

MIIKKA VILANDER

# Succinctness and Formula Size Games



MIIKKA VILANDER

# Succinctness and Formula Size Games

ACADEMIC DISSERTATION

To be presented, with the permission of  
the Faculty of Engineering and Natural Sciences  
of Tampere University,  
for public discussion in the auditorium B1096  
of the Pinni B building, Kanslerinrinne 1, Tampere,  
on 23 September 2022, at 12 o'clock.

## ACADEMIC DISSERTATION

Tampere University, Faculty of Engineering and Natural Sciences  
Finland

<i>Responsible supervisor</i>	Professor Esko Turunen Tampere University Finland	
<i>Supervisor</i>	Professor Lauri Hella Tampere University Finland	
<i>Pre-examiners</i>	Professor Hans van Ditmarsch Open University of the Netherlands Netherlands	Professor Martin Grohe RWTH Aachen University Germany
<i>Opponent</i>	Professor Juha Kontinen University of Helsinki Finland	
<i>Custos</i>	Professor Lauri Hella Tampere University Finland	

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Copyright ©2022 Miikka Vilander

Cover design: Roihu Inc.

ISBN 978-952-03-2540-4 (print)

ISBN 978-952-03-2541-1 (pdf)

ISSN 2489-9860 (print)

ISSN 2490-0028 (pdf)

<http://urn.fi/URN:ISBN:978-952-03-2541-1>



Carbon dioxide emissions from printing Tampere University dissertations have been compensated.

PunaMusta Oy – Yliopistopaino  
Joensuu 2022

# ACKNOWLEDGEMENTS

First and foremost I would like to thank my supervisor Lauri Hella. You have been an inspiring supervisor, a skilled co-author and a kind friend. This thesis would not have been possible without you.

I would also like to thank my other co-author Martin Lück and his thesis supervisor at the time Heribert Vollmer. Thank you for the opportunity to visit Hannover and work with you.

I would like to thank the pre-examiners Hans van Ditmarsch and Martin Grohe for your careful reading of this thesis and your kind comments on it.

Thank you to my colleagues in both campuses of Tampere University for a welcoming and inspiring atmosphere. Whether the topic was mathematics or the weather, I always had someone to talk to at work.

Thank you to my family for always being there for me. Thank you to my friends for all the good times and support. Thanks to all of you, even if work on the thesis was hard, life never was.

And finally thank you to my wife Anni for everything.

This thesis was financially supported by the Doctoral Programme in Engineering and Natural Sciences at TUT and later TAU as well as the Academy of Finland project *Explaining AI with logic* (XAILOG), project number 345612.

Tampere, September 2022

Miikka Vilander



# ABSTRACT

This thesis studies the succinctness of various logics using formula size games. The succinctness of a logic refers to the size of formulas required to express properties. Formula size games are some of the most successful methods of proof for results on succinctness. The contribution of the thesis is twofold. Firstly, we define formula size games for several logics, providing methods for future research. Secondly, we use these games and other methods to prove results on the succinctness of the studied logics.

More precisely, we develop new parameterized formula size games for basic modal logic, modal  $\mu$ -calculus, propositional team logic and generalized regular expressions. For the generalized regular expression game we introduce variants that correspond to regular expressions and the newly defined RE over star-free expressions, where stars do not occur inside complements.

We use the games to prove a number of succinctness results. We show that first-order logic is non-elementarily more succinct than both basic modal logic and modal  $\mu$ -calculus. We conduct a systematic study of the succinctness of defining common atoms of dependency in propositional team logic. We reprove a classic non-elementary succinctness gap between first-order logic and regular expressions in a much simpler way and establish a hierarchy of expressive power for the number of stars in RE over star-free expressions.

Many of the above results utilize explicit formulas in addition to game arguments. We use such formulas and a type counting argument to obtain non-elementary lower and upper bounds for the succinctness of defining single words in first-order logic and monadic second-order logic.





# TIIVISTELMÄ

Tämä väitöskirja tutkii erilaisten logiikoiden tiiviyttä kaavan pituuspelien avulla. Logiikan tiiviyys viittaa ominaisuuksien ilmaisemiseen tarvittavien kaavojen kokoon. Kaavan pituuspelit ovat hyväksi todettu menetelmä tiivistulosten todistamiseen. Väitöskirjan kontribuutio on kaksiosainen. Ensinnäkin väitöskirjassa määritellään kaavan pituuspelejä useille logiikoille ja tarjotaan näin uusia menetelmiä tulevaan tutkimukseen. Toiseksi näitä pelejä ja muita menetelmiä käytetään tiivistulosten todistamiseen tutkituille logiikoille.

Tarkemmin sanottuna väitöskirjassa määritellään uudet parametrisoidut kaavan pituuspelit perusmodaalilogiikalle, modaaliselle  $\mu$ -kalkyyllille, tiimilauselogiikalle ja yleistetyille säännöllisille lausekkeille. Yleistettyjen säännöllisten lausekkeiden pelistä esitellään myös variantit, jotka vastaavat säännöllisiä lausekkeitä ja uusia “RE over star-free” -lausekkeitä, joissa tähtiä ei esiinny komplementtien sisällä.

Pelejä käytetään useiden tiivistulosten todistamiseen. Predikaattilogiikan näytetään olevan epäelementaarisesti tiiviimpi kuin perusmodaalilogiikka ja modaalinen  $\mu$ -kalkyyli. Tiimilauselogiikassa tutkitaan systemaattisesti yleisten riippuvuuksia ilmaisevien atomien määrittämisen tiiviyttä. Klassinen epäelementaarinen tiiviysero predikaattilogiikan ja säännöllisten lausekkeiden välillä osoitetaan uudelleen yksinkertaisemmalla tavalla ja saadaan tähtien lukumäärälle “RE over star-free” -lausekkeissa hierarkia ilmaisuvoiman suhteen.

Monissa yllämainituista tuloksista hyödynnetään eksplisiittisiä kaavoja peliargumenttien lisäksi. Tällaisia kaavoja ja tyyppien laskemista hyödyntäen saadaan epäelementaarisia ala- ja ylärajoja yksittäisten sanojen määrittämisen tiiviydelle predikaattilogiikassa ja monadisessa toisen kertaluvun logiikassa.



# CONTENTS

1	Introduction . . . . .	1
1.1	Succinctness . . . . .	1
1.2	Formula size games . . . . .	3
1.3	Other methods . . . . .	6
1.4	Research objectives . . . . .	7
1.5	Structure of the dissertation . . . . .	8
2	Background . . . . .	9
2.1	First-order logic . . . . .	9
2.2	Modal logic and modal mu-calculus . . . . .	11
2.3	Team semantics . . . . .	14
2.4	Logics on words . . . . .	17
3	Results and discussion . . . . .	21
3.1	Modal logics . . . . .	21
3.2	Propositional team logics . . . . .	22
3.3	Logics on words . . . . .	24
4	Conclusions and future outlook. . . . .	27
4.1	Summary of results . . . . .	27
4.2	Future directions . . . . .	28
	References . . . . .	31
	Publication I . . . . .	39
	Publication II . . . . .	75
	Publication III . . . . .	105

Publication IV . . . . . 123

# LIST OF SYMBOLS AND ABBREVIATIONS

$D$	Delilah, the second player of formula size games
$D(\mathbb{A}, \mathbb{B})$	the density of the sets of teams $\mathbb{A}$ and $\mathbb{B}$
$\text{Dim}(\varphi)$	the upper dimension of the propositional team formula $\varphi$
$\text{DN}(L)$	the definability number of the fragment $L$
EF-game	Ehrenfeucht-Fraïssé game
FO	first-order logic
$\text{FO}_k$	the quantifier rank $k$ fragment of FO
$\text{FO}[n]$	the size $n$ fragment of FO
$\text{FS}_k^\Phi(\mathbb{A}, \mathbb{B})$	the formula size games for ML and propositional team logic
$\mu\text{-FS}_k^\Phi(\mathbb{A}, \mathbb{B})$	the formula size game for modal $\mu$ -calculus
GFP	greatest fixed point
GRE	generalized regular expressions with complement
$\text{GRES}(k, s, A, B)$	the GRE size game
$\text{H}(L)$	the Hanf number of the fragment $L$
LFP	least fixed point
$L_\mu$	modal $\mu$ -calculus
$\log^*$	the iterated logarithm function
$\text{LS}(L)$	the Löwenheim-Skolem number of the fragment $L$
ML	basic modal logic
$\text{ML}^2$	two-dimensional modal logic
MSO	monadic second-order logic

$\text{MSO}_k$	the quantifier rank $k$ fragment of MSO
$\text{MSO}[n]$	the size $n$ fragment of MSO
$\text{PL}(\Sigma)$	propositional team logic with the connectives in $\Sigma$
$\text{PL}(\{\wedge, \otimes, \vee, \dot{\vee}\})$	the existential fragment of propositional team logic
$\text{qr}(\varphi)$	the quantifier rank of the formula $\varphi$
RE	regular expressions
$\text{RES}(k, s, A, B)$	the RE size game
$\text{RESFS}(k, s, A, B)$	the RE over star-free expression size game
S	Samson, the first player of formula size games
$\text{sz}(\varphi)$	the size of the formula $\varphi$
$\text{twr}$	the exponential tower function
$V_n$	level $n$ of the cumulative hierarchy of finite sets
$\models$	the truth relation between models and formulas
$\vee$	disjunction connective, ‘or’, in propositional team logic the lax splitting disjunction
$\wedge$	conjunction connective, ‘and’
$\neg$	negation connective, ‘not’, in propositional team logic the dual negation, in RE the complement operator
$\top$	logical constant verum, always true
$\perp$	logical constant falsum, always false
$\exists$	the existential quantifier, ‘there is’
$\forall$	the universal quantifier, ‘for all’
$\diamond$	the existential diamond operator of modal logic
$\square$	the universal box operator of modal logic
$\mu X$	the least fixed point operator in modal $\mu$ -calculus
$\nu X$	the greatest fixed point operator in modal $\mu$ -calculus
$\otimes$	the intuitionistic or Boolean disjunction in propositional team logic

$\dot{\vee}$	the strict splitting disjunction in propositional team logic
$\otimes$	the lax splitting conjunction in propositional team logic
$\dot{\otimes}$	the strict splitting conjunction in propositional team logic
$\sim$	the contradictory or Boolean negation in propositional team logic
$=(\vec{\alpha}; \vec{\beta})$	a dependence atom
$\vec{\alpha} \perp \vec{\beta}$	an independence atom
$\vec{\alpha} \perp_{\vec{\beta}} \vec{\gamma}$	a conditional independence atom
$\vec{\alpha} \subseteq \vec{\beta}$	an inclusion atom
$\vec{\alpha} \mid \vec{\beta}$	an exclusion atom
$\vec{\alpha} \Upsilon \vec{\beta}$	an anonymity atom
$.^*$	the Kleene star
$\chi(\mathcal{G})$	the colouring number of the graph $\mathcal{G}$





## ORIGINAL PUBLICATIONS

- Publication I      Lauri Hella and Miikka Vilander. “Formula size games for modal logic and  $\mu$ -calculus”. In: *J. Log. Comput.* 29.8 (2019), pp. 1311–1344. DOI: 10.1093/logcom/exz025.
- Publication II     Martin Lück and Miikka Vilander. “On the Succinctness of Atoms of Dependency”. In: *Log. Methods Comput. Sci.* 15.3 (2019). DOI: 10.23638/LMCS-15(3:17)2019.
- Publication III    Miikka Vilander. “Games for Succinctness of Regular Expressions”. In: *Proceedings 12th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2021, Padua, Italy, 20-22 September 2021*. Ed. by Pierre Ganty and Davide Bresolin. Vol. 346. EPTCS. 2021, pp. 258–272. DOI: 10.4204/EPTCS.346.17.
- Publication IV    Lauri Hella and Miikka Vilander. “Defining Long Words Succinctly in FO and MSO”. In: *Revolutions and Revelations in Computability - 18th Conference on Computability in Europe, CiE 2022, Swansea, UK, July 11-15, 2022, Proceedings*. Ed. by Ulrich Berger et al. Vol. 13359. Lecture Notes in Computer Science. Springer, 2022, pp. 125–138. DOI: 10.1007/978-3-031-08740-0\_11.

### *Author's contribution*

- Publication I      Miikka Vilander wrote the entire article with the exception of the introduction, which was written by Lauri Hella. The results and proofs were jointly devised by both authors based on initial ideas of Lauri Hella.
- Publication II      The ideas for the results and proofs were devised jointly by both authors. Miikka Vilander wrote parts of the Introduction and Preliminaries, as well as Subsections 3.1 and 3.2 having to do with the formula size game. Martin Lück wrote the rest of the paper with constant feedback from Miikka Vilander.
- Publication III      Miikka Vilander wrote the paper with some comments and feedback from Lauri Hella.
- Publication IV      The ideas for the results and proofs were born in joint discussions. Lauri Hella wrote Section 3 with the upper bounds and Miikka Vilander wrote Sections 4 and 5, including the explicit formulas. The rest of the sections were written collaboratively.

# 1 INTRODUCTION

Mathematical logic is a field of mathematics that concerns formal languages called logics. These differ from natural language in that they have very strict syntax and semantics. Indeed logics are more akin to programming languages than natural language. A string of symbols formed correctly using the syntax of a logic is called a formula. On the semantic side we have models of the logic. These are the objects that give meaning to formulas of the logic. A given formula is either true or false in a given model. If we gather all of the models that satisfy a formula, this class of models is the property defined by the formula. We can then consider what kind of properties the logic can define. This is called expressive power and has been the subject of extensive research. Taking things one step further, one can consider the size of the formulas that define these properties. If a property can be defined in two different logics, there can be substantial differences between the sizes of formulas required. This phenomenon is often referred to as succinctness and is the topic of this dissertation.

## 1.1 Succinctness

The term succinctness refers to the size of formulas required to express properties. Formula size could in principle be defined as the number of symbols in the formula as a string. In practice, however, symbols deemed inconsequential such as parentheses are often omitted from the definition, ending up at something more akin to the number of nodes in the syntax tree of the formula. The relative succinctness of two logics is often discussed in the following way. We say that a logic  $\mathcal{L}_1$  is, for example, exponentially more succinct than another logic  $\mathcal{L}_2$  if there is a sequence of  $\mathcal{L}_1$ -formulas  $(\varphi_n)_{n \in \mathbb{N}}$  such that for any sequence  $(\psi_n)_{n \in \mathbb{N}}$  of equivalent  $\mathcal{L}_2$ -formulas, the size of each  $\psi_n$  is at least exponential in the size of  $\varphi_n$ . This can of course be formulated for any function  $f$  but the most commonly discussed cases are (double)

exponential and non-elementary gaps in succinctness. Here non-elementary means that the function  $f$  grows faster than any exponential tower of constant height. Polynomial differences in succinctness between logics are usually not considered relevant results.

Note that this definition of succinctness leaves open the possibility that two logics  $\mathcal{L}_1$  and  $\mathcal{L}_2$  can both be, say, exponentially more succinct than the other. This is intentional, as in some cases logics can be, perhaps by design, very apt at defining some types of properties succinctly while requiring large formulas for others. An example of this is given by [26], where Hoek et al. show that two extensions of multimodal logic are both exponentially more succinct than each other.

Succinctness can be seen as a more fine grained version of expressive power as logics with the exact same expressive power can be very different in terms of succinctness. In the context of comparing formalism for knowledge representation, Gogic et al. [14] argue that succinctness is “a much more interesting, but also more subtle question one can ask about a knowledge representation formalism” compared to expressive power.

Aside from independent interest, succinctness also has connections to complexity. Often, if a logic can express properties very succinctly, this comes at the cost of higher complexity of satisfiability or model checking. An example of this is given by two-variable first-order logic  $\text{FO}^2$  and a weak version of temporal logic called unary-TL. Etesami et al. proved in [9] that these two logics have the same expressive power over  $\omega$ -words, but  $\text{FO}^2$  is exponentially more succinct than unary-TL. Accordingly, the complexity of satisfiability for  $\text{FO}^2$  is NEXPTIME-complete, while the complexity of unary-TL is in NP [33]. However, more succinctness does not always imply higher complexity. For example, public announcement logic PAL is exponentially more succinct than epistemic logic EL, but the complexity of satisfiability is the same for both of them [30].

We would also argue that succinctness has connection to a notion of natural expression. Formulas with immense size can hardly be considered natural or understandable to a human reader as even reading them would take an unreasonable amount of time. Thus we could say that an extent of succinctness is necessary for a formula to be considered natural. In finite cases these immense formulas can essentially be just lists of the models that satisfy the property. The exponential formulas given for atoms of dependency in Publication II have this flavor as they utilize lists

of all propositional valuations in  $n$  variables. One could argue that if a property is very easy to convey in natural language, then listing the models is not a very natural way to express that property.

On the other hand, logics with high complexity can have very succinct formulas that are nevertheless difficult to understand. We would consider any property with more than one necessary fixed point alternation in modal  $\mu$ -calculus hard to understand. These kinds of definitions could also be considered unnatural so the relationship between succinctness and natural expression is not straightforward.

## 1.2 Formula size games

Succinctness is clearly an important area of research but it is also a challenging one. In [5] Buchfuhrer and Umans show that already in propositional logic, the problem of finding the smallest Boolean formula equivalent to a given formula is complete for the second level of the polynomial hierarchy under Turing reductions. It is reasonable to expect difficulties with the succinctness of more complex logics. Some of the foremost methods used to tackle the difficulty of succinctness are formula size games.

The lineage of formula size games can be traced back to a 1981 paper by Immerman [28]. He defined a so called separability game that characterized the number of quantifiers in formulas. Even though this is not the same as the size of the formula, it is markedly closer to it than the more traditionally considered quantifier depth. The first full formula size game for propositional logic was defined in 1990 by Razborov in [32]. Both the games of Immerman and Razborov seem to have gone largely unnoticed by the logic community and in 2003 Adler and Immerman defined their formula size game in [1] for first-order logic with a transitive closure operator and for a temporal logic called  $CTL^+$ . This is the version cited by many later works as inspiration for their games.

Formula size games are somewhat akin to the Ehrenfeucht-Fraïssé games, or EF-games, widely used in finite model theory. For more on EF-games see e.g. [7]. Both games have two players we refer to as S and D. In EF-games these players have the names Spoiler and Duplicator, reflecting the roles of the players in EF-games. For formula size games these names seem somewhat inaccurate so we instead use the more neutral Samson and Delilah and refer to them as he and she, respectively. While EF-

games characterize the quantifier depth of first-order formulas required to separate two structures, formula size games instead characterize the size of formulas required to separate two sets of structures. The main theorem of any formula size game has more or less the following form, formulated here as a parameterized version for an arbitrary logic  $\mathcal{L}$ .

**Theorem 1.** *Let  $k \in \mathbb{N}$  and let  $\mathcal{A}$  and  $\mathcal{B}$  be sets of  $\mathcal{L}$ -models. The following statements are equivalent.*

1. *S has a winning strategy for the formula size game  $FS(k, \mathcal{A}, \mathcal{B})$*
2. *There is an  $\mathcal{L}$ -formula  $\varphi$  with size at most  $k$  such that  $\mathcal{A} \models \varphi$  and  $\mathcal{B} \models \neg\varphi$ .*

The game starts with the position  $(k, \mathcal{A}, \mathcal{B})$ . As one can see from above the theorem, the goal of S is to show that  $\mathcal{A}$  and  $\mathcal{B}$  can be separated using a formula of size at most  $k$  and the goal of D is to refute this. The details of the game vary from logic to logic but the basic idea stays the same. Each connective, quantifier or other such operator in the logic has its own move. S builds the alleged separating formula starting from the outmost operator using the move corresponding to that operator. The rules of the move are based on the semantics of the operator and the sets  $\mathcal{A}$  and  $\mathcal{B}$  are modified to reflect this. D chooses which parts of the formula are constructed during the play and which are left vague.

We sketch an example of a disjunction move. S must split the left set  $\mathcal{A}$  into two parts  $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{A}$  with  $\mathcal{A}_1 \cup \mathcal{A}_2 = \mathcal{A}$  to reflect which models satisfy the left disjunct and which satisfy the right disjunct. The right set  $\mathcal{B}$  is duplicated to both following positions as the negation of a disjunction is a conjunction. S must also split the parameter  $k$  between the two positions and spend one of it so that  $k_1 + k_2 + 1 = k$ . It is now the role of D to choose which of the following positions  $k_1, \mathcal{A}_1, \mathcal{B}$  or  $k_2, \mathcal{A}_2, \mathcal{B}$  she thinks cannot be separated with the remaining value of  $k_i$ .

In the game of Adler and Immerman [1] there is no parameter  $k$ . Instead S actually builds the entire syntax tree of the formula and the size is checked after the game. This game could be defined as only having a single player as the only role of D is to play an easily defined optimal strategy. To see this, consider first that since the entire syntax tree of the formula will be constructed by the end of the game, the choice of disjunct in the previous example is only a matter of order and thus inconsequential. There are other moves where D gets to choose a subset of some set of models to

include in the following position, but this is also trivialized by the observation that adding more models can never be detrimental to  $D$ . Thus the optimal strategy of  $D$  is to always choose the entire permitted set of models to include in the game.

By contrast, the game of Razborov [32] as well as the games of Hella and Väänänen [21] are both parameterized versions, where a parameter  $k$  is given beforehand and this limits the size of the formula  $S$  can build. One can think of  $k$  as a resource at the disposal of  $S$ . In the case of a binary connective such as disjunction,  $D$  now has the important choice of choosing the disjunct the game proceeds from. For most logics, the game never returns to the unchosen disjunct during the play. So the parameter  $k$  both gives  $D$  a meaningful role as the second player and shortens the plays as only part of the formula is constructed during a single play. The optimal choice of the entire set of models in some moves is not affected by the parameterization, but in our games we remove the semblance of choice by fixing in the rules that the entire set is always used.

Most of the applications of formula size games are in the field of modal logics. Such games have been defined for a multitude of modal logics including epistemic logic [11], modal logic with contingency operators [6] and multimodal logics with union, intersection and quantification [25], to name just a few. Recently Balbiani et al. [2] even defined games of this nature for formulas defining properties of Kripke frames.

A different interpretation of the idea of formula size games was formulated by Grohe and Schweikardt [17] as extended syntax trees. This is a method inspired by the games of Adler and Immerman but with the game aspect removed in favor of a static object. As the Adler-Immerman game involved always constructing the entire syntax tree of the separating formula, it is natural to consider this final tree. In terms of our parameterized games, an extended syntax tree is closest to the entire winning strategy of  $S$ . In practice, an extended syntax tree consists of the syntax tree of a formula with a left and right set of models appended to each node. Proofs argue about the form of the tree and models included in the sets to arrive at a conclusion about the size or existence of the tree. In [17], extended syntax trees were used to prove that the four-variable fragment of FO is exponentially more succinct than the three-variable fragment. In addition van der Hoek et. al [26] used this method to show that two extensions of multimodal logic are both exponentially more succinct than each other.

### 1.3 Other methods

Formula size games are by no means the only method that has been used to study succinctness. Many other methods can be found in the literature.

One broad category of methods that can be identified is automata theoretic methods. Here the actual arguments pertaining to size are done in the context of the automata that correspond to the logic under study. The results can then be transferred to the logic using known translations. For an easy example, the folklore result found in [8] that states the most succinct regular expression, or RE, that defines a single word is the word itself, can easily be proven using finite automata. For more involved examples, see the proof of Wilke [37] of the exponential succinctness gap between the modal logics  $CTL^+$  and  $CTL$ , or the proof of Etesami et al. [9] of the exponential succinctness of  $FO^2$  over unary-TL.

Another way to approach succinctness is through complexity. An example of this is given by the work of Stockmeyer [34]. He is mainly concerned with complexity and shows that the satisfiability of FO over words is of non-elementary complexity. Etesami et al. note in [9] that from Stockmeyer's proof one can find FO formulas with linear size and only models with non-elementary size. Add to this the fact that all satisfiable formulas of linear temporal logic LTL have a satisfying model at most exponential in the size of the formula, and one obtains a non-elementary succinctness gap between FO and LTL.

For a very coarse lower bound for succinctness one can study quantifier depth. An example of this is found in [17] where Grohe and Schweikardt prove that MSO is non-elementarily more succinct than FO and the non-elementary size comes already from the quantifier depth of the FO formulas. Related to quantifier depth, the star height of regular expressions is more useful in terms of succinctness. In [18] Gruber and Holzer show that the number of alphabet symbols in an RE is exponential in its star height. They use this fact to show exponential lower bounds for an RE defining the intersection or shuffle of two RE as well as a double exponential lower bound for defining the complement of an RE.

In addition to the rough categorization we have done here, there are other more tailor-made methods for specific contexts. An example of this is the upper dimension of a modal team logic formula  $\varphi$  by Hella et al. in [22], defined as the number of maximal teams satisfying  $\varphi$ . They show that upper dimension is well-behaved with



respect to formula size and use it to show an exponential gap in succinctness between extended modal dependence logic and modal team logic with boolean disjunction. Another example is encoding a specific hard property via regular expressions with different added operators like Gelade and Neven in [13]. They generalize a theorem by Ehrenfeucht and Zeiger to obtain a single hard sequence of languages  $Z_n$ . They proceed to show that both complement and intersection can be used to define this sequence in a succinct way, giving rise to double exponential succinctness gaps in relation to regular expressions without added operations.

## 1.4 Research objectives

This thesis has two objectives. The first is to define formula size games for different logics. In particular, we focus on logics for which no formula size games have been considered before. The games establish new methods for further study of succinctness. The second goal is to study the succinctness of these logics in relation to others in the same field. What kinds of gaps can be found? In addition to furthering our understanding of succinctness, this demonstrates how and why formula size games are used as proof methods.

Publication I focuses on games for modal logics and comparing them to first-order logic. A particular goal of this research is to define a formula size game for modal  $\mu$ -calculus and find applications. Publication II is the first systematic study of succinctness in the team semantics setting. Instead of comparing propositional team logic to very different logics we instead ask: how succinctly can the commonly used atoms of dependency be defined in propositional team logic? We also define the first formula size game in the team setting. Publication III considers generalized regular expressions. The goal is to define formula size games for different variants of regular expressions and use them to study the interesting middle ground of succinctness between generalized regular expressions and regular expressions. Publication IV focuses on definitions of single words. In particular, the question is: what is the length of the longest word definable via a formula of size  $n$  in FO or MSO?

## 1.5 Structure of the dissertation

This thesis consists of four publications found at the end and a preceding summary. In Chapter 2 we introduce the logics considered in the thesis and discuss some known succinctness results for those logics. Section 2.1 briefly introduces and discusses first-order logic. Section 2.2 concerns basic modal logic and modal  $\mu$ -calculus. Section 2.3 discusses team semantics and finally Section 2.4 is about regular expressions and MSO on words. In Chapter 3 we present and discuss the results obtained in the publications. Section 3.1 presents the results of Publication I concerning modal logics. Section 3.2 discusses Publication II and propositional team logic. Section 3.3 presents the results of Publications III and IV having to do with logics on words. Chapter 4 concludes the summary with a recap of the results in Section 4.1 and directions for future research in Section 4.2.

## 2 BACKGROUND

In this chapter we briefly present the logics considered in the publications. We define the syntax and semantics of each logic and discuss results found in the literature about the succinctness of these logics.

### 2.1 First-order logic

First-order logic, or FO, is one of the most influential logics in existence, so we will not spend much time on it. We only present syntax and semantics to fix notation. For further reading FO from a finite model theory perspective we direct the reader to [7]. We only define FO for relational vocabularies as is common in the field of finite model theory. This makes some proofs simpler and we can always replace an  $n$ -ary function symbol with an  $n + 1$ -ary relation symbol with the functionality required in the formula. Although this does have a potential effect on succinctness, it is of no consequence to our results in this thesis.

**Definition 1.** Let  $\sigma$  be a relational vocabulary and let  $\text{Var}$  be an infinite set of variable symbols. The set  $\text{FO}(\sigma)$  of first-order formulas in the vocabulary  $\sigma$  is the smallest set satisfying the following conditions:

- If  $x, y \in \text{Var}$ , then  $x = y$  is a formula.
- If  $R \in \sigma$  and  $R$  is  $n$ -ary and  $x_1, \dots, x_n \in X$ , then  $R(x_1, \dots, x_n)$  is a formula.
- If  $\varphi$  and  $\psi$  are formulas and  $x \in \text{Var}$ , then  $\varphi \vee \psi$ ,  $\varphi \wedge \psi$ ,  $\neg\varphi$ ,  $\exists x\varphi$  and  $\forall x\varphi$  are formulas.

The semantics of FO involve relational structures as models and assignments of points in those models to variables. A relational structure is a tuple  $(M, (R_i)_{i \in I})$ , where  $M$  is a set and each  $R_i$  is a relation on  $M$ . When we need to differentiate a relation in a model from its relation symbol we use the notation  $R^M$  for the actual relation.

An occurrence of a variable  $x$  in a formula  $\varphi$  is bound if it is inside a quantifier  $\exists x$  or  $\forall x$ . If an occurrence of  $x$  is not bound it is free. The variable  $x$  is free in  $\varphi$  if it has a free occurrence. The set of free variables of  $\varphi$  is denoted  $\text{Fr}(\varphi)$ . An assignment is a function  $s : Y \rightarrow M$ , where  $Y \subseteq \text{Var}$ . A modified assignment  $s[a/x]$  assigns  $s[a/x](x) = a$  and is otherwise the same as  $s$ .

We define the truth of formulas in relation to relational structures and variable assignments. We use the symbol  $\models$  to denote that the model and assignment on the left side of the symbol satisfy the formula on the right. We use the same symbol for other logics below.

**Definition 2.** Let  $\mathcal{M}$  be a relational  $\sigma$ -structure, let  $\varphi \in \text{FO}(\sigma)$  and let  $s : Y \rightarrow M$  be an assignment, where  $\text{Fr}(\varphi) \subseteq Y$ . The truth relation of FO is defined as follows:

- $\mathcal{M}, s \models x = y$  iff  $s(x) = s(y)$ .
- $\mathcal{M}, s \models R(x_1, \dots, x_n)$  iff  $(s(x_1), \dots, s(x_n)) \in R^{\mathcal{M}}$ .
- $\mathcal{M}, s \models \psi \vee \theta$  iff  $\mathcal{M}, s \models \psi$  or  $\mathcal{M}, s \models \theta$ .
- $\mathcal{M}, s \models \psi \wedge \theta$  iff  $\mathcal{M}, s \models \psi$  and  $\mathcal{M}, s \models \theta$ .
- $\mathcal{M}, s \models \neg\psi$  iff  $\mathcal{M}, s \not\models \psi$ .
- $\mathcal{M}, s \models \exists x\psi$  iff there is  $a \in M$  such that  $\mathcal{M}, s[a/x] \models \psi$ .
- $\mathcal{M}, s \models \forall x\psi$  iff for all  $a \in M$ , it holds that  $\mathcal{M}, s[a/x] \models \psi$ .

We also use the implication  $\rightarrow$  and equivalence  $\leftrightarrow$  connectives in our formulas, but we consider these shorthand in the following way:

$$\varphi \rightarrow \psi := \neg\varphi \vee \psi \text{ and } \varphi \leftrightarrow \psi := \varphi \rightarrow \psi \wedge \psi \rightarrow \varphi.$$

As widely studied as FO is, it is still hard to find many results on its succinctness. It could be that there are numerous such results scattered in the literature under different terminology. We cite some of the more known examples. In [17] Grohe and Schweikardt study the succinctness of FO with a bounded number  $n$  of variables, denoted  $\text{FO}^n$ , on linear orders. They show that the succinctness gap from  $\text{FO}^2$  to  $\text{FO}^3$  is only polynomial, but  $\text{FO}^4$  is exponentially more succinct than  $\text{FO}^3$ . They also show that MSO is non-elementarily more succinct than FO on linear orders.

The work of Stockmeyer [34], though originally quite focused on complexity, implies a non-elementary succinctness gap between FO and linear temporal logic.

This is pointed out in [9], where Etessami et al. prove that  $\text{FO}^2$  is exponentially more succinct than unary-TL.

## 2.2 Modal logic and modal mu-calculus

Modal logic is a paradigm of logic that is central in many areas of theoretical computer science. There are a plethora of logics classified as modal logics such as temporal and epistemic logics. In this thesis we consider the basic modal logic ML, also known as  $\mathbf{K}$ , and modal  $\mu$ -calculus  $L_\mu$ . For further reading on modal logic we direct the reader to [3]. For more on modal  $\mu$ -calculus we cite [4].

**Definition 3.** Let  $\text{Prop}$  be a set of proposition symbols. The set  $\text{ML}(\text{Prop})$  of formulas of basic modal logic for propositions  $\text{Prop}$  is the smallest set that satisfies the following conditions:

- The constants  $\top$  and  $\perp$  are formulas.
- Every  $p \in \text{Prop}$  is a formula.
- If  $\varphi$  and  $\psi$  are formulas, then  $\varphi \vee \psi$ ,  $\varphi \wedge \psi$ ,  $\neg\varphi$ ,  $\diamond\varphi$  and  $\square\varphi$  are formulas.

The semantics of modal logic are based on Kripke models. A Kripke model is a triple  $\mathcal{M} = (W, R, V)$ , where  $W$  is a set of points or worlds,  $R$  is a binary relation on  $M$  and  $V : \text{Prop} \rightarrow \mathcal{P}(W)$  is a valuation function that assigns to each proposition the set of points that satisfy the proposition. The semantics of ML are identical to propositional logic besides the two new symbols  $\diamond$  and  $\square$ . These have to do with the binary relation  $R$ .

**Definition 4.** Let  $\mathcal{M}$  be a Kripke-model and  $w \in W$ . The truth relation between pointed Kripke models and formulas  $\varphi \in \text{ML}$  is defined as follows:

- $(\mathcal{M}, w) \models \top$  and  $(\mathcal{M}, w) \not\models \perp$ .
- $(\mathcal{M}, w) \models p$  iff  $w \in V(p)$  for every  $p \in \text{Prop}$ .
- The Boolean connectives are defined in the usual way, e.g.  
 $(\mathcal{M}, w) \models \psi \vee \theta$  iff  $(\mathcal{M}, w) \models \psi$  or  $(\mathcal{M}, w) \models \theta$ .
- $(\mathcal{M}, w) \models \diamond\psi$  iff there is  $v \in W$  such that  $(w, v) \in R$  and  $(\mathcal{M}, v) \models \psi$ .
- $(\mathcal{M}, w) \models \square\psi$  iff for all  $v \in W$  such that  $(w, v) \in R$  it holds that  $(\mathcal{M}, v) \models \psi$ .

The modal  $\mu$ -calculus is a more complex extension of ML. The syntax adds new operators  $\mu X$  and  $\nu X$  along with new variables  $X$ .

**Definition 5.** Let  $\text{Prop}$  be a set of proposition symbols and  $Q$  a set of variables. The set  $L_\mu(\text{Prop})$  of formulas of modal  $\mu$ -calculus for propositions  $\text{Prop}$  is the smallest set that satisfies the following conditions:

- The constants  $\top$  and  $\perp$  are formulas.
- If  $p \in \text{Prop}$ , then  $p$  and  $\neg p$  are formulas.
- Every  $X \in Q$  is a formula.
- If  $\varphi$  and  $\psi$  are formulas and  $X \in Q$  is a variable, then  $\varphi \vee \psi$ ,  $\varphi \wedge \psi$ ,  $\diamond \varphi$ ,  $\square \varphi$ ,  $\mu X.\varphi$  and  $\nu X.\varphi$  are formulas.

Note that we assume all formulas are in negation normal form, where negation only occurs on the level of literals. This is an easy way to ensure that the variables of the fixed points only occur positively, which is required for the semantics.

The compositional semantics of  $L_\mu$  are still based on Kripke models but also involve fixed point operators in connection with the new symbols  $\mu$  and  $\nu$ . It is convenient to present the semantics in terms of truth sets instead of a truth relation. The truth set of a formula is the set of points that satisfy the formula. Note that we leave the model implicit in the notation as it remains constant throughout the definition of the semantics.

$$\|\varphi\|_\rho = \{w \in W \mid \mathcal{M}, w \models_\rho \varphi\}.$$

Let  $\Gamma : \mathcal{P}(W) \rightarrow \mathcal{P}(W)$  be a function. A set  $A$  such that  $\Gamma(A) = A$  is called a fixed point of  $\Gamma$ . The famous Knaster-Tarski theorem states that if  $\Gamma$  is monotone, then  $\Gamma$  has a least fixed point  $\text{LFP}(\Gamma)$  and a greatest fixed point  $\text{GFP}(\Gamma)$  with respect to the subset relation. We associate such a function  $\Gamma$  to each formula  $\varphi$  and valuation of variables  $\rho$  as follows.

$$\Gamma_{\varphi, X, \rho} : \mathcal{P}(W) \rightarrow \mathcal{P}(W), \quad \Gamma_{\varphi, \rho}(A) = \|\varphi\|_{\rho[A/X]}.$$

Since all formulas are in negation normal form, all variables  $X$  occur within their fixed points  $\mu X$  or  $\nu X$  only positively. This ensures that the function  $\Gamma_{\varphi, X, \rho}$  is always monotone and  $\text{LFP}(\Gamma)$  and  $\text{GFP}(\Gamma)$  exist. We now define the compositional semantics of  $L_\mu$ .

**Definition 6.** Let  $(M, w)$  be a pointed Kripke model and let  $\rho$  be a valuation of variables. The truth set  $\|\varphi\|_\rho$  of a formula  $\varphi \in L_\mu$  under the valuation  $\rho$  is defined as follows:

- $\|p\|_\rho = V(p)$  for all  $p \in \text{Prop}$ .
- $\|X\|_\rho = \rho(X)$  for all  $X \in Q$ .
- $\|\psi \vee \theta\|_\rho = \|\psi\|_\rho \cup \|\theta\|_\rho$ .
- $\|\psi \wedge \theta\|_\rho = \|\psi\|_\rho \cap \|\theta\|_\rho$ .
- $\|\diamond\psi\|_\rho = \{w \in W \mid \text{there is } v \in W \text{ s. t. } (w, v) \in R \text{ and } v \in \|\psi\|_\rho\}$ .
- $\|\Box\psi\|_\rho = \{w \in W \mid \text{for all } v \in W \text{ s. t. } (w, v) \in R \text{ it holds that } v \in \|\psi\|_\rho\}$ .
- $\|\mu X.\psi\|_\rho = \text{LFP}(\Gamma_{\psi, X, \rho})$ .
- $\|\nu X.\psi\|_\rho = \text{GFP}(\Gamma_{\psi, X, \rho})$ .

The above compositional semantics are widely considered quite unintuitive so we informally sketch some intuitions about  $L_\mu$ -formulas in terms of game semantics. In the game semantics the variables  $X$  are essentially references back to the formula starting from  $\mu X$  or  $\nu X$ . The difference between  $\mu$  and  $\nu$  is how this looping affects the truth of formulas. The least fixed point operator  $\mu X$  can be seen as a reachability operator. In the semantic game the verifier is responsible for the loop of  $\mu X$  eventually reaching its destination and not looping anymore. In a dual fashion,  $\nu X$  corresponds to a safety condition. It is the responsibility of the falsifier to reach the point which breaks the safety condition and stops the loop. Straightforward game semantics based on these interpretations potentially have infinite plays as a safety condition naively takes an infinite time to verify. However, there is a way to make plays finite using ordinal clocks as in [19]. Even with these intuitions about reachability and safety, more complex  $L_\mu$ -formulas with alternation between  $\mu$  and  $\nu$  can be very hard to interpret intuitively.

There is quite a volume of research on succinctness in modal logic relative to other fields. The following overview is not an attempt at a complete listing.

The seminal paper of Adler and Immerman [1] includes a proof of a succinctness gap of order  $n!$  between modal logics called  $\text{CTL}^+$  and  $\text{CTL}$ , sharpening an earlier exponential gap result by Wilke [37].

In [11] French et al. consider multimodal logic extended with operators epistemically interpreted as ‘somebody knows’, ‘everybody knows’ and public announce-

ment. They show that all three of these extensions are exponentially more succinct than multimodal logic. In [26], van der Hoek et al. extend this by showing that the extensions with ‘somebody knows’ and ‘everybody knows’ are both exponentially more succinct than each other. In [25] van der Hoek and Iliev consider more operators. They prove seven different exponential gaps between multimodal logics with union, intersection, quantification and/or public announcement operators added.

Recently Balbiani et al. [2] introduced a novel version of formula size games characterizing the size of formulas that define properties of Kripke frames without valuations. They use these games to show that the formulas traditionally used to define frame properties such as transitivity or symmetry are indeed the smallest such formulas.

For the modal  $\mu$ -calculus, research on succinctness is more scarce. This perhaps has to do with the fact that even the correct notion of formula size seems still be a topic for discussion [29]. In [16] Grohe and Schweikardt show that monadic least fixed point logic with two variables MLFP<sup>2</sup> is exponentially more succinct on finite trees than modal  $\mu$  calculus with future and past modalities. This is only one of many succinctness results between monadic query languages on finite trees in the paper. Another example is found in [10], where Fernández-Duque and Iliev show that the spatial  $\mu$ -calculus is exponentially more succinct than equally expressive spatial logic with the tangled limit operator. They use a combination of formula size games for the basic modal connectives and some situational translations between the different logics to accomplish this.

## 2.3 Team semantics

Team semantics are a construct to discuss notions of dependency in logic. Dependencies can only be found in sets of data and so logics with team semantics define truth on a set of assignments or models instead of a single such object. These sets are called teams. Team semantics were originally suggested by Hodges [24] under the name ‘trump semantics’ as a semantic model for logics such as Hintikka’s independence friendly logic [23]. Another seminal work of team semantics is [36] by Väänänen, where he defines dependence logic as a new logic with team semantics. Since then, Väänänen and others have defined various logics with team semantics to discuss different notions of dependency. The new features were often added as atoms



to the logic as in the original dependence logic of Väänänen.

For this thesis, the relevant team logics are propositional, first studied by Yang and Väänänen [38]. Teams in this context are sets of propositional assignments. We need some notation for the semantics. If  $T$  is a propositional team, an ordered pair  $(S, U)$  of teams is a split of  $T$  if  $S, U \subseteq T$  and  $S \cup U = T$ . We say that a split  $(S, U)$  is strict if additionally  $S \cap U = \emptyset$ . We denote the set of splits of a team  $T$  by  $\text{Sp}(T)$  and the set of strict splits of  $T$  by  $\text{SSp}(T)$ .

We first define the semantics of all connectives we use and then define full propositional team logic and the existential fragment in terms of how these connectives are used.

**Definition 7.** Let  $\text{Prop}$  be a set of proposition symbols and let  $T$  be a Prop-team. The semantics of propositional team literals and connectives are defined as follows:

- $T \models \top$ .
- $T \models \perp$  iff  $T = \emptyset$ .
- $T \models p$  iff for all  $s \in T$ ,  $s(p) = 1$ .
- $T \models \neg p$  iff for all  $s \in T$ ,  $s(p) = 0$ .
- $T \models \sim\psi$  iff  $T \not\models \psi$ .
- $T \models \psi \wedge \theta$  iff  $T \models \psi$  and  $T \models \theta$ .
- $T \models \psi \otimes \theta$  iff  $T \models \psi$  or  $T \models \theta$ .
- $T \models \psi \vee \theta$  iff there is  $(S, U) \in \text{Sp}(T)$  s.t.  $S \models \psi$  and  $U \models \theta$ .
- $T \models \psi \dot{\vee} \theta$  iff there is  $(S, U) \in \text{SSp}(T)$  s.t.  $S \models \psi$  and  $U \models \theta$ .
- $T \models \psi \otimes \theta$  iff for all  $(S, U) \in \text{Sp}(T)$  it holds that  $S \models \psi$  or  $U \models \theta$ .
- $T \models \psi \dot{\otimes} \theta$  iff for all  $(S, U) \in \text{SSp}(T)$  it holds that  $S \models \psi$  or  $U \models \theta$ .

Using these connectives we define the two variations of propositional team logic considered in Publication II. Full propositional team logic is the logic with full unrestricted use of all above connectives. The existential fragment is the logic with full use of the connectives  $\wedge, \otimes, \vee$  and  $\dot{\vee}$  and with the contradictory negation  $\sim$  only occurring on the level of literals. With this limited use of the contradictory negation the existential fragment is expressively complete for propositional teams like full

propositional team logic, but still lacks the succinctness that full use of contradictory negation provides. Expressive completeness means that any set of propositional teams can be defined in either fragment.

Team logics can be augmented with many different atoms that express notions of dependency. We now present the propositional versions of the commonly used atoms we study in Publication II. Below  $\vec{\alpha}, \vec{\beta}$  and  $\vec{\gamma}$  are tuples of proposition symbols.

**Definition 8.** The semantics of propositional atoms of dependency are defined as follows:

- Dependence:  $=(\vec{\alpha}; \vec{\beta})$

$$T \models =(\vec{\alpha}; \vec{\beta}) \text{ iff } \forall s, s' \in T : s(\vec{\alpha}) = s'(\vec{\alpha}) \Rightarrow s(\vec{\beta}) = s'(\vec{\beta})$$

- Independence:  $\vec{\alpha} \perp \vec{\beta}$

$$T \models \vec{\alpha} \perp \vec{\beta} \text{ iff } \forall s, s' \in T : \exists s'' \in T : s(\vec{\alpha}) = s''(\vec{\alpha}) \text{ and } s'(\vec{\beta}) = s''(\vec{\beta})$$

- Conditional independence:  $\vec{\alpha} \perp_{\vec{\beta}} \vec{\gamma}$

$$T \models \vec{\alpha} \perp_{\vec{\beta}} \vec{\gamma} \text{ iff } \forall s, s' \in T : \text{if } s(\vec{\beta}) = s'(\vec{\beta}), \text{ then} \\ \exists s'' \in T : s(\vec{\alpha}\vec{\beta}) = s''(\vec{\alpha}\vec{\beta}) \text{ and } s'(\vec{\gamma}) = s''(\vec{\gamma})$$

- Inclusion:  $\vec{\alpha} \subseteq \vec{\beta}$ , where  $\vec{\alpha}$  and  $\vec{\beta}$  have equal length

$$T \models \vec{\alpha} \subseteq \vec{\beta} \text{ iff } \forall s \in T \exists s' \in T : s(\vec{\alpha}) = s'(\vec{\beta})$$

- Exclusion:  $\vec{\alpha} \mid \vec{\beta}$ , where  $\vec{\alpha}$  and  $\vec{\beta}$  have equal length

$$T \models \vec{\alpha} \mid \vec{\beta} \text{ iff } \forall s, s' \in T : s(\vec{\alpha}) \neq s'(\vec{\beta})$$

- Anonymity:  $\vec{\alpha} \Upsilon \vec{\beta}$

$$T \models \vec{\alpha} \Upsilon \vec{\beta} \text{ iff } \forall s \in T \exists s' \in T : s(\vec{\alpha}) = s'(\vec{\alpha}) \text{ and } s(\vec{\beta}) \neq s'(\vec{\beta})$$

We give some intuition for the formal definitions above. The dependence atom

$=(\vec{\alpha}; \vec{\beta})$ , originally introduced by Väänänen [36], states that the value of the tuple  $\vec{\beta}$  functionally depends on the value of the tuple  $\vec{\alpha}$ . The independence atom  $\vec{\alpha} \perp \vec{\beta}$ , introduced by Grädel and Väänänen [15], states that the values of the two tuples are completely independent, that is the team includes the complete Cartesian product of the two sets of values. The conditional independence atom  $\vec{\alpha} \perp_{\vec{\beta}} \vec{\gamma}$  states the same thing for the tuples  $\vec{\alpha}$  and  $\vec{\gamma}$  but separately within each value of the tuple  $\vec{\beta}$ . The inclusion atom  $\vec{\alpha} \subseteq \vec{\beta}$ , introduced by Galliani [12], states that the values of the tuple  $\vec{\alpha}$  are included in the values of the tuple  $\vec{\beta}$  in the team. The exclusion atom  $\vec{\alpha} \mid \vec{\beta}$ , introduced in the same Galliani paper, states that the values of the two tuples are completely separate. The anonymity atom  $\vec{\alpha} \Upsilon \vec{\beta}$ , due to Väänänen [35], states that no value of the tuple  $\vec{\alpha}$  determines functionally the value of the tuple  $\vec{\beta}$ . All of these atoms were originally introduced in the first-order setting. The propositional counterparts were first studied by Yang and Väänänen [38], except for the anonymity atom.

Unlike in other contexts, in the propositional case all atoms of dependency are trivially definable via other connectives due to the finite number of valuations in  $n$  propositional variables. Thus a natural question of succinctness arises: what is the size of these definitions? This question is answered systematically in Publication II.

As team semantics is a relatively new area of study, it is perhaps not surprising that succinctness has not received much attention. We are not aware of any studies dedicated to succinctness in the team setting but there are some results scattered among papers mainly concerned with expressive power.

In [22], Hella et al. study the expressive power of modal dependence logic. They use the notion of upper dimension to show that extended modal dependence logic is exponentially more succinct than modal logic with intuitionistic disjunction. In [20], Hella and Stumpf show that modal inclusion logic is exponentially more succinct than modal logic with the nonemptiness operator. They use the semantic game of the latter logic to show that  $2^n$  occurrences of nonemptiness are required to define an inclusion atom of arity  $n$ .

## 2.4 Logics on words

Regular languages and defining them are central topics in theoretical computer science. Perhaps the most canonical way to define regular languages is via regular ex-

pressions, or RE. Regular expressions are not usually considered a logic as such, but we do not differentiate them as they have all the trappings of a logic. They have a strict syntax and the language of an RE can be seen as the set of words that satisfy the expression as a formula. We also present generalized regular expressions, or GRE, where complement is added as an operation. This does not add any expressive power but is very significant in terms of succinctness. For further reading on regular languages and RE we refer the reader to [27].

**Definition 9.** Let  $\Sigma$  be an alphabet. The set  $\text{GRE}(\Sigma)$  of generalized regular expressions in the alphabet  $\Sigma$  is the smallest set that satisfies the following conditions:

- The symbols  $\emptyset$  and  $\epsilon$  are expressions.
- Every  $a \in \Sigma$  is an expression.
- If  $R_1$  and  $R_2$  are expressions, then  $R_1 \cup R_2$ ,  $R_1 R_2$ ,  $R_1^*$  and  $\neg R_1$  are expressions.

The semantics of GRE operate on words. A set of symbols  $\Sigma$  is called an alphabet and strings of those symbols are called words. The semantics of GRE are usually stated in the form of associating sets of words, called languages, with expressions. We follow this convention here.

**Definition 10.** Let  $\Sigma$  be an alphabet. The language  $L(R)$  of a generalized regular expression  $R$  of the alphabet  $\Sigma$  is defined as follows:

- $L(\emptyset) = \emptyset$
- $L(\epsilon) = \{\epsilon\}$ , where  $\epsilon$  is the empty word
- $L(a) = \{a\}$  for every  $a \in \Sigma$
- $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$
- $L(R_1 R_2) = L(R_1)L(R_2) = \{w_1 w_2 \mid w_1 \in R_1, w_2 \in R_2\}$
- $L(R_1^*) = L(R_1)^* = \{w_1 \cdots w_n \mid n \in \mathbb{N}, w_i \in L(R_1) \text{ for each } i \in \mathbb{N}\}$
- $L(\neg R_1) = \Sigma^* \setminus L(R_1)$

By restricting these definitions, we can easily define regular expressions and RE over star-free expressions. The set  $\text{RE}(\Sigma)$  of regular expressions of the alphabet  $\Sigma$  is defined as above but omitting the complement  $\neg$ . RE over star-free expressions are GREs, where no Kleene stars occur inside complement operations.

While the semantics of RE are defined to work on words as such, other logics often need to view words as relational structures to accommodate their semantics. This is the case for the logics we consider here, FO and monadic second-order logic MSO. In the following definition  $|w|$  denotes the length of the word  $w$ .

**Definition 11.** Let  $w \in \Sigma^*$ . The corresponding word model is the relational structure  $(M, \leq, (P_a)_{a \in \Sigma})$ , where  $M = \{1, \dots, |w|\}$ ,  $\leq$  is a linear order and each  $P_a$  is monadic and  $n \in P_a$  iff the  $n$ -th symbol of  $w$  is  $a$ .

Monadic second-order logic is a logic in the same vein as FO but with more freedom in quantification. In addition to quantifying single points like in FO, MSO can also quantify sets of points. For this MSO has set variables  $P$  from the set  $SVar$ . The syntax of MSO is the same as for FO, but with additional atomic formulas  $P(x)$ , where  $P \in SVar$  and  $x \in Var$ , and quantifiers  $\exists P$  and  $\forall P$ , where  $P \in SVar$ . The semantics add a separate assignment  $t : Z \rightarrow \mathcal{P}(M)$  for set variables, where  $Z \subseteq SVar$ . The set  $SFr(\varphi)$  of free set variables of the formula  $\varphi$  is defined in the same way as for first-order variables.

**Definition 12.** Let  $\mathcal{M}$  be a relational structure and let  $\varphi \in MSO(\Sigma)$ . Let  $s : Y \rightarrow M$  be a first-order assignment with  $Fr(\varphi) \subseteq Y$ . Let  $t : Z \rightarrow \mathcal{P}(M)$  be a set variable assignment with  $SFr(\varphi) \subseteq Z$ . The truth relation of MSO is defined as for FO with the following additions:

- $\mathcal{M}, s, t \models P(x)$  iff  $s(x) \in t(P)$ .
- $\mathcal{M}, s, t \models \exists P\psi$  iff there is  $A \subseteq M$  such that  $\mathcal{M}, s, t[A/P] \models \psi$ .
- $\mathcal{M}, s, t \models \forall P\psi$  iff for all  $A \subseteq M$  it holds that  $\mathcal{M}, s, t[A/P] \models \psi$ .

In the context of words, MSO is an important logic. The famous Büchi Theorem (see e.g. [27]) states that MSO and RE have the same expressive power on words, that is, MSO also captures regular languages. On the other hand, FO corresponds to star-free expressions, that is regular expressions with no Kleene stars in them. In Publication III we introduce a variant on RE motivated by this fact called RE over star-free expressions. These expressions combine star-free expressions with the full operations of RE, including stars. One can think of RE over star-free expressions as RE over FO definable properties. As RE over star-free expressions include RE and are included in GRE, they still capture regular languages but in terms of succinctness they present an interesting middle point between RE and GRE.

The succinctness gained by adding operators to RE has been studied independently and thoroughly by at least two groups of authors. On the one hand Gelade and Neven [13] and on the other Gruber and Holzer [18] both show exponential and double exponential gains in succinctness from adding operators such as complement, intersection, interleaving and more.

It is well known that MSO is non-elementarily more succinct than RE on words, but there are also some results on the succinctness of MSO in other contexts. Grohe and Schweikardt [16] study the relative succinctness of monadic query languages on finite trees. They show that in this context MSO is non-elementarily more succinct than monadic least fixed point logic MLFP, under some complexity theoretic assumptions. The logic MLFP in turn is non-elementarily more succinct than its two-variable fragment MLFP<sup>2</sup>. In [17] the same authors show that MSO is non-elementarily more succinct than FO on linear orders.

## 3 RESULTS AND DISCUSSION

The contribution of the thesis spans many different logics and is twofold. Firstly the formula size games defined in Publications I, II and III are of independent interest as methods to study the succinctness of the logics in question. They can be easily modified to study more restricted versions of logics or other measures like quantifier depth. An example of this is given in Publication III by the regular expression size game as a simple modification of the generalized regular expression game. Secondly these games and other methods are used in all four publications to obtain various results regarding succinctness. The following sections present each game and the results obtained.

### 3.1 Modal logics

Publication I is concerned with basic modal logic and modal  $\mu$ -calculus. Parameterized formula size games are defined for both logics. The basic modal logic game is quite simple and the novel part is the resource parameterization. For modal  $\mu$ -calculus the formulation of the game itself is new and highly complex.

In addition to the operators of basic modal logic, modal  $\mu$ -calculus also contains fixed point operators. This means the formula size game needs to be able to return to previously visited parts of the partial formula that the game is played on. Thus the game includes a full history of the formula defined thus far, unlike the modal logic game where this information can safely be forgotten. In addition, the semantics of the fixed point operators initially seem to lead to infinite plays of the formula size game to check conditions such as reachability. We use a method from [19] and use ordinals to finitize the duration of the game though it still has infinite branching in strategies.

These games give a method to study the succinctness of basic modal logic and modal  $\mu$ -calculus. It must be noted however that in the case of modal  $\mu$ -calculus, the

applications can in practice be limited by the considerable complexity of the game.

In terms of succinctness results, we prove that FO is non-elementarily more succinct than both basic modal logic and modal  $\mu$ -calculus.

**Theorem 2** (Publication I: Corollary 4.11 and Theorem 6.6). *First-order logic is non-elementarily more succinct than basic modal logic. The same applies for first-order logic and modal  $\mu$ -calculus.*

We give concrete formulas for FO that define a property based on the cumulative hierarchy of sets with a polynomial formula. We use the formula size games of modal logic and modal  $\mu$ -calculus to show that the same property requires a formula of non-elementary size to define in both of these logics. We also give a polynomial formula in two-dimensional modal logic  $ML^2$  for the same property. Essentially  $ML^2$  is basic modal logic evaluated on pairs of points  $(u, v)$  with separate relations and diamonds for the two points. See [31] for more on multi-dimensional modal logics. We obtain the same gaps for  $ML^2$  as for FO.

**Corollary 1** (Publication I: Corollary 4.14). *Two-dimensional modal logic  $ML^2$  is non-elementarily more succinct than basic modal logic. The same applies for  $ML^2$  and modal  $\mu$ -calculus.*

## 3.2 Propositional team logics

Publication II considers propositional team logics. Unlike in first-order team logics, in the propositional case all of the different atoms of dependency can be defined using other connectives. This is due to the finitary nature of propositional valuations compared to first-order quantification. The natural question then becomes: how succinct can these definitions be? We systematically study the succinctness of defining the most common atoms of dependency. The atoms considered are those of dependence, independence, inclusion, exclusion and anonymity. We show that in the existential fragment, where the splitting disjunction only occurs positively, defining any of these atoms requires a formula of size exponential in the number of propositions involved.

**Theorem 3** (Publication II: Theorems 3.4 and 3.10). *In the existential fragment of propositional team logic, formulas of size exponential in the number of propositions are required to define parity of cardinality of teams or the dependence, independence, inclusion,*



*exclusion or anonymity atoms.*

To prove the exponential lower bounds in the existential fragment, we define a new formula size game for propositional team logic. In accordance with team semantics, this game is played on sets of teams. The game resembles the formula size game for ordinary propositional logic in terms of conjunction and Boolean disjunction. The main difference is in the splitting disjunction. The semantics involve splitting teams into two subteams. This is of course reflected in the game and especially the negative side of the splitting disjunction move becomes quite involved.

In Section 3 we use the formula size game to prove exponential lower bounds in the existential fragment for defining the parity of cardinality of teams, a specific cardinality of teams and the inclusion, independence and anonymity atoms. For all of these we employ a measure called density, similar to [21]. Density is defined in terms of neighbours. Neighbours of a team are otherwise identical teams with one assignment missing. If a team has many neighbours on the opposite side of the game, then the density of the position is high and it will require a high amount of resource from  $S$  to win.

The remaining atoms of dependence and exclusion are downward closed and so the notion of density does not work as all neighbours of a given team on the left side will also be on the left side. We instead adapt another technique for lower bounds from [22] called upper dimension. Although the precise definition is a bit more involved, the upper dimension of a formula is essentially the number of subset-maximal teams that satisfy the formula. This is related to succinctness via a Lemma from [22] that links the upper dimension to the number of Boolean disjunctions in the formula.

On the other hand we show that in full propositional team logic with unrestricted Boolean negation, the same atoms can be defined with polynomial formulas.

**Theorem 4** (Publication II: Theorems 4.9 and 4.10). *In full propositional team logic there are formulas of size polynomial in the number of propositions, that define the parity of cardinality of teams and the dependence, independence, inclusion, exclusion and anonymity atoms.*

In Section 4 we obtain polynomial upper bounds for all considered atoms of dependency in full propositional team logic with unrestricted negation. More precisely, we define the negations of the atoms polynomially in the existential fragment. Besides the succinctness result between the existential and polynomial fragments, this

shows an interesting asymmetry between the common atoms of dependency and their negations. We also define parity of cardinality of teams polynomially, fully utilizing the unrestricted negation in the recursive definition of the formula.

### 3.3 Logics on words

Publications III and IV consider logics on word models. In Publication III the main focus is regular expressions while Publication IV deals with first-order logic and monadic second-order logic on words.

In Publication III we define a formula size game for generalized regular expressions, that is ones with complement as an added operation. We utilize two constrained variants of this game. The first is the game for regular expressions and the second is a game that only counts the number of stars in a generalized regular expression. In addition, we define a new natural middle ground between RE and GRE called RE over star-free expressions. These are generalized regular expressions that have no stars inside complements.

We use the regular expression size game and concrete FO formulas to reprove in a much simpler manner a known result by Stockmeyer [34] that states FO is non-elementarily more succinct than regular expressions. The original proof features encodings of Turing machines while our proof is a simple game argument in addition to some FO formulas.

Star height is a well-known open problem [8] for generalized regular expressions. From the point of view of succinctness it is then reasonable to instead study the number of stars in an expression. For regular expressions this measure trivially gives an infinite hierarchy in expressive power. We use the star counting game to show that such a hierarchy also exists for RE over star-free expressions. The case of generalized regular expressions remains open.

**Theorem 5** (Publication III: Theorem 5.1). *For each  $n \in \mathbb{N}$  there is a regular language  $L_n$  such that an RE over star-free expression  $R_n$  with  $L(R_n) = L_n$  has at least  $n$  stars.*

Publication IV concerns defining single words in FO and MSO. In particular, we consider the longest word definable in a fragment  $\mathcal{L}$  of FO or MSO. We call this length the definability number  $\text{DN}(\mathcal{L})$  of the fragment  $\mathcal{L}$ . In particular, we investigate the fragments  $\text{FO}[n]$  and  $\text{MSO}[n]$  with formulas up to size  $n$ .

We also consider two other related numbers. The Löwenheim-Skolem number

$\text{LS}(\mathcal{L})$  of a fragment  $\mathcal{L}$  is the smallest number  $m$  such that every formula in  $\mathcal{L}$  that has a model, has a model of length at most  $m$ . Similarly the Hanf number  $\text{H}(\mathcal{L})$  is the smallest number  $l$  such that if a formula of  $\mathcal{L}$  has a model of length greater than  $l$ , then it has arbitrarily long models.

Note that if  $\text{DN}(\mathcal{L}) = d$  for a fragment  $\mathcal{L}$ , then clearly  $\text{LS}(\mathcal{L}) \geq d$  since the formula defining the longest word has a model of length  $d$ . Similarly  $\text{H}(\mathcal{L}) \geq d$  since the same formula does not have arbitrarily long models. We obtain the following upper and lower bounds for these numbers, where  $\text{twr}$  is the exponential tower function and  $\log^*$  is the iterated logarithm function. Note that  $\log^*$  is essentially the inverse function of  $\text{twr}$ .

**Theorem 6** (Publication IV). *There is a constant  $c_1 \in \mathbb{N}$  such that*

$$\text{twr}(\sqrt[5]{n/c_1}) \leq \text{DN}(\text{FO}[n]) \leq \text{twr}(n/2 + \log^*((n/2)^2 + n/2) + 1).$$

*There is a constant  $c_2 \in \mathbb{N}$  such that*

$$\text{twr}(\sqrt{n/c_2}) \leq \text{DN}(\text{MSO}[n]) \leq \text{twr}(n/2 + \log^*((n/2 + 1)^2) + 1).$$

*The same bounds hold for the Löwenheim-Skolem and Hanf numbers of these fragments.*

The upper bounds are obtained by counting types of words with regards to quantifier depth. We relate quantifier depth to formula size with a crude estimate stating at most half of a meaningful formula can consist of quantifiers. Finally this relates to definability number by noting that if a word is long enough, it will have at least two prefixes that are of the same type and thus interchangeable. This means the word is not definable.

The lower bounds are obtained via concrete formulas in FO and MSO of sizes  $O(n^5)$  and  $O(n^3)$ , respectively. Both formulas define a single word representation of level  $n$  of the cumulative hierarchy of sets. The greater succinctness of MSO is gained by dividing the brackets into sets according to their depth and using a different method to single out only one such representation.



