Algorithmic Model Theory
SS 2010

Prof. Dr. Erich Grädel

Mathematische Grundlagen der Informatik
RWTH Aachen

Contents

1 The classical decision problem for FO

The classical decision problem for first-order logic was considered the main problem of mathematical logic by Hilbert and Ackermann and its undecidability was shown by Church and Turing.

> The Entscheidungsproblem is solved when we know a procedure that allows for any given logical expression to decide by finitely many operations its validity or satisfiability. [...] The Entscheidungsproblem must be considered the main problem of mathematical logic.
>
> (D. Hilbert and W. Ackermann, 1928)

We introduce the classical decision problem for first-order logic, for which we present three equivalent formulations. The importance of the decision problem for first-order logic results from the fact that first-order logic provides a framework to express almost all aspects of mathematics.

Satisfiability: Construct an algorithm that decides for any given formula of FO whether it has a model.

Validity: Construct an algorithm that decides for any given formula of FO whether it is valid, i.e. whether it holds in all models where it is defined.

Provability: Construct an algorithm that decides for any given formula ψ of FO whether $\vdash \psi$, meaning ψ is provable from the empty set of axioms in some formal system, e.g. sequential calculus.

Since ψ is satisfiable if and only if $\neg\psi$ is not valid, satisfiability and validity are equivalent problems with respect to computability. The equivalence with provability is a much more intricate result and in fact a consequence of the following

Theorem 1.1 (Completeness Theorem (Gödel))**.** For any given set of sentences $\Phi \subseteq \mathrm{FO}(\tau)$ and any sentence $\psi \in \mathrm{FO}(\tau)$ it holds that

$$\Phi \models \psi \iff \Phi \vdash \psi \,;$$

in particular $\emptyset \models \psi \Leftrightarrow \emptyset \vdash \psi$.

As a direct consequence we get the following

Theorem 1.2. The set of valid first-order formulae is recursively enumerable.

1.1 Basic notions on decidability

In our formulation of the decision problem it was not precisely specified what an algorithm is. It was not until the 1930s that Church and Kleene , Gödel and Turing provided a precise definition of an abstract algorithm. Their approaches are today known to be equivalent. We introduce the concept of a Turing machine.

Definition 1.3. A *Turing machine* (TM) M is a 6-tuple $M = (Q, \Sigma, \Gamma, q_0, F, \delta)$, where

- Q denotes a finite set of states,
- Σ, Γ denote finite alphabets, where Σ is the working alphabet with a special blank symbol $\square \in \Sigma$,
- $\Gamma \subseteq \Sigma \setminus \{\square\}$ is the input alphabet,
- $q_0 \in Q$ denotes the initial state,
- $F \subseteq Q$ is the set of final states and
- $\delta : (Q \setminus F) \times \Sigma \to Q \times \Sigma \times \{-1, 0, 1\}$ is the transition function.

A *configuration* is an element $C = (q, p, w = w_0 w_1 \dots w_k) \in Q \times \mathbb{N} \times \Sigma^*$. The transition function δ induces a partial function on the set of all configurations

$$C \mapsto \mathrm{Next(C)},$$

where for $\delta(q, w_p) = (q', a, m)$, the successor configuration of C is defined as $\mathrm{Next(C)} = (q', p + m, w_0 \dots w_{p-1} a w_{p+1} \cdots w_k)$. A *computation* of the TM M on an input word $x \in \Gamma^*$ is a configuration

sequence

$$C_0, C_1, \ldots$$

where $C_0 = C_0(x) := (q_0, 0, x)$ is the input configuration and $C_{i+1} = \text{Next}(C_i)$ for all i.

M *halts* on x if the computation of M on x is finite, i.e. ends in a final configuration $C_f = (q, p, w)$ with $q \in F$.

The language accepted by M is

$$L(M) := \{x \in \Gamma^* : M \text{ halts on } x\}.$$

M computes a partial function $f_M : \Gamma^* \to \Sigma^*$ with domain $L(M)$ such that $f_M(x) = y$ if and only if the computation of M on x ends in (q, p, y) for some $q \in F$, $y \in \Sigma^*$ and $p \in \mathbb{N}$.

Definition 1.4. A *Turing acceptor* is a Turing machine M with $F = F^+ \uplus F^-$ where M *accepts* x if the computation of M on x ends in a state in F^+. M *rejects* x if the computation of M on x ends in a state in F^-.

Definition 1.5.

- $L \subseteq \Gamma^*$ is *recursively enumerable (r.e.)* if there exists a TM M with $L(M) = L$.
- $L \subseteq \Gamma^*$ is *co-recursively enumerable (co-r.e.)* if $\overline{L} := \Gamma^* \setminus L$ is r.e..
- A (partial) function $f : \Gamma^* \to \Sigma^*$ is *(Turing) computable* if there is a TM M with $f_M = f$.
- $L \subseteq \Gamma^*$ is *decidable* if there is a Turing acceptor M such that for all $x \in \Gamma^*$

  $$x \in L \Rightarrow M \text{ accepts } x$$

  $$x \notin L \Rightarrow M \text{ rejects } x$$

  or, equivalently, L is decidable if its characteristic function $\chi_L : \Gamma^* \to \{0, 1\}$ is Turing computable.

Theorem 1.6. A language $L \subseteq \Gamma^*$ is decidable if and only if L is r.e. and co-r.e.

Definition 1.7. Let $A \subseteq \Gamma^*, B \subseteq \Sigma^*$. We say that A is *(many-to-one) reducible* to B, $A \leq B$, if there is a total computable function $f : \Gamma^* \to \Sigma^*$ such that for all $x \in \Gamma^*$ we have $x \in A \Leftrightarrow f(x) \in B$.

Lemma 1.8.

- $A \leq B$, B decidable $\Rightarrow$ A decidable
- $A \leq B$, B r.e. $\Rightarrow$ A r.e.
- $A \leq B$, A undecidable $\Rightarrow$ B undecidable.

There surely are undecidable languages since there are only countably many Turing machines but uncountably many languages. Unfortunately, among these languages there are quite relevant classes of languages. For example we cannot even decide whether a TM halts on a given input.

Definition 1.9 (Halting Problems)**.** The *general halting problem* is defined as

$$H := \{\rho(M)\#\rho(x) : M \text{ Turing machine}, x \in L(M)\}$$

where $\rho(M)$ and $\rho(x)$ are encodings of the TM M and the input x over a fixed alphabet $\{0,1\}$ such that the computation of M on x can be reconstructed from the encodings $\rho(M)$ and $\rho(x)$ in an effective way.

There is a universal TM U which, given $\rho(M)$ and $\rho(x)$, simulates the computation of M on x and halts if and only if M halts on x. Thus, $L(U) = H$ from which we conclude that H is r.e..

We introduce two special variants of the halting problem

- *Self-application problem*

$$H_0 := \{\rho(M) : \rho(M) \in L(M)\}$$

- *Halting on the empty word*

$$H_\varepsilon := \{\rho(M) : \varepsilon \in L(M)\}$$

Theorem 1.10. H, H_0, H_ε are undecidable.

Proof.

- H_0 is not co-r.e. and thus undecidable. Otherwise $\overline{H_0} = L(M_0)$ for some TM M_0. Then

$$\rho(M_0) \in H_0 \Leftrightarrow M_0 \text{ halts on } \rho(M_0) \Leftrightarrow \rho(M_0) \in \overline{H_0}.$$

- H_0 is a special case of H, $H_0 \leq H$, and thus H is undecidable.
- We can reduce H to H_ε, thus H_ε is undecidable. Q.E.D.

As a consequence of the next theorem we cannot algorithmically prove whether a program computes a given function, i.e. we cannot algorithmically prove the correctness of a program. Note that this does not mean that we cannot prove the correctness of a single given program. Instead the statement is that we cannot do so algorithmically for all programs.

Theorem 1.11 (Rice)**.** Let $\mathcal{R}$ be the set of all computable functions and let $S \subseteq \mathcal{R}$ be a set of computable functions such that $S \neq \emptyset$ and $S \neq \mathcal{R}$. Then $\text{code}(S) := \{\rho(M) : f_M \in S\}$ is undecidable.

Proof. Let $\Uparrow$ be the everywhere undefined function, i.e. $\text{Def}(\Uparrow) = \emptyset$. Obviously, $\Uparrow$ is computable. Assume that $\Uparrow \notin S$ (otherwise consider $\mathcal{R} \setminus S$ instead of S. Clearly if $\text{code}(\mathcal{R} \setminus S)$ is undecidable then so is $\text{code}(S)$.)

As $S \neq \emptyset$, there exists a function $f \in S$. Let M_f be a TM that computes f, i.e. $f_{M_f} = f$. We define a reduction $H_\varepsilon \leq \text{code}(S)$ by describing a total computable function $\rho(M) \mapsto \rho(M')$ such that

$$M \text{ halts on } \varepsilon \Leftrightarrow f_{M'} \in S.$$

Specifically, given $\rho(M)$, we construct the encoding of a TM M' which, given an input x, proceeds as follows:

- first simulate M on ε (i.e. apply the universal TM U to $\rho(M)\#\varepsilon$);
- then simulate M_f on x (i.e. apply the universal TM U to $\rho(M_f)\#\rho(x)$).

It is clear that the reduction function is computable. Furthermore, if M halts on ε then $f_{M'}(x) = f(x)$ for all inputs x, i.e. $f_{M'} = f$, so $f_{M'} \in S$. If M does not halt on ε then M' does not halt on x for any x, i.e. $f_{M'} = \Uparrow$, so $f_{M'} \notin S$. Q.E.D.

Definition 1.12 (Recursive inseparability)**.** Let $A, B \subseteq \Gamma^*$ be two disjoint sets. We say that A and B are *recursively inseparable* if there exists no recursive set $C \subseteq \Gamma^*$ such that $A \subseteq C$ and $B \cap C = \emptyset$.

Example. $(A, \overline{A})$ are recursively inseparable if and only if A is undecidable.

Lemma 1.13. Let $A, B \subseteq \Gamma^*, A \cap B = \emptyset$ be recursively inseparable. Let $X, Y \subseteq \Sigma^*, X \cap Y = \emptyset$, and let f be a total computable function such that $f(A) \subseteq X$ and $f(B) \subseteq Y$. Then X and Y are recursively inseparable.

Proof. Assume there exists a decidable set $Z \subseteq \Sigma^*$ such that $X \subseteq Z$ and $Y \cap Z = \emptyset$. Consider $C = \{x \in \Gamma^* : f(x) \in Z\}$. C is decidable, $A \subseteq C, B \cap C = \emptyset$, thus C separates A, B. Q.E.D.

Notation: We write $(A, B) \leq (X, Y)$ if such a function f exists.

Example. $(A, \overline{A}) \leq (B, \overline{B}) \Leftrightarrow A \leq B$.

As a preparation to prove Trakhtenbrot's theorem, we consider a refinement of H_ε

$$H_\varepsilon^+ := \{\rho(M) : M \text{ accepts } \varepsilon\}$$
$$H_\varepsilon^- := \{\rho(M) : M \text{ rejects } \varepsilon\}$$
$$H_\varepsilon^\infty := \{\rho(M) : \text{the computation of } M \text{ on } \varepsilon \text{ is infinite and does not cycle.}\}$$

H_0^+, H_0^-, H_0^∞ are defined analogously, with respect to self-application.

Theorem 1.14. $H_\varepsilon^+, H_\varepsilon^-$ and H_ε^∞ are pairwise recursively inseparable.

Proof.

- $(H_\varepsilon^+, H_\varepsilon^\infty)$: We show that every set C with $H_\varepsilon^+ \subseteq C$ and $H_\varepsilon^\infty \cap C = \emptyset$ is undecidable by reducing the halting problem H_ε to C. Define the function $\rho(M) \mapsto \rho(M')$ as follows. From a given code $\rho(M)$ construct the code of a TM M' that simulates M and simultaneously counts the number of computation steps since the start. If M halts (accepting or rejecting), M' accepts.

  It is clear that the reduction function is computable. If M halts on ε then M' halts on ε as well and accepts, so $\rho(M') \in H_\varepsilon^+ \subseteq C$. If M does not halt on ε then M' does not halt either, so $\rho(M') \in H_\varepsilon^\infty$ and as $H_\varepsilon^\infty \cap C = \emptyset$, we have $\rho(M') \notin C$.

- The statement for H_ε^- and H_ε^∞ is proven analogously.

- $(H_\varepsilon^-, H_\varepsilon^+)$: Show that $(H_0^-, H_0^+) \leq (H_\varepsilon^-, H_\varepsilon^+)$ and that (H_0^-, H_0^+) are recursively inseparable.

  – $(H_0^-, H_0^+) \leq (H_\varepsilon^-, H_\varepsilon^+)$:

  For a given input TM M construct a TM M' that ignores its own input and simulates M on $\rho(M)$. Obviously, M' can be constructed effectively, say by a computable function h. Now $h(M)$ accepts ε iff M accepts $\rho(M)$ and $h(M)$ rejects ε iff M rejects $\rho(M)$.

  – (H_0^-, H_0^+) recursively inseparable:

  Assume there exists a decidable C with $H_0^- \subseteq C$ and $H_0^+ \subseteq \overline{C}$. Consider a machine M_0 that decides C. There are two cases:

  (1) M_0 accepts $\rho(M_0)$. Then $\rho(M_0) \in C$ by definition of M_0. Then $\rho(M_0) \notin H_0^+$ by definition of C. On the other hand, if M_0 accepts $\rho(M_0)$ then $\rho(M_0) \in H_0^+$ (by definition of H_0^+), a contradiction.

  (2) M_0 rejects $\rho(M_0)$. Then $\rho(M_0) \notin C$ by definition of M_0. Then $\rho(M_0) \notin H_0^-$ by definition of C. On the other hand, if M_0 rejects $\rho(M_0)$ then $\rho(M_0) \in H_0^-$ (by definition of H_0^-), a contradiction.

Q.E.D.

1.2 Trakhtenbrot's Theorem

In the following, we consider FO, more precisely first-order logic with equality. We restrict ourselves to a countable signature

$$\tau_\infty := \{R_j^i : i, j \in \mathbb{N}\} \cup \{f_j^i : i, j \in \mathbb{N}\}$$

where R_j^i stands for a relation symbol of arity i and f_j^i stands for a function symbol of arity i.

We encode formulae over a fixed alphabet

$$\Gamma := \{R, f, x, 0, 1, [,]\} \cup \{=, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, \exists, \forall, (,)\},$$

and uniquely encode the relational and functional symbols

relation symbols:	R_j^i	$\longmapsto$	$R[\text{bin } i][\text{bin } j]$
functional symbols:	f_j^i	$\longmapsto$	$f[\text{bin } i][\text{bin } j]$
variables:	x_j	$\longmapsto$	$x[\text{bin } j]$.

Thus, every formula $\varphi \in \text{FO}$ is a word in Γ^*.

Let $X \subseteq \text{FO}$ be a class of formulae. We analyse the following decision problems:

$$\begin{aligned}
\mathit{Sat}(X) &:= \{\psi \in X : \psi \text{ has a model}\} \\
\mathit{Fin\text{-}sat}(X) &:= \{\psi \in X : \psi \text{ has a finite model}\} \\
\mathit{Val}(X) &:= \{\psi \in X : \psi \text{ is valid}\} \\
\mathit{Non\text{-}sat}(X) &:= X \setminus \mathit{Sat}(X) \\
\mathit{Inf\text{-}axioms}(X) &:= \mathit{Sat}(X) \setminus \mathit{Fin\text{-}sat}(X) \\
&= \{\psi \in X : \psi \text{ is an infinity axiom, i.e. } \psi \text{ has a} \\
&\qquad\qquad \text{model but no finite model}\}.
\end{aligned}$$

Theorem 1.15. Let $X \subseteq \text{FO}$ be decidable. Then

(1) $\mathit{Val}(X)$ is r.e.
(2) $\mathit{Non\text{-}sat}(X)$ is r.e.
(3) $\mathit{Sat}(X)$ is co-r.e.

(4) *Fin-sat*(X) is r.e.
(5) *Inf-axioms*(X) is co-r.e.

Proof. (1) φ is valid $\Leftrightarrow$ $\vdash \varphi$ (Completeness Theorem). Thus we can systematically enumerate all proofs and halt if a proof for φ is listed.
(2) φ valid $\Leftrightarrow$ $\neg\varphi$ is not satisfiable.
(3) Follows from Item (2).
(4) Systematically generate all finite models and halt if a model of φ is found.
(5) FO $\setminus$ *Inf-axioms*(X) = *Non-sat*$(X) \cup$ *Fin-sat*(X) is r.e. Q.E.D.

Definition 1.16. A class $X \subseteq$ FO has the *finite model property* (FMP) if every satisfiable $\varphi \in X$ has a finite model, i.e. if *Sat*(X) = *Fin-sat*(X).

Theorem 1.17. Suppose that $X \subseteq$ FO is decidable and that X has the FMP. Then *Sat*(X) is decidable.

Proof. *Sat*(X) is co-r.e. and since *Sat*(X) = *Fin-sat*(X) and *Fin-sat*(X) is r.e. also *Sat*(X) is r.e. Thus *Sat*(X) is decidable. Q.E.D.

In this case also *Fin-sat*(X), *Non-sat*(X), *Val*(X) are decidable and of course *Inf-axioms*$(X) = \emptyset$ is decidable.

Theorem 1.18 (Trakhtenbrot)**.** There is a finite vocabulary $\tau \subseteq \tau_\infty$ such that *Fin-sat*$(\mathrm{FO}(\tau))$, *Non-sat*$(\mathrm{FO}(\tau))$ and *Inf-axioms*$(\mathrm{FO}(\tau))$ are pairwise recursively inseparable and therefore undecidable.

The proof of Trakhtenbrot's theorem introduces a proof strategy that can be applied in many other undecidability proofs. (Do not focus on the technicalities but on the general idea to construct the reduction formulae.)

Proof. Let M be a deterministic Turing acceptor. We show that there is an effective reduction $\rho(M) \mapsto \psi_M$ such that

(1) M accepts $\varepsilon \implies \psi_M$ has a finite model.
(2) M rejects $\varepsilon \implies \psi_M$ is unsatisfiable.

(3) The computation of M on ε is infinite and non-periodic $\implies$ ψ_M is an infinity axiom.

Then the theorem follows by Lemma 1.13.

Let M be a Turing acceptor with states $Q = \{q_0, \ldots, q_r\}$, initial state q_0, alphabet $\Sigma = \{a_0, \ldots, a_s\}$ (where $a_0 = \square$), final states $F = F^+ \cup F^-$ and transition function δ.

ψ_M is defined over the vocabulary $\tau = \{0, f, q, p, w\}$ where 0 is a constant, f, q, p are unary functions and w is a binary function. Define the term $\boldsymbol{k}$ as $f^k 0$.

By constructing a formula we intend to have a model $\mathfrak{A}_M = (A, 0, f, q, p, w)$ describing a run of M on the input ε where

- universe $A = \{0, 1, 2, \ldots, n\}$ or $A = \mathbb{N}$;
- $f(t) = t + 1$ if $t + 1 \in A$ and $f(t) = t$, if t is the last element of A;
- $q(t) = i$ iff M is at time t in state q_i;
- $p(t)$ is the head position of M at time t;
- $w(s, t) = i$ iff symbol a_i is at time t on tape-cell s.

Note that we cannot enforce this model, but if ψ_M is satisfiable this one will be among its models.

$$\psi_M := \text{START} \wedge \text{COMPUTE} \wedge \text{END}$$

$$\text{START} := (q0 = 0 \wedge p0 = 0 \wedge \forall x\, w(x, 0) = 0).$$

[Enforces input configuration on ε at time 0]

$$\text{COMPUTE} := \text{NOCHANGE} \wedge \text{CHANGE}$$

$$\text{NOCHANGE} := \forall x \forall y (py \neq x \to w(x, fy) = w(x, y))$$

[content of currently not visited tape cells does not change]

$$\text{CHANGE} := \bigwedge_{\delta:(q_i, a_j) \mapsto (q_k, a_\ell, m)} \forall y (\alpha_{i,j} \to \beta_{k,\ell,m})$$

where

$$\alpha_{ij} := (qy = \boldsymbol{i} \wedge w(py, y) = \boldsymbol{j})$$

[M is at time y in state q_i and reads the symbol a_j]

$$\beta_{k,\ell,m} := (qfy = \boldsymbol{k} \wedge w(py, fy) = \boldsymbol{\ell} \wedge \text{MOVE}_m)$$

and

$$\mathrm{MOVE}_m := \begin{cases} pfy = py & \text{if } m = 0 \\ pfy = fpy & \text{if } m = 1 \\ \exists z(fz = py \wedge pfy = z) & \text{if } m = -1. \end{cases}$$

$$\mathrm{END} := \bigwedge_{\substack{\delta(q_i,a_j) \text{ undef.} \\ q_i \notin F^+}} \forall y \neg \alpha_{ij}$$

[The only way the computation ends is in an accepting state]

Remark 1.19.

- $\rho(M) \mapsto \psi_M$ is an effective construction.
- If M accepts ε, the intended model is finite and is indeed a model $\mathfrak{A}_M \models \psi_M$, thus $\psi_M \in \mathit{Fin\text{-}sat}(FO(\tau))$.
- If the computation of M on ε is infinite, the intended model is infinite and $\mathfrak{A}_M \models \psi_M$.

It remains to show that if M rejects ε, then ψ_M is unsatisfiable, and if the computation of M on ε is infinite and aperiodic, then ψ_M is an infinity axiom.

Suppose $\mathfrak{B} = (B, 0, f, q, p, w) \models \psi_M$.

Definition 1.20. $\mathfrak{B}$ *enforces* at time t the configuration (q_i, j, w) with $w = a_{i_0} \dots a_{i_m} \in \Sigma^*$ if

(1) $\mathfrak{B} \models q\boldsymbol{t} = \boldsymbol{i}$,
(2) $\mathfrak{B} \models p\boldsymbol{t} = \boldsymbol{j}$,
(3) for all $k \leq m$, $\mathfrak{B} \models w(\boldsymbol{k}, \boldsymbol{t}) = \boldsymbol{i_k}$ and for all $k > m$, $\mathfrak{B} \models w(\boldsymbol{k}, \boldsymbol{t}) = 0$.

Since $\mathfrak{B} \models \psi_M$, the following holds:

- $\mathfrak{B}$ enforces $C_0 = (q_0, 0, \varepsilon)$ at time 0 (since $\mathfrak{B} \models$ START.)
- If $\mathfrak{B}$ enforces at time t a non-final configuration C_t, then $\mathfrak{B}$ enforces the configuration $C_{t+1} = \mathrm{Next}(\mathrm{C_t})$ at time $t + 1$.
- Especially, the computation of M cannot reach a rejecting configuration. It follows that if M rejects ε, then ψ_M is unsatisfiable.

Consider an infinite and aperiodic computation of M, and assume $\mathfrak{B} \models \psi_M$ is finite. Since $\mathfrak{B}$ is finite, it enforces a periodic computation in contradiction to the assumption that the computation of M is aperiodic.

$$C_0 \vdash \ldots \vdash \underbrace{C_r \quad \vdash \ldots \vdash \quad C_{t-1}}$$

We have shown:

- If M accepts ε, then ψ_M has a finite model.
- If M rejects ε, then ψ_M is unsatisfiable.
- If the computation of M is infinite and aperiodic, then ψ_M is an infinity axiom. Q.E.D.

We now know that the sets of all finitely satisfiable, all unsatisfiable and all only infinitely satisfiable formulae are undecidable for FO(τ) where τ consists of only three unary functions and one binary function. This raises a number of questions.

(1) For which other vocabularies σ do we have similar undecidability results for FO(σ)?
(2) For which σ is satisfiability of FO(σ) decidable?
(3) Is there a complete classification? In this case, we want to find minimal vocabularies σ such that the above problems are undecidable, i.e. vocabularies such that any further restriction yields a class of formulae for which satisfiability is decidable.

We first define what it means that a fragment of FO is as hard for satisfiability as the whole FO.

Definition 1.21. $X \subseteq$ FO is a *reduction class* if there exists a computable function f : FO $\to X$ such that $\psi \in Sat(\text{FO}) \Leftrightarrow f(\psi) \in Sat(X)$.

Let $X, Y \subseteq$ FO. A *conservative reduction of X to Y* is a computable function $f : X \to Y$ with

- $\psi \in Sat(X) \Leftrightarrow f(\psi) \in Sat(Y)$, and
- $\psi \in Fin\text{-}sat(X) \Leftrightarrow f(\psi) \in Fin\text{-}sat(Y)$.

X is a *conservative reduction class* if there exists a conservative reduction of FO to X.

Corollary 1.22. Let X be a conservative reduction class. Then *Fin-sat*(X), *Inf-axioms*(X) and *Non-sat*(X) are pairwise recursively inseparable, and thus *Fin-sat*(X), *Sat*(X), *Val*(X), *Non-sat*(X), *Inf-axioms*(X) are undecidable.

Proof. A conservative reduction from FO to X yields a uniform reduction from *Fin-sat*(FO), *Inf-axioms*(FO) and *Non-sat*(FO) to *Fin-sat*(X), *Inf-axioms*(X) and *Non-sat*(X), respectively. Q.E.D.

We now observe that we can indeed give a complete classification of signatures σ such that FO(σ) is decidable.

Theorem 1.23. If $\sigma \subseteq \{P_0, P_1, \ldots\} \cup \{f\}$ consists of at most one unary function f and an arbitrary number of monadic relations $P_0, P_1, \ldots$, then *Sat*(FO(σ)) is decidable. In all other cases, *Sat*(FO(σ)), *Inf-axioms*(FO(σ)) and *Non-sat*(FO(σ)) are pairwise recursively inseparable, and FO(σ) is a conservative reduction class.

A full proof of this classification theorem is rather difficult. In particular, the decidability of the monadic theory of one unary function, which implies the decidability part, is a difficult theorem due to Rabin. On the other side, one has to show that Trakhtenbrot's theorem applies to the vocabularies

$\tau_1 = \{E\}$ where E is a binary relation,
$\tau_2 = \{f, g\}$ where f, g are unary functions,
$\tau_3 = \{F\}$ where F is a binary function,

and hence to all extensions of τ_1, τ_2, τ_3.

Of course, we may also look at other syntactic restrictions besides restricting the vocabulary. One possibility is to restrict the number of variables. This is only interesting for relational formulae. If we have functions, satisfiability is undecidable even for formulae with only one variable as we shall see.

Define FO^k as first-order logic with relational symbols only and a fixed amount of k variables, say $x_1, \ldots, x_k$.

Theorem 1.24.

- FO^2 has the finite model property and is decidable (see Chapter 2).
- FO^3 is a conservative reduction class.

Another possibility is to restrict the structure of quantifier prefixes.

Definition 1.25 (Prefix-Vocabulary Classes)**.** A string in $\{\forall, \exists\}^*$ is called *prefix*, and an *arity sequence* is a sequence assigning all positive integers values in $\mathbb{N} \cup \{\omega\}$.

For any set of prefixes Π and any arity sequences p and f, $[\Pi, p, f]$ and $[\Pi, p, f]_=$ denote the collection of all formulae $\varphi \in$ FO in prenex normal form without equality and with equality, respectively, such that

- the prefix of φ belongs to Π,
- the number of n-ary predicate symbols in φ is at most $p(n)$ and
- the number of n-ary function symbols in φ is at most $f(n)$.
- Except for the logical constants *true* and *false*, φ has no nullary predicate symbols, no nullary function symbols and no free variables.

The prefix set containing all prefixes and the arity sequence that assigns ω to each n will be denoted *all*.

We write arity sequences as tuples, e.g., $(2, 1, \omega), (0)$ to express that two predicate symbols of arity 1, one of arity 2, unboundedly many of arity 3 and no other predicate or function symbols are allowed.

Theorem 1.26 (Gurevich)**.** Let X be a prefix class, p, q two arity sequences and $X = [\Pi, p, q]_=$.

- X is a conservative reduction class if it contains any of
  (1) $[\forall, (0), (2)]_=$
  (2) $[\forall, (0), (0, 1)]_=$
  (3) $[\forall^2\exists, (\omega, 1), (0)]_=$
  (4) $[\exists^*\forall^2\exists, (0, 1), (0)]_=$
  (5) $[\forall^2\exists^*, (0, 1), (0)]_=$.
- If X is contained in one of the following classes, then $Sat(X)$ and *Inf-axioms*(X) are decidable

(6) $[\exists^*\forall^*, all, (0)]_=$
(7) $[\exists^*, all, all]_=$
(8) $[all, (\omega), (1)]_=$
(9) $[\exists^*\forall\exists^*, all, (1)]_=$.

This gives a complete classification.

1.3 Domino problems

Domino problems are a simple and yet general tool for proving undecidability without talking about Turing machines.

The informal idea is the following: a domino (type) is an oriented square with unit length and coloured edges. We consider the following decision problem.

Given: a finite set of domino types (infinite supply of each).

Question: does there exist a tiling of $\mathbb{N} \times \mathbb{N}$ such that adjacent edges have the same colour?

The undecidability of the stated problem is established by encoding computations of Turing machines in an appropriate way. A row of the tiling represents a configuration of a Turing machine.

Definition 1.27. A *domino system* is a structure $\mathcal{D} = (D, H, V)$ with

- a finite set D,
- horizontal and vertical compatibility relations $H, V \subseteq D \times D$.

The meaning of H and V is that

- $(d, d') \in H$ if the right colour of d is equal to the left colour of d',
- $(d, d') \in V$ if the top colour of d is equal to the bottom colour of d' (see Figure 1.1).

A tiling of $\mathbb{N} \times \mathbb{N}$ by $\mathcal{D}$ is a function $\sigma : \mathbb{N} \times \mathbb{N} \to D$ such that for all $x, y \in \mathbb{N}$

- $(\sigma(x, y), \sigma(x + 1, y)) \in H$ and
- $(\sigma(x, y), \sigma(x, y + 1)) \in V$.

A periodic tiling of $\mathbb{N} \times \mathbb{N}$ by $\mathcal{D}$ is a tiling σ for which two integers $h, v \in \mathbb{N}$ exist such that for all $x, y \in \mathbb{N}$ it holds $\sigma(x, y) = \sigma(x + h, y) = \sigma(x, y + v)$. The decision problem DOMINO is described as

$$\text{DOMINO} := \{\mathcal{D} : \text{there exists a tiling of } \mathbb{N} \times \mathbb{N} \text{ by } \mathcal{D}\}$$

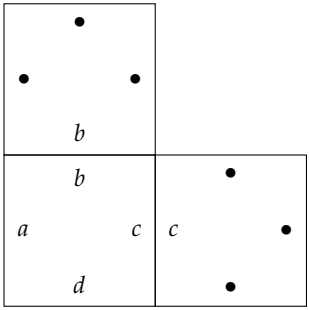


Figure 1.1. Domino adjacency condition

An important variant is the origin constrained tiling.

Definition 1.28. An *origin constrained domino system* is a system $(\mathcal{D}, D_0)$ with $D_0 \subseteq D$. A tiling with origin constraint D_0 is a tiling σ such that $\sigma(0, 0) \in D_0$. The corresponding decision problem is

$$\text{CORNER-DOMINO} := \{(\mathcal{D}, D_0) : \text{ there exists a tiling of } \mathbb{N} \times \mathbb{N} \text{ with origin constraint } D_0\}.$$

Theorem 1.29 (Wang, Büchi)**.** CORNER-DOMINO is undecidable.

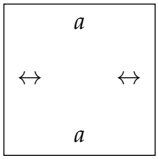
Proof. We reduce $H_\varepsilon^\omega = \{\rho(M) : \text{the computation of } M \text{ on } \varepsilon \text{ is infinite}\}$, which is co-r.e., to CORNER-DOMINO.

Consider a 1-tape TM $M = (Q, \Sigma, q_0, \delta, F)$, and construct $(\mathcal{D}, D_0)$ such that the computation of M on ε is infinite if and only if there exists a tiling of $\mathbb{N} \times \mathbb{N}$ by $\mathcal{D}$ with origin constraint D_0.

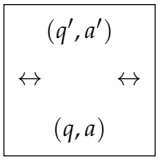
Assume w.l.o.g. that M never moves off-tape to the left, i.e. in configurations $(q, 0, w)$ it is never the case that $\delta(q, w_0) = (q', a, -1)$.

D consists of the following domino types.

