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 $\Box \in \Sigma$,
- $\Gamma \subseteq \Sigma \setminus \{\Box\}$ 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 $C = (q, p, w = w_0 w_1 \ldots w_k) \in Q \times \mathbb{N} \times \Sigma^*$. 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 $\mathrm{Next(C)} = (q', p + m, w_0 \ldots w_{p-1} a w_{p+1} \cdots w_k)$. 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 $C_{i+1} = \mathrm{Next(C_i)}$ 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 $F = F^+ \dot\cup F^-$ 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

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

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

$$\tau_\infty := \{R^i_j : i, j \in \mathbb{N}\} \cup \{f^i_j : i, j \in \mathbb{N}\}$$

where R^i_j stands for a relation symbol of arity i and f^i_j 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^i_j	$\longmapsto$	$R[\text{bin } i][\text{bin } j]$
functional symbols:	f^i_j	$\longmapsto$	$f[\text{bin } i][\text{bin } j]$
variables:	x_j	$\longmapsto$	$x[\text{bin } j]$.

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

Let $X \subseteq$ 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 $X \subseteq$ 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*(FO(τ)), *Non-sat*(FO(τ)) and *Inf-axioms*(FO(τ)) 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 $\implies \psi_M$ is an infinity axiom.

Then the theorem follows by Lemma 1.13.

Let M be a Turing acceptor with states $Q = \{q_0, \dots, q_r\}$, initial state q_0, alphabet $\Sigma = \{a_0, \dots, a_s\}$ (where $a_0 = \Box$), 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 $A = \{0, 1, 2, \dots, n\}$ 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.

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

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

[Enforces input configuration on ε at time 0]

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

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

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

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

where

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

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

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

and

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

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

[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 $\psi_M \in \textit{Fin-sat}(FO(\tau))$.
- 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 $k \le m$, $\mathfrak{B} \models w(\boldsymbol{k}, \boldsymbol{t}) = \boldsymbol{i_k}$ 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 $C_{t+1} = \mathrm{Next(C_t)}$ 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.

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

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

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

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

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 $\textit{Fin-sat}(X), \textit{Inf-axioms}(X)$ and $\textit{Non-sat}(X)$ are pairwise recursively inseparable, and thus $\textit{Fin-sat}(X)$, $\textit{Sat}(X), \textit{Val}(X), \textit{Non-sat}(X), \textit{Inf-axioms}(X)$ are undecidable.

Proof. A conservative reduction from FO to X yields a uniform reduction from $\textit{Fin-sat}(\mathrm{FO})$, $\textit{Inf-axioms}(\mathrm{FO})$ and $\textit{Non-sat}(\mathrm{FO})$ to $\textit{Fin-sat}(X)$, $\textit{Inf-axioms}(X)$ and $\textit{Non-sat}(X)$, respectively. Q.E.D.

We now observe that we can indeed give a complete classification of signatures σ such that $\mathrm{FO}(\sigma)$ 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 $\textit{Sat}(\mathrm{FO}(\sigma))$ is decidable. In all other cases, $\textit{Sat}(\mathrm{FO}(\sigma))$, $\textit{Inf-axioms}(\mathrm{FO}(\sigma))$ and $\textit{Non-sat}(\mathrm{FO}(\sigma))$ are pairwise recursively inseparable, and $\mathrm{FO}(\sigma)$ 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 $\varphi \in \mathrm{FO}$ 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 $Inf\text{-}axioms(X)$ 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

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

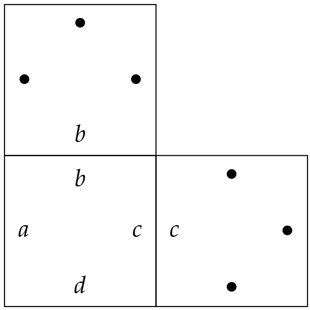


Figure 1.1. Domino adjacency condition

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.

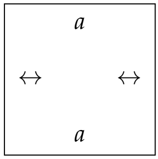
Proof. We reduce $H_\varepsilon^\omega = \{\rho(M) : \text{the computation of } M \text{ on } \varepsilon \text{ is infinite}\}$, which is co-r.e., to CORNER-DOMINO.

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.

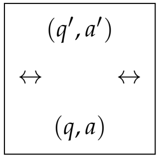
Assume w.l.o.g. that M never moves off-tape to the left, i.e. in configurations $(q, 0, w)$ it is never the case that $\delta(q, w_0) = (q', a, -1)$.

D consists of the following domino types.

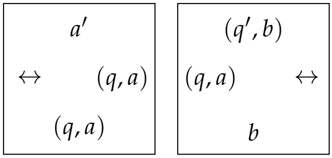
For each $a \in \Sigma$



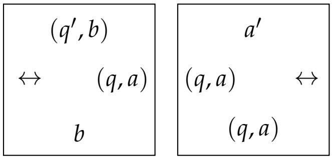
For each $(q, a) \in Q \times \Sigma$ with $\delta(q, a) = (q', a', 0)$



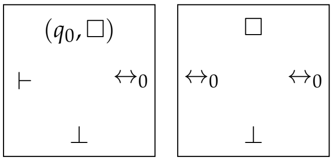
For each $(q, a) \in Q \times \Sigma$ with $\delta(q, a) = (q', a', 1)$, for each $b \in \Sigma$



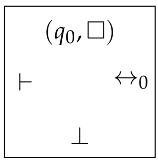
For each $(q, a) \in Q \times \Sigma$ with $\delta(q, a) = (q', a', -1)$ for each $b \in \Sigma$



Additionally there exist dominoes

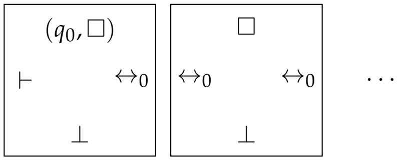


The origin constraint D_0 consists of



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 $C_{j+1} = \mathrm{Next(C_j)}$.

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 $X \in \mathrm{FO}$. 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

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

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:

$$\begin{tikzcd} \bullet \arrow[r, "f"'] & \bullet \arrow[r, "f"'] & b \cdots & & \cdots \bullet \arrow[ll, bend left, "f"] \end{tikzcd}$$

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

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

We claim that there exists a tiling $\sigma : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{D}$ 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 $\mathfrak{B} = (B, f, g, (h_d)_{d \in \mathcal{D}}) \models \psi_{\mathcal{D}}$.

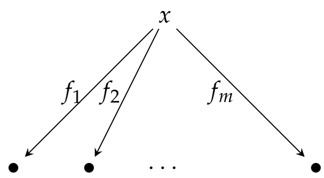
Choose an arbitrary $b \in B$ and define

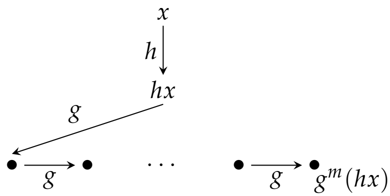
$$\sigma : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{D} : \sigma(i,j) := d \text{ iff } \mathfrak{B} \models h_d f^i g^j b = f^i g^j b.$$

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 $\mathfrak{A} = (A, f_1, \dots, f_m)$ with

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

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

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

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

This transformation preserves the meaning of terms: Let $t(x) = f_{i_1} \dots f_{i_k} x$ be a term in φ. Then $\tilde{t}(x) = hg^{i_1} \dots hg^{i_k} hx$, 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)

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

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

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.