## 4 CONCLUSIONS AND FUTURE OUTLOOK

We conclude with a summary of the results obtained in this thesis and some discussion on possible directions for future research on succinctness.

### 4.1 Summary of results

In this thesis we presented parameterized formula size games for basic modal logic, modal  $\mu$ -calculus, propositional team logic and generalized regular expressions. All of these games are useful proof methods for the study of succinctness in these logics. Among them, especially the modal  $\mu$ -calculus game is highly complex to define and a contribution in and of itself.

We used these games to prove a variety of results concerning the succinctness of the corresponding logics. For the modal logics we showed in Publication I that FO and two-dimensional modal logic are both non-elementarily more succinct than ML and  $L_\mu$ .

In the propositional team setting we conducted in Publication II a systematic investigation into the succinctness of defining commonly studied atoms of dependency. We obtained exponential lower bounds for all atoms and parity in the existential fragment. In the full fragment we gave formulas that polynomially define the atoms and parity. This is the first systematic study of succinctness we know of in the team semantics setting.

For regular expressions, we reproved in Publication III a known non-elementary succinctness gap between FO and RE in a much simpler way. In addition we defined a new class of generalized regular expressions called RE over star-free expressions and showed that the number of stars in such an expression gives a full hierarchy in terms of expressive power for these new expressions.

In Publication IV we investigated the longest words definable by formulas of bounded size in FO and MSO. We obtained exponential towers of various heights

as upper and lower bounds for the length of the longest word called the definability number. We also considered the related Löwenheim-Skolem and Hanf numbers and obtained the same bounds for these as well.

## 4.2 Future directions

We sketch some possible directions for future research. We begin by considering the questions left open in each of the four publications and then move on to a discussion of research on succinctness in general.

The study of succinctness in modal logic is generally quite a developed field of research so in terms of basic modal logic there are not many open problems we can point out. Instead, the open questions we still have relating to Publication I have to do with modal  $\mu$ -calculus.

The main theorem of the formula size game for modal  $\mu$ -calculus has a clause of uniformity. Since the game loops back to the same parts of the formula multiple times, it is possible for S to have a strategy where he chooses a different move depending on when some branch of the formula is chosen by D. Uniformity means that this is not allowed; S must always choose the same move for a branch of the formula, no matter when it is reached. We conjecture that this requirement of uniformity is not actually needed for the game to function, but we were unable to prove this in the paper.

The modal  $\mu$ -calculus game is also very complex, to the point where it becomes an issue when trying to use the game as a proof method. The proof it is used for in the paper succeeds essentially because the fixed point operators are not of any use when defining the property in question. If one were to attempt a succinctness proof relating to some property that requires alternating fixed points to define, it would be reasonable to expect great difficulties.

Finally, the modal  $\mu$ -calculus game characterizes formula size in terms of the size of the syntax tree of the formula. In contrast to this, researchers of  $\mu$ -calculus seem to find the number of subformulas or the size of the Fischer-Ladner closure a more pertinent measure, see e.g. [29]. It is very reasonable to count each subformula only once, since the logic already includes references to subformulas in the form of the variables  $X$ . Since the gap between the size of the syntax tree and the number of different subformulas is exponential in the worst case, one would need to prove a

double exponential gap in terms of syntax tree size in order to obtain an exponential one for subformula size.

Publication II considered propositional team logics. The paper is to our knowledge the first systematic study of succinctness for team semantics. This naturally means that the future of this research area is rife with possibilities. In the propositional setting we saw that all of the common atoms of dependency require exponential definitions in the existential fragment, but one could also compare them to each other. Moves for desired atoms could be added to the formula size game to study for example defining inclusion atoms in the existential fragment with free use of dependence atoms. Outside the propositional setting the atoms of dependency cannot generally be defined in the base logic without them, but comparisons between logics with different atoms would still be possible to study.

Publication III studied the succinctness of RE. We defined a formula size game for GRE with complement, but only used the restricted variants for RE and RE over star-free expressions. For RE over star-free expressions we showed that the number of stars gives a hierarchy of expressive power. The question of whether this happens for GRE is open even for the case where star height is restricted to one. The general case where nested stars are allowed is presumably quite difficult since it is close to the notoriously open star height problem [8]. The formula size game is in theory a plausible tool to study the star height problem itself, but even though it leads to a different characterization of the problem, it still remains too difficult for us to solve. Perhaps a more reasonable first goal would be to find any kind of application for the full GRE game.

Publication IV considered definitions of single words in FO and MSO. The bounds we obtained for the definability numbers of the size  $n$  fragments were the first attempt and as such quite loose. The upper bound has two different pieces to improve on. When counting types we count all sets of types as possible models when in reality a vast number of these sets are unsatisfiable. On the other hand we relate quantifier depth to formula size by a simple factor of two. We argue that for each formula of which the majority of symbols are quantifiers there is an equivalent formula with lower quantifier depth. A better factor than two is surely possible but would require a more sophisticated argument.

We could also consider the definability, Löwenheim-Skolem and Hanf numbers of the size  $n$  fragments of other logics on words. At least the two-variable fragment

FO<sup>2</sup> of first-order logic and temporal logics come to mind.

We move on to a more general discussion of succinctness. The very foundation of any study of succinctness is the definition of formula size. Definitions such as string length and syntax tree size are essentially equivalent, but there are other substantially different alternatives. One such possibility is the number of different subformulas, called subformula size or DAG-size. It has been argued [29] that at least for modal  $\mu$ -calculus this would be a more pertinent measure than ordinary formula size. Unfortunately the parameterized formula size game seems very difficult to adapt to this notion of size. The original Adler-Immerman game or extended syntax DAGs could be used instead, but these lack the dynamic nature of the parameterized game. It would be very interesting to see what kind of dynamic two-player game could be defined for subformula size.

The results proven in the field of succinctness often take the form of a single sequence of properties shown to be more succinctly definable in a logic  $\mathcal{L}_1$  than another logic  $\mathcal{L}_2$ . Additionally there is sometimes a polynomial translation from  $\mathcal{L}_2$  to  $\mathcal{L}_1$  or perhaps another sequence of properties showing  $\mathcal{L}_2$  can also be more succinct than  $\mathcal{L}_1$ . In both of these cases only some kinds of properties can be defined more succinctly in  $\mathcal{L}_1$  or  $\mathcal{L}_2$ . It would give a more complete picture of the phenomenon if one could give some kind of characterization for these properties, especially in the cases where succinctness results can be shown both ways.

In conclusion, we find succinctness to be an important and interesting research topic and a natural refinement of expressive power. Formula size games are a useful if sometimes unwieldy method to study the succinctness of any logic.



## REFERENCES

- [1] Micah Adler and Neil Immerman. “An  $n!$  lower bound on formula size”. In: *ACM Trans. Comput. Log.* 4.3 (2003), pp. 296–314. DOI: 10.1145/772062.772064. URL: <https://doi.org/10.1145/772062.772064>.
- [2] Philippe Balbiani et al. “Frame-validity Games and Lower Bounds on the Complexity of Modal Axioms”. In: *Log. J. IGPL* 30.1 (2022), pp. 155–185. DOI: 10.1093/jigpal/jzaa068. URL: <https://doi.org/10.1093/jigpal/jzaa068>.
- [3] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Vol. 53. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001. ISBN: 978-1-10705088-4. DOI: 10.1017/CBO9781107050884. URL: <https://doi.org/10.1017/CBO9781107050884>.
- [4] Julian C. Bradfield and Colin Stirling. “Modal mu-calculi”. In: *Handbook of Modal Logic*. Ed. by Patrick Blackburn, J. F. A. K. van Benthem, and Frank Wolter. Vol. 3. Studies in logic and practical reasoning. North-Holland, 2007, pp. 721–756. DOI: 10.1016/s1570-2464(07)80015-2. URL: [https://doi.org/10.1016/s1570-2464\(07\)80015-2](https://doi.org/10.1016/s1570-2464(07)80015-2).
- [5] David Buchfuhrer and Christopher Umans. “The complexity of Boolean formula minimization”. In: *J. Comput. Syst. Sci.* 77.1 (2011), pp. 142–153. DOI: 10.1016/j.jcss.2010.06.011. URL: <https://doi.org/10.1016/j.jcss.2010.06.011>.
- [6] Hans van Ditmarsch et al. “Some Exponential Lower Bounds on Formula-size in Modal Logic”. In: *Advances in Modal Logic 10, invited and contributed papers from the tenth conference on "Advances in Modal Logic," held in Groningen, The Netherlands, August 5-8, 2014*. Ed. by Rajeev Goré, Barteld P. Kooi, and Agi Kurucz. College Publications, 2014, pp. 139–157. URL: <http://www.aiml.net/volumes/volume10/Ditmarsch-Fan-Hoek-Iliev.pdf>.

- [7] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, 1995. ISBN: 978-3-540-60149-4.
- [8] Keith Ellul et al. “Regular Expressions: New Results and Open Problems”. In: *J. Autom. Lang. Comb.* 10.4 (2005), pp. 407–437. DOI: 10.25596/jalc-2005-407. URL: <https://doi.org/10.25596/jalc-2005-407>.
- [9] Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. “First-Order Logic with Two Variables and Unary Temporal Logic”. In: *Inf. Comput.* 179.2 (2002), pp. 279–295. DOI: 10.1006/inco.2001.2953. URL: <https://doi.org/10.1006/inco.2001.2953>.
- [10] David Fernández-Duque and Petar Iliev. “Succinctness in Subsystems of the Spatial  $\mu$ -Calculus”. In: *FLAP* 5.4 (2018), pp. 827–874. URL: <https://www.collegepublications.co.uk/downloads/ifcolog00024.pdf>.
- [11] Tim French et al. “On the succinctness of some modal logics”. In: *Artif. Intell.* 197 (2013), pp. 56–85. DOI: 10.1016/j.artint.2013.02.003. URL: <https://doi.org/10.1016/j.artint.2013.02.003>.
- [12] Pietro Galliani. “Inclusion and exclusion dependencies in team semantics - On some logics of imperfect information”. In: *Ann. Pure Appl. Log.* 163.1 (2012), pp. 68–84. DOI: 10.1016/j.apal.2011.08.005. URL: <https://doi.org/10.1016/j.apal.2011.08.005>.
- [13] Wouter Gelade and Frank Neven. “Succinctness of the Complement and Intersection of Regular Expressions”. In: *ACM Trans. Comput. Log.* 13.1 (2012), 4:1–4:19. DOI: 10.1145/2071368.2071372. URL: <https://doi.org/10.1145/2071368.2071372>.
- [14] Goran Gogic et al. “The Comparative Linguistics of Knowledge Representation”. In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*. Morgan Kaufmann, 1995, pp. 862–869. URL: <http://ijcai.org/Proceedings/95-1/Papers/111.pdf>.
- [15] Erich Grädel and Jouko A. Väänänen. “Dependence and Independence”. In: *Stud Logica* 101.2 (2013), pp. 399–410. DOI: 10.1007/s11225-013-9479-2. URL: <https://doi.org/10.1007/s11225-013-9479-2>.

- [16] Martin Grohe and Nicole Schweikardt. “Comparing the succinctness of monadic query languages over finite trees”. In: *RAIRO Theor. Informatics Appl.* 38.4 (2004), pp. 343–373. DOI: 10.1051/ita:2004017. URL: <https://doi.org/10.1051/ita:2004017>.
- [17] Martin Grohe and Nicole Schweikardt. “The succinctness of first-order logic on linear orders”. In: *Log. Methods Comput. Sci.* 1.1 (2005). DOI: 10.2168/LMCS-1(1:6)2005. URL: [https://doi.org/10.2168/LMCS-1\(1:6\)2005](https://doi.org/10.2168/LMCS-1(1:6)2005).
- [18] Hermann Gruber and Markus Holzer. “Finite Automata, Digraph Connectivity, and Regular Expression Size”. In: *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*. Ed. by Luca Aceto et al. Vol. 5126. Lecture Notes in Computer Science. Springer, 2008, pp. 39–50. DOI: 10.1007/978-3-540-70583-3\_4. URL: [https://doi.org/10.1007/978-3-540-70583-3%5C\\_4](https://doi.org/10.1007/978-3-540-70583-3%5C_4).
- [19] Lauri Hella, Antti Kuusisto, and Raine Rönnholm. “Bounded Game-Theoretic Semantics for Modal Mu-Calculus and Some Variants”. In: *Proceedings 11th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2020, Brussels, Belgium, September 21-22, 2020*. Ed. by Jean-François Raskin and Davide Bresolin. Vol. 326. EPTCS. 2020, pp. 82–96. DOI: 10.4204/EPTCS.326.6. URL: <https://doi.org/10.4204/EPTCS.326.6>.
- [20] Lauri Hella and Johanna Stumpf. “The expressive power of modal logic with inclusion atoms”. In: *Proceedings Sixth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2015, Genoa, Italy, 21-22nd September 2015*. Ed. by Javier Esparza and Enrico Tronci. Vol. 193. EPTCS. 2015, pp. 129–143. DOI: 10.4204/EPTCS.193.10. URL: <https://doi.org/10.4204/EPTCS.193.10>.
- [21] Lauri Hella and Jouko Väänänen. “The Size of a Formula as a Measure of Complexity”. In: *Logic Without Borders - Essays on Set Theory, Model Theory, Philosophical Logic and Philosophy of Mathematics*. Ed. by Åsa Hirvonen et al. Vol. 5. Ontos Mathematical Logic. De Gruyter, 2015, pp. 193–214. DOI: 10.1515/9781614516873.193. URL: <https://doi.org/10.1515/9781614516873.193>.

- [22] Lauri Hella et al. “The Expressive Power of Modal Dependence Logic”. In: *Advances in Modal Logic 10, invited and contributed papers from the tenth conference on "Advances in Modal Logic," held in Groningen, The Netherlands, August 5-8, 2014*. Ed. by Rajeev Goré, Barteld P. Kooi, and Agi Kurucz. College Publications, 2014, pp. 294–312. URL: <http://www.aiml.net/volumes/volume10/Hella-Luosto-Sano-Virtema.pdf>.
- [23] Jaakko Hintikka and Gabriel Sandu. “Informational Independence as a Semantical Phenomenon”. In: *Studies in Logic and the Foundations of Mathematics*. Vol. 126. Elsevier, 1989, pp. 571–589. ISBN: 978-0-444-70520-4. (Visited on 11/02/2015).
- [24] Wilfrid Hodges. “Compositional Semantics for a Language of Imperfect Information”. In: *Log. J. IGPL 5.4 (1997)*, pp. 539–563. DOI: 10.1093/jigpal/5.4.539. URL: <https://doi.org/10.1093/jigpal/5.4.539>.
- [25] Wiebe van der Hoek and Petar Iliev. “On the relative succinctness of modal logics with union, intersection and quantification”. In: *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*. Ed. by Ana L. C. Bazzan et al. IFAAMAS/ACM, 2014, pp. 341–348. URL: <http://dl.acm.org/citation.cfm?id=2615788>.
- [26] Wiebe van der Hoek, Petar Iliev, and Barteld P. Kooi. “On the Relative Succinctness of Two Extensions by Definitions of Multimodal Logic”. In: *How the World Computes - Turing Centenary Conference and 8th Conference on Computability in Europe, CiE 2012, Cambridge, UK, June 18-23, 2012. Proceedings*. Ed. by S. Barry Cooper, Anuj Dawar, and Benedikt Löwe. Vol. 7318. Lecture Notes in Computer Science. Springer, 2012, pp. 323–333. DOI: 10.1007/978-3-642-30870-3\_33. URL: [https://doi.org/10.1007/978-3-642-30870-3\\_33](https://doi.org/10.1007/978-3-642-30870-3_33).
- [27] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. 3rd. USA: Addison-Wesley Longman Publishing Co., Inc., 2006. ISBN: 0321455363.
- [28] Neil Immerman. “Number of Quantifiers is Better Than Number of Tape Cells”. In: *J. Comput. Syst. Sci.* 22.3 (1981), pp. 384–406. DOI: 10.1016/0022-0000(81)90039-8. URL: [https://doi.org/10.1016/0022-0000\(81\)90039-8](https://doi.org/10.1016/0022-0000(81)90039-8).

- [29] Clemens Kupke, Johannes Marti, and Yde Venema. “Size matters in the modal  $\mu$ -calculus”. In: *CoRR* abs/2010.14430 (2020). arXiv: 2010.14430. URL: <https://arxiv.org/abs/2010.14430>.
- [30] Carsten Lutz. “Complexity and succinctness of public announcement logic”. In: *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*. Ed. by Hideyuki Nakashima et al. ACM, 2006, pp. 137–143. DOI: 10.1145/1160633.1160657. URL: <https://doi.org/10.1145/1160633.1160657>.
- [31] Maarten Marx and Yde Venema. *Multi-dimensional modal logic*. Vol. 4. Applied logic series. Kluwer, 1997. ISBN: 978-0-7923-4345-5.
- [32] Alexander A. Razborov. “Applications of matrix methods to the theory of lower bounds in computational complexity”. In: *Comb.* 10.1 (1990), pp. 81–93. DOI: 10.1007/BF02122698. URL: <https://doi.org/10.1007/BF02122698>.
- [33] A. Prasad Sistla and Edmund M. Clarke. “The Complexity of Propositional Linear Temporal Logics”. In: *J. ACM* 32.3 (1985), pp. 733–749. DOI: 10.1145/3828.3837. URL: <https://doi.org/10.1145/3828.3837>.
- [34] L. Stockmeyer. “The complexity of decision problems in automata theory and logic”. PhD thesis. Massachusetts Institute of Technology, June 1974. URL: <https://dspace.mit.edu/handle/1721.1/15540>.
- [35] Jouko Väänänen. “An atom’s worth of anonymity”. Dagstuhl Seminar 19031 “Logics for Dependence and Independence”. 2019.
- [36] Jouko A. Väänänen. *Dependence Logic - A New Approach to Independence Friendly Logic*. Vol. 70. London Mathematical Society student texts. Cambridge University Press, 2007. ISBN: 978-0-521-70015-3. URL: [http://www.cambridge.org/de/knowledge/isbn/item1164246/?site%5C\\_locale=de%5C\\_DE](http://www.cambridge.org/de/knowledge/isbn/item1164246/?site%5C_locale=de%5C_DE).
- [37] Thomas Wilke. “CTL<sup>+</sup> is Exponentially more Succinct than CTL”. In: *Foundations of Software Technology and Theoretical Computer Science, 19th Conference, Chennai, India, December 13-15, 1999, Proceedings*. Ed. by C. Pandu Rangan, Venkatesh Raman, and Ramaswamy Ramanujam. Vol. 1738. Lecture Notes in Computer Science. Springer, 1999, pp. 110–121. DOI: 10.1007/3-540-46691-6\ 9. URL: [https://doi.org/10.1007/3-540-46691-6%5C\\_9](https://doi.org/10.1007/3-540-46691-6%5C_9).

- [38] Fan Yang and Jouko Väänänen. “Propositional logics of dependence”. In: *Ann. Pure Appl. Log.* 167.7 (2016), pp. 557–589. DOI: 10.1016/j.apal.2016.03.003. URL: <https://doi.org/10.1016/j.apal.2016.03.003>.

## PUBLICATIONS





# PUBLICATION

|

**Formula size games for modal logic and  $\mu$ -calculus**

Lauri Hella and Miikka Vilander

*J. Log. Comput.* 29.8 (2019), pp. 1311–1344

DOI: 10.1093/logcom/exz025

**Publication reprinted with the permission of the copyright holders.**



---

# Formula size games for modal logic and $\mu$ -calculus

LAURI T. HELLA, *Department of Computing Sciences, Tampere University, FI-33014, Tampere, Finland.*  
E-mail: lauri.hella@tuni.fi

MIIKKA S. VILANDER, *Department of Computing Sciences, Tampere University, FI-33014, Tampere, Finland.*  
E-mail: miikka.vilander@tuni.fi

## Abstract

We propose a new version of formula size game for modal logic. The game characterizes the equivalence of pointed Kripke models up to formulas of given numbers of modal operators and binary connectives. Our game is similar to the well-known Adler–Immerman game. However, due to a crucial difference in the definition of positions of the game, its winning condition is simpler, and the second player does not have a trivial optimal strategy. Thus, unlike the Adler–Immerman game, our game is a genuine two-person game. We illustrate the use of the game by proving a non-elementary succinctness gap between bisimulation invariant first-order logic FO and (basic) modal logic ML. We also present a version of the game for the modal  $\mu$ -calculus  $L_\mu$  and show that FO is also non-elementarily more succinct than  $L_\mu$ .

*Keywords:* Succinctness, formula size game, modal logic, modal  $\mu$ -calculus, bisimulation invariant first-order logic

## 1 Introduction

Logical languages are often compared in terms of expressiveness and computational complexity. The authors of [13] argue that another important semantic aspect of a logical language is the size of formulas needed for expressing properties of structures. If two logics  $L$  and  $L'$  are equivalent in terms of expressivity, one of them may be able to express interesting properties much more succinctly than the other. According to the standard terminology, for a given function  $f$  on natural numbers,  $L$  is said to be  $f$  times more succinct than  $L'$  if there is a sequence  $(\varphi_n)_{n \in \mathbb{N}}$  of  $L$ -formulas such that for any sequence  $(\psi_n)_{n \in \mathbb{N}}$  of equivalent  $L'$ -formulas, the size of  $\psi_n$  is at least  $f(m_n)$ , where  $m_n$  is the size of  $\varphi_n$ .

The succinctness of various modal and temporal logics has been an active area of research for the last couple of decades, see e.g. [1, 5, 21–23, 34] for earlier work on the topic and [7, 9, 25, 30, 32, 33] for recent work. Typical results in the area state an exponential succinctness gap between two equally expressive logics. Often such a gap is reflected in the complexity of the logics in question. For example, Etessami *et al.* proved in [5] that the two-variable fragment  $\text{FO}^2$  of first-order logic and unary-TL (a weak version of temporal logic) have the same expressive power over  $\omega$ -words, but  $\text{FO}^2$  is exponentially more succinct than unary-TL. Furthermore, the complexity of satisfiability for  $\text{FO}^2$  is NEXPTIME-complete, while the complexity of unary-TL is in NP [28]. However, being more succinct does not always imply higher complexity: for example, public announcement logic PAL is exponentially more succinct than epistemic logic EL, but the complexity of satisfiability is the same for both of them [21].

The most commonly used methods for proving succinctness results are formula size games and extended syntax trees. It seems that the first formula size game for propositional logic was defined

## 2 Formula size games for modal logic and $\mu$ -calculus

by Razborov in [27]<sup>1</sup>. Our work was inspired by a game for branching-time temporal logic CTL by Adler and Immerman in [1]. The method of extended syntax trees was originally formulated by Grohe and Schweikardt in [14] for first-order logic. Although the work of Karchmer in [19] can be seen as a precursor of extended syntax trees, the notion was actually inspired by the Adler–Immerman game, and in a certain sense these two methods are equivalent: an extended syntax tree can be interpreted as a winning strategy for one of the players of the corresponding formula size game. Both of these methods have been adapted to a large number of modal languages, including epistemic logic [8], multimodal logics with union and intersection operators on modalities [31] and modal logic with contingency operators [32].

The basic idea of the Adler–Immerman game is that one of the players, S (spoiler), tries to show that two sets of pointed models  $\mathbb{A}$  and  $\mathbb{B}$  can be separated by a formula of size  $n$ , while the other player, D (duplicator), aims to show that no formula of size at most  $n$  suffices for this. The moves that S makes in the game reflect directly the logical operators in a formula that is supposed to separate the sets  $\mathbb{A}$  and  $\mathbb{B}$ . Any pair  $(\sigma, \delta)$  of strategies for the players S and D produces a finite game tree  $T_{\sigma, \delta}$ , and S wins this play if the size of  $T_{\sigma, \delta}$  is at most  $n$ . The strategy  $\sigma$  is a winning strategy for S if using it, S wins every play of the game. If this is the case, then there is a formula of size at most  $n$  that separates the sets, and this formula can actually be read from the strategy  $\sigma$ .

A peculiar feature of the Adler–Immerman game is that the second player, duplicator, can be completely eliminated from it. This is because D has an optimal strategy  $\delta_{\max}$ , which is to always choose the maximal allowed answer; this strategy guarantees that the size of the tree  $T_{\sigma, \delta}$  is as large as possible. Thus, in this sense, the Adler–Immerman game is not a genuine two-person game but rather a one-person game. Extended syntax trees, on the other hand, do away with the game aspect entirely.

In the present paper, we propose another type of formula size game for modal logic. Our game is a natural adaptation of the game first introduced by Razborov in [27] for propositional logic and later by Hella and Väänänen [17] for propositional logic and first-order logic. The basic setting in our game is the same as in the Adler–Immerman game: there are two players, S and D, and two sets of structures that S claims can be separated by a formula of some given size. The crucial difference is that in our game we define positions to be tuples  $(k, \mathbb{A}, \mathbb{B})$  instead of just pairs  $(\mathbb{A}, \mathbb{B})$  of sets of structures, where  $k$  is a parameter referring to the number of modal operators and binary connectives in a formula. In each move, S has to decrease the parameter  $k$ . The game ends when the players reach a position  $(k^*, \mathbb{A}^*, \mathbb{B}^*)$  such that either there is a literal separating  $\mathbb{A}^*$  and  $\mathbb{B}^*$ , or S cannot make any moves because  $k^* = 0$ . In the former case, S wins the play; otherwise, D wins.

Thus, in contrast to the Adler–Immerman game, to determine the winner in our game it suffices to consider a single ‘leaf-node’  $(k^*, \mathbb{A}^*, \mathbb{B}^*)$  of the game tree. This also means that our game is a real two-person game: the final position  $(k^*, \mathbb{A}^*, \mathbb{B}^*)$  of a play depends on the moves of D, and there is no simple optimal strategy for D that could be used for eliminating the role of D in the game.

We believe that our game is more intuitive and thus, in some cases, it may be easier to use than the Adler–Immerman game. On the other hand, it should be remarked that the two games are essentially equivalent: the moves corresponding to connectives and modal operators are the same in both games (when restricting to the sets  $\mathbb{A}$  and  $\mathbb{B}$  in a position  $(k, \mathbb{A}, \mathbb{B})$ ). Hence, in principle, it is possible to translate a winning strategy in one of the games to a corresponding winning strategy in the other.

Additionally, we introduce a formula size game for the modal  $\mu$ -calculus. This game is obtained by adapting the formula size game of modal logic to the setting with fixed point operators  $\mu$  and  $\nu$ .

<sup>1</sup>We thank an anonymous referee for pointing out this very pertinent reference.

A new challenge in defining such a game is that if  $S$  uses a fixed point  $\eta X$  ( $\eta \in \{\mu, \nu\}$ ) as the logical operator in his move and later uses the corresponding variable  $X$ , then in the next round, the game has to return to the subformula that follows  $\eta X$ . This means that the play may become infinite, and defining the correct winning condition for infinite plays is complicated. We solve this problem by adding ordinal clocks to the pointed Kripke models in the sets  $\mathbb{A}$  and  $\mathbb{B}$ . The idea is that the ordinals corresponding to a fixed point variable  $X$  decrease each time the game returns to an earlier formula from a position with label  $X$ . This, in conjunction with keeping the sets  $\mathbb{A}$  and  $\mathbb{B}$  always finite, guarantees that every play of the game is finite. The idea of using ordinal clocks is also used in [16] to define finite semantic games for  $L_\mu$ .

We illustrate the use of our games by proving two non-elementary succinctness gaps: one between first-order logic FO and (basic) modal logic ML and the other between FO and the modal  $\mu$ -calculus  $L_\mu$ . More precisely, we define a property of pointed Kripke models, which is closed under bisimulation, by a first-order formula of linear size, and show that this property cannot be defined by any ML- or  $L_\mu$ -formula of size less than the exponential tower of height  $n - 1$ . Furthermore, we show that the same property of pointed Kripke models is already definable by a formula of size  $\mathcal{O}(2^n)$  in a version  $ML^2$  of two-dimensional modal logic. Hence, the same non-elementary succinctness result holds for  $ML^2$  over ML.

A similar gap between FO and temporal logic follows from a construction in the PhD thesis [29] of Stockmeyer. He proved that the satisfiability problem of FO over words is of non-elementary complexity. Etesami and Wilke [6] observed that from Stockmeyer's proof it is possible to extract FO-formulas of size  $\mathcal{O}(n)$  whose smallest models are words of length non-elementary in  $n$ . On the other hand, it is well known that any satisfiable formula of temporal logic has a model of size  $\mathcal{O}(2^n)$ , where  $n$  is the size of the formula. Another result related to ours can be found in [26], where Otto shows that FO is exponentially more succinct than ML by relating the modal depth of the ML-formula to the quantifier rank of the FO-formula. In contrast to this, our proof relies entirely on the number of disjunctions and conjunctions in the ML-formula.

For the modal  $\mu$ -calculus, the literature regarding succinctness is scarcer. In [15], Grohe and Schweikardt show several succinctness gaps between monadic second-order logics, many with fixed points. They use automata-theoretic techniques and cite a non-elementary succinctness gap between MSO and  $L_\mu$  as well known<sup>2</sup>.

The structure of the paper is as follows. In Section 2, we present the logics used in the paper, fix some notation and define our notion of formula size. In Section 3, we present the formula size game for ML and show some basic results for it. Section 4 is dedicated to the non-elementary succinctness gap between FO and ML and all necessary definitions and lemmas to prove it. In Section 5, we define the formula size game for  $L_\mu$  and show basic results. Finally, in Section 6, we show the non-elementary succinctness of FO over  $L_\mu$ . Section 7 is the conclusion.

The work on modal logic was previously published in the conference paper [18]. This version has some minor changes to the modal logic part and the sections on the modal  $\mu$ -calculus are completely new.

## 2 Preliminaries

In this section, we fix some notation, define the syntax and semantics of basic modal logic and the modal  $\mu$ -calculus and define our notions of formula size. For a detailed account on the logics used

---

<sup>2</sup>We were unable to find a source in the literature for this result but we are reasonably convinced that a non-elementary gap already between FO and  $L_\mu$  is a new result.

#### 4 Formula size games for modal logic and $\mu$ -calculus

in the paper, we refer to the textbook [2] of Blackburn *et al.* for basic modal logic and [3] for the modal  $\mu$ -calculus.

##### Basic modal logic and first-order logic

Let  $Prop$  be an infinite set of propositional symbols and let  $\Phi \subseteq Prop$ . Let  $\mathcal{M} = (W, R, V)$ , where  $W$  is a set,  $R \subseteq W \times W$  and  $V : \Phi \rightarrow \mathcal{P}(W)$ , and let  $w \in W$ . The structure  $(\mathcal{M}, w)$  is called a *pointed Kripke model for  $\Phi$* .

Let  $(\mathcal{M}, w)$  be a pointed Kripke model. We use the notation

$$\square(\mathcal{M}, w) := \{(\mathcal{M}, v) \mid v \in W, wR^M v\}.$$

If  $\mathbb{A}$  is a set of pointed Kripke models, we use the notation

$$\square\mathbb{A} := \bigcup_{(\mathcal{M}, w) \in \mathbb{A}} \square(\mathcal{M}, w).$$

Furthermore, if  $f$  is a function  $f : \mathbb{A} \rightarrow \square\mathbb{A}$  such that  $f(\mathcal{M}, w) \in \square(\mathcal{M}, w)$  for every  $(\mathcal{M}, w) \in \mathbb{A}$ , then we use the notation

$$\diamond_f \mathbb{A} := f(\mathbb{A}).$$

Intuitively,  $\square(\mathcal{M}, w)$  is the set of all successor models of  $(\mathcal{M}, w)$ ,  $\square\mathbb{A}$  is the collection of all successor models of all models  $(\mathcal{M}, w) \in \mathbb{A}$  and  $\diamond_f \mathbb{A}$  consists of one successor for each model in  $\mathbb{A}$ , where the successors are given by the function  $f$ . We now define the syntax and semantics of basic modal logic for pointed models.

Let  $\Phi \subseteq Prop$ . The set of formulas of  $ML(\Phi)$  is generated by the following grammar

$$\varphi := \top \mid \perp \mid p \mid \neg p \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid \diamond \varphi \mid \square \varphi,$$

where  $p \in \Phi$ .

As is apparent from the definition of the syntax, we assume that all  $ML$ -formulas are in negation normal form. This is useful for the formula size game that we introduce in the next section.

The satisfaction relation  $(\mathcal{M}, w) \models \varphi$  between pointed Kripke models  $(\mathcal{M}, w)$  and  $ML(\Phi)$ -formulas  $\varphi$  is defined as follows:

- (1)  $(\mathcal{M}, w) \models \top$  for all  $(\mathcal{M}, w)$ , and  $(\mathcal{M}, w) \not\models \perp$  for all  $(\mathcal{M}, w)$ ,
- (2)  $(\mathcal{M}, w) \models p \Leftrightarrow w \in V(p)$ , and  $(\mathcal{M}, w) \models \neg p \Leftrightarrow w \notin V(p)$ ,
- (3)  $(\mathcal{M}, w) \models (\varphi \wedge \psi) \Leftrightarrow (\mathcal{M}, w) \models \varphi$  and  $(\mathcal{M}, w) \models \psi$ ,
- (4)  $(\mathcal{M}, w) \models (\varphi \vee \psi) \Leftrightarrow (\mathcal{M}, w) \models \varphi$  or  $(\mathcal{M}, w) \models \psi$ ,
- (5)  $(\mathcal{M}, w) \models \diamond \varphi \Leftrightarrow$  there is  $(\mathcal{M}, v) \in \square(\mathcal{M}, w)$  such that  $(\mathcal{M}, v) \models \varphi$ ,
- (6)  $(\mathcal{M}, w) \models \square \varphi \Leftrightarrow$  for every  $(\mathcal{M}, v) \in \square(\mathcal{M}, w)$  it holds that  $(\mathcal{M}, v) \models \varphi$ .

Furthermore, if  $\mathbb{A}$  is a class of pointed Kripke models, then

$$\mathbb{A} \models \varphi \Leftrightarrow (\mathcal{A}, w) \models \varphi \text{ for every } (\mathcal{A}, w) \in \mathbb{A}.$$

For the sake of convenience, we also use the notation

$$\mathbb{A} \models \neg \varphi \Leftrightarrow (\mathcal{A}, w) \not\models \varphi \text{ for every } (\mathcal{A}, w) \in \mathbb{A}.$$

Note that this is only a notational convention as  $\neg \varphi$  is not in negation normal form and as such is generally not a formula in our syntax.

In Section 4, we consider the case  $\Phi = \emptyset$ . In this case, the only available literals are the constants  $\top$  and  $\perp$ , which are always true or false, respectively.

The syntax and semantics of first-order logic are defined in the standard way. Each ML-formula  $\varphi$  defines a class  $\text{Mod}(\varphi)$  of pointed Kripke models:

$$\text{Mod}(\varphi) := \{(\mathcal{M}, w) \mid (\mathcal{M}, w) \models \varphi\}.$$

In the same way, any FO-formula  $\psi(x)$  in the vocabulary consisting of the accessibility relation symbol  $R$  and unary relation symbols  $U_p$  for  $p \in \Phi$  defines a class  $\text{Mod}(\psi)$  of pointed Kripke models:

$$\text{Mod}(\psi) := \{(\mathcal{M}, w) \mid \mathcal{M} \models \psi[w/x]\}.$$

The formulas  $\varphi \in \text{ML}$  and  $\psi(x) \in \text{FO}$  are *equivalent* if  $\text{Mod}(\varphi) = \text{Mod}(\psi)$ .

For the sake of easier reading, we define here the standard notion of  $n$ -bisimulation.

#### DEFINITION 2.1

Let  $(\mathcal{M}, w)$  and  $(\mathcal{M}', w')$  be pointed  $\Phi$ -models. We say that  $(\mathcal{M}, w)$  and  $(\mathcal{M}', w')$  are  $n$ -bisimilar,  $(\mathcal{M}, w) \simeq_n (\mathcal{M}', w')$ , if there are binary relations  $Z_n \subseteq \dots \subseteq Z_0$  such that for every  $0 \leq i \leq n-1$  we have

- (1)  $(\mathcal{M}, w)Z_n(\mathcal{M}', w')$ ,
- (2) if  $(\mathcal{M}, v)Z_0(\mathcal{M}', v')$ , then  $(\mathcal{M}, v) \models p \Leftrightarrow (\mathcal{M}', v') \models p$  for each  $p \in \Phi$ ,
- (3) if  $(\mathcal{M}, v)Z_{i+1}(\mathcal{M}', v')$  and  $(\mathcal{M}, u) \in \Box(\mathcal{M}, v)$  then there is  $(\mathcal{M}', u') \in \Box(\mathcal{M}', v')$  such that  $(\mathcal{M}, u)Z_i(\mathcal{M}', u')$ ,
- (4) if  $(\mathcal{M}, v)Z_{i+1}(\mathcal{M}', v')$  and  $(\mathcal{M}', u') \in \Box(\mathcal{M}', v')$  then there is  $(\mathcal{M}, u) \in \Box(\mathcal{M}, v)$  such that  $(\mathcal{M}, u)Z_i(\mathcal{M}', u')$ .

It is well known that if  $\Phi$  is finite, two pointed  $\Phi$ -models are  $n$ -bisimilar if and only if they satisfy the same ML( $\Phi$ )-formulas of modal depth at most  $n$ .

The well known link between ML and FO is the following theorem.

#### THEOREM 2.2 (van Benthem characterization theorem).

A first-order formula  $\psi(x)$  is equivalent to some formula in ML if and only if  $\text{Mod}(\psi)$  is closed under bisimulation.

If a property of pointed Kripke models is closed under  $n$ -bisimulation for some  $n \in \mathbb{N}$ , then it is also closed under bisimulation. Thus, if a property of pointed Kripke models is FO-definable and closed under  $n$ -bisimulation, it is also ML-definable. We will use this version of van Benthem's characterization in Section 4.1 to show that a certain property is ML-definable.

#### Modal $\mu$ -calculus

Let  $\Phi \subseteq \text{Prop}$  and let  $\text{Var}$  be an infinite set of variables. The syntax of the modal  $\mu$ -calculus  $L_\mu(\Phi)$  is given by the grammar:

$$\varphi ::= \top \mid \perp \mid p \mid \neg p \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \diamond \varphi \mid \Box \varphi \mid X \mid \mu X.\varphi \mid \nu X.\varphi,$$

where  $p \in \Phi$  and  $X \in \text{Var}$ . Note that all formulas are again in negation normal form. We additionally assume for simplicity that variables of different fixed points are distinct.

Truth of formulas of  $L_\mu(\Phi)$  is, like ML, evaluated on pointed Kripke models  $(\mathcal{M}, w)$ , where  $\mathcal{M} = (W, R, V)$ . Let  $\varphi \in L_\mu(\Phi)$  and let  $\rho : \text{Var} \rightarrow \mathcal{P}(W)$  be a valuation of variables.

## 6 Formula size games for modal logic and $\mu$ -calculus

We define truth relation  $(\mathcal{M}, w) \models_{\rho} \varphi$  between pointed models and  $L_{\mu}(\Phi)$ -formulas. Let  $\|\varphi\|_{\rho} := \{w \in W \mid (\mathcal{M}, w) \models_{\rho} \varphi\}$  and let  $\Gamma_{\varphi, \rho} : \mathcal{P}(W) \rightarrow \mathcal{P}(W)$  be an operator, which maps  $W'$  to  $\|\varphi\|_{\rho[W'/X]}$ . The notation LFP stands for least fixed point of an operator and GFP for greatest fixed point. Since variables only occur positively in fixed point formulas,  $\Gamma_{\varphi, \rho}$  is a monotone operator. By the Knaster–Tarski theorem, the least and greatest fixed points of such a monotone operator always exist. The recursive definition of  $\models_{\rho}$  is as follows:

- $(\mathcal{M}, w) \models_{\rho} p \Leftrightarrow w \in V(p)$ ,
- $(\mathcal{M}, w) \models_{\rho} X \Leftrightarrow w \in \rho(X)$ ,
- $(\mathcal{M}, w) \models_{\rho} (\varphi \vee \psi) \Leftrightarrow (\mathcal{M}, w) \models_{\rho} \varphi$  or  $(\mathcal{M}, w) \models_{\rho} \psi$ ,
- $(\mathcal{M}, w) \models_{\rho} (\varphi \wedge \psi) \Leftrightarrow (\mathcal{M}, w) \models_{\rho} \varphi$  and  $(\mathcal{M}, w) \models_{\rho} \psi$ ,
- $(\mathcal{M}, w) \models_{\rho} \diamond\varphi \Leftrightarrow$  there is  $(\mathcal{M}, v) \in \Box(\mathcal{M}, w)$  such that  $(\mathcal{M}, v) \models_{\rho} \varphi$ ,
- $(\mathcal{M}, w) \models_{\rho} \Box\varphi \Leftrightarrow$  for every  $(\mathcal{M}, v) \in \Box(\mathcal{M}, w)$  it holds that  $(\mathcal{M}, v) \models_{\rho} \varphi$ ,
- $(\mathcal{M}, w) \models_{\rho} \mu X.\varphi \Leftrightarrow w \in \text{LFP}(\Gamma_{\varphi, \rho})$ ,
- $(\mathcal{M}, w) \models_{\rho} \nu X.\varphi \Leftrightarrow w \in \text{GFP}(\Gamma_{\varphi, \rho})$ .

### Formula size

We define notions of formula size for ML,  $L_{\mu}$  and FO. Note that many different notions are called formula size in the literature and our notion is close to the length of the formula as a string rather than, say, the DAG-size<sup>3</sup> of it.

#### DEFINITION 2.3

The *size* of a formula  $\varphi \in \text{ML}$ , denoted  $\text{sz}(\varphi)$ , is defined recursively as follows:

- (1) If  $\varphi$  is a literal, then  $\text{sz}(\varphi) = 1$ .
- (2) If  $\varphi = \psi \vee \vartheta$  or  $\varphi = \psi \wedge \vartheta$ , then  $\text{sz}(\varphi) = \text{sz}(\psi) + \text{sz}(\vartheta) + 1$ .
- (3) If  $\varphi = \diamond\psi$  or  $\varphi = \Box\psi$ , then  $\text{sz}(\varphi) = \text{sz}(\psi) + 1$ .

#### DEFINITION 2.4

The *size* of a formula  $\varphi \in L_{\mu}$ , denoted  $\text{sz}(\varphi)$ , is defined recursively as follows:

- (1)  $\text{sz}(l) = \text{sz}(X) = 1$ , where  $l$  is a literal and  $X$  is a variable,
- (2)  $\text{sz}(\varphi \vee \psi) = \text{sz}(\varphi \wedge \psi) = \text{sz}(\varphi) + \text{sz}(\psi) + 1$ ,
- (3)  $\text{sz}(\diamond\varphi) = \text{sz}(\Box\varphi) = \text{sz}(\mu X.\varphi) = \text{sz}(\nu X.\varphi) = \text{sz}(\varphi) + 1$ .

The size of a formula is essentially its length as a string. Note, however, that we do not count negations as we view them as parts of literals. Another aspect worth mentioning is the size of descriptions of propositional symbols. If we have an infinite set of propositional symbols, the size of the encoding of each symbol in a fixed size vocabulary necessarily grows logarithmically. Here we consider all propositional symbols to be of size one.

Similarly, we define formula size for FO to be the number of binary connectives, quantifiers and literals in the formula. In general, this could lead to an arbitrarily large difference between formula size and actual string length. For example, if  $f$  is a unary function symbol, then atomic formulas of

<sup>3</sup>The DAG-size of a formula  $\varphi$  is the number of nodes of the syntactic structure of  $\varphi$  in the form of a directed acyclic graph. This is the same as the number of subformulas of  $\varphi$ .



the form  $f(x) = x$ ,  $f(f(x)) = x$  and so on, all have size 1. In this paper, however, we only consider formulas with one binary relation so this is not an issue.

DEFINITION 2.5

The *size* of a formula  $\varphi \in \text{FO}$ , denoted by  $\text{sz}(\varphi)$ , is defined recursively as follows:

- (1) If  $\varphi$  is a literal, then  $\text{sz}(\varphi) = 1$ .
- (2) If  $\varphi = \neg\psi$ , then  $\text{sz}(\varphi) = \text{sz}(\psi)$ .
- (3) If  $\varphi = \psi \vee \vartheta$  or  $\varphi = \psi \wedge \vartheta$ , then  $\text{sz}(\varphi) = \text{sz}(\psi) + \text{sz}(\vartheta) + 1$ .
- (4) If  $\varphi = \exists x\psi$  or  $\varphi = \forall x\psi$ , then  $\text{sz}(\varphi) = \text{sz}(\psi) + 1$ .

To refer to some rather large formula sizes, we need the exponential tower function.

DEFINITION 2.6

We define the function  $\text{twr} : \mathbb{N} \rightarrow \mathbb{N}$  recursively as follows:

$$\begin{aligned} \text{twr}(0) &= 1 \\ \text{twr}(n + 1) &= 2^{\text{twr}(n)}. \end{aligned}$$

We will also use in the sequel the binary logarithm function, denoted by  $\log$ .

*Separating classes by formulas*

The definitions of the formula size games in Sections 3 and 5 are based on the notion of separating classes of pointed Kripke models by formulas. Recall that by the notation  $\mathbb{B} \models \neg\varphi$  we mean that for every model  $(\mathcal{B}, w) \in \mathbb{B}$ , we have  $(\mathcal{B}, w) \not\models \varphi$ . As formulas of ML are also in  $L_\mu$ , we only define the following for  $L_\mu$  and FO.

DEFINITION 2.7

Let  $\mathbb{A}$  and  $\mathbb{B}$  be classes of pointed Kripke models.

- (a) We say that a formula  $\varphi \in L_\mu$  *separates*  $\mathbb{A}$  from  $\mathbb{B}$  if  $\mathbb{A} \models \varphi$  and  $\mathbb{B} \models \neg\varphi$ .
- (b) Similarly, a formula  $\psi(x) \in \text{FO}$  separates  $\mathbb{A}$  from  $\mathbb{B}$  if for all  $(\mathcal{M}, w) \in \mathbb{A}$ ,  $\mathcal{M} \models \psi[w/x]$  and for all  $(\mathcal{M}, w) \in \mathbb{B}$ ,  $\mathcal{M} \models \neg\psi[w/x]$ .

In other words, a formula  $\varphi \in L_\mu$  separates  $\mathbb{A}$  from  $\mathbb{B}$  if  $\mathbb{A} \subseteq \text{Mod}(\varphi)$  and  $\mathbb{B} \subseteq \overline{\text{Mod}(\varphi)}$ , where  $\overline{\text{Mod}(\varphi)}$  is the complement of  $\text{Mod}(\varphi)$ .

### 3 The formula size game for ML

As in the Adler–Immerman game, the basic idea in our formula size game is that there are two players, S (Samson) and D (Delilah), who play on a pair  $(\mathbb{A}, \mathbb{B})$  of two sets of pointed Kripke models. The aim of S is to show that  $\mathbb{A}$  and  $\mathbb{B}$  can be separated by a formula with size at most  $k$ , while D tries to refute this. The moves of S reflect the connectives and modal operators of a formula that is supposed to separate the sets.

The crucial difference between our game and the Adler–Immerman game is that we define positions in the game to be tuples  $(k, \mathbb{A}, \mathbb{B})$  instead of just pairs  $(\mathbb{A}, \mathbb{B})$ . As in the A-I game, D chooses for connective moves, which branch she would like to see played next. However, our game never returns to the branch not chosen, so D has a genuine choice to make. The winning condition

## 8 Formula size games for modal logic and $\mu$ -calculus

of our game is based on a natural property of single positions instead of the size of the entire game tree.

We give now the precise definition of our game.

### DEFINITION 3.1

Let  $\mathbb{A}_0$  and  $\mathbb{B}_0$  be sets of pointed  $\Phi$ -Kripke models and let  $k_0 \in \mathbb{N}$ . The formula size game between the sets  $\mathbb{A}_0$  and  $\mathbb{B}_0$ , denoted  $\text{FS}_{k_0}^\Phi(\mathbb{A}_0, \mathbb{B}_0)$ , has two players, S and D. The number  $k_0$  is the *resource parameter* of the game. The starting position of the game is  $(k_0, \mathbb{A}_0, \mathbb{B}_0)$ . Let the position after  $n$  moves be  $(k, \mathbb{A}, \mathbb{B})$ . If  $k = 0$ , D wins the game. If  $k > 0$ , S has the following five moves to choose from the following:

- $\vee$ -move: First, S chooses natural numbers  $k_1$  and  $k_2$  and sets  $\mathbb{A}_1$  and  $\mathbb{A}_2$  such that  $k_1 + k_2 + 1 = k$  and  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$ . Then D decides whether the game continues from the position  $(k_1, \mathbb{A}_1, \mathbb{B})$  or the position  $(k_2, \mathbb{A}_2, \mathbb{B})$ .
- $\wedge$ -move: First, S chooses natural numbers  $k_1$  and  $k_2$  and sets  $\mathbb{B}_1$  and  $\mathbb{B}_2$  such that  $k_1 + k_2 + 1 = k$  and  $\mathbb{B}_1 \cup \mathbb{B}_2 = \mathbb{B}$ . Then D decides whether the game continues from the position  $(k_1, \mathbb{A}, \mathbb{B}_1)$  or the position  $(k_2, \mathbb{A}, \mathbb{B}_2)$ .
- $\diamond$ -move: S chooses a function  $f : \mathbb{A} \rightarrow \square\mathbb{A}$  such that  $f(\mathcal{A}, w) \in \square(\mathcal{A}, w)$  for all  $(\mathcal{A}, w) \in \mathbb{A}$  and the game continues from the position  $(k - 1, \diamond_f\mathbb{A}, \square\mathbb{B})$ .
- $\square$ -move: S chooses a function  $g : \mathbb{B} \rightarrow \square\mathbb{B}$  such that  $g(\mathcal{B}, w) \in \square(\mathcal{B}, w)$  for all  $(\mathcal{B}, w) \in \mathbb{B}$  and the game continues from the position  $(k - 1, \square\mathbb{A}, \diamond_g\mathbb{B})$ .
- *Lit-move*: S chooses a literal  $l \in \text{Lit}(\Phi)$ . If  $l$  separates the sets  $\mathbb{A}$  and  $\mathbb{B}$ , S wins. Otherwise, D wins. Note that if  $l = \top$ , S wins if  $\mathbb{A} \neq \emptyset$  and  $\mathbb{B} = \emptyset$ , and vice versa for  $\perp$ .

Since D wins if  $k$  runs out, the parameter  $k$  can be thought of as a resource of S that she spends on connectives and literals. In addition, if there is a model  $(\mathcal{M}, w) \in \mathbb{A}$  (or  $\mathbb{B}$ ) for which  $\square(\mathcal{M}, w) = \emptyset$ , then S cannot make a  $\diamond$ - (or  $\square$ -) move.

We prove that the formula size game indeed characterizes the separation of two sets of pointed Kripke models by a formula of a given size.

### THEOREM 3.2

Let  $\mathbb{A}$  and  $\mathbb{B}$  be sets of pointed  $\Phi$ -models and let  $k$  be natural number. Then the following conditions are equivalent:

- (win) $_k$  S has a winning strategy in the game  $\text{FS}_k^\Phi(\mathbb{A}, \mathbb{B})$ .
- (sep) $_k$  There is a formula  $\varphi \in \text{ML}(\Phi)$  such that  $\text{sz}(\varphi) \leq k$  and the formula  $\varphi$  separates  $\mathbb{A}$  from  $\mathbb{B}$ .

PROOF. The proof proceeds by induction on the number  $k$ . First, assume  $k = 1$ . If S makes any non-literal move, D wins since  $k = 0$  in the following position. So the only possibility for a winning strategy is a literal move. There is a winning literal move if and only if there is a literal, which separates  $\mathbb{A}_0$  from  $\mathbb{B}_0$ . Thus,  $(\text{win})_1 \Leftrightarrow (\text{sep})_1$ .

Suppose then that  $k > 1$  and  $(\text{win})_l \Leftrightarrow (\text{sep})_l$  for all  $l < k$ . Assume first that  $(\text{win})_k$  holds. Consider the following cases according to the first move in the winning strategy of S. For  $\vee$ - and  $\wedge$ -moves, we use the index  $i$  to always mean  $i \in \{1, 2\}$ .

- (a) Assume the first move of the winning strategy is a literal move and  $\varphi$  is the literal chosen by S. Then  $\varphi$  separates  $\mathbb{A}$  and  $\mathbb{B}$  and  $\text{sz}(\varphi) = 1$  so  $(\text{sep})_k$  trivially holds.
- (b) Assume that the first move of the winning strategy of S is a  $\vee$ -move choosing numbers  $k_1, k_2 \in \mathbb{N}$  such that  $k_1 + k_2 + 1 = k$ , and sets  $\mathbb{A}_1, \mathbb{A}_2 \subseteq \mathbb{A}$  such that  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$ . Since this

move is given by a winning strategy, S has a winning strategy for both possible continuations of the game,  $(k_1, \mathbb{A}_1, \mathbb{B})$  and  $(k_2, \mathbb{A}_2, \mathbb{B})$ . Since  $k_i < k$ , by induction hypothesis, there is a formula  $\psi_i$  such that  $\text{sz}(\psi_i) \leq k_i$  and  $\psi_i$  separates  $\mathbb{A}_i$  from  $\mathbb{B}$ . Thus,  $\mathbb{A}_i \models \psi_i$  so  $\mathbb{A} \models \psi_1 \vee \psi_2$ . On the other hand,  $\mathbb{B} \models \neg\psi_1$  and  $\mathbb{B} \models \neg\psi_2$  so  $\mathbb{B} \models \neg(\psi_1 \vee \psi_2)$ . Therefore, the formula  $\psi_1 \vee \psi_2$  separates  $\mathbb{A}$  from  $\mathbb{B}$ . In addition,  $\text{sz}(\psi_1 \vee \psi_2) = \text{sz}(\psi_1) + \text{sz}(\psi_2) + 1 \leq k_1 + k_2 + 1 = k$  so  $(\text{sep})_k$  holds.

- (c) The case in which the first move of the winning strategy of S is a  $\wedge$ -move is proved in the same way as the previous one, with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched, and disjunction replaced by conjunction.
- (d) Assume that the first move of the winning strategy of S is a  $\diamond$ -move choosing a function  $f : \mathbb{A} \rightarrow \square\mathbb{A}$  such that  $f(\mathcal{A}, w) \in \square(\mathcal{A}, w)$  for all  $(\mathcal{A}, w) \in \mathbb{A}$ . The game continues from the position  $(k-1, \diamond_f\mathbb{A}, \square\mathbb{B})$  and S has a winning strategy from this position. By induction hypothesis, there is a formula  $\psi$  such that  $\text{sz}(\psi) \leq k-1$  and  $\psi$  separates  $\diamond_f\mathbb{A}$  from  $\square\mathbb{B}$ . Now for every  $(\mathcal{A}, w) \in \mathbb{A}$  we have  $f(\mathcal{A}, w) \in \square(\mathcal{A}, w)$  and  $f(\mathcal{A}, w) \models \psi$ . Therefore,  $\mathbb{A} \models \diamond\psi$ . On the other hand,  $\square\mathbb{B} \models \neg\psi$  so for every  $(\mathcal{B}, w) \in \mathbb{B}$  and every  $(\mathcal{B}, v) \in \square(\mathcal{B}, w)$  we have  $(\mathcal{B}, v) \not\models \psi$ . Thus,  $\mathbb{B} \models \neg\diamond\psi$ . So the formula  $\diamond\psi$  separates  $\mathbb{A}$  from  $\mathbb{B}$  and since  $\text{sz}(\diamond\psi) = \text{sz}(\psi) + 1 \leq k$ ,  $(\text{sep})_k$  holds.
- (e) The case in which the first move of the winning strategy of S is a  $\square$ -move is similar to the case of the  $\diamond$ -move. It suffices to switch the classes  $\mathbb{A}$  and  $\mathbb{B}$  and replace  $\diamond$  with  $\square$ .

Now assume  $(\text{sep})_k$  holds, and  $\varphi$  is the formula separating  $\mathbb{A}$  from  $\mathbb{B}$ . We obtain a winning strategy of S for the game  $\text{FS}_k^\varphi(\mathbb{A}, \mathbb{B})$  using  $\varphi$  as follows:

- (a) If  $\varphi$  is a literal, S wins the game by making the corresponding literal move.
- (b) Assume that  $\varphi = \psi_1 \vee \psi_2$ . Let  $\mathbb{A}_i := \{(\mathcal{A}, w) \in \mathbb{A} \mid (\mathcal{A}, w) \models \psi_i\}$ . Since  $\mathbb{A} \models \varphi$ , we have  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$ . In addition, since  $\mathbb{B} \models \neg\varphi$ , we have  $\mathbb{B} \models \neg\psi_i$ . Thus,  $\psi_i$  separates  $\mathbb{A}_i$  from  $\mathbb{B}$ . Since  $\text{sz}(\psi_1) + \text{sz}(\psi_2) + 1 = \text{sz}(\varphi) \leq k$ , there are  $k_1, k_2 \in \mathbb{N}$  such that  $k_1 + k_2 + 1 = k$  and  $\text{sz}(\psi_i) \leq k_i$ . By induction hypothesis, S has winning strategies for the games  $\text{FS}_{k_i}^\varphi(\mathbb{A}_i, \mathbb{B})$ . Since  $k \geq \text{sz}(\varphi) \geq 1$ , S can start the game  $\text{FS}_k^\varphi(\mathbb{A}, \mathbb{B})$  with a  $\vee$ -move choosing the numbers  $k_1$  and  $k_2$  and the sets  $\mathbb{A}_1$  and  $\mathbb{A}_2$ . Then S wins the game by following the winning strategy for whichever position D chooses.
- (c) Assume that  $\varphi = \psi_1 \wedge \psi_2$ . Let  $\mathbb{B}_1 := \{(\mathcal{B}, w) \in \mathbb{B} \mid (\mathcal{B}, w) \not\models \psi_1\}$  and  $\mathbb{B}_2 := \{(\mathcal{B}, w) \in \mathbb{B} \mid (\mathcal{B}, w) \not\models \psi_2\}$ . Since  $\mathbb{B} \models \neg\varphi$ , we have  $\mathbb{B}_1 \cup \mathbb{B}_2 = \mathbb{B}$ . In addition, since  $\mathbb{A} \models \varphi$ , we have  $\mathbb{A} \models \psi_1$  and  $\mathbb{A} \models \psi_2$ . Thus,  $\psi_1$  separates  $\mathbb{A}$  from  $\mathbb{B}_1$  while  $\psi_2$  separates  $\mathbb{A}$  from  $\mathbb{B}_2$ . As in the previous case, there are  $k_1, k_2 \in \mathbb{N}$  such that  $k_1 + k_2 = k$ ,  $\text{sz}(\psi_1) \leq k_1$  and  $\text{sz}(\psi_2) \leq k_2$ . By induction hypothesis, S has a winning strategy for the games  $\text{FS}_{k_1}^\varphi(\mathbb{A}, \mathbb{B}_1)$  and  $\text{FS}_{k_2}^\varphi(\mathbb{A}, \mathbb{B}_2)$ . S wins the game  $\text{FS}_k^\varphi(\mathbb{A}, \mathbb{B})$  by starting with a  $\wedge$ -move choosing the numbers  $k_1$ , and  $k_2$  and the sets  $\mathbb{B}_1$  and  $\mathbb{B}_2$  and proceeding according to the winning strategies for the games  $\text{FS}_{k_1}^\varphi(\mathbb{A}, \mathbb{B}_1)$  and  $\text{FS}_{k_2}^\varphi(\mathbb{A}, \mathbb{B}_2)$ .
- (d) Assume that  $\varphi = \diamond\psi$ . Since  $\mathbb{A} \models \varphi$ , for every  $(\mathcal{A}, w) \in \mathbb{A}$  there is  $(\mathcal{A}, v_w) \in \square(\mathcal{A}, w)$  such that  $(\mathcal{A}, v_w) \models \psi$ . We define the function  $f : \mathbb{A} \rightarrow \square\mathbb{A}$  by  $f(\mathcal{A}, w) = (\mathcal{A}, v_w)$ . Clearly,  $\diamond_f\mathbb{A} \models \psi$ . On the other hand,  $\mathbb{B} \models \neg\varphi$  so for each  $(\mathcal{B}, w) \in \mathbb{B}$  and each  $(\mathcal{B}, v) \in \square(\mathcal{B}, w)$  we have  $(\mathcal{B}, v) \not\models \psi$ . Therefore,  $\square\mathbb{B} \models \neg\psi$  and the formula  $\psi$  separates  $\diamond_f\mathbb{A}$  from  $\square\mathbb{B}$ . Moreover,  $\text{sz}(\psi) = \text{sz}(\varphi) - 1 \leq k-1$  so by induction hypothesis S has a winning strategy for the game  $\text{FS}_{k-1}^\varphi(\diamond_f\mathbb{A}, \square\mathbb{B})$ . Since  $k \geq \text{sz}(\varphi) \geq 1$ , S can start the game  $\text{FS}_k^\varphi(\mathbb{A}, \mathbb{B})$  with a  $\diamond$ -move choosing the function  $f$ . Then S wins the game by following the winning strategy for the game  $\text{FS}_{k-1}^\varphi(\diamond_f\mathbb{A}, \square\mathbb{B})$ .

## 10 Formula size games for modal logic and $\mu$ -calculus

- (e) Assume finally that  $\varphi = \Box\psi$ . Since  $\mathbb{A} \models \varphi$ , as in the previous case, we obtain  $\Box\mathbb{A} \models \psi$ . On the other hand, since  $\mathbb{B} \models \neg\varphi$ , for every  $(\mathcal{B}, w) \in \mathbb{B}$  there is  $(\mathcal{B}, v_w) \in \Box(\mathcal{B}, w)$  such that  $(\mathcal{B}, v_w) \not\models \psi$ . We define the function  $g : \mathbb{B} \rightarrow \Box\mathbb{B}$  by  $g(\mathcal{B}, w) = (\mathcal{B}, v_w)$ . Clearly,  $\Diamond_g\mathbb{B} \models \neg\psi$  so the formula  $\psi$  separates the sets  $\Box\mathbb{A}$  and  $\Diamond_g\mathbb{B}$ . By induction hypothesis, S has a winning strategy for the game  $\text{FS}_{k-1}^\Phi(\Box\mathbb{A}, \Diamond_g\mathbb{B})$ . S wins the game  $\text{FS}_k^\Phi(\mathbb{A}, \mathbb{B})$  by starting with a  $\Box$ -move choosing the function  $g$  and proceeding according to the winning strategy of the game  $\text{FS}_{k-1}^\Phi(\Box\mathbb{A}, \Diamond_g\mathbb{B})$ .  $\square$

### REMARK 3.3

In this form, the game  $\text{FS}_k^\Phi(\mathbb{A}, \mathbb{B})$  tracks the size of the separating formula but with slight modifications it could track different things such as the number or nesting depth of specific operators. See e.g. the conference paper [18] where the game counts propositional connectives and modal operators with two separate parameters.

Note that in Theorem 3.2 we allow the set of propositional symbols  $\Phi$  to be infinite. This is in contrast with other similar games, such as the bisimulation game and the  $n$ -bisimulation game. For an example of two models, which satisfy the same  $\text{ML}(\Phi)$ -formulas for an infinite  $\Phi$ , but are not bisimilar, see [2, Figure 2.5, p. 68].

We prove next that  $k$ -bisimilarity implies that D has winning strategy in the formula size game with resource parameter  $k$ . This simple observation is used in the next section, when we apply the game  $\text{FS}_k^\Phi$  for proving a succinctness result for FO over ML.

### THEOREM 3.4

Let  $\mathbb{A}$  and  $\mathbb{B}$  be sets of pointed models and let  $k \in \mathbb{N}$ . If there are  $(k-1)$ -bisimilar pointed models  $(\mathcal{A}, w) \in \mathbb{A}$  and  $(\mathcal{B}, v) \in \mathbb{B}$ , then D has a winning strategy for the game  $\text{FS}_k^\Phi(\mathbb{A}, \mathbb{B})$ .

PROOF. The proof proceeds by induction on the number  $k \in \mathbb{N}$ . If  $k = 1$  and  $(\mathcal{A}, w) \in \mathbb{A}$  and  $(\mathcal{B}, v) \in \mathbb{B}$  are 0-bisimilar and thus satisfy the same literals. Thus, there is no literal  $\varphi \in \text{ML}$  that separates the sets  $\mathbb{A}$  and  $\mathbb{B}$ . Thus, any literal move by S leads to D winning. In addition, any non-literal move leads to a following position with  $k = 0$  so D wins the game  $\text{FS}_1^\Phi(\mathbb{A}, \mathbb{B})$ .