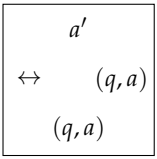
For each $a \in \Sigma$

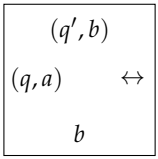


For each $(q, a) \in Q \times \Sigma$ with $\delta(q, a) = (q', a', 0)$

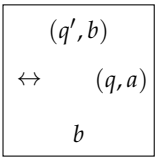


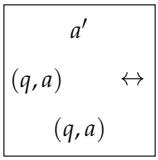
For each $(q, a) \in Q \times \Sigma$ with $\delta(q, a) = (q', a', 1)$, for each $b \in \Sigma$



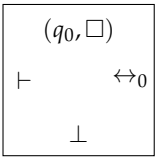


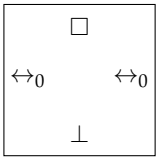
For each $(q, a) \in Q \times \Sigma$ with $\delta(q, a) = (q', a', -1)$ for each $b \in \Sigma$



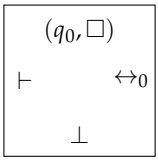


Additionally there exist dominoes



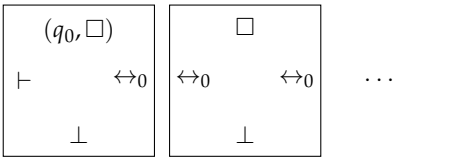


The origin constraint D_0 consists of



Note that $(\mathcal{D}, D_0)$ can be constructed effectively from M.

There is precisely one way of tiling the first row:



$\sigma(0,0)$ $\sigma(0,i)$ for all $i > 0$

Assume the first j rows have been tiled correctly. Then the top edge of row j reads

$$w_0 \dots w_{i-1}(q, w_i)w_{i+1} \dots$$

for $C_j = (q, i, w_0, w_1, \dots)$, the jth configuration of M on ε.

This tiling can be extended to a tiling of row $j+1$ if and only if there exists $C_{j+1} = \mathrm{Next}(\mathrm{C_j})$.

Conclusion: The computation of M on ε is infinite if and only if there exists a tiling of $\mathbb{N} \times \mathbb{N}$ by $(\mathcal{D}, D_0)$. Q.E.D.

Stronger forms of this result are the following

Theorem 1.30 (Berger, Robinson)**.** DOMINO (without origin constraint) is co-r.e. and undecidable.

Theorem 1.31. The problem of tiling $\mathbb{Z} \times \mathbb{Z}$ is reducible to the problem of tiling $\mathbb{N} \times \mathbb{N}$. (Proof via König's Lemma).

Theorem 1.32. The set of domino systems admitting a periodic tiling of $\mathbb{N} \times \mathbb{N}$, those that admit no tiling of $\mathbb{N} \times \mathbb{N}$ and those that admit a tiling but not a periodic one are pairwise recursively inseparable.

Definition 1.33. A computable function f is a *reduction from domino systems* to X if, for all domino systems $\mathcal{D}$, $f(\mathcal{D}) = \varphi_{\mathcal{D}}$ is in X and the following holds:

- $\mathcal{D}$ admits a periodic tiling of $\mathbb{N} \times \mathbb{N} \Rightarrow \psi_{\mathcal{D}}$ has a finite model
- $\mathcal{D}$ admits no tiling of $\mathbb{N} \times \mathbb{N} \Rightarrow \psi_{\mathcal{D}}$ is unsatisfiable
- $\mathcal{D}$ admits a tiling of $\mathbb{N} \times \mathbb{N}$ but no periodic one $\Rightarrow \psi_{\mathcal{D}}$ is an infinity axiom.

Remark 1.34. Let $X \in$ FO. If there exists a reduction from domino systems to X then X is a conservative reduction class.

Proof. Since *Fin-sat*(FO) and *Non-sat*(FO) are recursively enumerable and *Inf-axioms*(FO) is co-recursively enumerable, we can associate with every first-order formula ψ a Turing machine M such that

- $\psi \in$ *Fin-sat*(FO) $\Rightarrow \rho(M) \in H_\varepsilon^+$,
- $\psi \in$ *Non-sat*(FO) $\Rightarrow \rho(M) \in H_\varepsilon^-$,
- $\psi \in$ *Inf-axioms*(FO) $\Rightarrow \rho(M) \in H_\varepsilon^\infty$.

The proof of 1.32 reduces the halting problems $H_\varepsilon^+, H_\varepsilon^-, H_\varepsilon^\infty$, to the domino problems. There exists a recursive function that associates with every TM M a domino system $\mathcal{D}$ satisfying

- If $M \in H_\varepsilon^+$ then $\mathcal{D}$ admits a periodic tiling of $\mathbb{N} \times \mathbb{N}$.
- If $M \in H_\varepsilon^-$ then $\mathcal{D}$ admits no tiling of $\mathbb{N} \times \mathbb{N}$.
- If $M \in H_\varepsilon^\infty$ then $\mathcal{D}$ admits a tiling of $\mathbb{N} \times \mathbb{N}$ but no periodic one.

Finally, according to to the assumption, there is a reduction $\mathcal{D} \mapsto \varphi_\mathcal{D}$ from domino systems to X Thus, the domino method yields a conservative reduction from FO to X.

Q.E.D.

1.4 Applications of the domino method

We now apply the domino method to obtain several reduction classes.

Theorem 1.35. $[\forall\exists\forall, (0, \omega), (0)]$ is a conservative reduction class.

Proof. Due to Remark 1.34 it suffices to give a reduction from domino systems to X, i.e. find a mapping $\mathcal{D} \mapsto \psi_\mathcal{D}$ over a vocabulary consisting of binary relation symbols $(P_d)_{d \in D}$ such that

(1) $\mathcal{D}$ admits a periodic tiling of $\mathbb{N} \times \mathbb{N} \Rightarrow \psi_\mathcal{D}$ has a finite model
(2) $\mathcal{D}$ admits no tiling of $\mathbb{N} \times \mathbb{N} \Rightarrow \psi_\mathcal{D}$ is unsatisfiable
(3) $\mathcal{D}$ admits a tiling of $\mathbb{N} \times \mathbb{N}$ but no periodic one $\Rightarrow \psi_\mathcal{D}$ is an infinity axiom

The intended model is $\mathbb{N}$ with intended interpretation of $P_d = \{(i,j) \in \mathbb{N} \times \mathbb{N} : \tau(i,j) = d\}$ for all $d \in D$. We define $\psi_{\mathcal{D}}$ by

$$\psi_{\mathcal{D}} := \forall x \exists y \forall z \Big(\bigwedge_{d \neq d'} P_d xz \to \neg P_{d'} xz \\ \wedge \bigvee_{(d,d') \in H} (P_d xz \wedge P_{d'} yz) \wedge \bigvee_{(d,d') \in V} (P_d zx \wedge P_{d'} zy) \Big).$$

Obviously $\psi_{\mathcal{D}}$ is of the desired format, i.e. $\psi_{\mathcal{D}} \in [\forall\exists\forall, (0,\omega), (0)]$.

(1) If $\mathcal{D}$ admits a periodic tiling of $\mathbb{N} \times \mathbb{N}$, then $\psi_{\mathcal{D}}$ has a finite model. Let $\tau : \mathbb{N} \times \mathbb{N} \to D$ be a periodic tiling such that for some $h, v \in \mathbb{N}$ $\tau(x,y) = \tau(x+h,y) = \tau(x,y+v)$ for all x, y. Let $t := lcm(h,v)$ be the least common multiple of h and v. Then τ induces a tiling

$$\tau : \mathbb{Z}/t\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z} \to D$$

with $\tau'(x,y) = \tau(x(\mod t), y(\mod t))$.
Thus, $\mathfrak{A} = (\mathbb{Z}/t\mathbb{Z}, (P_d)_{d \in D})$ with $P_d = \{(i,j) : \tau'(i,j) = d\}$ is a finite model (for x in $\psi_{\mathcal{D}}$ choose $y := x + 1 \pmod t$ in $\psi_{\mathcal{D}}$.)

(2) If $\psi_{\mathcal{D}}$ has a model, then $\mathcal{D}$ admits a tiling.

(3) We want to show: if $\psi_{\mathcal{D}}$ has a finite model, then $\mathcal{D}$ admits a periodic tiling. (In the case that $\psi_{\mathcal{D}}$ is unsatisfiable, we show with the same arguments as in (1) that if $\mathcal{D}$ admits a tiling of $\mathbb{N} \times \mathbb{N}$, then $\psi_{\mathcal{D}}$ has a model $\mathfrak{A} = (\mathbb{N}, (P_d)_{d \in D})$.)

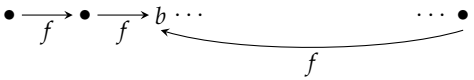
Let now $\psi_{\mathcal{D}}$ have a finite model. To show that if $\psi_{\mathcal{D}}$ has a (finite) model, then $\mathcal{D}$ admits a (periodic) tiling we consider the Skolem normal form $\varphi_{\mathcal{D}}$ of $\psi_{\mathcal{D}}$:

$$\varphi_{\mathcal{D}} := \forall x \forall z (\bigwedge_{d \neq d'} P_d xz \to \neg P_{d'} xz \\ \wedge \bigvee_{(d,d') \in H} (P_d xz \wedge P_{d'} fxz) \wedge \bigvee_{(d,d') \in V} (P_d zx \wedge P_{d'} zfx).$$

- Suppose $\mathfrak{B} = (B, f, (P_d)_{d \in D}) \models \varphi_{\mathcal{D}}$. Define a tiling $\tau : \mathbb{N} \times \mathbb{N} \to D$ as follows: choose $b \in B$, and set $\tau(i,j) := d$ for the unique

$d \in D$ such that $\mathfrak{B} \models P_d(f^i b, f^j b)$ for all $i, j \in \mathbb{N}$. Since $\mathfrak{B} \models \varphi_{\mathcal{D}}$, τ is a correct tiling.

- Suppose that $\mathfrak{B} \models \varphi_{\mathcal{D}}$ is finite:



Choose $b \in B$ such that, for some $t \geq 1$, $f^t b = b$. Then the defined tiling τ is periodic.

Q.E.D.

Corollary 1.36. FO^3 is a conservative reduction class.

Later we show that FO^2 has the FMP.

Consider sets of formulae $X \subseteq$ FO over functional vocabularies. FO(τ) is a conservative reduction class if τ contains

- two unary functions or
- one binary function.

This is even true for sentences of the form $\forall x \varphi$ where φ is quantifier-free.

Theorem 1.37. $[\forall, (0), (2)]_=$ and $[\forall, (0), (0,1)]_=$ are conservative reduction classes.

Proof. We apply the domino method for formulae $\forall x \varphi$ where φ is quantifier-free with any number of unary functions, and then apply a reduction/interpretation to reduce this to two unary/one binary function/s.

Define a mapping $\mathcal{D} = (D, H, V) \mapsto \psi_{\mathcal{D}}$ where $\psi_{\mathcal{D}}$ is a formula over the vocabulary $\{f, g, (h_d)_{d \in D}\}$ where all function symbols are unary. The intended model is $\mathbb{N} \times \mathbb{N}$ with successor functions f and g. The subformula $\forall x(fgx = gfx)$ ensures that the models of $\psi_{\mathcal{D}}$ contain a two-dimensional grid. The fact that a position x is tiled by $d \in D$ is

expressed by requiring that $h_d x = x$, i.e. that x is a fixed point of h_d. Now define

$$\begin{aligned}\psi_{\mathcal{D}} := \forall x\big(fgx = gfx \wedge \bigwedge_{d \neq d'} (h_d x = x \to h_{d'} x \neq x) \\ \wedge \bigvee_{(d,d') \in H} (h_d x = x \wedge h_{d'} fx = fx) \\ \wedge \bigvee_{(d,d') \in V} (h_d x = x \wedge h_{d'} gx = gx)\big)\,.\end{aligned}$$

We claim that there exists a tiling $\sigma : \mathbb{N} \times \mathbb{N} \to \mathcal{D}$ if and only if $\psi_{\mathcal{D}}$ is satisfiable.

" $\Rightarrow$ " Assume σ is a correct tiling. Construct the (intended) model $\mathfrak{A} = (\mathbb{N} \times \mathbb{N}, f, g, (h_d)_{d \in \mathcal{D}})$ with

- $f(i,j) = (i+1, j)$,
- $g(i,j) = (i, j+1)$,
- $h_d(i,j) \begin{cases} = (i,j) & \text{if } \sigma(i,j) = d \\ \neq (i,j) & \text{otherwise.} \end{cases}$

Clearly $\mathfrak{A} \models \psi_{\mathcal{D}}$.

" $\Leftarrow$ " Consider $\mathfrak{B} = (B, f, g, (h_d)_{d \in \mathcal{D}}) \models \psi_{\mathcal{D}}$.

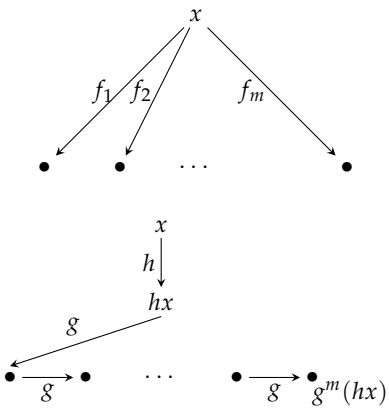
Choose an arbitrary $b \in B$ and define

$$\sigma : \mathbb{N} \times \mathbb{N} \to \mathcal{D} : \sigma(i,j) := d \text{ iff } \mathfrak{B} \models h_d f^i g^j b = f^i g^j b.$$

Note that every position is in exactly one of the h_d. Then σ is a correct tiling. If $\mathfrak{B}$ is finite, then σ is periodic, and thus the reduction is conservative.

We now show that we can sharpen the results, i.e. show that two unary function symbols are sufficient

Consider $\forall x \varphi \in [\forall, (0), (\omega)]_=$ with monadic function symbols $f_1, \ldots, f_m$. Transform φ into $\tilde{\varphi} := \varphi[x/hx, f_i/hg^i]$ where h, g are fresh unary function symbols. This procedure transforms formulae over the vocabulary $\{f_1, \ldots, f_m\}$ into formulae over the vocabulary $\{h, g\}$. The idea is to replace an application of f_i by i applications of g. The second function h takes care of unwanted equalities.



Claim: $\forall x\varphi$ is (finitely) satisfiable $\Leftrightarrow \forall x\tilde{\varphi}$ is (finitely) satisfiable.

" $\Leftarrow$ " Let $\mathfrak{B} = (B, h, g) \models \forall x\tilde{\varphi}$. Construct $\mathfrak{A} = (A, f_1, \ldots, f_m)$ with

- $A = \{hb : b \in B\}$
- $f_i(a) = (hg^i)(a)$

Then $\mathfrak{A} \models \forall x\varphi$.

" $\Rightarrow$ " Let $\mathfrak{A} = (A, f_1, \ldots, f_m) \models \forall x\varphi$. Construct $\mathfrak{B} = (B, g, h)$ with

- $B = A \times (\mathbb{Z}/(m+1)\mathbb{Z})$,
- $g(a, i) = (a, i+1)$,
- $h(a, 0) = (a, 0)$,
- $h(a, i) = (f_i a, 0)$.

This transformation preserves the meaning of terms: Let $t(x) = f_{i_1} \ldots f_{i_k} x$ be a term in φ. Then $\tilde{t}(x) = hg^{i_1} \ldots hg^{i_k} hx$, and it holds that $\tilde{t}^{\mathfrak{B}}[a, 0] = (t^{\mathfrak{A}}[a], 0)$. Now the claim follows via induction over the structure of φ.

We now show that we need at most one binary function. The idea is to find an interpretation of $g, h : A \to A$ in a structure $\mathfrak{A} = (A, F)$ with $F : A \times A \to A$ via

- $g(a) = F(a, F(a, a))$,
- $h(a) = F(F(a, a), a)$

where $F(a,a) \neq a$.

Formally, consider a formula $\forall x \varphi$ with unary function symbols f, g. Introduce a new binary function symbol F and translate

$$\varphi \mapsto \varphi_g \wedge \varphi_h$$

where

$$\varphi_g := \varphi[x/g^*x, g/g^*, h/h^*],$$

$$\varphi_h := \varphi[x/h^*x, g/g^*, h/h^*]$$

with

$$g^*t = F(t, Ftt),$$

$$h^*t = F(Ftt, t).$$

Claim: $\forall x \varphi$ (finitely) satisfiable $\Leftrightarrow$ $\forall x(\varphi_g \wedge \varphi_h)$ (finitely) satisfiable.

" $\Rightarrow$ " Let $\mathfrak{A} = (A, g, h) \models \forall x \varphi$ be a model. Set $\mathfrak{B} = (B, F)$ with

- $B := A \times \mathbb{Z}/3\mathbb{Z}$
- $F((a,i),(a,i)) := (a, i+1)$
- $F((a,i),(a,i+1)) := (ga, 0)$
- $F((a,i+1),(a,i)) := (ha, 0)$.

Now, for all $(a,i) \in B$

$$g^*(a,i) = F\big((a,i), F(a,i)(a,i)\big) = F\big((a,i),(a,i+1)\big) = (ga,0)$$

and

$$h^*(a,i) = (ha,0).$$

Thus $\mathfrak{A}$ is isomorphic to a copy of $\mathfrak{A}$ defined in $\mathfrak{B}$.

$$\mathfrak{A} \cong \mathfrak{A}^* := (\{(a,0) : a \in A\}, g^*, h^*).$$

Therefore, for all (a,i)

$$\mathfrak{B} \models \varphi_g(a,i) \Leftrightarrow \mathfrak{A}^* \models \varphi(ga,0)$$

$$\Leftrightarrow \mathfrak{A} \models \varphi(ga) \quad \text{and}$$
$$\mathfrak{B} \models \varphi_h(a,i) \Leftrightarrow \mathfrak{A}^* \models \varphi(ha,0)$$
$$\Leftrightarrow \mathfrak{A} \models \varphi(ha).$$

Thus, $\mathfrak{A} \models \forall x\varphi$ implies $\mathfrak{B} \models \forall x(\varphi_g \wedge \varphi_k)$.

" $\Leftarrow$ " For $\mathfrak{B} = (B, F) \models \forall x(\varphi_g \wedge \varphi_h)$ let $\mathfrak{A} = (A, g, h)$ with

- $A := g^*(B) \cup h^*(B)$
- $g := g^*$
- $h := h^*$

Then $\mathfrak{A} \models \forall x\varphi$. Q.E.D.

2 Finite Model Property

We study the finite model property for fragments of FO as a mean to show that these fragments are decidable, and also to better understand their expressive power and algorithmic complexity.

Recall that a class $X \subseteq \mathrm{FO}$ has the *finite model property* if $\mathit{Sat}(X) = \mathit{Fin\text{-}sat}(X)$. Since for any decidable class X, $\mathit{Fin\text{-}sat}(X)$ is r.e. and $\mathit{Sat}(X)$ is co-r.e., it follows that $\mathit{Sat}(X)$ is decidable if X has the FMP. In many cases, the proof that a class has the finite model property provides a bound on the model's cardinality, and thus a complexity bound for the satisfiability problem. To prove completeness for complexity classes we make use of a bounded variant of the domino problem.

2.1 Ehrenfeucht-Fraïssé Games

2.1.1 Atomic Types

Definition 2.1. The *atomic k-type* of $a_1, \ldots, a_k$ in $\mathfrak{A}$ is defined as

$$\mathrm{atp}_{\mathfrak{A}}(a_1, \ldots, a_k) := \{\gamma(x_1 \ldots, x_k) : \gamma \text{ atomic formula or negated atomic formula such that } \mathfrak{A} \models \gamma(a_1, \ldots, a_k)\}.$$

We assume that all structures contain unary or binary relations only. Hence, to describe a structure it suffices to define its universe and to specify the atomic 1-types and 2-types for all of its elements.

Example 2.2. Let $\mathfrak{A}$ be the structure $(A, E_1, \ldots, E_m)$ where the E_i are binary relations. Then for $a \in A$:

$$\mathrm{atp}_{\mathfrak{A}}(a) = \{E_i xx : \mathfrak{A} \models E_i aa\} \cup \{\neg E_i xx : \mathfrak{A} \models \neg E_i aa\}.$$

Definition 2.3. Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures over the same signature and

$\bar{a} \subseteq A$ and $\bar{b} \subseteq B$. We say that $\mathfrak{A}, \bar{a}$ is locally isomorphic to $\mathfrak{B}, \bar{b}$ and write $\mathfrak{A}, \bar{a} \equiv_0 \mathfrak{B}, \bar{b}$ if $\bar{a}$ has the same atomic type in $\mathfrak{A}$ as $\bar{b}$ in $\mathfrak{B}$.

2.1.2 The Game $EF_m(\mathfrak{A}, \mathfrak{B})$

The Ehrenfeucht-Fraïssé game $EF_m(\mathfrak{A}, \mathfrak{B})$ is played by two players according to the following rules.

The *arena* consists of the structures $\mathfrak{A}$ and $\mathfrak{B}$. We assume that $A \cap B = \emptyset$. The players are called *Spoiler* and *Duplicator*, and a play of $EF_m(\mathfrak{A}, \mathfrak{B})$ consists of m moves.

In the *i-th move*, Spoiler chooses either an element $a_i \in A$ or an element $b_i \in B$. Duplicator answers by choosing an element in the other structure.

After m moves, elements $a_1, \ldots, a_m$ from $\mathfrak{A}$ and $b_1, \ldots, b_m$ from $\mathfrak{B}$ are chosen. Duplicator wins the play if $\mathfrak{A}, (a_1, a_2, \ldots, a_m) \equiv_0 \mathfrak{B}, (b_1, b_2, \ldots, b_m)$. Otherwise Spoiler wins.

After i moves in $EF_m(\mathfrak{A}, \mathfrak{B})$ are made, a position $(a_1, \ldots, a_i, b_1, \ldots, b_i)$ is reached. We denote the remaining subgame in which $m - i$ moves are left by $EF_{m-i}(\mathfrak{A}, a_1, \ldots, a_i, \mathfrak{B}, b_1, \ldots, b_i)$.

A *winning strategy* of Spoiler for such a subgame is a function which, for every reachable position, determines a move such that Spoiler wins each play which is consistent with this strategy, no matter how Duplicator plays. Winning strategies for Duplicator are defined analogously.

We say that *Spoiler (respectively, Duplicator) wins the game* $EF_m(\mathfrak{A}, \mathfrak{B})$ if this player has a winning strategy for $EF_m(\mathfrak{A}, \mathfrak{B})$. By induction on the number of moves it is easy to show that for every (sub)game exactly one of the two players has a winning strategy.

Example 2.4.

- Let $\mathfrak{A} = (\mathbb{Z}, <)$, $\mathfrak{B} = (\mathbb{R}, <)$. Then Duplicator wins $EF_2(\mathfrak{A}, \mathfrak{B})$, but Spoiler wins $EF_3(\mathfrak{A}, \mathfrak{B})$.
- For a relational signature $\tau = \{E, P\}$ (where P has arity one and E has arity two), consider the structures $\mathfrak{A}$ and $\mathfrak{B}$ in Figure 2.1. Spoiler wins the game $EF_3(\mathfrak{A}, \mathfrak{B})$, but Duplicator wins $EF_2(\mathfrak{A}, \mathfrak{B})$.

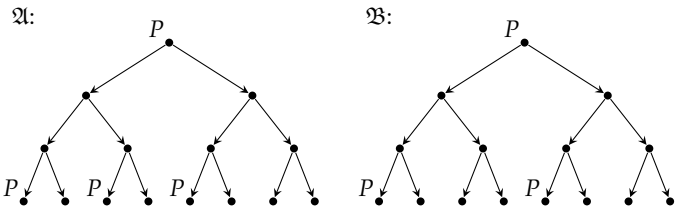


Figure 2.1. Two structures$\mathfrak{A}$ and $\mathfrak{B}$ with $\mathfrak{A} \equiv_2 \mathfrak{B}$ and $\mathfrak{A} \not\equiv_3 \mathfrak{B}$

2.1.3 The Game $EF(\mathfrak{A}, \mathfrak{B})$

An important variant is the Ehrenfeucht-Fraïssé game $EF(\mathfrak{A}, \mathfrak{B})$ in which plays of arbitrary length are possible. In each play, Spoiler first chooses an $m \in \mathbb{N}$, and then the players play the game $EF_m(\mathfrak{A}, \mathfrak{B})$.

Spoiler wins the game $EF(\mathfrak{A}, \mathfrak{B})$ if and only if there exists an $m \in \mathbb{N}$ such that he wins the game $EF_m(\mathfrak{A}, \mathfrak{B})$. In other words: Duplicator wins $EF(\mathfrak{A}, \mathfrak{B})$ if and only if she has a winning strategy for each of the games $EF_m(\mathfrak{A}, \mathfrak{B})$.

Recall that two structures $\mathfrak{A}$ and $\mathfrak{B}$ are said to be *elementarily m-equivalent*, written $\mathfrak{A} \equiv_m \mathfrak{B}$, if no first-order formula of quantifier rank at most m can separate both structures. If $\mathfrak{A} \equiv_m \mathfrak{B}$ for all $m \in \mathbb{N}$ we write $\mathfrak{A} \equiv \mathfrak{B}$ and say that $\mathfrak{A}$ and $\mathfrak{B}$ are *elementarily equivalent*. The following theorem shows that elementary equivalence and Ehrenfeucht-Fraïssé games are in some sense equivalent concepts.

Theorem 2.5 (Ehrenfeucht, Fraïssé)**.** Let τ be finite and relational, and let $\mathfrak{A}, \mathfrak{B}$ be τ-structures.

(1) The following statements are equivalent:
   (i) $\mathfrak{A} \equiv \mathfrak{B}$.
   (ii) Duplicator wins the Ehrenfeucht-Fraïssé game $EF(\mathfrak{A}, \mathfrak{B})$.

(2) For all $m \in \mathbb{N}$ the following statements are equivalent:
   (i) $\mathfrak{A} \equiv_m \mathfrak{B}$.
   (ii) Duplicator wins $EF_m(\mathfrak{A}, \mathfrak{B})$.

In fact, even the following, somewhat stronger proposition holds (for a proof see the lecture notes of mathematical logic).

Theorem 2.6. Let $\mathfrak{A}, \mathfrak{B}$ be τ-structures, $\bar{a} = a_1, \ldots, a_r \in A$, $\bar{b} = b_1, \ldots, b_r \in B$. If there exists a formula $\psi(\bar{x})$ with $\mathrm{qr}(\psi) = m$ such that $\mathfrak{A} \models \psi(\bar{a})$ and $\mathfrak{B} \models \neg\psi(\bar{b})$ holds, then Spoiler has a winning strategy for the game $G_m(\mathfrak{A}, \bar{a}, \mathfrak{B}, \bar{b})$.

We use the above to prove finite model property of the following fragment of FO.

Theorem 2.7. If τ contains only unary predicates then FO$[\tau]$ has FMP.

Proof. Let $\mathfrak{A} = (A, P_1, \ldots, P_n)$ and let $\mathrm{qr}(\varphi) = m$. For each sequence of bits $\alpha = \alpha_1 \ldots \alpha_n$ we define $P_\alpha = Q_1 \cap Q_2 \cap \ldots \cap Q_n$, where $Q_i = P_i$ if $\alpha_i = 1$ and Q_i is the complement of P_i else.

Note that $\{\alpha \mid x \in P_\alpha\}$ determines all atomic types of x. We construct $\mathfrak{B}$ by taking $\min(|P_\alpha|, m)$ elements into each $P_\alpha^{\mathfrak{B}}$. Observe that $\mathfrak{B}$ is defined in this way (take $P_i^{\mathfrak{B}} = \bigcup_{\alpha|\alpha_i=1} P_\alpha^{\mathfrak{B}}$). We show that $\mathfrak{A} \equiv_m \mathfrak{B}$ using the Ehrenfeucht-Fraïssé Theorem.

The following is a winning strategy for Duplicator in $EF(\mathfrak{A}, \mathfrak{B})$: Answer each Spoiler's choice of an element with an element of the same atomic type in the other structure. Due to the construction it is possible to do that for m moves. It also follows from the construction that $\equiv_0$ is never violated and Duplicator wins the game. Q.E.D.

You can see from the proof that the constructed finite model $\mathfrak{B}$ is a sub-model of $\mathfrak{A}$. It is not always the case, sometimes it is not possible to find a finite sub-model, even for fragments with FMP.

2.2 FMP of Modal Logic

We proceed with proving that propositional modal logic (ML), which is an important fragment of FO^2, has the finite model property. In fact we establish an even stronger result showing that every satisfiable ML-formula has a finite model that is a tree. Hence, we prove that ML has the *finite tree model property*.

2.2.1 Modal Logic

Let us first briefly review the syntax and semantics of propositional modal logic (ML).

Definition 2.8. For a given set of actions A and atomic properties $\{P_i : i \in I\}$, the syntax of ML is inductively defined as:

- All propositional logic formulae with propositional variables P_i are in ML.
- If $\psi, \varphi \in$ ML, then also $\neg\psi$, $(\psi \wedge \varphi)$ and $(\psi \vee \varphi) \in$ ML.
- If $\psi \in$ ML and $a \in A$, then $\langle a \rangle \psi$ and $[a]\psi \in$ ML.

Remark 2.9. If there is only one action $a \in A$, we write $\Diamond\psi$ and $\Box\psi$ instead of $\langle a \rangle \psi$ and $[a]\psi$, respectively.

Definition 2.10. A *transition system* or *Kripke structure* with actions from a set A and atomic properties $\{P_i : i \in I\}$ is a structure

$$\mathcal{K} = (V, (E_a)_{a \in A}, (P_i)_{i \in I})$$

with a universe V of states, binary relations $E_a \subseteq V \times V$ describing transitions between the states, and unary relations $P_i \subseteq V$ describing the atomic properties of states.

A transition system can be seen as a labelled graph where the nodes are the states of $\mathcal{K}$, the unary relations are labels of the states, and the binary transition relations are the labelled edges.

Definition 2.11. Let $\mathcal{K} = (V, (E_a)_{a \in A}, (P_i)_{i \in I})$ be a transition system, $\psi \in$ ML a formula and v a state of $\mathcal{K}$. The *model relationship* $\mathcal{K}, v \models \psi$, i.e. ψ holds at state v of $\mathcal{K}$, is inductively defined:

- $\mathcal{K}, v \models P_i$ if and only if $v \in P_i$.
- $\mathcal{K}, v \models \neg\psi$ if and only if $\mathcal{K}, v \not\models \psi$.
- $\mathcal{K}, v \models \psi \vee \varphi$ if and only if $\mathcal{K}, v \models \psi$ or $\mathcal{K}, v \models \varphi$.
- $\mathcal{K}, v \models \psi \wedge \varphi$ if and only if $\mathcal{K}, v \models \psi$ and $\mathcal{K}, v \models \varphi$.
- $\mathcal{K}, v \models \langle a \rangle \psi$ if and only if there exists w such that $(v, w) \in E_a$ and $\mathcal{K}, w \models \psi$.
- $\mathcal{K}, v \models [a]\,\psi$ if and only if $\mathcal{K}, w \models \psi$ holds for all w with $(v, w) \in E_a$.

Definition 2.12. For a transition system $\mathcal{K}$ and a formula ψ we define the *extension*

$$[\![\psi]\!]^{\mathcal{K}} := \{v : \mathcal{K}, v \models \psi\}$$

as the set of states of $\mathcal{K}$ where ψ holds.

2.2.2 Bisimulation

One of the most important notions in the analysis of modal logics is *bisimulation*. In fact bisimulation is closely related to logical equivalence of Kripke structures with respect to formulae from ML.

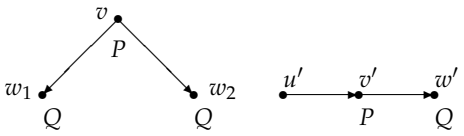
Definition 2.13. Let $\mathcal{K} = (V, (E_a)_{a\in A}, (P_i)_{i\in I})$ and $\mathcal{K}' = (V', (E'_a)_{a\in A}, (P'_i)_{i\in I})$ be transition systems. A *bisimulation* between $\mathcal{K}$ and $\mathcal{K}'$ is a relation $Z \subseteq V \times V'$ such that for all $(v, v') \in Z$

(Pred) $v \in P_i$ if and only if $v' \in P'_i$ for all $i \in I$,

(Forth) for all $a \in A$, $w \in V$ with $v \xrightarrow{a} w$ there exists a $w' \in V'$ with $v' \xrightarrow{a} w'$ and it is $(w, w') \in Z$,

(Back) for all $a \in A$, $w' \in V'$ with $v' \xrightarrow{a} w'$ there exists a $w \in V$ with $v \xrightarrow{a} w$ and it is $(w, w') \in Z$.