Assume that  $k > 1$  and  $(\mathcal{A}, w) \in \mathbb{A}$  and  $(\mathcal{B}, v) \in \mathbb{B}$  are  $(k-1)$ -bisimilar. We consider the cases of the first move of S in the game  $\text{FS}_k^\Phi(\mathbb{A}, \mathbb{B})$ .

If S makes a literal move, D will win as in the basic step.

If S starts with a  $\vee$ -move choosing the numbers  $k_1$  and  $k_2$  and the sets  $\mathbb{A}_1$  and  $\mathbb{A}_2$ , then since  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$ , D can choose the next position  $(k_i, \mathbb{A}_i, \mathbb{B})$ , in such a way that  $(\mathcal{A}, w) \in \mathbb{A}_i$ . Then we have  $k_i < k$  so by induction hypothesis D has a winning strategy for the game  $\text{FS}_{k_i}^\Phi(\mathbb{A}_i, \mathbb{B})$ . The case of a  $\wedge$ -move is similar.

If S starts with a  $\Diamond$ -move choosing a function  $f : \mathbb{A} \rightarrow \Box\mathbb{A}$ , then since  $(\mathcal{A}, w)$  and  $(\mathcal{B}, v)$  are  $(k-1)$ -bisimilar, there is a pointed model  $(\mathcal{B}, v') \in \Box(\mathcal{B}, v)$  that is  $(k-2)$ -bisimilar with the pointed model  $f(\mathcal{A}, w)$ . By induction hypothesis, D has a winning strategy in  $\text{FS}_{k-1}^\Phi(\Diamond_f\mathbb{A}, \Box\mathbb{B})$ . The case of a  $\Box$ -move is similar.  $\square$

## 4 Succinctness of FO over ML

In this section, we illustrate the use of the formula size game  $\text{FS}_k^\Phi$  by proving a non-elementary succinctness gap between bisimulation invariant first-order logic and modal logic. We also show that this gap is already present between a limited two-dimensional modal logic  $\text{ML}^2$  and basic modal logic.

A similar gap between FO and linear temporal logic LTL has already been established in the literature. In his PhD thesis [29], Stockmeyer proved that the satisfiability problem of FO over words is of non-elementary complexity. He reduced the problem of nonemptiness of star-free regular expressions to this satisfiability problem. Etessami and Wilke pointed out in [5] that careful examination of Stockmeyer's proof yields FO sentences with size  $\mathcal{O}(n)$  such that the minimal words satisfying these sentences have length non-elementary in  $n^3$ . Since all satisfiable formulas of LTL have a satisfying model at most exponential in the size of the formula, a non-elementary succinctness gap between FO and LTL is obtained.

#### 4.1 A property of pointed models

For the remainder of this section, we consider only the case where the set  $\Phi$  of propositional symbols is empty. This makes all points in Kripke models propositionally equivalent so the only formulas available for the win condition of S in the game  $\text{FS}_k^\Phi$  are  $\perp$  and  $\top$ . Thus, S can only win with a literal move from position  $(k, \mathbb{A}, \mathbb{B})$  if either  $\mathbb{A} = \emptyset$  and  $\mathbb{B} \neq \emptyset$ , or  $\mathbb{A} \neq \emptyset$  and  $\mathbb{B} = \emptyset$ .

We will use the following two classes in our application of the formula size game  $\text{FS}_k^\Phi$ :

- $\mathbb{A}_n$  is the class of all pointed models  $(\mathcal{A}, w)$  such that for all  $(\mathcal{A}, u), (\mathcal{A}, v) \in \square(\mathcal{A}, w)$ , the models  $(\mathcal{A}, u)$  and  $(\mathcal{A}, v)$  are  $n$ -bisimilar.
- $\mathbb{B}_n$  is the complement of  $\mathbb{A}_n$ .

##### LEMMA 4.1

For each  $n \in \mathbb{N}$ , there is a formula  $\varphi_n(x) \in \text{FO}$  that separates the classes  $\mathbb{A}_n$  and  $\mathbb{B}_n$  such that the size of  $\varphi_n(x)$  is linear with respect to  $n$ , i.e.  $\text{sz}(\varphi_n) = \mathcal{O}(n)$ .

PROOF. We first define formulas  $\psi_n(x, y) \in \text{FO}$  such that  $(\mathcal{M}, u) \simeq_n (\mathcal{M}, v)$  if and only if  $\mathcal{M} \models \psi_n[u/x, v/y]$ . We only use four variables. To make our reuse of variables explicit, we define formulas  $\psi_n(x, y)$  and  $\psi'_n(s, t)$  via mutual recursion as follows:

$$\begin{aligned} \psi_1(x, y) &:= \exists s R(x, s) \leftrightarrow \exists t R(y, t) \\ \psi'_1(s, t) &:= \exists x R(x, s) \leftrightarrow \exists y R(t, y) \\ \psi_{n+1}(x, y) &:= \forall s \exists t \left( (R(x, s) \rightarrow R(y, t)) \wedge (R(y, s) \rightarrow R(x, t)) \right. \\ &\quad \left. \wedge (R(x, s) \vee R(y, s) \rightarrow \psi'_n(s, t)) \right) \\ \psi'_{n+1}(s, t) &:= \forall x \exists y \left( (R(s, x) \rightarrow R(t, y)) \wedge (R(s, y) \rightarrow R(t, x)) \right. \\ &\quad \left. \wedge (R(s, x) \vee R(s, y) \rightarrow \psi_n(x, y)) \right) \end{aligned}$$

Clearly, the formulas  $\psi_n(x, y)$  express  $n$ -bisimilarity as intended. When we interpret the equivalence and implications as shorthand in the standard way, we get the sizes  $\text{sz}(\psi_1) = 11$  and  $\text{sz}(\psi_{n+1}) = \text{sz}(\psi_n) + 14$ . Thus,  $\text{sz}(\psi_n) = 14n - 3$ .

Now we can define the formulas  $\varphi_n$ :

$$\varphi_n(x) := \forall y \forall z (R(x, y) \wedge R(x, z) \rightarrow \psi_n(y, z)).$$

12 *Formula size games for modal logic and  $\mu$ -calculus*

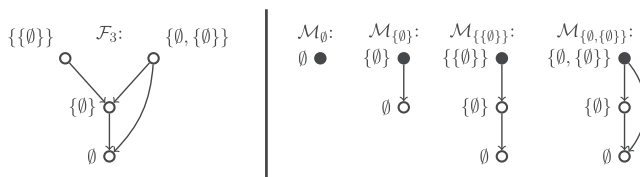


FIGURE 1 The model  $\mathcal{F}_3$  and its generated submodels.

Clearly, for every  $(\mathcal{A}, w) \in \mathbb{A}_n$  we have  $\mathcal{A} \models \varphi_n[w/x]$  and for every  $(\mathcal{B}, v) \in \mathbb{B}_n$  we have  $\mathcal{B} \models \neg\varphi_n[w/x]$  so the formula  $\varphi_n$  separates the classes  $\mathbb{A}_n$  and  $\mathbb{B}_n$ . Furthermore,  $\text{sz}(\varphi_n) = \text{sz}(\psi_n) + 6 = 14n + 3$  so the size of  $\varphi_n$  is linear with respect to  $n$ .  $\square$

LEMMA 4.2

For each  $n \in \mathbb{N}$ , the formula  $\varphi_n$  is  $(n + 1)$ -bisimulation invariant.

PROOF. Let  $(\mathcal{A}, w)$  and  $(\mathcal{B}, v)$  be  $(n + 1)$ -bisimilar pointed models. Assume that  $\mathcal{A} \models \varphi_n[w/x]$ . If  $(\mathcal{B}, v_1), (\mathcal{B}, v_2) \in \square(\mathcal{B}, v)$ , by  $(n + 1)$ -bisimilarity there are  $(\mathcal{A}, w_1), (\mathcal{A}, w_2) \in \square(\mathcal{A}, w)$  such that  $(\mathcal{A}, w_1) \simeq_n (\mathcal{B}, v_1)$  and  $(\mathcal{A}, w_2) \simeq_n (\mathcal{B}, v_2)$ . Since  $\mathcal{A} \models \varphi_n[w/x]$ , we have  $(\mathcal{B}, v_1) \simeq_n (\mathcal{A}, w_1) \simeq_n (\mathcal{A}, w_2) \simeq_n (\mathcal{B}, v_2)$  so  $\mathcal{B} \models \psi_n[v_1/x, v_2/y]$ . Thus, we see that  $\mathcal{B} \models \varphi_n[v/x]$ .  $\square$

It follows now from van Benthem’s characterization theorem that each  $\varphi_n$  is equivalent to some ML-formula. Thus, we get the following corollary.

COROLLARY 4.3

For each  $n \in \mathbb{N}$ , there is a formula  $\vartheta_n \in \text{ML}$  that separates the classes  $\mathbb{A}_n$  and  $\mathbb{B}_n$ .

4.2 *Set theoretic construction of pointed models*

We have shown that the classes  $\mathbb{A}_n$  and  $\mathbb{B}_n$  can be separated both in ML and in FO. Furthermore the size of the FO-formula is linear with respect to  $n$ . It only remains to ask: what is the size of the smallest ML-formula that separates the classes  $\mathbb{A}_n$  and  $\mathbb{B}_n$ ? To answer this, we will need suitable subsets of  $\mathbb{A}_n$  and  $\mathbb{B}_n$  to play the formula size game on.

DEFINITION 4.4

Let  $n \in \mathbb{N}$ . The finite levels of the cumulative hierarchy are defined recursively as follows:

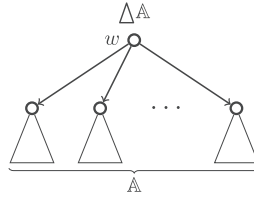
$$\begin{aligned} \mathbb{V}_0 &= \emptyset \\ \mathbb{V}_{n+1} &= \mathcal{P}(\mathbb{V}_n) \end{aligned}$$

For every  $n \in \mathbb{N}$ ,  $\mathbb{V}_n$  is a transitive set, i.e. for every  $a \in \mathbb{V}_n$  and every  $b \in a$  it holds that  $b \in \mathbb{V}_n$ . Thus, it is reasonable to define a model  $\mathcal{F}_n = (\mathbb{V}_n, R_n)$ , where for all  $a, b \in \mathbb{V}_n$  it holds that  $(a, b) \in R_n \Leftrightarrow b \in a$ .

For every point  $a \in \mathbb{V}_n$ , we denote by  $(\mathcal{M}_a, a)$  the pointed model, where  $\mathcal{M}_a$  is the submodel of  $\mathcal{F}_n$  generated by the point  $a$ .

LEMMA 4.5

Let  $n \in \mathbb{N}$  and  $a, b \in \mathbb{V}_{n+1}$ . If  $a \neq b$ , then  $(\mathcal{M}_a, a) \not\simeq_n (\mathcal{M}_b, b)$ .

FIGURE 2 The pointed model  $\Delta\mathbb{A}$ .

PROOF. We prove the claim by induction on  $n$ . The basic step  $n = 0$  is trivial since  $\mathbb{V}_1$  only has one element. For the induction step, assume that  $a, b \in \mathbb{V}_{n+1}$  and  $a \neq b$ . Assume further for contradiction that  $(\mathcal{M}_a, a) \simeq_n (\mathcal{M}_b, b)$ . Since  $a \neq b$ , by symmetry we can assume that there is  $x \in a$  such that  $x \notin b$ . By  $n$ -bisimilarity there is  $y \in b$  such that  $(\mathcal{M}_x, x)$  and  $(\mathcal{M}_y, y)$  are  $(n-1)$ -bisimilar. Since  $x \in a \in \mathbb{V}_{n+1}$  and  $y \in b \in \mathbb{V}_{n+1}$ , we have  $x, y \in \mathbb{V}_n$ . By induction hypothesis, we obtain  $x = y$ . This is a contradiction, since  $x \notin b$  and  $y \in b$ .  $\square$

If  $\mathbb{A}$  is a set of pointed models, the pointed model  $\Delta\mathbb{A}$  is formed by taking all the pointed models of  $\mathbb{A}$  and connecting a new root to their distinguished points as illustrated in Figure 2. To make sure that  $(\Delta\mathbb{A}, w)$  is bisimilar with  $(\mathcal{A}, v)$  for any  $(\mathcal{A}, v) \in \mathbb{A}$ , we require that the models in  $\mathbb{A}$  are compatible in possible intersections. The precise definition is the following.

Let  $\mathbb{A}$  be a set of pointed models. For all  $(\mathcal{A}, v), (\mathcal{A}', v') \in \mathbb{A}$ , let  $D(\mathcal{A}, \mathcal{A}') = \text{dom}(\mathcal{A}) \cap \text{dom}(\mathcal{A}')$  and assume that  $R^{\mathcal{A}} \cap (D(\mathcal{A}, \mathcal{A}') \times D(\mathcal{A}, \mathcal{A}')) = R^{\mathcal{A}'} \cap (D(\mathcal{A}, \mathcal{A}') \times D(\mathcal{A}, \mathcal{A}'))$ . Let  $w \notin \text{dom}(\mathcal{A})$  for all  $(\mathcal{A}, v) \in \mathbb{A}$ . We use the notation  $\Delta\mathbb{A} := (\mathcal{M}, w)$ , where

$$\text{dom}(\mathcal{M}) = \{w\} \cup \bigcup \{\text{dom}(\mathcal{A}) \mid (\mathcal{A}, v) \in \mathbb{A}\}, \text{ and}$$

$$R^{\mathcal{M}} = \{(w, v) \mid (\mathcal{A}, v) \in \mathbb{A}\} \cup \bigcup \{R^{\mathcal{A}} \mid (\mathcal{A}, v) \in \mathbb{A}\}.$$

For each  $n \in \mathbb{N}$ , we define the following sets of pointed models:

$$\mathbb{C}_n := \{\Delta\{(\mathcal{M}_a, a) \mid a \in \mathbb{V}_{n+1}\}\}$$

$$\mathbb{D}_n := \{\Delta\{(\mathcal{M}_a, a), (\mathcal{M}_b, b)\} \mid a, b \in \mathbb{V}_{n+1}, a \neq b\}.$$

In other words, the pointed models in  $\mathbb{C}_n$  have a single successor from level  $n+1$  of the cumulative hierarchy, whereas the pointed models in  $\mathbb{D}_n$  have two different successors from the same set. Therefore, clearly  $\mathbb{C}_n \subseteq \mathbb{A}_n$  and by Lemma 4.5 also  $\mathbb{D}_n \subseteq \mathbb{B}_n$ . In the next subsection, we will use these sets in the formula size game.

It is well known that the cardinality of  $\mathbb{V}_n$  is the exponential tower of  $n-1$ . Thus, the cardinality of  $\mathbb{C}_n$  is  $\text{twr}(n)$ .

LEMMA 4.6

If  $n \in \mathbb{N}$ , we have  $|\mathbb{C}_n| = |\mathbb{V}_{n+1}| = \text{twr}(n)$ .

#### 4.3 Graph colourings and winning strategies in $FS_k^\Phi$

Our aim is to prove that any ML-formula  $\vartheta_n$  separating the sets  $\mathbb{C}_n$  and  $\mathbb{D}_n$  is of size at least  $\text{twr}(n-1)$ . To do this, we make use of a surprising connection between the chromatic numbers of certain graphs

14 *Formula size games for modal logic and  $\mu$ -calculus*

related to pairs of the form  $(\mathbb{V}, \mathbb{E})$ , where  $\mathbb{V} \subseteq \mathbb{C}_n$  and  $\mathbb{E} \subseteq \mathbb{D}_n$ , and existence of a winning strategy for D in the game  $\text{FS}_k^\phi(\mathbb{V}, \mathbb{E})$ .

Let  $n \in \mathbb{N}$ ,  $\emptyset \neq \mathbb{V} \subseteq \mathbb{C}_n$  and  $\mathbb{E} \subseteq \mathbb{D}_n$ . Then  $\mathcal{G}(\mathbb{V}, \mathbb{E})$  denotes the graph  $(V, E)$ , where

$$V = \square \mathbb{V} \text{ and}$$

$$E = \{((\mathcal{M}, w), (\mathcal{M}', w')) \in V \times V \mid \Delta\{(\mathcal{M}, w), (\mathcal{M}', w')\} \in \mathbb{E}\}.$$

That is, since models on the left all have exactly one successor, and ones on the right have exactly two successors from the same basic set, we can take the graph where these successors are nodes and the pairs on the right define the edges. Note that a pair on the right only produces an edge if both elements of the pair are present on the left.

## DEFINITION 4.7

Let  $\mathcal{G} = (V, E)$  be a graph and let  $C$  be a set. A function  $\chi : V \rightarrow C$  is a *colouring* of the graph  $\mathcal{G}$  if for all  $u, v \in V$  it holds that if  $(u, v) \in E$ , then  $\chi(u) \neq \chi(v)$ . If the set  $C$  has  $k$  elements, then  $\chi$  is called a *k-colouring* of  $\mathcal{G}$ .

The *chromatic number* of  $\mathcal{G}$ , denoted by  $\chi(\mathcal{G})$ , is the smallest number  $k \in \mathbb{N}$  for which there is a *k-colouring* of  $\mathcal{G}$ .

When playing the formula size game  $\text{FS}_k^\phi(\mathbb{V}, \mathbb{E})$ , connective moves correspond with dividing either the vertex set or the edge set of the graph  $\mathcal{G}(\mathbb{V}, \mathbb{E})$  into two parts, forming two new graphs. In the next lemma, we get simple arithmetic estimates for the behaviour of chromatic numbers in such divisions. In the case of a vertex set split, if the two new graphs are coloured with separate colours, combining these colourings yields a colouring of the whole graph. For an edge split, the full graph is coloured with pairs of colours given by the two new colourings. If two vertices are adjacent in the full graph, at least one of the new colourings will colour them with a different colour and the pairs of colours will be different.

## LEMMA 4.8

Let  $\mathcal{G} = (V, E)$  be a graph.

1. Let  $V_1, V_2 \subseteq V$  be nonempty such that  $V_1 \cup V_2 = V$  and let  $\mathcal{G}_1 = (V_1, E \cap (V_1 \times V_1))$  and  $\mathcal{G}_2 = (V_2, E \cap (V_2 \times V_2))$ . Then we have  $\chi(\mathcal{G}) \leq \chi(\mathcal{G}_1) + \chi(\mathcal{G}_2)$ .
2. Let  $E_1, E_2 \subseteq E$  such that  $E_1 \cup E_2 = E$  and let  $\mathcal{G}_1 = (V, E_1)$  and  $\mathcal{G}_2 = (V, E_2)$ . Then  $\chi(\mathcal{G}) \leq \chi(\mathcal{G}_1)\chi(\mathcal{G}_2)$ .

## PROOF.

1. Let  $V_1, V_2, \mathcal{G}_1$  and  $\mathcal{G}_2$  be as in the claim and let  $k_1 = \chi(\mathcal{G}_1)$  and  $k_2 = \chi(\mathcal{G}_2)$ . Let  $\chi_1 : V_1 \rightarrow \{1, \dots, k_1\}$  be a  $k_1$ -colouring of the graph  $\mathcal{G}_1$  and let  $\chi_2 : V_2 \rightarrow \{k_1 + 1, \dots, k_1 + k_2\}$  be a  $k_2$ -colouring of the graph  $\mathcal{G}_2$ . Then it is straightforward to show that  $\chi = \chi_1 \cup (\chi_2 \upharpoonright (V_2 \setminus V_1))$  is a  $k_1 + k_2$ -colouring of the graph  $\mathcal{G}$ , whence  $\chi(\mathcal{G}) \leq k_1 + k_2 = \chi(\mathcal{G}_1) + \chi(\mathcal{G}_2)$ .
2. Let  $\chi_1 : V \rightarrow \{1, \dots, k_1\}$  and  $\chi_2 : V \rightarrow \{1, \dots, k_2\}$  be colourings of the graphs  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , respectively. Then it is easy to verify that the map  $\chi : V \rightarrow \{1, \dots, k_1\} \times \{1, \dots, k_2\}$  defined by  $\chi(v) = (\chi_1(v), \chi_2(v))$  is a colouring of  $\mathcal{G}$ . Thus, we obtain  $\chi(\mathcal{G}) \leq |\{1, \dots, k_1\} \times \{1, \dots, k_2\}| = \chi(\mathcal{G}_1)\chi(\mathcal{G}_2)$ .  $\square$

For the condition, D maintains to win the game, we use the logarithm of the chromatic number of  $\mathcal{G}(\mathbb{V}, \mathbb{E})$  as it behaves nicely with both kinds of splittings. Note that to achieve non-elementary



formula size, it suffices to consider the number of binary connectives required before any modal moves can be made.

LEMMA 4.9

Assume  $\emptyset \neq \mathbb{V} \subseteq \mathbb{C}_n$  and  $\mathbb{E} \subseteq \mathbb{D}_n$  for some  $n \in \mathbb{N}$  and let  $k \in \mathbb{N}$ . If  $k \leq \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E})))$ , then D has a winning strategy in the game  $\text{FS}_k^\Phi(\mathbb{V}, \mathbb{E})$ .

PROOF. Let  $n, k \in \mathbb{N}$  and assume that  $\emptyset \neq \mathbb{V} \subseteq \mathbb{C}_n, \mathbb{E} \subseteq \mathbb{D}_n$  and  $k \leq \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E})))$ . We prove the claim by induction on  $k$ .

If  $k = 0$ , then D wins the game.

If  $k = 1$ , any non-literal move of S leads to D winning. Since  $\mathbb{V}, \mathbb{E} \neq \emptyset$  and all models are propositionally equivalent, D will also win if S makes a literal move.

Assume then that  $k > 1$ . If S starts the game with a literal move, then D wins as described above.

Assume that S begins the game with a  $\diamond$ - or  $\square$ -move. Since  $\chi(\mathcal{G}(\mathbb{V}, \mathbb{E})) \geq 2$ , there are pointed models  $(\mathcal{M}, w), (\mathcal{M}', w') \in V$  such that  $((\mathcal{M}, w), (\mathcal{M}', w')) \in E$ . Thus,  $\Delta\{(\mathcal{M}, w)\}, \Delta\{(\mathcal{M}', w')\} \in \mathbb{V}$  and  $\Delta\{(\mathcal{M}, w), (\mathcal{M}', w')\} \in \mathbb{E}$ . In the following position  $(k-1, \mathbb{V}', \mathbb{E}')$  it holds that  $(\mathcal{M}, w) \in \mathbb{V}' \cap \mathbb{E}'$  or  $(\mathcal{M}', w') \in \mathbb{V}' \cap \mathbb{E}'$ . Thus, the same pointed model is present on both sides of the game and by Theorem 3.4, D has a winning strategy for the game  $\text{FS}_{k-1}^\Phi(\mathbb{V}', \mathbb{E}')$ .

Assume that S begins the game with a  $\vee$ -move choosing the numbers  $k_1, k_2 \in \mathbb{N}$  and the sets  $\mathbb{V}_1, \mathbb{V}_2 \subseteq \mathbb{V}$ . Consider the graphs  $\mathcal{G}(\mathbb{V}, \mathbb{E}) = (V, E)$  and  $\mathcal{G}(\mathbb{V}_i, \mathbb{E}) = (V_i, E_i)$ . Since  $\mathbb{V}_1 \cup \mathbb{V}_2 = \mathbb{V}$ , we have  $V_1 \cup V_2 = V$ . In addition, by the definition of the graphs  $\mathcal{G}(\mathbb{V}, \mathbb{E})$  and  $\mathcal{G}(\mathbb{V}_i, \mathbb{E})$  we see that  $E_i = E \cap (V_i \times V_i)$ . Thus by Lemma 4.8, we obtain  $\chi(\mathcal{G}(\mathbb{V}, \mathbb{E})) \leq \chi(\mathcal{G}(\mathbb{V}_1, \mathbb{E})) + \chi(\mathcal{G}(\mathbb{V}_2, \mathbb{E}))$ . It must hold that  $k_1 \leq \log(\chi(\mathcal{G}(\mathbb{V}_1, \mathbb{E})))$  or  $k_2 \leq \log(\chi(\mathcal{G}(\mathbb{V}_2, \mathbb{E})))$ , since otherwise we would have

$$\begin{aligned} k &\leq \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}))) \leq \log(\chi(\mathcal{G}(\mathbb{V}_1, \mathbb{E})) + \chi(\mathcal{G}(\mathbb{V}_2, \mathbb{E}))) \\ &\leq \log(\chi(\mathcal{G}(\mathbb{V}_1, \mathbb{E}))) + \log(\chi(\mathcal{G}(\mathbb{V}_2, \mathbb{E}))) + 1 < k_1 + k_2 + 1 = k. \end{aligned}$$

Thus, D can choose the next position of the game,  $(k_i, \mathbb{V}_i, \mathbb{E})$ , in such a way that  $k_i \leq \log(\chi(\mathcal{G}(\mathbb{V}_i, \mathbb{E})))$ . By induction hypothesis, D has a winning strategy in the game  $\text{FS}_{k_i}^\Phi(\mathbb{V}_i, \mathbb{E})$ .

Assume then that S begins the game with a  $\wedge$ -move choosing the numbers  $k_1, k_2 \in \mathbb{N}$  and the sets  $\mathbb{E}_1, \mathbb{E}_2 \subseteq \mathbb{E}$ . Consider now the graphs  $\mathcal{G}(\mathbb{V}, \mathbb{E}) = (V, E)$  and  $\mathcal{G}(\mathbb{V}, \mathbb{E}_i) = (V, E_i)$ . Clearly  $V_1 = V_2 = V$  and since  $\mathbb{E}_1 \cup \mathbb{E}_2 = \mathbb{E}$ , we have  $E_1 \cup E_2 = E$ . Thus by Lemma 4.8, we obtain  $\chi(\mathcal{G}(\mathbb{V}, \mathbb{E})) \leq \chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_1))\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_2))$ . It must hold that  $k_1 \leq \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_1)))$  or  $k_2 \leq \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_2)))$ , since otherwise we would have

$$\begin{aligned} k &\leq \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}))) \leq \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_1))\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_2))) \\ &= \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_1))) + \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_2))) < k_1 + k_2 + 1 = k. \end{aligned}$$

Thus, D can again choose the next position of the game,  $(k_i, \mathbb{V}, \mathbb{E}_i)$ , in such a way that  $k_i \leq \log(\chi(\mathcal{G}(\mathbb{V}, \mathbb{E}_i)))$ . By induction hypothesis, D has a winning strategy in the game  $\text{FS}_{k_i}^\Phi(\mathbb{V}, \mathbb{E}_i)$ .  $\square$

THEOREM 4.10

Let  $n \in \mathbb{N}$ . If a formula  $\vartheta_n \in \text{ML}$  separates  $\mathbb{A}_n$  from  $\mathbb{B}_n$ , then  $\text{sz}(\vartheta_n) > \text{twr}(n-1)$ .

PROOF. Assume that a formula  $\vartheta_n \in \text{ML}$  separates  $\mathbb{A}_n$  from  $\mathbb{B}_n$ . As observed in the end of Subsection 4.2, it holds that  $\mathbb{C}_n \subseteq \mathbb{A}_n$  and  $\mathbb{D}_n \subseteq \mathbb{B}_n$ . Therefore,  $\vartheta_n$  also separates the sets  $\mathbb{C}_n$  and  $\mathbb{D}_n$ .

Assume for contradiction that  $\text{sz}(\vartheta_n) \leq \text{twr}(n-1)$ . By Theorem 3.2, S has a winning strategy in the game  $\text{FS}_k^\Phi(\mathbb{C}_n, \mathbb{D}_n)$  for  $k = \text{sz}(\vartheta_n)$ .

On the other hand, by Lemma 4.6, we have  $|\mathbb{C}_n| = \text{twr}(n)$  and the set  $\mathbb{D}_n$  consists of all the pointed models  $\Delta\{(\mathcal{M}, w), (\mathcal{M}', w')\}$ , where  $\Delta\{(\mathcal{M}, w)\}, \Delta\{(\mathcal{M}', w')\} \in \mathbb{C}_n, (\mathcal{M}, w) \neq (\mathcal{M}', w')$ . Thus, the

## 16 Formula size games for modal logic and $\mu$ -calculus

graph  $\mathcal{G}(\mathbb{C}_n, \mathbb{D}_n)$  is isomorphic with the complete graph  $K_{\text{twr}(n)}$ . Therefore, we obtain

$$\chi(\mathcal{G}(\mathbb{C}_n, \mathbb{D}_n)) = \chi(K_{\text{twr}(n)}) = \text{twr}(n).$$

By the assumption,  $k \leq \text{twr}(n-1) = \log(\text{twr}(n)) = \log(\chi(\mathcal{G}(\mathbb{C}_n, \mathbb{D}_n)))$ , so by Lemma 4.9, D also has a winning strategy in the game  $\text{FS}_k^\Phi(\mathbb{C}_n, \mathbb{D}_n)$ , which is a contradiction.  $\square$

We now have everything we need for proving the non-elementary succinctness of FO over ML. By Lemma 4.1, for each  $n \in \mathbb{N}$  there is a formula  $\varphi_n(x) \in \text{FO}$  such that  $\varphi_n$  separates the classes  $\mathbb{A}_n$  and  $\mathbb{B}_n$  with  $s(\varphi) = \mathcal{O}(n)$ . On the other hand by Corollary 4.3, there is an equivalent formula  $\vartheta_n \in \text{ML}$ , but by Theorem 4.10 the size of  $\vartheta_n$  must be at least  $\text{twr}(n-1)$ . So the property of all successors of a pointed model being  $n$ -bisimilar with each other can be expressed in FO with a formula of linear size, but in ML expressing it requires a formula of non-elementary size.

### COROLLARY 4.11

Bisimulation invariant FO is non-elementarily more succinct than ML.

### REMARK 4.12

It is well known that the DAG-size of any formula  $\varphi$  is greater than or equal to the logarithm of the size of  $\varphi$ . Thus, if  $\vartheta_n$  is a formula as in Theorem 4.10, the DAG-size of  $\vartheta_n$  must be at least  $\text{twr}(n-2)$ . Consequently the result of Corollary 4.11 also holds for DAG-size.

## 4.4 Succinctness of two-dimensional modal logic

Our proof for the non-elementary succinctness gap between bisimulation invariant FO and ML is based on the fact that  $n$ -bisimilarity of two points  $u, v \in W$  of a Kripke model  $\mathcal{M} = (W, R)$  is definable by a linear FO-formula  $\psi_n(x, y)$  (see the proof of Lemma 4.1). We will now show that the property  $(\mathcal{M}, u) \rightleftharpoons_n (\mathcal{M}, v)$  is succinctly expressible also in *two-dimensional modal logic*.

The idea in two-dimensional modal logic is that the truth of formulas is evaluated on pairs  $(u, v)$  of points of Kripke models instead of single points. We refer to the book [24] of Marx and Venema and the series of papers [10–12] of Gabbay and Shehtman for a detailed exposition on two-dimensional and multi-dimensional modal logics. For our purposes, it suffices to consider the logic Gabbay and Shehtman call  $\mathbf{K}^2$ . For consistency of notation in this paper, we call the logic  $\text{ML}^2$  and introduce it only semantically.

A Kripke model  $\mathcal{T}$  for  $\text{ML}^2$  consists of a set  $W$  of points, two binary accessibility relations  $R_1$  and  $R_2$ , and a valuation  $V : \Phi \rightarrow \mathcal{P}(W^2)$ . Correspondingly,  $\text{ML}^2$  has two modal operators  $\diamond_1, \diamond_2$  and their duals  $\square_1, \square_2$ . The connectives  $\vee, \wedge$  and  $\neg$  are defined in the standard way. The rest of the semantics is defined as follows:

- $(\mathcal{T}, (u, v)) \models p \Leftrightarrow (u, v) \in V(p)$  for  $p \in \Phi$ ,
- $(\mathcal{T}, (u, v)) \models \diamond_1 \varphi \Leftrightarrow$  there is  $u' \in W$  such that  $uR_1u'$  and  $(\mathcal{T}, (u', v)) \models \varphi$ ,
- $(\mathcal{T}, (u, v)) \models \diamond_2 \varphi \Leftrightarrow$  there is  $v' \in W$  such that  $vR_2v'$  and  $(\mathcal{T}, (u, v')) \models \varphi$ ,
- $(\mathcal{T}, (u, v)) \models \square_1 \varphi \Leftrightarrow$  for all  $u' \in W$ , if  $uR_1u'$ , then  $(\mathcal{T}, (u', v)) \models \varphi$ ,
- $(\mathcal{T}, (u, v)) \models \square_2 \varphi \Leftrightarrow$  for all  $v' \in W$ , if  $vR_2v'$ , then  $(\mathcal{T}, (u, v')) \models \varphi$ .

Any pointed Kripke model  $(\mathcal{M}, w) = ((W, R, V), w)$  can be interpreted as the two-dimensional pointed model  $(\mathcal{M}_2, (w, w))$ , where  $\mathcal{M}_2 = (W, R, R, V)$ . This gives us a meaningful way of defining properties of pointed models  $(\mathcal{M}, w)$  by formulas of  $\text{ML}^2$ . In particular, we say that a formula  $\varphi \in \text{ML}^2$  separates two classes  $\mathbb{A}$  and  $\mathbb{B}$  of pointed models if for all  $(\mathcal{M}, w) \in \mathbb{A}$ ,  $(\mathcal{M}_2, (w, w)) \models \varphi$  and for all  $(\mathcal{M}, w) \in \mathbb{B}$ ,  $(\mathcal{M}_2, (w, w)) \not\models \varphi$ .

The *size*  $\text{sz}(\varphi)$  of a formula  $\varphi \in \text{ML}^2$  is defined in the same way as for formulas of ML, see Definition 2.3. In other words,  $\text{sz}(\varphi)$  is the total number of modal operators, binary connectives and literals occurring in  $\varphi$ .

Observe now that two pointed models  $(\mathcal{M}, u)$  and  $(\mathcal{M}, v)$  with no propositional symbols are 1-bisimilar if and only if  $(\mathcal{M}_2, (u, v)) \models \rho_1$ , where  $\rho_1 := \Diamond_1 \top \leftrightarrow \Diamond_2 \top$ . Furthermore if  $\rho_n \in \text{ML}_2$  defines the class of all two-dimensional pointed models  $(\mathcal{M}_2, (u, v))$  such that  $(\mathcal{M}, u) \simeq_n (\mathcal{M}, v)$ , then  $\rho_{n+1} := \Box_1 \Diamond_2 \rho_n \wedge \Box_2 \Diamond_1 \rho_n$  defines the class of all  $(\mathcal{M}_2, (u, v))$  such that  $(\mathcal{M}, u) \simeq_{n+1} (\mathcal{M}, v)$ .

LEMMA 4.13

For each  $n \in \mathbb{N}$ , there is a formula  $\zeta_n \in \text{ML}^2$  that separates the classes  $\mathbb{A}_n$  and  $\mathbb{B}_n$  such that the size of  $\zeta_n$  is exponential with respect to  $n$ , i.e.  $\text{sz}(\zeta_n) = \mathcal{O}(2^n)$ .

PROOF. Let  $\zeta_n$  be the formula  $\Box_1 \Box_2 \rho_n$ . Then  $(\mathcal{M}_2, (w, w)) \models \zeta_n$  if and only if  $(\mathcal{M}, u)$  and  $(\mathcal{M}, v)$  are  $n$ -bisimilar for all  $(\mathcal{M}, u), (\mathcal{M}, v) \in \Box(\mathcal{M}, w)$ , whence  $\zeta_n$  separates  $\mathbb{A}_n$  from its complement  $\mathbb{B}_n$ . An easy calculation shows that the size of  $\zeta_n$  is  $2^{n+4} - 3$ .  $\square$

By Theorem 4.3, for each  $n \in \mathbb{N}$  there is a formula  $\vartheta_n \in \text{ML}$  that is equivalent with  $\zeta_n$ . On the other hand, by Theorem 4.10 the size of  $\vartheta_n$  is at least  $\text{twr}(n-1)$ . Thus, we obtain the non-elementary succinctness gap already between  $\text{ML}^2$  and ML.

COROLLARY 4.14

The two-dimensional modal logic  $\text{ML}^2$  is non-elementarily more succinct than ML.

## 5 The formula size game for $L_\mu$

To define a formula size game similar to the one of ML for  $L_\mu$ , we will need some additional notation and concepts, since  $L_\mu$  is significantly more complex than ML.

Let  $(V, E)$  be a tree and let  $s, t \in V$ . We say that  $s$  is *above*  $t$  if there is an  $E$ -path from  $s$  to  $t$ . We say that  $s$  is *below*  $t$  if  $t$  is above  $s$ . A triple  $(V, E, B)$  is a *tree with back edges* if  $(V, E)$  is a tree and  $s$  is below  $t$  for every  $(s, t) \in B$ .

We define for each formula  $\varphi \in L_\mu$  its *syntax tree with back edges*,  $T_\varphi = (V_\varphi, E_\varphi, B_\varphi, \text{lab}_\varphi)$  as follows. The set  $V_\varphi$  consists of occurrences of subformulas of  $\varphi$  and the relation  $E_\varphi$  is the subformula relation between those occurrences. Additionally,  $\text{lab}_\varphi$  labels each vertex with its type (connective, modal operator, fixed point, literal or variable). Finally, the relation  $B_\varphi$  contains a back edge from each vertex labelled with a variable to the successor of the fixed point binding that variable.

A *partial function*  $f : M \rightarrow N$  is a function  $f' : M' \rightarrow N$  for some  $M' \subseteq M$ . For a partial function  $f : M \rightarrow N$ , we denote by

$$f[b_1/a_1, \dots, b_m/a_m, -/a_{m+1}, \dots, -/a_{m+n}] := \\ (f \setminus \{(a_i, b) \mid i \in \{1, \dots, m+n\}, b \in N\}) \cup \{(a_i, b_i) \mid i \in \{1, \dots, m\}\},$$

the partial function obtained from  $f$  by setting values for  $a_1, \dots, a_m \in M$  to  $b_1, \dots, b_m \in N$ , respectively, and the values for  $a_{m+1}, \dots, a_{m+n} \in M$  as undefined.

We add some features to pointed Kripke models for the game. A *clocked model* is a tuple  $(\mathcal{A}, w, c, a)$ , where  $(\mathcal{A}, w)$  is a pointed Kripke model,  $c : \text{Var} \rightarrow \kappa$  and  $a \in \{\text{new}, \text{old}\}$ . Here  $\kappa$  is a fixed cardinal larger than the size of the domain of  $\mathcal{A}$ . The partial function  $c$  associates to each fixed point a clock to show how many times the model can return to that fixed point. As clocked models traverse a graph in the game, we use the identifier *old* to keep track of where they have been

18 *Formula size games for modal logic and  $\mu$ -calculus*

previously. We suppress the age identifier  $a$  from the notation in cases where the distinction between new and old models does not matter.

For simplicity, we use the symbols  $w$  and  $c$  extensively and they should be read as ‘the distinguished point and clocks of the model currently discussed’ throughout the rest of the paper.

Let  $\mathfrak{A} = (\mathcal{A}, w, c, a)$  be a clocked model and  $\mathbb{A}$  a set of clocked models. We redefine the following notations from the ML case for clocked models:

- $\Box\mathfrak{A} = \Box(\mathcal{A}, w, c, a) := \{(\mathcal{A}, w', c, a) \mid wR^{\mathcal{A}}w'\}$ ,
- $\Box\mathbb{A} := \bigcup_{\mathfrak{A} \in \mathbb{A}} \Box\mathfrak{A}$ .
- Let  $f : \mathbb{A} \rightarrow \Box\mathbb{A}$  be a function such that  $f(\mathfrak{A}) \in \Box\mathfrak{A}$  for every  $\mathfrak{A} \in \mathbb{A}$ . Then  $\Diamond_f\mathbb{A} := f(\mathbb{A})$ .

As for the ML-game, the  $\Box$ -notation denotes the set of all successors of a single clocked model or a set of clocked models. The clocks and age identifier are inherited. The set  $\Diamond_f\mathbb{A}$  contains one successor for each clocked model in  $\mathbb{A}$ , given by the function  $f$ .

Now let  $\mathbb{A}$  be a set of clocked models,  $\mathbb{A}_0$  a set of pointed models and  $a \in \{\text{new}, \text{old}\}$ . We use the following new notations:

- $\text{PM}(\mathbb{A}) := \{(\mathcal{M}, w) \mid (\mathcal{M}, w, c, a) \in \mathbb{A}\}$ ,
- $\text{CM}(\mathbb{A}_0) := \{(\mathcal{M}, w, \emptyset, \text{new}) \mid (\mathcal{M}, w) \in \mathbb{A}_0\}$ ,
- $\mathbb{A}_a := \{(\mathcal{A}, w, c, b) \in \mathbb{A} \mid b = a\}$ ,
- $a(\mathbb{A}) := \{(\mathcal{A}, w, c, a) \mid (\mathcal{A}, w, c, b) \in \mathbb{A} \text{ for some } b \in \{\text{new}, \text{old}\}\}$ .

The set  $\text{PM}(\mathbb{A})$  contains the underlying pointed models of all clocked models in  $\mathbb{A}$  and the set  $\text{CM}(\mathbb{A}_0)$  is the set of clocked models with underlying pointed models from  $\mathbb{A}_0$  and empty clock functions. For an age identifier  $a$ , the set  $\mathbb{A}_a$  gives all clocked models in  $\mathbb{A}$  with that identifier and the set  $a(\mathbb{A})$  gives all the models in  $\mathbb{A}$  with the age identifier changed to  $a$ .

We define for  $L_\mu$  the standard approximant formulas that evaluate a fixed point only up to a bound. These approximants are formulas of *infinitary*  $L_\mu$ , where infinite conjunctions and disjunctions are allowed.

## DEFINITION 5.1

Let  $\alpha$  be an ordinal and  $\psi(X)$  a formula of infinitary  $L_\mu$ . Then the *approximant formulas*  $\mu^\alpha X.\psi(X)$  and  $\nu^\alpha X.\psi(X)$  are defined by recursion as follows:

- $\mu^0 X.\psi(X) = \perp$  and  $\nu^0 X.\psi(X) = \top$ ,
- $\mu^{\alpha+1} X.\psi(X) = \psi(\mu^\alpha X.\psi(X))$  and  $\nu^{\alpha+1} X.\psi(X) = \psi(\nu^\alpha X.\psi(X))$ ,
- $\mu^\lambda X.\psi(X) = \bigvee_{0 < \alpha < \lambda} \mu^\alpha X.\psi(X)$  and  $\nu^\lambda X.\psi(X) = \bigwedge_{0 < \alpha < \lambda} \nu^\alpha X.\psi(X)$  for a limit ordinal  $\lambda$ .

This definition differs from the usual one (see e.g. [20]) in that we leave out the  $\perp$  disjunct and  $\top$  conjunct in the limit ordinal cases, and more importantly, we do not necessarily approximate all fixed points so the resulting formula is not necessarily in infinitary ML but instead in infinitary  $L_\mu$ . During the game, we approximate several fixed points at once, starting from a specific point in the formula. We define our own approximate formulas to reflect this.

## DEFINITION 5.2

Let  $\varphi \in L_\mu$ . Let  $T_\varphi = (V_\varphi, E_\varphi, B_\varphi, \text{lab}_\varphi)$  be the syntax tree with back edges of  $\varphi$  and let  $s \in V_\varphi$ . Let  $r_1, \dots, r_n$  be the fixed point nodes above  $s$  in  $T_\varphi$  in order with  $r_1$  being the outermost and  $r_n$  the

innermost, and let  $\text{lab}_\varphi(r_i) = \eta_i X_i$  for each  $i \in \{1, \dots, n\}$ . Let  $c : \text{Var} \rightarrow \kappa$  be a partial function with  $\text{dom}(c) = \{X_1, \dots, X_n\}$ .

The  $(c, s)$ -approximant of  $\varphi$ ,  $\varphi_s^c$ , is defined recursively as follows:

- if  $\text{lab}(s) = l \in \text{Lit}$ , then  $\varphi_s^c = l$ ,
- if  $\text{lab}(s) = \nabla \in \{\vee, \wedge\}$  and  $s_1, s_2$  are the successors of  $s$ , then  $\varphi_s^c = \varphi_{s_1}^c \nabla \varphi_{s_2}^c$ ,
- if  $\text{lab}(s) = \Delta \in \{\diamond, \square\}$  and  $s_1$  is the successor of  $s$ , then  $\varphi_s^c = \Delta \varphi_{s_1}^c$ ,
- if  $\text{lab}(s) = \eta X$ , where  $\eta \in \{\mu, \nu\}$  and  $X \in \text{Var}$ , and  $s_1$  is the successor of  $s$ , then  $\varphi_s^c = \eta X. \varphi_{s_1}^c$ ,
- if  $\text{lab}(s) = X \in \text{Var} \setminus \text{dom}(c)$ , then  $\varphi_s^c = X$ .
- Let  $\text{lab}(s) = X_i$  and let  $u$  be the  $B_\varphi$ -successor of  $s$ .
- If  $c(X_i) = 0$ , then if  $\text{lab}(r_i) = \mu X_i$ ,  $\varphi_s^c = \perp$  and if  $\text{lab}(r_i) = \nu X_i$ ,  $\varphi_s^c = \top$ .
- If  $c(X_i) = \alpha + 1$  for some ordinal  $\alpha$ , let  $c_\alpha = c[\alpha/X_i, -/X_{i+1}, \dots, -/X_n]$ . Now  $\varphi_s^c = \varphi_u^{c_\alpha}$ .
- If  $c(X_i)$  is a limit ordinal, let  $c_\alpha = c[\alpha/X_i, -/X_{i+1}, \dots, -/X_n]$  for every  $\alpha < c(X_i)$ . Now

$$\varphi_s^c = \bigvee_{\alpha < c(X_i)} \varphi_u^{c_\alpha} \text{ if } \text{lab}(r_i) = \mu X_i \quad \text{and} \quad \varphi_s^c = \bigwedge_{\alpha < c(X_i)} \varphi_u^{c_\alpha} \text{ if } \text{lab}(r_i) = \nu X_i.$$

The formulas  $\varphi_s^c$  can contain infinite conjunctions and disjunctions but if all clocks are finite, then  $\varphi_s^c$  is an  $L_\mu$ -formula. For instance, if all models considered are finite, then finite clocks suffice. The following lemma formalizes the relationship of our approximant with the usual one.

#### LEMMA 5.3

Let  $\varphi \in L_\mu$  and let  $s$  be a vertex in the syntax tree of  $\varphi$  with  $\text{lab}(s) = \eta X$ , where  $\eta \in \{\mu, \nu\}$ . Let  $s_1$  be the successor of  $s$  and let  $c : \text{Var} \rightarrow \kappa$  be a partial function with  $X \notin \text{dom}(c)$ . Let  $c_\alpha = c[\alpha/X]$ . Now

$$\varphi_{s_1}^{c_\alpha} = \eta^{\alpha+1} X. \varphi_{s_1}^c(X).$$

PROOF. Since  $\eta^{\alpha+1} X. \varphi_{s_1}^{c_\alpha}(X) = \varphi_{s_1}^c(\eta^\alpha X. \varphi_{s_1}^c(X))$ , where the parentheses notation refers to substituting free occurrences of  $X$  with a formula, we may rewrite the claim in the form  $\varphi_{s_1}^{c_\alpha} = \varphi_{s_1}^c(\eta^\alpha X. \varphi_{s_1}^c(X))$ . We show this by transfinite induction on  $\alpha$ .

- If  $\alpha = 0$ , then it is easy to see that  $\varphi_{s_1}^{c_\alpha} = \varphi_{s_1}^c(\xi) = \varphi_{s_1}^c(\eta^0 X. \varphi_{s_1}^c(X))$ , where  $\xi = \perp$  if  $\eta = \mu$  and  $\xi = \top$  if  $\eta = \nu$ .
- Let  $\alpha = \beta + 1$ . By induction hypothesis,  $\varphi_{s_1}^{c_\beta} = \varphi_{s_1}^c(\eta^\beta X. \varphi_{s_1}^c(X)) = \eta^\alpha X. \varphi_{s_1}^c(X)$ . We show by induction on the definition of  $\varphi_t^{c_\alpha}$ , where  $t$  is below  $s$  in the syntax tree of  $\varphi$ , that  $\varphi_t^{c_\alpha} = \varphi_t^c(\eta^\alpha X. \varphi_{s_1}^c(X))$ . We first note that as  $s_1$  is the successor of the fixed point node  $s$ , the fixed point of  $X$  is the innermost one in  $\text{dom}(c_\alpha)$ .
  - If  $\text{lab}(t) = l \in \text{Lit}$ , then  $\varphi_t^{c_\alpha} = l = \varphi_t^c(\eta^\alpha X. \varphi_{s_1}^c(X))$ .
  - If  $\text{lab}(t) = Y \in \text{Var} \setminus \text{dom}(c_\alpha)$ , then  $\varphi_t^{c_\alpha} = Y = \varphi_t^c(\eta^\alpha X. \varphi_{s_1}^c(X))$ .
  - Let  $\text{lab}(t) = Y \in \text{dom}(c_\alpha) \setminus \{X\}$  and let  $u$  be the  $B$ -successor of  $t$ . Now for some  $c'$ ,  $\varphi_t^{c_\alpha} = \varphi_u^{c'}$ . Since the fixed point of  $X$  is inside that of  $Y$ ,  $\varphi_u^{c'}$  contains no free occurrences of  $X$ . Thus,  $\varphi_u^{c'} = \varphi_t^c(\eta^\alpha X. \varphi_{s_1}^c(X))$ .
  - If  $\text{lab}(t) = X$ , then  $\varphi_t^{c_\alpha} = \varphi_{s_1}^{c_\beta}$ . Because  $\eta X$  is the innermost approximated fixed point, no clocks need to be reset. Since  $\varphi_t^c = X$  and by the induction hypothesis on  $\alpha$ ,

$$\varphi_t^{c_\alpha} = \varphi_{s_1}^{c_\beta} = \eta^\alpha X. \varphi_{s_1}^c(X) = \varphi_t^c(\eta^\alpha X. \varphi_{s_1}^c(X)).$$

## 20 Formula size games for modal logic and $\mu$ -calculus

- If  $\text{lab}(t) = \nabla \in \{\vee, \wedge\}$  and  $t_1$  and  $t_2$  are the successors of  $t$ , then by induction hypothesis,  $\varphi_{t_1}^{c\alpha} = \varphi_{t_1}^c(\eta^\alpha X.\varphi_{s_1}^c(X))$  and  $\varphi_{t_2}^{c\alpha} = \varphi_{t_2}^c(\eta^\alpha X.\varphi_{s_1}^c(X))$ . Now

$$\begin{aligned}\varphi_t^{c\alpha} &= \varphi_{t_1}^{c\alpha} \nabla \varphi_{t_2}^{c\alpha} = \varphi_{t_1}^c(\eta^\alpha X.\varphi_{s_1}^c(X)) \nabla \varphi_{t_2}^c(\eta^\alpha X.\varphi_{s_1}^c(X)) \\ &= (\varphi_{t_1}^c \nabla \varphi_{t_2}^c)(\eta^\alpha X.\varphi_{s_1}^c(X)) = \varphi_t^c(\eta^\alpha X.\varphi_{s_1}^c(X)).\end{aligned}$$

- If  $\text{lab}(t) = \Delta \in \{\diamond, \square\}$  and  $t_1$  is the successor of  $t$ , then by induction hypothesis we have  $\varphi_{t_1}^{c\alpha} = \varphi_{t_1}^c(\eta^\alpha X.\varphi_{s_1}^c(X))$ . Thus,

$$\varphi_t^{c\alpha} = \Delta \varphi_{t_1}^{c\alpha} = \Delta \varphi_{t_1}^c(\eta^\alpha X.\varphi_{s_1}^c(X)) = (\Delta \varphi_{t_1}^c)(\eta^\alpha X.\varphi_{s_1}^c(X)) = \varphi_t^c(\eta^\alpha X.\varphi_{s_1}^c(X)).$$

- If  $\text{lab}(t) = \eta_1 Y$ , where  $\eta_1 \in \{\mu, \nu\}$  and  $Y \in \text{Var}$ , and  $t_1$  is the successor of  $t$ , then by induction hypothesis  $\varphi_{t_1}^{c\alpha} = \varphi_{t_1}^c(\eta^\alpha X.\varphi_{s_1}^c(X))$ . Thus,

$$\varphi_t^{c\alpha} = \eta_1 Y.\varphi_{t_1}^{c\alpha} = \eta_1 Y.\varphi_{t_1}^c(\eta^\alpha X.\varphi_{s_1}^c(X)) = (\eta_1 Y.\varphi_{t_1}^c)(\eta^\alpha X.\varphi_{s_1}^c(X)) = \varphi_t^c(\eta^\alpha X.\varphi_{s_1}^c(X)).$$

- Now let  $\alpha$  be a limit ordinal. By induction hypothesis,  $\varphi_{s_1}^{c\beta} = \varphi_{s_1}^c(\eta^\beta X.\varphi_{s_1}^c(X))$  for all  $\beta < \alpha$ . The induction on  $\varphi_t^{c\alpha}$  is handled in the same way as in the previous case with the exception of the  $X$ -case.

- If  $\text{lab}(t) = X$ , then  $\varphi_t^{c\alpha} = \nabla_{\beta < \alpha} \varphi_{s_1}^{c\beta}$ , where  $\nabla = \bigvee$  if  $\eta = \mu$  and  $\nabla = \bigwedge$  if  $\eta = \nu$ . By the induction hypothesis on  $\alpha$ ,  $\varphi_{s_1}^{c\beta} = \varphi_{s_1}^c(\eta^\beta X.\varphi_{s_1}^c(X))$  for all  $\beta < \alpha$  so

$$\begin{aligned}\varphi_t^{c\alpha} &= \nabla_{\beta < \alpha} \varphi_{s_1}^c(\eta^\beta X.\varphi_{s_1}^c(X)) = \nabla_{\beta < \alpha} \eta^{\beta+1} X.\varphi_{s_1}^c(X) = \nabla_{0 < \beta < \alpha} \eta^\beta X.\varphi_{s_1}^c(X) \\ &= \eta^\alpha X.\varphi_{s_1}^c(X) = \varphi_t^c(\eta^\alpha X.\varphi_{s_1}^c(X)).\end{aligned}$$

□

### The definition of the $L_\mu$ -game

Let  $\Phi$  be a fixed finite set of propositional symbols. The formula size game for  $L_\mu(\Phi)$ ,  $\mu\text{-FS}_k^\Phi(\mathbb{A}_0, \mathbb{B}_0)$ , has two players, S (Samson) and D (Delilah). The game has, as parameters, two sets of pointed  $\Phi$ -models,  $\mathbb{A}_0$  and  $\mathbb{B}_0$ , and a natural number  $k$ . S wants to show that the sets  $\mathbb{A}_0$  and  $\mathbb{B}_0$  can be separated with a  $L_\mu(\Phi)$ -formula of size at most  $k$ . D on the other hand wants to show this is not possible. During the game, S constructs step by step the syntax tree of a formula that, he claims, separates the sets. The number  $k$  is a resource that is spent when S adds vertices to the syntax tree. If the resource  $k$  ever runs out, S loses the game. S has to simultaneously show how the models in  $\mathbb{A}_0$  make the formula true and how the models in  $\mathbb{B}_0$  make it false. Each model traverses the incomplete syntax tree in a fashion similar to semantic games. The role of D is to keep S honest by deciding which branch of the tree she wants to see next.

The modal  $\mu$ -calculus has the special feature of fixed point formulas. In terms of models traversing the syntax tree of a formula, the truth of a least fixed point  $\mu X$  comes down to the model having to eventually stop returning to that fixed point. Thus, when entering such a fixed point, S must set a clock for each model that shows how many more times he will return the model to that fixed point. On the other hand, to show a  $\mu X$ -formula is false in a model, the model would have to keep returning to the fixed point forever. Here it is the responsibility of D to declare how many returns are enough for her to be satisfied that the formula is indeed false. We now present the quite complex formalization of the game.

Let  $\mathbb{A}_0$  and  $\mathbb{B}_0$  be sets of pointed models and let  $k_0 \in \mathbb{N}$ . Let  $V^*$  be a predefined infinite set of vertices. The formula size game  $\mu\text{-FS}_{k_0}^\Phi(\mathbb{A}_0, \mathbb{B}_0)$  for the modal  $\mu$ -calculus has two players, S and D. The positions of the game are of the form  $P = (V, E, B, \text{lab}, \text{res}, \text{left}, \text{right}, v)$ . Here  $V \subseteq V^*$  and  $(V, E, B)$  is a tree with back edges. The partial function

$$\text{lab} : V \rightarrow \{\wedge, \vee, \diamond, \square\} \cup \text{Var} \cup \{\mu X, \nu X \mid X \in \text{Var}\} \cup \text{Lit}(\Phi)$$

assigns a label to some vertices of the tree. The function  $\text{res} : V \rightarrow \mathbb{N}$  assigns to each vertex the remaining resource, i.e. an upper bound for the size of the subformula starting from the vertex. The function  $\text{left} : V \rightarrow \mathcal{P}(\mathbb{A}_0^*)$  assigns to each  $v \in V$  its left set of clocked models  $\text{left}(v)$ . Here  $\mathbb{A}_0^*$  contains all the clocked models obtainable from models of  $\mathbb{A}_0$  by altering the distinguished point, clocks and age identifier. Similarly,  $\text{right} : V \rightarrow \mathcal{P}(\mathbb{B}_0^*)$  assigns the right set  $\text{right}(v)$ . The clock function of each model is of the form  $c : \text{Var} \rightarrow \kappa$ , where  $\kappa$  is a fixed cardinal larger than the size of the domain of any model in  $\mathbb{A}_0 \cup \mathbb{B}_0$ . Finally, the vertex  $v \in V$  is the *current vertex* of the position. We will always assume that the position  $P$  has components with these names and  $P'$  always consists of the same components with primes.

The starting position of the game is

$$(\{v_0\}, \emptyset, \emptyset, \emptyset, \{(v_0, k_0)\}, \{(v_0, \text{CM}(\mathbb{A}_0))\}, \{(v_0, \text{CM}(\mathbb{B}_0))\}, v_0).$$

The first move is always D choosing finite subsets  $\mathbb{A} \subseteq \text{CM}(\mathbb{A}_0)$  and  $\mathbb{B} \subseteq \text{CM}(\mathbb{B}_0)$ . The following position is

$$(\{v_0\}, \emptyset, \emptyset, \emptyset, \{(v_0, k_0)\}, \{(v_0, \mathbb{A})\}, \{(v_0, \mathbb{B})\}, v_0).$$

Throughout the whole game, D wins if at any position  $P$ ,  $\text{res}(v) = 0$ . D also wins if S is unable to make the choices required by a move. Assume the game is in position  $P$  and let  $\text{left}(v) = \mathbb{A}$ ,  $\text{right}(v) = \mathbb{B}$  and  $\text{res}(v) = k > 0$ . We define two cases by whether  $v$  already has a label or not. In each case, we denote the following position by  $P'$ .

$v \notin \text{dom}(\text{lab})$ : S has a choice of eight different moves. Note that in this case  $\mathbb{A} = \mathbb{A}_{\text{new}}$  and  $\mathbb{B} = \mathbb{B}_{\text{new}}$ .

- **$\nu$ -move**: S chooses sets  $\mathbb{A}_1, \mathbb{A}_2 \subseteq \mathbb{A}$  s.t.  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$ , and numbers  $0 < k_1, k_2 \leq k$  s.t.  $k_1 + k_2 + 1 = k$ . Then D chooses a number  $i \in \{1, 2\}$ . Let  $V' = V \cup \{v_1, v_2\}$ ,  $E' = E \cup \{(v, v_1), (v, v_2)\}$ ,  $B' = B$ ,  $\text{lab}' = \text{lab}[v/v]$ ,  $\text{left}' = \text{left}[\mathbb{A}_1/v_1, \mathbb{A}_2/v_2, \text{old}(\mathbb{A})/v]$ ,  $\text{right}' = \text{right}[\mathbb{B}/v_1, \mathbb{B}/v_2, \text{old}(\mathbb{B})/v]$ ,  $\text{res}' = \text{res}[k_1/v_1, k_2/v_2]$  and  $v' = v_i$ , where  $v_1$  and  $v_2$  are new vertices.
- **$\wedge$ -move**: Same as the  $\nu$ -move with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.
- **$\diamond$ -move**: S chooses a function  $f : \mathbb{A} \rightarrow \square \mathbb{A}$  such that  $f(\mathfrak{A}) \in \square \mathfrak{A}$  for each  $\mathfrak{A} \in \mathbb{A}$ . Then D chooses finite subsets  $\mathbb{A}' \subseteq \diamond_f \mathbb{A}$  and  $\mathbb{B}' \subseteq \square \mathbb{B}$ . Let  $V' = V \cup \{v'\}$ ,  $E' = E \cup \{(v, v')\}$ ,  $B' = B$ ,  $\text{lab}' = \text{lab}[\diamond/v]$ ,  $\text{left}' = \text{left}[\mathbb{A}'/v', \text{old}(\mathbb{A})/v]$ ,  $\text{right}' = \text{right}[\mathbb{B}'/v', \text{old}(\mathbb{B})/v]$  and  $\text{res}' = \text{res}[k - 1/v']$ , where  $v'$  is a new vertex.
- **$\square$ -move**: Same as the  $\diamond$ -move with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.
- **$\mu X$ -move**: S chooses a variable  $X \in \text{Var}$  and for every  $\mathfrak{A} = (\mathcal{A}_{\mathfrak{A}}, w_{\mathfrak{A}}, c_{\mathfrak{A}}) \in \mathbb{A}$  an ordinal  $\alpha_{\mathfrak{A}}$ . Then D chooses for every  $\mathfrak{B} = (\mathcal{B}_{\mathfrak{B}}, w_{\mathfrak{B}}, c_{\mathfrak{B}}) \in \mathbb{B}$  an ordinal  $\alpha_{\mathfrak{B}}$ . Let  $c'_{\mathfrak{A}} = c_{\mathfrak{A}}[\alpha_{\mathfrak{A}}/X]$  and let  $\mathbb{A}' = \{(\mathcal{A}_{\mathfrak{A}}, w_{\mathfrak{A}}, c'_{\mathfrak{A}}) \mid \mathfrak{A} \in \mathbb{A}\}$ . Let  $c'_{\mathfrak{B}} = c_{\mathfrak{B}}[\alpha_{\mathfrak{B}}/X]$  and let  $\mathbb{B}' = \{(\mathcal{B}_{\mathfrak{B}}, w_{\mathfrak{B}}, c'_{\mathfrak{B}}) \mid \mathfrak{B} \in \mathbb{B}\}$ . Let  $V' = V \cup \{v'\}$ ,  $E' = E \cup \{(v, v')\}$ ,  $B' = B$ ,  $\text{lab}' = \text{lab}[\mu X/v]$ ,  $\text{left}' = \text{left}[\mathbb{A}'/v', \text{old}(\mathbb{A})/v]$ ,  $\text{right}' = \text{right}[\mathbb{B}'/v', \text{old}(\mathbb{B})/v]$  and  $\text{res}' = \text{res}[k - 1/v']$ .
- **$\nu X$ -move**: Same as the  $\mu$ -move with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.

## 22 Formula size games for modal logic and $\mu$ -calculus

- **$X$ -move:** S chooses  $X \in \text{Var}$ . Let  $u \in V$  be the closest vertex above  $v$  with  $\text{lab}(u) \in \{\mu X, \nu X\}$ . If no such vertex exists, D wins the game. Otherwise, if  $\mathbb{A}_{\text{new}} = \mathbb{B}_{\text{new}} = \emptyset$ , S wins the game.
- Let  $v'$  be the successor vertex of  $u$ . Let  $V' = V, E' = E, B' = B \cup \{(v, v')\}$ ,  $\text{lab}' = \text{lab}[X/v]$  and  $\text{res}' = \text{res}$ .
- Assume that  $\text{lab}(u) = \mu X$ . If  $c(X) = 0$  for some  $(\mathcal{A}, w, c) \in \mathbb{A}$ , D wins the game. Otherwise, for every  $\mathfrak{A} = (\mathcal{A}_{\mathfrak{A}}, w_{\mathfrak{A}}, c_{\mathfrak{A}}, \text{new}) \in \mathbb{A}$ , S chooses  $\alpha_{\mathfrak{A}} < c_{\mathfrak{A}}(X)$ .
- Let  $\mathbb{B}_+ = \{(\mathcal{B}, w, c, \text{new}) \in \mathbb{B} \mid c(X) \neq 0\}$ . For every  $\mathfrak{B} = (\mathcal{B}_{\mathfrak{B}}, w_{\mathfrak{B}}, c_{\mathfrak{B}}, \text{new}) \in \mathbb{B}_+$ , D chooses  $\alpha_{\mathfrak{B}} < c_{\mathfrak{B}}(X)$ .
- Let  $Y_1, \dots, Y_n$  be the variables for which there is a node  $t_i$  on the path from  $u$  to  $v$  with  $\text{lab}(t_i) \in \{\mu Y_i, \nu Y_i\}$ . Let  $c'_{\mathfrak{A}} = c_{\mathfrak{A}}[\alpha_{\mathfrak{A}}/X, -/Y_1, \dots, -/Y_n]$  and let  $\mathbb{A}' = \{(\mathcal{A}_{\mathfrak{A}}, w_{\mathfrak{A}}, c'_{\mathfrak{A}}, \text{new}) \mid \mathfrak{A} \in \mathbb{A}\}$ . Similarly, let  $c'_{\mathfrak{B}} = c_{\mathfrak{B}}[\alpha_{\mathfrak{B}}/X, -/Y_1, \dots, -/Y_n]$  and  $\mathbb{B}' = \{(\mathcal{B}_{\mathfrak{B}}, w_{\mathfrak{B}}, c'_{\mathfrak{B}}, \text{new}) \mid \mathfrak{B} \in \mathbb{B}_+\}$ . Let  $\text{left}' = \text{left}[\mathbb{A}' \cup \text{left}(v')/v', \text{old}(\mathbb{A})/v]$  and  $\text{right}' = \text{right}[\mathbb{B}' \cup \text{left}(v')/v', \text{old}(\mathbb{B})/v]$ .
- The case  $\text{lab}(u) = \nu X$  is the same with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.
- **Lit-move:** S chooses a  $\Phi$ -literal  $l$ . Let  $\text{lab}' = \text{lab}[l/v]$  and let  $\Delta' = \Delta$  for every other component  $\Delta$ . In the following position  $P'$ , if  $l$  separates  $\mathbb{A}$  and  $\mathbb{B}$ , then S wins the game. Otherwise, D wins.

$v \in \text{dom}(\text{lab})$ : In this case, S must perform the move dictated by  $\text{lab}(v)$  without creating any new vertices. These moves are essentially performed only on new models. We again denote the following position by  $P'$  and in each case we have  $\nabla' = \nabla$  for  $\nabla \in \{V, E, B, \text{lab}, \text{res}\}$ .

- If  $\text{lab}(v) = \vee$ , then let  $v_1$  and  $v_2$  be the successors of  $v$ . S chooses sets  $\mathbb{A}_1, \mathbb{A}_2 \subseteq \mathbb{A}_{\text{new}}$  s.t.  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}_{\text{new}}$ . Then D chooses a number  $i \in \{1, 2\}$ . Let  $\text{left}' = \text{left}[\mathbb{A}_1 \cup \text{left}(v_1)/v_1, \mathbb{A}_2 \cup \text{left}(v_2)/v_2, \text{old}(\mathbb{A})/v]$ ,  $\text{right}' = \text{right}[\mathbb{B} \cup \text{right}(v_1)/v_1, \mathbb{B} \cup \text{right}(v_2)/v_2, \text{old}(\mathbb{B})/v]$  and  $v' = v_i$ .
- The case  $\text{lab}(v) = \wedge$  is the same as  $\vee$  with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.
- If  $\text{lab}(v) = \diamond$ , then let  $v'$  be the successor of  $v$ . S chooses a function  $f : \mathbb{A}_{\text{new}} \rightarrow \square \mathbb{A}_{\text{new}}$  such that  $f(\mathfrak{A}) \in \square \mathfrak{A}$  for each  $\mathfrak{A} \in \mathbb{A}_{\text{new}}$ . Then D chooses finite  $\mathbb{A}' \subseteq \diamond_f \mathbb{A}_{\text{new}}$  and  $\mathbb{B}' \subseteq \square \mathbb{B}_{\text{new}}$ . Let  $\text{left}' = \text{left}[\mathbb{A}' \cup \text{left}(v')/v_1, \text{old}(\mathbb{A})/v]$  and  $\text{right}' = \text{right}[\mathbb{B}' \cup \text{right}(v')/v_1, \text{old}(\mathbb{B})/v]$ .
- The case  $\text{lab}(v) = \square$  is the same as  $\diamond$  with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.
- If  $\text{lab}(v) = \mu X$  for some  $X \in \text{Var}$ , then let  $v'$  be the successor of  $v$ . S chooses for every  $\mathfrak{A} = (\mathcal{A}_{\mathfrak{A}}, w_{\mathfrak{A}}, c_{\mathfrak{A}}) \in \mathbb{A}_{\text{new}}$  an ordinal  $\alpha_{\mathfrak{A}}$ . Then D chooses for every  $\mathfrak{B} = (\mathcal{B}_{\mathfrak{B}}, w_{\mathfrak{B}}, c_{\mathfrak{B}}) \in \mathbb{B}_{\text{new}}$  an ordinal  $\alpha_{\mathfrak{B}}$ . Let  $c'_{\mathfrak{A}} = c_{\mathfrak{A}}[\alpha_{\mathfrak{A}}/X]$  and let  $\mathbb{A}' = \{(\mathcal{A}_{\mathfrak{A}}, w_{\mathfrak{A}}, c'_{\mathfrak{A}}) \mid \mathfrak{A} \in \mathbb{A}_{\text{new}}\}$ . Let  $c'_{\mathfrak{B}} = c_{\mathfrak{B}}[\alpha_{\mathfrak{B}}/X]$  and let  $\mathbb{B}' = \{(\mathcal{B}_{\mathfrak{B}}, w_{\mathfrak{B}}, c'_{\mathfrak{B}}) \mid \mathfrak{B} \in \mathbb{B}_{\text{new}}\}$ . Let  $\text{left}' = \text{left}[\mathbb{A}' \cup \text{left}(v')/v', \text{old}(\mathbb{A})/v]$  and  $\text{right}' = \text{right}[\mathbb{B}' \cup \text{right}(v')/v', \text{old}(\mathbb{B})/v]$ .
- If  $\text{lab}(v) = X \in \text{Var}$ , then let  $v'$  be the  $B$ -successor of  $v$ . The rest is very similar to the unlabelled  $X$ -move; the only differences are that the move is again essentially only performed on new models and the condition for an immediate win for S is  $\mathbb{A}_{\text{new}} = \mathbb{B}_{\text{new}} = \emptyset$ .

Note that just like in the ML-game, the  $\diamond$ -move cannot be performed if  $\square \mathfrak{A} = \emptyset$  for some  $\mathfrak{A} \in \mathbb{A}$ , and dually for the  $\square$ -move.

### EXAMPLE 5.4

Figure 3 depicts the syntax tree with back edges of the  $L_{\mu}$ -formula  $\varphi := \nu X. \mu Y. (\diamond X \wedge p) \vee \diamond Y$  and two models,  $\mathcal{A}$  and  $\mathcal{B}$ , where the black points satisfy  $p$  and the white points do not. Consider a game  $\mu\text{-FS}_9^{\Phi}(\mathbb{A}_0, \mathbb{B}_0)$  with  $(\mathcal{A}, a_1) \in \mathbb{A}_0$  and  $(\mathcal{B}, b_1) \in \mathbb{B}_0$ . We follow along as S plays a strategy based on the formula  $\varphi$ .



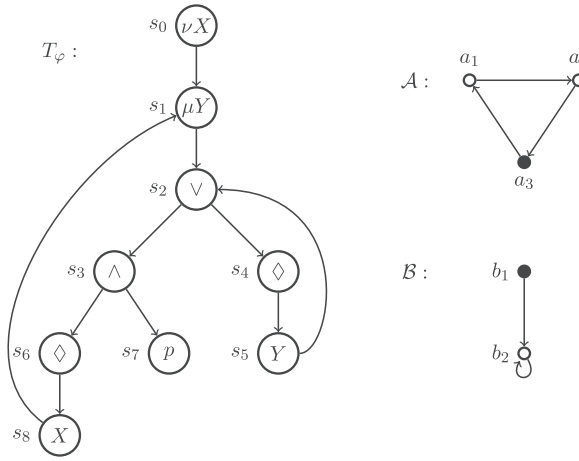


Figure 3 The formula and two models for Example 5.4.

We begin from the unlabelled vertex  $s_0$ . First, S makes a  $\nu X$ -move, labelling the vertex with  $\nu X$  and creating a new vertex,  $s_1$ . S sets the clock of  $(\mathcal{B}, b_1)$  to  $c_{\mathcal{B}}(X) = 1$ . Then D chooses  $c_{\mathcal{A}}(X)$  but since it is quite irrelevant for this example, we will not keep track of it.

Next, S makes a  $\mu Y$ -move at  $s_1$ . Now S must set the clock of  $\mathcal{A}$  and he chooses  $c_{\mathcal{A}}(Y) = 2$ . D chooses  $c_{\mathcal{B}}(Y)$ , which is again irrelevant for this example.

The clocks of S are always reachability clocks. The formula  $\varphi$  as a whole says that there is always a path to a point with  $p$ . Since S is claiming  $(\mathcal{B}, b_1)$  does not satisfy  $\varphi$ , the clock  $c_{\mathcal{B}}(X) = 1$  means that S claims he will inevitably reach a point, where  $p$  cannot be reached, in just one step. As for  $c_{\mathcal{A}}(Y) = 2$ , S claims he can reach  $p$  in two steps in the model  $(\mathcal{A}, a_1)$ . From Figure 3, we can see that both of these claims are true.

The next move of S is a  $\vee$ -move at  $s_2$ . S decides to put  $(\mathcal{A}, a_1, c_{\mathcal{A}})$  to the side of  $s_4$ , as he has not yet reached  $p$ . The clocked model  $(\mathcal{B}, b_1, c_{\mathcal{B}})$  is copied to both  $s_3$  and  $s_4$ . D decides that the game will continue from the node  $s_4$ . After the  $\diamond$ -move at  $s_4$ , the models are  $(\mathcal{A}, a_2, c_{\mathcal{A}})$  and  $(\mathcal{B}, b_2, c_{\mathcal{B}})$ .

Now S makes his first  $Y$ -move at  $s_5$ . He lowers the clock of  $\mathcal{A}$  to  $c_{\mathcal{A}}(Y) = 1$ . D also lowers her clock for  $\mathcal{B}$ . The game returns to  $s_2$ . As S still has not reached  $p$  in  $\mathcal{A}$ , he again puts  $\mathcal{A}$  on the side of  $s_4$  in the following  $\vee$ -move. D again decides to continue from  $s_4$  and the  $\diamond$ - and  $Y$ -moves are repeated. We end up at  $s_2$  with the models  $(\mathcal{A}, a_3, c_{\mathcal{A}})$  and  $(\mathcal{B}, b_2, c_{\mathcal{B}})$  with the clock of  $\mathcal{A}$  being  $c_{\mathcal{A}}(Y) = 0$ .

Now S has reached  $p$  in  $\mathcal{A}$  so he is ready to put the model on the  $s_3$ -side of the  $\vee$ -move. D decides to choose the  $s_3$ -side this time. Note that the set  $\text{right}(s_3)$  now has three different versions of the model  $\mathcal{B}$  from the previous  $\vee$ -moves. We will only consider the models  $\mathfrak{B}_1 = (\mathcal{B}, b_1, c_{\mathcal{B}})$  and  $\mathfrak{B}_2 = (\mathcal{B}, b_2, c_{\mathcal{B}})$  as the third one only differs from  $\mathfrak{B}_2$  with respect to a clock of D.

At  $s_3$ , S makes a  $\wedge$ -move. S decides to put  $\mathfrak{B}_1$  to the  $s_6$ -side and  $\mathfrak{B}_2$  to the  $s_7$ -side. The model  $(\mathcal{A}, a_3, c_{\mathcal{A}})$  is copied to both  $s_6$  and  $s_7$ . Now D sees that  $(\mathcal{A}, a_3) \models p$  and  $(\mathcal{B}, b_2) \not\models p$  so she chooses to continue from  $s_6$  so as to not lose the game to a *Lit*-move.

The  $\diamond$ -move at  $s_6$  again moves the models one step forward, resulting in  $(\mathcal{A}, a_1, c_{\mathcal{A}})$  and  $(\mathcal{B}, b_2, c_{\mathcal{B}})$ . Finally, at  $s_8$ , S makes an  $X$ -move. S lowers the clock of  $\mathcal{B}$  to  $c_{\mathcal{B}}(X) = 0$ . For  $\mathcal{A}$ , this lowers the clock of D, but more importantly, removes the clock  $c_{\mathcal{A}}(Y) = 0$  entirely. This means

## 24 Formula size games for modal logic and $\mu$ -calculus

that when S makes the next  $\mu Y$ -move dictated by the label of  $s_1$ , he can again set the clock at 2 and repeat the above process as many times as he wants. Meanwhile, each return to  $s_1$  lowers the clock of D so eventually the model  $\mathcal{A}$  will be removed from the game. The model  $\mathcal{B}$  on the other hand can be dropped off at  $s_7$ . It would seem that S is on the right track to win this play.

An important feature of our game is that, even though it contains infinite branching, every single play of the game is still finite.

LEMMA 5.5

Every play of the game  $\mu\text{-FS}_k^\Phi(\mathbb{A}, \mathbb{B})$  is finite.

PROOF. A play could be infinite only if at least one variable is reached infinitely many times. Of these variables, let  $X$  be the one with the outmost fixed point. Every time  $X$  is reached, if  $\text{left}(v)_{\text{new}} = \text{right}(v)_{\text{new}} = \emptyset$ , S wins and otherwise, the clock of at least one model is lowered. There are only finitely many models at any given position, since D always chooses finite subsets of models after modal moves. Since clocks are inherited by successor models in modal moves and ordinals are well founded, eventually either a clock of S will reach 0 and D will win or  $X$  will be reached with empty sets of models and S wins.  $\square$

For the essential theorem about how the game works, we assume that the strategy of S is uniform. This essentially means that S has a formula in mind and he follows the structure of that formula when constructing the syntax tree during the game.

DEFINITION 5.6

Let  $\varphi \in L_\mu$  and let  $T_\varphi = (V_\varphi, E_\varphi, B_\varphi, \text{lab}_\varphi)$  be the syntax tree with back edges of  $\varphi$ . Let  $P = (V, E, B, \text{lab}, \text{res}, \text{left}, \text{right}, \nu)$  be a position in a game  $\mu\text{-FS}_k^\Phi(\mathbb{A}, \mathbb{B})$ .

A function  $g : V \rightarrow V_\varphi$  is a *position embedding* if it satisfies the following conditions:

1.  $g(v_0)$  is the root of  $T_\varphi$ , where  $v_0$  is the vertex of the starting position,
2.  $g$  is an embedding of  $(V, E)$  to  $(V_\varphi, E_\varphi)$ ,
3.  $g \upharpoonright \text{dom}(\text{lab})$  is an embedding of  $(V, B, \text{lab})$  to  $(V_\varphi, B_\varphi, \text{lab}_\varphi)$ ,
4. for each  $u \in V$ ,  $\text{sz}(\varphi_{g(u)}) \leq \text{res}(u)$ .

Let  $\delta$  be a strategy of S. We say that  $\delta$  *follows  $\varphi$  from position  $P$  (via the function  $g$ )* if there is a position embedding  $g : V \rightarrow V_\varphi$  such that for each position  $P'$  reachable from  $P$  via the strategy  $\delta$ , the function  $g$  can be extended to a position embedding  $g' : V' \rightarrow V_\varphi$ .

Finally,  $\delta$  is *uniform* if  $\delta$  follows a formula  $\varphi \in L_\mu$  from the starting position.

We are now ready to prove that the game indeed works as we intended. In the following, if there is a vertex with label  $\mu X$  we shall call  $X$  a  $\mu$ -variable, and if there is one labelled  $\nu X$ , we call  $X$  a  $\nu$ -variable. Note that we assume all fixed points have separate variables.

THEOREM 5.7

Let  $\mathbb{A}_0$  and  $\mathbb{B}_0$  be sets of pointed models and let  $k \in \mathbb{N}$ . Then the following conditions are equivalent:

1. S has a uniform winning strategy in the game  $\mu\text{-FS}_k^\Phi(\mathbb{A}_0, \mathbb{B}_0)$ .
2. There is a sentence  $\varphi \in L_\mu(\Phi)$  s.t.  $\varphi$  separates  $\mathbb{A}_0$  and  $\mathbb{B}_0$  and  $\text{sz}(\varphi) \leq k$ .

PROOF. (2)  $\Rightarrow$  (1). Let  $\varphi \in L_\mu(\Phi)$  be a sentence such that  $\varphi$  separates  $\mathbb{A}_0$  and  $\mathbb{B}_0$  and  $\text{sz}(\varphi) \leq k$ . Let  $T_\varphi = (V_\varphi, E_\varphi, B_\varphi, \text{lab}_\varphi)$  be the syntax tree with back edges of  $\varphi$ . The strategy of S is to follow the structure of  $T_\varphi$  when forming  $(V, E, B, \text{lab})$ , to use the resource  $k$  accordingly and to choose maximal appropriate sets of models when necessary.

If  $P = (V, E, B, \text{lab}, \text{res}, \text{left}, \text{right}, v)$  is a position, let  $\mathbb{A} = \text{left}(v)$ ,  $\mathbb{B} = \text{right}(v)$  and  $k = \text{res}(v)$ . We define the strategy more precisely and simultaneously prove by induction that the strategy is uniform and the following condition holds for every position  $P$  of the game:

$$\begin{aligned} (\mathcal{A}, w) \models \varphi_{g(v)}^c \text{ for every } (\mathcal{A}, w, c) \in \mathbb{A}_{\text{new}} \text{ and} \\ (\mathcal{B}, w) \not\models \varphi_{g(v)}^c \text{ for every } (\mathcal{B}, w, c) \in \mathbb{B}_{\text{new}}, \end{aligned} \quad *$$

where  $g$  is a position embedding showing the uniformity of the strategy.

In the starting position, we set  $g(v_0)$  as the root of  $T_\varphi$ . We note that  $\text{sz}(\varphi_{g(v_0)}) = \text{sz}(\varphi) \leq k$  by assumption. Since there are no clocks yet,  $\varphi_{g(v_0)}^c = \varphi$  for every clocked model and since the sentence  $\varphi$  separates the sets  $\mathbb{A}_0$  and  $\mathbb{B}_0$ , (\*) holds no matter which subsets  $\mathbb{A} \subseteq \text{CM}(\mathbb{A}_0)$  and  $\mathbb{B} \subseteq \text{CM}(\mathbb{B}_0)$  D chooses.

We now divide the proof into cases based on whether  $v$  already has a label or not. We choose the move for  $S$  according to the label of  $g(v)$ . We only treat one of each pair of dual cases.

$v \notin \text{dom}(\text{lab})$  :

- $\text{lab}_\varphi(g(v)) = l \in \text{Lit}(\Phi)$ : Then by induction hypothesis  $l$  separates the sets  $\mathbb{A}$  and  $\mathbb{B}$  so S wins by making the corresponding *Lit*-move.
- $\text{lab}_\varphi(g(v)) = \vee$ : By induction hypothesis, (\*) holds for this position so for every  $(\mathcal{A}, w, c) \in \mathbb{A}$ ,  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ . Now  $\varphi_{g(v)}^c = \varphi_{s_1}^c \vee \varphi_{s_2}^c$ , where  $s_1, s_2 \in V_\varphi$  are the successors of  $g(v)$ , so  $(\mathcal{A}, w) \models \varphi_{s_1}^c \vee \varphi_{s_2}^c$ . Let  $\mathbb{A}_1 = \{(\mathcal{A}, w, c) \in \mathbb{A} \mid (\mathcal{A}, w) \models \varphi_{s_1}^c\}$  and  $\mathbb{A}_2 = \{(\mathcal{A}, w, c) \in \mathbb{A} \mid (\mathcal{A}, w) \models \varphi_{s_2}^c\}$ . On the other side, for every  $(\mathcal{B}, w, c) \in \mathbb{B}$ ,  $(\mathcal{B}, w) \not\models \varphi_{s_1}^c \vee \varphi_{s_2}^c$  so  $(\mathcal{B}, w) \not\models \varphi_{s_1}^c$  and  $(\mathcal{B}, w) \not\models \varphi_{s_2}^c$ . We set  $g' = g[v_1/s_1, v_2/s_2]$ , where  $v_1, v_2$  are the new vertices in  $V$ . Now (\*) holds in both of the possible following positions. Let  $k_1 = \text{sz}(\varphi_{s_1})$  and  $k_2 = k - k_1 - 1$ . Since  $\text{sz}(\varphi_{g(v)}) \leq k$ ,  $\text{sz}(\varphi_{s_2}) = \text{sz}(\varphi_{g(v)}) - \text{sz}(\varphi_{s_1}) - 1 \leq k - k_1 - 1 = k_2$ .
- $\text{lab}_\varphi(g(v)) = \diamond$ : By induction hypothesis, for every  $(\mathcal{A}, w, c) \in \mathbb{A}$ ,  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ . Since  $\varphi_{g(v)}^c = \diamond \varphi_{s_1}^c$ , where  $s_1$  is the successor of  $g(v)$ ,  $(\mathcal{A}, w) \models \diamond \varphi_{s_1}^c$ . Thus, there is  $(\mathcal{A}, w', c) \in \square \mathbb{A}$  s.t.  $(\mathcal{A}, w') \models \varphi_{s_1}^c$ . Let  $f : \mathbb{A} \rightarrow \square \mathbb{A}$  be a function mapping every  $(\mathcal{A}, w, c)$  to such a  $(\mathcal{A}, w', c)$ . Now  $(\mathcal{A}, w') \models \varphi_{s_1}^c$  for every  $(\mathcal{A}, w', c) \in \diamond_f \mathbb{A}$ . On the other side, for every  $(\mathcal{B}, w, c) \in \mathbb{B}$ , since  $(\mathcal{B}, w) \not\models \diamond \varphi_{s_1}^c$ , for every  $(\mathcal{B}, w', c) \in \square(\mathcal{B}, w, c)$  we get  $(\mathcal{B}, w') \not\models \varphi_{s_1}^c$ . Thus,  $(\mathcal{B}, w') \not\models \varphi_{s_1}^c$  for every  $(\mathcal{B}, w', c) \in \square \mathbb{B}$  so (\*) holds in the next position no matter which subsets  $\mathbb{A}' \subseteq \diamond_f \mathbb{A}$  and  $\mathbb{B}' \subseteq \square \mathbb{B}$  D chooses. For uniformity, we set  $g' = g[v'/s_1]$ , where  $v'$  is the new vertices. Now  $\text{sz}(\varphi_{s_1}) = \text{sz}(\varphi_{g(v)}) - 1 \leq k - 1$ .
- Let  $\text{lab}_\varphi(g(v)) = \mu X$ : By induction hypothesis, for every  $\mathfrak{A} = (\mathcal{A}, w, c) \in \mathbb{A}$ ,  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ . Since  $\varphi_{g(v)}^c = \mu X. \varphi_{s_1}^c(X)$ , where  $s_1$  is the successor of  $g(v)$ ,  $(\mathcal{A}, w) \models \mu X. \varphi_{s_1}^c(X)$ . Thus, there is an ordinal  $\alpha$  such that  $(\mathcal{A}, w) \models \mu^{\alpha+1} X. \varphi_{s_1}^c(X)$ . S chooses  $\alpha_{\mathfrak{A}} = \alpha$  as the new clock. By Lemma 5.3,  $(\mathcal{A}, w) \models \varphi_{s_1}^{c\alpha}$ . On the other side, by induction hypothesis, for every  $\mathfrak{B} = (\mathcal{B}, w, c) \in \mathbb{B}$ ,  $(\mathcal{B}, w) \not\models \mu X. \varphi_{s_1}^c(X)$  so for every ordinal  $\alpha$ ,  $(\mathcal{B}, w) \not\models \mu^{\alpha+1} X. \varphi_{s_1}^c(X)$ . Thus, no matter which ordinal  $\beta$  D chooses, we get  $(\mathcal{B}, w) \not\models \mu^{\beta+1} X. \varphi_{s_1}^c(X)$  and by Lemma 5.3,  $(\mathcal{B}, w) \not\models \varphi_{s_1}^{c\beta}$ . Therefore, (\*) holds in the following position. Uniformity is proved in the same way as in the  $\diamond$ -case.
- $\text{lab}_\varphi(g(v)) = X$ , where  $X \in \text{Var}$ : Let  $u$  be the  $B$ -successor of  $g(v)$ .
- Assume that  $X$  is a  $\mu$ -variable. Now by induction hypothesis, for every  $\mathfrak{A} = (\mathcal{A}, w, c) \in \mathbb{A}$ ,  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ . There are three cases according to  $c(X)$ .
  1. If  $c(X) = 0$ , we get a contradiction since then  $\varphi_{g(v)}^c = \perp$ .
  2. If  $c(X) = \alpha + 1$  for some  $\alpha$ , then S chooses  $\alpha$  as the new clock for  $X$  in  $(\mathcal{A}, w, c)$ . Now  $\varphi_{g(v)}^c = \varphi_u^{c\alpha}$  so  $(\mathcal{A}, w) \models \varphi_u^{c\alpha}$ .

26 *Formula size games for modal logic and  $\mu$ -calculus*

3. If  $c(X)$  is a limit ordinal, then

$$\varphi_{g(v)}^c = \bigvee_{\alpha < c(X)} \varphi_u^{c\alpha} \quad \text{so} \quad (\mathcal{A}, w) \models \bigvee_{\alpha < c(X)} \varphi_u^{c\alpha}.$$

S chooses the new clock  $\alpha$  such that  $(\mathcal{A}, w) \models \varphi_u^{c\alpha}$  holds.

- On the other side, by induction hypothesis, for every  $\mathfrak{B} = (\mathcal{B}, w, c) \in \mathbb{B}$ ,  $(\mathcal{B}, w) \not\models \varphi_{g(w)}^c$ . We again have three cases.
  1. If  $c(X) = 0$ ,  $\mathfrak{B}$  will be removed from the game and can be disregarded.
  2. Let  $c(X) = \alpha + 1$ . Now  $\varphi_{g(v)}^c = \varphi_u^{c\alpha}$  so  $(\mathcal{B}, w) \not\models \varphi_u^{c\alpha}$ . By Lemma 5.3,  $(\mathcal{B}, w) \not\models \mu^{\alpha+1}X.\varphi_u^{c'}(X)$ , where  $c' = c_\alpha[-/X]$ . Let  $\beta \leq \alpha$  be the choice of D for the new clock. Now by monotonicity,  $(\mathcal{B}, w) \not\models \mu^{\beta+1}X.\varphi_u^{c'}(X)$ . We use Lemma 5.3 again and obtain  $(\mathcal{B}, w) \not\models \varphi_u^{c\beta}$ .
  3. Finally, let  $c(X)$  be a limit ordinal. Now

$$\varphi_{g(v)}^c = \bigvee_{\alpha < c(X)} \varphi_u^{c\alpha} \quad \text{so} \quad (\mathcal{B}, w) \not\models \bigvee_{\alpha < c(X)} \varphi_u^{c\alpha}.$$

Thus,  $(\mathcal{B}, w) \not\models \varphi_u^{c\alpha}$  for any  $\alpha < c(X)$  D chooses.

- Uniformity is trivial here since no new vertices were created.
- The case where  $X$  is a  $\nu$ -variable is the same with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.

$\nu \in \mathbf{dom}(\mathbf{lab})$  : The moves are essentially the same as in the unlabelled case. The main differences are that the type of the move is already determined by  $\mathbf{lab}(\nu)$ , and the resource splittings are already fixed. In disjunction and conjunction moves, new models can be left to wait in the branch not chosen by D as the following position. We will consider only this special case of waiting new models here.

$\mathbf{lab}(\nu) = \vee$  : S chooses the sets  $\mathbb{A}_1$  and  $\mathbb{A}_2$  of new models as in the unlabelled case. There may however be some new models present in  $\nu_1$  or  $\nu_2$ . If so, these models are there because of previous  $\vee$ -moves, for the first of which  $\nu$  had no label. By induction hypothesis and the unlabelled case, (\*) held for both of the possible following positions and therefore  $(\mathcal{A}, w) \models \varphi_{s_i}^c$  for every model  $(\mathcal{A}, w, c)$  in the corresponding left model set. Inductively, we see that  $(\mathcal{A}, w) \models \varphi_{s_i}^c$  for every  $(\mathcal{A}, w, c) \in \mathbf{left}(\nu_i)$ . The same argument shows that  $(\mathcal{B}, w) \not\models \varphi_{s_i}^c$  for every  $(\mathcal{B}, w, c) \in \mathbf{right}(\nu_i)$ . Thus, (\*) holds for the sets  $\mathbb{A}_i \cup \mathbf{left}(\nu_i)$  and  $\mathbb{B} \cup \mathbf{right}(\nu_i)$  in both of the possible following positions of position  $P$ .

(1)  $\Rightarrow$  (2). Let  $\delta$  be a uniform winning strategy for S. Let  $\varphi \in L_\mu(\Phi)$  be the formula  $\delta$  follows. We denote the position embedding showing the uniformity of  $\delta$  in each position by  $g$ . By Lemma 5.5, every play of the game is finite so the game tree induced by the strategy  $\delta$  is well founded. We prove by well founded induction on the game positions reachable with  $\delta$  that the same condition (\*) as above holds in every position of the game.

$$\begin{aligned} (\mathcal{A}, w) \models \varphi_{g(v)}^c & \text{ for every } (\mathcal{A}, w, c) \in \mathbb{A}_{\mathbf{new}} \text{ and} \\ (\mathcal{B}, w) \not\models \varphi_{g(v)}^c & \text{ for every } (\mathcal{B}, w, c) \in \mathbb{B}_{\mathbf{new}}, \end{aligned} \quad *$$

In a position  $P$  reachable with  $\delta$ , let  $\mathbf{left}(\nu) = \mathbb{A}$ ,  $\mathbf{right}(\nu) = \mathbb{B}$  and  $\mathbf{res}(\nu) = k$ . We again consider the unlabelled and labelled case separately and only treat one of each pair of dual moves.

$\nu \notin \mathbf{dom}(\mathbf{lab})$  :

- $\mathbf{lab}(g(v)) \in \mathbf{Lit}(\Phi)$ : Since  $\delta$  follows  $\varphi$ , the next move according to  $\delta$  is a *Lit* move choosing that literal. Since  $\delta$  is a winning strategy, that literal separates the sets  $\mathbb{A}$  and  $\mathbb{B}$  and (\*) holds.

- $\text{lab}(g(v)) = \vee$ : Let  $s_1$  and  $s_2$  be the successors of  $g(v)$ . Let  $\mathbb{A}_1, \mathbb{A}_2, k_1$  and  $k_2$  be the selections of S according to  $\delta$ . By induction hypothesis, (\*) holds in both possible following positions so for every  $(\mathcal{A}, w, c) \in \mathbb{A}_i$ ,  $(\mathcal{A}, w) \models \varphi_{s_i}^c$  for  $i \in \{1, 2\}$ . Since  $\mathbb{A} = \mathbb{A}_1 \cup \mathbb{A}_2$ , for every  $(\mathcal{A}, w, c) \in \mathbb{A}$ ,  $(\mathcal{A}, w) \models \varphi_{s_1}^c \vee \varphi_{s_2}^c$ . In addition,  $\varphi_{s_1}^c \vee \varphi_{s_2}^c = \varphi_{g(v)}^c$  so  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ . Let  $(\mathcal{B}, w, c) \in \mathbb{B}$ . By (\*) in the following positions,  $(\mathcal{B}, w) \not\models \varphi_{s_1}^c$  and  $(\mathcal{B}, w) \not\models \varphi_{s_2}^c$  so  $(\mathcal{B}, w) \not\models \varphi_{s_1}^c \vee \varphi_{s_2}^c$ . Since  $\varphi_{s_1}^c \vee \varphi_{s_2}^c = \varphi_{g(v)}^c$ ,  $(\mathcal{B}, w) \not\models \varphi_{g(v)}^c$ . Thus, (\*) holds in  $P$ .
- $\text{lab}(g(v)) = \diamond$ : Let  $s_1$  be the successor of  $g(v)$ . Let  $f : \mathbb{A} \rightarrow \square \mathbb{A}$  be the function chosen by S according to  $\delta$ . Let  $(\mathcal{A}, w, c) \in \mathbb{A}$ . By induction hypothesis, (\*) holds in the following position no matter which subsets of  $\diamond_f \mathbb{A}$  and  $\square \mathbb{B}$  D chooses so for  $f(\mathcal{A}, w, c) = (\mathcal{A}, w', c) \in \diamond_f \mathbb{A}$ ,  $(\mathcal{A}, w') \models \varphi_{s_1}^c$ . Since  $w'$  is a successor of  $w$ , now  $(\mathcal{A}, w) \models \diamond \varphi_{s_1}^c$ . In addition,  $\diamond \varphi_{s_1}^c = \varphi_{g(v)}^c$  so  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ . Let  $(\mathcal{B}, w, c) \in \mathbb{B}$ . By induction hypothesis, (\*) holds for all possible following positions so for every  $(\mathcal{B}, w', c) \in \square \mathbb{B}$ ,  $(\mathcal{B}, w') \not\models \varphi_{s_1}^c$ . Therefore,  $(\mathcal{B}, w) \not\models \diamond \varphi_{s_1}^c$  and so  $(\mathcal{B}, w) \not\models \varphi_{g(v)}^c$ . Thus, (\*) holds in  $P$ .
- $\text{lab}(g(v)) = \mu X$ : Let  $s_1$  be the successor of  $g(v)$ . Let  $\mathfrak{A} = (\mathcal{A}, w, c) \in \mathbb{A}$  and let  $\alpha_{\mathfrak{A}} = \alpha$  be the choice of S according to  $\delta$ . By induction hypothesis, (\*) holds in the following position so  $(\mathcal{A}, w) \models \varphi_{s_1}^{\alpha}$ . By Lemma 5.3,  $(\mathcal{A}, w) \models \mu^{\alpha+1} X. \varphi_{s_1}^c$ . Thus,  $(\mathcal{A}, w) \models \mu X. \varphi_{s_1}^c$ . Since  $\mu X. \varphi_{s_1}^c = \varphi_{g(v)}^c$ ,  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ . On the other side, by induction hypothesis, for every  $\mathfrak{B} = (\mathcal{B}, w, c) \in \mathbb{B}$  and for any choice  $\alpha$  of D for the new clock,  $(\mathcal{B}, w) \not\models \varphi_{s_1}^{\alpha}$ . Thus by Lemma 5.3,  $(\mathcal{B}, w) \not\models \mu^{\alpha+1} X. \varphi_{s_1}^c(X)$  for every  $\alpha < \kappa$ . Since  $\kappa > \text{card}(\mathcal{B})$ , this means that  $(\mathcal{B}, w) \not\models \mu X. \varphi_{s_1}^c$  and so  $(\mathcal{B}, w) \not\models \varphi_{g(v)}^c$ . Thus, (\*) holds in  $P$ .
- $\text{lab}(g(v)) = X$ : If  $\mathbb{A}_{\text{new}} = \mathbb{B}_{\text{new}} = \emptyset$ , S wins and (\*) trivially holds. Let  $u$  be the  $B$ -successor of  $g(v)$ . Assume that  $X$  is a  $\mu$ -variable and let  $(\mathcal{A}, w, c) \in \mathbb{A}$ . We have three cases according to the ordinal  $c(X)$ .
  1. If  $c(X) = 0$ , D wins the game, which is a contradiction, since  $\delta$  is a winning strategy for S.
  2. Assume that  $c(X) = \alpha + 1$ . Let  $\beta \leq \alpha$  be the choice of S for the new clock according to  $\delta$ . By induction hypothesis, (\*) holds in the following position so  $(\mathcal{A}, w) \models \varphi_u^{c_\beta}$ . By Lemma 5.3,  $(\mathcal{A}, w) \models \mu^{\beta+1} X. \varphi_u^{c'}$ , where  $c' = c_\beta[-/X]$ . Thus by monotonicity,  $(\mathcal{A}, w) \models \mu^{\alpha+1} X. \varphi_u^{c'}$ . By Lemma 5.3 again,  $(\mathcal{A}, w) \models \varphi_u^{c_\alpha}$  and so  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ .
  3. Now assume  $c(X)$  is a limit ordinal and let  $\alpha < c(X)$  be the choice of S according to  $\delta$ . Now by induction hypothesis  $(\mathcal{A}, w) \models \varphi_u^{c_\alpha}$ . Thus,

$$(\mathcal{A}, w) \models \bigvee_{\alpha < c(X)} \varphi_u^{c_\alpha}$$

so  $(\mathcal{A}, w) \models \varphi_{g(v)}^c$ .

- For every  $(\mathcal{B}, w, c) \in \mathbb{B}$ , regardless of the choice of D for the new clock, (\*) holds in the following position. We again have three cases.
  1. If  $c(X) = 0$ , then since  $(\mathcal{B}, w) \not\models \perp$ , we get  $(\mathcal{B}, w) \not\models \varphi_{g(v)}^c$ .
  2. If  $c(X) = \alpha + 1$ , then  $\alpha$  is a choice available to D so  $(\mathcal{B}, w) \not\models \varphi_u^{c_\alpha}$ . Thus,  $(\mathcal{B}, w) \not\models \varphi_{g(v)}^c$ .
  3. If  $c(X)$  is a limit ordinal, every  $\alpha < c(X)$  is a choice available to D so  $(\mathcal{B}, w) \not\models \varphi_u^{c_\alpha}$  for every  $\alpha < c(X)$ . Thus,

$$(\mathcal{B}, w) \not\models \bigvee_{\alpha < c(X)} \varphi_u^{c_\alpha}$$

Therefore,  $(\mathcal{B}, w) \not\models \varphi_{g(v)}^c$  so (\*) holds in  $P$ .

## 28 Formula size games for modal logic and $\mu$ -calculus

$v \in \mathbf{dom}(\mathbf{lab})$  : All the moves in this case are proved the same way as in the unlabelled case. Note that since  $(*)$  refers only to new models, it trivially holds for terminal positions where  $\mathbb{A}_{\mathbf{new}} = \mathbb{B}_{\mathbf{new}} = \emptyset$  for an  $X$ -move.

For the very first move of the game, where D chooses finite subsets of the original sets of clocked models  $\mathbf{CM}(\mathbb{A}_0)$  and  $\mathbf{CM}(\mathbb{B}_0)$ , by induction hypothesis  $(*)$  holds in the following position no matter which subsets D chooses. Therefore, all models in  $\mathbb{A}_0$  and  $\mathbb{B}_0$  also satisfy the condition  $(*)$ . Since there are no clocks in the starting position, this means that  $\varphi$  separates the sets  $\mathbb{A}_0$  and  $\mathbb{B}_0$ . By the uniformity of  $\delta$ ,  $\text{sz}(\varphi) = \text{sz}(\varphi_{g(v_0)}) \leq \text{res}(v_0) = k$ .  $\square$

Note that condition  $(*)$  does not depend on old models and so we do not refer to them in this proof. We add old models to the game to make the proof of Lemma 6.3 in the next section easier.

Unlike other similar theorems, Theorem 5.7 has the added requirement of uniformity for the strategy of S. We conjecture that the theorem would still hold even without this condition but proving this has turned out to be difficult. Note, however, that to prove undefinability results for  $L_\mu(\Phi)$ , one need only define a winning strategy for D, and so the uniformity of strategies for S need not be considered. Note further that if a property of  $\Phi$ -models is not definable in  $L_\mu(\Phi)$ , then clearly it is not definable in  $L_\mu$ .

## 6 Succinctness of FO over $L_\mu$

We move on to the definitions and lemmas needed to show that FO is non-elementarily more succinct than  $L_\mu$ . We need a lemma similar to Lemma 3.4 that gives D a winning strategy if bisimilar models are produced on both sides of a vertex. In the case of the  $L_\mu$ -game, the clocks of the clocked models must also be taken into account. We define a sufficient condition for clocked models to be useful for D in the game and call them relevant models.

### DEFINITION 6.1

The *depth* of a pointed finite tree model  $(\mathcal{M}, w)$ ,  $d(\mathcal{M}, w)$ , is the length of a maximal path of transitions in the model starting from  $w$ .

### DEFINITION 6.2

In a position  $P$  of a game  $\mu\text{-FS}_k^\Phi(\mathbb{A}, \mathbb{B})$ , let  $u$  be a vertex and let  $\mathfrak{M} = (\mathcal{M}, w, c, a)$  be a clocked finite tree model in  $\text{left}(u) \cup \text{right}(u)$ . We say the model  $\mathfrak{M}$  is *relevant*, if for every  $X \in \text{dom}(c)$ , the clock of D is equal to or greater than the depth of the model, i.e.

- if  $\mathfrak{M} \in \text{left}(u)$ , for every  $v$ -variable  $X \in \text{dom}(c)$ ,  $c(X) \geq d(\mathcal{M}, w)$ ,
- if  $\mathfrak{M} \in \text{right}(u)$ , for every  $\mu$ -variable  $X \in \text{dom}(c)$ ,  $c(X) \geq d(\mathcal{M}, w)$ .

We also say the model  $\mathfrak{M}$  is *strictly relevant* if the above condition holds for strict inequality  $>$  instead of  $\geq$ .

We prove the analogue of Lemma 3.4 for relevant clocked models.

### LEMMA 6.3

Let  $P$  be a position of a game  $\mu\text{-FS}_k^\Phi(\mathbb{A}, \mathbb{B})$ . If there are strictly relevant clocked models  $(\mathcal{A}, w_{\mathcal{A}}, c_{\mathcal{A}}, \mathbf{new}) \in \text{left}(v)$  and  $(\mathcal{B}, w_{\mathcal{B}}, c_{\mathcal{B}}, \mathbf{new}) \in \text{right}(v)$  such that  $(\mathcal{A}, w_{\mathcal{A}})$  and  $(\mathcal{B}, w_{\mathcal{B}})$  are bisimilar finite tree models, then D has a winning strategy from position  $P$ .

PROOF. We show that D can maintain a slightly modified condition where we only require the models to be relevant and allow one of the two models to be old. For  $\vee$ - and  $\wedge$ -moves D need only choose the side for which the two models are both present. For modal moves, we see by bisimilarity that no matter which successor S chooses, a bisimilar model will be present on the opposite side. Moreover, the depth of the models is decreased and the clocks are inherited so the condition is maintained. For new fixed points, D need only set her clock to be equal to the depth of the models. On a literal move, D will win since the bisimilar models cannot be separated by a literal.

For  $X$ -moves, let  $u$  be the vertex the game returns to. If this is the first time since position  $P$  the game returns to  $u$ , D will lower the clock and since the models are strictly relevant, D can now decrease the clock to the same value as the depth. If on the other hand there has been a previous return to  $u$ , then there are two cases. If there has been a modal move in between this and the previous return, then the depth has decreased and D will decrease the clock to the same value. If there have been no modal moves, the pointed models have not changed and since we allow one of the models to be old, D will now consider the old version of her model, with a clock larger by one, instead of the new one. Consider the position  $P'$  right after D switches a new model for an old one in this fashion. Assume by symmetry that this model is  $\mathfrak{B}$  on the right side. Consider the path from the current vertex  $u$ , to the vertex  $s$  where the  $X$ -move was made. If there are no vertices with label  $\vee$  on this path, then D can just follow this path to  $s$  until the clock  $c_{\mathcal{A}}(X)$  runs out and D wins. Assume there are some vertices with label  $\vee$  on the path. For each of those vertices, the child that is not on the path from  $u$  to  $s$  has a new version of  $\mathfrak{B}$  in the right set. This is because the model  $\mathfrak{B}$  has passed through the disjunction before the  $X$ -move and models on the right side are always copied on both sides of a disjunction. If S splits the left model  $\mathfrak{A}$  away from the path to  $s$ , then D will consider the new copy of  $\mathfrak{B}$  from then on. If  $\mathfrak{A}$  stays on the path from  $u$  to  $s$  indefinitely, D wins when the clock  $c_{\mathcal{A}}(X)$  runs out.

If D uses this strategy, eventually S will either make a literal move and lose, or a clock of S for one of the bisimilar models will eventually run out. In either case, D wins.  $\square$

We want to use the same graph-based invariant for the proof as we did for the ML-case. The only question that remains is, which models should determine the graph of the current vertex  $v$ ? In other words, which models is S claiming to be able to separate in each position of the game? Certainly the models in  $\text{left}(v)$  and  $\text{right}(v)$  should be included, but they will not suffice, since other models can already be below  $v$  and thus involved with the subformula beginning from  $v$ . We define a way for D to collect all the models in the tree below a vertex  $s$  to see which models are, in a sense, ‘currently in  $s$ ’. We define the collected sets separately for the left and the right side. Recall that  $\text{PM}(\mathbb{A})$  is the set of underlying pointed models of clocked models in  $\mathbb{A}$ .

#### DEFINITION 6.4

Let  $P$  be a position such that  $\text{lab}(v)$  is not a literal and no vertices have label  $\diamond$  or  $\square$ . For each vertex  $s \in V$ , we define the *left collection of  $s$  in  $P$* , denoted by  $\mathbb{L}(s)$ , recursively starting from the leaves of  $(V, E, B)$ :

- if  $s$  is an unlabelled leaf, then  $\mathbb{L}(s) = \text{PM}(\text{left}(s))$ ,
- if  $\text{lab}(s)$  is a  $\mu$ -variable, then  $\mathbb{L}(s) = \emptyset$ ,
- if  $\text{lab}(s)$  is a  $\nu$ -variable, then  $\mathbb{L}(s) = \text{PM}(\text{left}(s))$ ,
- if  $\text{lab}(s) = \eta X$  and the successor is  $s_1$ , then  $\mathbb{L}(s) = \text{PM}(\text{left}(s)_{\text{new}}) \cup \mathbb{L}(s_1)$ ,
- if  $\text{lab}(s) = \vee$  and the successors are  $s_1, s_2$ , then  $\mathbb{L}(s) = \text{PM}(\text{left}(s)_{\text{new}}) \cup \mathbb{L}(s_1) \cup \mathbb{L}(s_2)$
- if  $\text{lab}(s) = \wedge$  and the successors are  $s_1, s_2$ , then  $\mathbb{L}(s) = \text{PM}(\text{left}(s)_{\text{new}}) \cup (\mathbb{L}(s_1) \cap \mathbb{L}(s_2))$ .

### 30 Formula size games for modal logic and $\mu$ -calculus

The *right collection of  $s$  in  $P, \mathbb{R}(s)$* , is defined symmetrically with left and right, as well as  $\mu$  and  $\nu$ , switched at every point of the definition.

Note that since  $\nu$  does not have a literal label, a leaf of  $(V, E, B)$  can only be either an unlabelled vertex or a vertex with a variable label. This is because any vertex labelled with an operator will have at least one successor and the game ends in the next position after any literal move.

In the following, we will associate superscripted sets like  $\mathbb{L}'(s)$  with the position  $P'$  with the same superscript just like we have done so far with the components of the position.

#### LEMMA 6.5

Let  $P^\circ$  be a position in a game  $\mu\text{-FS}_k^\phi(\mathbb{A}, \mathbb{B})$ , where the current vertex  $u = \nu^\circ$  is a successor of a fixed point vertex and no modal moves are made. Let  $P'$  be a position after  $P^\circ$  such that no  $X$ -move has returned to a vertex above  $u$  since  $P^\circ$ . Then  $\mathbb{L}^\circ(u) \subseteq \mathbb{L}'(u)$  and  $\mathbb{R}^\circ(u) \subseteq \mathbb{R}'(u)$ .

PROOF. The proof proceeds by induction. We assume that for position  $P$ ,  $\mathbb{L}^\circ(u) \subseteq \mathbb{L}(u)$  and  $\mathbb{R}^\circ(u) \subseteq \mathbb{R}(u)$  and we show that the inclusion also holds for the next position  $P'$ .

If  $S$  makes a  $\eta X$ -move, the new models in  $\text{left}(\nu)$  are moved to  $\text{left}'(\nu')$  but they still remain in  $\mathbb{L}'(\nu)$  so  $\mathbb{L}'(u) = \mathbb{L}(u)$ . Note that clocks of the models do change but  $\mathbb{L}'(\nu)$  only looks at the underlying pointed models. For the same reasons,  $\mathbb{R}'(u) = \mathbb{R}(u)$ .

For  $\vee$ -moves, the models in  $\text{left}(\nu)$  are split among the successors  $\nu_1$  and  $\nu_2$  so they are still in  $\mathbb{L}'(\nu)$  in position  $P'$ . The models in  $\text{right}(\nu)$  are copied to both  $\nu_1$  and  $\nu_2$  so they are still in  $\mathbb{R}'(\nu)$ . Thus,  $\mathbb{L}'(u) = \mathbb{L}(u)$  and  $\mathbb{R}'(u) = \mathbb{R}(u)$ . The case of  $\wedge$ -moves is symmetric with the two sides switched everywhere.

If  $S$  makes an  $X$ -move, it can either be a return to  $u$  or to a vertex  $s$  below  $u$ . Assume that the return is made to  $u$  and that  $X$  is a  $\mu$ -variable. Now the models in  $\text{left}(\nu)$  are moved to  $\text{left}'(u)$  so any that were already in  $\mathbb{L}(u)$  stay there and more may be added so  $\mathbb{L}(u) \subseteq \mathbb{L}'(u)$ . The models in  $\text{right}(\nu)$  cease to be relevant in  $\text{right}'(u)$  but they remain as old models in  $\text{right}'(\nu)$  and are still counted for  $\mathbb{R}'(\nu)$  in  $P'$  as they were in  $P$  so  $\mathbb{R}'(u) = \mathbb{R}(u)$ . The case of a  $\nu$  is symmetric with the two sides switched.

Finally assume that an  $X$ -move is made returning to a vertex  $s$  below  $u$ . Assume again that  $X$  is a  $\mu$ -variable. The models in  $\text{left}(\nu)$  are moved to  $\text{left}'(s)$  so  $\mathbb{L}(s) \subseteq \mathbb{L}'(s)$ . Everything not below  $s$  remains unchanged so  $\mathbb{L}(u) \subseteq \mathbb{L}'(u)$ . The models in  $\text{right}(\nu)$  cease to be relevant in  $\text{right}'(s)$  but they remain as old models in  $\text{right}'(\nu)$  and are still counted for  $\mathbb{R}'(s)$  and therefore also for  $\mathbb{R}'(u)$  just like in  $P$ . Thus,  $\mathbb{R}'(u) = \mathbb{R}(u)$ . The case of a  $\nu$  is again symmetric.  $\square$

We finally have all of the required notation and lemmas to show that the non-elementary succinctness gap is present also between FO and the modal  $\mu$ -calculus.

#### THEOREM 6.6

First-order logic is non-elementarily more succinct than the modal  $\mu$ -calculus.

PROOF. We prove an analogous result to Lemma 4.9 for the  $L_\mu$  game. We use the notation  $\mathcal{G}(\mathbb{V}, \mathbb{E})$  for the same graph as in Subsection 4.3. The precise statement we prove is as follows:

Let  $n \in \mathbb{N}$  and let  $k_0 \in \mathbb{N}$ . If  $k_0 < \log(\chi(\mathcal{G}(\mathbb{C}_n, \mathbb{D}_n)))$ , then  $D$  has a winning strategy in the game  $\mu\text{-FS}_{k_0}^\phi(\mathbb{C}_n, \mathbb{D}_n)$ .

In this proof, we only consider relevant models. Many positions in the game  $\mu\text{-FS}_{k_0}^\phi(\mathbb{C}_n, \mathbb{D}_n)$  have also non-relevant models but they are not needed for the strategy of  $D$  we describe here and can safely be ignored. We will assume all models are relevant and occasionally comment on why models remain or cease to be relevant after some moves of the game.



We show by induction that D has a strategy to maintain the following condition in any position  $P = (V, E, B, \text{lab}, \text{res}, \text{left}, \text{right}, v)$ :

$$\text{res}(v) < \log(\chi(\mathcal{G}(\mathbb{L}(v), \mathbb{R}(v)))) \quad (*)$$

At the start of the game,  $(*)$  holds by assumption. Since the sets  $\mathbb{C}_n$  and  $\mathbb{D}_n$  are already finite, D can keep the full sets for the first move of the game.

We first show that if S ever makes a modal move while  $(*)$  holds, D gets a winning strategy for the game. We assume  $v \notin \text{dom}(\text{lab})$  since the first modal move in a game must always be made in an unlabelled vertex. In this case there are no other vertices below  $v$  so  $\mathbb{L}(v) = \text{left}(v)$  and  $\mathbb{R}(v) = \text{right}(v)$ . We assume  $\text{res}(v) > 1$  so that S can make a modal move and not lose immediately due to the resource running out. From  $(*)$ , we obtain  $\chi(\mathcal{G}(\text{left}(v), \text{right}(v))) > 2$  so there are relevant clocked models  $(\Delta(\mathcal{M}_1, w_1), c_1, \text{new}), (\Delta(\mathcal{M}_2, w_2), c_2, \text{new}) \in \text{left}(v)$  and  $(\Delta\{\mathcal{M}_1, w_1\}, \mathcal{M}_2, w_2\}, c_3, \text{new}) \in \text{right}(v)$ . Now if S makes a  $\diamond$ - or  $\square$ -move, then in the following position  $P'$  there is  $i \in \{1, 2\}$  s.t.  $(\mathcal{M}_i, w_i, c_i, \text{new}) \in \text{left}(v')$  and  $(\mathcal{M}_i, w_i, c_3, \text{new}) \in \text{right}(v')$ . As these two share the same underlying pointed model they are bisimilar and moreover, since the depth has decreased by at least 1 from the previous position, the models are strictly relevant. By Lemma 6.3, D now has a winning strategy from position  $P'$ .

If S makes a  $\vee$ -move, let  $v_1$  and  $v_2$  be the successors of  $v$ . In the following position, whichever it may be, we have  $\mathbb{L}(v) = \mathbb{L}(v_1) \cup \mathbb{L}(v_2)$  and  $\mathbb{R}(v) = \mathbb{R}(v_1) \cap \mathbb{R}(v_2)$ . Let  $\mathcal{G}_s = \mathcal{G}(\mathbb{L}(s), \mathbb{R}(s)) = (V_s, E_s)$  for  $s \in \{v, v_1, v_2\}$ . We obtain  $V_v = V_{v_1} \cup V_{v_2}$  and  $E_v \cap (V_{v_i} \times V_{v_i}) \subseteq E_{v_i}$  for  $i \in \{1, 2\}$ . By Lemma 4.8,

$$\chi(\mathcal{G}_v) \leq \chi(V_{v_1}, E_v \cap (V_{v_1} \times V_{v_1})) + \chi(V_{v_2}, E_v \cap (V_{v_2} \times V_{v_2})) \leq \chi(\mathcal{G}_{v_1}) + \chi(\mathcal{G}_{v_2}).$$

Thus (just like in the proof of Theorem 4.9), we obtain  $\text{res}(v_i) < \log(\chi(\mathcal{G}_{v_i}))$  for some  $i \in \{1, 2\}$  so  $(*)$  holds in the following position after D chooses this  $i$ .

If S makes a  $\wedge$ -move, let  $v_1$  and  $v_2$  be the successors of  $v$ . In the following position, we have  $\mathbb{L}(v) = \mathbb{L}(v_1) \cap \mathbb{L}(v_2)$  and  $\mathbb{R}(v) = \mathbb{R}(v_1) \cup \mathbb{R}(v_2)$ . We use the notation  $\mathcal{G}_s = (V_s, E_s)$  from the previous case and obtain  $V_v = V_{v_1} \cap V_{v_2}$  and  $E_v = (E_{v_1} \cap (V_v \times V_v)) \cup (E_{v_2} \cap (V_v \times V_v))$ . By Lemma 4.8,

$$\chi(\mathcal{G}_v) \leq \chi(V_{v_1}, E_{v_1} \cap (V_v \times V_v)) \chi(V_{v_2}, E_{v_2} \cap (V_v \times V_v)) \leq \chi(\mathcal{G}_{v_1}) \chi(\mathcal{G}_{v_2}).$$

Thus, we again obtain  $\text{res}(v_i) < \log(\chi(\mathcal{G}_{v_i}))$  for some  $i \in \{1, 2\}$  so  $(*)$  holds in the following position after D chooses this  $i$ .

If S makes a  $\eta X$ -move, where  $\eta \in \{\mu, v\}$ , then D sets her clock for each model at the same value as the depth of the model. All relevant models remain relevant and  $(*)$  is maintained.

If S makes an  $X$ -move, by  $(*)$ , S does not immediately win the game. Assume that  $u$  is the vertex returned to and  $P^\circ$  is the previous position when  $u$  was the current vertex. Let  $P'$  be the position after this  $X$ -move. By Lemma 6.5, we obtain  $\mathbb{L}^\circ(u) \subseteq \mathbb{L}'(u)$  and  $\mathbb{R}^\circ(u) \subseteq \mathbb{R}'(u)$ . By induction hypothesis,  $(*)$  held in  $P^\circ$  and clearly  $\text{res}'(u) = \text{res}^\circ(u)$ , so  $(*)$  still holds in  $P'$ .

If S makes a *Lit*-move, by  $(*)$ ,  $\text{left}(v) \neq \emptyset$  and  $\text{right}(v) \neq \emptyset$ . Since all the models are propositionally equivalent, D wins the game.

By Theorem 5.7, we obtain that there is no sentence  $\varphi \in L_\mu(\emptyset)$  that separates  $\mathbb{C}_n$  from  $\mathbb{D}_n$  with  $\text{sz}(\varphi) \leq \log(\chi(\mathcal{G}(\mathbb{C}_n, \mathbb{D}_n))) = \text{twr}(n - 1)$ . Thus, FO is non-elementarily more succinct than  $L_\mu$ .  $\square$

#### REMARK 6.7

Just like in the case of ML, we remark that the result of Theorem 6.6 also holds for DAG-size.

This is again because the difference between the size of an  $L_\mu$  formula in our sense and the DAG-size of the same formula is at most exponential.

## 7 Conclusion

We have defined formula size games for basic modal logic and the modal  $\mu$ -calculus. The games utilize resource parameters to achieve a truly two-player game. In the case of modal logic, the players only construct one branch of the game tree. This is in contrast with the original Adler–Immerman game, where the players form the whole tree in a single play. For the modal  $\mu$ -calculus, the recursive nature of fixed point operators necessitates returning to previously visited nodes. However, the game still traverses only one possible path through the game tree in a single play and some branches can remain unvisited for the entire play. The  $\mu$ -calculus game has infinite branching but the use of decreasing ordinal clocks, as in [16], makes each play of the game finite.

We used the games to show that the property ‘all successor models are  $n$ -bisimilar with each other’ cannot be defined in basic modal logic or the modal  $\mu$ -calculus with a formula of size less than the exponential tower of height  $n - 1$ . On the other hand, this property can be defined in FO with a formula of size linear in  $n$ . This means that FO is non-elementarily more succinct than both ML and  $L_\mu$ . We also show that the same property can be defined in two-dimensional modal logic  $ML^2$  with a formula of size exponential in  $n$ . Thus, the non-elementary succinctness gap is also present between  $ML^2$  and both ML and  $L_\mu$ .

We find the ML-game to be a useful tool for proving lower bounds on the size of ML-formulas. Depending on the desired result, the game can also be modified to count a more specific parameter instead of formula size, such as the number or nesting depth of a specific operator.

The  $\mu$ -calculus game is also functional for proving lower bounds but with some caveats. The main theorem stating the usefulness of the game, Theorem 5.7, requires uniform strategies for S. This means that we assume S has a single formula in mind and always plays according to that formula. It may be that this restriction could be removed but we have been unable to prove this. However, to show succinctness results we only need one direction of the equivalence so the issue is usually not relevant in practice. The greater concern is whether the game can be successfully used to prove succinctness results for  $\mu$ -calculus in more complicated contexts. Here we only generalize a result already obtained with the ML game and we have so far failed to produce any other results with the  $L_\mu$  game due to its sheer complexity. A question related to this difficulty would be whether the game could be simplified significantly while still preserving its functionality. It would be especially interesting to apply the game to open problems related to  $\mu$ -calculus and succinctness, such as whether there is a polynomial transformation from  $L_\mu$  to the guarded fragment or from vectorial form to regular  $L_\mu$  [4].

## Acknowledgements

We are grateful to Martin Lück for coming up with a linear size formula  $\psi_n(x, y)$  to replace our previous one that was of exponential size.

## References

- [1] M. Adler and N. Immerman. An  $n!$  lower bound on formula size. *ACM Transactions on Computational Logic*, **4**, 296–314, 2003.
- [2] P. Blackburn, M. de Rijke and Y. V. Venema. *Modal Logic*. Cambridge University Press, New York, NY, USA, 2001.

- [3] J. Bradfield and C. Stirling. Chapter 12—modal mu-calculi. In *Handbook of Modal Logic*, P. Blackburn, J. V. Benthem, and F. Wolter, eds, Studies in Logic and Practical Reasoning, vol. 3, pp. 721–756. Elsevier, 2007.
- [4] F. Bruse, O. Friedmann and M. Lange. On guarded transformation in the modal mu-calculus. *Logic Journal of the IGPL*, **23**, 194–216, 2015.
- [5] K. Etessami, M. Y. Vardi and T. Wilke. First-order logic with two variables and unary temporal logic. *Information and Computation*, **179**, 279–295, 2002.
- [6] K. Etessami and T. Wilke. An until hierarchy and other applications of an Ehrenfeucht-Fraïssé game for temporal logic. *Information and Computation*, **160**, 88–108, 2000.
- [7] D. Fernández-Duque and P. Iliev. Succinctness in subsystems of the spatial mu-calculus. *Journal of Applied Logics - IfCoLoG Journal of Logics and their Applications*, College Publications, **5**(4), 827–873, 2018.
- [8] T. French, W. van der Hoek, P. Iliev and B. P. Kooi. Succinctness of epistemic languages. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16–22, 2011*, pp. 881–886, 2011.
- [9] T. French, W. van der Hoek, P. Iliev and B. P. Kooi. On the succinctness of some modal logics. *Artificial Intelligence*, **197**, 56–85, 2013.
- [10] D. Gabbay and V. Shehtman. Products of modal logics, part 1. *Logic Journal of the IGPL*, **6**, 73–146, 1998.
- [11] D. Gabbay and V. Shehtman. Products of modal logics. Part 2: relativised quantifiers in classical logic. *Logic Journal of the IGPL*, **8**, 165–210, 2000.
- [12] D. Gabbay and V. Shehtman.. Products of modal logics. Part 3: products of modal and temporal logics. *Studia Logica*, **72**, 157–183, 2002.
- [13] G. Gogic, H. Kautz, C. Papadimitriou and B. Selman. The comparative linguistics of knowledge representation. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence—Volume 1, IJCAI’95*, pp. 862–869. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.
- [14] M. Grohe and N. Schweikardt. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science*, **1**, 2005.
- [15] M. Grohe and N. Schweikardt. Comparing the succinctness of monadic query languages over finite trees. *RAIRO Theoretical Informatics and Applications*, **38**, 343–373, 2004.
- [16] L. Hella, A. Kuusisto and R. Rönholm. Bounded game-theoretic semantics for modal mu-calculus. ArXiv e-prints, 2017.
- [17] L. Hella and J. Väänänen. The size of a formula as a measure of complexity. In *Logic Without Borders—Essays on Set Theory, Model Theory, Philosophical Logic and Philosophy of Mathematics*, pp. 193–214. De Gruyter, 2015.
- [18] L. Hella and M. Vilander. The succinctness of first-order logic over modal logic via a formula size game. In *Advances in Modal Logic 11, Budapest, Hungary, August 30–September 2, 2016*, pp. 401–419, 2016.
- [19] M. Karchmer. On proving lower bounds for circuit size. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference*, IEEE, pp. 112–118, 1993.
- [20] G. Lenzi. The modal mu-calculus: a survey. *Task Q*, **9**, 293–316, 2005.
- [21] C. Lutz.. Complexity and succinctness of public announcement logic. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8–12, 2006*, pp. 137–143, 2006.