Example 2.14.



$Z = \{(v, v'), (w_1, w'), (w_2, w')\}$ is a bisimulation.

Definition 2.15. Let $\mathcal{K}, \mathcal{K}'$ be Kripke structures and let $u \in V$, $u' \in V'$. $(\mathcal{K}, u)$ and $(\mathcal{K}', u')$ are *bisimilar* (for short, $\mathcal{K}, u \sim \mathcal{K}', u'$), if there exists a bisimulation Z between $\mathcal{K}$ and $\mathcal{K}'$ such that $(u, u') \in Z$.

2.2.3 Bisimulation Invariance of Formulae of Modal Logic

The fundamental importance of bisimulation origins in the fact that formulae of modal logic are not able to distinguish between bisimilar

states. A more refined analysis considers the modal depth of formulae, i.e. the maximal depth of nesting of modal operators in a formula.

Definition 2.16. The *modal depth* of a formula $\psi \in \mathrm{ML}$ is defined inductively by

(1) $\mathrm{md}(\psi) = 0$ for propositional formulae ψ,
(2) $\mathrm{md}(\neg\psi) = \mathrm{md}(\psi)$,
(3) $\mathrm{md}(\psi \circ \varphi) = \max(\mathrm{md}(\psi), \mathrm{md}(\varphi))$ for $\circ \in \{\wedge, \vee, \to\}$,
(4) $\mathrm{md}(\langle a \rangle \psi) = \mathrm{md}([a]\psi) = \mathrm{md}(\psi) + 1$.

Definition 2.17. Let $\mathcal{K}$ and $\mathcal{K}'$ be two Kripke structures and let $v \in \mathcal{K}$, $v' \in \mathcal{K}'$.

(1) $\mathcal{K}, v \equiv_{\mathrm{ML}} \mathcal{K}', v'$ if for all $\psi \in \mathrm{ML}$ we have $\mathcal{K}, v \models \psi$ if and only if $\mathcal{K}', v' \models \psi$.
(2) $\mathcal{K}, v \equiv^n_{\mathrm{ML}} \mathcal{K}', v'$ if for all $\psi \in \mathrm{ML}$ with $\mathrm{md}(\psi) \leq n$ we have $\mathcal{K}, v \models \psi$ if and only if $\mathcal{K}', v' \models \psi$.

One can refine the definition of the bisimilarity relation between transition systems as well. We say that $(\mathcal{K}, u)$ and $(\mathcal{K}', u')$ are *n-bisimilar* (for short, $\mathcal{K}, u \sim_n \mathcal{K}', u'$), if there exists a relation Z between $\mathcal{K}$ and $\mathcal{K}'$ such that $(u, u') \in Z$ and Z has the property 'Pred' and the 'Forth' and 'Back' property for all pairs of nodes $(v, v') \in Z$ with distance at most n from (u, u'). For a formal (game theoretical) definition, see the lectures notes of mathematical logic.

Theorem 2.18. For Kripke structures $\mathcal{K}$, $\mathcal{K}'$ and $u \in \mathcal{K}$, $u' \in \mathcal{K}'$ the following holds:

(1) $\mathcal{K}, u \sim \mathcal{K}', u' \Rightarrow \mathcal{K}, u \equiv_{\mathrm{ML}} \mathcal{K}', u'$.
(2) $\mathcal{K}, u \sim_n \mathcal{K}', u' \Rightarrow \mathcal{K}, u \equiv^n_{\mathrm{ML}} \mathcal{K}', u'$.

Statement (1) is called the *bisimulation invariance of modal logic*:

$$\text{If } \mathcal{K}, v \models \psi \text{ and } \mathcal{K}, v \sim \mathcal{K}', v', \text{ then } \mathcal{K}', v' \models \psi.$$

The reverse only holds for finitely branching systems. A transition system is *finitely branching* if for all states v and all actions a the set $vE_a := \{w : (v, w) \in E_a\}$ of a-successors of v is finite. (for proofs see the lecture notes of mathematical logic).

Theorem 2.19. Let $\mathcal{K}, \mathcal{K}'$ be finitely branching transitions systems. Then

$$\mathcal{K}, u' \sim \mathcal{K}', u' \text{ if and only if } \mathcal{K}, u \equiv_{\mathrm{ML}} \mathcal{K}', u'.$$

2.2.4 Tree Model Property

Definition 2.20. A transition system $\mathcal{K} = (V, (E_a)_{a \in A}, (P_i)_{i \in I})$ with a marked node w is a *tree* if

(1) $E_a \cap E_b = \emptyset$ for all actions $a \neq b$,
(2) (V, E) is a (directed) tree with root w in the graph theoretical sense, where $E = \bigcup_{a \in A} E_a$.

Definition 2.21. Let Φ be a set of formulae (of some logic, e.g. of modal logic or first-order logic) over a signature which contains at most binary relations and no functions.

(1) Φ has the *finite model property* (FMP) if every satisfiable formula $\varphi \in \Phi$ has a finite model.
(2) Φ has the *tree model property* (TMP) if every satisfiable formula in Φ has a tree as a model.
(3) Φ has *finite tree model property* if every satisfiable formula in Φ has a finite tree as a model.

We shall prove that formulae of modal logic have the finite tree model property. For that consider *unfoldings* of transition systems. The unfolding of $\mathcal{K}$ from state v consists of all paths in $\mathcal{K}$ that start with v. Hereby every path is considered as a distinguished object, i.e. even if two paths intersect, the unfolding $\mathcal{T}$ contains several copies of the intersection points and each state from $\mathcal{K}$ that is reachable from v via a path is added to the unfolding, no matter whether is has already been reached. Self-loops in $\mathcal{K}$ correspond thus to infinite paths in the unfolding. Formally, unfoldings are defined as follows.

Definition 2.22. Let $\mathcal{K} = (V^{\mathcal{K}}, (E_a^{\mathcal{K}})_{a \in A}, (P_i^{\mathcal{K}})_{i \in I})$ be a Kripke structure and let $v \in V^{\mathcal{K}}$. The *unfolding* of $\mathcal{K}$ from v is the Kripke structure $\mathcal{T}_{\mathcal{K},v} = (V^{\mathcal{T}}, (E_a^{\mathcal{T}})_{a \in A}, (P_i^{\mathcal{T}})_{i \in I})$ with

$$V^{\mathcal{T}} = \{\bar{v} = v_0 a_0 v_1 a_1 v_2 \dots v_{m-1} a_{m-1} v_m : m \in \mathbb{N},$$

$$v_0 = v,\ v_i \in V^{\mathcal{K}},\ a_i \in A,\ (v_i, v_{i+1}) \in E^{\mathcal{K}}_{a_i} \text{ for all } i < m\}$$
$$E^{\mathcal{T}}_a = \{(\bar{v}, \bar{w}) \in V^{\mathcal{T}} \times V^{\mathcal{T}} : \bar{w} = \bar{v}aw \text{ for some } w \in V^{\mathcal{K}}, a \in A\}$$
$$P^{\mathcal{T}}_i = \{\bar{v} = v_0 a_0 \dots v_m \in V^{\mathcal{T}} : v_m \in P^{\mathcal{K}}_i\}.$$

We write $\mathrm{End}(\bar{v})$ for the last state on the path $\bar{v}$, so we have $\bar{v} \in P^{\mathcal{T}}_i$ if and only if $\mathrm{End}(\bar{v}) \in P^{\mathcal{K}}_i$.

Lemma 2.23. For all Kripke structures $\mathcal{K}$ and all states v in $\mathcal{K}$ we have $\mathcal{K}, v \sim \mathcal{T}_{K,v}, v$.

Proof. $Z := \{(w, \bar{w}) \in V^{\mathcal{K}} \times V^{\mathcal{T}} : \mathrm{End}(\bar{w}) = w\}$ is a bisimulation from $\mathcal{K}$ to $\mathcal{T}_{\mathcal{K},v}$ with $(v, v) \in Z$. Q.E.D.

Theorem 2.24. ML has the tree model property.

Proof. Let ψ be an arbitrary satisfiable formula from ML. Then there is a model $\mathcal{K}, v \models \psi$. Let $\mathcal{T} := \mathcal{T}_{\mathcal{K},v}$ be he unfolding of $\mathcal{K}, v$. As $\mathcal{K}, v \sim \mathcal{T}, v$, due to the bisimulation invariance of modal logic we have $\mathcal{T}, v \models \psi$. Thus ψ has a tree model. Q.E.D.

The same argument shows that *every* class of bisimulation invariant formulae has the tree model property.

2.2.5 Finite Model Property

For ML, we can prove a stronger result. For this, we use the notion of the closure $C(\psi)$ of a formula ψ.

Definition 2.25. For every formula $\psi \in$ ML we inductively define for all $n \in \mathbb{N}$ the sets of formulae $C_n(\psi)$ as follows:

(1) $\psi \in C_0(\psi)$.
(2) If $\neg\varphi \in C_n(\psi)$ then also $\varphi \in C_n(\psi)$.
(3) If $(\varphi \wedge \vartheta) \in C_n(\psi)$ or $(\varphi \vee \vartheta) \in C_n(\psi)$ then also $\varphi \in C_n(\psi)$ and $\vartheta \in C_n(\psi)$.
(4) If $\langle a \rangle \varphi \in C_n(\psi)$ or $[a]\varphi \in C_n(\psi)$ then $\varphi \in C_{n+1}(\psi)$.

Finally let $C(\psi) := \bigcup_{n \in \mathbb{N}} C_j(\psi)$.

The closure $C(\psi)$ contains those formulae from ML that are substantial for the evaluation of ψ; $C_j(\psi)$ are hereby formulae that appear in ψ within j nested modal operators. Notice that $|C(\psi)| \leq 2|\psi|$ (negated formulas are added) and that $C_n(\psi) = \emptyset$ for all $n > \mathrm{md}(\psi)$.

Theorem 2.26. For every satisfiable formula $\psi \in \mathrm{ML}$ there is a finite tree structure $\mathcal{T}, v$ of depth $\leq \mathrm{md}(\psi)$ and branching factor $\leq |C(\psi)|$ such that $\mathcal{T}, v \models \psi$. Thus ML has finite tree model property.

Proof. Without loss of generality we can assume that ψ is in negation normal form. As ψ is satisfiable, there exists a tree model $\mathcal{T}, u \models \psi$. The depth of a node of $\mathcal{T}$ is its distance from the root. We define now a labelling function S which assigns a subset of $C_m(\psi)$ to every node v of $\mathcal{T}$ of depth m, namely

$$S(v) := \{\varphi \in C_m(\psi) : \mathcal{T}, v \models \varphi\}.$$

We transform $\mathcal{T}$ in a finite tree structure by successively deleting unnecessary subtrees. Let $\mathcal{T}' \subseteq \mathcal{T}$ be some subtree of $\mathcal{T}$ and let v be a node of $\mathcal{T}'$. Notice that $\mathcal{T}, v \models S(v)$. The following lemma provides a sufficient condition for $\mathcal{T}', v \models S(v)$.

Lemma 2.27. Let the subtree $\mathcal{T}' \subseteq \mathcal{T}$ be constructed in a way that the following conditions are fulfilled.

(1) For every successor w of v in $\mathcal{T}'$ we have $\mathcal{T}', w \models S(w)$.
(2) For every formula of the form $\langle a \rangle \varphi \in S(v)$ there exists an a-successor $w_{\langle a \rangle \varphi}$ of v in the tree $\mathcal{T}'$ such that $\mathcal{T}', w_{\langle a \rangle \varphi} \models \varphi$.

Then it is $\mathcal{T}', v \models S(v)$.

Proof. Each formula in $S(v)$ is a combination of formulae of the form P_i, $\neg P_i$, $\langle a \rangle \varphi$ and $[a]\varphi$ that are built with $\wedge$ and $\vee$. So it suffices to show for every formula ϑ of this form that $\mathcal{T}, v \models \vartheta$ implies $\mathcal{T}', v \models \vartheta$. For $\vartheta = P_i$ and $\vartheta = \neg P_i$ this is clear as the atomic properties of the node v are the same in $\mathcal{T}$ and $\mathcal{T}'$. For formulae $[a]\varphi$ this follows from condition (1) and for formulae $\langle a \rangle \varphi$ from condition (2). Q.E.D.

Now we can construct a finite subtree $\mathcal{T}'$ as follows. First, let v be the root of $\mathcal{T}$. For every formula of the form $\langle a \rangle \varphi \in S(v)$ we choose an a-successor $w_{\langle a \rangle \varphi} \in vE_a$ such that $\mathcal{T}, w_{\langle a \rangle \varphi} \models \varphi$ holds and delete all not chosen successor nodes of v and the trees that have those nodes as roots from $\mathcal{T}$. We continue this process for all remaining nodes of depth $1, 2, \ldots$. As the labelling $S(v)$ of nodes of depth $m = \mathrm{md}(\psi)$ only consists of formulae P_i and $\neg P_i$, the resulting tree has depth at most m. Every node v has at most $|S(v)| \leq C(\psi)$ successors such that the branching factor of $\mathcal{T}'$ is bounded by $|C(\psi)|$.

It follows by inductively proceeding from leaves to the root of $\mathcal{T}'$ that $\mathcal{T}', v \models S(v)$, in particular, $\mathcal{T}', v \models \psi$. Q.E.D.

2.3 Finite Model Property of FO^2

We denote relational first-order logic over k variables by FO^k, i.e.

$$\mathrm{FO}^k := \{\varphi \in \mathrm{FO} : \varphi \text{ relational, } \varphi \text{ only contains } k \text{ variables}\}.$$

One result of the previous chapter was that $[\forall\exists\forall, \mathrm{all}, (0)] \subseteq \mathrm{FO}^3$ is a conservative reduction class. We now prove that FO^2 has the finite model property and is thus decidable. Note that FO^k formulae are not necessarily in prenex normal form. A further motivation for the study of FO^2 is that propositional modal logic can be viewed as a fragment of FO^2 (in fact ML can be proven to be precisely the bisimulation invariant fragment of FO^2).

Before we proceed to prove the finite model property for FO^2, as a first step we establish a normal form for formulae in FO^2.

Lemma 2.28 (Scott)**.** For each sentence $\psi \in \mathrm{FO}^2$ one can construct in polynomial time a sentence $\varphi \in \mathrm{FO}^2$ of the form

$$\varphi := \forall x \forall y \alpha \wedge \bigwedge_{i=1}^{n} \forall x \exists y \beta_i$$

such that $\alpha, \beta_1, \ldots, \beta_n$ are quantifier free and such that ψ and φ are satisfiable over the same universe. Moreover, we have $|\varphi| = \mathcal{O}(|\psi| \cdot \log |\psi|)$.

Proof. First of all, we can assume that formulae $\varphi \in \mathrm{FO}^2$ only contain unary and binary relation symbols. This is no restriction since relations of higher arity can be substituted by introducing new binary and unary relation symbols. For example, if R is a relation of arity three, one could add a unary relation R_x and three binary relations $R_{x,x,y}$, $R_{x,y,x}$ and $R_{x,y,y}$ and replace each atom $R(x,x,x)$ (or $R(y,y,y)$) by $R_x(x)$ (or $R_x(y)$) and atoms as $R(x,x,y)$ or $R(x,y,x)$ by $R_{x,x,y}(x,y)$ and $R_{x,y,x}(x,y)$ respectively. By adding appropriate new subformulae one can ensure that the semantics are preserved, i.e. that the newly introduced relations partition a ternary relation in the intended sense. For example we would introduce as a new subformula $\forall x(R_x(x) \leftrightarrow R_{x,x,y}(x,x))$.

With ψ containing at most binary relations, we iterate the following steps until ψ has the desired form. We choose a subformula $Qy\eta$ of ψ ($Q \in \{\forall, \exists\}$, η quantifier free) and add a new unary relation R:

$$\begin{aligned} \psi' &:= \psi[Qy\eta / Rx] \\ \psi &\mapsto \psi' \wedge \forall x(Rx \leftrightarrow Qy\eta). \end{aligned}$$

R captures those x that satisfy $Qy\eta$. The resulting formula φ is not yet of the desired form, but it is equivalent to the following:

(a) if $Q = \exists$, then

$$\varphi \equiv \psi' \wedge \forall x \forall y(\eta \to Rx) \wedge \forall x \exists y(Rx \to \eta)$$

(b) else if $Q = \forall$, then

$$\varphi \equiv \psi' \wedge \forall x \forall y(Rx \to \eta) \wedge \forall x \exists y(\eta \to Rx)$$

Now use that conjunctions of $\forall\forall$-formulae are equivalent to a $\forall\forall$-formula and obtain $\psi \equiv \forall x \forall y \alpha \wedge \bigwedge_{i=1}^{n} \forall x \exists y \beta_i$. Q.E.D.

Theorem 2.29. FO^2 has the finite model property. In fact, every satisfiable formula $\psi \in \mathrm{FO}^2$ has a model with at most $2^{|\psi|}$ elements.

Proof. The proof strategy is as follows: we start with a model $\mathfrak{A}$ of ψ and

proceed by constructing a new model $\mathfrak{B}$ of ψ such that $|\mathfrak{B}| \leq 2^{\mathcal{O}(|\psi|)}$. For the construction the following definitions will be essential.

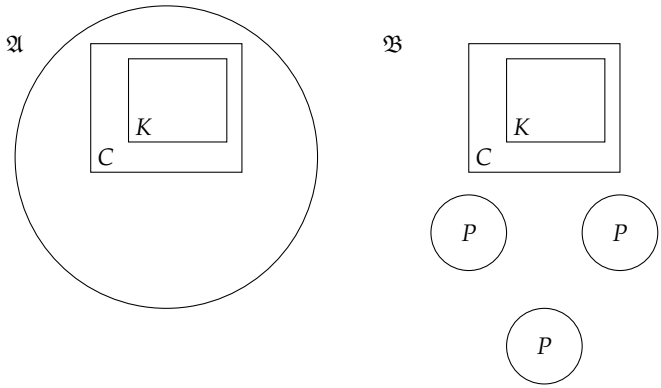
An element $a \in A$ is said to be a *king of* $\mathfrak{A}$ if its atomic 1-type is unique in $\mathfrak{A}$, i.e. if $\mathrm{atp}_{\mathfrak{A}}(b) \neq \mathrm{atp}_{\mathfrak{A}}(a)$ for all $b \neq a$. We let

- $K := \{a \in A : a \text{ is a king of } \mathfrak{A}\}$ be the set of kings of $\mathfrak{A}$, and
- $P := \{\mathrm{atp}_{\mathfrak{A}}(a) : a \in A, a \notin K\}$ be the set of atomic 1-types which are realized at least twice in $\mathfrak{A}$.

Since $\mathfrak{A} \models \forall x \exists y \beta_i$ for $i = 1, \ldots, n$, there exist (Skolem) functions $f_1, \ldots, f_n : A \to A$ such that $\mathfrak{A} \models \beta_i(a, f_i a)$ for all $a \in A$. The *court* of $\mathfrak{A}$ is defined as

$$C := K \cup \{f_i k : k \in K, i = 1, \ldots, n\}.$$

Let $\mathfrak{C}$ be the substructure of $\mathfrak{A}$ induced by C. We construct a model $\mathfrak{B} \models \psi$ with universe $B = C \cup (P \times \{1, \ldots, n\} \times \{0, 1, 2\})$.



To specify $\mathfrak{B}$ we set $\mathfrak{B}|_C = \mathfrak{C}$ and for all other elements we specify the 1- and 2-types (in this way fixing $\mathfrak{B}$ on the remaining part). However,

(1) This must be done consistently:
   - $\mathrm{atp}_{\mathfrak{A}}(b, b')$ and $\mathrm{atp}_{\mathfrak{A}}(b, b'')$ must agree on $\mathrm{atp}_{\mathfrak{A}}(b)$, and
   - $\gamma(x, y) \in \mathrm{atp}_{\mathfrak{B}}(b, b') \Leftrightarrow \gamma(y, x) \in \mathrm{atp}_{\mathfrak{B}}(b', b)$.

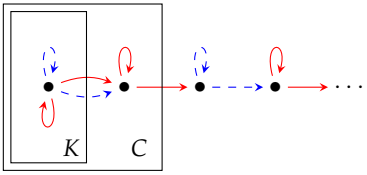
(2) Of course we have to ensure that $\mathfrak{B} \models \psi$.

We illustrate the construction with the following example.

Example 2.30. Consider the formula ψ over the signature $\tau = \{R, B\}$ (red edges and blue edges).

$$\begin{aligned}
\psi \quad &= \quad \exists x(Rxx \wedge Bxx) \\
&\wedge \quad \forall x \forall y((Rxx \wedge Bxx \wedge Ryy \wedge Byy \rightarrow x = y) \\
&\qquad \wedge (Rxx \vee Bxx) \\
&\qquad \wedge (Rxy \wedge Ryx \rightarrow x = y) \\
&\qquad \wedge (Bxy \wedge Byx \rightarrow x = y) \\
&\qquad \wedge (Bxy \wedge x \neq y \rightarrow Ryy)) \\
&\wedge \quad \forall x \exists y(x \neq y \wedge (Rxx \rightarrow Rxy) \\
&\wedge \quad (Bxx \rightarrow Bxy)).
\end{aligned}$$

Let $\mathfrak{A} \models \psi$, then $\mathfrak{A}$ looks like follows:



In this case $P = \{\{Rxx, \neg Bxx\}, \{\neg Rxx, Bxx\}\}$ and the universe of $\mathfrak{B}$ is $B = C \cup (P \times \{1\} \times \{0, 1, 2\})$.

We proceed to construct $\mathfrak{B}$ by specifying the 1-types and 2-types of its elements as follows.

(1) The atomic 1-types of elements (p, i, j) are set to $\mathrm{atp}_{\mathfrak{B}}((p, i, j)) = p$.
(2) The atomic 2-types $\mathrm{atp}_{\mathfrak{B}}(b, b')$ will be set so that $\mathfrak{B} \models \forall x \exists y \beta_i$ for $i = 1, \ldots, m$.
Choose for each $p \in P$ an element $h(p) \in A$ with $\mathrm{atp}_{\mathfrak{A}}(h(p)) = p$.
Find for each $b \in \mathfrak{B}$ and each i a suitable element b' such that $\mathfrak{B} \models \beta_i(b, b')$ (by defining $\mathrm{atp}_{\mathfrak{B}}(b, b')$ appropriately).
  (a) If b is a king, set $b' := f_i(b) \in C \subseteq B$. Then $\mathfrak{B} \models \beta_i(b, b')$.
  (b) If $b \in C \setminus K$ (non-royal member of the court), distinguish:
    - If $f_i(b) \in K$, then set $b' := f_i(b) \in K \subseteq B$.

- Otherwise it holds that $\mathrm{atp}_{\mathfrak{A}}(f_i(b)) = p \in P$.
  In this case, set $b' := (p, i, 0)$. Now set $\mathrm{atp}_{\mathfrak{B}}(b, b') := \mathrm{atp}_{\mathfrak{A}}(b, f_i(b))$. Thus $\mathfrak{B} \models \beta_i(b, b')$ since $\mathfrak{A} \models \beta_i(b, f_i(b))$.

(c) If $b = (p, j, \ell)$ for some $p \in P, j \in \{1, \ldots, n\}, \ell \in \{0, 1, 2\}$, let $a := h(p)$ and consider $f_i(a)$.
If $f_i(a) \in K$, set $b' = f_i(a)$ and $\mathrm{atp}_{\mathfrak{B}}(b, b') := \mathrm{atp}_{\mathfrak{A}}(a, b')$.
If $f_i(a) \notin K$, then $\mathrm{atp}_{\mathfrak{A}}(f_i(a)) = p' \in P$.
Set $b' := (p', i, (\ell + 1) \pmod 3)$.
Then set $\mathrm{atp}_{\mathfrak{B}}(b, b') := \mathrm{atp}_{\mathfrak{A}}(a, f_i(a))$, and thus $\mathfrak{B} \models \beta_i(b, b')$.

To complete the construction of $\mathfrak{B}$, let $b_1, b_2 \in B$ be such that $\mathrm{atp}_{\mathfrak{B}}(b_1, b_2)$ is not yet specified. Choose $a_1, a_2 \in A$ so that

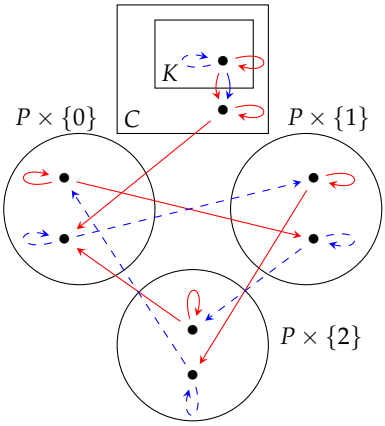
$$\begin{aligned} \mathrm{atp}_{\mathfrak{A}}(a_1) &= \mathrm{atp}_{\mathfrak{B}}(b_1) \text{ and} \\ \mathrm{atp}_{\mathfrak{A}}(a_2) &= \mathrm{atp}_{\mathfrak{B}}(b_2) \end{aligned}$$

and set

$$\mathrm{atp}_{\mathfrak{B}}(b_1, b_2) := \mathrm{atp}_{\mathfrak{A}}(a_1, a_2).$$

Since $\mathfrak{A} \models \alpha(a_1, a_2)$, also $\mathfrak{B} \models \alpha(b_1, b_2)$.

For the previously considered example, $\mathfrak{B}$ looks as follows:



Overall, we obtain $\mathfrak{B} \models \forall x \forall y \alpha \wedge \bigwedge_{i=1}^{n} \forall x \exists y \beta_i = \psi$, and the size of B

is restricted by

$$|B| = \underbrace{|C|}_{\leq |K|(n+1)} + 3n|P| = \mathcal{O}(n \cdot \#\,(\text{atomic 1-types}))\,.$$

For k relation symbols, there are 2^k atomic 1-types, hence $|B| = 2^{\mathcal{O}(|\psi|)}$.

Q.E.D.

This result implies that $Sat(\mathrm{FO}^2)$ is in NEXPTIME (indeed it is NEXPTIME-complete), since we can simply guess a finite structure $\mathfrak{A}$ of exponential size (in the length of ψ) and verify that $\mathfrak{A} \models \psi$.

Corollary 2.31. $Sat(\mathrm{FO}^2) \in \mathrm{NEXPTIME} = (\bigcup_k \mathrm{NTIME}(2^{n^k}))$.

This is a typical complexity level for decidable fragments of FO. In fact, $Sat(\mathrm{FO}^2)$ is even complete for NEXPTIME. For showing this, we reduce a bounded version of the domino problem to $Sat(\mathrm{FO}^2)$.

Definition 2.32. Let $\mathcal{D} = (D, H, V)$ be a domino system and let $Z(t)$ denote $\mathbb{Z}/t\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$. For a word $w = w_0, \ldots, w_{n-1} \in D^n$ we say that $\mathcal{D}$ tiles $Z(t)$ with initial condition w if there is $\tau : Z(t) \to D$ such that

- if $\tau(x,y) = d$ and $\tau(x+1,y) = d'$ then $(d,d') \in H$ for all $(x,y) \in Z(t)$,
- if $\tau(x,y) = d, \tau(x,y+1) = d'$ then $(d,d') \in V$ for all $(x,y) \in Z(t)$ and
- $\tau(i,0) = w_i$ for all $i = 0, \ldots, n-1$.

Let $\mathcal{D}$ be a domino system and $T : \mathbb{N} \to \mathbb{N}$ a mapping. Define

$$\mathrm{DOMINO}(\mathcal{D}, T) := \{w \in D^* : \mathcal{D} \text{ tiles } Z(T(|w|)) \text{ with initial condition } w\}\,.$$

As before we describe a computation of a (in this case nondeterministic) Turing machine by a domino tiling in such a way that the input condition of the domino problem relates to the initial configuration of the Turing machine. The restrictions on the size of the tiled rectangle correspond to the time and space restrictions of the Turing

machine. To prove that a problem A is NEXPTIME-hard, it suffices to show that $\mathrm{DOMINO}(\mathcal{D}, 2^n) \leq_p A$.

Our goal is to show that $\mathrm{DOMINO}(\mathcal{D}, 2^n)$ reduces to $Sat(X)$ for relatively simple classes $X \subseteq \mathrm{FO}$. Set

$$X = \{\varphi \in \mathrm{FO}^2 : \varphi = \forall x \forall y\, \alpha \wedge \forall x \exists y\, \beta, \text{ s.t. } \alpha, \beta \text{ quantifier-free,} \\ \text{without } =, \text{ and with only monadic predicates}\}.$$

We show that $Sat(X)$ is NEXPTIME-complete and hence also $Sat(\mathrm{FO}^2)$ is NEXPTIME-complete.

Lemma 2.33. For each domino system $\mathcal{D} = (D, H, V)$ there exists a polynomial time reduction $w \in D^n \mapsto \psi_w \in X$ such that $\mathcal{D}$ tiles $Z(2^n)$ with initial condition w if and only if ψ_w is satisfiable.

Proof. The intended model of ψ_w is a description of a tiling $\tau : Z(2^n) \to D$ in the universe $Z(2^n)$.

Let $z = (a, b) \in Z(2^n)$ with $a = \sum_{i=0}^{n-1} a_i 2^i$ and $b = \sum_{i=0}^{n-1} b_i 2^i$. Encode the tuple as $(a_o, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}) \in \{0,1\}^{2n}$.

To encode the tiling, we define ψ_w with the monadic predicates X_i, X_i^*, Y_i, Y_i^*, N_i for $0 \leq i < n$ and $P_d (d \in D)$ with the following intended meaning:

$$\begin{aligned} X_i z \quad &\text{iff} \quad a_i = 1. \\ X_i^* z \quad &\text{iff} \quad a_j = 1 \text{ for all } j < i. \\ Y_i z \quad &\text{iff} \quad b_j = 1. \\ Y_i^* z \quad &\text{iff} \quad b_j = 1 \text{ for all } j < i. \\ N_i z \quad &\text{iff} \quad z = (i, 0). \\ P_d z \quad &\text{iff} \quad \tau(z) = d. \end{aligned}$$

ψ_w will have the form $\psi_w = \forall x \forall y \alpha \wedge \forall x \exists y \beta$, where β accounts for the correct interpretation of $X_i, X_i^*, Y_i, Y_i^*, N_i$ and ensures that every element has a successor, and α accounts for the description of a correct tiling.

Now β is the the following formula:

$$\begin{aligned}
\beta = {} & X_0^* x \wedge Y_0^* x \\
\wedge \quad & \bigwedge_{i=1}^{n-1} X_i^* x \leftrightarrow (X_{i-1}^* x \wedge X_{i-1} x) \\
\wedge \quad & \bigwedge_{i=1}^{n-1} Y_i^* x \leftrightarrow (Y_{i-1}^* x \wedge Y_{i-1} x) \\
\wedge \quad & \bigwedge_{i=0}^{n-1} X_i y \leftrightarrow (X_i x \oplus X_i^* x) \\
\wedge \quad & \bigwedge_{i=0}^{n-1} Y_i y \leftrightarrow (Y_i x \oplus (Y_i^* x \wedge X_{n-1} x \wedge X_{n-1}^* x)) \\
\wedge \quad & N_0 x \leftrightarrow (\bigwedge_{i=0}^{n-1} \neg X_i x \wedge \neg Y_i x) \\
\wedge \quad & \bigwedge_{i=0}^{n-1} N_i x \leftrightarrow N_{i+1} y.
\end{aligned}$$

We define the following shorthands for use in α:

$$\begin{aligned}
H(x,y) \quad &:= \quad \bigwedge_{i=0}^{n-1} (Y_i y \leftrightarrow Y_i x) \wedge \bigwedge_{i=0}^{n-1} (X_i y \leftrightarrow (X_i x \oplus X_i^* x)) \\
V(x,y) \quad &:= \quad \bigwedge_{i=0}^{n-1} (X_i y \leftrightarrow X_i x) \wedge \bigwedge_{i=0}^{n-1} (Y_i y \leftrightarrow (Y_i x \oplus Y_i^* x)).
\end{aligned}$$

Now α is defined to be

$$\begin{aligned}
\alpha = {} & \bigwedge_{d \neq d'} \neg (P_d x \wedge P_{d'} x) \\
\wedge \quad & (H(x,y) \rightarrow \bigvee_{(d,d') \in H} (P_d x \wedge P_{d'} y)) \\
\wedge \quad & (V(x,y) \rightarrow \bigvee_{(d,d') \in V} (P_d x \wedge P_{d'} y)) \\
\wedge \quad & (\bigwedge_{i=i}^{n-1} (N_i x \rightarrow P_{w_i} x)).
\end{aligned}$$

Claim 2.34. ψ_w is satisfiable if and only if $\mathcal{D}$ tiles $Z(2^n)$ with initial condition w.