- [22] C. Lutz, U. Sattler and F. Wolter. Modal logic and the two-variable fragment. In *Computer Science Logic, Paris, France, September 10–13, 2001*, pp. 247–261, 2001.
- [23] N. Markey. Temporal logic with past is exponentially more succinct, concurrency column. *Bulletin of the European Association for Theoretical Computer Science*, **79**, 122–128, 2003.
- [24] M. Marx and Y. Venema. *Multi-dimensional Modal Logic*. Applied Logic Series, vol. **4**. Kluwer Academic Publishers, Dordrecht, 1997.
- [25] J. C. McCabe-Dansted, T. French, S. Pinchinat and M. Reynolds. Expressiveness and succinctness of a logic of robustness. *Journal of Applied Non-Classical Logics*, **25**, 193–228, 2015.
- [26] M. Otto. Bisimulation invariance and finite models. In *Logic Colloquium '02*, Z. Chatzidakis, P. Koepke and W. Pohlers, eds, Lecture Notes in Logic, pp. 276–298. Cambridge University Press, 2006.
- [27] A. A. Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, **10**, 81–93, 1990.
- [28] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, **32**, 733–749, 1985.
- [29] L. J. Stockmeyer. *The Complexity of Decision Problems in Automata Theory and Logic*. PhD Thesis, Massachusetts Institute of Technology, 1974.
- [30] W. van der Hoek and P. Iliev. On the relative succinctness of modal logics with union, intersection and quantification. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS'14, Paris, France, May 5–9, 2014*, pp. 341–348, 2014.
- [31] W. van der Hoek, P. Iliev and B. P. Kooi. On the relative succinctness of two extensions by definitions of multimodal logic. In *How the World Computes—Turing Centenary Conference and 8th Conference on Computability in Europe, CiE 2012, Cambridge, UK, June 18–23, 2012*, Springer-Verlag Berlin Heidelberg, pp. 323–333, 2012.
- [32] H. van Ditmarsch, J. Fan, W. van der Hoek and P. Iliev. Some exponential lower bounds on formula-size in modal logic. In *Advances in Modal Logic 10, Groningen, The Netherlands, August 5–8, 2014*, College Publications, pp. 139–157, 2014.
- [33] H. van Ditmarsch and P. Iliev. The succinctness of the cover modality. *Journal of Applied Non-Classical Logics*, **25**, 373–405, 2015.
- [34] T. Wilke. CTL<sup>+</sup> is exponentially more succinct than CTL. In *Foundations of Software Technology and Theoretical Computer Science, 19th Conference, Chennai, India, December 13–15, 1999*, Springer-Verlag Berlin Heidelberg, pp. 110–121, 1999.

Received 00 Month 20xx

# PUBLICATION

II

## On the Succinctness of Atoms of Dependency

Martin Lück and Miikka Vilander

*Log. Methods Comput. Sci.* 15.3 (2019)

DOI: 10.23638/LMCS-15(3:17)2019

**Publication reprinted with the permission of the copyright holders.**



## ON THE SUCCINCTNESS OF ATOMS OF DEPENDENCY

MARTIN LÜCK<sup>a</sup> AND MIIKKA VILANDER<sup>b</sup>

<sup>a</sup> Leibniz University Hannover  
e-mail address: lueck@thi.uni-hannover.de

<sup>b</sup> Tampere University  
e-mail address: miikka.vilander@tuni.fi

ABSTRACT. Propositional team logic is the propositional analog to first-order team logic. Non-classical atoms of dependence, independence, inclusion, exclusion and anonymity can be expressed in it, but for all atoms except dependence only exponential translations are known. In this paper, we systematically compare their succinctness in the existential fragment, where the splitting disjunction only occurs positively, and in full propositional team logic with unrestricted negation. By introducing a variant of the Ehrenfeucht–Fraïssé game called formula size game into team logic, we obtain exponential lower bounds in the existential fragment for all atoms. In the full fragment, we present polynomial upper bounds also for all atoms.

### 1. INTRODUCTION

As a novel extension of classical logic, *team semantics* provides a framework for reasoning about whole collections of entities at once, as well as their relation with each other. Such a collection of entities is called a *team*. Originally, team semantics was introduced by Hodges [Hod97] to provide a compositional approach to logic of incomplete information, such as Hintikka’s and Sandu’s *independence-friendly logic (IF-logic)* [HS89].

In his seminal work, Väänänen [Vä07] introduced *dependence logic* which extends first-order logic by so-called *dependence atoms*, atomic formulas  $=(x_1, \dots, x_n; y)$  that intuitively express that the value of  $y$  depends only on the values of  $x_1, \dots, x_n$ . While in IF-logic dependencies between variables are expressed with annotated quantifiers such as  $\exists y/\{x_1, \dots, x_n\}$ , in team semantics these can be expressed without changing the quantifiers. Accordingly, dependence logic formulas are evaluated on sets of first-order assignments (called *teams*). Besides the dependence atom, a multitude of other notions of interdependencies between variables were studied, such as the *independence* of variables [GV13], written  $x_1 \cdots x_n \perp y_1 \cdots y_m$ , the *inclusion*  $x_1 \cdots x_n \subseteq y_1 \cdots y_n$  [Gal12], *exclusion*  $x_1 \cdots x_n \mid y_1 \cdots y_n$ , and *anonymity*  $x_1 \cdots x_n \Upsilon y_1 \cdots y_n$  [Vä19], also known as *non-dependence* [Rö18]. We generally refer to these expressions as *atoms of dependency*. In its original formulation, dependence logic does not have a Boolean negation but only a so called dual negation  $\neg$ . For this negation, basic laws such as the law of the excluded middle—that either  $\alpha$  or  $\neg\alpha$  holds in

2012 ACM CCS: Theory of computation  $\rightarrow$  Complexity theory and logic;  
Key words and phrases: team semantics, succinctness, dependence atom.

any given interpretation—fail. By adding a Boolean negation operator, often written  $\sim$ , Väänänen [Vä07] introduced *team logic* as a strictly more powerful extension of dependence logic.

In the last decade, research on logics with team semantics outside of the first-order setting has thrived as well. A plethora of related systems has been introduced, most prominently for modal logic [Vä08], propositional logic [Yan14, YV16], and temporal logic [KMV15, KMVZ18]. Analogously to first-order team logics, variants with a Boolean negation were studied extensively [YV17, Mü14, KMSV15]. The atoms of dependency in these logics feature a fundamental difference to their first-order counterparts: First-order dependencies range over individuals of the universe, whereas propositional dependency atoms only range over truth values, of which there are only finitely many. Based on this fact, unlike in first-order logic, they can be finitely defined in terms of other logical connectives.

Gogic et al. [GKPS95] argue that in addition to the computational complexity of a logic and which properties it can express, it is also important to consider how *succinctly* the logic can express those properties. The succinctness of especially modal and temporal logics has been an active area of research for the last couple of decades; see e.g. [Wil99, LSW01, EVW02, AI03, Mar03] for earlier work on the topic and [FvdHIK11, FvdHIK13, vDFvdHI14, vdHI14] for recent work. A typical result states that a logic  $\mathcal{L}_1$  is exponentially more succinct than another logic  $\mathcal{L}_2$ . This means that there is a sequence of properties  $(P_n)_{n \in \mathbb{N}}$  such that  $P_n$  is definable by  $\mathcal{L}_1$ -formulas  $(\varphi_n)_{n \in \mathbb{N}}$ , but every family  $(\psi_n)_{n \in \mathbb{N}}$  of  $\mathcal{L}_2$ -formulas that defines  $(P_n)_{n \in \mathbb{N}}$  is exponentially larger than  $(\varphi_n)_{n \in \mathbb{N}}$ .

In team semantics, the question of succinctness has received only little attention so far. In their paper, Hella et al. [HLSV14] are primarily concerned with the expressive power of modal dependence logic, but they also show that defining the dependence atom in modal logic with Boolean disjunction requires a formula of exponential size. Similarly, Kontinen et al. [KMSV17] investigate many aspects of modal independence logic and among them show that modal independence logic is exponentially more succinct than basic modal logic. Our paper is, to our knowledge, the first systematic look at succinctness for team semantics.

The most commonly used systematic methods for proving succinctness results are formula size games and extended syntax trees. Formula size games are a variant of Ehrenfeucht-Fraïssé games made to correspond to the size of formulas instead of the usual depth of some operator. They were first introduced by Adler and Immerman [AI03] for branching-time temporal logic CTL. The method of extended syntax trees was originally formulated by Grohe and Schweikardt [GS05] for first-order logic. The notion of extended syntax tree was actually inspired by the Adler-Immerman game, and in a certain sense these two methods are equivalent: an extended syntax tree can be interpreted as a winning strategy for one of the players of the corresponding formula size game. Both of these methods have been adapted to many languages, especially in the modal setting, see e.g. [FvdHIK11, vdHIK12, vDFvdHI14].

The formula size game we define in this paper is an adaptation of the games defined by Hella and Väänänen for propositional and first-order logic [HV15] and later by Hella and Vilander for basic modal logic [HV16]. The new games of Hella and Väänänen are variations of the original Adler-Immerman game with a key difference. In the original game, the syntax tree of the formula in question is constructed in its entirety and consequently the second player has an easy optimal strategy. Thus the original game is in some sense a single player game. The new variant uses a predefined resource that bounds the size of the constructed formula and only one branch of the syntax tree is constructed in one play. The second player's decisions now truly matter as she gets to decide which branch that is.



Property		Connectives in $\Sigma$	Result
Dependence	$\sim=(\cdot;\cdot)$	$\wedge, \emptyset, *$	poly
	$=(\cdot;\cdot)$	$\wedge, \emptyset, *$	exp
	$\dot{=}(\cdot;\cdot)$	$\wedge, \sim, *$	poly
Independence	$\sim\perp_c$	$\wedge, \emptyset, \vee$	poly
	$\perp$	$\wedge, \emptyset, *$	exp
	$\perp_c$	$\wedge, \sim, *$	poly
Inclusion	$\sim\subseteq$	$\wedge, \emptyset, \vee$	poly
	$\subseteq$	$\wedge, \emptyset, *$	exp
	$\dot{\subseteq}$	$\wedge, \sim, *$	poly
Exclusion	$\sim $	$\wedge, \emptyset, *$	poly
	$ $	$\wedge, \emptyset, *$	exp
	$\dot{ }$	$\wedge, \sim, *$	poly
Anonymity	$\sim\Upsilon$	$\wedge, \emptyset, \vee$	poly
	$\Upsilon$	$\wedge, \emptyset, *$	exp
	$\dot{\Upsilon}$	$\wedge, \sim, *$	poly
Parity	$\sim\oplus$	$\wedge, \emptyset, *$	exp
	$\oplus$	$\wedge, \emptyset, *$	exp
	$\dot{\oplus}$	$\wedge, \sim, \dot{\vee}$	poly

TABLE 1. The succinctness of team properties in propositional team logic. “\*” means that the entry holds if  $\vee, \dot{\vee}$ , or both are available. The bounds are sharp in the following sense: “poly” means that there is a polynomial translation to  $\text{PL}(\Sigma)$ . “exp” means that there is an exponential translation to  $\text{PL}(\Sigma)$ , but no sub-exponential translation.

*Contribution.* In this paper we consider the succinctness of atoms of dependency. So far, it is known that these atoms can be expressed by exponentially large formulas (see Table 2), with only the dependence atom having a known polynomial size formula [HKVV18].

In Section 2 we define propositional team logic and the fragments we consider, and recall some useful known results. In Section 3 we obtain exponential lower bounds in the *existential fragment* of propositional team logic, where the splitting disjunction  $\vee$  may only occur positively. Our lower bounds imply succinctness results between logics with no atoms of dependency, and ones expanded with a single such atom. The lower bounds also show that the known translations to the existential fragment (see Table 2) are asymptotically optimal.

Most of the lower bounds are obtained via the new formula size game for propositional team logic, including a lower bound for the parity of the cardinality of teams. The lower bounds for dependence and exclusion atoms are obtained via the notion of *upper dimension*, adapted from [HLSV14].

In Section 4 we polynomially define the *negations* of the considered atoms of dependency in the existential fragment. From this, as a corollary we obtain polynomial upper bounds for full propositional team logic. Moreover, we define parity polynomially in the full logic, even though both even and odd parities have exponential lower bounds in the existential fragment. See Table 1 for an overview of all results. For each property, the three rows

correspond to defining the Boolean negation of the property with no free use of the Boolean negation operator  $\sim$ , defining the property itself in the same setting, and finally defining the property with free use of Boolean negation. The required formula is classified to be either polynomial or exponential with respect to the size of the corresponding atom. We always have the Boolean disjunction  $\otimes$  available and either the lax disjunction  $\vee$  or the strict disjunction  $\dot{\vee}$  or both.

Finally, we consider algorithmic applications of our results and show that the complexities of satisfiability, validity and model checking for propositional and modal team logic remain the same after extension by some atoms of dependency.

## 2. PRELIMINARIES

**Definition 2.1** (Teams). A *domain*  $\Phi$  is a finite set of atomic propositions. A  $\Phi$ -*assignment* is a function  $s: \Phi \rightarrow \{0, 1\}$ . A  $\Phi$ -*team*  $T$  is a (possibly empty) set of  $\Phi$ -functions,  $T \subseteq \Phi \rightarrow \{0, 1\}$ . The set of all  $\Phi$ -teams is denoted by  $\text{Tms}(\Phi)$ .

**Definition 2.2** (Splits). Let  $T$  be a team. We say that an ordered pair  $(T_1, T_2)$  of teams is a *split* of  $T$ , if  $T_1, T_2 \subseteq T$  and  $T_1 \cup T_2 = T$ . We say that a split  $(T_1, T_2)$  is *strict* if  $T_1 \cap T_2 = \emptyset$ . Otherwise it is *lax*. We denote the set of splits of  $T$  by  $\text{Sp}(T)$ , and the set of strict splits of  $T$  by  $\text{SSp}(T)$ .

**Definition 2.3** (PL( $\Sigma, \Phi$ )-formulas). Let  $\Sigma$  be a set of connectives  $\circ$  each with a designated arity  $\text{ar}(\circ) \geq 0$ . A  $\Phi$ -*literal* is a string of the form  $\top, \perp, \sim\top, \sim\perp, p, \neg p, \sim p$ , or  $\sim\neg p$ , where  $p \in \Phi$ . The set of PL( $\Sigma, \Phi$ )-formulas is then the smallest set containing all  $\Phi$ -literals and closed under connectives in  $\Sigma$ , i.e., if  $\varphi_1, \dots, \varphi_n \in \text{PL}(\Sigma, \Phi)$  and  $\text{ar}(\circ) = n$ , then  $\circ(\varphi_1, \dots, \varphi_n) \in \text{PL}(\Sigma, \Phi)$ .

Note that when we consider a logic with free usage of Boolean negation in front of arbitrary formulas, we include  $\sim$  in the set  $\Sigma$ . Otherwise, we always allow the Boolean negation  $\sim$  to occur in literals. In our setting the usual empty team property of every formula being true on the empty team, fails. We motivate this choice below after Proposition 2.13.

Let  $\text{Prop}(\varphi) \subseteq \Phi$  denote the set of propositional variables that occur in the formula  $\varphi$ . We will omit the domain  $\Phi$  if it is clear from the context or makes no difference, and write only PL( $\Sigma$ ). We consider the following connectives:

$$\begin{aligned}
T \vDash \top & \quad \text{always,} \\
T \vDash \perp & \quad \Leftrightarrow T = \emptyset \\
T \vDash p & \quad \Leftrightarrow \forall s \in T: s(p) = 1, \\
T \vDash \neg p & \quad \Leftrightarrow \forall s \in T: s(p) = 0, \\
T \vDash \sim\psi & \quad \Leftrightarrow T \not\vDash \psi, \\
T \vDash \psi \wedge \theta & \quad \Leftrightarrow T \vDash \psi \text{ and } T \vDash \theta, \\
T \vDash \psi \otimes \theta & \quad \Leftrightarrow T \vDash \psi \text{ or } T \vDash \theta, \\
T \vDash \psi \vee \theta & \quad \Leftrightarrow \exists (S, U) \in \text{Sp}(T): S \vDash \psi \text{ and } U \vDash \theta, \\
T \vDash \psi \dot{\vee} \theta & \quad \Leftrightarrow \exists (S, U) \in \text{SSp}(T): S \vDash \psi \text{ and } U \vDash \theta, \\
T \vDash \psi \otimes \theta & \quad \Leftrightarrow \forall (S, U) \in \text{Sp}(T): S \vDash \psi \text{ or } U \vDash \theta, \\
T \vDash \psi \dot{\otimes} \theta & \quad \Leftrightarrow \forall (S, U) \in \text{SSp}(T): S \vDash \psi \text{ or } U \vDash \theta,
\end{aligned}$$

Note that, as usually in the context of team logic, we have two different negations: a *dual negation*  $\neg$  and a *contradictory negation*  $\sim$ . For example, we have the equivalences  $\neg(p \vee q) \equiv \neg p \wedge \neg q$  and  $\sim(p \vee q) \equiv \sim p \otimes \sim q$ , but  $\neg p \wedge \neg q \not\equiv \sim p \otimes \sim q$ . Also note that in team logic we have four different logical constants, namely  $\top = \neg \perp$  (always true),  $\sim \top$  (always false),  $\perp = \neg \top$  (true in the empty team) and  $\sim \perp$  (true in non-empty teams).

We say  $\varphi$  *entails*  $\psi$ , in symbols  $\varphi \vDash \psi$ , if  $T \vDash \varphi$  implies  $T \vDash \psi$  for all domains  $\Phi \supseteq \text{Prop}(\varphi) \cup \text{Prop}(\psi)$  and  $\Phi$ -teams  $T$ . If  $\varphi \vDash \psi$  and  $\psi \vDash \varphi$ , then we write  $\varphi \equiv \psi$  and say that  $\varphi$  and  $\psi$  are *equivalent*.

**Definition 2.4.** A  $\text{PL}(\{\wedge, \vee\})$ -formula that contains no  $\sim$  is a *purely propositional* formula.

We will consistently use the letters  $\alpha, \beta, \gamma, \dots$  for purely propositional formulas, whereas  $\varphi, \psi, \theta, \dots$  will denote arbitrary formulas.

We define the shorthands  $\text{NE} := \sim \perp$ , which defines non-emptiness of teams, and  $\text{E}\alpha := \top \vee (\text{NE} \wedge \alpha)$ , which expresses that at least one assignment in the team satisfies the purely propositional formula  $\alpha$ .

Many formulas of team logic enjoy useful closure properties:

**Definition 2.5.** Let  $\varphi$  be a  $\text{PL}(\Sigma, \Phi)$ -formula.

- $\varphi$  is *union closed* if, for any set of  $\Phi$ -teams  $\mathcal{T}$  such that  $\forall T \in \mathcal{T} : T \vDash \varphi$  we have  $\bigcup \mathcal{T} \vDash \varphi$ .
- $\varphi$  is *downward closed* if, for any  $\Phi$ -teams  $T_1, T_2$ , if  $T_2 \vDash \varphi$  and  $T_1 \subseteq T_2$ , we have  $T_1 \vDash \varphi$ .
- $\varphi$  is *upward closed* if, for any  $\Phi$ -teams  $T_1, T_2$ , if  $T_2 \vDash \varphi$  and  $T_1 \supseteq T_2$ , we have  $T_1 \vDash \varphi$ .
- $\varphi$  has the *empty team property* if  $\emptyset \vDash \varphi$ .
- $\varphi$  is *flat* if, for any  $\Phi$ -team  $T$ ,  $T \vDash \varphi$  if and only if  $\{s\} \vDash \varphi$  for all  $s \in T$ .

A formula is flat if and only if it is union closed, downward closed, and has the empty team property.

**Proposition 2.6.** Let  $\varphi, \psi \in \text{PL}(\Sigma)$  such that at least one of  $\varphi$  and  $\psi$  is downward closed. Then  $\varphi \vee \psi \equiv \varphi \dot{\vee} \psi$ .

*Proof.* Obviously,  $\varphi \dot{\vee} \psi$  entails  $\varphi \vee \psi$ . Conversely, if  $T \vDash \varphi \vee \psi$  via some split  $(T_1, T_2)$  of  $T$ , then either  $T_1 \setminus T_2$  will still satisfy  $\varphi$  or  $T_2 \setminus T_1$  will satisfy  $\psi$ . So either  $(T_1 \setminus T_2, T_2)$  or  $(T_1, T_2 \setminus T_1)$  is a strict split of  $T$  witnessing  $\varphi \dot{\vee} \psi$ .  $\square$

**Proposition 2.7.** Every  $\sim$ -free  $\text{PL}(\{\wedge, \vee, \dot{\vee}\})$ -formula is flat. In particular, every purely propositional formula is flat.

*Proof.* An easy inductive proof.  $\square$

An important property of propositional (and other) logics is *locality*, which means that formulas depend only on the assignment to variables that actually occur in the formula. This property can be generalized to team semantics.

**Definition 2.8.** If  $T$  is a  $\Psi$ -team and  $\Phi \subseteq \Psi$ , the *projection of  $T$  onto  $\Phi$* , denoted  $T \upharpoonright \Phi$ , is defined as the  $\Phi$ -team  $\{s \upharpoonright \Phi \mid s \in T\}$ , where  $s \upharpoonright \Phi$  is the restriction of the function  $s$  to the domain  $\Phi$ .

**Definition 2.9.** A formula  $\varphi \in \text{PL}(\Sigma, \Phi)$  is *local* if, for any domain  $\Psi \supseteq \Phi$  and  $\Psi$ -team  $T$ , it holds  $T \vDash \varphi$  if and only if  $T \upharpoonright \Phi \vDash \varphi$ .

**Proposition 2.10** [YV17]. Every  $\text{PL}(\{\wedge, \sim, \vee\})$ -formula is local.

Note that locality quickly fails if we admit strict splitting  $\dot{\vee}$  (cf. [YV17]). The formula  $\psi := \sim p \dot{\vee} \sim p \dot{\vee} \sim p$  is an easy counter-example to the locality of  $\text{PL}(\{\dot{\vee}\})$ . No team with domain  $\{p\}$  does satisfy  $\psi$ , since it needs at least three assignments in the team, but for example the maximal  $\{p, q\}$ -team satisfies  $\psi$ .

**Definition 2.11** (Satisfiability). A formula  $\varphi$  is  $\Phi$ -satisfiable if  $T \models \varphi$  for at least one  $\Phi$ -team  $T$ .

The domain is crucial here: The previous example formula  $\psi$  is  $\{p, q\}$ -satisfiable, but not  $\{p\}$ -satisfiable.

Often the empty team is excluded in the definition of satisfiability, especially in logics with the empty team property where otherwise every formula would be satisfiable. This is not necessary here as these definitions are interchangeable;  $\varphi$  is satisfiable in a non-empty team iff  $\varphi \wedge \text{NE}$  is satisfiable, and  $\varphi$  is satisfiable iff  $\top \vee \varphi$  is satisfiable in a non-empty team.

Usually, for propositional team logic,  $\otimes$  and  $\odot$  are omitted since they are definable as  $\varphi \otimes \psi \equiv \sim(\sim\varphi \vee \sim\psi)$ , and  $\varphi \odot \psi \equiv \sim(\sim\varphi \dot{\vee} \sim\psi)$ . If they are removed entirely, then the splitting disjunction may occur only positively, that is, splits of team may only be quantified existentially. This fragment plays an important role in the paper.

**Definition 2.12.** The *existential fragment* is  $\text{PL}(\{\wedge, \otimes, \vee, \dot{\vee}\})$ .

It is well known that team logic is inherently second-order in nature: First-order dependence logic is actually equivalent to existential second-order logic [Vä07], and equivalent to full second-order logic if arbitrary negation is added [KN09]. In the same vein, propositional team logic is equivalent to second-order logic over  $\{0, 1\}$ , and to existential second-order logic if  $\sim$  is restricted [HKLV16]. In all these results, the splitting disjunction  $\vee$  simulates set quantification. From this perspective, we call the fragment with only positive  $\vee$  “existential”.

Note that, unlike in the first-order setting, for propositional logics the difference between existential and full logic emerges only in succinctness, not in expressive power. Indeed Yang and Väänänen [YV17] showed that already the existential fragment is expressively complete:

**Proposition 2.13.** *For every set  $P$  of  $\Phi$ -teams there is a formula  $\varphi$  in the existential fragment such that  $T \in P \Leftrightarrow T \models \varphi$  for all  $\Phi$ -teams  $T$ . In particular, for every  $\Sigma$  and formula  $\psi \in \text{PL}(\Sigma, \Phi)$  there is a formula  $\varphi$  of the existential fragment such that  $\psi \equiv \varphi$ .*

Essentially this is the reason we keep Boolean negation in literals. While expressively complete, the fragment lacks the succinctness of full propositional team logic with free use of Boolean negation. For this reason, we find the existential fragment to be a suitable logic to compare in terms of succinctness to full propositional team logic.

We proceed with the definition of the *size* of a formula. The literature contains many different accounts of what should be considered formula size. We take as our basic concept the length of the formula as a string. Since in team semantics the domain is often fixed and finite, we consider each proposition symbol to be only one symbol in the string. In Section 3 we define another measure of formula size called width because it is more convenient for the formula size game. Since we only use width for lower bounds and length is always greater than width, we refer to length in the theorems for the lower bounds.

**Definition 2.14.** The *length* of a formula  $\varphi \in \text{PL}(\Sigma)$ , denoted by  $|\varphi|$ , is the length of  $\varphi$  as a string, counting proposition symbols as one symbol.

If  $\alpha$  is a purely propositional formula and not an atomic proposition, then technically  $\neg\alpha$  is not a formula; then by  $\neg\alpha$  we refer to the formula that is obtained from  $\alpha$  by pushing

$\neg$  inwards using classical laws, i.e.,  $\neg(\beta \wedge \gamma) := (\neg\beta \vee \neg\gamma)$  and  $\neg(\beta \vee \gamma) := (\neg\beta \wedge \neg\gamma)$ . For tuples  $\vec{\alpha} = (\alpha_1, \dots, \alpha_n)$  and  $\vec{\beta} = (\beta_1, \dots, \beta_n)$  of purely propositional formulas, we write  $\vec{\alpha} \leftrightarrow \vec{\beta}$  for the formula  $\bigwedge_{i=1}^n ((\alpha_i \wedge \beta_i) \vee (\neg\alpha_i \wedge \neg\beta_i))$  and  $\vec{\alpha} \leftrightarrow\!\!\!\leftrightarrow \vec{\beta}$  for  $\neg(\vec{\alpha} \leftrightarrow \vec{\beta})$ . Note that since the formula  $\vec{\alpha} \leftrightarrow \vec{\beta}$  is purely propositional, we may use the dual negation  $\neg$  here.

By slight abuse of notation, we will write  $s(\alpha)$  even if  $\alpha$  is not an atomic proposition, and mean

$$s(\alpha) = \begin{cases} 1 & \text{if } \{s\} \models \alpha, \\ 0 & \text{else.} \end{cases}$$

Finally,  $s(\vec{\alpha})$  is short for the vector  $(s(\alpha_1), \dots, s(\alpha_n)) \in \{0, 1\}^n$ .

We consider the following atoms of dependency, where  $\vec{\alpha}$ ,  $\vec{\beta}$ ,  $\vec{\gamma}$  are (possibly empty) tuples of formulas:

**Dependence:**  $=(\vec{\alpha}; \vec{\beta})$ :

$$T \models =(\vec{\alpha}; \vec{\beta}) \iff \forall s, s' \in T : s(\vec{\alpha}) = s'(\vec{\alpha}) \Rightarrow s(\vec{\beta}) = s'(\vec{\beta})$$

**Independence:**  $\vec{\alpha} \perp \vec{\beta}$ :

$$T \models \vec{\alpha} \perp \vec{\beta} \iff \forall s, s' \in T : \exists s'' \in T : s(\vec{\alpha}) = s''(\vec{\alpha}) \text{ and } s'(\vec{\beta}) = s''(\vec{\beta})$$

**Conditional independence:**  $\vec{\alpha} \perp_{\vec{\beta}} \vec{\gamma}$ :

$$T \models \vec{\alpha} \perp_{\vec{\beta}} \vec{\gamma} \iff \forall s, s' \in T : \text{if } s(\vec{\beta}) = s'(\vec{\beta}) \text{ then} \\ \exists s'' \in T : s(\vec{\alpha}\vec{\beta}) = s''(\vec{\alpha}\vec{\beta}) \text{ and } s'(\vec{\gamma}) = s''(\vec{\gamma})$$

**Inclusion:**  $\vec{\alpha} \subseteq \vec{\beta}$ , where  $\vec{\alpha}$  and  $\vec{\beta}$  have equal length:

$$T \models \vec{\alpha} \subseteq \vec{\beta} \iff \forall s \in T \exists s' \in T : s(\vec{\alpha}) = s'(\vec{\beta})$$

**Exclusion:**  $\vec{\alpha} \mid \vec{\beta}$ , where  $\vec{\alpha}$  and  $\vec{\beta}$  have equal length:

$$T \models \vec{\alpha} \mid \vec{\beta} \iff \forall s \in T \forall s' \in T : s(\vec{\alpha}) \neq s'(\vec{\beta})$$

**Anonymity:**  $\vec{\alpha} \Upsilon \vec{\beta}$ :

$$T \models \vec{\alpha} \Upsilon \vec{\beta} \iff \forall s \in T \exists s' \in T : s(\vec{\alpha}) = s'(\vec{\alpha}) \text{ and } s(\vec{\beta}) \neq s'(\vec{\beta})$$

Originally, the dependence and independence atoms were introduced in the first-order setting by Väänänen [Vä07] and Grädel and Väänänen [GV13]. Inclusion and exclusion were considered by Galliani [Gal12]. The anonymity atom is due to Väänänen [Vä19]. The propositional counterparts we study here, except for the anonymity atom, were first studied by Yang [Yan14].

**Proposition 2.15.** *Let  $\Sigma = \{\wedge, \oplus, \vee\}$  or  $\Sigma = \{\wedge, \oplus, \dot{\vee}\}$ . The atoms of dependence, conditional independence, inclusion, exclusion and anonymity are expressible by  $\text{PL}(\Sigma)$ -formulas of size  $2^{\mathcal{O}(n)}$ .*

*Proof.* See Table 2 for  $\text{PL}(\{\wedge, \oplus, \vee\})$ -formulas defining each atom. We prove the case of the inclusion atom and leave the rest to the reader.

Let  $\varphi$  be the defining formula of Table 2 for the inclusion atom and let  $\vec{\alpha} = (\alpha_1, \dots, \alpha_n)$  and  $\vec{\beta} = (\beta_1, \dots, \beta_n)$  be tuples of purely propositional formulas. Assume  $T \models \vec{\alpha} \subseteq \vec{\beta}$  and let  $\vec{c} \in \{\top, \perp\}^n$ . If there is an assignment  $t \in T$  such that  $t(\vec{\alpha}) = t(\vec{c})$ , then by the inclusion

---

$\models(\vec{\alpha}; \vec{\beta})$	$\equiv \bigvee_{\vec{c} \in \{\top, \perp\}^n} ((\vec{\alpha} \leftrightarrow \vec{c}) \wedge \bigwedge_{i=1}^m (\beta_i \otimes \neg\beta_i))$
$\vec{\alpha} \perp_{\vec{\gamma}} \vec{\beta}$	$\equiv \bigvee_{\vec{c} \in \{\top, \perp\}^k} ((\vec{\gamma} \leftrightarrow \vec{c}) \wedge (\vec{\alpha} \perp \vec{\beta}))$
$\vec{\alpha} \perp \vec{\beta}$	$\equiv \bigwedge_{\substack{\vec{c} \in \{\top, \perp\}^n \\ \vec{c}' \in \{\top, \perp\}^m}} (\vec{\alpha} \leftrightarrow \vec{c}) \otimes (\vec{\beta} \leftrightarrow \vec{c}') \otimes \mathbf{E}((\vec{\alpha} \leftrightarrow \vec{c}) \wedge (\vec{\beta} \leftrightarrow \vec{c}'))$
$\vec{\alpha} \subseteq \vec{\beta}$	$\equiv \bigwedge_{\vec{c} \in \{\top, \perp\}^n} (\vec{\alpha} \leftrightarrow \vec{c}) \otimes \mathbf{E}(\vec{\beta} \leftrightarrow \vec{c})$
$\vec{\alpha} \mid \vec{\beta}$	$\equiv \bigwedge_{\vec{c} \in \{\top, \perp\}^n} (\vec{\alpha} \leftrightarrow \vec{c}) \otimes (\vec{\beta} \leftrightarrow \vec{c})$
$\vec{\alpha} \Upsilon \vec{\beta}$	$\equiv \bigvee_{\vec{c} \in \{\top, \perp\}^n} (\vec{\alpha} \leftrightarrow \vec{c} \wedge \bigotimes_{i=1}^m (\mathbf{E}\beta_i \wedge \mathbf{E}\neg\beta_i))$

---

TABLE 2. Exponential translations of atoms in the existential fragment, where  $\vec{\alpha} = (\alpha_1, \dots, \alpha_n)$ ,  $\vec{\beta} = (\beta_1, \dots, \beta_m)$  (with  $n = m$  for  $\subseteq$  and  $\mid$ ) and  $\vec{\gamma} = (\gamma_1, \dots, \gamma_k)$ .

atom there is another assignment  $t' \in T$  such that  $t(\vec{c}) = t'(\vec{c}) = t'(\vec{\beta})$ . Thus  $\mathbf{E}(\vec{\beta} = \vec{c})$  holds. If there is no such assignment  $t$ , then  $T \models \vec{\alpha} \neq \vec{c}$  holds. For every  $\vec{c}$  the Boolean disjunction  $(\vec{\alpha} \neq \vec{c}) \otimes \mathbf{E}(\vec{\beta} = \vec{c})$  holds, so  $T \models \varphi$ .

Conversely, assume  $T \models \varphi$ . Let  $t \in T$  be an assignment. Let  $t(\vec{\alpha}) = \vec{b} \in \{0, 1\}^n$ , and let  $\vec{s} \in \{\top, \perp\}^n$  such that  $t(\vec{s}) = \vec{b}$ . Now clearly  $\vec{\alpha} \neq \vec{s}$  does not hold so  $\mathbf{E}(\vec{\beta} = \vec{s})$  holds. Consequently, there is an assignment  $t' \in T$  such that  $s'(\beta) = \vec{b} = s(\vec{\alpha})$ , so  $T$  satisfies the inclusion atom.

For  $\text{PL}(\{\wedge, \otimes, \check{\vee}\})$ , it is easy to check that replacing each occurrence of  $\vee$  with  $\check{\vee}$  leads to an equivalent formula.  $\square$

### 3. EXPONENTIAL LOWER BOUNDS FOR TEAM PROPERTIES

Though the length of a formula is the most immediate measure of formula size, it is not the most practical one in terms of defining a formula size game. For a measure better suited to the game we have chosen the number of literals in a formula, which we call width. As the name suggests, width corresponds to the number of leaves in the syntax tree of the formula.

**Definition 3.1.** The *width* of a formula  $\varphi \in \text{PL}(\Sigma)$ , denoted by  $\text{wd}(\varphi)$ , is defined recursively as follows:

- $\text{wd}(l) = 1$  for a literal  $l$ ,
- $\text{wd}(\psi \circ \theta) = \text{wd}(\psi) + \text{wd}(\theta)$ , where  $\circ \in \Sigma$  is binary,
- $\text{wd}(\circ\psi) = \text{wd}(\psi)$ , where  $\circ \in \Sigma$  is unary.

For the actual upper and lower bounds we prove, the difference between length and width is inconsequential. The number of binary connectives, and therefore parentheses, depends on

the number of literals and the number of negations of either kind for a minimal formula is bounded by the number of literals. Note that for the game we also assume formulas to be in negation normal form, but this doesn't affect the width of formulas.

**3.1. A formula size game for team semantics.** Let  $\mathbb{A}_0$  and  $\mathbb{B}_0$  be sets of  $\Phi$ -teams and let  $k_0$  be a natural number. Let  $\Sigma \subseteq \{\emptyset, \wedge, \vee, \oplus, \dot{\vee}, \dot{\oplus}\}$  be a set of connectives. Note that if the strong negation  $\sim$  is freely available in the fragment under consideration, then either none or both of each pair of dual operators must be included in  $\Sigma$ .

The formula size game  $\text{FS}_{k_0}^\Sigma(\mathbb{A}_0, \mathbb{B}_0)$  for  $\text{PL}(\Sigma)$  has two players, S (Samson) and D (Delilah). Positions of the game are of the form  $(k, \mathbb{A}, \mathbb{B})$ , where  $\mathbb{A}$  and  $\mathbb{B}$  are sets of teams and  $k$  is a natural number.

The goal of S is to construct a formula  $\varphi$  that *separates*  $\mathbb{A}$  from  $\mathbb{B}$ , which means that  $T \models \varphi$  for every team  $T \in \mathbb{A}$ , denoted  $\mathbb{A} \models \varphi$  and  $T \not\models \varphi$  for every team  $T \in \mathbb{B}$ , denoted  $\mathbb{B} \not\models \varphi$ . Note that  $\mathbb{B} \models \sim\varphi$  is different from  $\mathbb{B} \not\models \varphi$  since the first states that no team in  $\mathbb{B}$  satisfies  $\varphi$  and the second only that not all teams in  $\mathbb{B}$  satisfy  $\varphi$ .

The starting position is  $(k_0, \mathbb{A}_0, \mathbb{B}_0)$ . If  $k_0 = 0$ , D wins the game. In a position  $(k, \mathbb{A}, \mathbb{B})$  with  $k \geq 1$ , S must make one of  $|\Sigma| + 1$  moves to continue the game. The available moves are the ones given by  $\Sigma$  and the literal move. The moves work as follows:

- $\oplus$ -move: S chooses subsets  $\mathbb{A}_1, \mathbb{A}_2 \subseteq \mathbb{A}$  such that  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$  and natural numbers  $k_1, k_2 > 0$  such that  $k_1 + k_2 = k$ . Then D chooses  $i \in \{1, 2\}$ . The game continues from the position  $(k_i, \mathbb{A}_i, \mathbb{B})$ .
- $\wedge$ -move: Same as the  $\oplus$ -move with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.
- $\vee$ -move: For every team  $A \in \mathbb{A}$ , S chooses a split  $(A_1, A_2)$ . Let  $\mathbb{A}_i = \{A_i \mid A \in \mathbb{A}\}$  for  $i \in \{1, 2\}$ . For every team  $B \in \mathbb{B}$ , S chooses a function  $f_B : \text{Sp}(B) \rightarrow \{1, 2\}$ . Let  $\mathbb{B}_i = \{B_i \mid f_B(B_1, B_2) = i, (B_1, B_2) \in \text{Sp}(B), B \in \mathbb{B}\}$  for  $i \in \{1, 2\}$ . Finally, S chooses natural numbers  $k_1, k_2 > 0$  such that  $k_1 + k_2 = k$ . Then D chooses a number  $i \in \{1, 2\}$ . The game continues from the position  $(k_i, \mathbb{A}_i, \mathbb{B}_i)$ .
- $\oplus$ -move: Same as the  $\vee$ -move with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.
- $\dot{\vee}$ -move: Same as the  $\vee$ -move except all splits  $(A_1, A_2)$  and  $(B_1, B_2)$  considered are strict.
- $\dot{\oplus}$ -move: Same as the  $\dot{\vee}$ -move with the roles of  $\mathbb{A}$  and  $\mathbb{B}$  switched.
- Literal move: S chooses a  $\Phi$ -literal  $l$ . If  $l$  separates  $\mathbb{A}$  from  $\mathbb{B}$ , S wins. Otherwise, D wins.

While the definition of the  $\vee$ -move is quite technical, the intuition is well grounded in the semantics of the connective  $\vee$ . Let us assume S has a formula in mind with  $\vee$  as the outermost connective. On the  $\mathbb{A}$ -side S simply splits each team  $A$  into two teams,  $A_1$  and  $A_2$ , such that  $A_1$  satisfies the left disjunct and  $A_2$  satisfies the right one. The  $\mathbb{B}$ -side is more involved. S claims that no team in  $\mathbb{B}$  satisfies the disjunction so for each team  $B$  and each split of that team,  $(B_1, B_2)$ , he must choose which  $B_i$  does not satisfy the corresponding disjunct. These choices are gathered in the function  $f_B$  for each team. Finally  $\mathbb{B}_1$  gathers all of the teams S has claimed to not satisfy the first disjunct, and the same for  $\mathbb{B}_2$  and the second disjunct.

The number  $k$  can be considered a resource for S in the following sense. Since for all the connective moves  $k_1, k_2 > 0$ , the number  $k$  decreases in each move, and if  $k = 1$ , only the literal move is available. Thus, in a finite number of moves, S expends his resource  $k$  and must eventually make a literal move which will end the game and one of the players will win.

We first prove that winning strategies for the formula size game  $\text{FS}_{k_0}^\Sigma(\mathbb{A}_0, \mathbb{B}_0)$  correspond to  $\text{PL}(\Sigma)$ -formulas of size at most  $k_0$  that separate  $\mathbb{A}_0$  from  $\mathbb{B}_0$ .

**Theorem 3.2.** *Let  $\mathbb{A}_0$  and  $\mathbb{B}_0$  be sets of teams and let  $k_0 \in \mathbb{N}$ . Then the following conditions are equivalent:*

- (1) $_{k_0}$  *S has a winning strategy for the game  $\text{FS}_{k_0}^\Sigma(\mathbb{A}_0, \mathbb{B}_0)$ .*
- (2) $_{k_0}$  *There is a formula  $\varphi \in \text{PL}(\Sigma)$  with  $\text{wd}(\varphi) \leq k_0$  which separates  $\mathbb{A}_0$  from  $\mathbb{B}_0$ .*

*Proof.* We prove the equivalence of (1) $_{k_0}$  and (2) $_{k_0}$  by induction on  $k_0$ .

Let  $k_0 = 1$ . The only type of move available for S is the literal move, so S if has a winning strategy, then there is a literal that separates  $\mathbb{A}_0$  from  $\mathbb{B}_0$ . Conversely, the only formulas with size at most 1 are literals so if such a formula exists, then S wins by choosing that formula for a literal move.

Let  $k_0 > 1$  and assume that the equivalence of (1) $_k$  and (2) $_k$  holds for all natural numbers  $k < k_0$  and all sets of teams  $\mathbb{A}$  and  $\mathbb{B}$ .

(1) $_{k_0} \Rightarrow$  (2) $_{k_0}$ : Let  $\delta$  be a winning strategy of S for the game  $\text{FS}_{k_0}^\Sigma(\mathbb{A}_0, \mathbb{B}_0)$ . We divide the proof into cases according to the first move of  $\delta$ . We handle all operators possibly in  $\Sigma$  except for dual cases.

- **Literal move:** Since S is playing according to the winning strategy  $\delta$ , the literal  $l$  chosen by S separates  $\mathbb{A}_0$  from  $\mathbb{B}_0$ . In addition,  $\text{wd}(l) = 1 \leq k_0$ .
- **$\otimes$ -move:** Let  $(k_1, \mathbb{A}_1, \mathbb{B}_0)$  and  $(k_2, \mathbb{A}_2, \mathbb{B}_0)$  be the successor positions chosen by S according to  $\delta$ . Since  $\delta$  is a winning strategy, S has a winning strategy for both games  $\text{FS}_{k_i}^\Sigma(\mathbb{A}_i, \mathbb{B}_0)$ . By induction hypothesis, there are formulas  $\psi_i$  with  $\text{wd}(\psi_i) \leq k_i$  that separate  $\mathbb{A}_i$  from  $\mathbb{B}_0$ . Let  $\varphi = \psi_1 \otimes \psi_2$ . We have  $\mathbb{A}_0 = \mathbb{A}_1 \cup \mathbb{A}_2$  so  $\mathbb{A}_0 \models \varphi$ . On the other side we have  $\mathbb{B}_0 \models \sim \psi_1$  and  $\mathbb{B}_0 \models \sim \psi_2$  so  $\mathbb{B}_0 \not\models \varphi$ . Finally  $\text{wd}(\varphi) = \text{wd}(\psi_1) + \text{wd}(\psi_2) \leq k_1 + k_2 = k_0$ .
- **$\vee$ -move:** Let  $(k_1, \mathbb{A}_1, \mathbb{B}_1)$  and  $(k_2, \mathbb{A}_2, \mathbb{B}_2)$  be the successor positions chosen by S according to  $\delta$ . Again by induction hypothesis there are formulas  $\psi_i$  with  $\text{wd}(\psi_i) \leq k_i$  which separate  $\mathbb{A}_i$  from  $\mathbb{B}_i$ . Let  $\varphi = \psi_1 \vee \psi_2$ . For each  $A \in \mathbb{A}_0$  S chose a split  $(A_1, A_2)$ . Now  $A_1 \models \psi_1$  and  $A_2 \models \psi_2$  so  $A \models \varphi$ . On the other side, for each  $B \in \mathbb{B}_0$ , S chose a function  $f_B : \text{Sp}(B) \rightarrow \{1, 2\}$ . For each  $(B_1, B_2) \in \text{Sp}(B)$ , if  $f_B(B_1, B_2) = i$ , then  $B_i \not\models \psi_i$ . Thus  $B \not\models \varphi$ . The width of  $\varphi$  is as in the previous case.
- **$\dot{\vee}$ -move:** Same as the  $\vee$ -move except all splits considered are strict.

(2) $_{k_0} \Rightarrow$  (1) $_{k_0}$ : Let  $\varphi \in \text{PL}(\Sigma)$  with  $\text{wd}(\varphi) \leq k_0$  which separates  $\mathbb{A}_0$  from  $\mathbb{B}_0$ . We give the first move of the winning strategy of S for the game  $\text{FS}_{k_0}^\Sigma(\mathbb{A}_0, \mathbb{B}_0)$ . Then the following position  $(k, \mathbb{A}, \mathbb{B})$  is a valid starting position for a game  $\text{FS}_k^\Sigma(\mathbb{A}, \mathbb{B})$ . We can obtain a winning strategy for S in this new game using the induction hypothesis. We finally obtain the full winning strategy for S by combining the first move described below to the strategy given by the induction hypothesis. We divide the proof into cases according to the outermost connective of  $\varphi$ . We again handle only one of each pair of dual cases.

- $\varphi$  is a literal: We know that  $\varphi$  separates  $\mathbb{A}_0$  from  $\mathbb{B}_0$  so S wins by making a literal move choosing  $\varphi$ .
- $\varphi = \psi_1 \otimes \psi_2$ : S chooses  $\mathbb{A}_i = \{A \in \mathbb{A}_0 \mid A \models \psi_i\}$  for  $i \in \{1, 2\}$ ,  $k_1 = \text{wd}(\psi_1)$  and  $k_2 = k - k_1$ . Since  $\varphi$  separates  $\mathbb{A}_0$  from  $\mathbb{B}_0$ , we have  $\mathbb{A}_0 \models \varphi$  so  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$ . On the other side,  $\mathbb{B}_0 \models \sim \varphi$  so  $B \not\models \psi_1$  and  $B \not\models \psi_2$  for every  $B \in \mathbb{B}_0$ . Now, no matter which number  $i \in \{1, 2\}$  D chooses, in the following position  $(k_i, \mathbb{A}_i, \mathbb{B}_0)$ , the formula  $\psi_i$  will separate  $\mathbb{A}_i$  from  $\mathbb{B}_0$ . In addition,  $k_1 \leq \text{wd}(\psi_1)$  and  $k_2 = k_0 - k_1 \leq \text{wd}(\varphi) - \text{wd}(\psi_1) = \text{wd}(\psi_2)$ . By induction hypothesis S has a winning strategy for both games  $\text{FS}_{k_i}^\Sigma(\mathbb{A}_i, \mathbb{B}_0)$ .
- $\varphi = \psi_1 \vee \psi_2$ : Again we have  $\mathbb{A}_0 \models \varphi$  so for every  $A \in \mathbb{A}_0$ , there is a split  $(A_1, A_2)$  such that  $A_1 \models \psi_1$  and  $A_2 \models \psi_2$ . S chooses such a split for every  $A \in \mathbb{A}_0$ . On the other side,



$\mathbb{B}_0 \models \sim\varphi$  so for every  $B \in \mathbb{B}_0$  and every split  $(B_1, B_2)$  we have  $B_1 \not\models \psi_1$  or  $B_2 \not\models \psi_2$ . For each  $B \in \mathbb{B}_0$ , S chooses  $f_B$  so that if  $f_B(B_1, B_2) = i$ , then  $B_i \not\models \psi_i$ . Now, no matter which number  $i \in \{1, 2\}$  D chooses, in the following position  $(k_i, \mathbb{A}_i, \mathbb{B}_i)$ , the formula  $\psi_i$  will separate  $\mathbb{A}_i$  from  $\mathbb{B}_i$ . S deals with the resource  $k_0$  just like in the previous case. By induction hypothesis S has a winning strategy for both games  $\text{FS}_{k_i}^\Sigma(\mathbb{A}_i, \mathbb{B}_i)$ .

- $\varphi = \psi_1 \dot{\vee} \psi_2$ : Same as the  $\vee$ -case except all splits considered are strict.  $\square$

Before we move on to the lower bounds, we prove a very standard lemma for formula size games stating that if at any time the same team ends up on both sides of the game, D wins.

**Lemma 3.3.** *If in a position  $P = (k, \mathbb{A}, \mathbb{B})$  there is a team  $T \in \mathbb{A} \cap \mathbb{B}$ , D has a winning strategy from position  $P$ .*

*Proof.* As long as there is  $T \in \mathbb{A} \cap \mathbb{B}$ , if S makes a literal move, D wins. We show that D can maintain this condition. We again omit the cases of dual operators.

- $\odot$ -move: S chooses sets  $\mathbb{A}_1, \mathbb{A}_2 \subseteq \mathbb{A}$ . Since  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$ , we have  $T \in \mathbb{A}_i$  for some  $i \in \{1, 2\}$ . Then D chooses the following position  $(k_i, \mathbb{A}_i, \mathbb{B})$  and we have  $T \in \mathbb{A}_i \cap \mathbb{B}$ .
- $\vee$ -move: Let  $(T_1, T_2)$  be the split S chooses for  $T$  on the left side. On the right side S must choose  $i = f_T(T_1, T_2) \in \{1, 2\}$ . Then D chooses the following position  $(k_i, \mathbb{A}_i, \mathbb{B}_i)$  and we have  $T_i \in \mathbb{A}_i \cap \mathbb{B}_i$ .
- $\dot{\vee}$ -move: Same as the  $\vee$ -move except the split must be strict.

Since S must eventually make a literal move, D wins the game.  $\square$

**3.2. Lower bounds via the formula size game.** In this section we use the formula size game to show lower bounds for the lengths of formulas defining atoms of dependency in the positive fragment of propositional team logic. We first state all of the bounds as a theorem and prove them in the rest of the section.

For natural numbers  $k$  and  $m$ ,  $[k]_m$  is the remainder of  $k$  modulo  $m$ .

**Theorem 3.4.** *Let  $\Sigma = \{\odot, \wedge, \vee, \dot{\vee}\}$ ,  $n, m \geq 1$ , and  $\Phi_n = \{p_1, \dots, p_n\}$ .*

- (1) *If  $m \leq 2^n$  and  $k < m$ , then a  $\text{PL}(\Sigma)$ -formula, that defines the property  $|T| \equiv k \pmod{m}$  of  $\Phi_n$ -teams  $T$ , has length at least  $2^n - [2^n - k]_m$ . In particular, a formula that defines even parity has length at least  $2^n$ .*
- (2) *A  $\text{PL}(\Sigma)$ -formula that defines cardinality  $k \leq 2^n$  of  $\Phi_n$ -teams has length at least  $k$ .*
- (3) *A  $\text{PL}(\Sigma)$ -formula that defines  $p_1 \cdots p_n \subseteq q_1 \cdots q_n$  has length at least  $2^n$ .*
- (4) *A  $\text{PL}(\Sigma)$ -formula that defines  $p_1 \cdots p_n \perp q_1 \cdots q_m$  has length at least  $2^{n+m}$ .*
- (5) *A  $\text{PL}(\Sigma)$ -formula that defines  $p_1 \cdots p_n \Upsilon q$  has length at least  $2^{n+1}$ .*

Note that for  $\Upsilon$  we only consider a single argument on the right-hand side. While this is an exponential lower bound (in  $n$ ), a tight bound in both  $n$  and  $m$  (cf. Table 2) is still open.

Our approach to proving these bounds is similar to that of Hella and Väänänen in [HV15]. They used a formula size game for propositional logic to show that defining the parity of the number of ones in a propositional assignment of length  $n$  requires a formula of length  $n^2$ . We focus on teams that differ only by one assignment and define a measure named *density* as in [HV15], although our definition is slightly different.

**Definition 3.5.** Let  $T$  be a team. A team  $T'$  is a *neighbour* of  $T$ , if  $T' = T \setminus \{s\}$  for some assignment  $s \in T$ .

Let  $\mathbb{A}$  be a set of teams. The number of neighbours of  $T$  in the set  $\mathbb{A}$  is denoted by  $N(T, \mathbb{A})$ ,

$$N(T, \mathbb{A}) := |\{A \in \mathbb{A} \mid A \text{ is a neighbour of } T\}|.$$

The *density* of the pair  $(\mathbb{A}, \mathbb{B})$  is

$$D(\mathbb{A}, \mathbb{B}) := \max\{N(A, \mathbb{B}) \mid A \in \mathbb{A}\}.$$

We shall use density as an invariant for the formula size game. Essentially we will show that a certain number of the resource  $k$  must be expended before a literal move can be made. First we show that  $S$  cannot make a successful literal move when density is too high.

**Lemma 3.6.** *If  $D(\mathbb{A}, \mathbb{B}) > 1$ , then no literal separates  $\mathbb{A}$  from  $\mathbb{B}$ .*

*Proof.* If  $D(\mathbb{A}, \mathbb{B}) > 1$ , at least one team  $A \in \mathbb{A}$  has two neighbours  $B_1, B_2 \in \mathbb{B}$ . Now any positive literal  $l$  (with respect to  $\sim$ ) true in  $A$  is also true in  $B_1$  and  $B_2$  since they are subteams of  $A$ . On the other hand, since  $B_1$  and  $B_2$  are different neighbours of  $A$ , we have  $B_1 \cup B_2 = A$ . For a negative literal  $\sim l$ , assume that  $B_1 \not\models \sim l$  and  $B_2 \not\models \sim l$ . This means that  $B_1 \models l$  and  $B_2 \models l$ , so by union closure,  $A \models l$  and consequently  $A \not\models \sim l$ .  $\square$

We proceed to show that density behaves well with respect to the moves of the game.

**Lemma 3.7.** *Let  $(k, \mathbb{A}, \mathbb{B})$  be a position in a game  $\text{FS}_{k_0}^\Sigma(\mathbb{A}_0, \mathbb{B}_0)$ .*

- (1) *If  $S$  makes a  $\odot$ -move, and the possible following positions are  $(k_1, \mathbb{A}_1, \mathbb{B})$  and  $(k_2, \mathbb{A}_2, \mathbb{B})$ , then  $D(\mathbb{A}_1, \mathbb{B}) + D(\mathbb{A}_2, \mathbb{B}) \geq D(\mathbb{A}, \mathbb{B})$ .*
- (2) *If  $S$  makes a  $\wedge$ -move, and the possible following positions are  $(k_1, \mathbb{A}, \mathbb{B}_1)$  and  $(k_2, \mathbb{A}, \mathbb{B}_2)$ , then  $D(\mathbb{A}, \mathbb{B}_1) + D(\mathbb{A}, \mathbb{B}_2) \geq D(\mathbb{A}, \mathbb{B})$ .*
- (3) *If  $S$  makes a  $\vee$ -move or  $\check{\vee}$ -move, and the possible following positions are  $(k_1, \mathbb{A}_1, \mathbb{B}_1)$  and  $(k_2, \mathbb{A}_2, \mathbb{B}_2)$ , then  $D(\mathbb{A}_1, \mathbb{B}_1) + D(\mathbb{A}_2, \mathbb{B}_2) \geq D(\mathbb{A}, \mathbb{B})$  or  $D$  has a winning strategy from one of the following positions.*

*Proof.* Let  $A$  be one of the teams in  $\mathbb{A}$  with most neighbours in  $\mathbb{B}$ .

- (1) Since  $\mathbb{A}_1 \cup \mathbb{A}_2 = \mathbb{A}$ , we may assume by symmetry that  $A \in \mathbb{A}_1$ . Since all the same neighbours of  $A$  are still in  $\mathbb{B}$ , we get  $D(\mathbb{A}_1, \mathbb{B}) + D(\mathbb{A}_2, \mathbb{B}) \geq D(\mathbb{A}_1, \mathbb{B}) \geq D(\mathbb{A}, \mathbb{B})$ .
- (2) Since  $\mathbb{B}_1 \cup \mathbb{B}_2 = \mathbb{B}$ , the neighbours of  $A$  are split between  $\mathbb{B}_1$  and  $\mathbb{B}_2$  so  $D(\mathbb{A}, \mathbb{B}_1) + D(\mathbb{A}, \mathbb{B}_2) \geq N(A, \mathbb{B}_1) + N(A, \mathbb{B}_2) \geq N(A, \mathbb{B}) = D(\mathbb{A}, \mathbb{B})$ .
- (3) Let  $(A_1, A_2)$  be the (strict) split of  $A$  chosen by  $S$ . Suppose  $B = A \setminus \{a\}$  is a neighbour of  $A$  in  $\mathbb{B}$ . Then  $(B_1, B_2) := (A_1 \setminus \{a\}, A_2 \setminus \{a\})$  is a split of  $B$ , and is strict if  $(A_1, A_2)$  is strict. Let  $f_B : (\text{S})\text{Sp}(B) \rightarrow \{1, 2\}$  be the function chosen by  $S$  for the team  $B$  and  $i := f_B(B_1, B_2)$ . If  $a \notin A_i$ , then  $A_i = B_i \in \mathbb{A}_i \cap \mathbb{B}_i$  and by Lemma 3.3,  $D$  has a winning strategy from the position  $(k_i, \mathbb{A}_i, \mathbb{B}_i)$ . Consequently, we proceed with the case where  $a \in A_i$  for all  $A, B$  as above. Then  $B_i = A_i \setminus \{a\}$  is a neighbour of  $A_i$  in  $\mathbb{B}_i$ , i.e., on the opposite side in the position  $(k_i, \mathbb{A}_i, \mathbb{B}_i)$ . We see that for each neighbour  $B$  of  $A$ , we obtain a neighbour of  $A_1$  in  $\mathbb{B}_1$ , or one of  $A_2$  in  $\mathbb{B}_2$ . Furthermore, if  $B = A \setminus \{a\}$  and  $B' = A \setminus \{a'\}$  are distinct neighbours of  $A$ , then  $A_i \setminus \{a\}$  and  $A_i \setminus \{a'\}$  are distinct neighbours of  $A_i$ . For this reason,  $D(\mathbb{A}_1, \mathbb{B}_1) + D(\mathbb{A}_2, \mathbb{B}_2) \geq D(\mathbb{A}, \mathbb{B})$ .  $\square$

For the rest of this section, we study a fragment  $\text{PL}(\Sigma)$  with operators from  $\Sigma = \{\odot, \wedge, \vee, \check{\vee}\}$ . All results are lower bounds for this fragment and are naturally preserved by any fragment  $\text{PL}(\Sigma')$  with  $\Sigma' \subseteq \Sigma$ .

We gather the above lemmas as the following theorem stating the usefulness of density.

**Theorem 3.8.** *If  $k_0 < D(\mathbb{A}_0, \mathbb{B}_0)$ , then  $D$  has a winning strategy in the game  $\text{FS}_{k_0}^\Sigma(\mathbb{A}_0, \mathbb{B}_0)$ .*

*Proof.* We define a strategy  $\delta$  for D and show that if D plays according to  $\delta$ , the condition  $k < D(\mathbb{A}, \mathbb{B})$  is maintained in all positions  $(k, \mathbb{A}, \mathbb{B})$ .

Let  $(k, \mathbb{A}, \mathbb{B})$  be a position of the game  $\text{FS}_{k_0}^{\Sigma}(\mathbb{A}_0, \mathbb{B}_0)$ . By induction hypothesis,  $k < D(\mathbb{A}, \mathbb{B})$ .

- If S makes a  $\heartsuit$ -move, then by the first item of Lemma 3.7,  $D(\mathbb{A}_1, \mathbb{B}) + D(\mathbb{A}_2, \mathbb{B}) \geq D(\mathbb{A}, \mathbb{B})$ . Assume for contradiction that  $k_i \geq D(\mathbb{A}_i, \mathbb{B})$  for  $i \in \{1, 2\}$ . Then

$$k = k_1 + k_2 \geq D(\mathbb{A}_1, \mathbb{B}) + D(\mathbb{A}_2, \mathbb{B}) \geq D(\mathbb{A}, \mathbb{B}) > k,$$

which is a contradiction. Therefore  $k_i < D(\mathbb{A}_i, \mathbb{B})$  for some  $i \in \{1, 2\}$  and D chooses that  $i$  to continue the game.

- The case of a  $\wedge$ -move is similar, the second item of Lemma 3.7.
- If S makes a  $\vee$ -move, then by the third item of Lemma 3.7, D has a winning strategy from a following position  $(k_i, \mathbb{A}_i, \mathbb{B}_i)$  or  $D(\mathbb{A}_1, \mathbb{B}_1) + D(\mathbb{A}_2, \mathbb{B}_2) \geq D(\mathbb{A}, \mathbb{B})$ . In the first case D chooses the position  $(k_i, \mathbb{A}_i, \mathbb{B}_i)$  and follows the strategy given by the lemma. In the second case D chooses a following position that maintains the condition  $k < D(\mathbb{A}, \mathbb{B})$  just like in the  $\heartsuit$ -case above.
- If S makes a literal move, since  $D(\mathbb{A}, \mathbb{B}) > k \geq 1$ , by Lemma 3.6, D wins the game. Note that the case  $k = 0$  is not possible since all binary connective moves lead to positions with positive  $k$ , and a literal move always ends the game.  $\square$

**Lemma 3.9.** *No set  $\mathbb{A}$  of  $\Phi$ -teams can be defined with a  $\text{PL}(\Sigma)$ -formula of width less than  $D(\mathbb{A}, \text{Tms}(\Phi) \setminus \mathbb{A})$ .*

*Proof.* Let  $\mathbb{B} := \text{Tms}(\Phi) \setminus \mathbb{A}$ . Now defining  $\mathbb{A}$  amounts to separating  $\mathbb{A}$  from  $\mathbb{B}$ . If  $k < D(\mathbb{A}, \mathbb{B})$ , then by Theorem 3.8, D has a winning strategy in the game  $\text{FS}_{k_0}^{\Sigma}(\mathbb{A}, \mathbb{B})$  and by Theorem 3.2,  $\mathbb{A}$  and  $\mathbb{B}$  cannot be separated by a formula with width  $k$ .  $\square$

With the above lemma, we are now in the position to prove the main theorem of this section.

*Proof of Theorem 3.4.* We find in each case a team which satisfies the desired property  $\mathbb{A}$  and has the desired number of neighbours which do not. Then  $D(\mathbb{A}, \text{Tms}(\Phi) \setminus \mathbb{A})$  is greater than or equal to the desired number and the claim follows from Lemma 3.9 along with the fact that length is always greater than width.

- (1) First is the cardinality  $k \pmod{m}$  of  $\Phi_n$ -teams. Let  $k' = 2^n - [2^n - k]_m$ . We first note that  $k' \leq 2^n$  so there is a  $\Phi_n$ -team  $T_1$  with cardinality  $k'$ . Furthermore,

$$k' \equiv [2^n - 2^n + k]_m \equiv k \pmod{m}.$$

Now  $|T_1| \equiv k \pmod{m}$  and  $T_1$  has  $k'$  neighbours with smaller cardinality.

- (2) For a specific cardinality  $k \leq 2^n$ , if  $T_2$  is any team with cardinality  $k$ , then  $T_2$  clearly has  $k$  neighbours with a smaller cardinality.
- (3) Next is the inclusion atom  $p_1 \cdots p_n \subseteq q_1 \cdots q_n$ . If  $s(p_1) \cdots s(p_n)$  is a binary representation of the number  $i$ , we denote this by  $s(\vec{p}) = i$ . For  $i \in \{0, \dots, 2^n - 1\}$ , let  $s_i$  be the assignment with  $s_i(\vec{p}) = i$  and  $s_i(\vec{q}) = [i + 1]_{2^n}$ . Let  $T_3 := \{s_i \mid i \in \{0, \dots, 2^n - 1\}\}$ . Now  $\vec{p}$  and  $\vec{q}$  both get all possible values so  $T_3 \models p_1 \cdots p_n \subseteq q_1 \cdots q_n$ . Furthermore, for any  $s_i \in T_3$ , we have  $T_3 \setminus \{s_i\} \not\models p_1 \cdots p_n \subseteq q_1 \cdots q_n$  since  $\vec{p}$  gets the value  $[i + 1]_{2^n}$  but  $\vec{q}$  does not. Thus there are  $|T_3| = 2^n$  neighbours of  $T_3$  which do not satisfy the inclusion atom.

- (4) For the independence atom  $p_1 \cdots p_n \perp q_1 \cdots q_m$ , let  $T_4$  be the full team with domain  $\{p_1, \dots, p_n, q_1, \dots, q_m\}$ . Clearly  $T_4 \models p_1 \cdots p_n \perp q_1 \cdots q_m$  and  $(T_4 \setminus \{s\}) \not\models p_1 \cdots p_n \perp q_1 \cdots q_m$  for any assignment  $s \in T_4$ . Thus there are  $|T_4| = 2^{n+m}$  neighbours of  $T_4$  which do not satisfy the independence atom.
- (5) Finally, for the anonymity atom  $p_1 \cdots p_n \Upsilon q$ , let  $T_5$  be the full team with domain  $\{p_1, \dots, p_n, q\}$ . Clearly  $T_5 \models p_1 \cdots p_n \Upsilon q$  and  $T_5 \setminus \{s\} \not\models p_1 \cdots p_n \Upsilon q$  for any  $s \in T_5$ . We now have  $|T_5| = 2^{n+1}$  neighbours of  $T_5$  which do not satisfy the anonymity atom.  $\square$

In the above, we did not prove lower bounds for the atoms of dependence and exclusion. The reason for this is that the invariant we use for the formula size game is density, which is defined via the neighbourhood relation. The remaining two atoms are downward closed, so a team which satisfies such an atom cannot have any neighbours which do not satisfy the same atom. For this reason, the above strategy fails for these two atoms. We present a different approach in the next section.

**3.3. Lower bounds via upper dimension.** For the lower bounds of dependence and exclusion atoms we employ the notion of *upper dimension*, which was successfully used to prove lower bounds by Hella et al. [HLSV14]. Their paper mainly concerns the expressive power of modal dependence logic, but at the end it is shown that defining the dependence atom in modal logic with Boolean disjunction  $\otimes$  requires a formula with length at least  $2^n$ . However, the logic they consider again has downward closure. The existential fragment is not downward closed, so we adapt the technique of Hella et al. accordingly. We first state the lower bounds as a theorem and then prove it in this section.

**Theorem 3.10.** *Let  $\Sigma = \{\otimes, \wedge, \vee, \dot{\vee}\}$  and  $n \geq 1$ .*

- *A PL( $\Sigma$ )-formula that defines  $\models (p_1 \cdots p_n; q)$  has length at least  $2^n$ .*
- *A PL( $\Sigma$ )-formula that defines  $p_1 \cdots p_n \mid q_1 \cdots q_n$  has length at least  $2^n$ .*

For now, we will assume that  $\Sigma = \{\otimes, \wedge, \vee\}$ . We will show in the next subsection that this imposes no restriction on the results, as for every PL( $\{\otimes, \wedge, \vee, \dot{\vee}\}$ )-formula that is local there is an equivalent PL( $\{\otimes, \wedge, \vee\}$ )-formula of the same size.

**Definition 3.11.** Let  $\varphi \in \text{PL}(\Sigma, \Phi)$ . A *generator* of  $\varphi$  is a set  $\mathbb{G}(\varphi)$  of pairs  $(S, U)$  such that  $S \subseteq U$ , and for each  $\Phi$ -team  $T$  it holds that  $T \models \varphi$  precisely if there is  $(S, U) \in \mathbb{G}(\varphi)$  such that  $S \subseteq T \subseteq U$ . The *upper dimension*  $\text{Dim}(\mathbb{G})$  of  $\mathbb{G}$  is the number of distinct upper bounds in  $\mathbb{G}$ :

$$\text{Dim}(\mathbb{G}) := |\{U : (S, U) \in \mathbb{G}\}|.$$

The *upper dimension* of  $\varphi$ , denoted  $\text{Dim}(\varphi)$ , is the minimal upper dimension of a generator of  $\varphi$ :

$$\text{Dim}(\varphi) := \min\{\text{Dim}(\mathbb{G}) \mid \mathbb{G} \text{ is a generator of } \varphi\}.$$

That we count only the upper bounds  $U$  is analogous to Hella et al. [HLSV14], who considered downward closed formulas and defined generators only in terms of  $U$ . Indeed, with downward closure we could simply set  $S := \emptyset$  and obtain a definition equivalent to theirs. For arbitrary formulas  $\varphi$  however (even with the empty team property), we could have  $(S, U) \in \mathbb{G}(\varphi)$ , but  $\emptyset \subsetneq X \subsetneq S \subseteq U$  for some  $X$  such that  $X \not\models \varphi$ . Since the subformulas defining a downward closed formula are not necessarily downward closed, the inductive proofs in our results only work if we additionally keep track of the  $S$ .

**Lemma 3.12.** *Let  $\varphi, \psi \in \text{PL}(\Sigma, \Phi)$  and  $\Phi = \{p_1, \dots, p_n\}$ . We have the following estimates:*

- $\text{Dim}(l) \leq 1$  for any  $\Phi$ -literal  $l$ ,
- $\text{Dim}(\varphi \wedge \psi) \leq \text{Dim}(\varphi) \cdot \text{Dim}(\psi)$ ,
- $\text{Dim}(\varphi \vee \psi) \leq \text{Dim}(\varphi) \cdot \text{Dim}(\psi)$ ,
- $\text{Dim}(\varphi \otimes \psi) \leq \text{Dim}(\varphi) + \text{Dim}(\psi)$ ,

*Proof.* For the binary connectives, let  $\mathbb{G}(\varphi)$  and  $\mathbb{G}(\psi)$  be minimal generators of  $\varphi$  and  $\psi$ , respectively.

- Let  $T$  be the full  $\Phi$ -team. Any positive literal  $l \in \{p, \neg p, \top, \perp \mid p \in \Phi\}$  has flatness, so  $\{(\emptyset, \{s \in T \mid \{s\} \models l\})\}$  generates  $l$ . The negative literals  $l \in \{\sim p, \sim \neg p, \sim \perp \mid p \in \Phi\}$  are upward closed, so  $\{(\{s\}, T) \mid s \in T : \{s\} \models l\}$  generates  $l$ . Finally,  $\sim \top$  is unsatisfiable, so it has the empty generator.
- For the conjunction, it is easy to check that  $\mathbb{G}(\cap) := \{(S_1 \cup S_2, U_1 \cap U_2) \mid (S_1, U_1) \in \mathbb{G}(\varphi), (S_2, U_2) \in \mathbb{G}(\psi)\}$  is a generator of  $\varphi \wedge \psi$ , so

$$\text{Dim}(\varphi \wedge \psi) \leq \text{Dim}(\mathbb{G}(\cap)) \leq \text{Dim}(\mathbb{G}(\varphi)) \cdot \text{Dim}(\mathbb{G}(\psi)) = \text{Dim}(\varphi) \cdot \text{Dim}(\psi).$$

- For the lax disjunction, let  $\mathbb{G}(\cup) := \{(S_1 \cup S_2, U_1 \cup U_2) \mid (S_1, U_1) \in \mathbb{G}(\varphi), (S_2, U_2) \in \mathbb{G}(\psi)\}$ . If  $T \models \varphi \vee \psi$  via some split  $(T_1, T_2)$ , there are  $(S_1, U_1) \in \mathbb{G}(\varphi)$  and  $(S_2, U_2) \in \mathbb{G}(\psi)$  such that  $S_i \subseteq T_i \subseteq U_i$  for  $i \in \{1, 2\}$ . Then  $S_1 \cup S_2 \subseteq T \subseteq U_1 \cup U_2$ .

Conversely, assume  $(S_1, U_1) \in \mathbb{G}(\varphi)$  and  $(S_2, U_2) \in \mathbb{G}(\psi)$  such that  $S_1 \cup S_2 \subseteq T \subseteq U_1 \cup U_2$ . Define  $T_i := (T \cap U_i) \cup S_i$ . Then  $(T_1, T_2)$  is a split of  $T$ , and  $S_i \subseteq T_i \subseteq U_i$  (w.l.o.g.  $S_i \subseteq U_i$ ). Consequently,  $T_1 \models \varphi$  and  $T_2 \models \psi$ , so  $T \models \varphi \vee \psi$ . Thus  $\mathbb{G}(\cup)$  is a generator of  $\varphi \vee \psi$  and

$$\text{Dim}(\varphi \vee \psi) \leq \text{Dim}(\mathbb{G}(\cup)) \leq \text{Dim}(\mathbb{G}(\varphi)) \cdot \text{Dim}(\mathbb{G}(\psi)) = \text{Dim}(\varphi) \cdot \text{Dim}(\psi).$$

- For the Boolean disjunction, clearly  $\mathbb{G}(\varphi) \cup \mathbb{G}(\psi)$  is a generator of  $\varphi \otimes \psi$ . □

Let  $\text{occ}_{\otimes}(\varphi)$  denote the number of occurrences of  $\otimes$  inside  $\varphi$ .

**Lemma 3.13** [HLSV14, Proposition 5.9]. *Let  $\varphi \in \text{PL}(\Sigma)$ . Then  $\text{Dim}(\varphi) \leq 2^{\text{occ}_{\otimes}(\varphi)}$ .*

*Proof.* By induction on  $\varphi$ , using the previous lemma. For literals  $\varphi = l$ ,  $\text{Dim}(l) \leq 1 = 2^{\text{occ}_{\otimes}(l)}$ . For  $\nabla \in \{\wedge, \vee\}$ , it holds that

$$\begin{aligned} \text{Dim}(\psi \nabla \theta) &\leq \text{Dim}(\psi) \cdot \text{Dim}(\theta) \\ &\leq 2^{\text{occ}_{\otimes}(\psi)} \cdot 2^{\text{occ}_{\otimes}(\theta)} = 2^{\text{occ}_{\otimes}(\psi) + \text{occ}_{\otimes}(\theta)} = 2^{\text{occ}_{\otimes}(\varphi)} \end{aligned}$$

and for the Boolean disjunction,

$$\begin{aligned} \text{Dim}(\psi \otimes \theta) &\leq \text{Dim}(\psi) + \text{Dim}(\theta) \leq \text{Dim}(\psi) \cdot \text{Dim}(\theta) + 1 \\ &\leq 2^{\text{occ}_{\otimes}(\psi)} \cdot 2^{\text{occ}_{\otimes}(\theta)} + 1 \leq 2^{\text{occ}_{\otimes}(\psi) + \text{occ}_{\otimes}(\theta) + 1} = 2^{\text{occ}_{\otimes}(\varphi)}. \end{aligned} \quad \square$$

Next, we show that the upper dimension of the dependence atom and the exclusion atom is at least doubly exponential.

**Lemma 3.14.** *Let  $n \geq 1$ , let  $p_1, \dots, p_n, q_1, \dots, q_n \in \Phi$  be pairwise distinct propositions,  $\vec{p} = (p_1, \dots, p_n)$ , and  $\vec{q} = (q_1, \dots, q_n)$ . Then  $\text{Dim}(=(\vec{p}; q_1)) \geq 2^{2^n}$  and  $\text{Dim}(\vec{p} \mid \vec{q}) \geq 2^{2^n} - 2$ .*

*Proof.* We prove a more general result and then apply it to the two atoms. Let  $\varphi$  be any formula and  $\Phi = \text{Prop}(\varphi)$ . We show that the size of a generator of  $\varphi$  is always at least the number of maximal  $\Phi$ -teams of  $\varphi$ , where a  $\Phi$ -team  $X$  is *maximal* if it satisfies  $\varphi$  but no  $\Phi$ -team  $Y$  with  $Y \supsetneq X$  satisfies  $\varphi$ . Suppose that  $\varphi$  has  $m$  distinct maximal teams, but  $\mathbb{G}$  is a generator of  $\varphi$  with  $|\mathbb{G}| < m$ . Then there are distinct maximal teams  $X_1, X_2$  and pairs

$(S_1, U), (S_2, U) \in \mathbb{G}$  such that  $X_1, X_2 \subseteq U$ . Since  $X_1$  is maximal and  $U \models \varphi$  by definition of generator, we have  $X_1 = U$ . But by the same argument  $X_2 = U$ , contradiction.

Next, we show that the atoms have at least  $2^{2^n}$  maximal teams. We start with the dependence atom. For each  $f: \{0, 1\}^n \rightarrow \{1, 0\}$ , let

$$X(f) := \{s: \Phi \rightarrow \{0, 1\} \mid f(s(\vec{p})) = s(q_1)\}.$$

Then  $X(f)$  is maximal for  $(\vec{p}; q_1)$ . Since  $f_1 \neq f_2$  implies  $X(f_1) \neq X(f_2)$ , there are at least  $2^{2^n}$  distinct maximal teams.

For the exclusion atom, we proceed similarly. A function  $f: \{0, 1\}^n \rightarrow \{1, 0\}$  is *non-constant* if  $f(\vec{b}) \neq f(\vec{b}')$  for some  $\vec{b}, \vec{b}' \in \{0, 1\}^n$ . Now, for all non-constant  $f: \{0, 1\}^n \rightarrow \{1, 0\}$ , let

$$X(f) := \{s: \Phi \rightarrow \{0, 1\} \mid f(s(\vec{p})) = 1 \text{ and } f(s(\vec{q})) = 0\}.$$

Clearly  $X(f) \models \vec{p} \mid \vec{q}$ , as for every  $s \in X(f)$  we have  $f(s(\vec{p})) \neq f(s(\vec{q}))$ , hence  $s(\vec{p}) \neq s(\vec{q})$ .

Next, we show that these are distinct teams, i.e.,  $f_1 \neq f_2$  implies  $X(f_1) \neq X(f_2)$ . Suppose  $f_1 \neq f_2$ , w.l.o.g. there is  $\vec{b} \in \{0, 1\}^n$  such that  $f_1(\vec{b}) = 1$  and  $f_2(\vec{b}) = 0$ . Consider the assignment  $s$  defined by  $s(\vec{p}) = \vec{b}$  and  $s(\vec{q})$  defined in a way such that  $f_1(s(\vec{q})) = 0$  (recall that  $f_1$  is non-constant). Then  $s \in X(f_1)$  but  $s \notin X(f_2)$ . Consequently, there are  $2^{2^n} - 2$  such teams (as there are  $2^{2^n} - 2$  non-constant functions).

It remains to show that these teams are maximal, i.e.,  $X(f) \subsetneq Y$  implies  $Y \not\models \vec{p} \mid \vec{q}$  for all  $\Phi$ -teams  $Y$ . Suppose  $s \in Y \setminus X(f)$ . Then  $f(s(\vec{p})) = 0$  or  $f(s(\vec{q})) = 1$ . By symmetry, we consider only the first case. As  $f$  is non-constant, there exists  $\vec{b} \in \{0, 1\}^n$  with  $f(\vec{b}) = 1$ . Now, define an assignment  $s'$  such that  $s'(\vec{p}) = \vec{b}$  and  $s'(\vec{q}) = s(\vec{p})$ . Then  $f(s'(\vec{p})) = 1$  and  $f(s'(\vec{q})) = 0$ , so  $s' \in X(f) \subseteq Y$ . Hence  $s, s' \in Y$ , but  $s'(\vec{q}) = s(\vec{p})$ , so  $Y \not\models \vec{p} \mid \vec{q}$ .  $\square$

We conclude the section with the following exponential lower bounds.

*Proof of Theorem 3.10.* We consider the exclusion atom, the dependence atom works analogously. Suppose that  $\varphi \in \text{PL}(\Sigma)$  is equivalent to  $p_1 \cdots p_n \mid q_1 \cdots q_n$ . Then by Lemma 3.14,  $\text{Dim}(\varphi) \geq 2^{2^n} - 2$ , as the upper dimension is a purely semantical property. However, by Lemma 3.13,  $\text{Dim}(\varphi) \leq 2^{\text{occ}_{\mathbb{Q}}(\varphi)} \leq 2^{|\varphi|} - 2$ . With  $n \geq 1$ , the resulting inequality  $2^{2^n} - 2 \leq 2^{|\varphi|} - 2$  implies  $|\varphi| \geq 2^n$ .  $\square$

**3.4. From lax to strict lower bounds.** Before, we proved lower bounds for the dependence and exclusion atom for the restricted operator set  $\Sigma = \{\otimes, \wedge, \vee\}$ , in particular with only lax disjunction. Next, we incorporate the strict disjunction  $\dot{\vee}$ .

The idea is the following: Define the *relaxation*  $\varphi^*$  of a formula  $\varphi$  as the formula where every occurrence of  $\dot{\vee}$  is replaced by  $\vee$ . We will prove that a formula  $\varphi$  and its relaxation are equivalent, provided  $\varphi$  is local. This is the case in particular for the dependence and the exclusion atom, for which all lower bounds with  $\vee$  then also hold with  $\dot{\vee}$ . This additional assumption of locality is needed, since formulas containing  $\dot{\vee}$  can be non-local. For example,  $\text{NE} \dot{\vee} \text{NE}$  is not equivalent to its relaxation  $\text{NE} \vee \text{NE} \equiv \text{NE}$ .

The intuition is that if  $\varphi^*$  is satisfiable, then  $\varphi$  is also satisfiable if we just make the domain larger, since the only way  $\varphi$  could be false while  $\varphi^*$  is true is that we “run out of assignments” for  $\dot{\vee}$ . But if now  $\varphi$  is local, then enlarging the domain should have no effect so that then we have  $\varphi \equiv \varphi^*$ .

We begin with proving the first part formally. If  $T$  is a  $\Phi$ -team and  $\Psi \supseteq \Phi$ , then the  $\Psi$ -expansion of  $T$  is

$$T[\Psi] := \{ s : \Psi \rightarrow \{0, 1\} \mid s \upharpoonright \Phi \in T \}.$$

Intuitively it is obtained from  $T$  by duplicating all assignments in  $T$  for all possible values for propositions  $p \in \Psi \setminus \Phi$ . Observe that  $T[\Psi] \upharpoonright \Phi = T$ .

**Lemma 3.15.** *Let  $\varphi \in \text{PL}(\{\wedge, \vee, \dot{\vee}\})$ . If a  $\Phi$ -team  $T$  satisfies  $\varphi^*$ , then there is a domain  $\Psi \supseteq \Phi$  such that  $T[\Psi]$  satisfies  $\varphi$ .*

*Proof.* The idea is that any lax splitting can be simulated by a strict splitting by duplicating assignments in the team such that no assignment needs to be used in both halves of the splitting. We show the following stronger statement by induction on  $\varphi$ : If  $\varphi^*$  is satisfied by a  $\Phi$ -team  $T$ , then there is a domain  $\Psi \supseteq \Phi$  such that, for all domains  $\Psi' \supseteq \Psi$  and  $\Psi'$ -teams  $X$ , it holds that  $X \upharpoonright \Psi = T[\Psi]$  implies  $X \models \varphi$ .

The case where  $\varphi$  is a literal or a conjunction is straightforward. So suppose  $\varphi = \psi_1 \dot{\vee} \psi_2$  or  $\varphi = \psi_1 \vee \psi_2$ , and assume  $T \models \varphi^* = \psi_1^* \vee \psi_2^*$  via a (lax) split  $(S_1, S_2)$  of  $T$ , i.e.,  $S_i \models \psi_i^*$ . For  $i \in \{1, 2\}$ , there is  $\Psi_i \supseteq \Phi$  such that for all  $\Psi'_i \supseteq \Psi_i$  and  $\Psi'_i$ -teams  $X_i$  it holds that  $X_i \upharpoonright \Psi_i = S_i \upharpoonright \Psi_i$  implies  $X_i \models \psi_i$ . We pick  $p \in \text{Prop} \setminus (\Psi_1 \cup \Psi_2)$ , and let  $\Psi := \Psi_1 \cup \Psi_2 \cup \{p\}$ . Now assume  $X \upharpoonright \Psi = T[\Psi]$  for some  $\Psi'$ -team  $X$ , where  $\Psi' \supseteq \Psi$ . We have to show that  $X \models \psi_1 \dot{\vee} \psi_2$ . This holds via the strict split  $(Y_1 \cup Z_1, Y_2 \cup Z_2)$  of  $X$ , where

$$Y_1 := \{s \in X \mid s \upharpoonright \Phi \in S_1 \cap S_2 \text{ and } s(p) = 1\}$$

$$Y_2 := \{s \in X \mid s \upharpoonright \Phi \in S_1 \cap S_2 \text{ and } s(p) = 0\}$$

$$Z_1 := \{s \in X \mid s \upharpoonright \Phi \in S_1 \setminus S_2\}$$

$$Z_2 := \{s \in X \mid s \upharpoonright \Phi \in S_2 \setminus S_1\}$$

□

We now prove the second part.

**Theorem 3.16.** *A formula  $\varphi \in \text{PL}(\{\otimes, \wedge, \vee, \dot{\vee}\})$  is local if and only if it is equivalent to its relaxation  $\varphi^*$ .*

*Proof.* If  $\varphi$  is equivalent to  $\varphi^*$ , then  $\varphi$  is local by Proposition 2.10. For the converse, let  $\varphi$  be local. We have to prove  $\varphi \equiv \varphi^*$ . The direction  $\varphi \models \varphi^*$  is easy to prove by induction. For the other direction,  $\varphi^* \models \varphi$ , we first transform  $\varphi$  and  $\varphi^*$  into a disjunction of  $\text{PL}(\{\wedge, \vee, \dot{\vee}\})$ -formulas using the distributive laws

$$\theta_1 \circ (\theta_2 \otimes \theta_3) \equiv (\theta_1 \circ \theta_2) \otimes (\theta_1 \circ \theta_3)$$

$$(\theta_1 \otimes \theta_2) \circ \theta_3 \equiv (\theta_1 \circ \theta_3) \otimes (\theta_2 \circ \theta_3)$$

for  $\circ \in \{\wedge, \vee, \dot{\vee}\}$ . We obtain  $\varphi \equiv \bigotimes_{i=1}^n \psi_i$  and  $\varphi^* \equiv \bigotimes_{i=1}^n \psi_i^*$  for suitable  $\psi_1, \dots, \psi_n \in \text{PL}(\{\wedge, \vee, \dot{\vee}\})$ .<sup>1</sup>

Let now  $T$  be a  $\Phi$ -team such that  $T \models \varphi^*$ , where  $\Phi \supseteq \text{Prop}(\varphi)$ . Then  $T \models \psi_i^*$  for some  $i$ . By Lemma 3.15, there is a domain  $\Psi \supseteq \Phi$  such that  $T[\Psi] \models \psi_i$ , which implies that  $T[\Psi] \models \varphi$ . Since  $T[\Psi] \upharpoonright \Phi = T$  and  $\varphi$  is local, we conclude  $T \models \varphi$ , as desired. □

<sup>1</sup>Such normal forms with  $\otimes$  are standard in team logic (cf. [HLSV14, Theorem 3.5], [KMSV15, Theorem 3.4], [YV16, Lemma 4.9], [Vir17, Proposition 6.2]).

## 4. POLYNOMIAL UPPER BOUNDS FOR TEAM PROPERTIES

In this section, we complement the exponential lower bounds presented in Theorem 3.4 by polynomial upper bounds in the fragment  $\text{PL}(\{\emptyset, \wedge, \vee\})$ . Notably, among these polynomially definable properties are the *negations* of all atoms of dependency considered previously. This exhibits an interesting asymmetry of succinctness between the standard atoms of dependency and their negations. For the parity of teams there is no such asymmetry and we have exponential lower bounds for both even and odd cardinality. Nevertheless, in the subsequent subsection, we will present a polynomial upper bound for parity in a stronger logic than  $\text{PL}(\{\emptyset, \wedge, \vee, \dot{\vee}\})$ .

**4.1. Upper bounds for the atoms of dependency.** As with the lower bounds, we will first state the theorem and prove it with a series of lemmas. The length of a tuple  $\vec{\varphi} = (\varphi_1, \dots, \varphi_n)$  of formulas is  $|\vec{\varphi}| := \sum_{i=1}^n |\varphi_i|$ . The *negation* of a formula  $\varphi$  is  $\sim\varphi$ . Throughout this section, let  $\vec{\alpha}, \vec{\beta}, \vec{\gamma}$  always denote tuples of purely propositional formulas  $\vec{\alpha} = (\alpha_1, \dots, \alpha_n)$ ,  $\vec{\beta} = (\beta_1, \dots, \beta_m)$ , and  $\vec{\gamma} = (\gamma_1, \dots, \gamma_k)$ , where  $n, m, k \geq 0$ .

**Theorem 4.1.** *Let  $\Sigma \supseteq \{\emptyset, \wedge, \vee\}$ .*

- *The dependence atom  $=(\vec{\alpha}; \vec{\beta})$  is equivalent to the negation of a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(|\vec{\alpha}\vec{\beta}|)$ .*
- *The exclusion atom  $\vec{\alpha} \mid \vec{\beta}$  is equivalent to the negation of a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(n|\vec{\alpha}\vec{\beta}|)$ .*
- *The inclusion atom  $\vec{\alpha} \subseteq \vec{\beta}$  is equivalent to the negation of a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(n|\vec{\alpha}\vec{\beta}|)$ .*
- *The conditional independence atom  $\vec{\alpha} \perp_{\vec{\gamma}} \vec{\beta}$  is equivalent to the negation of a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(n(n+m+k)|\vec{\alpha}\vec{\beta}\vec{\gamma}|)$ .*
- *The anonymity atom  $\vec{\alpha}\Upsilon\beta$  is equivalent to the negation of a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(n|\beta| + |\vec{\alpha}|)$ .*

*Additionally, for the dependence and exclusion atoms,  $\Sigma \supseteq \{\emptyset, \wedge, \dot{\vee}\}$  yields the same result. Furthermore, all these formulas are logspace-computable.*

*Proof.* We prove these results in Lemmas 4.2 to 4.7. For the formulas that are equivalent to the negations of the dependence and exclusion atom, note that every occurrence of  $\vee$  in them is of the form  $\alpha \vee \varphi$  for purely propositional  $\alpha$ . But then  $\alpha \vee \varphi \equiv \alpha \dot{\vee} \varphi$  by Proposition 2.6. For this reason, these results hold for  $\Sigma \supseteq \{\emptyset, \wedge, \dot{\vee}\}$  as well.  $\square$

**Dependence atom.** It is well-known that the dependence atom can be efficiently rewritten by means of other connectives in most flavors of team logic that have unrestricted negation (see, e.g., [Vä07, KMSV15, HKVV18]). For the sake of completeness, we will also state such a formula here.

The following formula expresses the negation of the dependence atom  $=(\vec{\alpha}; \vec{\beta})$  and has length  $\mathcal{O}(|\vec{\alpha}\vec{\beta}|)$ . Recall the defined abbreviations  $\text{E}\alpha := \top \vee (\text{NE} \wedge \alpha)$  and  $(\vec{\alpha} \leftrightarrow \vec{\beta}) := \bigwedge_{i=1}^n ((\alpha_i \wedge \beta_i) \vee (\neg\alpha_i \wedge \neg\beta_i))$ , which we will extensively use in this section.



The following formula defines  $\sim=(\vec{\alpha}; \vec{\beta})$ .

$$\varphi(\vec{\alpha}; \vec{\beta}) := \top \vee \left( \bigwedge_{i=1}^n (\alpha_i \otimes \neg \alpha_i) \wedge \bigvee_{i=1}^m (\mathbf{E}\beta_i \wedge \mathbf{E}\neg\beta_i) \right)$$

**Lemma 4.2.**  $\sim=(\vec{\alpha}; \vec{\beta}) \equiv \varphi(\vec{\alpha}; \vec{\beta})$ .

*Proof.* Analogously to [HKVV18, Proposition 2.5].  $\square$

Next, we require the abbreviation  $\alpha \hookrightarrow \varphi := \neg\alpha \vee (\alpha \wedge \varphi)$ , or equivalently, with strict splitting,  $\alpha \hookrightarrow \varphi := \neg\alpha \dot{\vee} (\alpha \wedge \varphi)$ . It was introduced by Galliani [Gal15] and has the semantics  $T \models \alpha \hookrightarrow \varphi \Leftrightarrow T_\alpha \models \varphi$ , where  $T_\alpha := \{s \in T \mid s \models \alpha\}$ .

Before we define the next atom, we introduce two helper formulas  $\theta^=$  and  $\theta^\neq$ , which we will explain below.

$$\begin{aligned} \theta^=(\vec{\alpha}; \vec{\beta}; \gamma) &:= \bigwedge_{i=1}^n \bigvee_{l \in \{\top, \perp\}} \left( (\gamma \wedge (\alpha_i \leftrightarrow l)) \vee (\neg\gamma \wedge (\beta_i \leftrightarrow l)) \right) \\ \theta^\neq(\vec{\alpha}; \vec{\beta}; \gamma) &:= \bigvee_{i=1}^n \left( \mathbf{E}\gamma \wedge \bigvee_{l \in \{\top, \perp\}} \left( (\gamma \wedge (\alpha_i \leftrightarrow l)) \vee (\neg\gamma \wedge (\beta_i \leftrightarrow l)) \right) \right) \end{aligned}$$

These are PL( $\{\otimes, \wedge, \vee\}$ )-formulas of length  $\mathcal{O}(n|\gamma| + |\vec{\alpha}| + |\vec{\beta}|)$ .

The purpose of  $\theta^=(\vec{\alpha}, \vec{\beta}, \gamma)$  and  $\theta^\neq(\vec{\alpha}, \vec{\beta}, \gamma)$  is the following. The definitions of the various dependency atoms are all based on comparison of pairs of assignments in a team. For instance,  $\vec{\alpha} \mid \vec{\beta}$  holds if  $s(\vec{\alpha}) \neq s'(\vec{\beta})$  for all  $s, s' \in T$ , and so on. Loosely speaking,  $\theta^=(\vec{\alpha}, \vec{\beta}, \gamma)$  and  $\theta^\neq(\vec{\alpha}, \vec{\beta}, \gamma)$  test the values  $s(\vec{\alpha})$  and  $s'(\vec{\beta})$  for equality resp. inequality for pairs  $(s, s') \in T_\gamma \times T_{\neg\gamma}$ . The restriction to  $T_\gamma \times T_{\neg\gamma}$  is unfortunately necessary in our implementation of  $\theta^=$  and  $\theta^\neq$ , so  $s$  and  $s'$  must differ in some formula  $\gamma$  that is known *a priori*. While this seems to complicate the matter, we can actually find such  $\gamma$  for all of the atoms of dependency.

Before we proceed with defining the atoms, we prove the semantics of  $\theta^=$  and  $\theta^\neq$ . Another constraint is that they work only for the subclass of teams  $T$  where  $|\{s(\vec{\alpha}) \mid s \in T_\gamma\}| = 1$ , i.e., all  $s \in T_\gamma$  agree on the value  $s(\vec{\alpha})$ , but this again suffices for our purpose.

**Lemma 4.3.** *Let  $T$  be a team such that  $|\{s(\vec{\alpha}) \mid s \in T_\gamma\}| = 1$ . Then the following holds:*

$$\begin{aligned} T \models \theta^=(\vec{\alpha}; \vec{\beta}; \gamma) &\Leftrightarrow \forall (s, s') \in T_\gamma \times T_{\neg\gamma} : s(\vec{\alpha}) = s'(\vec{\beta}) \\ T \models \theta^\neq(\vec{\alpha}; \vec{\beta}; \gamma) &\Leftrightarrow \forall (s, s') \in T_\gamma \times T_{\neg\gamma} : s(\vec{\alpha}) \neq s'(\vec{\beta}) \end{aligned}$$

*Proof.* As  $\theta^=$  is straightforward, let us consider  $\theta^\neq$ .

For “ $\Rightarrow$ ”, by the formula,  $T$  can be divided into  $Y_1 \cup \dots \cup Y_n$  such that  $Y_i \cap T_\gamma \neq \emptyset$  and additionally  $Y_i$  satisfies the respective Boolean disjunction. Now let  $s \in T_\gamma$  and  $s' \in T_{\neg\gamma}$ . For some  $i \geq 1$ ,  $s' \in Y_i$ . Furthermore, there is  $l \in \{\top, \perp\}$  such that  $Y_i \models \mathbf{E}\gamma \wedge (\gamma \wedge (\alpha_i \leftrightarrow l)) \vee (\neg\gamma \wedge (\beta_i \leftrightarrow l))$ . As  $Y_i \models \mathbf{E}\gamma$ , some  $s^* \in Y_i \cap T_\gamma$  exists, and we conclude  $s(\alpha_i) = s^*(\alpha_i) \neq s'(\beta_i)$ .

For “ $\Leftarrow$ ”, we divide  $T$  into teams  $Y_1 \cup \dots \cup Y_n$  as follows. For every  $i \in \{1, \dots, n\}$ , choose  $l \in \{\top, \perp\}$  such that

$$Y_i := \{s \in T_{\neg\gamma} \mid s(\beta_i) \neq s'(\alpha_i), s' \in T_\gamma\}.$$

$Y_i$  is well-defined as  $s'(\alpha_i)$  is constant for all  $s' \in T_\gamma$ . This is a split of  $T$ , as otherwise some  $s \in T_{\neg\gamma}$  is left over with  $s(\beta_i) = s'(\alpha_i)$  for all  $i \in \{1, \dots, n\}$ , contradicting the assumption. Clearly  $Y_i \models E\gamma$ , as  $T_\gamma \neq \emptyset$  and  $T_\gamma \subseteq Y_i$ . It remains to check that setting  $l := \top$  if  $a_i = 1$  (resp.  $l := \perp$  if  $a_i = 0$ ) renders  $(\gamma \wedge (\alpha_i \leftrightarrow l)) \vee (\neg\gamma \wedge (\beta_i \leftrightarrow l))$  true in  $Y_i$ .  $\square$

With  $\theta^\equiv$  and  $\theta^\neq$  we can now define the remaining atoms. To define the condition  $|\{s(\vec{\alpha}) \mid s \in T_\gamma\}| = 1$  in a formula, we use  $\gamma \hookrightarrow \mathbf{1}_\alpha$ , where  $\mathbf{1}_\alpha := \sim\perp \wedge \bigwedge_{i=1}^n =(\alpha_i)$ . Let us call an assignment  $s$  in  $T_\gamma$  that is unique up to  $\alpha$  a *pivot*.

**Exclusion atom.** With the exclusion atom, we exemplify how the formula  $\theta^\equiv$  can be used. A team  $T$  violates the exclusion atom  $\vec{\alpha} \mid \vec{\beta}$  if either some assignment  $s$  satisfies  $\vec{\alpha} \leftrightarrow \vec{\beta}$ , or otherwise if  $s(\vec{\alpha}) = s'(\vec{\beta})$  for distinct  $s, s'$ . Assuming we are only in the latter case, however,  $s$  and  $s'$  must disagree on some  $\alpha_i$ , say,  $s \models \alpha_i$  and  $s' \models \neg\alpha_i$ , since otherwise we again have  $s'(\vec{\alpha}) = s(\vec{\alpha}) = s'(\vec{\beta})$ . Taking now  $\gamma := \alpha_i$ , we can with  $\vee$  split off everything from  $T_\gamma$  except the pivot  $s$ , retain the team  $\{s\} \cup T_{\neg\gamma}$ , and search for  $s'$  in  $T_{\neg\gamma}$  with  $\theta^\equiv$ .

We apply these ideas in the following formula which expresses  $\sim(\vec{\alpha} \mid \vec{\beta})$  and has length  $\mathcal{O}(n|\vec{\alpha}\vec{\beta}|)$ .

$$\varphi(\vec{\alpha}; \vec{\beta}) := E(\vec{\alpha} \leftrightarrow \vec{\beta}) \otimes \bigvee_{\substack{i=1 \\ \gamma \in \{\alpha_i, \neg\alpha_i\}}}^n \left( \top \vee ((E\neg\gamma) \wedge (\gamma \hookrightarrow \mathbf{1}_{\vec{\alpha}}) \wedge \theta^\equiv(\vec{\alpha}; \vec{\beta}; \gamma)) \right)$$

**Lemma 4.4.**  $\sim\vec{\alpha} \mid \vec{\beta} \equiv \varphi(\vec{\alpha}; \vec{\beta})$ .

*Proof.* Suppose  $T \not\models \vec{\alpha} \mid \vec{\beta}$ , so there are  $s, s' \in T$  such that  $s(\vec{\alpha}) = s'(\vec{\beta})$ . First, if  $s(\vec{\alpha}) = s'(\vec{\alpha})$ , then  $T \models E(\vec{\alpha} \leftrightarrow \vec{\beta})$  and we are done. Otherwise,  $s$  and  $s'$  disagree on some  $\gamma \in \{\alpha_i, \neg\alpha_i \mid 1 \leq i \leq n\}$  such that  $s(\gamma) = 1$  and  $s'(\gamma) = 0$ . Then the split  $(T \setminus \{s, s'\}, \{s, s'\})$  satisfies the Boolean disjunct with index  $\gamma$ , as clearly  $\{s, s'\}$  satisfies  $E\neg\gamma$ ,  $\gamma \hookrightarrow \mathbf{1}_{\vec{\alpha}}$ , and  $\theta^\equiv(\vec{\alpha}; \vec{\beta}; \gamma)$ . For the other direction, assume that  $T \models \varphi(\vec{\alpha}; \vec{\beta})$ . Then either  $T \models E(\vec{\alpha} \leftrightarrow \vec{\beta})$  and we are done, or there exist  $\gamma$  and some split  $(S, U)$  of  $T$  such that  $U \models (E\neg\gamma) \wedge (\gamma \hookrightarrow \mathbf{1}_{\vec{\alpha}}) \wedge \theta^\equiv(\vec{\alpha}; \vec{\beta}; \gamma)$ . This implies  $U_\gamma, U_{\neg\gamma} \neq \emptyset$ , so  $s_1(\vec{\alpha}) = s_2(\vec{\beta})$  for some  $(s_1, s_2) \in U_\gamma \times U_{\neg\gamma}$ .  $\square$

**Inclusion atom.** A team  $T$  falsifies the inclusion atom  $\vec{\alpha} \subseteq \vec{\beta}$  if there exists  $s^* \in T$  such that  $s^*(\vec{\alpha}) \neq s^*(\vec{\beta})$  for all  $s \in T$ . In particular, some  $s^* \in T$  must exist such that  $s^*(\alpha_i) \neq s^*(\beta_i)$  for some  $i$ . Similar as for the exclusion atom, it suffices to compare  $s^*(\vec{\alpha})$  and  $s^*(\vec{\beta})$  only for assignments  $s$  such that  $s(\beta_i) \neq s^*(\beta_i)$ , as  $s(\beta_i) = s^*(\beta_i)$  already ensures  $s^*(\vec{\alpha}) \neq s^*(\vec{\beta})$ . Hence  $s^*$  is a pivot for  $\gamma := \beta_i$ , and it suffices to compare pairs from  $\{s^*\} \times T_{\neg\gamma}$  with  $\theta^\neq$ .

The following formula expresses the negation of the inclusion atom  $\vec{\alpha} \subseteq \vec{\beta}$  and has length  $\mathcal{O}(n|\vec{\alpha}\vec{\beta}|)$ .

$$\varphi(\vec{\alpha}; \vec{\beta}) := \bigvee_{\substack{i=1 \\ \gamma \in \{\beta_i, \neg\beta_i\}}}^n \left( \gamma \vee \left( (\gamma \hookrightarrow ((\alpha_i \leftrightarrow \beta_i) \wedge \mathbf{1}_{\vec{\alpha}})) \wedge \theta^\neq(\vec{\alpha}; \vec{\beta}; \gamma) \right) \right)$$

**Lemma 4.5.**  $\sim\vec{\alpha} \subseteq \vec{\beta} \equiv \varphi(\vec{\alpha}; \vec{\beta})$ .

*Proof.* Let  $T \not\models \vec{\alpha} \subseteq \vec{\beta}$ . We show that  $T \models \varphi(\vec{\alpha}; \vec{\beta})$ . By definition, there is  $s^* \in T$  such that  $s^*(\vec{\alpha}) \neq s^*(\vec{\beta})$  for all  $s \in T$ . In particular,  $s^*(\alpha_i) \neq s^*(\beta_i)$  for some  $i \in \{1, \dots, n\}$ . Let  $\gamma \in \{\beta_i, \neg\beta_i\}$  such that  $s^*(\gamma) = 1$ , and consider the subteam  $S := \{s^*\} \cup T_{\neg\gamma}$  of  $T$ . We show that the Boolean disjunct with index  $\gamma$  is satisfied by the split  $(T \setminus S, S)$ . Clearly,  $T \setminus S \models \gamma$ . Moreover,  $S_\gamma = \{s^*\} \models (\alpha_i \leftrightarrow \beta_i) \wedge \mathbf{1}_{\vec{\alpha}}$ . Finally,  $S \models \theta^\neq(\vec{\alpha}; \vec{\beta}; \gamma)$  holds since  $s^*(\vec{\alpha}) \neq s^*(\vec{\beta})$  for all  $s \in T_{\neg\gamma}$  by assumption.

Conversely, assume  $T \models \varphi(\vec{\alpha}; \vec{\beta})$  with  $1 \leq i \leq n$  and  $\gamma \in \{\beta_i, \neg\beta_i\}$  chosen according to a satisfying disjunct of  $\varphi(\vec{\alpha}; \vec{\beta})$ . By the formula,  $T$  can be divided into  $X \cup S$  with  $X \models \gamma$ ,  $S_\gamma \models (\alpha_i \neq \beta_i) \wedge \mathbf{1}_{\vec{\alpha}}$ , and  $S \models \theta^\neq(\vec{\alpha}; \vec{\beta}; \gamma)$ . In particular  $S_\gamma = \{s^*\}$  for some  $s^*$ . We show that for all  $s \in T$  we have  $s^*(\vec{\alpha}) \neq s^*(\vec{\beta})$ , so  $T \not\models \vec{\alpha} \subseteq \vec{\beta}$ . For all  $s \in T_{\neg\gamma}$ , this follows since  $T_{\neg\gamma} = S_{\neg\gamma}$  and  $S \models \theta^\neq(\vec{\alpha}; \vec{\beta}; \gamma)$ . For all  $s \in T_\gamma$ , this follows since  $s(\beta_i) = s^*(\beta_i)$  (recall that  $\gamma \in \{\beta_i, \neg\beta_i\}$ ) and  $s^*(\beta_i) \neq s^*(\alpha_i)$ .  $\square$

**Independence atom.** The independence atom  $\vec{\alpha} \perp_{\vec{\gamma}} \vec{\beta}$  is a bit more complicated: It is false if there are  $s^*, s^\circ \in T$  that agree on  $\vec{\gamma}$ , and any  $s \in T$  disagrees either with  $s^*$  on  $\vec{\alpha}\vec{\gamma}$  or with  $s^\circ$  on  $\vec{\beta}$ . We separate the team along two “axes”, with  $\delta$  and  $\varepsilon$ , have one pivot ( $s^*$  or  $s^\circ$ ) for each, and two occurrences of  $\theta^\neq$ .

The following formula expresses the negation of the conditional independence atom  $\vec{\alpha} \perp_{\vec{\gamma}} \vec{\beta}$  and has length  $\mathcal{O}(n(n+m+k)|\vec{\alpha}\vec{\beta}\vec{\gamma}|)$ , where  $\vec{\alpha} = (\alpha_1, \dots, \alpha_n)$ ,  $\vec{\beta} = (\beta_1, \dots, \beta_m)$ , and  $\vec{\gamma} = (\gamma_1, \dots, \gamma_k)$ .

$$\varphi(\vec{\alpha}; \vec{\beta}; \vec{\gamma}) := \bigvee_{\substack{\delta \in \{\alpha_i, \neg\alpha_i \mid 1 \leq i \leq n\} \\ \varepsilon \in \{\beta_j, \neg\beta_j \mid 1 \leq j \leq m\}}} (\delta \vee \varepsilon) \vee \left( [(-\delta \wedge \theta^\neq(\vec{\alpha}\vec{\gamma}; \vec{\alpha}\vec{\gamma}; \varepsilon)) \vee (-\varepsilon \wedge \theta^\neq(\vec{\beta}; \vec{\beta}; \delta))] \wedge \mathbf{E}\delta \wedge \mathbf{E}\varepsilon \wedge ((\delta \vee \varepsilon) \leftrightarrow \mathbf{1}_{\vec{\gamma}}) \right)$$

**Lemma 4.6.**  $\sim \vec{\alpha} \perp_{\vec{\gamma}} \vec{\beta} \equiv \varphi(\vec{\alpha}; \vec{\beta}; \vec{\gamma})$ .

*Proof.* For the direction from left to right, assume  $T \not\models \vec{\alpha} \perp_{\vec{\gamma}} \vec{\beta}$ . Then there are  $s^*, s^\circ \in T$  such that  $s^*(\vec{\gamma}) = s^\circ(\vec{\gamma})$ , but for all  $s \in T$  it holds either  $s(\vec{\alpha}\vec{\gamma}) \neq s^*(\vec{\alpha}\vec{\gamma})$  or  $s(\vec{\beta}) \neq s^\circ(\vec{\beta})$ . In particular, there must be  $i, j$  such that  $s^*(\alpha_i) \neq s^\circ(\alpha_i)$  and  $s^*(\beta_j) \neq s^\circ(\beta_j)$ . Let  $\delta \in \{\alpha_i, \neg\alpha_i\}$  and  $\varepsilon \in \{\beta_j, \neg\beta_j\}$  such that  $s^*(\varepsilon) = s^\circ(\delta) = 1$  and  $s^*(\delta) = s^\circ(\varepsilon) = 0$ . In order to now satisfy the Boolean disjunct with index  $\delta, \varepsilon$ , we define subteams

$$S := \{s^*\} \cup \{s \in T \mid s(\delta) = s(\varepsilon) = 0, s(\vec{\alpha}\vec{\gamma}) \neq s^*(\vec{\alpha}\vec{\gamma})\}$$

$$U := \{s^\circ\} \cup \{s \in T \mid s(\delta) = s(\varepsilon) = 0, s(\vec{\beta}) \neq s^\circ(\vec{\beta})\}$$

of  $T$ . We show that the (in fact strict) split  $(T \setminus (S \cup U), S \cup U)$  satisfies the disjunction. First,  $T \setminus (S \cup U) \models \delta \vee \varepsilon$  due to the fact that  $T \setminus (S \cup U) \subseteq T_\delta \cup T_\varepsilon$ . Furthermore,  $S \cup U \models \mathbf{E}\delta \wedge \mathbf{E}\varepsilon \wedge (\delta \vee \varepsilon) \leftrightarrow \mathbf{1}_{\vec{\gamma}}$ , since  $(S \cup U)_{\delta \vee \varepsilon} = \{s^*, s^\circ\}$ . For the part in brackets, consider the (again strict) split  $(S, U \setminus S)$  of  $S \cup U$ . Again, clearly  $S \models \neg\delta$  and  $U \models \neg\varepsilon$ . Finally, both  $S \models \theta^\neq(\vec{\alpha}\vec{\gamma}; \vec{\alpha}\vec{\gamma}; \varepsilon)$  and  $U \models \theta^\neq(\vec{\beta}; \vec{\beta}; \delta)$  hold.

For the other direction, assume  $T \models \varphi(\vec{\alpha}; \vec{\beta}; \vec{\gamma})$  with the Boolean disjunction satisfied with indices  $\delta \in \{\alpha_i, \neg\alpha_i \mid 1 \leq i \leq n\}$  and  $\varepsilon \in \{\beta_j, \neg\beta_j \mid 1 \leq j \leq m\}$ . Then  $T$  can be divided into  $X \cup S \cup U$  where

- $X \models \delta \vee \varepsilon$ ,
- $S \models \neg\delta \wedge \theta^\neq(\vec{\alpha}\vec{\gamma}; \vec{\alpha}\vec{\gamma}; \varepsilon)$ ,

- $U \models \neg \varepsilon \wedge \theta^\neq(\vec{\beta}; \vec{\beta}; \delta)$  and
- $S \cup U \models E\delta \wedge E\varepsilon \wedge ((\delta \vee \varepsilon) \hookrightarrow \mathbf{1}_{\vec{\gamma}})$ .

By the final line, assignments  $s^* \in S_\varepsilon$  and  $s^\circ \in U_\delta$  exist. Now, for the sake of contradiction, suppose that  $T \models \vec{\alpha} \perp_{\vec{\gamma}} \vec{\beta}$ . As  $S_\varepsilon \cup U_\delta \models \mathbf{1}_{\vec{\gamma}}$  and hence  $s^*(\vec{\gamma}) = s^\circ(\vec{\gamma})$ , due to independence, another assignment  $s \in T$  must exist such that  $s(\vec{\alpha}\vec{\gamma}) = s^*(\vec{\alpha}\vec{\gamma})$  and  $s(\vec{\beta}) = s^\circ(\vec{\beta})$ .

However,  $s \notin X$ , since  $s(\vec{\alpha}) = s^*(\vec{\alpha})$  implies  $s \neq \delta$  and  $s(\vec{\beta}) = s^\circ(\vec{\beta})$  implies  $s \neq \varepsilon$ . Consequently,  $s \in S \cup U$ . For this reason, either  $s(\vec{\alpha}\vec{\gamma}) \neq s^*(\vec{\alpha}\vec{\gamma})$ , or  $s(\vec{\beta}) \neq s^\circ(\vec{\beta})$ , contradiction to  $s(\vec{\alpha}\vec{\gamma}) = s^*(\vec{\alpha}\vec{\gamma})$  and  $s(\vec{\beta}) = s^\circ(\vec{\beta})$ .  $\square$

**Anonymity atom.** Finally, the following formula expresses the negation of the unary anonymity atom  $\vec{\alpha}\Upsilon\beta$  and has length  $\mathcal{O}(n|\beta| + |\vec{\alpha}|)$ .

Roughly speaking, the anonymity atom  $\vec{\alpha}\Upsilon\beta$  is false if there is  $s^* \in T$  such that no  $s \in T$  with identical  $\vec{\alpha}$  but different  $\beta$  exists, or in other words, all  $s \in T$  with different  $\beta$  are also different in  $\vec{\alpha}$ . So we can directly let  $\gamma := \beta$  or  $\gamma := \neg\beta$ , pick  $s$  as pivot, and apply  $\theta^\neq$  to  $\alpha$ :

$$\varphi(\vec{\alpha}; \beta) := \bigvee_{\gamma \in \{\beta, \neg\beta\}} \left( \gamma \vee ((\gamma \hookrightarrow \mathbf{1}_{\vec{\alpha}}) \wedge \theta^\neq(\vec{\alpha}; \vec{\alpha}; \gamma)) \right)$$

**Lemma 4.7.**  $\sim \vec{\alpha}\Upsilon\beta \equiv \varphi(\vec{\alpha}; \beta)$ .

*Proof.* Suppose  $T \not\models \vec{\alpha}\Upsilon\beta$ . Then there is  $s^* \in T$  such that  $s(\vec{\alpha}) = s^*(\vec{\alpha})$  implies  $s(\beta) = s^*(\beta)$  for all  $s \in T$ . Let  $\gamma \in \{\beta, \neg\beta\}$  such that  $s^* \models \gamma$ , and consider the split  $(T \setminus S, S)$  of  $T$  defined by  $S := \{s^*\} \cup T_{\neg\gamma}$ . Then  $T \setminus S \models \gamma$ . Moreover,  $S \models \gamma \hookrightarrow \mathbf{1}_{\vec{\alpha}}$  and  $S \models \theta^\neq(\vec{\alpha}; \vec{\alpha}; \gamma)$ , since  $S_\gamma = \{s^*\}$  and  $s(\vec{\alpha}) \neq s^*(\vec{\alpha})$  for all  $s \in S_{\neg\gamma}$ .

For the other direction, suppose there is  $\gamma \in \{\beta, \neg\beta\}$  such that  $S \models \gamma$  and  $U \models (\gamma \hookrightarrow \mathbf{1}_{\vec{\alpha}}) \wedge \theta^\neq(\vec{\alpha}; \vec{\alpha}; \gamma)$  for some split  $(S, U)$  of  $T$ . Then there exists  $s^* \in U_\gamma$  such that  $s^*(\vec{\alpha}) \neq s(\vec{\alpha})$  for all  $s \in U_{\neg\gamma}$ . Clearly, now  $s^*(\beta) = s(\beta)$  for all  $s \in S \cup U_\gamma$ , so ultimately  $s^*(\beta) = s(\beta)$  or  $s^*(\vec{\alpha}) \neq s(\vec{\alpha})$  for all  $s \in T$ , hence  $T \not\models \vec{\alpha}\Upsilon\beta$ .  $\square$

In the first-order setting, Rönholm [Rö18, Remark 2.31] demonstrated that the general anonymity atom can be expressed via the unary anonymity atom and the splitting disjunction. In the lemma below, we show that this can also be done via strict splitting. This yields a formula expressing  $\vec{\alpha}\Upsilon\vec{\beta}$  of length  $\mathcal{O}(n|\vec{\beta}| + m|\vec{\alpha}|)$ .

**Lemma 4.8.** *The following formulas are equivalent:*

- (1)  $\vec{\alpha}\Upsilon\vec{\beta}$ ,
- (2)  $\bigvee_{i=1}^m \vec{\alpha}\Upsilon\beta_i$ ,
- (3)  $\bigvee_{i=1}^m \vec{\alpha}\Upsilon\beta_i$ .

*Proof.* For (2)  $\Rightarrow$  (1), we follow Rönholm [Rö18]. Suppose  $T \models \bigvee_{i=1}^m \vec{\alpha}\Upsilon\beta_i$  via the split of  $T$  into  $Y_1 \cup \dots \cup Y_m$ , where  $Y_i \models \vec{\alpha}\Upsilon\beta_i$ . To see that  $T \models \vec{\alpha}\Upsilon\vec{\beta}$ , let  $s \in T$  be arbitrary. For some  $i$ , now  $s \in Y_i$ . Consequently, there is  $s' \in Y_i$  such that  $s(\vec{\alpha}) = s'(\vec{\alpha})$  but  $s(\beta_i) \neq s'(\beta_i)$ . But as  $Y_i \subseteq T$  and  $s$  was arbitrary, (1) follows.

The step (3)  $\Rightarrow$  (2) is clear, since every strict split of a team is a split.

It remains to show (1)  $\Rightarrow$  (3). Here, we adapt the proof of Rönholm [Rö18] for  $\vec{\vee}$ . Suppose that  $T \models \vec{\alpha}\Upsilon\vec{\beta}$  holds. Define subteams  $Y_i$  of  $T$  by

$$Y_i := \{ s \in T \mid \exists s' \in T : s'(\vec{\alpha}) = s(\vec{\alpha}) \text{ but } s(\beta_i) \neq s'(\beta_i) \},$$

as in the proof of Rönnholm [Rö18], but additionally define teams  $Z_i := Y_i \setminus \bigcup_{j < i} Y_j$  for  $1 \leq i \leq m$ , where  $Y_0 := \emptyset$ . We show that  $Z_1 \cup \dots \cup Z_m$  forms a strict split of  $T$ . The sets  $Z_1, \dots, Z_m$  are pairwise disjoint, as  $Z_i \subseteq Y_i$  but  $Z_j \cap Y_i = \emptyset$  when  $i < j$ . Next, let  $s \in T$  be arbitrary. Define

$$I := \{i \in \{1, \dots, m\} \mid \exists s' \in T : s(\vec{\alpha}) = s'(\vec{\alpha}) \text{ but } s(\beta_i) \neq s'(\beta_i)\}.$$

By assumption (1),  $I$  is non-empty and hence contains a minimal element  $i$ . But then  $s \in Y_i \setminus \bigcup_{j < i} Y_j = Z_i$ . Consequently,  $T = \bigcup_{i=1}^m Z_i$ .

Finally, we need to show that  $Z_i \models \vec{\alpha} \Upsilon \beta_i$ . For this, let now  $s \in Z_i$  be arbitrary. By definition of  $Z_i$ , there exists  $s' \in T$  with  $s(\vec{\alpha}) = s'(\vec{\alpha})$  and  $s(\beta_i) \neq s'(\beta_i)$ . It suffices to show that  $s' \in Z_i = Y_i \setminus \bigcup_{j < i} Y_j$ . As  $s' \in Y_i$  follows from the definition of  $Y_i$ , assume  $s' \in Y_j$  for some  $j < i$ . Then by symmetry also  $s \in Y_j$ , contradiction to  $s \in Z_i$ . Hence  $s' \notin Y_j$  for all  $j < i$ , so  $s' \in Z_i$ .  $\square$

With the negations of dependency atoms definable in  $\text{PL}(\{\emptyset, \wedge, \vee\})$ , it is an easy corollary that the atoms themselves are definable when additionally the strong negation  $\sim$  is available. In the next theorem, we prove this, generalize the part on the anonymity atom  $\Upsilon$ , and furthermore expand the results to also work with  $\dot{\vee}$ , which we previously considered only for the downward closed atoms  $=(\cdot, \cdot)$  and  $\mid$  in Theorem 4.1.

**Theorem 4.9.** *Let  $\Sigma = \{\sim, \wedge, \vee\}$  or  $\Sigma = \{\sim, \wedge, \dot{\vee}\}$ . Let  $\vec{\alpha} = (\alpha_1, \dots, \alpha_n)$ ,  $\vec{\beta} = (\beta_1, \dots, \beta_m)$ , and  $\vec{\gamma} = (\gamma_1, \dots, \gamma_k)$  be tuples of purely propositional formulas.*

- *The dependence atom  $=(\vec{\alpha}; \vec{\beta})$  is equivalent to a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(|\vec{\alpha}\vec{\beta}|)$ .*
- *The exclusion atom  $\vec{\alpha} \mid \vec{\beta}$  is equivalent to a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(n|\vec{\alpha}\vec{\beta}|)$ .*
- *The inclusion atom  $\vec{\alpha} \subseteq \vec{\beta}$  is equivalent to a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(n|\vec{\alpha}\vec{\beta}|)$ .*
- *The conditional independence atom  $\vec{\alpha} \perp_{\vec{\gamma}} \vec{\beta}$  is equivalent to a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(n(n+m+k)|\vec{\alpha}\vec{\beta}\vec{\gamma}|)$ .*
- *The anonymity atom  $\vec{\alpha} \Upsilon \vec{\beta}$  is equivalent to a  $\text{PL}(\Sigma)$ -formula of length  $\mathcal{O}(n|\vec{\beta}| + m|\vec{\alpha}|)$ .*

*Furthermore, all these formulas are logspace-computable.*

*Proof.* We essentially take the formulas of Theorem 4.1 (and Lemma 4.8 for the anonymity atom) and add a Boolean negation in front of them. For  $\Sigma = \{\sim, \wedge, \vee\}$ , the only remaining thing to do is to rewrite  $\otimes$  via  $\wedge$  and  $\sim$ .

For  $\Sigma = \{\sim, \wedge, \dot{\vee}\}$ , we must also remove all occurrences of  $\vee$  and use only  $\dot{\vee}$ . We see that this comes down to expressing the subformulas  $\theta^=$  and  $\theta^{\neq}$  in  $\{\sim, \wedge, \dot{\vee}\}$ . In  $\theta^=$ , the lax splitting  $\vee$  can equivalently be replaced by  $\dot{\vee}$  due to Proposition 2.6, as any occurrence of  $\vee$  has at least one purely propositional argument. The same does not hold for  $\theta^{\neq}$ , but it is easy to see that  $\theta^{\neq}(\vec{\alpha}; \vec{\beta}; \gamma)$  can be replaced by

$$\gamma \otimes \sim(\top \dot{\vee} (\mathbf{E}\gamma \wedge \mathbf{E}\neg\gamma \wedge \theta^=(\vec{\alpha}; \vec{\beta}; \gamma))). \quad \square$$

**4.2. Upper bounds for parity.** Next, we again consider the parity of the cardinality of teams, i.e., is there a formula that is true precisely on teams with even cardinality? This differs from the other considered team properties in that both the property and its negation have exponential lower bounds in  $\text{PL}(\{\emptyset, \wedge, \vee, \dot{\vee}\})$  (see Theorem 3.4). Nevertheless, we show that it is polynomially definable when linearly many negations are nested inside the formula, which was not necessary for the results of Theorem 4.9.

**Theorem 4.10.** *Let  $|\Phi| = n$ . The class of  $\Phi$ -teams of odd cardinality is defined by a  $\text{PL}(\wedge, \sim, \dot{\vee})$ -formula of length  $\mathcal{O}(n^2)$ .*

We write  $\text{=}(X)$ , for a finite set  $X \subseteq \Phi$  of propositions, as abbreviation for  $\bigwedge_{p \in X} \text{=}(p)$ . Based on this, the formula  $\mathbf{1} := \sim \perp \wedge \text{=}(\Phi)$  defines singletons, that is, a  $\Phi$ -team  $T$  satisfies  $\mathbf{1}$  iff  $|T| = 1$ . The formula expressing odd cardinality is now recursively defined as follows:

$$\varphi() := \mathbf{1}$$

$$\varphi(p\vec{q}) := \mathbf{1} \dot{\vee} \sim \left( \left[ \mathbf{1} \otimes (\sim \text{=}(p) \wedge (\mathbf{1} \dot{\vee} \text{=}(p))) \right] \dot{\vee} [\text{=}(p) \wedge \sim \varphi(\vec{q})] \right)$$

We prove its correctness in the lemma below. The rough idea is that a team is even precisely if  $T_p$  and  $T_{\neg p}$  are either both even or both odd, regardless of which proposition  $p$  is.

**Lemma 4.11.** *Let  $T \in \text{Tms}(\Phi)$  and let  $\vec{q}$  list all propositions in  $\Phi$ . Then  $T \models \varphi(\vec{q})$  if and only if  $|T|$  is odd.*

*Proof.* The proof is by induction on  $|\vec{q}|$ . Since the domain of  $T$  exceeds the arguments of the recursive subformulas  $\varphi$ , we prove the following stronger statement. Let  $\vec{q} = (q_1, \dots, q_m)$ . Then, for any  $\Phi$ -team  $S$  satisfying  $\text{=}(\Phi \setminus \{q_1, \dots, q_m\})$ , it holds that that  $S \models \varphi(\vec{q})$  if and only if  $|S|$  is odd. The base case is clear as the only  $\emptyset$ -teams are  $\emptyset$  and  $\{\emptyset\}$ .

We proceed with the inductive step, and first provide some intuition. The crucial subformula is

$$\psi := \left[ \mathbf{1} \otimes (\sim \text{=}(p) \wedge (\mathbf{1} \dot{\vee} \text{=}(p))) \right] \dot{\vee} [\text{=}(p) \wedge \sim \varphi(\vec{q})].$$

We will show below that it is true iff at least one of  $|S_p|$  and  $|S_{\neg p}|$  is odd. Then  $\sim \psi$  means that both  $|S_p|$  and  $|S_{\neg p}|$  are even. This is sufficient for  $|S|$  to be even but of course not necessary. However, the following holds:  $|S|$  is odd precisely when we can remove one assignment  $S$  such that afterwards both  $|S_p|$  and  $|S_{\neg p}|$  are even. Hence, oddness is defined by  $\mathbf{1} \dot{\vee} \sim \psi$ .

Intuitively,  $\psi$  allows to split off an even subteam of either  $S_p$  or  $S_{\neg p}$  by  $\dots \dot{\vee} (\text{=}(p) \wedge \sim \varphi(\vec{q}))$ , reducing either  $S_p$  or  $S_{\neg p}$ , depending on which is odd, to a singleton. Afterwards the team then satisfies  $\mathbf{1} \otimes \sim \text{=}(p) \wedge (\mathbf{1} \dot{\vee} \text{=}(p))$ . We prove this formally, i.e., that  $S \models \psi$  iff  $|S_p|$  or  $|S_{\neg p}|$  is odd.

“ $\Rightarrow$ ” Suppose  $S \models \psi$  via the strict split  $(U, V)$  such that  $V \models \text{=}(p) \wedge \sim \varphi(\vec{q})$ , and either  $U \models \mathbf{1}$  or  $U \models \sim \text{=}(p) \wedge (\mathbf{1} \dot{\vee} \text{=}(p))$ . Note that  $|V|$  is even by induction hypothesis. We distinguish the two possible cases for  $U$ .