Proof. We show both directions.

($\Leftarrow$) Consider the intended model, ψ_w holds in it.
($\Rightarrow$) Consider $\mathfrak{C} = (C, X_1, \ldots) \models \psi_w$ and define a mapping

$$\begin{aligned} f: \quad C &\to Z(2^n) \\ c &\mapsto (a,b) \equiv (a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}) \end{aligned}$$

$$\begin{aligned} \text{with } a_i = 1 \quad &\text{iff} \quad \mathfrak{C} \models X_i c \quad \text{and} \\ b_i = 1 \quad &\text{iff} \quad \mathfrak{C} \models Y_i c. \end{aligned}$$

As $\mathfrak{C} \models \forall x \exists y \beta$, f is surjective. Choose for each $z \in Z(2^n)$ an element $c \in f^{-1}(z)$ and set $\tau(z) = d$ for the unique d that satisfies $\mathfrak{C} \models P_d c$. Then τ is a correct tiling with initial condition w. Q.E.D.

Since the length of ψ_w is $|\psi_w| = O(n \log n)$, the above claim completes the proof of the lemma. Q.E.D.

3 Descriptive Complexity

In this chapter we study the relationship between logical definability and computational complexity on finite structures. In contrast to the theory of computational complexity we do not measure resources as time and space required to decide a property but the logical resources needed to define it. The ultimate goal is to characterize the complexity classes known from computational complexity theory by means of logic.

We first define what it means for a logic to capture a complexity class. One of the main results is due to Fagin, stating that existential second order logic captures NP, while it is still unknown whether there exists a logic capturing PTIME on all finite structures. A deeper analysis of the proof of Fagin's Theorem shows that SO-HORN logic captures PTIME on all ordered finite structures.

We further introduce least-fixed point logic, LFP, and prove the result of Immerman and Vardi which states that least fixed-point logic also captures PTIME on all ordered finite structures. We compare LFP to inflationary fixed-point logic (IFP), which turns out to be equivalent to LFP. Finally, we present partial fixed-point logic PFP and logics with counting.

3.1 Logics Capturing Complexity Classes

Assume we have a class of finite τ-structures. To measure the complexity of problems we have to represent the structures by strings over a finite alphabet Σ so that they can be used as inputs for Turing machines. Since Turing machines accept *words* and logics do not distinguish between isomorphic structures, for encoding a structure it is necessary to fix an ordering on the universe before.

By $\mathrm{Ord}(\tau)$ we denote the class of all finite structures $(\mathfrak{A}, <)$, where

$\mathfrak{A}$ is a τ-structure and $<$ is a linear order on its universe. For any structure $\mathfrak{A} \in \mathrm{Ord}(\tau)$ with universe of size n, and for a fixed k, we can identify A^k with the set $\{0, 1, \ldots, n^k - 1\}$. This is done by associating each k-tuple $\bar{a}$ with its rank in the lexicographic ordering induced by $<$ on A^k. When we talk about the $\bar{a}$-th element, we understand it in this sense.

Definition 3.1. An *encoding* is a function mapping ordered structures to words. An encoding $\mathrm{code}(\cdot) : \mathrm{Ord}(\tau) \to \Sigma^*$ is good if it identifies isomorphic structures, is polynomially bounded, first-order definable and allows to compute the values of atomic statements efficiently. Formally, the following abstract conditions must be satisfied.

- $\mathrm{code}(\mathfrak{A}, <) = \mathrm{code}(\mathfrak{B}, <)$ iff $(\mathfrak{A}, <) \cong (\mathfrak{B}, <)$.
- There is a fixed polynomial p such that $|\mathrm{code}(\mathfrak{A}, <)| \leq p(|A|)$ for all $(\mathfrak{A}, <) \in \mathrm{Ord}(\tau)$.
- For all $k \in \mathbb{N}$ and all $\sigma \in \Sigma$ there exists a first-order formula $\beta_\sigma(x_1, \ldots, x_k)$ of vocabulary $\tau \cup \{<\}$ so that for all $(\mathfrak{A}, <)$ and all $\bar{a} \in A^k$ it holds that

$$(\mathfrak{A}, <) \models \beta_\sigma(\bar{a}) \Leftrightarrow \text{ the } \bar{a}\text{-th symbol of } \mathrm{code}(\mathfrak{A}, <) \text{ is } \sigma.$$

- Given $\mathrm{code}(\mathfrak{A}, <)$ a relation symbol R of τ and a tuple $\bar{a}$ one can efficiently decide whether $\mathfrak{A} \models R\bar{a}$.

The meaning of "efficiently" in the last condition may depend on the context, here we understand it is as evaluated in linear time and logarithmic space.

Example 3.2. Let $\mathfrak{A} = (A, R_1, \ldots, R_m)$ be a structure with a linear order $<$ on A. Let $|A| = n$ and let s_i be the arity of R_i. Let ℓ be the maximal arity of $R_1, \ldots, R_m$. For each relation we define

$$\chi(R_j) = w_0 \ldots w_{n^{s_j}-1} 0^{n^\ell - n^{s_j}} \in \{0,1\}^{n^\ell},$$

where $w_i = 1$ if the i-th element of A^{s_j} is in R_j. Now

$$\mathrm{code}(\mathfrak{A}, <) := 1^n 0^{n^\ell - n} \chi(R_1) \ldots \chi(R_m).$$

When we say that an algorithm decides a class $\mathcal{K}$ of finite τ-structures we actually mean that it decides

$$\text{code}(\mathcal{K}) = \{\text{code}(\mathfrak{A}, <) : \mathfrak{A} \in \mathcal{K}, < \text{ a linear order on } A\}.$$

Definition 3.3. A *model class* is a class $\mathcal{K}$ of structures of a fixed vocabulary τ that is closed under isomorphism, i.e. if $\mathfrak{A} \in \mathcal{K}$ and $\mathfrak{A} \cong \mathfrak{B}$, then $\mathfrak{B} \in \mathcal{K}$.

A *domain* is an isomorphism closed class $\mathcal{D}$ of structures where the vocabulary is not fixed. For a domain $\mathcal{D}$ and vocabulary τ, we write $\mathcal{D}(\tau)$ for the class of τ-structures in $\mathcal{D}$.

Definition 3.4. Let L be a logic, Comp a complexity class and $\mathcal{D}$ a domain of finite structures. L captures Comp on $\mathcal{D}$ if

(1) For every vocabulary τ and every (fixed) sentence $\psi \in L(\tau)$, the model-checking problem for ψ on $\mathcal{D}(\tau)$ is in Comp.
(2) For every vocabulary τ and any model class $\mathcal{K} \subseteq \mathcal{D}(\tau)$ whose membership problem is in Comp, there exists a sentence $\psi \in L(\tau)$ such that

$$\mathcal{K} = \{\mathfrak{A} \in \mathcal{D}(\tau) : \mathfrak{A} \models \psi\}.$$

3.2 Fagin's Theorem

Existential second-order logic (Σ^1_1) is the fragment of second-order logic consisting of formulae of the form $\exists R_1 \dots \exists R_m \varphi$ where $\varphi \in$ FO and $R_1, \dots, R_m$ are relation symbols. As we will see in this chapter, the logic Σ^1_1 captures the complexity class NP on the domain of all finite structures.

Example 3.5. 3-Colorability of a graph $G = (V, E)$ is in NP and indeed there is a Σ^1_1-formula defining the class of graphs which possess a valid 3-coloring:

$$\begin{aligned} \exists R \exists B \exists Y \quad (\quad & \forall x (Rx \vee Bx \vee Yx) \\ \wedge \quad & \forall x \forall y (Exy \to \neg((Rx \wedge Ry) \vee (Bx \wedge By) \vee (Yx \wedge Yy)) \end{aligned}$$

Theorem 3.6 (Fagin)**.** Existential second-order logic captures NP on the domain of all finite structures.

Proof.
The proof consists of two parts. First of all, let $\psi = \exists R_1 \dots \exists R_m \varphi \in \Sigma^1_1$ be an existential second-order sentence. We show that it can be decided in non-deterministic polynomial time whether a given structure $\mathfrak{A}$ is a model of ψ.

In a first step, we guess relations $R_1, \dots, R_m$ on A. Recall that relations can be identified with binary strings of length n^{s_i}, where s_i is the arity of R_i. Then we check whether $(\mathfrak{A}, R_1, \dots, R_m) \models \varphi$ which can be done in LOGSPACE and hence in PTIME. Thus the computation consists of guessing a polynomial number of bits followed by a deterministic polynomial time computation, showing that the problem is in NP.

For the other direction, let $\mathcal{K}$ be an isomorphism-closed class of τ-structures and let M be a non-deterministic TM deciding code($\mathcal{K}$) in polynomial time. We construct a sentence $\psi \in \Sigma^1_1$ such that for all finite τ-structure $\mathfrak{A}$ it holds that

$$\mathfrak{A} \models \psi \Leftrightarrow M \text{ accepts code}(\mathfrak{A}, <) \text{ for any linear order } < \text{ on } A.$$

Let $M = (Q, \Sigma, q_0, F^+, F^-, \delta)$ with accepting and rejecting states F^+ and F^- and $\delta : (Q \times \Sigma) \to \mathcal{P}(Q \times \Sigma \times \{0, 1, -1\})$ which, given an input code$(\mathfrak{A}, <)$, decides in non-deterministic polynomial time whether $\mathfrak{A}$ belongs to $\mathcal{K}$ or not. We assume that all computations of M reach an accepting or rejecting state after precisely n^k steps ($n := |A|$).

We encode a computation of M on code$(\mathfrak{A}, <)$ by relations $\overline{X}$ and construct a first-order sentence $\varphi_M \in \text{FO}(\tau \cup \{<\} \cup \{\overline{X}\})$ such that for every linear order $<$ there exists $\overline{X}$ with $(\mathfrak{A}, <, \overline{X}) \models \varphi_M$ if and only if code$(\mathfrak{A}, <) \in L(M)$. To this end we show that

- If $\overline{X}$ represents an accepting computation of M on code$(\mathfrak{A}, <)$ then $(\mathfrak{A}, <, \overline{X}) \models \varphi_M$.
- If $(\mathfrak{A}, <, \overline{X}) \models \varphi_M$ then $\overline{X}$ *contains* a representation of an accepting computation of M on code$(\mathfrak{A}, <)$.

Accordingly the desired formula ψ is then obtained via existential second-order quantification

$$\psi := (\exists <)(\exists \overline{X})(\text{“}< \text{ is a linear order ”} \wedge \varphi_M).$$

Details:

- We represent numbers up to n^k as tuples in A^k.
- For each state $q \in Q$ we introduce a predicate

$$X_q := \{\overline{t} \in A^k : \text{ at time } \overline{t} \text{ the TM } M \text{ is in state } q\}.$$

- For each symbol $\sigma \in \Sigma$ we define

$$Y_\sigma := \{(\overline{t}, \overline{a}) \in A^k \times A^k : \text{ at time } \overline{t} \text{ the cell } \overline{a} \text{ contains } \sigma\}.$$

- The head predicate is

$$Z := \{(\overline{t}, \overline{a}) \in A^k \times A^k : \text{ at time } \overline{t} \text{ the head of } M \text{ is at position } \overline{a}\}.$$

Now φ_M is the universal closure of START $\wedge$ COMPUTE $\wedge$ END.

$$\text{START} := X_{q_0}(\overline{0}) \wedge Z(\overline{0}, \overline{0}) \wedge \bigwedge_{\sigma \in \Sigma} (\beta_\sigma(\overline{x}) \to Y_\sigma(\overline{0}, \overline{x})).$$

Recall that β_σ states that the symbol at position $\overline{x}$ in code$(\mathfrak{A}, <)$ is σ. The existence of the formulae β_σ is guaranteed by the fact that code$(\cdot)$ is a good encoding. In what follows, we denote by $\overline{x}+1$ and $\overline{x}-1$ a first-order formula that defines the direct successor and predecessor of the tuple $\overline{x}$ (in the lexicographical ordering on tuples that is induced by the linear order $<$), respectively.

$$\text{COMPUTE} := \text{NOCHANGE} \wedge \text{CHANGE}.$$

$$\text{NOCHANGE} := \bigwedge_{\sigma\in\Sigma} (Y_\sigma(\bar{t},\bar{x}) \wedge Z(\bar{t},\bar{y}) \wedge \bar{y} \neq \bar{x} \wedge \bar{t}' = \bar{t}+1 \to Y_\sigma(\bar{t}',\bar{x})).$$

$$\text{CHANGE} := \bigwedge_{q\in Q,\sigma\in\Sigma} (\text{PRE}[q,\sigma] \to \bigvee_{(q',\sigma',m)\in\delta(q,\sigma)} \text{POST}[q',\sigma',m]),$$

where

$$\text{PRE}[q,\sigma] := X_q(\bar{t}) \wedge Z(\bar{t},\bar{x}) \wedge Y_\sigma(\bar{t},\bar{x}) \wedge \bar{t}' = \bar{t}+1,$$

$$\text{POST}[q',\sigma',m] := X_{q'}(\bar{t}') \wedge Y_{\sigma'}(\bar{t}',\bar{x}) \wedge \text{MOVE}_m[\bar{t}',\bar{x}],$$

and

$$\text{MOVE}_m[\bar{t}',\bar{x}] := \begin{cases} \exists\bar{y}(\bar{x}-1=\bar{y} \wedge Z(\bar{t}',\bar{y})), & m=-1 \\ Z(\bar{t}',\bar{x}), & m=0 \\ \exists\bar{y}(\bar{x}+1=\bar{y} \wedge Z(\bar{t}',\bar{y})), & m=1. \end{cases}$$

Finally, we let

$$\text{END} := \bigwedge_{q\in F^-} \neg X_q(\bar{t}).$$

It remains to show the following two claims.
Claim 1. If $\overline{X}$ represents an accepting computation of M on code$(\mathfrak{A},<)$ then $(\mathfrak{A},<,\overline{X}) \models \varphi_M$. This, however, follows immediately from the construction of φ_M.
Claim 2. If $(\mathfrak{A},<,\overline{X}) \models \varphi_M$, then $\overline{X}$ contains a representation of an accepting computation of M on code$(\mathfrak{A},<)$. We define

$$\text{CONF}[C,j] := X_q(\bar{j}) \wedge Z(\bar{j},\bar{p}) \wedge \bigwedge_{i=0}^{n^k-1} Y_{w_i}(\bar{j},\bar{i})$$

for configurations $C = (w_0 \dots w_{n^k-1}, q, p)$ (tape content $w_0 \dots w_{n^k-1}$, state q, head position p), i.e. the conjunction of the atomic statements that hold for C at time j. Let C_0 be the input configuration of M on code$(\mathfrak{A},<)$. Since $(\mathfrak{A},<,\overline{X}) \models$ START it follows that

$$(\mathfrak{A}, <, \overline{X}) \models \mathrm{CONF}[C_0, 0].$$

Since $(\mathfrak{A}, <, \overline{X}) \models \mathrm{COMPUTE}$ and $(\mathfrak{A}, <, \overline{X}) \models \mathrm{CONF}[C_i, t]$, for some $C_i \vdash C_{i+1}$ it holds that $(\mathfrak{A}, <, \overline{X}) \models \mathrm{CONF}[C_{i+1}, t+1]$.

Finally, no rejecting configuration can be encoded in $\overline{X}$ because $(\mathfrak{A}, <, \overline{X}) \models \mathrm{END}$. Thus an accepting computation

$$C_0 \vdash C_1 \vdash \ldots \vdash C_{n^k-1}$$

of M on code$(\mathfrak{A}, <)$ exists, with $(\mathfrak{A}, <, \overline{X}) \models \mathrm{CONF}[C_i, i]$ for all $i \leq n^k - 1$. This completes the proof of Fagin's Theorem. Q.E.D.

Theorem 3.7 (Cook, Levin)**.** SAT is NP-complete.

Proof. Obviously SAT $\in$ NP. We show that for any Σ^1_1-definable class $\mathcal{K}$ of finite structures the membership problem $\mathfrak{A} \in \mathcal{K}$ can be reduced to SAT. By Fagin's Theorem, there exists a first-order sentence ψ such that

$$\mathcal{K} = \{\mathfrak{A} \in \mathrm{Fin}(\tau) : \mathfrak{A} \models \exists R_1 \ldots \exists R_m \psi\}.$$

Given $\mathfrak{A}$, construct a propositional formula $\psi_{\mathfrak{A}}$ as follows.

- replace $\exists x_i \varphi$ by $\bigvee_{a \in A} \varphi[x_i/a]$,
- replace $\forall x_i \varphi$ by $\bigwedge_{a \in A} \varphi[x_i/a]$,
- replace all closed τ-atoms $P\bar{a}$ in ψ with their truth values,
- replace all atoms $R\bar{a}$ with propositional variables $P_{R\bar{a}}$.

This is a polynomial transformation and it holds that

$$\mathfrak{A} \in \mathcal{K} \Leftrightarrow \mathfrak{A} \models \exists R_1 \ldots \exists R_m \psi \Leftrightarrow \psi_{\mathfrak{A}} \in \mathrm{SAT}.$$

Q.E.D.

3.3 Second Order Horn Logic on Ordered Structures

The problem of whether there exists a logic capturing PTIME on all finite structures is still open. The theorem of Immerman and Vardi states that least fixed-point logic captures PTIME on the class of all ordered finite structures. We first present the result of Grädel that

on ordered finite structures SO-HORN captures PTIME. This result follows from a careful analysis of the proof of Fagin's Theorem (indeed, the construction we used in its proof is not the standard one, but an optimized version so that it can be adapted for showing that SO-HORN captures PTIME on ordered structures).

Definition 3.8. *Second-order Horn logic*, denoted by SO-HORN, is the set of second-order sentences of the form

$$Q_1 R_1 \dots Q_m R_m \forall y_1 \dots \forall y_s \bigwedge_{i=1}^{t} C_i,$$

where $Q_i \in \{\exists, \forall\}$ and the C_i are Horn clauses, i.e. implications

$$\beta_1 \wedge \dots \wedge \beta_m \to H,$$

where each β_j is either a positive atom $R_k \bar{z}$ or an FO-formula that does not contain $R_1, \dots, R_m$. H is either a positive atom $R_j \bar{z}$ or the Boolean constant 0.

Σ^1_1-HORN denotes the existential fragment of SO-HORN, i.e. the set of SO-HORN sentences where all second-order quantifiers are existential.

Theorem 3.9. Every sentence $\psi \in$ SO-HORN is equivalent to a sentence $\psi' \in \Sigma^1_1$-HORN.

Proof. It suffices to prove the theorem for formulae of the form

$$\psi = \forall P \exists R_1 \dots \exists R_m \forall \bar{z} \varphi,$$

where φ is a conjunction of Horn clauses and $m \geq 0$ (for $m = 0$, the formula has the form $\forall P \forall \bar{z} \varphi$). Indeed we can then eliminate universal quantifiers beginning with the inner most one by considering only the part starting with that universal quantifier.

Lemma 3.10. A formula $\exists \overline{R} \forall \bar{z} \varphi(P, \overline{R}) \in \Sigma^1_1$-HORN holds for all relations P on a structure $\mathfrak{A}$ if and only if it holds for those P that are false at at most one point.

Proof. Let k be the arity of P. For every k-tuple $\overline{a}$, let $P^{\overline{a}} = A^k - \{\overline{a}\}$, i.e. the relation that is false at $\overline{a}$ and true at all other points. By assumption, there exist $\overline{R}^{\overline{a}}$ such that

$$(\mathfrak{A}, P^{\overline{a}}, \overline{R}^{\overline{a}}) \models \forall \overline{z} \varphi.$$

Now consider any $P \neq A^k$ and let $R_i := \bigcap_{\overline{a} \notin P} R_i^{\overline{a}}$. We show that $(\mathfrak{A}, P, \overline{R}) \models \forall \overline{z} \varphi$ where $\overline{R}$ is the tuple consisting of all R_i.

Suppose that this is false, then there exists a relation $P \neq A^k$, a clause C of φ and an assignment $\rho : \{z_1, \ldots, z_s\} \to A$ such that $(\mathfrak{A}, P, \overline{R}) \models \neg C[\rho]$. We proceed to show that in this case there exists a tuple $\overline{a}$ such that $(\mathfrak{A}, P^{\overline{a}}, \overline{R}^{\overline{a}}) \models \neg C[\rho]$ and thus

$$(\mathfrak{A}, P^{\overline{a}}, \overline{R}^{\overline{a}}) \models \neg \forall \overline{z} \varphi$$

which contradicts the assumption.

- If the head of $C[\rho]$ is $P\overline{a}$, then take $\overline{a} = \overline{u} \notin P$.
- If the head of $C[\rho]$ is $R_i \overline{u}$, then choose $\overline{a} \notin P$ such that $\overline{u} \notin R_i^{\overline{a}}$, which exists because $\overline{u} \notin R_i$.
- If the head is 0, take an arbitrary $\overline{a} \notin P$.

The head of $C[\rho]$ is clearly false in $(\mathfrak{A}, P^{\overline{a}}, \overline{R}^{\overline{a}})$. $P\overline{a}$ does not occur in the body of $C[\rho]$, because $\overline{a} \notin P$ and all atoms in the body of $C[\rho]$ are true in $(\mathfrak{A}, P, \overline{R})$. All other atoms of the form P_i that might occur in the body of the clause remain true for $P^{\overline{a}}$. Moreover, every atom $R_i \overline{v}$ in the body remains true if R_i is replaced by $R_i^{\overline{a}}$ because $R_i \subseteq R_i^{\overline{a}}$. This implies $(\mathfrak{A}, P^{\overline{a}}, \overline{R}^{\overline{a}}) \models \neg C[\rho]$. Q.E.D.

Using the above lemma, the original formula $\psi = \forall P \exists R_1 \ldots \exists R_m \forall \overline{z} \varphi$ is equivalent to

$$\exists \overline{R}\ \forall \overline{z} \varphi[P\overline{u}/\overline{u} = \overline{u}] \wedge \forall \overline{y}\ \exists \overline{R}\ \forall \overline{z} \varphi[P\overline{u}/\overline{u} \neq \overline{y}].$$

This formula can be converted again to Σ_1^1-HORN; in the second part we push the external first-order quantifiers inside while increasing the

arity of quantified relations by $|\overline{y}|$ to compensate it, i.e. we get

$$\exists \overline{R'}\ \forall \overline{y}\,\overline{z}\varphi[P\overline{u}/\overline{u} \neq \overline{y}, R(\overline{x})/R'(\overline{x},\overline{y})].$$

Q.E.D.

Theorem 3.11. If $\psi \in$ SO-HORN, then the set of finite models of ψ, $\mathrm{Mod}_0(\psi)$, is in PTIME.

Proof. Given $\psi' \in$ SO-HORN, transform it to Σ^1_1-HORN, $\psi = \exists R_1 \ldots \exists R_m \forall \overline{z} \bigwedge_i C_i$. Given a finite structure $\mathfrak{A}$ reduce the problem of whether $\mathfrak{A} \models \psi$ to HORNSAT (as in the proof of the theorem of Cook and Levin).

- Omit quantifiers $\exists R_i$.
- Replace the universal quantifiers $\forall z_i \eta(z_i)$ by $\bigwedge_{a \in A} \eta[z_i/a]$.
- If there is a clause that is already made false by this interpretation, i.e. $C = 1 \wedge \ldots \wedge 1 \rightarrow 0$, reject ψ. Else interpret atoms $R_i\overline{u}$ as propositional variables.

The resulting formula is a propositional Horn formula with length polynomially bounded in $|A|$ and which is satisfiable iff $\mathfrak{A} \models \psi$. The satisfiability problem HORNSAT can be solved in linear time. Q.E.D.

Theorem 3.12 (Grädel)**.** On ordered finite structures SO-HORN and Σ^1_1-HORN capture PTIME.

Proof. We analyze the formula φ_M constructed in the proof of Fagin's Theorem in the case of a deterministic TM M. Recall that φ_M is the universal closure of START $\wedge$ NOCHANGE $\wedge$ CHANGE $\wedge$ END. START, NOCHANGE and END are already in Horn form. CHANGE has the form

$$\bigwedge_{q \in Q, \sigma \in \Sigma} (\mathrm{PRE}[q,\sigma] \rightarrow \bigvee_{(q',\sigma',m) \in \delta(q,\sigma)} \mathrm{POST}[q',\sigma',m]).$$

For a deterministic M for each (q,σ) there is a unique $\delta(q,\sigma) = (q',\sigma',m)$. In this case $\mathrm{PRE}[q,\sigma] \rightarrow \mathrm{POST}[q',\sigma',m]$ can be replaced by the conjunction of the Horn clauses

- $\mathrm{PRE}[q, \sigma] \to X_{q'}(\overline{t}')$
- $\mathrm{PRE}[q, \sigma] \to Y_{\sigma'}(\overline{t}', \overline{x})$
- $\mathrm{PRE}[q, \sigma] \wedge \overline{y} = \overline{x} + m \to Z(\overline{t}', \overline{y})$.

Q.E.D.

Remark 3.13. The assumption that a linear order is explicitly available cannot be eliminated, since linear orderings are not definable by Horn formulae.

4 LFP and Infinitary Logics

One of the distinguishing features of finite model theory compared with other branches of logic is the eminent role of various kinds of fixed-point logics. Fixed-point logics extend a basic logical formalism (such as first-order logic, conjunctive queries, or propositional modal logic) by a constructor for expressing *fixed points of relational operators*.

What do we mean by a *relational operator*? Note that any formula $\psi(R, \overline{x})$ of vocabulary $\tau \cup \{R\}$ where R is a relational symbol of arity k and $\overline{x}$ is a k-tuple of variables that are free in ψ can be viewed as defining, for every τ-structure $\mathfrak{A}$, an update operator $F_\psi : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$ on the class of k-ary relations on A, namely

$$F_\psi : R \mapsto \{\overline{a} : (\mathfrak{A}, R) \models \psi(R, \overline{a})\}.$$

A fixed point of F_ψ is a relation R for which $F_\psi(R) = R$. In general, a fixed point of F_ψ need not exist, or there may exist many of them. However, if R happens to occur only positively in ψ, then the operator F_ψ is monotone, and in that case there exists a *least* relation $R \subseteq A^k$ such that $F_\psi(R) = R$. The most influential fixed-point formalisms in logic are concerned with least (and greatest) fixed points, so we shall discuss these first. We start by reviewing the necessary mathematical foundations and we also show how least fixed-point logic is related to infinitary first-order logic.

4.1 Ordinals

The standard basic notion used in mathematics is the notion of a set, and all mathematical theorems follow from *the axioms of set theory*. The standard set of axioms is known as *Zermelo-Fraenkel Set Theory ZF*. These axioms guarantee, for instance, the existence of an empty set, an infinite

set, the power set of any set, and that no set is a member of itself (i.e. $\forall x\ \neg x \in x$). It is common in mathematics to extend ZF by *the axiom of choice AC* and to denote the resulting set of axioms by ZFC.

In particular, the notion of *numbers* can be formalised by sets. The standard way to do this is to start with the empty set, i.e. let $0 = \emptyset$, and proceed by induction, defining $n + 1 = n \cup \{n\}$. Here are the first few numbers in this representation:

- $0 = \emptyset$,
- $1 = \{\emptyset\}$,
- $2 = \{\emptyset, \{\emptyset\}\}$,
- $3 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$.

In this way we can construct all natural numbers. Observe that for each such number n (viewed as a set) it holds that

$$m \in n \implies m \subseteq n.$$

In particular, the relation $\in$ is *transitive* in such sets, i.e. if $k \in m$ and $m \in n$ then $k \in n$. We use this property of sets to define a more general class of numbers.

Definition 4.1. A set α is *an ordinal number* if $\in$ is transitive in α.

Besides natural numbers, what other ordinal numbers are there? The smallest example is $\omega = \bigcup_{n \in \mathbb{N}} n$, the union of all natural numbers. Indeed, it is easy to check that the union of ordinals is always an ordinal as well (as long as it is a set).

What is the next ordinal number after ω? We can again apply the $+1$ operation in the same way as for natural numbers, so

$$\omega + 1 = \omega \cup \{\omega\} = \{0, 1, 2, \ldots, \{0, 1, \ldots\}\}.$$

But does it make sense to say that $\omega + 1$ is the *next* ordinal, or, to put it more generally: is there an order on ordinals? In fact both, each ordinal as a set and all ordinals as a class, are well-ordered, i.e. the following holds:

- for any two ordinal numbers α and β either $\alpha \subseteq \beta$ or $\beta \subseteq \alpha$;

- there exists no infinite descending sequence of ordinals

$$\alpha_0 \supsetneq \alpha_1 \supsetneq \alpha_3 \supsetneq \cdots ;$$

- each ordinal α is well-ordered by $\in$.

Ordinals are intimately connected to well-orders, in fact any structure $(A, <)$ where $<$ is a well-order is isomorphic to some ordinal α. To get an intuition on how ordinals look like, consider the following examples of countable ordinals: $\omega + 1, \omega + \omega, \omega^2, \omega^3, \omega^\omega$.

The well-order of ordinals allows to define and prove the principle of *transfinite induction*. This principle states that the class of *all ordinals* is generated from $\emptyset$ by taking the successor ($+1$) and the union on limit steps, as shown on the examples before. Specifically, for each ordinal α it holds that either

- there exists an ordinal $\beta < \alpha$ such that $\alpha = \beta + 1 = \beta \cup \{\beta\}$, or
- there exist ordinals $\beta_\gamma < \alpha$ such that $\alpha = \bigcup_\gamma \beta_\gamma$.

Besides ordinals, we sometimes need cardinal numbers Cn which formalise the notion of cardinalities of sets. A *cardinal number* $\kappa \in \mathrm{Cn}$ is *a smallest ordinal number*, i.e. κ is an ordinal number with which no strictly smaller ordinal number can be put into bijection. For example, every natural number and ω itself are cardinal numbers, but $\omega^2 \notin \mathrm{Cn}$. We denote the class of infinite cardinal numbers by Cn^∞.

4.2 Some Fixed-Point Theory

There is a well-developed mathematical theory of fixed points of monotone operators on complete lattices. A *complete lattice* is a partial order $(A, \leq)$ such that each set $X \subseteq A$ has a supremum (a least upper bound) and an infimum (a greatest lower bound). Here we are interested mainly in power set lattices $(\mathcal{P}(A^k), \subseteq)$ (where A is the universe of a structure), and later in product lattices $(\mathcal{P}(B_1) \times \cdots \times \mathcal{P}(B_m), \subseteq)$. For simplicity, we shall describe the basic facts of fixed-point theory for lattices $(\mathcal{P}(B), \subseteq)$, where B is an arbitrary (finite or infinite) set.

Definition 4.2. Let $F : \mathcal{P}(B) \to \mathcal{P}(B)$ be an operator.

(1) $X \subseteq B$ is a *fixed point* of F if $F(X) = X$.

(2) A *least fixed point* or a *greatest fixed point* of F is a fixed point X of F such that $X \subseteq Y$ or $Y \subseteq X$, respectively, for each fixed point Y of F.

(3) F is *monotone*, if $X \subseteq Y \Longrightarrow F(X) \subseteq F(Y)$ for all $X, Y \subseteq B$.

Theorem 4.3 (Knaster and Tarski)**.** Every monotone operator $F : \mathcal{P}(B) \to \mathcal{P}(B)$ has a *least fixed point* $\mathrm{lfp}(F)$ and a *greatest fixed point* $\mathrm{gfp}(F)$. Further, these fixed points may be written as

$$\mathrm{lfp}(F) = \bigcap\{X : F(X) = X\} = \bigcap\{X : F(X) \subseteq X\}$$
$$\mathrm{gfp}(F) = \bigcup\{X : F(X) = X\} = \bigcup\{X : F(X) \supseteq X\}.$$

Proof. Let $S = \{X \subseteq B : F(X) \subseteq X\}$ and $Y = \bigcap S$. We first show that Y is a fixed point of F.