- $U \models \mathbf{1}$ : Then  $U, V \models \text{=}(p)$ . Additionally, Both  $U$  and  $U \cup V$  have odd size, and one of them equals  $S_p$  or  $S_{\neg p}$ , depending on whether  $U$  and  $V$  agree on  $p$  or not.
- $U \models \sim \text{=}(p) \wedge (\mathbf{1} \dot{\vee} \text{=}(p))$ : Due to symmetry, we can assume  $V \subsetneq S_p$  and  $S_{\neg p} \subsetneq U$ . By the formula,  $U$  has a strict split  $(X, Y)$  such that  $|X| = 1$  and  $Y \models \text{=}(p)$ . Let  $Z = S_p \setminus V$ . Either  $Z \subseteq X$  or  $Z \subseteq Y$ , as  $X$  and  $Y$  do not agree on  $p$ , but each is constant in  $p$ . If  $Z \subseteq X$ , then  $Z = X$  and  $|V \cup X| = |S_p|$  is odd and we are done. If  $Z \subseteq Y$ , then  $S_{\neg p} \subseteq X$ , hence  $S_{\neg p} = X$  and  $|S_{\neg p}|$  is odd.

“ $\Leftarrow$ ” W.l.o.g.  $|S_p|$  is odd. Pick  $s \in S_p$  arbitrarily and consider the split  $(S_{\neg p} \cup \{s\}, S_p \setminus \{s\})$  of  $S$ . For the second component,  $S_p \setminus \{s\} \models \text{=}(p) \wedge \sim \varphi(\vec{q})$  by induction hypothesis. For the first component, either  $S_{\neg p}$  is empty and  $S_{\neg p} \cup \{s\} \models \mathbf{1}$ , or  $S_{\neg p}$  is non-empty and  $S_{\neg p} \cup \{s\} \models \sim \text{=}(p) \wedge (\mathbf{1} \dot{\vee} \text{=}(p))$ . In both cases,  $S \models \psi$ .  $\square$

We have shown an exponential lower bound for parity in the existential fragment. For the matching upper bound, the following formulas define parity by mutual recursion:

$$\begin{aligned}\varphi^{\text{even}}() &:= \perp \\ \varphi^{\text{odd}}() &:= \text{NE} \\ \varphi^{\text{even}}(p\vec{q}) &:= ((p \wedge \varphi^{\text{odd}}(\vec{q})) \vee (\neg p \wedge \varphi^{\text{odd}}(\vec{q}))) \otimes ((p \wedge \varphi^{\text{even}}(\vec{q})) \vee (\neg p \wedge \varphi^{\text{even}}(\vec{q}))) \\ \varphi^{\text{odd}}(p\vec{q}) &:= ((p \wedge \varphi^{\text{odd}}(\vec{q})) \vee (\neg p \wedge \varphi^{\text{even}}(\vec{q}))) \otimes ((p \wedge \varphi^{\text{even}}(\vec{q})) \vee (\neg p \wedge \varphi^{\text{odd}}(\vec{q})))\end{aligned}$$

**Theorem 4.12.** *Let  $|\Phi| = n$ . If  $\Sigma = \{\wedge, \otimes, \vee\}$  or  $\Sigma = \{\wedge, \otimes, \dot{\vee}\}$ , then the class of  $\Phi$ -teams of odd resp. even cardinality is definable by a  $\text{PL}(\Sigma)$ -formula of length  $2^{\mathcal{O}(n)}$ .*

*Proof.* First of all, observe that the formula  $(p \wedge \varphi) \vee (\neg p \wedge \varphi')$  is equivalent to  $(p \wedge \varphi) \dot{\vee} (\neg p \wedge \varphi')$  for all  $\varphi, \varphi'$  and propositions  $p$ , since any split satisfying the former formula is necessarily strict. As a consequence, it suffices to consider  $\Sigma = \{\wedge, \otimes, \dot{\vee}\}$ .

Let  $\Phi = \{p_1, \dots, p_n\}$ , and  $T$  a  $\Phi$ -team. Let  $\vec{p} = (p_1, \dots, p_n)$  list all variables in  $\Phi$ . We prove by induction on  $n$  that  $T \models \varphi^{\text{even}}(\vec{p})$  iff  $|T|$  is even, and  $T \models \varphi^{\text{odd}}(\vec{p})$  iff  $|T|$  is odd.

First, if  $\Phi = \emptyset$ , then either  $T = \emptyset$  and  $T \models \perp = \varphi^{\text{even}}()$ , or  $T = \{\emptyset\}$  and  $T \models \text{NE} = \varphi^{\text{odd}}()$ . For the inductive step, observe that  $|T|$  is even iff  $|T_p|$  and  $|T_{\neg p}|$  have equal parity, and is odd iff they have different parity, where  $p \in \Phi$  is an arbitrary proposition. Furthermore,  $T_p$  and  $T_p \upharpoonright (\Phi \setminus \{p\})$  have the same cardinality (the same goes for  $T_{\neg p}$ ). Additionally,  $T_p$  and  $T_p \upharpoonright (\Phi \setminus \{p\})$  satisfy the same  $\text{PL}(\Phi \setminus \{p\}, \Sigma)$ -formulas by Proposition 2.10. Hence the equivalence immediately follows by induction hypothesis.  $\square$

**4.3. Modal team logic.** In this final section, we consider *modal team logic* MTL, introduced by Müller [Mül14], which extends both classical modal logic ML and propositional team logic  $\text{PL}(\{\wedge, \sim, \vee\})$ . Beginning with modal dependence logic by Väänänen [Vä08], several atoms of dependency have been transferred from the first-order setting also to the modal setting (cf. [EHM<sup>+</sup>13, KMSV17, HS15]). Using the results of this paper, we show that the computational complexity of MTL does not change if it is augmented with any of the dependency atoms we considered before.

For each  $k \geq 0$ , we define the function  $\text{exp}_k$  as  $\text{exp}_0(n) := n$  and  $\text{exp}_{k+1}(n) := 2^{\text{exp}_k(n)}$ .

For  $k \geq 0$ ,  $\text{ATIME-ALT}(\text{exp}_k, \text{poly})$  is the class of problems decidable by an alternating Turing machine (see [CKS81]) with at most  $p(n)$  alternations and runtime at most  $\text{exp}_k(p(n))$ , for a polynomial  $p$ . Likewise,  $\text{TOWER}(\text{poly})$  is the class of problems that are decidable by a deterministic Turing machine in time  $\text{exp}_{p(n)}(1)$  for some polynomial  $p$ .

The syntax of MTL is given by the following grammar, where  $p$  is an atomic proposition:

$$\varphi ::= \top \mid \perp \mid p \mid \neg p \mid \sim \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \square \varphi \mid \diamond \varphi,$$

Observe that classical modal logic ML is the  $\sim$ -free fragment of MTL. Let  $\text{md}(\varphi)$  denote the *modal depth* of  $\varphi$ , i.e., the nesting depth of  $\diamond$  and  $\square$  inside  $\varphi$ . A *Kripke structure* over  $\Phi$ , where  $\Phi$  is a set of propositions, is a tuple  $K = (W, R, V)$  where  $(W, R)$  is a directed graph and  $V: \Phi \rightarrow 2^W$ . A *team* in  $K$  is a subset of  $W$ . Let  $RT := \{v \mid (w, v) \in R, w \in T\}$  and  $R^{-1}T := \{w \mid (w, v) \in R, v \in T\}$ . The set  $\text{Prop}(\varphi)$  is defined as for propositional logic.

MTL-formulas  $\varphi$  are evaluated as follows on pairs  $(K, T)$ , where  $K$  is a Kripke structure over some set  $\Phi' \supseteq \text{Prop}(\varphi)$  of propositions and  $T$  is a team in  $K$ :

$$\begin{aligned} (K, T) \models p &\Leftrightarrow T \subseteq V(p) \text{ for } p \in \Phi, \\ (K, T) \models \neg p &\Leftrightarrow T \cap V(p) = \emptyset \text{ for } p \in \Phi, \\ (K, T) \models \diamond\psi &\Leftrightarrow \exists T' \subseteq RT : T \subseteq R^{-1}T' \text{ and } (K, T') \models \psi \\ (K, T) \models \Box\psi &\Leftrightarrow (K, RT) \models \psi, \end{aligned}$$

with  $\wedge, \sim, \top$  and  $\perp$  analogously to propositional logic. An MTL-formula  $\varphi$  is *satisfiable* (*valid*) if  $(K, T) \models \varphi$  for some (every) Kripke structure  $K$  over  $\text{Prop}(\varphi)$  and team  $T$  in  $K$ . The *model checking problem* is, given  $\varphi \in \text{MTL}$  and a Kripke structure with team  $(K, T)$ , to decide whether  $(K, T) \models \varphi$ .

The modal atoms of dependence  $=(\vec{\alpha}; \vec{\beta})$ , independence  $\vec{\alpha} \perp_{\vec{\beta}} \vec{\gamma}$ , inclusion  $\vec{\alpha} \subseteq \vec{\beta}$ , exclusion  $\vec{\alpha} \mid \vec{\beta}$ , and anonymity  $\vec{\alpha} \Upsilon \vec{\beta}$ , are defined completely analogous as the propositional variants (cf. p. 7), but with  $\vec{\alpha}, \vec{\beta}, \vec{\gamma}$  being tuples of ML-formulas instead of PL-formulas.

**Theorem 4.13.** *For MTL extended by the atoms  $=(\cdot, \cdot), \perp_c, \subseteq, \mid$ , and  $\Upsilon$ ,*

- *satisfiability and validity is TOWER(poly)-complete,*
  - *satisfiability and validity for modal depth at most  $k$  is ATIME-ALT( $\text{exp}_k, \text{poly}$ )-complete,*
  - *model checking is PSPACE-complete,*
- with respect to logspace-reductions.*

*Proof.* For the logic without any atoms, the complexity was shown by Müller [Mül14] and Lück [Lül18]. The upper bounds of Theorem 4.1 immediately carry over to MTL, so we can substitute every such atom by a polynomially long equivalent MTL-formula.  $\square$

## 5. CONCLUSION

In this paper, we classified common atoms of dependency with respect to their succinctness in various fragments of propositional team logic. We showed that the negations of these atoms all can be polynomially expressed in the positive fragment of propositional team logic, while the atoms themselves can only be expressed in this fragment in formulas of exponential size. This implies polynomial upper bounds for the atoms in full propositional team logic with unrestricted contradictory negation. For the lower bounds, we adapted formula size games to the team semantics setting, and refined the approach with the notion of upper dimension.

In further research, comparing the atoms of dependency in terms of succinctness could be interesting. For example, do the lower bounds for the inclusion atoms still hold if we consider the positive fragment together with dependence atoms? Adding moves corresponding to atoms of dependency to the formula size game would enable looking into the relative succinctness.



## ACKNOWLEDGMENT

The authors want to thank Lauri Hella and Raine Rönholm for pointing out references essential for this work, and Juha Kontinen for being the initiating force behind it. We also thank the anonymous referees for their very detailed comments and suggestions on how to simplify several proofs. This work was supported in part by a joint grant of the DAAD (57348395) and the Academy of Finland (308099).

## REFERENCES

- [AI03] Micah Adler and Neil Immerman. An  $n!$  lower bound on formula size. *ACM Trans. Comput. Log.*, 4(3):296–314, 2003.
- [CKS81] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, January 1981.
- [EHM<sup>+</sup>13] Johannes Ebbing, Lauri Hella, Arne Meier, Julian-Steffen Müller, Jonni Virtema, and Heribert Vollmer. Extended modal dependence logic. In *WoLLIC*, volume 8071 of *Lecture Notes in Computer Science*, pages 126–137. Springer, 2013.
- [EVW02] Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. *Inf. Comput.*, 179(2):279–295, 2002.
- [FvdHIK11] Tim French, Wiebe van der Hoek, Petar Iliev, and Barteld P. Kooi. Succinctness of epistemic languages. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 881–886, 2011.
- [FvdHIK13] Tim French, Wiebe van der Hoek, Petar Iliev, and Barteld P. Kooi. On the succinctness of some modal logics. *Artif. Intell.*, 197:56–85, 2013.
- [Gal12] Pietro Galliani. Inclusion and exclusion dependencies in team semantics — On some logics of imperfect information. *Annals of Pure and Applied Logic*, 163(1):68–84, January 2012.
- [Gal15] Pietro Galliani. Upwards closed dependencies in team semantics. *Information and Computation*, 245:124–135, 2015.
- [GKPS95] Goran Gogic, Henry Kautz, Christos Papadimitriou, and Bart Selman. The comparative linguistics of knowledge representation. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'95*, pages 862–869, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [GS05] Martin Grohe and Nicole Schweikardt. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science*, 1(1), 2005.
- [GV13] Erich Grädel and Jouko Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013.
- [HKL16] Miika Hannula, Juha Kontinen, Martin Lück, and Jonni Virtema. On quantified propositional logics and the exponential time hierarchy. In *GandALF*, volume 226 of *EPTCS*, pages 198–212, 2016.
- [HKVV18] Miika Hannula, Juha Kontinen, Jonni Virtema, and Heribert Vollmer. Complexity of propositional logics in team semantic. *ACM Trans. Comput. Log.*, 19(1):2:1–2:14, 2018.
- [HLSV14] Lauri Hella, Kerkko Luosto, Katsuhiko Sano, and Jonni Virtema. The expressive power of modal dependence logic. In *Advances in Modal Logic 10, invited and contributed papers from the tenth conference on “Advances in Modal Logic”, 2014*, pages 294–312, 2014.
- [Hod97] Wilfrid Hodges. Compositional semantics for a language of imperfect information. *Logic Journal of the IGPL*, 5(4):539–563, 1997.
- [HS89] Jaakko Hintikka and Gabriel Sandu. Informational Independence as a Semantical Phenomenon. In *Studies in Logic and the Foundations of Mathematics*, volume 126, pages 571–589. Elsevier, 1989.
- [HS15] Lauri Hella and Johanna Stumpf. The expressive power of modal logic with inclusion atoms. In *GandALF*, volume 193 of *EPTCS*, pages 129–143, 2015.
- [HV15] Lauri Hella and Jouko Väänänen. The size of a formula as a measure of complexity. In *Logic Without Borders - Essays on Set Theory, Model Theory, Philosophical Logic and Philosophy of Mathematics*, pages 193–214. De Gruyter, 2015.

- [HV16] Lauri Hella and Mikka Vilander. The succinctness of first-order logic over modal logic via a formula size game. In *Advances in Modal Logic 11, Budapest, Hungary, August 30 - September 2, 2016*, pages 401–419, 2016.
- [KMSV15] Juha Kontinen, Julian-Steffen Müller, Henning Schnoor, and Heribert Vollmer. A Van Benthem Theorem for Modal Team Semantics. In *CSL*, volume 41 of *LIPICs*, pages 277–291, 2015.
- [KMSV17] Juha Kontinen, Julian-Steffen Müller, Henning Schnoor, and Heribert Vollmer. Modal independence logic. *J. Log. Comput.*, 27(5):1333–1352, 2017.
- [KMV15] Andreas Krebs, Arne Meier, and Jonni Virtema. A team based variant of CTL. In *TIME*, pages 140–149. IEEE Computer Society, 2015.
- [KMVZ18] Andreas Krebs, Arne Meier, Jonni Virtema, and Martin Zimmermann. Team semantics for the specification and verification of hyperproperties. In *MFCS*, volume 117 of *LIPICs*, pages 10:1–10:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
- [KN09] Juha Kontinen and Ville Nurmi. Team Logic and Second-Order Logic. In *Logic, Language, Information and Computation*, volume 5514, pages 230–241. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [LSW01] Carsten Lutz, Ulrike Sattler, and Frank Wolter. Modal logic and the two-variable fragment. In *Computer Science Logic, Paris, France, September 10-13, 2001*, pages 247–261, 2001.
- [Lü18] Martin Lück. Canonical models and the complexity of modal team logic. In *CSL*, volume 119 of *LIPICs*, pages 30:1–30:23. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
- [Mar03] Nicolas Markey. Temporal logic with past is exponentially more succinct, concurrency cullm. *Bulletin of the EATCS*, 79:122–128, 2003.
- [Mül14] Julian-Steffen Müller. *Satisfiability and model checking in team based logics*. PhD thesis, University of Hanover, 2014.
- [Rö18] Raine Rönholm. *Arity Fragments of Logics with Team Semantics*. PhD thesis, Tampere University, Tampere University Press, 2018.
- [vDFvdHI14] Hans van Ditmarsch, Jie Fan, Wiebe van der Hoek, and Petar Iliev. Some exponential lower bounds on formula-size in modal logic. In *Advances in Modal Logic 10, held in Groningen, The Netherlands, August 5-8, 2014*, pages 139–157, 2014.
- [vdHI14] Wiebe van der Hoek and Petar Iliev. On the relative succinctness of modal logics with union, intersection and quantification. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 341–348, 2014.
- [vdHIK12] Wiebe van der Hoek, Petar Iliev, and Barteld P. Kooi. On the relative succinctness of two extensions by definitions of multimodal logic. In *How the World Computes - Turing Centenary Conference and 8th Conference on Computability in Europe, CiE 2012, Cambridge, UK, June 18-23, 2012. Proceedings*, pages 323–333, 2012.
- [Vir17] Jonni Virtema. Complexity of validity for propositional dependence logics. *Inf. Comput.*, 253:224–236, 2017.
- [Vä07] Jouko Väänänen. *Dependence logic: A New Approach to Independence Friendly Logic*. Number 70 in London Mathematical Society student texts. Cambridge University Press, Cambridge ; New York, 2007.
- [Vä08] Jouko Väänänen. Modal dependence logic. *New perspectives on games and interaction*, 4:237–254, 2008.
- [Vä19] Jouko Väänänen. An atom’s worth of anonymity. Dagstuhl Seminar 19031 “Logics for Dependence and Independence”, 2019.
- [Wil99] Thomas Wilke. CTL<sup>+</sup> is exponentially more succinct than CTL. In *Foundations of Software Technology and Theoretical Computer Science, 19th Conference, Chennai, India, December 13-15, 1999*, pages 110–121, 1999.
- [Yan14] Fan Yang. *On extensions and variants of dependence logic*. PhD thesis, University of Helsinki, 2014.
- [YV16] Fan Yang and Jouko Väänänen. Propositional logics of dependence. *Ann. Pure Appl. Logic*, 167(7):557–589, 2016.
- [YV17] Fan Yang and Jouko Väänänen. Propositional team logics. *Ann. Pure Appl. Logic*, 168(7):1406–1441, 2017.

# PUBLICATION

## III

### **Games for Succinctness of Regular Expressions**

Miikka Vilander

In: *Proceedings 12th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2021, Padua, Italy, 20-22 September 2021*. Ed. by Pierre Ganty and Davide Bresolin. Vol. 346EPTCS. 2021, pp. 258–272

DOI: 10.4204/EPTCS.346.17

**Publication reprinted with the permission of the copyright holders.**



# Games for Succinctness of Regular Expressions

Miikka Vilander

Computing Sciences  
Tampere University  
Tampere, Finland

miikka.vilander@tuni.fi

We present a version of so called formula size games for regular expressions. These games characterize the equivalence of languages up to expressions of a given size. We use the regular expression size game to give a simple proof of a known non-elementary succinctness gap between first-order logic and regular expressions. We also use the game to only count the number of stars in an expression instead of the overall size. For regular expressions this measure trivially gives a hierarchy in terms of expressive power. We obtain such a hierarchy also for what we call RE over star-free expressions, where star-free expressions, that is ones with complement but no stars, are combined using the operations of regular expressions.

## 1 Introduction

Even though regular expressions, abbreviated RE, are a very thoroughly studied topic in computer science, little work has been done on their succinctness, or size, until recently. The pioneering paper on the size of RE seems to be in 1974 by Ehrenfeucht and Zeiger [4]. They define the size of an RE as the number of occurrences of alphabet symbols in it and show that there is a deterministic finite automata with  $n$  states such that the smallest RE defining the same language has size  $2^{n-1}$ . In 2005, Ellul et al. [5] noted the lack of work on succinctness and presented several open problems as well as some results of their own. Some of these open problems were related to the succinctness of RE expanded with operations such as intersection. These and other similar problems were independently solved by Gelade and Neven [6, 7] on the one hand and Gruber and Holzer [8, 9] on the other.

Gelade and Neven use a generalization of the result of Ehrenfeucht and Zeiger [4] to obtain double exponential lower bounds for the size of an RE defining the complement of a single RE or the intersection of a finite number of RE in a fixed size alphabet [7]. Gelade uses the same technique to also obtain double exponential lower bounds for the added operations of interleaving and counting [6]. Gruber and Holzer go even further, obtaining tighter bounds for all of the above in a two-letter alphabet [8, 9]. They link the size of RE to their star height via a measure on the connectivity of the underlying DFA. The measure is called cycle rank and was first introduced by Eggan and Büchi [3]. These two groups worked independently although they were clearly aware of the other group's work.

Many problems in finite model theory have been solved via the use of games such as the famous Ehrenfeucht-Fraïssé game that characterizes quantifier rank or depth in first-order logic. A similar game for RE was presented by Yan [15]. This so called split game characterizes the depth of both catenation and stars for generalized regular expressions, or GRE, where complement is added as an operation. Catenation depth is sometimes referred to as dot-depth and star depth is more commonly known as star height. For RE, Hashiguchi famously proved that star height gives a full hierarchy in terms of expressive power [10]. For GRE, it is notoriously not even known if a language that requires an expression of star height two exists. Yan offers his game as a possible way to attack the generalized star height problem but is only able to complete results on infinite  $\omega$ -words.

In the vein of EF-games, there are also games for succinctness. These are often called formula size games. They are games of definability just as the EF-game, but instead of quantifier rank they measure the size of the defining formula. To our knowledge, the earliest example of such a game is for propositional logic by Razborov [13]. Perhaps more well known is the later game by Adler and Immerman [1] for a modal logic called CTL. To our knowledge, ours are the first formula size games presented for regular expressions.

While EF-games are played on two structures, formula size games are instead played on two sets of structures,  $A$  and  $B$ . In the context of regular expressions, these sets are languages. Our version of the games also has a resource parameter  $k$ . The first player  $S$  is trying to show that there is an expression  $R$  with  $A \subseteq L(R)$ ,  $B \subseteq \Sigma^* \setminus L(R)$  and size at most  $k$ .  $S$  essentially sketches the syntax tree of such a separating expression as the game goes on, but in a single game only one branch of the tree is visited. It is the role of the second player  $D$  to choose which branch this is, and try to find the error in the strategy of  $S$ . A separating expression of appropriate size exists if and only if  $S$  has a winning strategy. In addition to the size, in this paper we are also interested in the number of stars in an expression. Thus we add a separate parameter  $s$  to the game to track this. The game is very easy to modify in this way to track the number or depth of whatever operators one is interested in.

We use the RE-version of the game to give a simpler proof for a known non-elementary succinctness gap between FO and RE. Stockmeyer [14] showed that star-free expressions are non-elementarily more succinct than RE and together with an elementary translation from FO to star-free by McNaughton and Papert [12], the result follows. In addition, we consider the number of stars in an expression as a measure of complexity. For RE a hierarchy in terms of expressive power can be trivially obtained in star height one. For GRE this presents a difficult problem as the full use of complement ramps up the complexity of the game significantly. We present RE over star-free expressions as a natural middle ground between RE and GRE. These include all star-free expressions with complement and their combinations using the operations of RE. For RE over star-free expressions we use a corresponding version of the game to show that the number of stars also gives a full hierarchy in terms of expressive power already in star height one.

The outline of the paper is as follows. In Section 2 we introduce RE, GRE and RE over star-free expressions. We also discuss our definition of size for these expressions and define some notation for the rest of the paper. In Section 3 we present the GRE size game and prove that it works as intended. We also present variations of the game for RE and RE over star-free, and prove some useful lemmas for later. In Section 4 we use the game for RE to show that defining a large finite language requires a large RE. We then define a finite language of non-elementary size via a FO-formula of exponential size, thus reproving the succinctness gap between FO and RE. In Section 5 we show that the number of stars in an expression gives a hierarchy in terms of expressive power for RE over star-free expressions. We conclude in Section 6.

## 2 Preliminaries

We begin by defining some basic notions such as regular expressions and our concept of the size of a regular expression. For more on regular expressions we refer the reader to [11]. We omit the syntax and semantics of first-order logic and direct the reader to [2] for a textbook with a finite model theory approach.

Let  $\Sigma$  be an alphabet. Strings of symbols from the alphabet are called *words* and sets of words are called *languages*. We denote the length of a word  $w$  with  $|w|$ .

The *regular expressions*, or RE, of  $\Sigma$  are defined recursively as follows:  $\emptyset$ ,  $\varepsilon$  and every  $a \in \Sigma$  are regular expressions. If  $R_1$  and  $R_2$  are regular expressions, then also  $R_1 \cup R_2$ ,  $R_1 R_2$  and  $R_1^*$  are regular expressions. The *generalized regular expressions*, or GRE, of  $\Sigma$  are defined in the same way with the following addition: if  $R$  is a GRE, then  $\neg R$  is also a GRE. Sometimes GRE are also defined to include a separate intersection operation. As the effect on succinctness is negligible, we define intersection as the shorthand  $R_1 \cap R_2 := \neg(\neg R_1 \cup \neg R_2)$  to keep the number of moves in our game smaller.

The *language of a regular expression*  $R$ , denoted by  $L(R)$  is defined as follows:

- $L(\emptyset) = \emptyset$ ,
- $L(\varepsilon) = \{\varepsilon\}$  (the empty word),
- $L(a) = \{a\}$  for  $a \in \Sigma$ ,
- $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$ ,
- $L(R_1 R_2) = L(R_1)L(R_2) = \{uv \mid u \in L(R_1), v \in L(R_2)\}$  and
- $L(R_1^*) = L(R_1)^* = \{w_1 \cdots w_n \mid n \in \mathbb{N}, w_i \in L(R_1) \text{ for each } i \in \mathbb{N}\}$ .

For generalized regular expressions, additionally  $L(\neg R_1) = \Sigma^* \setminus L(R_1)$ .

We will also refer to *star-free expressions*. These are generalized regular expressions with the  $*$ -rule removed. A classical result by McNaughton and Papert [12] states that star-free expressions have the same expressive power over words as first-order logic. Note that this means many languages naturally expressed by a RE with stars are also expressible by star-free expressions. For example, if  $\Sigma = \{a, b\}$ , then  $L((ab)^*) = L(\varepsilon \cup (a-\emptyset \cap \neg \emptyset b \cap \neg(\neg \emptyset a a - \emptyset) \cap \neg(\neg \emptyset b b - \emptyset)))$ .

Finally we present a middle ground between RE and GRE we call *RE over star-free expressions*. These expressions are defined by  $R$  in the following grammar (we omit parentheses for simplicity):

$$\begin{aligned} R &::= R \cup R \mid RR \mid R^* \mid S \\ S &::= S \cup S \mid SS \mid \neg S \mid \emptyset \mid \varepsilon \mid a \text{ for every } a \in \Sigma \end{aligned}$$

As the name suggests, RE over star-free expressions include all star-free expressions in the sense of GRE and can combine them using only the operations of RE. Essentially this means that stars cannot occur inside a complement. Since star-free expressions correspond to FO-definable properties of words, we feel this is a natural variation of RE to consider in terms of succinctness. It is quite possible someone else has already presented it but we could not find it in the literature.

There are several ways one could define the size of a regular expression. Gruber and Holzer [8] use alphabetic width defined as the number of occurrences of symbols from  $\Sigma$  in the expression. Gelade and Neven [7] on the other hand note that this is not sufficient for GRE since one can construct non-trivial expressions with no symbols from  $\Sigma$ . Thus they count also operations, ending up with the size of the syntax tree of the expression. This is also sometimes called *reverse polish length* [5]. We use the latter concept here but the game can easily be adapted to alphabetic width or actual string length with parentheses if desired.

**Definition 2.1.** The *size* of a GRE is defined recursively as follows:

- $\text{sz}(\emptyset) = \text{sz}(\varepsilon) = \text{sz}(a) = 1$  for every  $a \in \Sigma$ ,
- $\text{sz}(R^*) = \text{sz}(\neg R) = \text{sz}(R) + 1$  and
- $\text{sz}(R_1 \cup R_2) = \text{sz}(R_1 R_2) = \text{sz}(R_1) + \text{sz}(R_2) + 1$ .

In the sequel we will deal with some rather large expression sizes. In particular, we will show a *non-elementary* succinctness gap between FO and RE. This means that the difference in required size is not expressible by an elementary function. In practice, it suffices to show that the size of the RE is above an exponential tower. For this, we define the function  $\text{twr}$  as follows:

- $\text{twr}(0) = 1$ ,
- $\text{twr}(n+1) = 2^{\text{twr}(n)}$ .

We also use the shorthand

$$[n] := \{1, \dots, n\}.$$

Finally we define some concepts and notations for the RE size game. First is the concept of regular expressions separating languages.

**Definition 2.2.** Let  $A, B \subseteq \Sigma^*$ . A GRE  $R$  separates  $A$  from  $B$  if  $A \subseteq L(R)$  and  $B \subseteq \Sigma^* \setminus L(R)$ .

Note that if  $A = L(R)$  and  $B = \Sigma^* \setminus L(R)$ , then  $R$  defines the language  $A$ , so separation is a sort of partial version of defining languages with expressions.

To consider catenation and star in the game, we will need notation for the different ways one can split a word into two or more shorter words.

Let  $w \in \Sigma^*$  and  $n \in \mathbb{N}$ . The set of  $n$ -splits of  $w$  is the set

$$\text{Sp}^n(w) = \{(w_1, \dots, w_n) \mid w_1 \dots w_n = w\}.$$

We also use the notation

$$\text{Sp}(w) := \bigcup_{n \in \mathbb{N}} \text{Sp}^n(w)$$

for the set of all splits of  $w$ .

### 3 Generalized regular expression size game

In this section we define a game for generalized regular expressions that is the equivalent of so called formula size games previously developed for different logics. Since we consider both overall size and number of stars in this paper, we present a game with a separate parameter for stars.

The GRE size game has two players, Samson (S) and Delilah (D). The game has four parameters: two sets of  $\Sigma$ -words,  $A_0$  and  $B_0$ , and two natural numbers  $k_0$  and  $s_0$  with  $k_0 \geq s_0$ . Samson wants to show that  $A_0$  can be separated from  $B_0$  using a GRE with size at most  $k_0$  and at most  $s_0$  stars. Delilah wants to refute this. The GRE size game with the above parameters is denoted by  $\text{GRES}(k_0, s_0, A_0, B_0)$ .

Positions of the game are of the form  $(k, s, A, B)$  where  $A$  and  $B$  are sets of words,  $k, s \in \mathbb{N}$  and  $k \geq s$ . The starting position is  $(k_0, s_0, A_0, B_0)$ . In a position  $P = (k, s, A, B)$ , if  $k = 0$ , then the game ends and D wins. Otherwise S has a choice of six moves (note that the empty word  $\varepsilon$  is covered in the  $a$ -move):

- $a$ -move: S chooses  $a \in \Sigma \cup \{\varepsilon\}$ . If  $A \subseteq \{a\}$  and  $a \notin B$ , the game ends and S wins. Otherwise D wins.
- $\emptyset$ -move: If  $A = \emptyset$ , S wins. Otherwise D wins.
- $\cup$ -move: S chooses subsets  $A_1, A_2 \subseteq A$  such that  $A_1 \cup A_2 = A$  and natural numbers  $k_1, k_2, s_1, s_2$  such that  $k_i \geq s_i$ ,  $k_1 + k_2 + 1 = k$  and  $s_1 + s_2 = s$ . Then D chooses a number  $i \in \{1, 2\}$ . The game continues from the position  $(k_i, s_i, A_i, B)$ .



- **cat-move:** For every  $w \in A$ , S chooses a 2-split  $(w_1, w_2)$ . Let  $A_i = \{w_i \mid w \in A\}$ . Then for every  $v \in B$ , S chooses a function  $f_v : \text{Sp}^2(v) \rightarrow \{1, 2\}$ . Let  $B_i = \{v_i \mid f_v(v_1, v_2) = i, (v_1, v_2) \in \text{Sp}^2(v)\}$ . S chooses numbers  $k_1, k_2, s_1, s_2$  such that  $k_i \geq s_i$ ,  $k_1 + k_2 + 1 = k$  and  $s_1 + s_2 = s$ . Finally D chooses a number  $i \in \{1, 2\}$ . The game continues from the position  $(k_i, s_i, A_i, B_i)$ .
- **\*-move:** If  $\varepsilon \in B$ , D wins. Otherwise, for every  $w \in A \setminus \{\varepsilon\}$ , S chooses a natural number  $n(w) > 0$  and an  $n(w)$ -split  $(w_1, \dots, w_{n(w)})$  with  $w_i \neq \varepsilon$  for every  $i \in [n(w)]$ . Let  $A' = \{w_i \mid i \in [n(w)], w \in A\}$ . Then for every  $v \in B$ , S chooses a function  $f_v : \text{Sp}(v) \rightarrow \mathbb{N}$  such that  $f_v(v_1, \dots, v_n) \in [n]$ . Let  $B' = \{v_i \mid f_v(v_1, \dots, v_n) = i, (v_1, \dots, v_n) \in \text{Sp}(v)\}$ . The game continues from the position  $(k-1, s-1, A', B')$ .
- **$\neg$ -move:** The game continues from the position  $(k-1, s, B, A)$ .

Note that since every move either ends the game or decreases the resource  $k$ , the game always ends in a finite number of moves and one of the players wins.

We now prove the crucial theorem that states the connection of the game to the succinctness of generalized regular expressions.

**Theorem 3.1.** *Let  $A, B \subseteq \Sigma^*$  and  $k, s \in \mathbb{N}$  with  $k \geq s$ . The following are equivalent:*

1. *S has a winning strategy in the game  $\text{GRES}(k, s, A, B)$ .*
2. *There is a generalized regular expression that separates  $A$  from  $B$  with size at most  $k$  and at most  $s$  stars.*

*Proof.* In the following we will always have  $i \in \{1, 2\}$  without explicit statement. We show the equivalence of 1 and 2 for all  $A$  and  $B$  by induction on the number  $k$ . The case  $k = 0$  is clear.

$1 \Rightarrow 2$ : Let  $\delta$  be a winning strategy for S in the game  $\text{GRES}(k, A, B)$ . Since  $\delta$  is a winning strategy, we have  $k > 0$ . The proof is divided into cases according to the first move of  $\delta$ :

- **$a$ -move:** If the first move is an  $a$ -move, because  $\delta$  is a winning strategy, we have  $A \subseteq \{a\} = L(a)$  and  $a \notin B$  so  $B \subseteq \Sigma^* \setminus L(a)$ . Thus the regular expression  $a$  separates  $A$  from  $B$ .
- **$\emptyset$ -move:** Now  $A = \emptyset$  so  $\emptyset$  separates  $A$  from  $B$ .
- **$\cup$ -move:** S chooses  $A_1, A_2 \subseteq A$  and  $k_1, k_2, s_1, s_2$  according to  $\delta$ . Since  $\delta$  is a winning strategy, S has winning strategies from both of the possible following positions  $(k_i, s_i, A_i, B)$ . Thus by induction hypothesis there are GREs  $R_1$  and  $R_2$  such that  $R_i$  separates  $A_i$  from  $B$ ,  $\text{sz}(R_i) \leq k_i$  and  $R_i$  has at most  $s_i$  stars. Now  $A_i \subseteq R_i$  and  $B \subseteq \Sigma^* \setminus L(R_i)$ . Therefore

$$A_0 = A_1 \cup A_2 \subseteq L(R_1) \cup L(R_2) = L(R_1 \cup R_2).$$

and  $B \subseteq (\Sigma^* \setminus L(R_1)) \cap (\Sigma^* \setminus L(R_2)) = \Sigma^* \setminus L(R_1 \cup R_2)$  so  $R_1 \cup R_2$  separates  $A$  from  $B$ . In addition,  $\text{sz}(R_1 \cup R_2) = \text{sz}(R_1) + \text{sz}(R_2) + 1 \leq k_1 + k_2 + 1 = k$  and  $R_1 \cup R_2$  has at most  $s_1 + s_2 = s$  stars.

- **cat-move:** S makes his choices according to  $\delta$ . Now S has a winning strategy for both positions  $(k_i, s_i, A_i, B_i)$  so by induction hypothesis there are GREs  $R_1$  and  $R_2$  such that  $R_i$  separates  $A_i$  from  $B_i$ ,  $\text{sz}(R_i) \leq k_i$  and  $R_i$  has at most  $s_i$  stars. Now  $A_i \subseteq L(R_i)$ . For every  $w \in A$  there are  $w_1 \in A_1$  and  $w_2 \in A_2$  such that  $w_1 w_2 = w$  so  $A \subseteq L(R_1)L(R_2) = L(R_1 R_2)$ . On the other side  $B_i \subseteq \Sigma^* \setminus L(R_i)$ . For every  $v \in B$  and every  $(v_1, v_2) \in \text{Sp}^2(v)$ , either  $v_1 \in B_1$  or  $v_2 \in B_2$ . Thus  $v \notin L(R_1)L(R_2) = L(R_1 R_2)$  so  $B \subseteq \Sigma^* \setminus L(R_1 R_2)$ . The GRE  $R_1 R_2$  thus separates  $A$  from  $B$ . The size and number of stars are handled as in the previous case.

- \*-move: S makes his choices according to  $\delta$ . S has a winning strategy for the following position  $(k-1, s-1, A', B')$  so by induction hypothesis there is a GRE  $R$  such that  $R$  separates  $A'$  from  $B'$ ,  $\text{sz}(R) \leq k-1$  and  $R$  has at most  $s-1$  stars. We have  $A' \subseteq L(R)$ . For every  $w \in A$  there is  $n(w) \in \mathbb{N}$  and an  $n(w)$ -split  $(w_1, \dots, w_{n(w)})$  such that  $w_j \in A'$  for  $j \in [n(w)]$ . Thus  $A \subseteq L(R)^* = L(R^*)$ . On the other side,  $B' \subseteq \Sigma^* \setminus L(R)$ . For every  $v \in B$  and every  $(v_1, \dots, v_n) \in \text{Sp}(v)$ , there is  $j \in [n]$  such that  $v_j \in B'$ . Thus  $v \notin L(R)^* = L(R^*)$  so  $B \subseteq \Sigma \setminus L(R^*)$ . The GRE  $R^*$  thus separates  $A$  from  $B$ . In addition,  $\text{sz}(R^*) = \text{sz}(R) + 1 \leq k$  and  $R^*$  has at most  $s-1+1 = s$  stars.
- $\neg$ -move: S has a winning strategy from the following position  $(k-1, s, B, A)$  so there is a GRE  $R$  that separates  $B$  from  $A$  with  $\text{sz}(R) \leq k-1$  and at most  $s$  stars. Now the GRE  $\neg R$  separates  $A$  from  $B$ . In addition,  $\text{sz}(\neg R) = \text{sz}(R) + 1 \leq k$  and  $\neg R$  has at most  $s$  stars.

2  $\Rightarrow$  1: Let  $R$  be a GRE that separates  $A$  and  $B$  with size at most  $k$  and at most  $s$  stars. The proof is divided into cases according to the outermost operator in  $R$ :

- $R = a \in \Sigma \cup \{\varepsilon\}$ : Since  $R$  separates  $A$  from  $B$ , we have  $A \subseteq \{a\}$  and  $B \subseteq \Sigma^* \setminus \{a\}$  so  $a \notin B$ . Thus S wins by making an  $a$ -move.
- $R = \emptyset$ : Now  $A = \emptyset$  so S wins by making a  $\emptyset$ -move.
- $R = R_1 \cup R_2$ : Since  $R$  separates  $A$  from  $B$ , we have  $A \subseteq L(R) = L(R_1) \cup L(R_2)$ . Let  $A_i = A \cap L(R_i)$ , let  $k_1 = \text{sz}(R_1)$  and let  $k_2 = k - k_1 - 1$ . Similarly let  $s_1$  be the number of stars in  $R_1$  and let  $s_2 = s - s_1$ . Now  $A_1 \cup A_2 = A$ ,  $k_i > s_i$ ,  $k_1 + k_2 + 1 = k$  and  $s_1 + s_2 = s$  so these are valid choices for a  $\cup$ -move. After the  $\cup$ -move,  $A_i \subseteq L(R_i)$  and  $B \subseteq \Sigma^* \setminus L(R) = (\Sigma^* \setminus L(R_1)) \cap (\Sigma^* \setminus L(R_2))$  so  $B \subseteq \Sigma^* \setminus L(R_i)$ . Now  $R_i$  separates  $A_i$  from  $B$ . In addition,  $\text{sz}(R_1) = k_1$ ,  $\text{sz}(R_2) = \text{sz}(R) - \text{sz}(R_1) - 1 \leq k - k_1 - 1 = k_2$ . Similarly  $R_1$  has  $s_1$  stars and  $R_2$  has at most  $s - s_1 = s_2$  stars. By induction hypothesis, S has a winning strategy for the game  $\text{GRES}(k_i, s_i, A_i, B)$ . Together with the first move, this is a winning strategy for the game  $\text{GRES}(k, s, A, B)$ .
- $R = R_1 R_2$ : Since  $R$  separates  $A$  from  $B$ , we have  $A \subseteq L(R) = L(R_1)L(R_2)$ . Thus for every  $w \in A_0$  there is  $(w_1, w_2) \in \text{Sp}^2(w)$  such that  $w_1 \in L(R_1)$  and  $w_2 \in L(R_2)$ . S makes a cat-move and chooses such a split for each  $w \in A$ . On the other side we have  $B \subseteq \Sigma^* \setminus L(R) = \Sigma^* \setminus L(R_1)L(R_2)$ . Thus for every  $v \in B$  and every  $(v_1, v_2) \in \text{Sp}^2(v)$ , we have  $v_1 \notin L(R_1)$  or  $v_2 \notin L(R_2)$ . For the function  $f_v : \text{Sp}(v) \rightarrow \mathbb{N}$ , S chooses  $i = f_v(v_1, v_2)$  so that  $v_i \notin L(R_i)$ . S chooses  $k_i$  and  $s_i$  as in the previous case. Finally we have  $A_i \subseteq L(R_i)$  and  $B_i \subseteq \Sigma^* \setminus L(R_i)$  so  $R_i$  separates  $A_i$  from  $B_i$ . The resources  $k$  and  $s$  are handled like in the previous case. By induction hypothesis, S has a winning strategy from the position  $(k_i, s_i, A_i, B_i)$ .
- $R = R_1^*$ : Since  $R$  separates  $A$  from  $B$ , we have  $A \subseteq L(R) = L(R_1)^*$ . Thus for every  $w \in A$  there is  $(w_1, \dots, w_n) \in \text{Sp}(w)$  such that  $w_j \in L(R_1)$  for all  $j \in [n]$ . S makes a \*-move and chooses such a split for each  $w \in A$ . On the other side we have  $B \subseteq \Sigma^* \setminus L(R) = \Sigma^* \setminus L(R_1)^*$ . Note that  $\varepsilon \notin B$  so D does not win outright. Now for every  $v \in B$  and every  $(v_1, \dots, v_n) \in \text{Sp}(v)$  we have  $v_j \notin L(R_1)$  for some  $j \in [n]$ . For the function  $f_v : \text{Sp}(v) \rightarrow \mathbb{N}$ , S chooses  $j = f_v(v_1, \dots, v_n)$  so that  $v_j \notin L(R_1)$ . Finally we have  $A' \subseteq L(R_1)$  and  $B' \subseteq \Sigma^* \setminus L(R_1)$  so  $R_1$  separates  $A'$  from  $B'$ . In addition,  $\text{sz}(R_1) = \text{sz}(R) - 1 \leq k - 1$  and  $R_1$  has at most  $s - 1$  stars. By induction hypothesis, S has a winning strategy from the position  $(k-1, s-1, A', B')$ .
- $R = \neg R_1$ : S makes a  $\neg$ -move. Since  $R$  separates  $A$  from  $B$ , it follows that  $R_1$  separates  $B$  from  $A$ . In addition,  $\text{sz}(R_1) = \text{sz}(R) - 1 \leq k - 1$  and  $R_1$  has at most  $s$  stars. By induction hypothesis, S has a winning strategy from the position  $(k-1, s, B, A)$ .

□

We have defined the game for generalized regular expressions but this full game turns out to be very complex in a combinatorial sense. For the results in this paper we will use simpler games for RE and RE over star-free.

The RE size game  $\text{RES}(k, A, B)$  is the game  $\text{GRES}(k, s, A, B)$  with the  $\neg$ -move and the star parameter  $s$  removed. The proof of Theorem 3.1 with the  $\neg$ -move cases and  $s$  removed proves the following analogue for this game:

**Theorem 3.2.** *Let  $A, B \subseteq \Sigma^*$ ,  $k \in \mathbb{N}$ . The following are equivalent:*

1. *S has a winning strategy in the game  $\text{RES}(k, A, B)$ .*
2. *There is a regular expression that separates A from B with size at most k.*

The RE over star-free size game  $\text{RESFS}(k, s, A, B)$  is the game  $\text{GRES}(k, s, A, B)$  with the following modification: after a  $\neg$ -move, the following position is  $(k, 0, B, A)$  instead of the normal  $(k, s, B, A)$ . This corresponds with the syntax of RE over star-free, where stars cannot occur under complement. We omit the proof of the analogous theorem for this game:

**Theorem 3.3.** *Let  $A, B \subseteq \Sigma^*$  and  $k, s \in \mathbb{N}$  with  $k \geq s$ . The following are equivalent:*

1. *S has a winning strategy in the game  $\text{RESFS}(k, s, A, B)$ .*
2. *There is a RE over star-free expression that separates A from B with size at most k and at most s stars.*

As is usual with these sorts of games, we will need a simple lemma stating that if the same word is present on both sides of the game, D has a winning strategy. We prove the lemma for the GRE game and note that it can just as easily be proven for the other variations.

**Lemma 3.4.** *In a position  $P = (k, s, A, B)$  of a game  $\text{GRES}(k_0, s_0, A_0, B_0)$ , if there is  $w \in A \cap B$ , then D has a winning strategy from position P.*

*Proof.* Under the assumptions, we describe a strategy for D. For any move of S, this strategy either wins or maintains the condition of having  $w \in A \cap B$ . It is thus a winning strategy. We consider the cases for each possible move of S.

- *a-move:* Assume S chooses  $a \in \Sigma \cup \{\varepsilon\}$ . If  $A \subseteq \{a\}$ , then  $a = w \in B$ , so D wins.
- $\emptyset$ -move: Since  $w \in A$ ,  $A \neq \emptyset$  and D wins.
- $\cup$ -move: Assume S chooses subsets  $A_1, A_2 \subseteq A$ . Since  $A_1 \cup A_2 = A$ , there is  $i \in \{1, 2\}$  such that  $w \in A_i$ . D chooses this  $i$  and in the following position  $(k_i, s_i, A_i, B)$ , we have  $w \in A_i \cap B$ .
- cat-move: Let  $(w_1, w_2)$  be the split S chooses for  $w$  on the A-side and let  $f_w : \text{Sp}^2(w) \rightarrow \{1, 2\}$  be the function S chooses for  $w$  on the B-side. D chooses the number  $i := f_w(w_1, w_2)$ . In the following position  $(k_i, s_i, A_i, B_i)$ , we have  $w_i \in A_i \cap B_i$ .
- \* -move: If  $w = \varepsilon$ , D wins. Otherwise, let  $(w_1, \dots, w_n)$  be the split S chooses for  $w$  on the A-side and let  $f_w : \text{Sp}(w) \rightarrow \mathbb{N}$  be the function S chooses for  $w$  on the B-side. Let  $i := f_w(w_1, \dots, w_n)$ . In the following position  $(k-1, s-1, A', B')$  we have  $w_i \in A' \cap B'$ .
- $\neg$ -move: In the following position  $(k-1, s, B, A)$ , we have  $w \in B \cap A$ .

□

For the RE over star-free game, we need a further lemma that gives an easy condition to guarantee that the current sets  $A$  and  $B$  cannot be separated via a star-free expression. The language we use for the game has words with long strings of the same symbol in them. We call these  $a$ -chains for  $a \in \Sigma$ . For example, the word  $baabbbaaa$  has two  $a$ -chains of lengths 2 and 3 respectively. We use the GRE game with  $s = 0$  to argue about star-free expressions.

**Lemma 3.5.** *In a position  $P = (k, 0, A, B)$  of a game  $\text{GRES}(k_0, s_0, A_0, B_0)$ , if there are  $w \in A$  and  $w' \in B$  such that they only differ from each other by lengths of one or more chains of symbols, each of length more than  $k$  in both, then  $D$  has a winning strategy from position  $P$ .*

*Proof.* We describe a strategy for  $D$ . For each move of  $S$ , this strategy either wins or maintains the assumptions of the lemma so it is a winning strategy. We consider each possible move of  $S$ :

- $a$ -move:  $S$  chooses  $a \in \Sigma \cup \varepsilon$ . Since  $w$  has a chain with length more than  $k > 0$ , clearly  $w \neq a$  so  $D$  wins.
- $\emptyset$ -move: Since  $w \in A$ ,  $A \neq \emptyset$  and  $D$  wins.
- $\cup$ -move:  $S$  chooses subsets  $A_1, A_2 \subseteq A$ . Since  $A_1 \cup A_2 = A$ , we have  $w \in A_i$  for some  $i \in \{1, 2\}$ .  $D$  chooses this  $i$  and in the following position  $(k_i, 0, A_i, B)$  we have  $w \in A_i$  and  $w' \in B$ . In addition, the chains of  $w$  and  $w'$  that differ are of length more than  $k > k_i$ . Thus the assumptions still hold.
- cat-move: Let  $(w_1, w_2)$  be the split  $S$  chooses for  $w \in A$  and let  $f_{w'} : \text{Sp}^2(w') \rightarrow \{1, 2\}$  be the function  $S$  chooses for  $w' \in B$ . Let  $k_1, k_2$  be the numbers chosen by  $S$  with  $k_1 + k_2 + 1 = k$ . Since  $w$  and  $w'$  only differ by the lengths of some chains, for each chain in  $w$  we can find the corresponding chain in  $w'$ .

If the split  $(w_1, w_2)$  splits no chains where  $w$  and  $w'$  differ, then we consider the split  $(w'_1, w'_2)$  of  $w'$  at the corresponding point and in the following position  $(k_i, 0, A_i, B_i)$ , the assumptions hold since  $k_i < k$ .

Now assume  $(w_1, w_2)$  splits a chain of length more than  $k$  and the length of this chain is different but still more than  $k$  in  $w'$ . If the length of the chain in  $w_i$  is at more than  $k_i$  for both  $i$ , then we consider a split  $(w'_1, w'_2)$  of  $w'$  where the same holds. Recall such a split can be found since  $k_1 + k_2 + 1 = k$  and the length of the chain is more than  $k$  in  $w'$  also. Now the assumptions hold in the following position.

Otherwise, by symmetry we assume that the length of the chain in  $w_1$  is less than or equal to  $k_1$ . In this case we consider the split  $(w'_1, w'_2)$  of  $w'$  where the length of the chain in  $w'_1$  is identical to  $w_1$ . Now the lengths of the chains in  $w_2$  and  $w'_2$  are more than  $k_2$  since  $k_1 + k_2 + 1 = k$ . Thus if the following position is  $(k_2, 0, A_2, B_2)$ , then the assumptions hold. If the following position is  $(k_1, 0, A_1, B_1)$ , then either there are still other differing chains of length more than  $k > k_1$  and the assumptions hold, or  $w_1 = w'_1$  and  $D$  has a winning strategy by Lemma 3.4.

- $*$ -move: We assume that the star resource  $s = 0$  in the position  $P$  so  $S$  cannot make a  $*$ -move.
- $\neg$ -move: In the following position  $(k - 1, 0, B, A)$ , the assumptions still hold as they are symmetric w.r.t.  $A$  and  $B$  and  $k - 1 < k$ .

□

**Remark 3.6.** The GRE size game can be modified in several ways to obtain different games. The games for RE and RE over star-free are examples of this. Additional operations can be included by adding moves. For example the move corresponding to intersection is the union move with the roles of  $A$  and

$B$  switched. One could also have separate resources for different operations or ignore some operations entirely. It is also possible to modify how the resources work with binary moves to track the nesting depth of an operation instead of the number.

## 4 The succinctness gap between FO and RE

To compare the succinctness of FO and RE, we must restrict the models of FO to *word models*. These are finite models with a linear order and unary predicates to indicate which letter of the alphabet  $\Sigma$  is in each spot. Thus properties of words are often defined in a language of the form  $\text{FO}(<, P_1, \dots, P_n)$ .

In his thesis [14] Stockmeyer showed that star-free generalized regular expressions are non-elementarily more succinct than regular expressions. Since there is an elementary translation from FO to star-free expressions [12], this implies that FO is non-elementarily more succinct than RE. The proof of Stockmeyer is quite involved as he encodes computations of Turing machines into star-free expressions. In this section, we show a simple way to obtain the gap between FO and RE via the RE size game. Our proof relies on the following proposition which states that to define a large finite language with a RE, the RE must be quite large as well.

**Proposition 4.1.** *A finite language  $L$  cannot be defined via a RE with size less than  $\log|L|$ .*

*Proof.* Let  $L$  be a finite language and  $k_0 < \log|L|$ . We consider the game  $\text{RES}(k_0, L, \Sigma^* \setminus L)$ . We will show that after every move of S, D will either gain a winning strategy via Lemma 3.4, or D can maintain the following two conditions in any position  $(k, A, B)$  of the game:

1.  $k \leq \log(|A|)$
2.  $\Sigma^{>N} := \{w \in \Sigma^* \mid |w| > N\} \subseteq B$  for some  $N \in \mathbb{N}$

In the starting position  $(k_0, L, \Sigma^* \setminus L)$ , we have  $k_0 \leq \log(|L|)$  so condition 1 holds. For condition 2, note that since  $L$  is finite,  $\Sigma^* \setminus L$  includes every word with length greater than the maximum length of words in the language  $L$ .

Consider a position  $(k, A, B)$  of the game  $\text{RES}(k_0, L, \Sigma^* \setminus L)$  and assume conditions 1 and 2 hold. S has five different moves to choose from:

- $*$ -move: Since  $0 < k \leq \log(|A|)$ , we have  $|A| \geq 2$  so there is  $w \in A$  with  $w \neq \varepsilon$ . Let  $(w_1, w_2, \dots, w_m)$  be the split chosen by S for  $w$ . By condition 2, there is  $N \in \mathbb{N}$  such that  $\Sigma^{>N} \subseteq B$ . Let  $v = w_1^{N+1}$ . Now  $|v| > N$  so  $v \in B$ . For the split  $(w_1, w_1, \dots, w_1)$  of  $v$  S must choose the piece  $w_1$  so in the following position  $(k-1, A', B')$ , we have  $w_1 \in A' \cap B'$  and by Lemma 3.4, D has a winning strategy from this position.
- $\cup$ -move: Let  $A_1, A_2 \subseteq A$  and  $k_1, k_2 < k$  be the choices of S. If either  $A_i$  is empty, D chooses the other one and both conditions are trivially maintained. Assume both  $A_i$  are non-empty. Since  $A_1 \cup A_2 = A$ , we obtain  $|A_1| + |A_2| \geq |A|$ . Now we have  $k_i \leq \log(|A_i|)$  for some  $i \in \{1, 2\}$ , since otherwise

$$\begin{aligned} k &= k_1 + k_2 + 1 > \log(|A_1|) + \log(|A_2|) + 1 \\ &= \log(|A_1||A_2|) + 1 \geq \log(|A_1| + |A_2|) \geq \log(|A|) \geq k, \end{aligned}$$

which is a contradiction. D chooses such an  $i$ , fulfilling condition 1 in the following position is  $(k_i, A_i, B)$ . Condition 2 is trivially maintained since  $B$  remains unchanged in  $\cup$ -moves.

- **cat-move:** Let the two possible following positions be  $P_i = (k_i, A_i, B_i)$  for  $i \in \{1, 2\}$ . We consider condition 2 first. Let  $w \in \Sigma^{>N}$ . Let  $v \in A$  and let  $(v_1, v_2) = v$  be the split chosen by S for  $v$ . Now  $u = v_1w \in \Sigma^{>N} \subseteq B$ . For the split  $(v_1, w)$  of  $u$ , if S chooses the piece  $v_1$ , then  $v_1 \in A_1 \cap B_1$  and by Lemma 3.4, D has a winning strategy from position  $P_1$ . Thus we assume that S chooses the piece  $w$  and  $w \in B_2$ . In the same way using the word  $wv_2$ , we get  $w \in B_1$ . Thus, in order to not give D a winning strategy via Lemma 3.4, S must maintain condition 2 for both positions  $P_i$ .

Now let us address condition 1. Since for every  $w \in A$  there is  $w_1 \in A_1$  and  $w_2 \in A_2$  such that  $w_1w_2 = w$ , we obtain  $|A_1||A_2| \geq |A|$ . We again have  $k_i \leq \log(|A_i|)$  for some  $i \in \{1, 2\}$ , since otherwise

$$k = k_1 + k_2 + 1 > \log(|A_1|) + \log(|A_2|) + 1 = \log(|A_1||A_2|) + 1 \geq \log(|A|) \geq k,$$

which is a contradiction. D again fulfills condition 1 by choosing such an  $i$ .

- **$a$ - or  $\emptyset$ -move:** Since  $0 < k \leq \log(|A|)$ , we have  $|A| \geq 2$  so  $A \not\subseteq \{a\}$  and  $A \neq \emptyset$  and D wins the game.

□

The language we use encodes sets of *the cumulative hierarchy*, defined as follows:

$$\begin{aligned} V_0 &:= \emptyset \\ V_{n+1} &:= \mathcal{P}(V_n). \end{aligned}$$

For each set in the cumulative hierarchy, we define a set of natural encodings. The encodings correspond to the different ways the set could be written down using only set brackets  $\{$  and  $\}$ . To differentiate the encoded words from actual set notation, we will use parentheses  $($  and  $)$  instead. The encodings are defined as follows:

$$\begin{aligned} \text{enc}(\emptyset) &:= \{()\} \\ \text{enc}(X) &:= \{(e_1 \cdots e_n) \mid e_i \in \text{enc}(x_i), x_1 < \cdots < x_n \text{ is a linear order of } X\}. \end{aligned}$$

A set has several encodings corresponding to different orders of the elements. For example, the set  $V_2 = \{\emptyset, \{\emptyset\}\}$  has the encodings  $((()))$  and  $((())())$ .

Let  $\Sigma$  be the alphabet with  $($  and  $)$  and let  $n \in \mathbb{N}$ . We consider the following language:

$$L_n = \bigcup_{X \in V_{n+1}} \text{enc}(X).$$

We first define  $L_n$  in first-order logic with linear order  $<$  and a unary predicate symbol  $P$ .

We define some auxiliary formulas. We interpret the predicate  $P$  so that the left parentheses satisfy  $P$  and the right parentheses do not. We use the formulas  $L(x)$  and  $R(x)$  to indicate this. We also define the formula  $S(x, y)$  that says  $y$  is the successor of  $x$ .

$$L(x) := P(x), R(x) := \neg P(x), S(x, y) := x < y \wedge \neg \exists z(x < z < y)$$

We will often want to say that the subword from position  $x_1$  to  $x_2$  encodes an instance of a set  $X$ . For easy readability of these kinds of statements, we adopt a flexible notation, where capital letters are used as shorthand for pairs of variables, that is to say  $X := (x_1, x_2)$ . Whenever possible, we shall use only the capital letters but in some cases we need the singular variables also.

We define the formulas  $\text{set}_i(X)$  and  $X =_i Y$  by mutual recursion. We additionally define formulas  $X \in_i Y$ , but since these only refer to the formula  $\text{set}_i$ , they are not essential in the recursion but rather shorthand to make the formulas more readable. The formula  $\text{set}_i(X)$  says that  $X$  correctly encodes a set in  $V_i$  with no repetition. The formula  $X \in_i Y$  assumes  $Y$  encodes a set and says that  $X$  encodes a set in  $V_i$  and is an element of the set encoded by  $Y$ . Finally, the formula  $X =_i Y$  assumes  $X$  and  $Y$  both encode sets in  $V_i$  and says that these sets are the same. The definition by mutual recursion is as follows:

$$\begin{aligned} \text{set}_0(X) &:= L(x_1) \wedge R(x_2) \wedge S(x_1, x_2) \\ \text{set}_{i+1}(X) &:= x_1 < x_2 \wedge L(x_1) \wedge R(x_2) \\ &\quad \wedge \forall u(x_1 < u < x_2 \rightarrow \exists v(x_1 < v < x_2 \wedge (\text{set}_i(u, v) \vee \text{set}_i(v, u)))) \\ &\quad \wedge \forall A \forall B((A \in_i X \wedge B \in_i X \wedge a_1 \neq b_1) \rightarrow A \neq_i B) \\ \\ X \in_i Y &:= y_1 < x_1 < x_2 < y_2 \wedge \text{set}_i(X) \\ &\quad \wedge \neg \exists U(y_1 < u_1 < x_1 \wedge x_2 < u_2 < y_2 \wedge \text{set}_i(U)) \\ \\ X =_0 Y &:= \top \\ X =_{i+1} Y &:= \forall A(A \in_i X \rightarrow \exists B(B \in_i Y \wedge A =_i B)) \\ &\quad \wedge \forall B(B \in_i Y \rightarrow \exists A(A \in_i X \wedge A =_i B)) \end{aligned}$$

We use these auxiliary formulas to define the formula  $\varphi_n$ , which defines the language  $L_n$ . The formula  $\varphi_n$  says that the first and last symbol of the word encode a set in  $V_n$  with no repetition.

$$\varphi_n := \exists X(\forall z(x_1 \leq z \wedge z \leq x_2) \wedge \text{set}_n(X))$$

From the form of the formulas we see that  $\text{sz}(\varphi_n) = \mathcal{O}(c^n)$  for some small constant  $c$ .<sup>1</sup>

Now Proposition 4.1 allows us to easily prove a non-elementary succinctness gap between FO and RE. This gap already follows from the work of Stockmeyer [14]. He found a similar gap between star-free expressions and RE and an elementary translation from FO to star-free expressions [12] leads to this result.

**Theorem 4.2.** *FO( $\langle, P$ ) is non-elementarily more succinct than RE on words.*

*Proof.* The language  $L_n$  is finite and  $|L_n| \geq \text{twr}(n)$ . We have shown that  $L_n$  can be defined in FO( $\langle, P$ ) via a formula exponential in  $n$ . However, if  $k < \log(\text{twr}(n)) = \text{twr}(n-1)$ , by Theorem 4.1, D has a winning strategy in the game RES( $k, L, \Sigma^n \setminus L$ ). Thus, by Theorem 3.2, there is no RE that defines  $L$  with size less than  $\text{twr}(n-1)$ .  $\square$

## 5 Number of stars in RE over star-free

We shift our attention from the overall size of regular expressions to only the number of stars. Star height famously gives a hierarchy in terms of expressive power for RE [10] and the corresponding result for GRE is a notorious open problem. For the number of stars, a full hierarchy can be trivially obtained already in star height one. On the other hand, for GRE, we have so far been unable to prove results of this nature due to the added complexity brought to the game with full use of complement. We present

<sup>1</sup>Numerical calculations performed with Maple seem to indicate  $\text{sz}(\varphi_n) = \mathcal{O}(8^n)$ .

an interesting middle ground between RE and GRE we call RE over star-free. For these expressions, star-free, that is FO-definable, properties are combined using the operations of RE. For RE over star-free we show that the number of stars gives a hierarchy in terms of expressive power.

The aforementioned trivial hierarchy for RE is obtained via the expression  $a_1^* \cup \dots \cup a_n^*$  but we omit that proof since we prove the stronger hierarchy for RE over star-free expressions. The language we use is actually definable with  $n$  stars already in RE but we show that even if we allow RE over star-free expressions, it still requires  $n$  stars to define.

Let  $\Sigma_n = \{a_1, \dots, a_n\}$  be a set of  $n$  symbols. We consider the following  $\Sigma_n$ -language:

$$L_n := L\left(\bigcup_{i \in [n]} (a_1 \cup \dots \cup a_{i-1} \cup a_i^2 \cup a_{i+1} \cup \dots \cup a_n)^*\right)$$

In other words, for each word in  $w \in L_n$ , there is  $i \in [n]$  such that every  $a_i$ -chain in  $w$  has even length. We don't need the whole language  $L_n$  for the game so we use a simple subset instead. For  $k \in \mathbb{N}$  and  $i \in [n]$ , we define

$$L_{n,k} := \{\ell_1, \dots, \ell_n\} = \{a_1^{2k+1} \dots a_i^{2k} \dots a_n^{2k+1} \mid i \in [n]\}.$$

Each  $\ell_i$  is a word that consists of a chain of each symbol  $a_j$  in order. The chain of the specific symbol  $a_i$  has even length and all other chains of  $a_j$  have odd length.

**Theorem 5.1.** *Any RE over star-free expression  $R_n$  with  $L(R_n) = L_n$  has at least  $n$  stars.*

*Proof.* Let  $n \in \mathbb{N}$  and  $k_0 \geq n$ . We consider the languages  $A_0 := L_{n,k_0}$  and  $B_0 := \Sigma_n^* \setminus L_n$ . We will show that D has a winning strategy for the game  $\text{RES}(k_0, n-1, A_0, B_0)$ . Since  $A_0 \subseteq L_n$  and  $B_0 = \Sigma_n^* \setminus L_n$ , D then also has a winning strategy for the game  $\text{RES}(k_0, n-1, L_n, \Sigma_n^* \setminus L_n)$ . The number  $k_0$  is arbitrary so by Theorem 3.1 the claim follows.

Let  $(k, s, A, B)$  be a position in the game  $\text{RES}(k_0, n-1, A_0, B_0)$ . We will show that D can maintain the following conditions while a  $*$ -move has not been made. We will also see that if a  $*$ -move is made while the conditions hold, D gains a winning strategy. The conditions are:

There is  $I \subseteq [n]$  such that

1.  $|I| > s$ ,
2. for every  $i \in I$  there is  $w_i \in A$  and  $u_i, v_i \in \Sigma_n^*$  s.t.  $\ell_i = u_i w_i v_i$  and  $(a_i)^{k+1}$  is a subword of  $w_i$ ,
3. for every  $r \in \Sigma_n^*$  if there are  $i, j \in I$  with  $u_i r v_j \in B_0$ , then  $r \in B$ .

Intuitively condition 2 says that in the position  $(k, s, A, B)$ , the set  $A$  has some ‘descendants’  $w_i$  of the original words  $\ell_i$  in  $A_0$ . The words  $u_i$  and  $v_i$  are the parts that have been removed from  $\ell_i$  via cat-moves to obtain  $w_i$ . The set  $I$  contains the indices that still have descendants in play. Condition 1 states that the number of such indices is always larger than the star resource  $s$ . Finally condition 3 says that the set  $B$  has versions of the original words in  $B_0$  with some prefix  $u_i$  and some suffix  $v_j$  removed.

In the starting position  $(k_0, n-1, A_0, B_0)$  the conditions hold with  $I = [n]$  and for every  $i \in I$ ,  $w_i = \ell_i$  and  $u_i = v_i = \varepsilon$ . We consider each possible move of S and show that in every case either the above conditions are maintained or D wins eventually by a winning strategy described in a previous lemma.

- $\neg$ -move: We must first check that while the conditions hold, a  $\neg$ -move from S leads to a win for D. Let  $i \in I$ . By condition 2, the word  $w_i$  has  $(a_i)^{k+1}$  as a subword. Let  $r$  be a word obtained from  $w_i$  by adding one  $a_i$  to this  $a_i$ -chain. Since  $\ell_i = u_i w_i v_i$  and the  $a_i$ -chain in  $\ell_i$  is even, we know the chain in  $u_i r v_i$  is odd. The chains of all other  $a_j$  are odd in  $\ell_i$  and thus also in  $u_i r v_i$  so  $u_i r v_i \in B_0$ . By



condition 3, we have  $r \in B$ . If S makes a  $\neg$ -move, his star resource  $s$  becomes 0. In the following position  $(k-1, 0, B, A)$ , we have  $r \in B$  and  $w_i \in A$  and the two words only differ by the length of a chain with length more than  $k-1$  so Lemma 3.5 gives D a winning strategy. This means that while the conditions hold, S can only attempt  $\cup$ -moves, cat-moves and  $*$ -moves if he hopes to win.

- $\cup$ -move: Let  $A_1, A_2 \subseteq A$  be the subsets S chooses. For each  $i \in I$ ,  $w_i \in A_1$  or  $w_i \in A_2$ . Let  $I_1, I_2 \subseteq I$  be the sets of indices generated this way. Since  $|I| > s$ , we have  $|I_1| > s_1$  or  $|I_2| > s_2$ . D chooses the position where this holds. Condition 2 still clearly holds and since  $B$  remains unchanged in this move, so does condition 3.
- cat-move: Let  $i \in I$  and let  $(w_{i,1}, w_{i,2})$  be the split S chooses for  $w_i$ . Let  $k_1 + k_2 + 1 = k$  and  $s_1 + s_2 = s$  be the resource splits of S. Since  $w_i$  has  $(a_i)^{k+1}$  as a subword,  $w_{i,1}$  has  $(a_i)^{k_1+1}$  as a subword or  $w_{i,2}$  has  $(a_i)^{k_2+1}$  as a subword. We divide  $I$  into subsets  $I_1, I_2$  according to this condition. Since  $|I| > s$ , we have  $|I_1| > s_1$  or  $|I_2| > s_2$ . Assume the former. Now condition 2 is satisfied for  $w_{i,1}$  by letting  $u_{i,1} := u_i$  and  $v_{i,1} := w_{i,2}v_i$ . For condition 3, let  $u_{i,1}rv_{j,1} \in B_0$  for some  $r \in \Sigma_n^*$  and  $i, j \in I_1$ . Now  $u_irw_{j,2}v_j \in B_0$  so by condition 3 in the position before this move,  $rw_{j,2} \in B$ . For the split  $(r, w_{j,2})$  of  $rw_{j,2}$  S must choose  $r$  to have a chance, since choosing  $w_{j,2}$  would result in an identical word on both sides for the position  $(k_2, s_2, A_2, B_2)$ . So either D has a winning strategy by Lemma 3.4 or  $r \in B_1$  for every such  $r$  and condition 3 holds for the position  $(k_1, s_1, A_1, B_1)$  and D chooses this position. The case of  $|I_2| > s_2$  is handled in the same way.
- $*$ -move: S can only make this move if  $1 \leq s < |I|$  so we have  $i, j \in I$  with  $i < j$ . We will show that this is enough to give D a winning strategy if S makes a  $*$ -move. Our aim is to show that a word of the form  $(w_j)^{m_1}(w_i)^{m_2}$  is in  $B$ . We will use condition 3 to show this. Condition 3 requires a word of the form  $u_irv_j$  to be in  $B_0$  and words in  $B_0$  have odd chains of all symbols  $a_p$ . Thus we begin by finding odd chains of all symbols in our words.

Recall that by condition 2, there are  $w_i \in A$  and  $u_i, v_i \in \Sigma^*$  such that  $\ell_i = u_iw_iv_i$  and  $(a_i)^{k+1}$  is a subword of  $w_i$ . The same holds for  $j$ . Let  $u \in \{u_i, u_j\}$  be the one of the two words with more odd chains of symbols. If they have the same number of odd chains, we choose, say, the longer word. Choose  $v \in \{v_i, v_j\}$  the same way. Next, we will show that for each  $p \in [n]$ , at least one of the words  $w_i, w_j, u$  and  $v$  has an odd  $a_p$ -chain.

Recall that the words in  $A_0$  have chains of symbols  $a_p$  in order and only the  $a_i$ -chain in a word  $\ell_i$  is even while all the others are odd. Furthermore,  $\ell_i = u_iw_iv_i$  and  $w_i$  has  $(a_i)^{k+1}$  as a subword so all chains in  $u_i$  are odd except possibly the last. Thus for each odd chain in  $u_i$  there is also one of the same symbol in  $u$  and the same goes for  $u_j$ . Similarly for each odd chain in  $v_i$  or  $v_j$  there is one in  $v$ .

We now show that for every  $p \in [n]$  there is an odd chain in at least one of the words  $w_i, w_j, u$  and  $v$ . First, let  $p < i$ . If there is an odd  $a_p$ -chain in  $w_i$  we are done so let us assume there is not. Now the  $a_p$ -chain in  $w_i$  is even (possibly empty) and since the chain in  $u_iw_iv_i = \ell_i$  is odd, we know the one in  $u_i$  is odd. As noted above, an odd chain in  $u_i$  means there is also one in  $u$ . So in this case there is an odd  $a_p$ -chain in  $w_i$  or  $u$ . The case  $p > i$  is very similar and we obtain an  $a_p$ -chain in  $w_i$  or  $v$ . Finally let  $p = i$ . Now  $p < j$  so like above we obtain an odd  $a_p$ -chain in  $w_j$  or  $u$ .

We now have an odd chain of each  $a_p$  among the words  $w_i, w_j, u$  and  $v$ , but we still need to make sure the specific way we catenate these words does not remove the only odd chains of a symbol by merging them into an even one. Let  $f(w)$  be the index of the first symbol of a word  $w$  and  $l(w)$  the index of the last. By condition 2 we have  $f(w_i) \leq i \leq l(w_i)$ . The same goes for  $f(w_j) \leq j \leq l(w_j)$ . We start with  $w_jw_i$ . By the above we obtain  $f(w_i) \leq i < j \leq l(w_j)$  so this catenation cannot result

in any merging of odd chains. Next we add  $u$  to the left. If  $l(u) = f(w_j)$  and both chains are odd, this merges the chains into an even one. Here we consider two cases. First, if  $w_j$  is just an odd  $a_j$ -chain, then for some  $m_1 \in \{1, 2\}$  the  $a_j$ -chain in the word  $u(w_j)^{m_1}w_i$  is odd. If  $w_j$  has other symbols besides  $a_j$ , then the word  $u(w_j)^2w_i$  has an odd  $a_{f(w_j)}$ -chain at the start of the second  $w_j$ . We have thus obtained  $u(w_j)^{m_1}w_i$  with an odd chain of  $a_{f(w_j)}$ . We finally add  $v$  to the right in a similar fashion. If  $l(w_i) = f(v)$  and both chains are odd, we again consider the cases of  $w_i$  being just an odd  $a_i$ -chain or a larger word and we obtain  $m_2 \in \{1, 2\}$  such that  $u(w_j)^{m_1}(w_i)^{m_2}v$  has an odd chain of  $a_{l(w_i)}$ .

As the words  $w_i$ ,  $w_j$ ,  $u$  and  $v$  have an odd chain of each symbol and we have made sure the concatenations did not lose any, our concatenated word  $u(w_j)^{m_1}(w_i)^{m_2}v$  is now in  $B_0$ . Since  $u \in \{u_i, u_j\}$  and  $v \in \{v_i, v_j\}$ , by condition 3,  $(w_j)^{m_1}(w_i)^{m_2} \in B$ .

Let us finish by showing how this gives D a winning strategy after the  $*$ -move in progress. S must give splits for  $w_i$  and  $w_j$  and every piece of these splits is in the left set of the following position,  $A'$ . S must also choose a piece of every split of  $(w_j)^{m_1}(w_i)^{m_2}$  to add to the right set,  $B'$ . The split of  $(w_j)^{m_1}(w_i)^{m_2}$  we are interested in is the one where each subword  $w_i$  and  $w_j$  is split according to the splits given by S for  $w_i$  and  $w_j$ . For this split, S must choose one of the pieces already in  $A'$  to also be in  $B'$ . Thus, in the following position  $(k-1, s-1, A', B')$ , there is an identical word on both sides and D has a winning strategy by Lemma 3.4. Thus if S makes a  $*$ -move while the conditions hold, D eventually wins.

□

## 6 Conclusion

We have presented a formula size game for GRE, RE and a middle ground between these we call RE over star-free expressions. We used the RE version to reprove a non-elementary succinctness gap between FO and RE via a large finite language. For RE over star-free we showed that the number of stars gives a full hierarchy in terms of expressive power. As the astute reader has noted, we have not used the full GRE size game in this paper. This is due to the considerable combinatorial complexity of the game. A clear goal for further research is to find some handle on this complexity at least for some problems. A good first candidate is to prove that there is a star height one language that requires two stars to define via a GRE.

As noted in Remark 3.6, the games can be modified to isolate different operations with different resources or counting the nesting depth of some operations instead of the number. This means that the games could naturally be used to investigate any problem having to do with bounds on operators such as the generalized star height problem.

## References

- [1] M. Adler & N. Immerman (2003): *An  $n!$  lower bound on formula size*. *ACM Trans. Comput. Log.* 4(3), pp. 296–314, doi:10.1145/772062.772064.
- [2] H-D. Ebbinghaus & J. Flum (1995): *Finite Model Theory: First Edition*. Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-662-03182-7.
- [3] L. C. Eggan (1963): *Transition graphs and the star-height of regular events*. *Michigan Math. J.* 10(4), pp. 385–397, doi:10.1307/mmj/1028998975.

- [4] A. Ehrenfeucht & P. Zeiger (1974): *Complexity Measures for Regular Expressions*. In: *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, STOC '74*, Association for Computing Machinery, New York, NY, USA, p. 75–79, doi:10.1145/800119.803886.
- [5] K. Ellul, B. Krawetz, J. Shallit & M. Wang (2005): *Regular Expressions: New Results and Open Problems*. *J. Autom. Lang. Comb.* 10(4), p. 407–437, doi:10.25596/jalc-2005-407.
- [6] W. Gelade (2010): *Succinctness of regular expressions with interleaving, intersection and counting*. *Theor. Comput. Sci.* 411(31-33), pp. 2987–2998, doi:10.1016/j.tcs.2010.04.036.
- [7] W. Gelade & F. Neven (2012): *Succinctness of the Complement and Intersection of Regular Expressions*. *ACM Trans. Comput. Logic* 13(1), doi:10.1145/2071368.2071372.
- [8] H. Gruber & M. Holzer (2008): *Finite Automata, Digraph Connectivity, and Regular Expression Size*. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfssdóttir & I. Walukiewicz, editors: *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Lecture Notes in Computer Science 5126*, Springer, pp. 39–50, doi:10.1007/978-3-540-70583-3\_4.
- [9] H. Gruber & M. Holzer (2009): *Tight Bounds on the Descriptive Complexity of Regular Expressions*. In V. Diekert & D. Nowotka, editors: *Developments in Language Theory*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 276–287, doi:10.1007/978-3-642-02737-6\_22.
- [10] K. Hashiguchi (1988): *Algorithms for Determining Relative Star Height and Star Height*. *Inf. Comput.* 78(2), pp. 124–169, doi:10.1016/0890-5401(88)90033-8.
- [11] J. E. Hopcroft, R. Motwani & J. D. Ullman (2006): *Introduction to Automata Theory, Languages, and Computation*, 3rd edition. Addison-Wesley Longman Publishing Co., Inc., USA.
- [12] R. McNaughton & S. A. Papert (1971): *Counter-Free Automata (M.I.T. Research Monograph No. 65)*. The MIT Press.
- [13] A. A. Razborov (1990): *Applications of matrix methods to the theory of lower bounds in computational complexity*. *Combinatorica* 10(1), pp. 81–93, doi:10.1007/BF02122698.
- [14] L. Stockmeyer (1974): *The complexity of decision problems in automata theory and logic*. Ph.D. thesis, Massachusetts Institute of Technology.
- [15] Q. Yan (2007): *Classifying regular languages by a split game*. *Theoretical Computer Science* 374(1), pp. 181–190, doi:10.1016/j.tcs.2006.12.041.



# PUBLICATION

## IV

### Defining Long Words Succinctly in FO and MSO

Lauri Hella and Miikka Vilander

In: *Revolutions and Revelations in Computability - 18th Conference on Computability in Europe, CiE 2022, Swansea, UK, July 11-15, 2022, Proceedings*. Ed. by Ulrich Berger et al.

Vol. 13359 Lecture Notes in Computer Science. Springer, 2022, pp. 125–138

DOI: 10.1007/978-3-031-08740-0\_11

**Publication reprinted with the permission of the copyright holders.**





# Defining Long Words Succinctly in FO and MSO

Lauri Hella<sup>(ID)</sup> and Miikka Vilander<sup>(✉)</sup><sup>(ID)</sup>

Tampere University, 33100 Tampere, Finland  
`miikka.vilander@tuni.fi`

**Abstract.** We consider the length of the longest word definable in FO and MSO via a formula of size  $n$ . For both logics we obtain as an upper bound for this number an exponential tower of height linear in  $n$ . We prove this by counting types with respect to a fixed quantifier rank. As lower bounds we obtain for both FO and MSO an exponential tower of height in the order of a rational power of  $n$ . We show these lower bounds by giving concrete formulas defining word representations of levels of the cumulative hierarchy of sets. In addition, we consider the Löwenheim-Skolem and Hanf numbers of these logics on words and obtain similar bounds for these as well.

**Keywords:** Logic on words · Monadic second-order logic · Succinctness

## 1 Introduction

We consider the succinctness of defining words. More precisely, if we allow formulas of size up to  $n$  in some logic, we want to know the length of the longest word definable by such formulas.

This question is not very interesting for all formalisms. An example where this is the case is given by regular expressions. There is no smaller regular expression that defines a word than the word itself. This result is spelled out at least in the survey [3]. However, the situation is completely different for monadic second-order logic MSO over words with linear order and unary predicates for the letters. Even though MSO has the same expressive power as regular expressions over words, it is well-known that MSO is non-elementarily more succinct. This follows from the results in the PhD thesis [12] of Stockmeyer. In fact, he proved that the problem whether the language defined by a given star-free generalized regular expression has non-empty complement is of non-elementary complexity with respect to the length of the expression. Since star-free generalized expressions can be polynomially translated into first-order logic FO, it follows that already FO is non-elementarily more succinct than regular expressions. In the article [11], Reinhardt

---

M. Vilander acknowledges the financial support of the Academy of Finland project *Explaining AI via Logic* (XAILOG), project number 345612.

uses a variation of Stockmeyer's method for proving similar non-elementary succinctness gaps between finite automata and the logics MSO and FO.

In this paper our focus is in the definability of words in MSO and FO. As far as we know, this aspect of succinctness has not been considered previously in the context of words. We show that these logics can define words of non-elementary length via formulas of polynomial size.

In order to argue about definability via formulas of bounded size, we define the size  $n$  fragments FO $[n]$  and MSO $[n]$  that include only formulas of size up to  $n$ . We also define similar quantifier rank  $k$  fragments FO $_k$  and MSO $_k$  and use them to prove our upper bounds. Both of these types of fragments are essentially finite in the sense that they contain only a finite number of non-equivalent formulas. We call the length of the longest word definable in a fragment the definability number of that fragment. Using this concept, our initial question is reframed as studying the definability numbers of FO $[n]$  and MSO $[n]$ .

The definability number of a fragment is closely related to the Löwenheim-Skolem and Hanf numbers of the fragment. The Löwenheim-Skolem number of a fragment is the smallest number  $m$  such that each satisfiable formula in the fragment has a model of size at most  $m$ . The Hanf number is the smallest number  $l$  such that any formula with a model of size greater than  $l$  has arbitrarily large models. These were originally defined for extensions of first-order logic in the context of model theory of infinite structures, but they are also meaningful in the context of finite structures. For a survey on Löwenheim-Skolem and Hanf numbers both on infinite and finite structures see [1]. For previous research on finite Löwenheim-Skolem type results see [4] and [5].

Aside from what we have already mentioned, related work includes the article [9] of Pikhurko and Verbitsky, where they consider the complexity of single finite structures. They study the minimal quantifier rank in FO of both defining a single finite structure and separating it from other structures of the same size. In [8] the same authors and Spencer consider quantifier rank and formula size required to define single graphs in FO. The survey [10] by Pikhurko and Verbitsky covers the above work and more on the logical complexity of single graphs in FO. By logical complexity they mean minimal quantifier rank, number of variables and length of a defining formula as functions of the size of the graph. They give an extensive account of these measures and relate them to each other, the Ehrenfeucht-Fraïssé game and the Weisfeiler-Lehman algorithm. An important difference between our approach and theirs is that we take formula size as the parameter and look for the longest definable word, whereas they do the opposite.

Our contributions are upper and lower bounds for the definability, Löwenheim-Skolem and Hanf numbers of the size  $n$  fragments of FO and MSO on words. The upper bounds in Sect. 3 are obtained by counting types with respect to the quantifier rank  $n/2$  fragment. The upper bounds for both FO and MSO are exponential towers of height  $n/2 + \log^*(t) + 1$  where  $t$  is a polynomial term. The lower bounds in Sects. 4 and 5 are given by concrete polynomial size formulas that define words of non-elementary length based on the cumulative hierarchy of sets. The lower bounds are exponential towers of height  $\sqrt[5]{n/c}$  for FO and  $\sqrt{n/c}$  for MSO, respectively.



An anonymous referee pointed out that lower bounds similar to ours can be obtained by adapting the method used by Reinhardt in [11], which in turn is based on the work of Stockmeyer [12]. However, our formulas are based on the cumulative hierarchy of sets instead of the binary counters used in Stockmeyer and Reinhardt. Furthermore, we emphasize defining single words and relate the bounds to Löwenheim-Skolem and Hanf numbers.

Note that our results only apply in the context of words. If finite structures over arbitrary finite vocabularies are allowed, then there are no computable upper bounds for the Löwenheim-Skolem or Hanf numbers of the size  $n$  fragments of FO. For the Löwenheim-Skolem number, this follows from Trakhtenbrot's theorem<sup>1</sup> (see, e.g., [7]), and for the Hanf number, this follows from a result of Grohe in [4]. Clearly the same applies for the size  $n$  fragments of MSO as well.

## 2 Preliminaries

The logics we consider in this paper are first-order logic FO and monadic second-order logic MSO and their (typically finite) fragments. The syntax and semantics of these are standard and well-known. Due to space restrictions we will not present them here, instead directing the reader to [2] and [7].

In terms of structures we limit our consideration to words of the two letter alphabet  $\Sigma = \{l, r\}$ . We have chosen to use letters for readability but intuitively the  $l$  stands for the left brace  $\{$  and  $r$  for the right brace  $\}$ . We use these later to encode sets as words. The empty set would be encoded as  $lr$ , or  $\{\}$ .

When we say that a word satisfies a logical sentence, we mean the natural corresponding word model does. A word model is a finite structure with linear order and unary predicates  $P_l$  and  $P_r$  for the two symbols. Since we only consider words over the two letter alphabet  $\Sigma$ , we will tacitly assume that all formulas of MSO are in the vocabulary  $\{<, P_l, P_r\}$  of the corresponding word models (and similarly for FO-formulas).

**Definition 1.** *The size  $\text{sz}(\varphi)$  of a formula  $\varphi \in \text{MSO}$  is defined recursively as follows:*

- $\text{sz}(\varphi) = 1$  for atomic  $\varphi$ ,
- $\text{sz}(\neg\psi) = \text{sz}(\psi) + 1$ ,
- $\text{sz}(\psi \wedge \theta) = \text{sz}(\psi \vee \theta) = \text{sz}(\psi) + \text{sz}(\theta) + 1$ ,
- $\text{sz}(\exists x\psi) = \text{sz}(\forall x\psi) = \text{sz}(\exists U\psi) = \text{sz}(\forall U\psi) = \text{sz}(\psi) + 1$ .

For  $n \in \mathbb{N}$  the size  $n$  fragment of MSO, denoted  $\text{MSO}[n]$ , consists of the formulas of MSO with size at most  $n$ . Size as well as size  $n$  fragments are defined in the same way for FO.

**Definition 2.** *The quantifier rank  $\text{qr}(\varphi)$  of a formula  $\varphi \in \text{MSO}$  is defined recursively as follows:*

<sup>1</sup> Trakhtenbrot's theorem states that the finite satisfiability problem of FO is undecidable. Hence there cannot exist any computable upper bound for the size of models that need to be checked to see whether a given formula is satisfiable.

- $\text{qr}(\varphi) = 0$  for atomic  $\varphi$ ,
- $\text{qr}(\neg\psi) = \text{qr}(\psi)$ ,
- $\text{qr}(\psi \wedge \theta) = \text{qr}(\psi \vee \theta) = \max\{\text{qr}(\psi), \text{qr}(\theta)\}$ ,
- $\text{qr}(\exists x\psi) = \text{qr}(\forall x\psi) = \text{qr}(\exists U\psi) = \text{qr}(\forall U\psi) = \text{qr}(\psi) + 1$ .

For  $k \in \mathbb{N}$ , the quantifier rank  $k$  fragment of MSO, denoted  $\text{MSO}_k$ , consists of the formulas  $\varphi \in \text{MSO}$  with  $\text{qr}(\varphi) \leq k$ . The quantifier rank  $k$  fragment of FO is defined in the same way and denoted  $\text{FO}_k$ .

Note that both size  $n$  fragments and quantifier rank  $k$  fragments are essentially finite in the sense that they contain only finitely many non-equivalent formulas.

**Definition 3.** For each (finite) fragment  $L$  of MSO or FO, we define the relation  $\equiv_L$  on  $\Sigma$ -words as

$$w \equiv_L v, \text{ if } w \text{ and } v \text{ agree on all } L\text{-sentences.}$$

Clearly  $\equiv_L$  is an equivalence relation. We denote the set of equivalence classes of  $\equiv_L$  by  $\Sigma^* / \equiv_L$  and define a notation for the number of these classes.

**Definition 4.** For each (finite) fragment  $L$  of MSO or FO, we denote the number of equivalence classes of  $\equiv_L$  by  $N_L$ , i.e.

$$N_L := |\Sigma^* / \equiv_L|.$$

Note that each equivalence class of  $\equiv_L$  is uniquely determined by a subset  $\text{tp}_L(w) = \{\varphi \in L \mid w \models \varphi\}$  of  $L$  sentences, which we call the  $L$ -type of  $w$ . Thus,  $N_L$  is the number of  $L$ -types. In the case  $L = \text{MSO}_k$  or  $L = \text{FO}_k$ , we talk about quantifier rank  $k$  types.

**Definition 5.** We say that a sentence  $\varphi \in \text{MSO}$  defines a word  $w \in \Sigma^+$  if  $w \models \varphi$  and  $v \not\models \varphi$  for all  $v \in \Sigma^+ \setminus \{w\}$ .

For a fragment  $L$  of MSO or FO, we denote by  $\text{Def}(L)$  the set of words definable in  $L$ , i.e.

$$\text{Def}(L) := \{w \in \Sigma^+ \mid \text{there is } \varphi \in L \text{ s.t. } \varphi \text{ defines } w\}.$$

In order to discuss words of non-elementary length and make our bounds precise, we define the exponential tower function  $\text{twr}$  for the positive reals as well as the, essentially inverse, iterated logarithm function  $\log^*$ .

**Definition 6.** The exponential tower function  $\text{tower} : \mathbb{N} \rightarrow \mathbb{N}$  is defined recursively by setting  $\text{tower}(0) := 1$  and  $\text{tower}(n + 1) := 2^{\text{tower}(n)}$ . We extend this definition to a function  $\text{twr} : [0, \infty[ \rightarrow \mathbb{N}$  by setting  $\text{twr}(x) = \text{tower}(\lceil x \rceil)$ . The iterated logarithm function  $\log^* : [1, \infty[ \rightarrow \mathbb{N}$  is defined by setting  $\log^*(x)$  as the smallest  $m \in \mathbb{N}$  that has  $\text{tower}(m) \geq x$ .

## 2.1 Definability, Löwenheim-Skolem and Hanf Numbers

Löwenheim-Skolem and Hanf numbers were originally introduced for studying the behaviour of extensions of first-order logic on infinite structures. See the article [1] of Ebbinghaus for a nice survey on the infinite case. As observed in [4], with suitable modifications, it is possible to give meaningful definitions for these numbers also on finite structures. We will now give such definitions for finite fragments  $L$  of FO and MSO, and in addition, we introduce the closely related definability number of  $L$ .

Let  $\varphi$  be a sentence in MSO over  $\Sigma$ -words. If it has a model, we denote by  $\mu(\varphi)$  the minimal length of a model of  $\varphi$ :  $\mu(\varphi) = \min\{|w| \mid w \in \Sigma^+, w \models \varphi\}$ . If  $\varphi$  has no models, we stipulate  $\mu(\varphi) = 0$ . Furthermore, we denote by  $\nu(\varphi)$  the maximum length of a model of  $\varphi$ , assuming the maximum is well-defined. If the maximum is not defined, i.e., if  $\varphi$  has no models or has arbitrarily long models, we stipulate  $\nu(\varphi) = 0$ .

**Definition 7.** *Let  $L$  be a finite fragment of MSO or FO with  $\text{Def}(L) \neq \emptyset$ .*

(a) *The definability number of  $L$  is*

$$\text{DN}(L) = \max\{|w| \mid w \in \Sigma^+, w \in \text{Def}(L)\}.$$

(b) *The Löwenheim-Skolem number of  $L$  is  $\text{LS}(L) = \max\{\mu(\varphi) \mid \varphi \in L\}$ .*

(c) *The Hanf number of  $L$  is  $\text{H}(L) = \max\{\nu(\varphi) \mid \varphi \in L\}$ .*

Thus,  $\text{DN}(L)$  is the length of the longest  $L$ -definable word. Note further that  $\text{LS}(L)$  is the smallest number  $m$  such that every  $\varphi \in L$  that has a model, has a model of length at most  $m$ . Similarly  $\text{H}(L)$  is the smallest number  $\ell$  such that if  $\varphi \in L$  has a model of length greater than  $\ell$ , then it has arbitrarily long models.

Since every sentence  $\varphi$  of MSO defines a regular language over  $\Sigma$ , and there is an effective translation from MSO to equivalent finite automata, it is clear that we can compute the numbers  $\mu(\varphi)$  and  $\nu(\varphi)$  from  $\varphi$ . Consequently, for any finite fragment  $L$  of MSO,  $\text{LS}(L)$  and  $\text{H}(L)$  can be computed from  $L$ .

As we mentioned in the Introduction,  $\text{LS}(\text{FO}[n])$  and  $\text{H}(\text{FO}[n])$  are not computable from  $n$  if we consider arbitrary finite models instead of words. Clearly the same holds also for the fragments  $\text{FO}_k$ ,  $\text{MSO}[n]$  and  $\text{MSO}_k$ .

It follows immediately from Definition 7 that the definability number of any finite fragment of MSO is bounded above by its Löwenheim-Skolem number and its Hanf number:

**Proposition 1.** *If  $L$  is a finite fragment of MSO, then  $\text{DN}(L) \leq \text{LS}(L), \text{H}(L)$ .*

*Proof.* It suffices to observe that if  $w \in \text{Def}(L)$ , then  $\mu(\varphi) = \nu(\varphi) = |w|$ , where  $\varphi \in L$  is the sentence that defines  $w$ .

Note that all three cases for the relationship between  $\text{LS}(L)$  and  $\text{H}(L)$  are possible. Indeed, if  $L$  consists of existential first-order sentences, then any  $\varphi \in L$  that has a model, has arbitrarily long models, whence  $\text{H}(L) = 0$ . Clearly  $\text{LS}(L)$  can be arbitrarily large for such an  $L$ . On the other hand, if  $L$  consists of universal first-order sentences, then any satisfiable  $\varphi \in L$  has a model of length 1, whence  $\text{LS}(L) \leq 1$ . If  $L$  contains, e.g., the sentence  $\forall x_0 \dots \forall x_\ell \bigvee_{i < j \leq \ell} x_i = x_j$  for  $\ell > 1$ ,

then  $H(L) \geq \ell > \text{LS}(L)$ . Finally, combining existential and universal sentences it is easy to construct a finite fragment  $L$  of FO such that  $\text{LS}(L) = H(L)$ .

### 3 Upper Bounds for the Length of Definable Words

#### 3.1 Definability and Types

It is well-known that equivalence of words up to a quantifier rank is preserved in catenation:

**Theorem 1.** *Let  $L \in \{\text{FO}_k, \text{MSO}_k\}$  for some  $k \in \mathbb{N}$ . Assume that  $v, v', w, w' \in \Sigma^+$  are words such that  $v \equiv_L v'$  and  $w \equiv_L w'$ . Then  $vw \equiv_L v'w'$ .*

*Proof.* The claim is proved by a straightforward Ehrenfeucht-Fraïssé game argument (see Proposition 2.1.4 in [2]).

Using Theorem 1, we get the following upper bounds for the numbers  $\mu(\varphi)$  and  $\nu(\varphi)$  in terms of the quantifier rank of  $\varphi$ :

**Proposition 2.** *Let  $L \in \{\text{FO}_k, \text{MSO}_k\}$  for some  $k \in \mathbb{N}$ . If  $\varphi$  is a sentence of  $L$ , then  $\mu(\varphi), \nu(\varphi) \leq N_L$ .*

*Proof.* If  $|w| \leq N_L$  for all words  $w \in \Sigma^+$  such that  $w \models \varphi$ , the claim is trivial. Assume then that  $w \models \varphi$  and  $|w| > N_L$ . Then there are two initial segments  $u$  and  $u'$  of  $w$  such that  $|u| < |u'|$  and  $u \equiv_L u'$ . Let  $v$  and  $v'$  be the corresponding end segments, i.e.,  $w = uv = u'v'$ . Then by Theorem 1,  $uv' \equiv_L u'v' = w$ , and similarly  $u'v \equiv_L uv = w$ , whence  $uv' \models \varphi$  and  $u'v \models \varphi$ .

Since  $|uv'| < |w|$ , we see that  $w$  is not the shortest word satisfying  $\varphi$ . The argument applies to any word  $w$  with  $|w| > N_L$ , whence we conclude that  $\mu(\varphi) \leq N_L$ . On the other hand  $|u'v| > |w|$ , whence  $w$  is neither the longest word satisfying  $\varphi$ . Applying this argument repeatedly, we see that  $\varphi$  is satisfied in arbitrarily long words, whence  $\nu(\varphi) = 0 \leq N_L$ .

From Propositions 1 and 2 we immediately obtain the following upper bound for the definability numbers of quantifier rank fragments of MSO:

**Corollary 1.** *Let  $k \in \mathbb{N}$  and  $L \in \{\text{FO}_k, \text{MSO}_k\}$ . Then  $\text{LS}(L), H(L) \leq N_L$ , and consequently  $\text{DN}(L) \leq N_L$ .*

This  $N_L$  upper bound for the definability, Löwenheim-Skolem and Hanf numbers shows that the quantifier rank fragments  $L$  of FO and MSO behave quite tamely on words: Clearly every type  $\text{tp}_L(w)$  is definable by a sentence of  $L$ , whence the number of non-equivalent sentences in  $L$  is  $2^{N_L}$ . Thus, any collection of representatives of non-equivalent sentences of  $L$  necessarily contains sentences of size close to  $N_L$ . But in spite of this, it is not possible to define words that are longer than  $N_L$  by sentences of  $L$ .

This shows that quantifier rank is not a good starting point if we want to prove interesting succinctness results for definability. Hence we turn our attention to the size  $n$  fragments  $\text{FO}[n]$  and  $\text{MSO}[n]$ . Note first that for any  $n \in \mathbb{N}$ ,  $\text{FO}[n]$  is trivially contained in  $\text{FO}_n$ , and similarly,  $\text{MSO}[n]$  is contained in  $\text{MSO}_n$ . A simple argument shows that this can be improved by a factor of 2:

**Lemma 1.** *For any  $n \in \mathbb{N}$ ,  $\text{FO}[2n] \leq \text{FO}_n$  and  $\text{MSO}[2n] \leq \text{MSO}_n$ .*

*Proof.* (Idea) Any sentence  $\varphi$  with quantifier rank  $n$  is equivalent to one with smaller quantifier rank unless it contains atomic formulas of the form  $x < y$  mentioning each quantified variable, and more than one of them at least twice. Counting the quantifiers, the atomic formulas, and the connectives needed, we see that  $\text{sz}(\varphi) \geq 2n$ .

Note that we have not tried to be optimal in the formulation of Lemma 1. We believe that with a more careful analysis,  $2n$  could be replaced with  $3n$ , and possibly with an even larger number.

**Corollary 2.** *For any  $n \in \mathbb{N}$ ,  $\text{DN}(\text{FO}[2n]), \text{LS}(\text{FO}[2n]), \text{H}(\text{FO}[2n]) \leq N_{\text{FO}_n}$  and  $\text{DN}(\text{MSO}[2n]), \text{LS}(\text{MSO}[2n]), \text{H}(\text{MSO}[2n]) \leq N_{\text{MSO}_n}$ .*

### 3.2 Number of Types

As we have seen in the previous section, the numbers of  $\text{FO}_k$ -types and  $\text{MSO}_k$ -types give upper bounds for the corresponding definability, Löwenheim-Skolem and Hanf-numbers. It is well known that on finite relational structures, for  $\text{FO}_k$  this number is bound above by an exponential tower of height  $k + 1$  with a polynomial, that depends on the vocabulary, on top (see, e.g., [10] for the case of graphs). It is straightforward to generalize this type of upper bound to  $\text{MSO}_k$ . On the class of  $\Sigma$ -words, we can prove the following explicit upper bounds. For the proof of this result, see the Appendix in the pre-print [6].

**Theorem 2.** *For any  $k \in \mathbb{N}$ ,  $N_{\text{FO}_k} \leq \text{twr}(k + \log^*(k^2 + k) + 1)$  and  $N_{\text{MSO}_k} \leq \text{twr}(k + \log^*((k + 1)^2) + 1)$ .*

By Corollary 1, we obtain the same upper bounds for the definability, Löwenheim-Skolem and Hanf numbers of the quantifier rank fragments.

**Corollary 3.** *For any  $k \in \mathbb{N}$ ,  $\text{DN}(\text{FO}_k), \text{LS}(\text{FO}_k), \text{H}(\text{FO}_k) \leq \text{twr}(k + \log^*(k^2 + k) + 1)$  and  $\text{DN}(\text{MSO}_k), \text{LS}(\text{MSO}_k), \text{H}(\text{MSO}_k) \leq \text{twr}(k + \log^*((k + 1)^2) + 1)$ .*

As we discussed after Corollary 1, from the point of view of succinctness it is more interesting to consider the definability numbers of the size fragments of FO and MSO than those of the quantifier rank fragments. Using Corollary 2, we obtain the following upper bounds for  $\text{FO}[n]$  and  $\text{MSO}[n]$ .

**Corollary 4.** *For any  $n \in \mathbb{N}$ ,  $\text{DN}(\text{FO}[n]), \text{LS}(\text{FO}[n]), \text{H}(\text{FO}[n]) \leq \text{twr}(n/2 + \log^*((n/2)^2 + n/2) + 1)$  and  $\text{DN}(\text{MSO}[n]), \text{LS}(\text{MSO}[n]), \text{H}(\text{MSO}[n]) \leq \text{twr}(n/2 + \log^*((n/2 + 1)^2) + 1)$ .*

In the next two sections we will prove lower bounds for the definability numbers of  $\text{FO}[n]$  and  $\text{MSO}[n]$  by providing explicit polynomial size sentences that define words that are of exponential tower length.

## 4 Lower Bounds for FO

In order to obtain a lower bound for  $\text{DN}(\text{FO}[n])$  we need a relatively small FO-formula that defines a long word. The long word we define has to do with the cumulative hierarchy of finite sets.

The finite levels  $V_i$  of the cumulative hierarchy are defined by  $V_0 = \emptyset$  and  $V_{i+1} = \mathcal{P}(V_i)$ . We represent finite sets as words using only braces  $\{$  and  $\}$  in a straightforward fashion. For example  $V_0$  is encoded as  $\{\}$  and  $V_1$  as  $\{\{\}\}$ .  $V_2$  has two possible encodings:  $\{\{\}\{\}\}$  and  $\{\{\{\}\}\}$ . It is well known that  $|V_{i+1}| = \text{twr}(i)$ . Thus the encodings of  $V_{i+1}$  have length at least  $\text{twr}(i)$ . We will define one such word via an FO-formula of polynomial size with respect to  $i$ .

For readability, we define  $L(x) := P_l(x)$  and  $R(x) := P_r(x)$  that say  $x$  is a left or right brace, respectively. We also define  $S(x, y) := x < y \wedge \neg \exists z(x < z < y)$  that says  $y$  is the successor of  $x$ .

As each set in the encoding can be identified by its outermost braces, the formula mostly operates on pairs of variables. For readability we adopt the convention  $\bar{x} := (x_1, x_2)$ , and similarly for different letters, to denote these pairs. To ensure that our formula defines a single encoding of  $V_i$ , we also define a linear order on encoded sets and require that the elements are in that order.

We define our formula recursively in terms of many subformulas. We briefly list the meanings and approximate sizes of each subformula involved:

- $\text{core}(\bar{x}, \theta(s, t))$ : the common core formula used in the formulas  $\text{set}_i$  and  $\text{oset}_i$  defined below. States that every brace  $y$  between  $x_1$  and  $x_2$  has a pair  $z$  such that the pair satisfies  $\theta$ . In practice,  $\theta$  will be another step of a similar recursion. The variables  $s$  and  $t$  are used to deal with both cases  $y < z$  and  $z < y$  at once, making the formula smaller.

$$\begin{aligned} \text{core}(\bar{x}, \theta(s, t)) &:= x_1 < x_2 \wedge L(x_1) \wedge R(x_2) \\ &\wedge \forall y(x_1 < y < x_2 \rightarrow \exists z(x_1 < z < x_2 \wedge y \neq z) \\ &\wedge \exists s \exists t((y < z \rightarrow (s = y \wedge t = z)) \\ &\wedge (z < y \rightarrow (s = z \wedge t = y)) \wedge \theta(s, t))) \end{aligned}$$

- $\text{set}_i(\bar{x})$ :  $\bar{x}$  correctly encodes a set in  $V_i$ , possibly with repetition. Size linear in  $i$ .

$$\begin{aligned} \text{set}_0(\bar{x}) &:= L(x_1) \wedge R(x_2) \wedge S(x_1, x_2) \\ \text{set}_{i+1}(\bar{x}) &:= \text{core}(\bar{x}, \text{set}_i(s, t)) \end{aligned}$$

- $\bar{x} \in_i \bar{y}$ :  $\bar{x}$  is an element of  $\bar{y}$ . Size linear in  $i$ . Assumes that  $\bar{x}$  encodes a set in  $V_i$  and  $\bar{y}$  encodes a set in  $V_{i+1}$ . The part with  $\bar{z}$  is used to ensure that  $\bar{x}$  is an element of  $\bar{y}$  and not for example an element of an element.

$$\bar{x} \in_i \bar{y} := y_1 < x_1 < x_2 < y_2 \wedge \neg \exists \bar{z}(\text{set}_i(\bar{z}) \wedge y_1 < z_1 < x_1 \wedge x_2 < z_2 < y_2)$$

- $\bar{x} \sim_i \bar{y}$ :  $\bar{x}$  and  $\bar{y}$  encode the same set, possibly in a different order. Size  $\mathcal{O}(i^2)$ . Assumes  $\bar{x}$  and  $\bar{y}$  encode sets in  $V_i$ . The two implications on the second line

are used to deal with the symmetry of  $\bar{x}$  and  $\bar{y}$  at once, making the formula smaller.

$$\begin{aligned}\bar{x} \sim_0 \bar{y} &:= \top \\ \bar{x} \sim_{i+1} \bar{y} &:= \forall \bar{a}(\text{set}_i(\bar{a}) \rightarrow \exists \bar{b}(\text{set}_i(\bar{b}) \\ &\quad \wedge (\bar{a} \in_i \bar{x} \rightarrow \bar{b} \in_i \bar{y}) \wedge (\bar{a} \in_i \bar{y} \rightarrow \bar{b} \in_i \bar{x}) \wedge \bar{a} \sim_i \bar{b}))\end{aligned}$$

- $\bar{x} \prec_i \bar{y}$ : the  $\prec_{i-1}$ -greatest element of the symmetric difference of  $\bar{x}$  and  $\bar{y}$  is in  $\bar{y}$ . Size  $\mathcal{O}(i^3)$ . Defines a linear order for encoded sets in  $V_i$ . The set  $\bar{z}$  is in  $\bar{y}$ , is not in  $\bar{x}$  and is larger than any  $\bar{a}$  in  $\bar{x}$ .

$$\begin{aligned}\bar{x} \prec_0 \bar{y} &:= \perp \\ \bar{x} \prec_{i+1} \bar{y} &:= \exists \bar{z}(\text{set}_i(\bar{z}) \wedge \bar{z} \in_i \bar{y} \wedge \forall \bar{a}((\text{set}_i(\bar{a}) \wedge \bar{a} \in_i \bar{x}) \\ &\quad \rightarrow (\bar{a} \sim_i \bar{z} \wedge (\forall \bar{b}((\text{set}_i(\bar{b}) \wedge \bar{b} \in_i \bar{y}) \rightarrow \bar{a} \sim_i \bar{b}) \rightarrow \bar{a} \prec_i \bar{z}))))\end{aligned}$$

- $\text{oset}_i(\bar{x})$ :  $\bar{x}$  correctly encodes a set in  $V_i$  with no repetition and with the elements in the linear order given by the formula  $\bar{x} \prec_i \bar{y}$ . Size  $\mathcal{O}(i^4)$ . Ensures that only a singular word satisfies our formula.

$$\begin{aligned}\text{oset}_0(\bar{x}) &:= L(x_1) \wedge R(x_2) \wedge S(x_1, x_2) \\ \text{oset}_{i+1}(\bar{x}) &:= \text{core}(\bar{x}, \text{oset}_i(s, t)) \wedge \forall \bar{a} \forall \bar{b}((\text{set}_i(\bar{a}) \wedge \text{set}_i(\bar{b}) \\ &\quad \wedge \bar{a} \in_i \bar{x} \wedge \bar{b} \in_i \bar{x} \wedge a_1 < b_1) \rightarrow \bar{a} \prec_i \bar{b})\end{aligned}$$

- $\text{add}_i(\bar{x}, \bar{y}, \bar{z})$ : States that  $\bar{x} = \bar{y} \cup \{\bar{z}\}$ . Size  $\mathcal{O}(i^2)$ . Assumes  $\bar{x}$  and  $\bar{y}$  encode sets in  $V_i$  and  $\bar{z}$  encodes a set in  $V_{i-1}$ . The first line states that  $\bar{y} \subseteq \bar{x}$ , the second line states  $\bar{z} \in \bar{x}$  and the two final lines state  $\bar{x} \setminus \{\bar{z}\} \subseteq \bar{y}$ .

$$\begin{aligned}\text{add}_{i+1}(\bar{x}, \bar{y}, \bar{z}) &:= \forall \bar{a}((\text{set}_i(\bar{a}) \wedge \bar{a} \in_i \bar{y}) \rightarrow \exists \bar{b}(\text{set}_i(\bar{b}) \wedge \bar{b} \in_i \bar{x} \wedge \bar{a} \sim_i \bar{b})) \\ &\quad \wedge \exists \bar{c}(\text{set}_i(\bar{c}) \wedge \bar{c} \in_i \bar{x} \wedge \bar{c} \sim_i \bar{z}) \\ &\quad \wedge \forall \bar{d}((\text{set}_i(\bar{d}) \wedge \bar{d} \in_i \bar{x} \wedge d_1 \neq c_1) \\ &\quad \rightarrow \exists \bar{e}(\text{set}_i(\bar{e}) \wedge \bar{e} \in_i \bar{y} \wedge \bar{e} \sim_i \bar{d}))\end{aligned}$$

- $V_i(\bar{x})$ :  $\bar{x}$  encodes the set  $V_i$ . Size  $\mathcal{O}(i^5)$ . States that  $\bar{x}$  is an ordered encoding,  $\emptyset \in \bar{x}$ ,  $V_{i-1} \in \bar{x}$  and for all  $\bar{c} \in \bar{x}$  and  $\bar{d} \in V_{i-1}$ , we have  $\bar{c} \cup \{\bar{d}\} \in \bar{x}$ .

$$\begin{aligned}V_0(\bar{x}) &:= \text{set}_0(\bar{x}) \\ V_{i+1}(\bar{x}) &:= \text{oset}_{i+1}(\bar{x}) \wedge \exists \bar{a}(V_0(\bar{a}) \wedge S(x_1, a_1)) \wedge \exists \bar{b}(V_i(\bar{b}) \wedge S(b_2, x_2) \\ &\quad \wedge \forall \bar{c} \forall \bar{d}((\text{set}_i(\bar{c}) \wedge \bar{c} \in_i \bar{x} \wedge \text{set}_{i-1}(\bar{d}) \wedge \bar{d} \in_{i-1} \bar{b}) \\ &\quad \rightarrow \exists \bar{e}(\text{set}_i(\bar{e}) \wedge \bar{e} \in_i \bar{x} \wedge \text{add}_i(\bar{e}, \bar{c}, \bar{d})))\end{aligned}$$

- $\psi_i$ : the entire word is the ordered encoding of the set  $V_i$ . Size  $\mathcal{O}(i^5)$ .

$$\psi_i := \exists x \exists y \forall z (x \leq z \wedge z \leq y \wedge V_i(x, y))$$

The formula  $\psi_{i+1}$  defines a word  $w$  that, as an encoding of the set  $V_{i+1}$ , has length at least  $\text{twr}(i)$ . The size of  $\psi_{i+1}$  is  $\mathcal{O}((i+1)^5)$  and thus  $\mathcal{O}(i^5)$ . Let  $c$  be a constant such that  $\text{sz}(\psi_{i+1}) \leq c \cdot i^5$  so  $w \in \text{Def}(\text{FO}[c \cdot i^5])$ . As we want to relate the length of  $w$  to the size of  $\psi_i$ , we set  $n = c \cdot i^5$  and obtain the following result:

**Theorem 3.** *For some constant  $c \in \mathbb{N}$  there are infinitely many  $n \in \mathbb{N}$  satisfying*

$$\text{DN}(\text{FO}[n]) \geq \text{twr}(\sqrt[5]{n/c}).$$

Proposition 1 immediately gives the same bound for the Hanf number.

**Corollary 5.** *For some constant  $c \in \mathbb{N}$  there are infinitely many  $n \in \mathbb{N}$  satisfying*

$$\text{H}(\text{FO}[n]) \geq \text{twr}(\sqrt[5]{n/c}).$$

By omitting the subformula  $\text{oset}_{i+1}$  from the above we get a formula of size  $\mathcal{O}(i^3)$  that is no longer satisfied by only one word but still only has large models. With this formula we obtain a lower bound for the Löwenheim-Skolem number.

**Corollary 6.** *For some  $c \in \mathbb{N}$  there are arbitrarily large  $n \in \mathbb{N}$  satisfying*

$$\text{LS}(\text{FO}[n]) \geq \text{twr}(\sqrt[3]{n/c}).$$

## 5 Lower Bounds for MSO

In this section, we define a similar formula for MSO as we did above for FO. The formula again defines an encoding of  $V_i$  but for MSO our formula is of size  $\mathcal{O}(i^2)$  compared to the  $\mathcal{O}(i^5)$  of FO. We achieve this by quantifying a partition of so called levels for the braces and thus the encoded sets and using a different method to define only a single encoding.

The level of the entire encoded set will be equal to the maximum depth of braces inside the set. The level of an element of a set will always be one less than the level of the parent set. This means that there will be instances of the same set with different levels in our encoding. For example in the encoding  $\{\{\}\{\{\}\}\}$  the outermost braces are level 2, both of the elements are level 1 and the empty set in the second element is level 0.

We again define our formula in terms of many subformulas and briefly list the meaning and size of each subformula:

- $\text{set}_i(\bar{x})$ :  $\bar{x}$  encodes a set of level  $i$ . Size constant. Here we only require that there are no braces of the same level between  $x_1$  and  $x_2$ , leaving the rest to the formula  $\text{levels}_i$  below.

$$\begin{aligned} \text{set}_0(\bar{x}) &:= S(x_1, x_2) \wedge L(x_1) \wedge R(x_2) \wedge D_0(x_1) \wedge D_0(x_2) \\ \text{set}_i(\bar{x}) &:= x_1 < x_2 \wedge L(x_1) \wedge R(x_2) \wedge D_i(x_1) \wedge D_i(x_2) \\ &\quad \wedge \forall y (x_1 < y < x_2 \rightarrow \neg D_i(y)) \end{aligned}$$



- **levels<sub>i</sub>**: The relations  $D_j$  define the levels of sets as intended and there are no odd braces without pairs. Size  $\mathcal{O}(i^2)$ . States that every brace has a level, no brace has two different levels, every set encloses only braces of lower levels and every brace has a pair of the same level to form a set.

$$\begin{aligned}
 \text{levels}_i := & \forall x \left( \bigvee_{j=0}^i D_j(x) \wedge \bigwedge_{\substack{j,k \in \{0, \dots, i\} \\ j \neq k}} \neg(D_j(x) \wedge D_k(x)) \right. \\
 & \wedge \forall \bar{x} \left( \bigwedge_{j=0}^i (\text{set}_j(\bar{x}) \rightarrow \forall y (x_1 < y < x_2 \rightarrow \bigvee_{k=0}^{j-1} D_k(y))) \right) \\
 & \wedge \forall x_1 \left( \bigwedge_{j=0}^i ((L(x_1) \wedge D_j(x_1)) \rightarrow \exists x_2 \text{set}_j(x_1, x_2)) \right) \\
 & \wedge \bigwedge_{j=0}^i (R(x_1) \wedge D_j(x_1)) \rightarrow \exists x_2 \text{set}_j(x_2, x_1) \left. \right)
 \end{aligned}$$

- $\bar{x} \in \bar{y}$ :  $\bar{x}$  is an element of  $\bar{y}$ . Size constant. Assumes  $\bar{x}$  encodes a set of level  $i$  and  $\bar{y}$  encodes a set of level  $i - 1$ .

$$\bar{x} \in \bar{y} := y_1 < x_1 \wedge x_2 < y_2$$

- $\bar{x} \sim_i \bar{y}$ :  $\bar{x}$  and  $\bar{y}$  encode the same set. Size linear in  $i$ . Assumes  $\bar{x}$  and  $\bar{y}$  encode sets of level  $i$ . Similar to the FO case.

$$\begin{aligned}
 \bar{x} \sim_0 \bar{y} &:= \top \\
 \bar{x} \sim_{i+1} \bar{y} &:= \forall \bar{a} (\text{set}_i(\bar{a}) \rightarrow \exists \bar{b} (\text{set}_i(\bar{b}) \\
 & \wedge (\bar{a} \in \bar{x} \rightarrow \bar{b} \in \bar{y}) \wedge (\bar{a} \in \bar{y} \rightarrow \bar{b} \in \bar{x}) \wedge \bar{a} \sim_i \bar{b}))
 \end{aligned}$$

- **add<sub>i</sub>( $\bar{x}, \bar{y}, \bar{z}$ )**: States that  $\bar{x} = \bar{y} \cup \{\bar{z}\}$ . Size linear in  $i$ . Assumes  $\bar{x}$  and  $\bar{y}$  encode sets of level  $i$  and  $\bar{z}$  encodes a set of level  $i - 1$ . Similar to the FO case.

$$\begin{aligned}
 \text{add}_{i+1}(\bar{x}, \bar{y}, \bar{z}) := & \forall \bar{a} ((\text{set}_i(\bar{a}) \wedge \bar{a} \in \bar{y}) \rightarrow \exists \bar{b} (\text{set}_i(\bar{b}) \wedge \bar{b} \in \bar{x} \wedge \bar{a} \sim_i \bar{b})) \\
 & \wedge \exists \bar{c} (\text{set}_i(\bar{c}) \wedge \bar{c} \in \bar{x} \wedge \bar{c} \sim_i \bar{z}) \\
 & \wedge \forall \bar{d} ((\text{set}_i(\bar{d}) \wedge \bar{d} \in \bar{x} \wedge d_1 \neq c_1) \\
 & \rightarrow \exists \bar{e} (\text{set}_i(\bar{e}) \wedge \bar{e} \in \bar{y} \wedge \bar{e} \sim_i \bar{d}))
 \end{aligned}$$

- $V_i(\bar{x})$ :  $\bar{x}$  encodes the set  $V_i$ . Size  $\mathcal{O}(i^2)$ . Assumes the level partition is given. Similar to the FO case with no ordering.

$$\begin{aligned}
 V_0(\bar{x}) &:= \text{set}_0(\bar{x}) \\
 V_{i+1}(\bar{x}) &:= \text{set}_{i+1}(\bar{x}) \wedge \exists \bar{a} (\text{set}_i(\bar{a}) \wedge \bar{a} \in \bar{x} \wedge S(a_1, a_2)) \\
 & \wedge \exists \bar{b} (V_i(\bar{b}) \wedge \bar{b} \in \bar{x} \wedge \forall \bar{c} \forall \bar{d} ((\text{set}_i(\bar{c}) \wedge \bar{c} \in \bar{x} \wedge \text{set}_{i-1}(\bar{d}) \wedge \bar{d} \in \bar{b}) \\
 & \rightarrow \exists \bar{e} (\text{set}_i(\bar{e}) \wedge \bar{e} \in \bar{x} \wedge \text{add}_i(\bar{e}, \bar{c}, \bar{d}))))
 \end{aligned}$$

- $\varphi_i(x, y)$ : Quantifies the level partition and states the subword from  $x$  to  $y$  encodes  $V_i$ . Size  $\mathcal{O}(i^2)$ .

$$\varphi_i(x, y) := \exists D_0 \dots \exists D_i(\text{levels}_i \wedge V_i(x, y))$$

We now have a formula  $\varphi_i(x, y)$  that says the subword from  $x$  to  $y$  encodes the set  $V_i$ . There are still multiple words that satisfy this formula, since different orders of the sets and even repetition are still allowed. To pick out only one such word, we use a lexicographic order, where a shorter word always precedes a longer one.

Let  $\varphi'_i$  be the formula obtained from  $\varphi_i$  by replacing each occurrence of  $L(x)$  with  $P_1(x)$  and  $R(x)$  with  $P_2(x)$ . We define the final formula  $\psi_i$  of size  $\mathcal{O}(i^2)$  that says the entire word model is the least word in the lexicographic order that satisfies the property of  $\varphi_i$ . We check that no lexicographically smaller word satisfies  $\varphi_i$  by quantifying the word under consideration on top of the same word model using the variables  $P_1$  and  $P_2$  for the two letters. We first ensure that  $P_1$  and  $P_2$  partition the model and then use  $y'$  as the cut-off point for the possibly shorter word we want to quantify. If  $y' = y$  we check the lexicographic order with  $z$  as the first different symbol. Finally we state that the quantified word does not satisfy  $\varphi_i$ .

$$\begin{aligned} \psi_i := & \exists x \exists y (\forall z (x \leq z \wedge z \leq y) \wedge \varphi_i(x, y) \\ & \wedge \forall P_1 \forall P_2 (\forall z ((P_1(z) \vee P_2(z)) \wedge \neg(P_1(z) \wedge P_2(z))) \\ & \wedge \forall y' ((y' < y \vee \exists z (\forall a (a < z \rightarrow (L(a) \leftrightarrow P_1(a) \wedge R(a) \leftrightarrow P_2(a))) \\ & \wedge (P_1(z) \wedge R(z))) \rightarrow \neg \varphi'_i(x, y')))) \end{aligned}$$

We have used the lexicographic order here to select only one of the possible words that satisfy our property. Note that this can be done for any property. The size of such a formula will depend polynomially on the size of the alphabet, as well as linearly on the size of the formula defining the property in question.

We obtain the lower bound for the definability number as in the FO case.

**Theorem 4.** *For some constant  $c \in \mathbb{N}$  there are infinitely many  $n \in \mathbb{N}$  satisfying*

$$\text{DN}(\text{MSO}[n]) \geq \text{twr}(\sqrt{n/c}).$$

We get the same bounds for  $\text{LS}(\text{MSO}[n])$  and  $\text{H}(\text{MSO}[n])$  via Proposition 1.

**Corollary 7.** *For some constant  $c \in \mathbb{N}$  there are infinitely many  $n \in \mathbb{N}$  satisfying*

$$\text{LS}(\text{MSO}[n]), \text{H}(\text{MSO}[n]) \geq \text{twr}(\sqrt{n/c}).$$

## 6 Conclusion

We considered the definability number, the Löwenheim-Skolem number and the Hanf number on words in the size  $n$  fragments of first-order logic and monadic

second-order logic. We obtained exponential towers of various heights as upper and lower bounds for each of these numbers.

For FO, we obtained the bounds

$$\text{twr}(\sqrt[5]{n/c}) \leq \text{DN}(\text{FO}[n]) \leq \text{twr}(n/2 + \log^*((n/2)^2 + n/2) + 1)$$

for some constant  $c$ . As corollaries, we obtained the same bounds for  $\text{LS}(\text{FO}[n])$  and  $\text{H}(\text{FO}[n])$ . In addition, by modifying the formula we used for the lower bounds, we obtained a slightly better lower bound of  $\text{twr}(\sqrt[3]{n/c})$  for  $\text{LS}(\text{FO}[n])$ .

In the case of MSO, the bounds are similarly

$$\text{twr}(\sqrt{n/c}) \leq \text{DN}(\text{MSO}[n]) \leq \text{twr}(n/2 + \log^*((n/2 + 1)^2) + 1)$$

for a different constant  $c$ . We again immediately obtained the same bounds for  $\text{LS}(\text{MSO}[n])$  and  $\text{H}(\text{MSO}[n])$ .

The gaps between the lower bounds and upper bounds we have proved are quite big. In absolute terms, they are actually huge, as each upper bound is non-elementary with respect to the corresponding lower bound. However, it is more fair to do the comparison in the iterated logarithmic scale, which reduces the gap to be only polynomial. Nevertheless, a natural task for future research is to look for tighter lower and upper bounds.

Finally, we remark that the technique for proving an exponential tower upper bound for the number of types in the quantifier rank fragments of some logic  $\mathcal{L}$  is completely generic: it works in the same way irrespective of the type of quantifiers allowed in  $\mathcal{L}$ . Thus, it can be applied for example in the case where  $\mathcal{L}$  is the extension of FO with some generalized quantifier (or a finite set of generalized quantifiers). Assuming further that the quantifier rank fragments  $L$  of  $\mathcal{L}$  satisfy Theorem 1, we can obtain this way an exponential tower upper bound for the numbers  $\text{DN}(L)$ ,  $\text{LS}(L)$  and  $\text{H}(L)$ . On the other hand, note that if the quantifier rank fragments  $L$  satisfy Theorem 1, then each  $\equiv_L$  is an invariant equivalence relation, whence  $\mathcal{L}$  can only define regular languages. Therefore it seems that our technique for proving upper bounds cannot be used for logics with expressive power beyond regular languages.

## References

1. Ebbinghaus, H.D.: Löwenheim-Skolem theorems. In: Gabbay, D., Thagard, P., Woods, J., Jacquette, D. (eds.) *Philosophy of Logic, Handbook of the Philosophy of Science*. Elsevier Science (2006)
2. Ebbinghaus, H., Flum, J.: *Finite model theory. Perspectives in Mathematical Logic*, Springer, Germany (1995)
3. Ellul, K., Krawetz, B., Shallit, J., Wang, M.W.: Regular expressions: new results and open problems. *J. Autom. Lang. Comb.* **10**(4), 407–437 (2005). <https://doi.org/10.25596/jalc-2005-407>
4. Grohe, M.: Some remarks on finite Löwenheim-Skolem theorems. *Math. Log. Q.* **42**, 569–571 (1996). <https://doi.org/10.1002/malq.19960420145>

5. Grohe, M.: Large finite structures with few  $L^k$ -types. *Inf. Comput.* **179**(2), 250–278 (2002). <https://doi.org/10.1006/inco.2002.2954>
6. Hella, L., Vilander, M.: Defining long words succinctly in FO and MSO (2022). <https://doi.org/10.48550/arxiv.2202.10180>, pre-print
7. Libkin, L.: Elements of finite model theory. Texts in Theoretical Computer Science. An EATCS Series, Springer (2004). <https://doi.org/10.1007/978-3-662-07003-1>
8. Pikhurko, O., Spencer, J., Verbitsky, O.: Succinct definitions in the first order theory of graphs. *Ann. Pure Appl. Log.* **139**(1–3), 74–109 (2006). <https://doi.org/10.1016/j.apal.2005.04.003>
9. Pikhurko, O., Verbitsky, O.: Descriptive complexity of finite structures: saving the quantifier rank. *J. Symb. Log.* **70**(2), 419–450 (2005). <https://doi.org/10.2178/jsl/1120224721>
10. Pikhurko, O., Verbitsky, O.: Logical complexity of graphs: a survey. In: Grohe, M., Makowsky, J.A. (eds.) *Model Theoretic Methods in Finite Combinatorics - AMS-ASL Joint Special Session*, Washington, DC, 5–8 January 2009. *Contemporary Mathematics*, vol. 558, pp. 129–180. American Mathematical Society (2009)
11. Reinhardt, K.: The complexity of translating logic to finite automata. In: Grädel, E., Thomas, W., Wilke, T. (eds.) *Automata Logics, and Infinite Games*. LNCS, vol. 2500, pp. 231–238. Springer, Heidelberg (2002). [https://doi.org/10.1007/3-540-36387-4\\_13](https://doi.org/10.1007/3-540-36387-4_13)
12. Stockmeyer, L.J.: The complexity of decision problems in automata theory and logic. Ph.D. thesis, Massachusetts Institute of Technology (1974)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