$F(Y) \subseteq Y$. Clearly, $Y \subseteq X$ for all $X \in S$. As F is monotone, it follows that $F(Y) \subseteq F(X) \subseteq X$. Hence $F(Y) \subseteq \bigcap S = Y$.

$Y \subseteq F(Y)$. As $F(Y) \subseteq Y$, we have $F(F(Y)) \subseteq F(Y)$, and hence $F(Y) \in S$. Thus $Y = \bigcap S \subseteq F(Y)$.

By definition, Y is contained in all X such that $F(X) \subseteq X$. In particular Y is contained in all fixed points of F. Hence Y is the least fixed point of F.

The argument for the greatest fixed point is analogous. Q.E.D.

Least fixed points can also be constructed inductively. We call an operator $F : \mathcal{P}(B) \to \mathcal{P}(B)$ *inductive* if the sequence of its *stages* X^α (where α is an ordinal), defined by

$$\begin{aligned} X^0 &:= \emptyset, \\ X^{\alpha+1} &:= F(X^\alpha), \text{ and} \\ X^\lambda &:= \bigcup_{\alpha<\lambda} X^\alpha \text{ for limit ordinals } \lambda, \end{aligned}$$

is increasing, i.e. if $X^\beta \subseteq X^\alpha$ for all $\beta < \alpha$. Obviously, monotone operators are inductive. The sequence of stages of an inductive operator eventually reaches a fixed point, which we denote by X^∞. The least

ordinal β for which $X^\beta = X^{\beta+1} = X^\infty$ is called $\mathrm{cl}(F)$, the *closure ordinal* of F.

Lemma 4.4. For every inductive operator $F : \mathcal{P}(B) \to \mathcal{P}(B)$, $|\mathrm{cl}(F)| \leq |B|$.

Proof. Let $|B|^+$ denote the smallest cardinal greater than $|B|$. Suppose that the claim is false for F. Then for each $\alpha < |B|^+$ there exists an element $x_\alpha \in X^{\alpha+1} - X^\alpha$. The set $\{x_\alpha : \alpha < |B|^+\}$ is a subset of B of cardinality $|B|^+ > |B|$, which is impossible. Q.E.D.

Proposition 4.5. For monotone operators, the inductively constructed fixed point coincides with the least fixed point, i.e. $X^\infty = \mathrm{lfp}(F)$.

Proof. As X^∞ is a fixed point, $\mathrm{lfp}(X) \subseteq X^\infty$. For the converse, we show by induction that $X^\alpha \subseteq \mathrm{lfp}(F)$ for all α. As $\mathrm{lfp}(F) = \bigcap\{Z : F(Z) \subseteq Z\}$, it suffices to show that X^α is contained in all Z for which $F(Z) \subseteq Z$.

For $\alpha = 0$, this is trivial. By monotonicity and the induction hypothesis, we have $X^{\alpha+1} = F(X^\alpha) \subseteq F(Z) \subseteq Z$. For limit ordinals λ with $X^\alpha \subseteq Z$ for all $\alpha < \lambda$ we also have $X^\lambda = \bigcup_{\alpha<\lambda} \subseteq Z$. Q.E.D.

The greatest fixed point can be constructed by a dual induction, starting with $Y^0 = B$, by setting $Y^{\alpha+1} := F(Y^\alpha)$ and $Y^\lambda = \bigcap_{\alpha<\lambda} Y^\alpha$ for limit ordinals. The *decreasing* sequence of these stages then eventually converges to the greatest fixed point $Y^\infty = \mathrm{gfp}(F)$.

The least and greatest fixed points are dual to each other. For every monotone operator F, the dual operator $F^d : X \mapsto \overline{F(\overline{X})}$ (where $\overline{X}$ denotes the complement of X) is also monotone, and we have that

$$\mathrm{lfp}(F) = \overline{\mathrm{gfp}(F^d)} \text{ and } \mathrm{gfp}(F) = \overline{\mathrm{lfp}(F^d)}.$$

Everything said so far holds for operators on arbitrary (finite or infinite) power set lattices. In *finite model theory*, we consider operators $F : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$ for finite A only. In this case the inductive constructions will reach the least or greatest fixed point in a polynomial number of steps. As a consequence, these fixed points can be constructed efficiently.

Lemma 4.6. Let $F : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$ be a monotone operator on a finite set A. If F is computable in polynomial time (with respect to $|A|$), then so are the fixed points $\mathrm{lfp}(F)$ and $\mathrm{gfp}(F)$.

4.3 Least Fixed-Point Logic

LFP is the logic obtained by adding least and greatest fixed points to first-order logic.

Definition 4.7. *Least fixed-point logic* (LFP) is defined by adding to the syntax of first-order logic the following *least fixed-point formation rule*: If $\psi(R, \overline{x})$ is a formula of vocabulary $\tau \cup \{R\}$ with only positive occurrences of R, if $\overline{x}$ is a tuple of variables, and if $\overline{t}$ is a tuple of terms (such that the lengths of $\overline{x}$ and $\overline{t}$ match the arity of R), then

$$[\mathrm{lfp}\, R\overline{x} \,.\, \psi](\overline{t}) \text{ and } [\mathrm{gfp}\, R\overline{x} \,.\, \psi](\overline{t})$$

are formulae of vocabulary τ. The free first-order variables of these formulae are those in $(\mathrm{free}(\psi) \setminus \{x : x \text{ in } \overline{x}\}) \cup \mathrm{free}(\overline{t})$.

Semantics. For any τ-structure $\mathfrak{A}$ providing interpretations for all free variables in the formula, we have that $\mathfrak{A} \models [\mathrm{lfp}\, R\overline{x} \,.\, \psi](\overline{t})$ if $\overline{t}^{\mathfrak{A}}$ (the tuple of elements of $\mathfrak{A}$ interpreting $\overline{t}$) is contained in $\mathrm{lfp}(F_\psi)$, where F_ψ is the update operator defined by ψ on $\mathfrak{A}$. The semantic for greatest fixed point operators is defined analogously.

Example 4.8. Here is a fixed-point formula that defines the transitive closure of the binary predicate E:

$$\mathrm{TC}(u, v) := [\mathrm{lfp}\, Txy \,.\, Exy \vee \exists z(Exz \wedge Tzy)](u, v).$$

Note that in a formula $[\mathrm{lfp}\, R\overline{x} \,.\, \varphi](\overline{t})$, there may be free variables in φ additional to those in $\overline{x}$, and these remain free in the fixed-point formula. They are often called *parameters* of the fixed-point formula. For instance, the transitive closure can also be defined by the formula

$$\varphi(u, v) := [\mathrm{lfp}\, Ty \,.\, Euy \vee \exists x(Tx \wedge Exy)](v)$$

which has u as a parameter.

It can be shown that every LFP-formula is equivalent to one without parameters (at the cost of increasing the arity of the fixed-point variables). The proof is left to the reader.

Example 4.9. Let $\varphi := \forall y(y < x \to Ry)$ and let $(A, <)$ be a partial order. The formula $[\mathrm{lfp}\, Rx \,.\, \varphi](x)$ then defines the well-founded part of $<$. The closure ordinal of F_φ on $(A, <)$ is the length of the longest well-founded initial segment of $<$, and $(A, <) \models \forall x[\mathrm{lfp}\, Rx \,.\, \varphi](x)$ if, and only if, $(A, <)$ is well-founded.

Example 4.10. The LFP-sentence

$$\psi := \forall y \exists z Fyz \wedge \forall y[\mathrm{lfp}\, Ry \,.\, \forall x(Fxy \to Rx)](y)$$

is an infinity axiom, i.e. it is satisfiable but does not have a finite model.

Example 4.11. The Game query asks, given a finite game $\mathcal{G} = (V, V_0, V_1, E)$, to compute the set of winning positions for Player 0. The Game query is LFP-definable, by use of $[\mathrm{lfp}\, Wx \,.\, \varphi](x)$ with

$$\varphi(W, x) := (V_0 x \wedge \exists y(Exy \wedge Wy)) \vee (V_1 \wedge \forall y(Exy \to Wy)).$$

The Game query plays an important role for LFP. It can be shown that every LFP-definable property of finite structures can be reduced to Game by a quantifier-free interpretation. Hence Game is complete for LFP via this notion of reduction, and thus a natural candidate if one is trying to separate a weaker logic from LFP.

Example 4.12. Maximal bisimulation B on a Kripke structure $\mathcal{K} = (K, \{E_j\}, \{P_j\})$ is defined by the formula $\psi(u, v) =$

$$\Big[\mathrm{gfp}\, Bxy.\Big(\bigwedge_i (P_i x \leftrightarrow P_i y) \wedge \bigwedge_j (\forall x'(E_j(x, x') \to \exists y'(E_j(y, y') \wedge B(x', y'))) \wedge \forall y'(E_j(y, y') \to \exists x'(E_j(x, x') \wedge B(x', y'))))\Big)\Big](u, v)\,,$$

i.e. $u^{\mathcal{K}}$ and $v^{\mathcal{K}}$ are bisimilar if and only if $\mathcal{K}, u^{\mathcal{K}}, v^{\mathcal{K}} \models \psi(u, v)$.

The duality between the least and greatest fixed points implies that for any formula ψ,

$$[\text{gfp } R\bar{x} \,.\, \psi](\bar{t}) \equiv \neg[\text{lfp } R\bar{x} \,.\, \neg\psi[R/\neg R]](\bar{t}),$$

where $\psi[R/\neg R]$ is the formula obtained from ψ by replacing all occurrences of R-atoms by their negations. (As R occurs only positively in ψ, the same is true for $\neg\psi[R/\neg R]$.) Because of this duality, greatest fixed points are often omitted in the definition of LFP. On the other hand, it is sometimes convenient to keep the greatest fixed points, and to use the duality (and de Morgan's laws) to translate LFP-formulae to *negation normal form*, i.e. to push negations all the way to the atoms.

4.3.1 Capturing Polynomial Time

From the fact that first-order operations are polynomial-time computable and from Lemma 4.6, we can conclude that every LFP-definable property of finite structures is computable in polynomial time.

Proposition 4.13. Let ψ be a sentence in LFP. It is decidable in polynomial time whether a given finite structure $\mathfrak{A}$ is a model of ψ. In short, LFP $\subseteq$ PTIME.

Obviously LFP, is a fragment of second-order logic. Indeed, by the Knaster-Tarski Theorem,

$$[\text{lfp } R\bar{x} \,.\, \psi(R,\bar{x})](\bar{y}) \equiv \forall R((\forall\bar{x}(\psi(R,\bar{x}) \to R\bar{x})) \to R\bar{y}).$$

We next relate LFP to SO-HORN.

Theorem 4.14. Every formula $\psi \in$ SO-HORN is equivalent to some formula $\psi^* \in$ LFP.

Proof. By Theorem 3.9, we can assume that $\psi = (\exists R_1)\cdots(\exists R_m)\varphi \in \Sigma^1_1$-HORN. By combining the predicates $R_1, \ldots, R_m$ into a single predicate R of larger arity and by renaming variables, it is easy to transform

ψ into an equivalent formula

$$\psi' := \exists R \forall \overline{x} \forall \overline{y} \bigwedge_i C_i \wedge \bigwedge_j D_j,$$

where the C_i are clauses of the form $R\overline{x} \leftarrow \alpha_i(R, \overline{x}, \overline{y})$ (with exactly the same head $R\overline{x}$ for every i) and the D_j are clauses of the form $0 \leftarrow \beta_j(R, \overline{x}, \overline{y})$. The clauses C_i define, on every structure $\mathfrak{A}$, a monotone operator $F : R \mapsto \{\overline{x} : \bigvee_i \exists \overline{y} \alpha_i(\overline{x}, \overline{y})\}$. Let R^ω be the least fixed point of this operator. Obviously $\mathfrak{A} \models \neg\psi$ if and only if $\mathfrak{A} \models \beta_i(R^\omega, \overline{a}, \overline{b})$ for some i and some tuple $\overline{a}$, $\overline{b}$. But R^ω is defined by the fixed-point formula

$$\alpha^\omega(\overline{x}) := [\mathrm{lfp}\, R\overline{x} \,.\, \bigvee_i \exists \overline{y} \alpha_i(\overline{x}, \overline{y})](\overline{x}).$$

Hence, for $\beta := \exists \overline{x} \exists \overline{y} \bigvee_j \beta_j(\overline{x}, \overline{y})$, ψ is equivalent to the formula $\psi^* := \neg\beta[R\overline{z}/\alpha^\omega(\overline{z})]$ obtained from $\neg\beta$ by substituting all occurrences of atoms $R\overline{z}$ by $\alpha^\omega(\overline{z})$. Clearly, this formula is in LFP. Q.E.D.

Hence SO-HORN $\leq$ LFP $\leq$ SO. As an immediate consequence of Theorems 3.12 and 4.14 we obtain the Immerman–Vardi Theorem.

Theorem 4.15 (Immerman and Vardi)**.** On ordered structures, least fixed-point logic captures polynomial time.

However, on unordered structures, SO-HORN is strictly weaker than LFP.

4.4 Infinitary First-Order Logic

Definition 4.16. Let $\kappa \in \mathrm{Cn}^\infty$ be an infinite cardinal number and τ a signature. The *infinitary logic* $L_{\kappa\omega}(\tau)$ is inductively defined as follows.

- Each atomic formula in $\mathrm{FO}(\tau)$ is in $L_{\kappa\omega}(\tau)$.
- If $\varphi \in L_{\kappa\omega}(\tau)$, then also $\neg\varphi$, $\exists x\varphi$, $\forall x\varphi \in L_{\kappa\omega}(\tau)$.
- If $\Phi \subseteq L_{\kappa\omega}(\tau)$ is a set of formulae with $|\Phi| < \kappa$, then $\bigvee \Phi$, $\bigwedge \Phi \in L_{\kappa\omega}(\tau)$.

Further, we write $L_{\infty\omega}(\tau)$ for $\bigcup_{\kappa \in \mathrm{Cn}^\infty} L_{\kappa\omega}(\tau)$.

Note that the second parameter ω is always fixed as an index of our logics. This indicates that we only allow finite sequences of quantifiers.

The logic $L_{\omega\omega}(\tau)$ is precisely the logic $\mathrm{FO}(\tau)$. The logic $L_{\aleph_1\omega}(\tau)$, in which disjunctions and conjunctions can be built over countable sets of formulae, is denoted by $L_{\omega_1\omega}$.

The semantics of the infinitary logic is defined in an obvious way. Clearly, we only have to treat the cases of $\bigwedge\Phi$ and $\bigvee\Phi$. Let $\overline{a} \subseteq A$ be an assignment of at most κ free variables, then

- $\mathfrak{A}, \overline{a} \models \bigwedge\Phi$ if and only if $\mathfrak{A}, \overline{a} \models \varphi$ for all $\varphi \in \Phi$.
- $\mathfrak{A}, \overline{a} \models \bigvee\Phi$ if and only if there exists a $\varphi \in \Phi$ such that $\mathfrak{A}, \overline{a} \models \varphi$.

In all other cases the semantics of infinitary logic coincides with that of first-order logic.

Example 4.17. Finiteness can be expressed in $L_{\omega_1\omega}$. Let

$$\varphi_{\geq n} := \exists x_1 \dots \exists x_n \bigwedge_{1 \leq i < j \leq n} (x_i \neq x_j)$$

and $\varphi_{\mathrm{fin}} := \bigvee\{\neg\varphi_{\geq n} \mid n < \omega\} \in L_{\omega_1\omega}$. Then for each structure $\mathfrak{A}$, $\mathfrak{A} \models \varphi_{\mathrm{fin}}$ if and only if $\mathfrak{A}$ is finite.

Remark 4.18. The Compactness Theorem does not hold for the logic $L_{\omega_1\omega}$. Consider for example the set of formulas $\varphi_{\mathrm{fin}} \cup \{\varphi_{\geq n} \mid n < \omega\}$. It is unsatisfiable, but each of its finite subsets is satisfiable.

Theorem 4.19. Let $\kappa \in \mathrm{Cn}^\infty$. For each formula $\varphi(\overline{x}) \in \mathrm{LFP}$ there is a formula $\widehat{\varphi} \in L_{\kappa\omega}$ such that for all structures $\mathfrak{A}$ with $|\mathfrak{A}| < \kappa$ and all $\overline{a} \subseteq A$, we have $\mathfrak{A} \models \varphi(\overline{a})$ if and only if $\mathfrak{A} \models \widehat{\varphi}(\overline{a})$.

Proof. By using the duality between least and greatest fixed points we may assume that formulas in LFP only contain operators expressing least fixed points. We inductively define the translation from LFP to formulas of $L_{\infty\omega}$ as follows:

- for atomic formulas ψ we set $\widehat{\psi} = \psi$,
- $\widehat{\neg\psi} = \neg\widehat{\psi}$,
- $\widehat{\psi_1 \wedge \psi_2} = \widehat{\psi_1} \wedge \widehat{\psi_2}$, and $\widehat{\psi_1 \vee \psi_2} = \widehat{\psi_1} \vee \widehat{\psi_2}$.

For the case of $[\widehat{\text{lfp } R\bar{x}.\psi}](\bar{t})$, we build by transfinite induction a sequence of formulas $\psi^\alpha(\bar{x})$ for all ordinals $\alpha \leq \kappa$. These formulas intuitively correspond to the stages in the inductive evaluation of the least fixed-point. Accordingly, we start with the empty relation and set $\psi^0(\bar{x}) = \bot$. The induction proceeds as follows:

- $\psi^{\alpha+1}(\bar{x}) = \widehat{\psi}[R\bar{z}/\psi^\alpha(\bar{z})]$,
- for $\alpha = \bigcup_{\beta<\alpha} \beta$, let $\psi^\alpha(\bar{x}) = \bigvee_{\beta<\alpha}\{\psi^\beta(\bar{x}) \mid \beta < \alpha\}$.

Using induction on α and the definition of the semantics of $L_{\infty\omega}$, we see that the formulas ψ^α correspond exactly to the stages of fixed-point induction, i.e. $R^\alpha = \{\bar{x} \mid \psi^\alpha(\bar{x})\}$.

On structures $\mathfrak{A}$ with $|\mathfrak{A}| < \kappa$ we have $R^\kappa = R^\infty$ is the least fixed-point and which is thus defined by $\psi^\kappa(\bar{x})$. The claim follows. Q.E.D.

In general we can not drop the condition of bounded cardinality of the structures. In fact, the class of all well-orderings is definable in LFP by the following sentence:

$$\psi_{\text{wo}} := \varphi_{\text{lin}} \wedge \forall x[\text{lfp } Wx(\forall y(y < x \to Wy))](x),$$

where φ_{lin} is a formula that expresses that $<$ is a linear order. One can show that this class is not definable in $L_{\infty\omega}$.

We also observe that the structure $(\omega, 0, S)$ is axiomatizable in $\text{LFP}(0, S)$ up to isomorphism. To see this, note that $\{S^n(0) \mid n < \omega\}$ is the least fixed-point of the expression $x = 0 \vee \exists y(Ry \wedge Sy = x)$ (with respect to the variable R). Thus, $(\omega, 0, S)$ can be axiomatized by

$$\forall x \forall y(Sx = Sy \to x = y) \wedge (\forall x Sx \neq 0) \wedge \\ \forall x[\text{lfp } Rx(x = 0 \vee \exists y(Ry \wedge Sy = x))](x).$$

(The two first formulae in the conjunction are the first Peano axioms.) We conclude that the upward Löwenheim-Skolem theorem for LFP fails:

Remark 4.20. There exists a sentence $\varphi \in \text{LFP}$ that has an infinite model and no uncountable model.

Next, we want to show that the Compactness Theorem does not

hold for LFP either. For this we give an LFP(S)-sentence φ such that φ has arbitrary large finite models, but no infinite one.

Theorem 4.21. There is a sentence $\varphi \in \mathrm{LFP}(S)$ where S is a function symbol of arity one such that φ has arbitrary large finite models, but no infinite one.

Proof. Define

$$\psi(x,z) := [\mathrm{lfp}\, Rx.(x = z \vee \exists y(Ry \wedge Sy = x))](x).$$

If $\mathfrak{A}$ is an S-structure then for all elements $a, b \in A$, we have $\mathfrak{A} \models \psi(b,a)$ if and only if there is some $n < \omega$ such that $(S^{\mathfrak{A}})^n(a) = b$. Now let

$$\varphi := \forall x \exists y(Sy = x) \wedge \exists x \forall y \psi(y,x).$$

For some S-structure $\mathfrak{A}$, we have $\mathfrak{A} \models \varphi$ if and only if $S^{\mathfrak{A}}$ is surjective and there is an $a \in A$ that generates the whole structure in the sense that $A = \{(S^{\mathfrak{A}})^n(a) \mid n < \omega\}$. For any $n < \omega$, the structure $\mathfrak{A} = (\{1,\dots,n\}, S^{\mathfrak{A}})$ where $S^{\mathfrak{A}}(k) = k+1$ for $k \in \{1,\dots,n-1\}$ and $S^{\mathfrak{A}}(n) = 1$ is a model of φ. Thus, φ has arbitrary large finite models.

On the other hand, φ has no infinite model. Let $\mathfrak{A} = (A, S^{\mathfrak{A}})$ be an S-structure with an infinite universe A such that there is an $a \in A$ with $A = \{(S^{\mathfrak{A}})^n(a) \mid n < \omega\}$, then $a \notin \mathrm{Img}(S^{\mathfrak{A}})$, so $S^{\mathfrak{A}}$ is not surjective. Indeed, assume that $a \in \mathrm{Img}(S^{\mathfrak{A}})$. Then $a = S^{\mathfrak{A}}(b)$ for some $b \in A$. Because $A = \{(S^{\mathfrak{A}})^n(a) \mid n < \omega\}$, it would follow $b = (S^{\mathfrak{A}})^n(a)$, so $(S^{\mathfrak{A}})^{n+1}(a) = a$. Then it would be $|\{(S^{\mathfrak{A}})^n(a) \mid n < \omega\}| \leq n$, in contradiction to the fact that A is infinite and $A = \{(S^{\mathfrak{A}})^n(a) \mid n < \omega\}$. It follows that $\mathfrak{A} \not\models \varphi$, and the statement is proven. Q.E.D.

Corollary 4.22. There exists an unsatisfiable set of sentences $\Phi \subseteq \mathrm{LFP}$ such that every finite subset of Φ is satisfiable, i.e. the Compactness Theorem fails for LFP.

Proof. According to Theorem 4.21 there is a sentence $\varphi \in \mathrm{LFP}(S)$ that has arbitrary large finite models, but no infinite one. As before, consider the set of sentences $\Phi = \{\varphi\} \cup \{\exists x_1 \dots \exists x_n \bigwedge_{i<j} x_i \neq x_j : n \in \omega\}$. Q.E.D.

We mention yet another property of LFP, that we do not prove here: the downward Löwenheim-Skolem theorem holds for LFP.

Theorem 4.23. Let $\varphi \in \mathrm{LFP}$ be a satisfiable sentence. Then φ has a countable model.

In particular, it follows that there is a sentence in $\varphi \in L_{\infty\omega}(\tau)$ for some appropriate signature τ that is not equivalent to any sentence $\psi \in \mathrm{LFP}(\tau)$. For example, we can choose an uncountable set of constant symbols as τ and a conjunction of all sentences $c \neq d$ for pairwise distinct $c, d \in \tau$ as φ, which has no countable model.

5 Modal, Inflationary and Partial Fixed Points

In finite model theory, a number of other fixed-point logics, in addition to LFP, play an important role. The structure, expressive power, and algorithmic properties of these logics have been studied intensively, and we review these results in this chapter.

5.1 The Modal μ-Calculus

A fragment of LFP that is of fundamental importance in many areas of computer science (e.g. controller synthesis, hardware verification, and knowledge representation) is the modal μ-calculus (L_μ). It is obtained by adding least and greatest fixed points to propositional modal logic (ML). In this way L_μ relates to ML in the same way as LFP relates to FO.

Definition 5.1. The *modal μ-calculus* L_μ extends ML (including propositional variables $X, Y, \ldots$, which can be viewed as monadic second-order variables) by the following rule for building fixed point formulae: If ψ is a formula in L_μ and X is a propositional variable that only occurs positively in ψ, then $\mu X.\psi$ and $\nu X.\psi$ are also L_μ-formulae.

The semantics of these fixed-point formulae is completely analogous to that for LFP. The formula ψ defines on G (with universe V, and with interpretations for other free second-order variables that ψ may have besides X) the monotone operator $F_\psi : \mathcal{P}(V) \to \mathcal{P}(V)$ assigning to every set $X \subseteq V$ the set $\psi^G(X) := \{v \in V : (G, X), v \models \psi\}$. The semantics of fixed-points is defined by

$$G, v \models \mu X.\psi \text{ iff } v \in \mathrm{lfp}(F_\psi)$$
$$G, v \models \nu X.\psi \text{ iff } v \in \mathrm{gfp}(F_\psi).$$

Example 5.2. The formula $\mu X.(\varphi \vee \langle a \rangle X)$ asserts that there exists a path along a-transitions to a node where φ holds.

The formula $\psi := \nu X.\Big((\bigvee_{a \in A} \langle a \rangle \mathit{true}) \wedge (\bigwedge_{a \in A} [a] X)\Big)$ expresses the assertion that the given transition system is deadlock-free. In other words, $G, v \models \psi$ if no path from v in G reaches a dead end (i.e. a node without outgoing transitions).

Finally, the formula $\nu X.\mu Y.\Big(\langle a \rangle \big((\varphi \wedge X) \vee Y\big)\Big)$ says that there exists a path from the current node on which φ holds infinitely often.

The embedding from ML into FO is readily extended to a translation from L_μ into LFP, by inductively replacing formulas of the form $\mu X.\varphi$ by $[\mathrm{lfp}\, Xx.\varphi^*](x)$.

Proposition 5.3. Every formula $\psi \in L_\mu$ is equivalent to a formula $\psi^*(x) \in \mathrm{LFP}$.

Further the argument proving that LFP can be embedded into SO also shows that L_μ is a fragment of MSO.

As for LFP, a fixed μ-calculus formula can be evaluated on a structure $\mathfrak{A}$ in time polynomial in $|\mathfrak{A}|$. The question whether evaluating μ-calculus formulas on a structure when both the formula and the structure are part of the input is in PTIME is a major open problem. On the other hand, it is not difficult to see that the μ-calculus does not suffice to capture PTIME, even in very restricted scenarios such as word structures. Indeed, as L_μ is a fragment of MSO, it can only define *regular languages*, and of course, not all PTIME-languages are regular. However, we shall see in Section 5.5 that there is a multidimensional variant of L_μ that captures the *bisimulation-invariant* fragment of PTIME. Before we do this, let us first show that L_μ is itself invariant under bisimulation. To this end, we translate L_μ formulas into formulas of *infinitary modal logic* $ML_{\infty\omega}$, similar to the embedding of LFP into $L_{\infty\omega}$.

5.1.1 Infinitary Modal Logic and Bisimulation Invariance

Infinitary modal logic extends ML in an analogous way as how infinitary first-order logic extends FO.

Definition 5.4. Let $\kappa \in \mathrm{Cn}^\infty$ be an infinite cardinal number. The *infinitary logic* $ML_{\kappa\omega}$ is inductively defined as follows.

- Predicates P_i are in $ML_{\kappa\omega}$.
- If $\varphi \in ML_{\kappa\omega}$, then also $\neg\varphi$, $\Box\varphi$, $\Diamond\varphi \in ML_{\kappa\omega}$.
- If $\Phi \subseteq ML_{\kappa\omega}$ is a set of formulae with $|\Phi| < \kappa$, then $\bigvee \Phi$, $\bigwedge \Phi \in ML_{\kappa\omega}$.

Further, we write $ML_{\infty\omega}$ to denote $\bigcup_{\kappa\in\mathrm{Cn}^\infty} ML_{\kappa\omega}$.

The semantics of $ML_{\infty\omega}$ on Kripke structures is defined analogously to the semantics of ML, with the following obvious extension for the case of infinite disjuntions and conjunctions.

- $\mathcal{K}, v \models \bigwedge \Phi$ if and only if $\mathcal{K}, v \models \varphi$ for all $\varphi \in \Phi$.
- $\mathcal{K}, v \models \bigvee \Phi$ if and only if there exists a $\varphi \in \Phi$ such that $\mathcal{K}, v \models \varphi$.

The same proof that shows invariance of ML under bisimulation works for $ML_{\infty\omega}$, because the introduction of infinite conjunctions and disjunctions does not interfere with the arguments in the proof at all.

Theorem 5.5. The logic $ML_{\infty\omega}$ is invariant under bisimulation, i.e. if $\varphi \in ML_{\infty\omega}$ is a formula and $\mathcal{K}, v \sim \mathcal{K}', v'$ are two bisimilar Kripke structures, then

$$\mathcal{K}, v \models \varphi \text{ iff } \mathcal{K}', v' \models \varphi.$$

Similarly, the proof of Theorem 5.6 can be adapted to give a translation from L_μ formulas to $ML_{\infty\omega}$, as stated below.

Theorem 5.6. Let $\kappa \in \mathrm{Cn}^\infty$. For each formula $\varphi \in L_\mu$ there exists a formula $\widehat{\varphi} \in ML_{\kappa\omega}$ such that for all transition systems $\mathcal{K}$ with $|\mathcal{K}| < \kappa$ and all $v \in \mathcal{K}$, we have $\mathcal{K}, v \models \varphi$ if and only if $\mathcal{K}, v \models \widehat{\varphi}$.

Combining these two theorems, we get bisimulation invariance of L_μ.

Corollary 5.7. The logic L_μ is invariant under bisimulation.

5.2 Inflationary Fixed-Point Logic

LFP is only one instance of a logic with an explicit operator for forming fixed points. A number of other fixed-point extensions of first-order logic (or fragments of it) have been extensively studied in finite model theory. These include inflationary, partial, non-deterministic, and alternating fixed-point logics. All of these have in common that they allow the construction of fixed points of operators that are not necessarily monotone.

An operator $G : \mathcal{P}(B) \to \mathcal{P}(B)$ is called *inflationary* if $G(X) \supseteq X$ for all $X \subseteq B$. With any operator F one can associate an inflationary operator G, defined by $G(X) := X \cup F(X)$. In particular, inflationary operators are inductive, so iterating G yields a fixed point, called the *inflationary fixed point* of F.

To be more precise, the inflationary fixed-point of any operator $F\colon \mathcal{P}(B) \to \mathcal{P}(B)$ is defined as the limit of the increasing sequence of sets (R^α) defined as $R^0 := \emptyset$, $R^{\alpha+1} := R^\alpha \cup F(R^\alpha)$, and $R^\lambda := \bigcup_{\alpha<\lambda} R^\alpha$ for limit ordinals λ. The *deflationary fixed point* of F is constructed in the dual way starting with B as the initial stage and taking intersections at successor and limit ordinals.

Remark 5.8.

(1) Monotone operators need not be inflationary, and inflationary operators need not be monotone.
(2) An inflationary operator need not have a least fixed point.
(3) The least fixed point of an inflationary operator (if it exists) may be different from the inductive fixed point.
(4) However, if F is a monotone operator, then its inflationary fixed point and its least fixed point coincide.

The logic IFP is defined with a syntax similar to that of LFP, but without the requirement that the fixed-point variable occurs only positively in the formula defining the operator, and with semantics given by the associated inflationary operator.

Definition 5.9. IFP is the extension of first-order logic by the following fixed-point formation rules. For every formula $\psi(R, \overline{x})$, every tuple

$\overline{x}$ of variables, and every tuple $\overline{t}$ of terms (such that the lengths of $\overline{x}$ and $\overline{t}$ match the arity of R), we can build formulas $[\mathrm{ifp}\, R\overline{x}\ .\ \psi](\overline{t})$ and $[\mathrm{dfp}\, R\overline{x}\ .\ \psi](\overline{t})$.

Semantics. For a given structure $\mathfrak{A}$, we have that $\mathfrak{A} \models [\mathrm{ifp}\, R\overline{x}\ .\ \psi](\overline{t})$ and $\mathfrak{A} \models [\mathrm{dfp}\, R\overline{x}\ .\ \psi](\overline{t})$ if $\overline{t}^{\mathfrak{A}}$ is contained in the inflationary and deflationary fixed point of F_ψ, respectively.

By the last item of Remark 5.8, least and inflationary inductions are equivalent for positive formulae, and hence IFP is at least as expressive as LFP. On finite structures, inflationary inductions reach the fixed point after a polynomial number of iterations, hence every IFP-definable class of finite structures is decidable in polynomial time.

Proposition 5.10. IFP captures PTIME on ordered finite structures.

5.2.1 Least Versus Inflationary Fixed-Points

As both logics capture PTIME, IFP and LFP are equivalent on ordered finite structures. What about unordered structures? It was shown by Gurevich and Shelah that the equivalence of IFP and LFP holds on all finite structures. Their proof does not work on infinite structures, and indeed there are some important aspects in which least and inflationary inductions behave differently. For instance, there are first-order operators (on arithmetic, say) whose inflationary fixed point is not definable as the least fixed point of a first-order operator. Further, the alternation hierarchy in LFP is strict, whereas IFP has a positive normal form (see Proposition 5.17 below). Hence it was conjectured by many that IFP might be more powerful than LFP. However, Kreutzer showed recently that IFP is equivalent to LFP on arbitrary structures. Both proofs, by Gurevich and Shelah and by Kreutzer, rely on constructions showing that the *stage comparison relations* of inflationary inductions are definable by lfp inductions.

Definition 5.11. For every inductive operator $F : \mathcal{P}(B) \to \mathcal{P}(B)$, with stages X^α and an inductive fixed point X^∞, the *F-rank* of an element $b \in B$ is $|b|_F := \min\{\alpha : b \in X^\alpha\}$ if $b \in X^\infty$, and $|b|_F = \infty$ otherwise.

The *stage comparison relations* of G are defined by

$$a \leq_F b \quad \text{iff} \quad |a|_F \leq |b|_F < \infty$$
$$a \prec_F b \quad \text{iff} \quad |a|_F < |b|_F.$$

Given a formula $\varphi(R, \overline{x})$, we write $\leq_\varphi$ and $\prec_\varphi$ for the stage comparison relations defined by the operator F_φ (assuming that it is indeed inductive), and $\leq_\varphi^{\text{inf}}$ and $\prec_\varphi^{\text{inf}}$ for the stage comparison relations of the associated inflationary operator $G_\varphi : R \mapsto R \cup \{\overline{a} : \mathfrak{A} \models \varphi(R, \overline{a})\}$.

Example 5.12. For the formula $\varphi(T, x, y) := Exy \vee \exists z(Exz \wedge Tzy)$ the relation $\prec_\varphi$ on a graph (V, E) is distance comparison:

$$(a, b) \prec_\varphi (c, d) \text{ iff } \text{dist}(a, b) < \text{dist}(c, d).$$

Stage comparison theorems are results about the definability of stage comparison relations. For instance, Moschovakis proved that the stage comparison relations $\leq_\varphi$ and $\prec_\varphi$ of any positive first-order formula φ are definable by a simultaneous induction over positive first-order formulae. For results on the equivalence of IFP and LFP one needs a stage comparison theorem for IFP inductions.

We first observe that the stage comparison relations for IFP inductions are easily definable in IFP. For any formula $\varphi(T, \bar{x})$ with free variables $\overline{x}$ and free occuring predicate T, the stage comparison relation $\prec_\varphi^{\text{inf}}$ is defined by the formula

$$\psi(\overline{x}'\overline{y}') = [\text{ifp}\, \overline{w} \prec \overline{z}\,.\, \varphi[T\overline{u}/\overline{u} \prec \overline{w}](\overline{w}) \wedge \neg\varphi[T\overline{u}/\overline{u} \prec \overline{z}](\overline{z})](\overline{x}', \overline{y}').$$

Here we syntactically substitute $T, \overline{u}$ by $\overline{u} \prec \overline{w}$ in $\varphi(T\overline{x})$ and, additionally, free variables again by $\overline{w}$. (Note that $\overline{u}$ may contain free variables.) In $\neg\varphi(T, \overline{x})$, we substitute $T, \overline{u}$ by $\overline{u} \prec \overline{z}$ and, additionally, free variables again by $\overline{z}$. Thus free variables become parameter variables of the fixed-point. Now, for the first iteration, T_0 is empty as well as $\prec_0$, so the formula $\varphi(T_0, \overline{w})$ is satisfied by the same $\overline{a}$ as $\varphi(\prec_0, \overline{w})$. So in the first interation, the first components of $\prec_1$ contain the same elements as T_1. The second components of $\prec_1$ contain all other elements. In general, in the i-th iteration, $\prec_i$ consists of pairs $(\overline{a}, \overline{b})$ such that $\overline{a} \in T_i$

and $\overline{b} \notin T_i$. In the next step, precisely those $\overline{a}$ satisfy $\varphi[T\overline{u}/\overline{u} \prec \overline{w}](\prec_i)$ that satisfy $\varphi(T_i)$ (instead of $\varphi[T, \overline{u}]$ we now have $\varphi[\overline{u} \prec \overline{w}]$, i.e. $T\overline{a}$ holds if and only if $\overline{u} \prec \overline{a}$ holds if and only if $\overline{a}$ has come to T in the previous steps). So those $\overline{b}$ that do not satisfy $\varphi[T\overline{u}/\overline{u} \prec \overline{w}](\prec_i)$, satisfy $\neg\varphi[T\overline{u}/\overline{u} \prec \overline{w}](\prec_i)$. Summing up, pairs $\overline{a}, \overline{b}$ are included to $\prec_{i+1}$ if and only if $\overline{a}$ is included into T_{i+1}, but not earlier, and $\overline{b}$ is not in T_{i+1}.

However, what we need to show is that the stage comparison relation for IFP inductions is in fact LFP-definable.

Theorem 5.13 (Inflationary Stage Comparison)**.** For any formula $\varphi(R, \overline{x})$ in FO or LFP, the stage comparison relation $\prec_\varphi^{\inf}$ is definable in LFP. On finite structures, it is even definable in positive LFP.

From this result, the equivalence of LFP and IFP follows easily.

Theorem 5.14 (Kreutzer)**.** For every IFP-formula, there is an equivalent LFP-formula.

Proof. For any formula $\varphi(R, \overline{x})$,

$$[\text{ifp } R\overline{x} \,.\, \varphi](\overline{x}) \equiv \varphi(\{\overline{y} : \overline{y} \prec_\varphi^{\inf} \overline{x}\}, \overline{x}).$$

This holds because, by definition, an inductive fixed-point can only increase. Thus a tuple is added to it if and only if there is a stage, at which the relation R contains all previously added elements (thus $R = \{\overline{y} : \overline{y} \prec_\varphi^{\inf} \overline{x}\}$), and at that stage $\varphi(R, \overline{x})$ holds. Due to Theorem 5.13, the relation $\{\overline{y} : \overline{y} \prec_\varphi^{\inf} \overline{x}\}$ is definable in LFP, so the statement follows directly. Q.E.D.

Positive LFP. While LFP and the modal μ-calculus allow arbitrary nesting of least and greatest fixed points, and arbitrary interleaving of fixed points with Boolean operations and quantifiers, we can also ask about their more restricted forms. Let LFP_1 (sometimes also called positive LFP) be the extension of first-order logic that is obtained by taking least fixed points of positive first-order formulae (without parameters) and closing them under disjunction, conjunction, and existential and universal quantification, but *not* under negation. LFP_1 can be

conveniently characterized in terms of simultaneous least fixed points, defined in the next chapter.

Theorem 5.15. A relation is definable in LFP_1 if and only if it is definable by a formula of the form $[\mathrm{lfp}\, R : S](\overline{x})$, where S is a system of update rules $R_i\overline{x} := \varphi_i(\overline{R}, \overline{x})$ with first-order formulae φ_i. Moreover, we can require, without diminishing the expressive power, that each of the formulae φ_i in the system is either a purely existential formula or a purely universal formula.

One interesting consequence of the stage comparison theorems is that on finite structures, greatest fixed points (i.e. negations of least fixed points) can be expressed in positive LFP. This gives a normal form for LFP and IFP.

Theorem 5.16 (Immerman)**.** On finite structures, every LFP-formula (and hence also every IFP-formula) is equivalent to a formula in LFP_1.

This result fails on infinite structures. On infinite structures, there exist LFP formulae that are not equivalent to positive formulae, and in fact the alternation hierarchy of least and greatest fixed points is strict. This is not the case for IFP.

Proposition 5.17. It can be proven that every IFP-formula is equivalent to one that uses ifp-operators only positively.

Proof. Assume that structures contain at least two elements and that a constant 0 is available. Then a formula $\neg[\mathrm{ifp}\, R\overline{x} \,.\, \psi(R, \overline{x})]$ is equivalent to an inflationary induction on a predicate $T\overline{x}\,y$ which, for $y \neq 0$, simulates the induction defined by ψ, checks whether the fixed point has been reached, and then makes atoms $T\overline{x}0$ true if $\overline{x}$ is not contained in the fixed point. Q.E.D.

In finite model theory, owing to the Gurevich-Shelah Theorem, the two logics LFP and IFP have often been used interchangeably. However, there are significant differences that are sometimes overlooked. Despite the equivalence of IFP and LFP, inflationary inductions are a more powerful concept than monotone inductions. The translation from IFP-formulae to equivalent LFP-formulae can make the formulae much more

complicated, requires an increase in the arity of fixed-point variables and, in the case of infinite structures, introduces alternations between least and greatest fixed points. Therefore it is often more convenient to use inflationary inductions in explicit constructions, the advantage being that one is not restricted to inductions over positive formulae. For an example, see the proof of Theorem 5.29 below. Furthermore, IFP is more robust, in the sense that inflationary fixed points remain well defined even when other non-monotone operators (e.g. generalized quantifiers) are added to the language.

5.3 Simultaneous Inductions

A more general variant of LFP permits simultaneous inductions over several formulae. A simultaneous induction is based on a system of operators of the form

$$\begin{aligned} F_1 : \mathcal{P}(B_1) \times \cdots \times \mathcal{P}(B_m) &\to \mathcal{P}(B_1) \\ &\vdots \\ F_m : \mathcal{P}(B_1) \times \cdots \times \mathcal{P}(B_m) &\to \mathcal{P}(B_m), \end{aligned}$$

forming together an operator

$$F = (F_1, \ldots, F_m) : \mathcal{P}(B_1) \times \cdots \times \mathcal{P}(B_m) \to \mathcal{P}(B_1) \times \cdots \times \mathcal{P}(B_m).$$

Inclusion on the product lattice $\mathcal{P}(B_1) \times \cdots \times \mathcal{P}(B_m)$ is componentwise. Accordingly, F is monotone if, whenever $X_i \subseteq Y_i$ for all i, then also $F_i(\overline{X}) \subseteq F_i(\overline{Y})$ for all i.

Everything said above about least and greatest fixed points carries over to simultaneous induction. In particular, a monotone operator F has a least fixed point $\mathrm{lfp}(F)$ which can be constructed inductively, starting with $\overline{X}^0 = (\emptyset, \ldots, \emptyset)$ and iterating F until a fixed point $\overline{X}^\infty$ is reached.

One can extend the logic LFP by a simultaneous fixed point formation rule.

Definition 5.18. *Simultaneous least fixed-point logic*, denoted by S-LFP, is the extension of first-order logic by the following rule.

Syntax. Let $\psi_1(\overline{R}, \overline{x}_1), \ldots, \psi_m(\overline{R}, \overline{x}_m)$ be formulae of vocabulary $\tau \cup \{R_1, \ldots, R_m\}$, with only positive occurrences of $R_1, \ldots, R_m$, and, for each $i \leq m$, let $\overline{x}_i$ be a sequence of variables matching the arity of R_i. Then

$$S := \begin{cases} R_1\overline{x}_1 & := \quad \psi_1 \\ & \vdots \\ R_m\overline{x}_m & := \quad \psi_m \end{cases}$$

is a *system of update rules*, which is used to build formulae $[\text{lfp } R_i : S](\overline{t})$ and $[\text{gfp } R_i : S](\overline{t})$ (for any tuple $\overline{t}$ of terms whose length matches the arity of R_i).

Semantics. On each structure $\mathfrak{A}$, S defines a monotone operator $S^{\mathfrak{A}} = (S_1, \ldots, S_m)$ mapping tuples $\overline{R} = (R_1, \ldots, R_m)$ of relations on A to $S^{\mathfrak{A}}(\overline{R}) = (S_1(\overline{R}), \ldots, S_m(\overline{R}))$ where $S_i(\overline{R}) := \{\overline{a} : (\mathfrak{A}, \overline{R}) \models \psi_i(\overline{R}, \overline{a})\}$. As the operator is monotone, it has a least fixed point $\text{lfp}(S^{\mathfrak{A}}) = (R_1^\infty, \ldots, R_m^\infty)$. Now $\mathfrak{A} \models [\text{lfp } R_i : S](\overline{a})$ if $\overline{a} \in R_i^\infty$. Similarly for greatest fixed points.

As in the case of LFP, one can also extend IFP and PFP (defined in the next section) by simultaneous inductions over several formulae. In all of these cases, simultaneous fixed-point logics S-LFP, S-IFP and S-PFP are not more expressive than their simple variants. This can be proven easily by taking a fixed-point over a relation R with bigger arity, e.g. one higher than the maximum arity of $R_1, \ldots, R_m$. The atoms $R_i(\overline{x})$ can then be replaced by $R(c_i, \overline{x})$ for chosen m constants $c_1, \ldots, c_m$. The fixed-point of R is then sufficient to describe the simultaneous fixed-point of S, yielding the following.

Theorem 5.19. For every formula $\varphi \in$ S-LFP ($\varphi \in$ S-IFP,S-PFP) there exists an equivalent formula $\varphi \in$ LFP ($\varphi \in$ IFP,PFP).

5.4 Partial Fixed-Point Logic

Another fixed-point logic that is relevant to finite structures is the partial fixed-point logic (PFP). Let $\psi(R, \bar{x})$ be an arbitrary formula defining on a finite structure $\mathfrak{A}$ a (not necessarily monotone) operator $F_\psi : R \mapsto \{\bar{a} : \mathfrak{A} \models \psi(R, \bar{a})\}$, and consider the sequence of its finite stages $R^0 := \emptyset$, $R^{m+1} = F_\psi(R^m)$.

This sequence is not necessarily increasing. Nevertheless, as $\mathfrak{A}$ is finite, the sequence either converges to a fixed point, or reaches a cycle with a period greater than one. We define the *partial fixed point* of F_ψ as the fixed point that is reached in the former case, and as the empty relation otherwise. The logic PFP is obtained by adding to first-order logic the *partial-fixed-point formation rule*, which allows us to build from any formula $\psi(R, \bar{x})$ a formula $[\mathrm{pfp}\ R\bar{x}\ .\ \psi(R, \bar{x})](\bar{t})$, saying that $\bar{t}$ is contained in the partial fixed point of the operator F_ψ.

Note that if R occurs only positively in ψ, then

$$[\mathrm{lfp}\ R\bar{x}\ .\ \psi(R, \bar{x})](\bar{t}) \equiv [\mathrm{pfp}\ R\bar{x}\ .\ \psi(R, \bar{x})](\bar{t}),$$

so we have that LFP $\leq$ PFP. However, PFP seems to be much more powerful than LFP. For instance, while a least-fixed-point induction on finite structures always reaches the fixed point in a polynomial number of iterations, a partial-fixed-point induction may need an exponential number of stages.

Example 5.20. Consider the sequence of stages R^m defined by the formula

$$\psi(R, x) := \Big(Rx \wedge \exists y(y < x \wedge \neg Ry)\Big) \vee \Big(\neg Rx \wedge \forall y(y < x \to Ry)\Big) \vee \forall y Ry$$

on a finite linear order $(A, <)$. It is easily seen than the fixed point reached by this induction is the set $R = A$, but before this fixed point is reached, the induction goes in lexicographic order through all possible subsets of A. Hence the fixed point is reached at stage $2^n - 1$, where $n = |A|$.

Complexity. Although a PFP induction on a finite structure may go through exponentially many stages (with respect to the cardinality of the structure), each stage can be represented with polynomial storage space. As first-order formulae can be evaluated efficiently, it follows by a simple induction that PFP-formulae can be evaluated in polynomial space.

Proposition 5.21. For every formula $\psi \in$ PFP, the set of finite models of ψ is in PSPACE; in short: PFP $\subseteq$ PSPACE.

On ordered structures, one can use techniques similar to those used in previous capturing results, to simulate polynomial-space-bounded computation by PFP-formulae.

Theorem 5.22 (Abiteboul, Vianu, and Vardi)**.** On ordered finite structures, PFP captures PSPACE.

Proof. It remains to prove that every class $\mathcal{K}$ of finite ordered structures that is recognizable in PSPACE, can be defined by a PFP-formula.

Let M be a polynomially space-bounded deterministic Turing machine with state set Q and alphabet Σ, recognizing (an encoding of) an ordered structure $(\mathfrak{A}, <)$ if and only if $(\mathfrak{A}, <) \in \mathcal{K}$. Without loss of generality, we can make the following assumptions. For input structures of cardinality n, M requires space less than $n^k - 2$, for some fixed k. For any configuration C of M, let Next(C) denote its successor configuration. The transition function of M is adjusted so that Next(C) = C if, and only if, C is an accepting configuration.

We represent any configuration of M with a current state q, tape inscription $w_1 \cdots w_m$, and head position i, by the word $\#w_1 \cdots w_{i-1}(qw_i)w_{i+1} \cdots w_{m-1}\#$ over the alphabet $\Gamma := \Sigma \cup (Q \times \Sigma) \cup \{\#\}$, where $m = n^k$ and # is merely used as an end marker to make the following description more uniform. When moving from one configuration to the next, Turing machines make only local changes. We can therefore associate with M a function $f : \Gamma^3 \to \Gamma$ such that, for any configuration $C = c_0 \cdots c_m$, the successor configuration Next(C) = $c'_0 \cdots c'_m$ is determined by the rules

$$c'_0 = c'_m = \# \quad \text{and} \quad c'_i = f(c_{i-1}, c_i, c_{i+1}) \text{ for } 1 \leq i \leq m-1.$$

Recall that we encode structures so that there exist first-order formulae $\beta_\sigma(\overline{y})$ such that $(\mathfrak{A}, <) \models \beta_\sigma(\overline{a})$ if and only the $\overline{a}$th symbol of the input configuration of M for input code$(()\mathfrak{A}, <)$ is σ. We now represent any configuration C in the computation of M by a tuple $\overline{C} = (C_\sigma)_{\sigma\in\Gamma}$ of k-ary relations, where

$$C_\sigma := \{\overline{a} : \text{ the } \overline{a}\text{-th symbol of } C \text{ is } \sigma\}.$$

The configuration at time t is the stage $t+1$ of a simultaneous pfp induction on $(\mathfrak{A}, <)$, defined by the rules

$$C_\# \overline{y} := \forall z(\overline{y} \leq \overline{z}) \vee \forall \overline{z}(\overline{z} \leq \overline{y})$$

and, for all $\sigma \in \Gamma - \{\#\}$,

$$C_\sigma \overline{y} := \Big(\beta_\sigma(\overline{y}) \wedge \bigwedge_{\gamma\in\Gamma} \forall \overline{x} \neg C_\gamma \overline{x}\Big) \vee \exists \overline{x} \exists \overline{z}\Big(\overline{x} + 1 = \overline{y} \wedge \overline{y} + 1 = \overline{z} \wedge \bigvee_{f(\alpha,\beta,\gamma)=\sigma} C_\alpha \overline{x} \wedge C_\beta \overline{y} \wedge C_\gamma \overline{z})\Big)$$

The first rule just says that each stage represents a word starting and ending with #. The other rules ensure that (1) if the given sequence $\overline{C}$ contains only empty relations (i.e. if we are at stage 0), then the next stage represents the input configuration, and (2) if the given sequence represents a configuration, then the following stage represents its successor configuration.

By our convention, M accepts its input if and only the sequence of configurations becomes stationary (i.e. reaches a fixed point). Hence M accepts code$(()\mathfrak{A}, <)$ if and only if the relations defined by the simultaneous pfp induction on $\mathfrak{A}$ of the rules described above are non-empty. Hence $\mathcal{K}$ is PFP-definable. Q.E.D.

5.4.1 Least Versus Partial Fixed-Point Logic

From the capturing results for PTIME and PSPACE we immediately obtain the result that PTIME = PSPACE if, and only if, LFP = PFP on

ordered finite structures. The natural question arises of whether LFP and PFP can be separated on the domain of all finite structures. For a number of logics, separation results on arbitrary finite structures can be established by relatively simple methods, even if the corresponding separation on ordered structures would solve a major open problem in complexity theory. For instance, we have proved by quite a simple argument that DTC $\subsetneq$ TC, and it is also not very difficult to show that TC $\subsetneq$ LFP (indeed, TC is contained in stratified Datalog, which is also strictly contained in LFP). Further, it is trivial that LFP is less expressive than Σ^1_1 on all finite structures. However the situation is different for LFP vs. PFP.

Theorem 5.23 (Abiteboul and Vianu)**.** LFP and PFP are equivalent on finite structures if, and only if, PTIME = PSPACE.

5.5 Capturing PTIME up to Bisimulation

In mathematics, we consider isomorphic structures as identical. Indeed, it almost goes without saying that relevant mathematical notions do not distinguish between isomorphic objects. As classical algorithmic devices work on ordered *representations of structures* rather than the structures themselves, our capturing results rely on an ability to reason about canonical ordered representations of isomorphism classes of finite structures.

However, in many application domains of logic, structures are distinguished only up to equivalences coarser than isomorphism. Perhaps the best-known example is the modelling of the computational behaviour of (concurrent) programs by transition systems. The meaning of a program is usually not captured by a unique transition system. Rather, transition systems are distinguished only up to appropriate notions of behavioural equivalence, the most important of these being *bisimulation*.

In such a context, the idea of a logic capturing PTIME gets a new twist. One would like to express in a logic precisely those properties of structures that are

(1) decidable in polynomial time, and

(2) invariant under the notion of equivalence being studied.

A class S of rooted transition systems or Kripke structures is *invariant under bisimulation* if, whenever $\mathcal{K}, v \in S$ and $\mathcal{K}, v \sim \mathcal{K}', v'$, then also $\mathcal{K}', v' \in S$. We say that a class S of finite rooted transition systems is in *bisimulation-invariant PTIME* if it is invariant under bisimulation, and if there exists a polynomial-time algorithm deciding whether a given pair $\mathcal{K}, v$ belongs to S. A logic L is invariant under bisimulation if all L-definable properties of rooted transition systems are.

Clearly, $L_\mu \subseteq$ bisimulation-invariant PTIME. However, as pointed out in Section 5.1, L_μ is far too weak to *capture* this class, mainly because it is essentially a monadic logic. Instead, we have to consider a *multidimensional* variant L_μ^ω of L_μ.

But before we define this logic, we should explain the main technical step, which relies on definable canonization, but of course with respect to bisimulation rather than isomorphism. For simplicity of notation, we consider only Kripke structures with a single transition relation E. The extension to the case of several transition relations E_a is straightforward.

With a rooted Kripke structure $\mathcal{K} = (V, E, (P_b)_{b \in B}), u$, we associate a new transition system

$$\mathcal{K}_u^\sim := (V_u^\sim, E^\sim, (P_b^\sim)_{b \in B}),$$

where $V_u^\sim$ is the set of all $\sim$-equivalence classes $[v]$ of nodes $v \in V$ that are reachable from u. More formally, let $[v]$ denote the bisimulation equivalence class of a node $v \in V$. Then

$$\begin{aligned} V_u^\sim &:= \{[v] : \text{there is a path in } G \text{ from } u \text{ to } v\} \\ P_b^\sim &:= \{[v] \in V_u^\sim : v \in P_b\} \\ E^\sim &:= \{([v], [w]) : (v, w) \in E\}. \end{aligned}$$

The pair $\mathcal{K}_u^\sim, [u]$ is, up to isomorphism, a *canonical representant* of the bisimulation equivalence class of $\mathcal{K}, u$. To see this one can prove

that (1) $(\mathcal{K}, u) \sim (\mathcal{K}_u^{\sim}, [u])$, and (2) if $(\mathcal{K}, u) \sim (\mathcal{G}, v)$, then $(\mathcal{K}_u^{\sim}, [u]) \cong (\mathcal{G}_v^{\sim}, [v])$.

It follows that a class S of rooted transition systems is bisimulation-invariant if and only if $S = \{(\mathcal{K}, u) : (\mathcal{K}_u^{\sim}, [u]) \in S\}$. Let $\mathcal{CR}^{\sim}$ be the domain of canonical representants of finite transition systems, i.e.

$$\mathcal{CR}^{\sim} := \{\mathcal{K}, u \mid (\mathcal{K}_u^{\sim}, [u]) \cong (\mathcal{K}, u)\}.$$

Proposition 5.24. $\mathcal{CR}^{\sim}$ admits LFP-definable linear orderings, i.e. for every vocabulary $\tau = \{E\} \cup \{P_b : b \in B\}$, there exists a formula $\psi(x, y) \in \mathrm{LFP}(\tau)$ which defines a linear order on every transition system in $\mathcal{CR}^{\sim}(\tau)$.

Proof. Recall that bisimulation equivalence on a transition system is a greatest fixed point. Its complement, bisimulation inequivalence, is a least fixed point, which is the limit of an increasing sequence $\not\sim_i$ defined as follows: $u \not\sim_0 v$ if u and v do not have the same atomic type, i.e. if there exists some b such that one of the nodes u, v has the property P_b and the other does not. Further, $u \not\sim_{i+1} v$ if the sets of $\sim_i$-classes that are reachable in one step from u and v are different. The idea is to refine this inductive process, by defining relations $\prec_i$ that order the $\sim_i$-classes. On the transition system itself, these relations are pre-orders. The inductive limit $\prec$ of the pre-orders $\prec_i$ defines a linear order of the bisimulation equivalence classes. But in transition systems in $\mathcal{CR}^{\sim}$, bisimulation classes have only one element, so $\prec$ actually defines a linear order on the set of nodes.

To make this precise, we choose an order on B and define $\prec_0$ by enumerating the $2^{|B|}$ atomic types with respect to the propositions P_b, i.e.

$$x \prec_0 y := \bigvee_{b \in B} \Big(\neg P_b x \wedge P_b y \wedge \bigwedge_{b' < b} P_{b'} x \leftrightarrow P_{b'} y \Big).$$

In other words, there is some b such that P_b separates x from y and for the least such b, P_b holds on y and not on x.

In what follows, $x \sim_i y$ can formally be taken as an abbreviation for $\neg(x \prec_i y \vee y \prec_i x)$, and similarly for $x \sim y$. We define $x \prec_{i+1} y$ by the condition that either $x \prec_i y$, or $x \sim_i y$ and the set of $\sim_i$-classes reachable from x is lexicographically smaller than the set of $\sim_i$-classes reachable from y. Note that this inductive definition of $\prec$ is not monotone, so it cannot be directly captured by an LFP-formula. However, as we know that LFP $\equiv$ IFP, we can use an IFP-formula instead. Explicitly, $\prec$ is defined by $[\mathrm{ifp}\, x \prec y \,.\, \psi(\prec, x, y)](x, y)$, where

$$\psi(\prec, x, y) := x \prec_0 y \vee \Big(x \sim y \wedge (\exists y' \,.\, Eyy')\Big((\forall x' \,.\, Exx')x' \not\sim y' \wedge (\forall z.z \prec y')\big(\exists x''(Exx'' \wedge x'' \sim z) \leftrightarrow \exists y''(Eyy'' \wedge y'' \sim z)\big)\Big)\Big).$$

Q.E.D.

Corollary 5.25. On the domain $\mathcal{CR}^{\sim}$, LFP captures PTIME.

Since LFP is not invariant under bisimulation, we will strengthen the above result and capture bisimulation-invariant PTIME in terms of a natural logic, the multidimensional μ-calculus L_μ^ω.

Definition 5.26. The syntax of the *k-dimensional μ-calculus* L_μ^k (for transition systems $\mathcal{K} = (V, E, (P_b)_{b\in B})$) is the same as the syntax of the usual μ-calculus L_μ with modal operators $\langle i\rangle$, $[i]$, and $\langle\sigma\rangle$, $[\sigma]$ for every substitution $\sigma : \{1, \ldots, k\} \to \{1, \ldots, k\}$. Let $S(k)$ be the set of all these substitutions.

The semantics is different, however. A formula ψ of L_μ^k is interpreted on a transition system $\mathcal{K} = (V, E, (P_b)_{b\in B})$ at node v by evaluating it as a formula of L_μ on the modified transition system

$$\mathcal{K}^k = (V^k, (E_i)_{1\le i\le k}, (E_\sigma)_{\sigma\in S(k)}, (P_{b,i})_{b\in B, 1\le i\le k})$$

at node $\underline{v} := (v, v, \ldots, v)$. Here $V^k = V \times \cdots \times V$ and

$$E_i := \{(\overline{v}, \overline{w}) \in V^k \times V^k : (v_i, w_i) \in E \text{ and } v_j = w_j \text{ for } j \neq i\}$$
$$E_\sigma := \{(\overline{v}, \overline{w}) \in V^k \times V^k : w_i = v_{\sigma(i)} \text{ for all } i\}$$
$$P_{b,i} := \{\overline{v} \in V^k : v_i \in P_b\}$$

That is, $\mathcal{K}, v \models_{L_\mu^k} \psi$ iff $\mathcal{K}^k, (v, \ldots, v) \models_{L_\mu} \psi$. The *multidimensional μ-calculus* is $L_\mu^\omega = \bigcup_{k<\omega} L_\mu^k$.

Remark. Instead of evaluating a formula $\psi \in L_\mu^k$ at single nodes v of G, we can also evaluate it at k-tuples of nodes: $\mathcal{K}, \overline{v} \models_{L_\mu^k} \psi$ iff $\mathcal{K}^k, \overline{v} \models_{L_\mu} \psi$.

Example 5.27. Bisimulation is definable in L_μ^2 (in the sense of the remark just made). Let

$$\psi^\sim := \nu X \,.\, \big(\bigwedge_{b \in B} (P_{b,1} \leftrightarrow P_{b,2}) \wedge [1]\langle 2\rangle X \wedge [2]\langle 1\rangle X \big).$$

For every transition system $\mathcal{K}$, we have that $\mathcal{K}, v_1, v_2 \models \psi^\sim$ if, and only if, v_1 and v_2 are bisimilar in $\mathcal{K}$. Further, we have that

$$\mathcal{K}, v \models \mu Y \,.\, \langle 2\rangle(\psi^\sim \vee \langle 2\rangle Y)$$

if, and only if, there exists in $\mathcal{K}$ a point w that is reachable from v (by a path of length ≥ 1) and bisimilar to v.

One can see that L_μ^ω is invariant under bisimulation (because if $\mathcal{K}, v_i \sim \mathcal{G}, u_i$ for all i then also $\mathcal{K}^k, \overline{v} \sim \mathcal{G}, \overline{u}$) and that L_μ^ω can be embedded in LFP. This establishes the easy direction of the desired result: $L_\mu^\omega \subseteq$ bisimulation-invariant PTIME.

For the converse, it suffices to show that LFP and L_μ^ω are equivalent on the domain $\mathcal{CR}^\sim$. Let S be a class of rooted transition systems in bisimulation-invariant PTIME. For any $\mathcal{K}, u$, we have that $\mathcal{K}, u \in S$ if its canonical representant $\mathcal{K}_u^\sim, [u] \in S$. If LFP and L_μ^ω are equivalent on $\mathcal{CR}^\sim$, then there exists a formula $\psi \in L_\mu^\omega$ such that $\mathcal{K}_u^\sim, [u] \models \psi$ iff $\mathcal{K}_u^\sim, [u] \in S$. By the bisimulation invariance of ψ, it follows that $\mathcal{K}, u \models \psi$ iff $\mathcal{K}, u \in S$.

The *width* of an LFP-formula φ is the maximal number of free variables occuring in a subformula of φ.

Proposition 5.28. On the domain $\mathcal{CR}^{\sim}$, LFP $\leq L_\mu^\omega$. More precisely, for each formula $\psi(x_1, \ldots, x_k) \in$ LFP of width $\leq k$, there exists a formula $\psi^* \in L_\mu^{k+1}$ such that for each $\mathcal{K}, u \in \mathcal{CR}^{\sim}$, we have that $\mathcal{K} \models \psi(u, \overline{v})$ iff $\mathcal{K}, u, \overline{v} \models \psi^*$.

Note that although, ultimately, we are interested only in formulae $\psi(x)$ with just one free variable, we need more general formulae, and evaluation of L_μ^k-formulae over k-tuples of nodes, for the inductive treatment. In all formulae, we shall have at least x_1 as a free variable, and we always interpret x_1 as u (the root of the transition system). We remark that, by an obvious modification of the formula given in Example 5.27, we can express in L_μ^k the assertion that $x_i \sim x_j$ for any i, j.

Atomic formulae are translated from LFP to L_μ^ω according to

$$\begin{aligned}(x_i = x_j)^* &:= x_i \sim x_j\\ (P_b x_i)^* &:= P_{b,i}\overline{x}\\ (Ex_ix_j)^* &:= \langle i\rangle x_i \sim x_j\\ (Xx_{\sigma(1)} \cdots x_{\sigma(r)})^* &:= \langle\sigma\rangle X.\end{aligned}$$

Boolean connectives are treated in the obvious way, and *quantifiers* are translated by use of fixed points. To find a witness x_j satisfying a formula ψ, we start at u (i.e. set $x_j = x_1$), and search along transitions (i.e. use the μ-expression for reachability). That is, let $j/1$ be the substitution that maps j to 1 and fixes the other indices, and translate $\exists x_j \psi(\overline{x})$ into

$$\langle j/1\rangle \mu Y \,.\, \psi^* \vee \langle j\rangle Y.$$

Finally, *fixed points* are first brought into normal form so that variables appear in the right order, and then they are translated literally, i.e. $[\mathrm{lfp}\, X\overline{x} \,.\, \psi](\overline{x})$ translates into $\mu X \,.\, \psi^*$.

The proof that the translation has the desired property is a straight-

forward induction, which we leave as an exercise. Altogether we have established the following result.

Theorem 5.29 (Otto)**.** The multidimensional μ-calculus captures bisimulation-invariant PTIME.

Otto has also established capturing results with respect to other equivalences. For finite structures $\mathfrak{A}, \mathfrak{B}$, we say that $\mathfrak{A} \equiv_k \mathfrak{B}$ if no first-order sentence of width k can distinguish between $\mathfrak{A}$ and $\mathfrak{B}$. Similarly, $\mathfrak{A} \equiv_k^C \mathfrak{B}$ if $\mathfrak{A}$ and $\mathfrak{B}$ are indistinguishable by first-order sentences of width k with counting quantifiers of the form $\exists^{\geq i} x$, for any $i \in \mathbb{N}$.

Theorem 5.30 (Otto)**.** There exist logics that effectively capture $\equiv_2$-invariant PTIME and $\equiv_2^C$-invariant PTIME on the class of all finite structures.

6 Fixed-point logic with counting

The (machine-independent) characterisation of complexity classes by logics (in the sense of Definition 3.4) yields deep insights into the structure of the classified problems. The theorem of Fagin (cf. Chapter 3) is a seminal result in the field of descriptive complexity theory, and gives such a correspondence between algorithmic and logical resources for the important class NP. If we restrict to ordered structures, we can also find such characterisation for PTIME as shown e.g. in the Immerman-Vardi theorem (cf. Chapter 4). However, it is still one of the major open questions in finite model theory whether there is a logic capturing PTIME on *all* finite structures. Note that if no such logic exists this would necessarily imply PTIME $\neq \exists\mathrm{SO} = \mathrm{NP}$.

As we will see, fixed-point logics, such as LFP or IFP, do not suffice to capture PTIME on arbitrary structures, and most of the naturally considered examples to separate them from PTIME involve some kind of counting. For instance, the simple class EVEN $= \{\mathfrak{A} : |A| \text{ is even}\}$ turns out to be not definable in LFP. Therefore Immerman proposed that counting quantifiers should be added to logics and asked whether a suitable variant of fixed-point logic with counting would suffice to capture PTIME.

Although Cai, Fürer and Immerman eventually answered this question negatively, the extension of fixed-point logic by counting terms (FPC) has turned out to be an important and robust logic, that defines a natural level of expressiveness. In this chapter we study the logic FPC and present the construction of Cai, Fürer and Immerman which yields the separation of FPC from PTIME. To be precise, we even present a slightly more general result which uses the concept of treewidth and which is due to Dawar and Richerby.

6.1 Logics with Counting Terms

There are different ways of adding counting mechanisms to a logic, which are not necessarily equivalent. The most straightforward possibility is the addition of quantifiers of the form $\exists^{\geq 2}$, $\exists^{\geq 3}$, etc., with the obvious meaning. While this is perfectly reasonable for bounded-variable fragments of first-order logic or infinitary logic it does not increase the expressiveness of logics such as FO or LFP, since they are closed under the replacement of $\exists^{\geq i}$ by i existential quantifiers. For fixed-point logic another severe restriction is that it does not allow for recursion over the counting parameters i in quantifiers $\exists^{\geq i} x$. These counting parameters should therefore be considered as variables that range over natural numbers. To define in a precise way a logic with counting and recursion, one extends the original objects of study, namely finite (one-sorted) structures $\mathfrak{A}$, to two-sorted auxiliary structures $\mathfrak{A}^*$ with a second numerical (but also finite) sort.

Definition 6.1. With any one-sorted finite structure $\mathfrak{A}$ with universe A, we associate the two-sorted structure $\mathfrak{A}^* := \mathfrak{A} \dot{\cup} \langle \{0, \ldots, |A|\}; \leq, 0, e \rangle$, where $\leq$ is the canonical ordering on $\{0, \ldots, |A|\}$, and 0 and e stand for the first and the last element. Thus, $\mathfrak{A}^*$ is the disjoint union of $\mathfrak{A}$ with a linear order of length $|A| + 1$.

For all logics we studied so far, we naturally obtain two-sorted variants definining properties of the extended structures $\mathfrak{A}^*$. For instance, formulas of two-sorted first-order logic over two-sorted vocabularies $\sigma \cup \{\leq, 0, e\}$ are evaluated in structures $\mathfrak{A}^*$ where semantics are defined in the obvious way. From now on, we stick to the convention to use Latin letters $x, y, z, \ldots$ for the variables over the first sort, and Greek letters $\lambda, \mu, \nu, \ldots$ for variables over the second sort (the numerical sort). In counting logics, these two sorts are related by *counting terms*, defined by the following rule. Let $\varphi(x)$ be a formula with a variable x (over the first sort) among its free variables. Then $\#_x[\varphi]$ is a term in the second sort, with the set of free variables $\text{free}(\#_x[\varphi]) = \text{free}(\varphi) - \{x\}$. The value of $\#_x[\varphi]$ is the number of elements a that satisfy $\varphi(a)$.

We introduce counting logics starting with first-order logic with

counting, denoted by FOC, which is the closure of two-sorted first-order logic under counting terms. Here are two simple examples that illustrate the use of counting terms.

Example 6.2. On an undirected graph $G = (V, E)$, the formula $\forall x \forall y(\#_z[Exz] = \#_z[Eyz])$ expresses the assertion that every node has the same degree, i.e., that G is regular.

Example 6.3. We present below a formula $\psi(E_1, E_2) \in$ FOC which expresses the assertion that two equivalence relations E_1 and E_2 are isomorphic; of course a necessary and sufficient condition for this is that for every i, they have the same number of elements in equivalence classes of size i:

$$\psi(E_1, E_2) \equiv (\forall \mu)(\#_x[\#_y[E_1 xy] = \mu] = \#_x[\#_y[E_2 xy] = \mu]).$$

6.2 Fixed-Point Logic with Counting

We now define *(inflationary) fixed point logic with counting* (FPC) and *partial fixed point logic with counting* PFPC by adding to FOC the usual rules for building inflationary or partial fixed points, ranging over both sorts.

Definition 6.4. Inflationary fixed point logic with counting, FPC, is the closure of two-sorted first-order logic under the following rules:

(1) The rule for building counting terms.

(2) The usual rules of first-order logic for building terms and formulae.

(3) The fixed-point formation rule. Suppose that $\psi(R, \overline{x}, \overline{\mu})$ is a formula of vocabulary $\tau \cup \{R\}$ where $\overline{x} = x_1, \ldots, x_k$, $\overline{\mu} = \mu_1, \ldots, \mu_\ell$, and R has mixed arity (k, ℓ), and that $(\overline{u}, \overline{v})$ is a $k + \ell$-tuple of first- and second-sort terms, respectively. Then

$$[\text{ifp } R\overline{x}\overline{\mu} \,.\, \psi](\overline{u}, \overline{v})$$

is a formula of vocabulary τ.

The semantics of $[\text{ifp } R\overline{x}\overline{\mu} \,.\, \psi]$ on $\mathfrak{A}^*$ is defined in the same way as

for the logic IFP, namely as the inflationary fixed point of the operator

$$F_\psi : R \longmapsto R \cup \{(\bar{a}, \bar{i}) \mid (\mathfrak{A}^*, R) \models \psi(\bar{a}, \bar{i})\}.$$

The definition of PFPC is analogous, where we replace inflationary fixed points by partial ones. In the literature, one also finds different variants of fixed-point logic with counting where the two sorts are related by *c*ounting quantifiers rather than counting terms. Counting quantifiers have the form $(\exists i\, x)$ for 'there exist at least i different x', where i is a second-sort variable. It is obvious that the two definitions are equivalent. In fact, FPC is a very robust logic. For instance, its expressive power does not change if one permits counting over tuples, even of mixed type, i.e. terms of the form $\#_{\bar{x},\bar{\mu}}\varphi$ (see exercise class). One can of course also define least fixed-point logic with counting, LFPC, but one has to be careful with the positivity requirement (which is more natural when one uses counting quantifiers rather than counting terms). The equivalence of LFP and IFP readily translates to LFPC $\equiv$ IFPC.

Example 6.5. An interesting example of an FPC-definable query is the method of *s*table colourings for graph-canonization. Given a graph G with a colouring $f : V \to \{0, \dots, r\}$ of its vertices, we define a refinement f' of f, giving to a vertex x the new colour $f'x = (fx, n_1, \dots, n_r)$ where $n_i = \#y[Exy \wedge (fy = i)]$. The new colours can be sorted lexicographically so that they again form an initial subset of $\mathbb{N}$. Then the process can be iterated until a fixed point, the *s*table colouring of G is reached. It is easy to see that the stable colouring of a graph is polynomial-time computable and uniformly definable in FPC.

On many graphs, the stable colouring uniquely identifies each vertex, i.e. no two distinct vertices (i.e. vertices in different orbits of the automorphism group) get the same stable colour. In this way stable colourings provide a polynomial-time graph canonization algorithm for such classes of graphs. For instance, this is the case for the class of all trees or, more generally, any class of graphs with bounded treewidth.

We now discuss the expressive power and evaluation complexity of fixed-point logic with counting. We are mainly interested in FPC-formulae and PFPC-formulae without free variables over the sec-

ond sort, so that we can compare them with the usual logics without counting.

Exercise 6.1. Even without making use of counting terms, IFP over two-sorted structures $\mathfrak{A}^*$ is more expressive than IFP over $\mathfrak{A}$. To prove this, construct a two-sorted IFP-sentence ψ such that $\mathfrak{A}^* \models \psi$ if, and only if, $|A|$ is even.

It is clear that counting terms can be computed in polynomial-time. Hence the data complexity remains in PTIME for FPC and in PSPACE for PFPC. We shall see below that these inclusions are strict.

Theorem 6.6. On finite structures,

(1) IFP $\subsetneq$ FPC $\subsetneq$ PTIME.
(2) PFP $\subsetneq$ PFPC $\subsetneq$ PSPACE.

6.2.1 Infinitary Logic with Counting

Let $C^k_{\infty\omega}$ be the infinitary logic with k variables $L^k_{\infty\omega}$, extended by the quantifiers $\exists^{\geq m}$ ('there exist at least m') for all $m \in \mathbb{N}$. Further, let $C^\omega_{\infty\omega} := \bigcup_k C^k_{\infty\omega}$.

Proposition 6.7. PFPC $\subseteq C^\omega_{\infty\omega}$.

Due to the two-sorted framework, the proof of this result is a bit more involved than for the corresponding result without counting, but not really difficult (see exercise class).

The separation of FPC from PTIME has been established by Cai, Fürer, and Immerman. Their proof also provides an analysis of the method of stable colourings for graph canonization. We have described this method in its simplest form in Example 6.1. More sophisticated variants compute and refine colourings of k-tuples of vertices. This is called the *k-dimensional Weisfeiler–Lehman method* and, in logical terms, it amounts to labelling each k-tuple by its type in $k+1$-variable logic with counting quantifiers. It was conjectured that this method could provide a polynomial-time algorithm for graph isomorphism, at least for graphs of bounded degree. However, Cai, Fürer, and Immerman were able to construct two families $(G_n)_{n\in\mathbb{N}}$ and $(H_n)_{n\in\mathbb{N}}$ of graphs

such that on one hand, G_n and H_n have $\mathcal{O}(n)$ nodes and degree three, and admit a linear-time canonization algorithm, but on the other hand, in first-order (or infinitary) logic with counting, $\Omega(n)$ variables are necessary to distinguish between G_n and H_n. In particular, this implies Theorem 6.6.

6.3 The k-pebble bijection game

In Chapter 2 we introduced Ehrenfeucht-Fraïssé games to characterize the equivalence of structures (or, to put it in another way, definability of classes) in first-order logic. More specifically, two relational structures $\mathfrak{A}$ and $\mathfrak{B}$ can be distinguished by an FO-sentence of quantifier-rank $\leq m$ if, and only if, Spoiler has a winning strategy in the m-move Ehrenfeucht-Fraïssé game played on $\mathfrak{A}$ and $\mathfrak{B}$ which was denoted by $EF_m(\mathfrak{A}, \mathfrak{B})$.

Our next aim is to introduce the *k-pebble bijection game* which is an extension of the standard Ehrenfeucht-Fraïssé game to capture definability in $C^\omega_{\infty\omega}$. We will use these games to show that a certain (polynomial-time decidable) class of graphs is not definable in $C^\omega_{\infty\omega}$. In particular, this yields the separation of FPC from PTIME by Proposition 6.7.

Definition 6.8. The *k-pebble bijection game* k-BG$(\mathfrak{A}, \mathfrak{B})$ is a two-player game played on relational structures $\mathfrak{A}$ and $\mathfrak{B}$ using k pairs of pebbles $(x_1, y_1), \ldots, (x_n, y_n)$ that can be placed on pairs of elements $(a_1, b_1), \ldots, (a_n, b_n) \in A \times B$ during a play. The goal of Player I, who is called *Spoiler*, is to show that $\mathfrak{A} \not\equiv^{C^k_{\infty\omega}} \mathfrak{B}$ while Player II, the *Duplicator*, claims that $\mathfrak{A} \equiv^{C^k_{\infty\omega}} \mathfrak{B}$.

A *position* in the game k-BG$(\mathfrak{A}, \mathfrak{B})$ is a (partial) assignment $(a_1, b_1), \ldots, (a_n, b_n)$ of pebbles on $A \times B$, so formally, a position is a (partial) mapping $p : \{1, \ldots, k\} \to A \times B$. The initial position is $p = \emptyset$.

At position p a play proceeds as follows: First, Spoiler selects a pair of pebbles $i \leq k$. Duplicator has to react with a bijection $h : A \to B$ which respects all remaining pairs of pebbled elements (except for i), i.e. for all $i \neq j \in \mathrm{dom}(p)$ and $p(j) = (a_j, b_j)$ we have $h(a_j) = b_j$. Spoiler

then chooses $a \in A$ and the position is updated to $(p|i \mapsto (a_i, b_i))$ where

$$(p|i \mapsto (a_i, b_i))(j) := \begin{cases} p(j) & j \neq i \\ (a, h(a)) & j = i. \end{cases}.$$

Spoiler wins a play, if either $|A| \neq |B|$ (i.e. Duplicator cannot respond with a bijection), or the play eventually reaches a position p such that the induced mapping $p(\{1, \ldots, k\})$ is not a partial isomorphism of $\mathfrak{A}$ and $\mathfrak{B}$, i.e. if $p(\{1, \ldots, k\}) \notin \mathrm{Loc}(\mathfrak{A}, \mathfrak{B})$. Infinite plays are won by Duplicator.

Theorem 6.9. If Duplicator wins the game k-BG$(\mathfrak{A}, \mathfrak{B})$, then $\mathfrak{A} \equiv^{C^k_{\infty\omega}} \mathfrak{B}$.

Proof. We prove by induction that for all formulae $\varphi(x_1, \ldots, x_k) \in C^k_{\infty\omega}$, structures $\mathfrak{A}$ and $\mathfrak{B}$ and all $a_1, \ldots, a_k \in A$ and $b_1, \ldots, b_k \in B$ we have that if $\mathfrak{A} \models \varphi(a_1, \ldots, a_k)$ and $\mathfrak{B} \not\models \varphi(a_1, \ldots, a_k)$ then Spoiler has a winning strategy for k-BG$(\mathfrak{A}, \mathfrak{B})$ starting from position $p(i) = (a_i, b_i)$.

The cases of quantifier-free formulae, Boolean connectivities and first-order quantifier follow as in the case of Ehrenfeucht-Fraïssé games (cf. lecture notes of mathematical logic). Hence, we only consider $\varphi = \exists^{\geq i} x_j \psi(x_1, \ldots, x_k)$. For this case, a winning strategy for Spoiler can be defined in the following way:

- Spoiler selects the pair $j \leq k$.
- Duplicator reacts with a bijection $h : A \to B$ respecting the remaining pebbled pairs.

We set $X = \{a \in A : \mathfrak{A} \models \psi(a_1, \ldots, a_n)\}$ and $Y = \{b \in B : \mathfrak{B} \models \psi(b_1, \ldots, b_n)\}$. From the assumption we know that $|X| \geq i$ and $|Y| < i$, hence there is an $a \in X$ such that $h(a) \notin Y$. Spoiler selects the element a and the position is updated to $(p|j \mapsto (a_j, b_j))$. As we have $\mathfrak{A} \models \varphi(a_1, \ldots, a_n)$ and $\mathfrak{B} \not\models \varphi(b_1, \ldots, b_{j-1}, h(a), b_{j+1}, \ldots, b_n)$ the claim follows by induction. Q.E.D.

We can use Theorem 6.9 to show that a class $\mathcal{K}$ of finite structures is not definable in $C^\omega_{\infty\omega}$. In particular, note that $\mathcal{K} \notin C^\omega_{\infty\omega}$ also implies that $\mathcal{K} \notin$ FPC since we have FPC $\leq C^\omega_{\infty\omega}$.

Proposition 6.10. Let $(\mathfrak{A}_k)_{k\geq 1}$ and $(\mathfrak{B}_k)_{k\geq 1}$ be two sequences of structures such that for infinitely many k we have $\mathfrak{A}_k \in \mathcal{K}$, $\mathfrak{B}_k \notin \mathcal{K}$ and Duplicators wins k-BG$(\mathfrak{A}_k, \mathfrak{B}_k)$. Then $\mathcal{K}$ cannot be defined in $C^\omega_{\infty\omega}$.

6.4 The construction of Cai, Fürer and Immerman

We now present the construction of Cai, Fürer and Immmerman which yields the separation of FPC from PTIME. Throughout this section, let $G = (V, E)$ denote a connected graph with $\deg(v) \geq 2$ for all $v \in V$. Starting from G we define a family of graphs $(X_S(G))_{S\subseteq E}$ that result by replacing every vertex v in a G by a gadget $Z(v)$ and interconnecting different gadgets according to edge relation in G.

For every v we define the set of new vertices $Z(v)$ as

$$Z(v) := \{a_{vw}, b_{vw}, c_{vw}, d_{vw} : w \in vE\} \cup \{v^X : X \subseteq vE, |X| \text{ even}\}.$$

Vertices of the form a_{vw}, b_{vw} are called *outer vertices* and they are intended to connect the two gadgets $Z(v)$ and $Z(w)$. The vertices c_{vw}, d_{vw} are *colour vertices* which are used only to make the set of outer nodes first-order definable. The remaining vertices v^S are called the *inner vertices*.

Let $X_\emptyset(G)$ denote the graph over the vertex set $\bigcup_{v\in V} Z(v)$ with the following edges:

- $(a_{vw}, c_{vw}), (b_{vw}, c_{vw}), (d_{vw}, c_{vw})$ for $(v, w) \in E$,
- (a_{vw}, v^X) for $w \in X$,
- (b_{vw}, v^X) for $w \notin X$, and
- (a_{vw}, a_{wv}) and (b_{vw}, b_{wv}) for all $(v, w) \in E$.

In Figure 6.1 the construction of a gadget $Z(v)$ is illustrated for the case of a vertex v with degree three. The pairs of outer nodes a_{vx}, b_{vx}, a_{vy}, b_{vy} and a_{vz}, b_{vz} are connected to the corresponding outer nodes of the gadgets $Z(x)$, $Z(y)$ and $Z(z)$, respectively (this is indicated by the dashed lines in the figure).

We now extend the construction: for any (symmetric) set $S \subseteq E$ we define $X_S(G)$ to be the graph $X_\emptyset(G)$ in which for all $(v, w) \in S$ the edges (a_{vw}, a_{wv}) and (b_{vw}, b_{wv}) are replaced by (a_{vw}, b_{wv}) and (a_{wv}, b_{wv}).

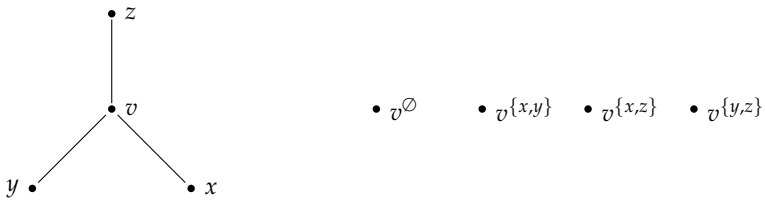


Figure 6.1. Example: gadget for a vertex v of degree three

We say that the edges in S have been *twisted*. In this way we obtain for every subset $S \subseteq E$ of edges a *CFI-graph* $X_S(G)$. Interestingly, we are going to show that these CFI-graphs $X_S(G)$ are completely determined by the parity of the set S:

Lemma 6.11. For all $S, T \subseteq E$ we have:

$$X_S(G) \cong X_T(G) \Leftrightarrow |S| \equiv |T| \mod 2.$$

Before we prove this claim in general, we consider some special cases. First of all, let all twisted edges be incident with a single vertex v.

Lemma 6.12. Let $S, T \subseteq vE$ be sets of neighbours of some vertex $v \in V$. If $S\Delta T = (S \setminus T) \cup (T \setminus S)$ is even, then

$$X_{v\times S}(G) \cong X_{v\times T}(G).$$

Proof. The mapping $\pi_{v;S;T} : X_{v\times S}(G) \to X_{v\times T}(G)$ defined by

$$\pi_{v;S;T}(z) := \begin{cases} z, & z \notin Z(v) \text{ or } z \text{ colour vertex}, \\ z, & z \in \{a_{vw}, b_{vw}\}, (v,w) \in S \cap T, \\ b_{vw}, & z = a_{vw}, (v,w) \in S\Delta T, \\ a_{vw}, & z = b_{vw}, (v,w) \in S\Delta T, \\ v^{X\Delta(S\Delta T)}, & z = v^X, \end{cases}$$

is an isomorphism (use that since X and $S\Delta T$ are even, the same holds for the symmetric difference $X\Delta(S\Delta T)$). Q.E.D.

We proceed to explain how one obtains an isomorphism between $X_{\{e\}}(G)$ and $X_{\{f\}}(G)$ for two distinct edges e and f of G.

Lemma 6.13. $X_{\{e\}}(G) \cong X_{\{f\}}(G)$.

Proof. If e and f are incident with the same vertex v, then the claim follows by Lemma 6.12. Hence, let $e = (u, v)$ and $f = (x, y)$ be such that $\{u, v\} \cap \{x, y\} = \emptyset$. Choose a path $v = v_1, v_2, \ldots, v_\ell = x$ connecting v and x with $v_i \notin \{u, y\}$ for all $i \geq 1$. Then

$$\pi_{e \mapsto f} := \pi_{v_1;u;v_2} \circ \pi_{v_2;v_1;v_3} \circ \cdots \circ \pi_{v_{l-1};v_{l-2};x} \circ \pi_{v_l;v_{l-1};y},$$

is an isomorphism of $X_{\{e\}}(G) \cong X_{\{f\}}(G)$: the twist at edge (u, v) is moved along the path to the twist at edge (x, y) where both twists cancel out each other. Note than along the path, at every inner node v_i we have precisely two twists of edges for the gadget $Z(v_i)$ which, again by Lemma 6.12, preserves the structure of the inner nodes. Q.E.D.

We are now ready to prove Lemma 6.11.

Proof (of Lemma 6.11). First of all, let $|S| \equiv |T| \mod 2$. If $|S| = |T| = 1$, then the claim follows by Lemma 6.13, so assume that $|S| \geq 2$ (or analogously, $|T| \geq 2$). Choose $e, f \in S$ with $e \neq f$. If e and f are incident with the same vertex $v \in V$ we know that $X_{S \setminus \{e,f\}}(G) \cong X_S(G)$ by Lemma 6.13. In the other case, we use the isomorphism $\pi_{e \mapsto f}$ and see that $X_{S \setminus \{e,f\}}(G) \cong X_S(G)$. The claim follows by induction on $|S \Delta T|$.

For the other direction assume that $\pi : X_{\{f=(x,y)\}}(G) \to X_\emptyset(G)$ is an isomorphism. Clearly, π maps outer (inner, colour) nodes to outer (inner, coulour) nodes, and since π also induces an isomorphism of G, we can assume that for all $v \in V$ we have $\pi(Z(v)) = Z(v)$ and $\pi(\{a_{vw}, b_{vw}\}) = \{a_{vw}, b_{vw}\}$ for all $(v, w) \in E$. At this point we observe that if π interchanges a_{vw} and b_{vw} it necessarily interchanges a_{wv} and b_{wv} for all edges $(v, w) \in E$ except for (x, y). Hence, the total number of interchanges of a's and b's in π is odd. This contradicts, Lemma 6.12, however, as the number of interchanges of a's and b's in π for each gadget has to be even. Q.E.D.

We conclude that, up to isomorphism, there are precisely two CFI-graphs for G and we fix two canonical representatives from the isomorphism classes:

- $X(G) := X_{\emptyset}(G)$ (the *even CFI-graph for G*)
- $\tilde{X}(G) := X_{\{e\}}(G)$ for some edge $e \in E$ (the *odd CFI-graph for G*)

The *CFI-query* is to decide, given a CFI-graph $X_S(G)$, whether $X_S(G)$ is even or odd, i.e. whether $X_S(G) \cong X(G)$ or $X_S(G) \cong \tilde{X}(G)$.

Theorem 6.14. The CFI-query can be decided in polynomial time.

Proof. In order to count the number of twists, we need to identify the a and b-vertices. To this end it suffices to fix in every gadget $Z(v)$ an arbitrary inner node and to associate the intended labeling to the gadget $Z(v)$ (e.g. declare this node to be $v^{\emptyset}$ and assign to all connected vertices b-labels and to the remaining outer ones a-labels). Then it is straightforward to count the number of twists modulo two. Lemma 6.11 guarantees that the isomorphism class of the resulting $\{a, b\}$-labeled graph is independent of the initial choice of inner vertices. Q.E.D.

We conclude that the even and odd CFI-graphs can be distinguished in polynomial time. However, we are going to show that they cannot be separated by sentences in $C^{\omega}_{\infty\omega}$ if we start from a class of graphs G with sufficient complexity. In order to measure the complexity of graphs we introduce the important and well-studied concept of *treewidth*. Intuitively the treewidth of a graph formalises to what extent an (undirected) graph resembles a tree, and one of the reasons for its importance is that many NP-hard problems (and even some PSPACE-hard ones) become tractable on classes of graphs with bounded treewidth. There are various equivalent ways to characterize the treewidth of a graph, of which we sketch two: an algebraic and a game theoretic approach.

Definition 6.15. Let $G = (V, E)$ be an undirected graph. A *tree decomposition of G* is an undirected tree $\mathcal{T} = (T, E_T)$ where T is a family of subsets of V, i.e. $T \subseteq \mathcal{P}(V)$ and

(a) $\bigcup T = V$, and

(b) for all $(u, v) \in E$ there is some $X \in T$ so that $\{u, v\} \subseteq X$, and

(c) for every vertex $v \in V$ the set $\{X \in T : v \in X\}$ is connected in $\mathcal{T}$.

Nodes in the tree $\mathcal{T}$ are called *bags*. The *width* of the tree decomposition $\mathcal{T} = (T, E_T)$ is $(\max\{|X| : X \in T\} - 1)$, and the *treewidth of* G, denoted by $\mathrm{tw}(G)$, is defined to be the minimal width for which a tree decomposition of G exists.

Next, we describe a game which characterises the notion of treewidth. The *k-cops and robber game* on G is played by two players, Player I (the cops) and Player II (the robber). The rules are as follows: the cops possess k pebbles (cops) which they can place on vertices of the graph. The robber has one pebble which is moved along paths. In each move the cops first choose some pebble which is either currently not placed on a vertex of the graph or which is removed from its current position w. Secondly, the cops determine a vertex v to be the new position for this pebble. After that, the robber reacts by moving his pebble along a path to a new vertex (which may be the old one). The chosen path has to be cop-free where the vertices v and w count as cop-free for the current move. The cops win a play if, and only if, they can reach a position such that the robber cannot move. All other plays, i.e. all infinite ones, are won by the robber.

Seymour proved that a graph G has treewidth k if, and only if, the cops have a winning strategy in the game with $k + 1$ pebbles, but the robber wins the game if the cops are restricted to k pebbles. We use this game-theoretic characterisation of to show:

Theorem 6.16. Let $G = (V, E)$ be graph with $\delta(G) \geq 2$ and $\mathrm{tw}(G) \geq k$. Then

$$X(G) \equiv^{C^k_{\infty\omega}} \tilde{X}(G).$$

Proof. For two vertices u, v let $\sigma[u, v]$ be the permutation which exchanges u and v and fixes all other points. We say that a bijection $h : X(G) \to \tilde{X}(G)$ is *good except at node* $u \in V$ if

- $h(Z(v)) = Z(v)$ for all $v \in V$,

- h maps inner vertices to inner vertices and outer vertices to outer vertices,
- h is an isomorphism between the subgraphs $X(G) \setminus \{v^X : X \subseteq vE\}$ and $\tilde{X}(G) \setminus \{v^X : X \subseteq vE\}$, and
- for every pair $(a_{uv}, b_{uv}) \in Z(u)$, the mapping $h \circ \sigma[a_{uv}, b_{uv}]$ is an isomorphism from $X(G)[Z(u)]$ to $\tilde{X}(G)[Z(u)]$.

Let $\tilde{X}(G) = X_{(u,v)}(G)$. Then for instance $\sigma[a_{uv}, b_{uv}]$ is good except at u and $\sigma[a_{vu}, b_{vu}]$ is good except at v. Note that if $\eta \in \mathrm{Aut}(\tilde{X}(G))$ with $\eta(Z(v)) = Z(v)$ for all $v \in V$ and h is good except at vertex u, then $h \circ \eta$ is good except at u as well.

The property of being good at some vertex can be propagated along path in G: let P be a simple path in G from u to v, $P : u = v_1, v_2, \ldots, v_{l-1}, v_l = v$, and let h be a bijection which is good except at vertex u. Then the bijection $h' := h \circ \eta_P$ where

$$\eta_P := \sigma[a_{uv_2}, b_{uv_2}] \circ \pi_{v_2;v_1;v_3} \circ \cdots \circ \pi_{v_{l-1};v_{l-2};v_o mel} \circ \sigma[a_{vv_{l-1}}, b_{vv_{l-1}}],$$

is good except at v and for $w \notin P, x \in Z(w)$ we have $h'(x) = h(x)$.

Finally, we describe a winning strategy for Duplicator in the k-pebble bijection game played on $X(G)$ and $\tilde{X}(G)$. The strategy satisfies that pairs of pebbles (a_i, b_i) are always placed on vertices in a common gadget $Z(v)$. First of all, we initialize an instance of the k-cops and robber game played on G where we identify each of the k pairs of pebbles with one of the cops, and we assume that the robber makes his moves according to a fixed winning strategy (recall that $\mathrm{tw}(G) \geq k$). The positions in the two games are related as follows: the vertex in G occupied by the i-th cop is precisely the vertex $v \in V$ for which the corresponding gadget $Z(v)$ in $X(G)$ and $X(\tilde{G})$ is pebbled with the i-th pair (a_i, b_i) of pebbles in the k-pebble bijection game. We update the positions in the cops and robber game after each round of the k-pebble bijection game accordingly. Furthermore, whenever the robber is at some vertex $v \in V$, then Duplicator chooses in her current move some bijection which is good except at vertex v. For convenience, we assume that the robber starts at node u, and that in the first round Duplicator

answers with the bijection $\sigma[a_{uv}, b_{uv}]$. Recall that this bijection is good except at vertex u.

We proceed to show that Duplicator can maintain the following invariant during each play: let $((a_1, \ldots, a_k), (b_1, \ldots, b_k))$ be the current position in the k-pebble bijection game, then

there is a bijection $g : X(G) \to \tilde{X}(G)$ with $g(a_i) = b_i$ for $i \leq k$ such that g is good except at a vertex $u \in V$ and for $i \leq k$ we have $a_i, b_i \notin Z(u)$ (u is the robber's position in the cops and robber game).

This can be seen as follows: assume Spoiler chooses the i-th pair of pebbles. Duplicator answers with the bijection g and Spoiler puts the i-th pair of pebbles onto some tuple $(a, g(a))$. By the condition on g of being good except at u, the new position in the k-pebble bijection game is indeed a partial isomorphism (g is an isomorphism except at gadget $Z(u)$, and Spoiler would need more than one pebble there to uncover the difference). The move of Spoiler induces an update for the ith cop in the cops and robber game, which yields a respond of the robber according to his winning strategy, i.e. a move along a cop-free path P to some vertex v. Hence, as shown above, the bijection $g' := g \circ \eta_P$ respects all pebbled pairs of elements and is good except at v. Since, $Z(v)$ is cop-free (and hence not pebbled), the claim follows. Q.E.D.

Theorem 6.17. FPC $\subsetneq$ PTIME on every class of graphs which contains CFI-graphs $X(G)$ and $\tilde{X}(G)$ for graphs G of arbitrary large treewidth.

In fact, Grohe and Marino proved that FPC $\equiv$ PTIME on every class of graphs with bounded treewidth. Their theorem allows us to reformulate the result in a very neat way.

We first observe that the treewidth of $X(G)$ is bounded by $(O)(\mathrm{tw}(G))$: from a tree-decomposition of G one obtains a tree decomposition of $X(G)$ by replacing in all bags the vertices by their corresponding gadgets. Furthermore, the size of a gadget $Z(v)$ in $X(G)$ is bounded by $(4\Delta(G) \cdot 2^{\Delta(G)-1}) \in \mathcal{O}(\Delta(G))$. Now let G_n be the $n \times n$ grid, then $\mathrm{tw}(G_n) = n$, $\Delta(G_n) = 4$ and

$$\mathrm{tw}(X(G)) \leq (4\Delta(G) \cdot 2^{\Delta(G)-1}) \, \mathrm{tw}(G_n) = 24n \in \mathcal{O}(|G|).$$

For a function $f : \mathbb{N} \to \mathbb{N}$ we define the class of graphs

$$\mathrm{TW}_f := \{G : \mathrm{tw}(G) \leq f(|G|)\}.$$

Theorem 6.18. FPC $\equiv$ PTIME on TW_f if, and only if, $f \in \mathcal{O}(1)$.

Proof. The direction from right two left is mentioned theorem due to Marino and Grohe. For the other direction, assume $f \notin \mathcal{O}(1)$; then for every $n > 0$, there exists $k > |X(G_n)|$ with $f(k) \geq 24n$. Hence, TW_f contains $X(G_n)$ and $\tilde{X}(G_n)$ for every $n \geq 0$. Q.E.D.

7 Zero-one laws

Introduction, maybe we can use the introduction from the seminar?

7.1 Random graphs

We consider the class $\mathcal{G}_n$ of (undirected) graphs over $\{0, \dots, n-1\}$, i.e.

$$\mathcal{G}_n := \{G = (V, E) : G \text{ graph}, V = \{0, \dots, n-1\}\},$$

In order to introduce *random graphs* we consider a sequence of probability distributions $\overline{\mu} = (\mu_1, \mu_2, \dots)$ on $(\mathcal{G}_1, \mathcal{G}_2, \dots)$, i.e. $\mu_n : \mathcal{G}_n \to [0,1]$ and $\sum_{G \in \mathcal{G}_n} \mu(G) = 1$ for all $n \geq 1$. This defines a sequence of probability spaces $(\mathcal{G}_1, \mu_1), (\mathcal{G}_2, \mu_2), \dots$ on classes of graphs of increasing size.

Example 7.1.

(1) The *uniform distribution* μ_n assigns equal probability to each graph:

$$\mu_n(G) = \frac{1}{2^{\binom{n}{2}}}.$$

(2) Let $p : \mathbb{N} \to [0,1]$ be an arbitrary mapping. Then the probability space $\mathcal{G}_{n,p} = (\mathcal{G}_n, \mu_{p,n})$ is defined by the following random experiment: determine for every pair (u, v) with $0 \leq u < v < n$ whether $(u, v) \in E$ using a random variable X taking values $0, 1$ (False and True) with $\Pr[X = 1] = p(n)$ and $\Pr[X = 0] = (1 - p(n))$. Observe that for $p = \frac{1}{2}$ one obtains the uniform distribution.

We make the following convention: unless otherwise stated, μ_n denotes the uniform distribution. For a class $\mathcal{K}$ of graphs we set

$$\mu_n(\mathcal{K}) := \mu_n(\mathcal{K} \cap \mathcal{G}_n) = \sum_{G \in \mathcal{K} \cap \mathcal{G}_n} \mu_n(G).$$

This definition formalises what it means that a random graph $G \in \mathcal{G}_n$ has a certain property $\mathcal{K}$. However, in what follows, we are not interested in random graphs of some fixed size $n \in \mathbb{N}$ but much more in the behaviour of the probability $\mu_n(K)$ if we increase the size of graphs, i.e. if we let n approach infinity.

Definition 7.2. The *asymptotic probability* of a class $\mathcal{K}$ of graphs (with respect to $\overline{\mu}$) is defined as

$$\mu(\mathcal{K}) := \lim_{n\to\infty} \mu_n(\mathcal{K}),$$

in the case that this sequence has a limit. In particular, if ψ is a sentence over vocabulary $\{E\}$ in some logic $\mathcal{L}$, then the *asymptotic probability* of ψ (with respect to $\overline{\mu}$) is defined as

$$\mu(\psi) := \lim_{n\to\infty} \mu_n(\{G \in \mathcal{G}_n : G \models \psi\},$$

again only for the case that the limit exists.

Example 7.3.

(1) Let $\mathcal{K} = \{G : G \text{ is a clique}\}$. Then

$$\lim_{n\to\infty} \mu_n(\mathcal{K}) = \lim_{n\to\infty} \frac{1}{2^{\binom{n}{2}}} = 0.$$

(2) Let H be a graph and let $\mathcal{K}_H = \{G : G \text{ contains } H \text{ as subgraph}\}$. For $n > k \cdot |H|$ we have

$$\mu_n(\mathcal{K}_H) \geq 1 - (1 - (2^{-|E(H)|}))^k,$$

hence $\mu(K_H) = 1$ since $k \to \infty$ for $n \to \infty$.

(3) Let $\mathcal{K} = \{G : G \text{ is three-colourable}\}$. Then

$$\lim_{n\to\infty} \mu_n(\mathcal{K}) \leq 1 - \lim_{n\to\infty} \mu_n(\{G \in \mathcal{G}_n : G \text{ contains } K_4\}) = 0.$$

(4) Recall that we have $\lim_{n\to\infty} \mu_n(\{G : (3,17) \in E\}) = \frac{1}{2}$.

(5) The asymptotic probability is not defined for every class of graphs. For instance, consider $\mathcal{K} = \{G : G \text{ has an even number of nodes}\}$. Then the sequence $(\mu_n(\mathcal{K}))_{n\geq 1} = (0, 1, 0, 1, \dots)$ has no limit.

7.2 Zero-one law for first-order logic

In this section we prove the *zero-one law* for first-order logic:

Theorem 7.4. For sentences $\psi \in \mathrm{FO}$ (over relational vocabulary) we have

$$\mu(\psi) = 0 \quad \text{or} \quad \mu(\psi) = 1.$$

To put it in words, every first-order definable property of graphs either holds *almost never* or *almost surely* on random graphs of increasing size.

Definition 7.5. An *atomic graph k-type* is a maximal consistent set t of $\mathrm{FO}(\{E\})$-literals in variables $x_1, \ldots, x_k$, i.e. $Ex_ix_j, \neg Ex_ix_j, x_i = x_j, x_i \neq x_j$, which is consistent with the graph axioms $(\forall x_1 \forall x_2(\neg Ex_1x_1 \wedge Ex_1x_2 \leftrightarrow Ex_2x_1)$. Furhtermore, for a graph $G = (V, E)$ and $\overline{a} \in V^k$ we define the *atomic graph k-type of* $\overline{a}$ by

$$t_G(\overline{a}) := \{\varphi(x_i, x_j) : \varphi \text{ an } \mathrm{FO}(\{E\})\text{-literal such that } G \models \varphi(a_i, a_j)\}.$$

Formally, an atomic k-type t is a set but we frequently identify it with the formula $t(\overline{x}) = \bigwedge_{\varphi \in t} \varphi(\overline{x})$ (this formula is an FO-formula, since there are only finitely many $\{E\}$-literals in k variables).

In what follows, let $s(\overline{x})$ and $t(\overline{x})$ denote atomic graph types of tuples of distinct elements, i.e. $s, t \models \bigwedge_{i<j\leq k} x_i \neq x_k$. We say that an atomic $(m+1)$-type $t(x_1, \ldots, x_m, x_{m+1})$ *extends* an atomic m-type $s(x_1, \ldots, x_m)$ if $s \subseteq t$, or equivalently, if $t \models s$.

Definition 7.6. Let $s(x_1, \ldots, x_m)$ and $t(x_1, \ldots, x_m, x_{m+1})$ be atomic types such that $s \subseteq t$. We define the *extension axiom* $\sigma_{s,t}$ by

$$\sigma_{s,t} := \forall x_1 \cdots \forall x_m (s(\overline{x}) \rightarrow \exists x_{m+1} t(\overline{x}, x_{m+1})).$$

Furthermore, we let T be the set of all extension axioms together with the graph axioms.

The proof of the zero-one law for FO relies on the following properties of the extension axioms and the set T:

(1) $\mu(\sigma_{s,t}) = 1$ for all $\sigma_{s,t} \in T$.

(2) T is ω-categorical, i.e. there is, up to isomorphism, only one countable model of T. This structure is known as the *Rado graph*.
(3) T is complete, i.e. for all $\psi \in \mathrm{FO}$ either $T \models \psi$ or $T \models \neg\psi$.

We proceed to establish these three properties.

Lemma 7.7. Let $\sigma_{s,t} \in T$ be an extension axiom. Then $\mu(\sigma_{s,t}) = 1$.

Proof. Let $\sigma_{s,t} := \forall x_1 \cdots \forall x_m (s(\overline{x}) \rightarrow \exists x_{m+1} t(\overline{x}, x_{m+1}))$. For every $i = 1, \ldots, m$ we have $t \models Ex_i x_{m+1}$ or $t \models \neg Ex_i x_{m+1}$. Let $G \in \mathcal{G}_n$ be a random graph and $a_1, \ldots, a_m \in \{0, \ldots, n-1\}$. For every fixed $a_{m+1} \in V \setminus \{a_1, \ldots, a_m\}$, the experiments $G \models Ea_i a_{m+1}$ are stochastically independent and have probability $\frac{1}{2}$. Hence

$$\Pr[G \models t(\overline{a}, a_{m+1}) | G \models s(\overline{a})] = \frac{1}{2^m}.$$

Thus, probability that *no* element $a_{m+1} \in V \setminus \{a_1, \ldots, a_m\}$ extends a realisation $\overline{a}$ of s to a realisation of $(\overline{a}, a_{m+1})$ of t is $(1 - \frac{1}{2^m})^{n-m}$. In conclusion, we obtain

$$\begin{aligned} \mu_n(\neg\sigma_{s,t}) &= \mu_n(\exists x_1 \cdots \exists x_n (s(\overline{x}) \wedge \forall x_{m+1} \neg t(\overline{x}, x_{m+1}))) \\ &\leq n^m \cdot (1 - \frac{1}{2^m})^{n-m} \quad \overset{\text{exp. fast}}{\longrightarrow} \quad 0, \end{aligned}$$

and thus $\mu(\sigma_{s,t}) = 1$. Q.E.D.

The compactness theorem implies that also every logical consequence of the extensions axioms almost surely holds in a random graph.

Corollary 7.8. If $T \models \psi$ then $\mu(\psi) = 1$, and the set T is satisfiable.

Proof. If $T \models \psi$, then by the compactness theorem there is a finite set $T_0 \subseteq T$ such that $T_0 \models \psi$. Hence, we have $\mu_n(\psi) \geq \prod_{\sigma \in T_0} \mu_n(\sigma)$ for all $n \geq 1$ and thus $\lim_{n\to\infty}(\psi) = 1$ by Lemma 7.7. In particular $T \not\models \forall x(x \neq x)$ since $\mu(\forall x(x \neq x)) = 0$. Q.E.D.

Interestingly, one can give explicit description of models of T and we present two different possibilities here. However, as we show later that T is ω-categorical, these models are isomorphic.

Definition 7.9 (Rado graph)**.** The following graphs are models of T.

(1) Let p_i denote the i-th prime number. We define $G = (\mathbb{N}, E)$ with

$$E := \{(i,j) \in \mathbb{N} \times \mathbb{N} : p_i \mid j \text{ or } p_j \mid i.\}$$

We claim that $G \models T$. To see this, we choose an arbitrary extension axiom $\sigma_{s,t} := \forall x_1 \cdots \forall x_m (s(\overline{x}) \to \exists x_{m+1} t(\overline{x}, x_{m+1})) \in T$.
Let $I \cup J = \{1, \ldots, m\}$ be the partition defined by t with respect to the following condition

- If $t \models Ex_i x_{m+1}$ then $i \in I$, and
- if $t \models \neg Ex_i x_{m+1}$ then $i \in J$.

Let $a_1, \ldots, a_k \in A$ such that $G \models s(a_1, \ldots, a_k)$. We set $a_{m+1} := \prod_{i \in I} p_{a_i} q$ where q is a prime number with $q > p_{a_1} \cdots p_{a_m}$. Then it is easy to check that $G \models Ea_i a_{m+1}$ for all $i \in I$ and $G \models \neg Ea_j a_{m+1}$ for all $j \in J$.

(2) The set HF of *heriditarily finite sets* is defined by:

- $\emptyset \in \mathrm{HF}$
- If $a_1, \ldots, a_k \in \mathrm{HF}$, then also $\{a_1, \ldots, a_k\} \in \mathrm{HF}$.

Let $G = (\mathrm{HF}, E)$ with $E := \{(a,b) : a \in b \text{ or } b \in a\}$. Similarly as above, one can show that $G \models T$.

Theorem 7.10. Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two countable models of T. Then $G \cong H$. The unique countable model of T is known as the *Rado graph* $\mathcal{R}$.

Proof. First of all, it is clear that T has no finite models, hence G and H are infinite graphs. We fix two enumerations of V_G and V_H and inductively construct a sequence of partial isomorphism $p_0, p_1, p_2, \ldots$ between G and H such that $p_0 \subseteq p_1 \subseteq p_2 \subseteq \cdots$. For the base case, we set $p_0 := \emptyset$. For the induction step let $p_i = \{(a_1, b_1), \ldots, (a_i, b_i)\} \in \mathrm{Loc}(G, H)$ be already defined. We distinguish between the following two cases:

- If i is even, choose $a_{i+1} \in V_G$ to be the minimal element (with respect to the enumeration of V_G) which is not in the domain of p_i, i.e. $a_{i+1} \notin \{a_1, \ldots, a_i\}$. Let $s := t_G(a_1, \ldots, a_i)$ and $t :=$

$t_G(a_1, \ldots, a_{i+1})$. Since p_i is a partial isomorphism we know that $H \models s(b_1, \ldots, b_i)$. Since $H \models \sigma_{s,t}$ there exists an element $b_{i+1} \in V_H$ such that $H \models t(b_1, \ldots, b_{i+1})$. We set $p_{i+1} := p_i \cup \{(a_{i+1}, b_{i+1})\}$ and obtain a partial isomorphism extending p_i.

- If i is odd, we proceed analogously, but this time we let $b_{i+1} \in V_H$ be the minimal element (with respect to the enumeration of V_H) which is not in the image of p_i, i.e. $b_{i+1} \notin \{b_1, \ldots, b_i\}$. For $s := t_H(b_1, \ldots, b_i)$ and $t := t_H(b_1, \ldots, b_{i+1})$, the same reasoning as above yields an element $a_{i+1} \in V_G$ such that $G \models t(a_1, \ldots, a_{i+1}$. Again we obtain an extended partial isomorphism by setting $p_{i+1} := p_i \cup \{(a_{i+1}, b_{i+1})\}$.

Finally we let $p := \bigcup_{i \geq 0} p_i$. By construction we have that $\mathrm{dom}(p) = V_G$ and $\mathrm{im}(p) = V_H$, hence $p : G \xrightarrow{\sim} H$. Q.E.D.

In particular, ω-categorical theories are complete:

Theorem 7.11. *T axiomatises a complete theory, i.e. for all sentences $\psi \in \mathrm{FO}(\{E\})$ we have $T \models \psi$ or $T \models \neg\psi$.*

Proof. Assume for some sentence $\psi \in \mathrm{FO}(\{E\})$ it holds that $T \not\models \psi$ and $T \not\models \neg\psi$. Then by the downwards Löwenheim-Skolem theorem, there exist two countable graphs G and H with $G \models T \cup \{\psi\}$ and $H \models T \cup \{\neg\psi\}$. In particular this implies $G \ncong H$, which contradicts Theorem 7.10. Q.E.D.

Theorem 7.12. [Glebskiĭ et al., R. Fagin] *For all $\psi \in \mathrm{FO}(\{E\})$ it holds:*

$$\mu(\psi) = 0 \quad \text{or} \quad \mu(\psi) = 1.$$

Proof. If $T \models \psi$, then $\mu(\psi) = 1$. Otherwise, $T \models \neg\psi$, and hence $\mu(\psi) = 1 - \mu(\neg\psi) = 0$. Q.E.D.

In particular, we can give a precise characterisation of those first-order properties which hold almost surely in random graphs.

Corollary 7.13. *Let $\psi \in \mathrm{FO}(\{E\})$. Then*

$$\mu(\psi) = 1 \quad \text{iff} \quad T \models \psi \quad \text{iff} \quad \mathcal{R} \models \psi.$$

7.2.1 Applications

We can use Theorem 7.12 to show that certain classes of graphs are not definable in first-order logic: if a class $\mathcal{K}$ of graphs has undefined asymptotic probability or an asymptotic probability different from 0 and 1, then clearly $\mathcal{K}$ cannot be defined in first-order logic. More generally, this method yields non-definability of $\mathcal{K}$ for *every* logic that has a 0-1-law, e.g. for $L^{\omega}_{\infty\omega}$ as we see later. For instance, consider the class EvenV $= \{G = (V, E) : |V| \text{ is even}\}$ with undefined asymptotic probability or the class EvenE $= \{G = (V, E) : |E| \text{ is even}\}$ with $\mu(\text{EvenE}) = \frac{1}{2}$. Moreover, we can use our results as a convenient method to determine the asymptotic probability for many natural classes of graphs.

(1) We want to determine $\mu(\text{Con})$ where Con denotes the class of connected graphs. Let s be an atomic 2-type in variables x, y containing $\neg Exy$ and let t be the atomic 3-type in variables x, y, z which extends s and contains $Exz \wedge Eyz$. Then $G \models \sigma_{s,t}$ iff G has diameter at most 2. Hence, $G \models \sigma_{s,t}$ implies $G \in \text{Con}$, which means that $\mu(\text{Con}) = 1$.

(2) Let $\mathcal{K}$ be any class of graphs which exclude a forbidden subgraph $H = (\{v_1, \ldots, v_k\}, E)$. Then $\mu(\mathcal{K}) = 0$. To see this, we set $s_i(x_1, \ldots, x_i) := t_H(v_1, \ldots, v_i)$ for $i \leq k$ and consider the extension axioms $\sigma_{s_i s_{i+1}}$. Then clearly $\psi := \bigwedge_{i<k} \sigma_{s_i s_{i+1}}$ is a logical consequence of T, which means that $\mu(\psi) = 1$. Moreover, if $G \models \psi$, then G contains H as an induced subgraph. We conclude that $\mu(\mathcal{K}) \leq 1 - \mu(\psi) = 0$. As an application, consider the class of planar graphs which exclude K_5 (the complete graph on 5 vertices) and the class of k-colourable graphs which exclude K_{k+1} (where k is fixed). To put it in words, a random graph is almost never planar nor k-colourable.

7.3 Generalised zero-one laws

In this section we want to generalise our considerations in two different ways. Firstly, instead of restricting ourselves to graphs, we want to work on more general classes of structures and analyse whether the

zero-one-law for FO still holds. Secondly, as FO has rather limited expressive power, we look for zero-one laws for more powerful logics as well.

Let τ be an arbitrary vocabulary (not necessarily relational). By $\mathrm{Str}_n(\tau)$ we denote the set of all τ-structures over the universe $\{0, \dots, n-1\}$. As before we define a sequence $\overline{\mu} = (\mu_1, \mu_2, \dots)$ of uniform probability distributions $\mu_n : \mathrm{Str}_n(\tau) \to [0,1]$, i.e. for every $\mathfrak{A} \in \mathrm{Str}_n(\tau)$ we set

$$\mu_n(\mathfrak{A}) = \frac{1}{|\mathrm{Str}_n(\tau)|}.$$

We claim that $\mathrm{FO}(\tau)$ has a zero-one law if, and only if, τ contains no function symbols. To this end, we first consider the case where τ contains function symbols:

(1) Assume $\{P, c\} \subseteq \tau$ where c is a constant symbol and P a monadic relation. Then for $\psi := Pc$ we have $\mu_n(\psi) = \frac{1}{2}$ for all $n \geq 1$, hence $\mu(\psi) = \frac{1}{2}$, i.e. the zero-one law does hold in this case.

(2) Assume $f \in \tau$ where f is a unary function symbol. Consider the $\mathrm{FO}(\tau)$-sentence $\psi := \exists x(fx = x)$ stating that f has a fixed point. For $n \geq 1$ we have

$$\mu_n(\psi) = 1 - \prod_{i=0}^{n-1} \underbrace{\left(\frac{n-1}{n}\right)}_{=\Pr[f(i) \neq i]} = 1 - \left(1 - \frac{1}{n}\right)^n.$$

Since $\left(1 - \frac{1}{n}\right)^n \longrightarrow e^{-1}$ for $n \to \infty$, the zero-one law does not hold in this case either.

For the other direction, let τ be purely relational, $\tau = \{R_1, \dots, R_k\}$. The proof strategy we used over graphs generalises for this general in a straightforward way:

- An *atomic τ-type in k variables* is a maximal, consistent set of τ-literals over variables $x_1, \dots, x_k$. For a τ-structure $\mathfrak{A}$ and $\overline{a} \in \mathfrak{A}$ we set $t_{\mathfrak{A}}(\overline{a}) = \{\varphi(\overline{x}) : \varphi \text{ a } \tau\text{-literal with } \mathfrak{A} \models \varphi(\overline{a})\}$.
- The *τ-extension axiom $\sigma_{s,t}$* for two atomic τ-types s and t (in k and

$k+1$ variables, respectively) with $s \subseteq t$ is defined as

$$\sigma_{s,t} := \forall \overline{x}(s(\overline{x}) \to \exists x_{k+1} t(\overline{x}, x_{k+1})).$$

As before, we let T denote the set of all τ-extension axioms

- Again we can show that $\mu(\sigma_{s,t}) = 1$ for all $\sigma_{s,t} \in T$. Let r denote the number of literals in t which contain x_{m+1}. Then, for a random structure $\mathfrak{A} \in \mathrm{Str}_n(\tau)$, $\overline{a} \in A$ and a_{m+1} it holds

$$\Pr[\mathfrak{A} \models t(\overline{a}, a_{m+1}) \mid \mathfrak{A} \models s(\overline{a})] = 2^{-r}.$$

Thus

$$\begin{aligned}\mu_n(\neg\sigma_{s,t}) &= \mu_n(\exists\overline{x}(s(\overline{x}) \wedge \forall x_{m+1} \neg t(\overline{x}, x_{m+1})))\\ &\leq n^m(1-2^{-r})^{n-m} \quad \overset{\text{exp. fast}}{\longrightarrow} \quad 0.\end{aligned}$$

- T is ω-categorical: analogously!

Our analysis raises the question why even basic functions but not arbitrary relations inhibit a zero-one law. The reason is that atomic experiments are not longer stochastically independent. For instance, consider the experiments $f(a) = b$ and $f(a) = c$ (for $b \neq c$), then $\Pr[f(a) = c \mid f(a) = b] = 0 \neq \Pr[f(a) = c]$.

7.3.1 Zero-one law for $L^\omega_{\infty\omega}$

We proceed to show that the zero-one law holds for $L^\omega_{\infty\omega}$ as well (restricted to relational vocabularies). In particular, since LFP $\leq L^\omega_{\infty\omega}$, this means that a random graph either almost surely has an LFP-definable property or almost never does. With FO^k we denote the k-variable fragment of FO, i.e. $\mathrm{FO}^k = \mathrm{FO} \cap L^k_{\infty\omega} = \{\varphi \in \mathrm{FO} : \varphi \text{ only contains variables } x_1, \ldots, x_k\}$. If we restrict the set of extension axioms T to FO^k we obtain finite sets of approximations of T which are again sentences in FO^k; more specifically, we set

$$\Theta_k := \bigwedge T \cap \mathrm{FO}^k = \bigwedge\{\sigma_{s,t} : \sigma_{s,t} \in T \cap \mathrm{FO}^k\} \in \mathrm{FO}^k.$$

The central property of these approximations for T is stated in the following theorem: in models of Θ_k, every $L^k_{\infty\omega}$-formula is equivalent to a simple Boolean combinations of atomic k-types. In particular, every $L^k_{\infty\omega}$-sentence is either true or false in all models of Θ_k.

Theorem 7.14. Let $m \leq k$, $s(x_1, \ldots, x_m)$ an atomic m-type and $\varphi(x_1, \ldots, x_m) \in L^k_{\infty\omega}$. Then

$$\begin{array}{ll} \text{either} & \Theta_k \models \forall \overline{x}(s(\overline{x}) \to \varphi(\overline{x})) \\ \text{or} & \Theta_k \models \forall \overline{x}(s(\overline{x}) \to \neg\varphi(\overline{x})). \end{array}$$

Proof. We proceed by induction on φ and simultaneously show the claim for all $m \leq k$ and atomic types s. If φ is atomic, then either $\varphi \in s$ or $\neg\varphi \in s$. If $\varphi = \neg\psi$, the claim directly follows.

Let $\varphi = \bigwedge \Psi$, $\Psi \subseteq L^k_{\infty\omega}$. By induction hypothesis for all $\psi \in \Psi$

$$\begin{array}{ll} \text{either} & \Theta_k \models \forall \overline{x}(s(\overline{x}) \to \psi(\overline{x})) \\ \text{or} & \Theta_k \models \forall \overline{x}(s(\overline{x}) \to \neg\psi(\overline{x})). \end{array}$$

If $\Theta_k \models \forall \overline{x}(s(\overline{x}) \to \psi(\overline{x}))$ for all $\psi \in \Psi$, then $\Theta_k \models \forall \overline{x}(s(\overline{x}) \to \bigwedge \Psi(\overline{x}))$. Otherwise, $\Theta_k \models \forall \overline{x}(s(\overline{x}) \to \neg \bigwedge \Psi(\overline{x}))$.

Let $\varphi(\overline{x}) = \exists y \psi(\overline{x}, y)$ and assume that $\Theta_k \not\models \forall \overline{x}(s(\overline{x}) \to \neg\varphi(\overline{x}))$. Choose a structure $\mathfrak{A} \models \Theta_k$ with $\mathfrak{A} \models \exists \overline{x}(s(\overline{x}) \wedge \exists y \psi(\overline{x}, y))$ and consider the following two cases

- If $y \notin \{x_1, \ldots, x_m\}$, i.e. $y \in \{x_{m+1}, \ldots, x_k\}$; let $a_1, \ldots, a_m, b \in A$ such that $\mathfrak{A} \models s(\overline{a}) \wedge \psi(\overline{a}, b)$. We define the atomic type $t(x_1, \ldots, x_m, y) := t_{\mathfrak{A}}(\overline{a}, b)$ with $s \subseteq t$. In particular,

  $$\mathfrak{A} \models \exists \overline{x} \exists y (t(\overline{x}, y) \wedge \psi(\overline{x}, y)).$$

  By induction hypothesis we know that

  $$\mathfrak{A} \models \forall \overline{x} \forall y (t(\overline{x}, y) \to \psi(\overline{x}, y)),$$

  and since $\sigma_{s,t} = \forall \overline{x}(s(x) \to \exists y t(\overline{x}, y))$ is an extension axiom con-

tained in Θ_k we finally obtain

$$\mathfrak{A} \models \forall \overline{x}(s(\overline{x}) \to \exists y \psi(\overline{x}, y)).$$

- If $y \in \{x_1, \ldots, x_m\}$, i.e. $y = x_j$ for $j \leq m$; let $\overline{a} \in A$ such that $\mathfrak{A} \models s(\overline{a}) \wedge \exists x_j \psi(\overline{a})$, and let $\overline{x}^*$ and $\overline{a}^*$ denote the tuples $\overline{x}$ and $\overline{a}$ without the j-th componenent, i.e.

$$\overline{x}^\star := x_1 \cdots x_{j-1} x_{j+1} \cdots x_k$$
$$\overline{a}^\star := a_1 \cdots a_{j-1} a_{j+1} \cdots a_k.$$

Similarly, let $s^\star(\overline{x}^\star) := t_{\mathfrak{A}}(\overline{a}^\star)$ be the atomic type of $\overline{a}^\star$ in $\mathfrak{A}$. Then $s^\star \subseteq s$ and there is $b \in A$ such that

$$\mathfrak{A} \models s^\star(\overline{a}^\star) \wedge \psi(\overline{a}\frac{b}{a_j}), \quad \text{where } \overline{a}\frac{b}{a_j} := a_1 \cdots a_{j-1} b a_{j+1} \cdots a_m.$$

For $t^\star(\overline{x}) := t_{\mathfrak{A}}(\overline{a}\frac{b}{a_j})$ we thus have $\mathfrak{A} \models \exists(t^\star(\overline{x}) \wedge \psi(\overline{x}))$, and the induction hypothesis yields

$$\Theta_k \models \forall \overline{x}(t^\star(\overline{x}) \to \psi(\overline{x})).$$

As above, since $s^\star \subseteq t^\star$, it holds that $\Theta_k \models \forall \overline{x}^\star(s^\star(\overline{x}^\star) \to \exists x_j t^\star(\overline{x}))$, and altogether we obtain

$$\Theta_k \models \forall \overline{x}(s(\overline{x}) \to \exists x_j \psi(\overline{x})).$$

Q.E.D.

Corollary 7.15. For every $L^k_{\infty\omega}$-sentence ψ we either have $\Theta_k \models \psi$ or $\Theta_k \models \neg\psi$.

Corollary 7.16. If $\mathfrak{A} \models \Theta_k$ and $\mathfrak{B} \models \Theta_k$, then $\mathfrak{A} \equiv_{L^k_{\infty\omega}} \mathfrak{B}$.

Corollary 7.17 (Kolaitis, Varidi 1992)**.** For every sentence $\psi \in L^\omega_{\infty\omega}$ (over a relational signature) we have $\mu(\psi) = 0$ or $\mu(\psi) = 1$.

Proof. Let $\psi \in L^k_{\infty\omega}$ for some $k \geq 1$. Then by Corollary 7.15 we have $\Theta_k \models \psi$ or $\Theta_k \models \neg\psi$. Since $\Theta_k \subseteq T$ is finite, we have $\mu(\Theta_k) = 1$ and thus the claim follows. Q.E.D.