Finite Model Theory

George Voutsadakis[1]

[1]Mathematics and Computer Science
Lake Superior State University

LSSU Math 600

Subsection 1

Some Extensions of First Order Logic

Operators on Power Sets

- Let M be a finite nonempty set.
- Denote by $\text{Pow}(M)$ the power set of M.
- Let $F : \text{Pow}(M) \to \text{Pow}(M)$ be a function.
- F induces a sequence $\varnothing, F(\varnothing), F(F(\varnothing)), \ldots$ of subsets of M.
- For its members we write $F_0, F_1, \ldots$.
- So we have

$$F_0 = \varnothing \quad \text{and} \quad F_{n+1} = F(F_n), \quad n \geq 0.$$

Fixed-Points and Inflationarity

- Let M be a finite nonempty set and $F : \text{Pow}(M) \to \text{Pow}(M)$.
- Suppose there is an $n_0 \geq 0$ such that $F_{n_0+1} = F_{n_0}$, i.e., $F(F_{n_0}) = F_{n_0}$.
- Then $F_m = F_{n_0}$, for all $m \geq n_0$.
- We denote F_{n_0} by F_∞.
- Moreover, we say that the **fixed-point** F_∞ of F **exists**.
- In case the fixed-point F_∞ does not exist, we agree to set $F_\infty := \varnothing$.
- F is said to be **inflationary** if

$$X \subseteq F(X), \quad \text{for all } X \subseteq M.$$

Periodicity and Fixed-Points

Lemma

(a) The sequence $(F_n)_{n\geq 0}$ is periodic. More precisely, there are $m < 2^{\|M\|}$ and $\ell \geq 1$ such that

$$F_k = F_{k+\ell}, \quad \text{for all } k \geq m.$$

(b) If F_∞ exists then $F_\infty = F_{2^{\|M\|}-1}$.

(c) If F is inflationary then F_∞ exists and $F_\infty = F_{\|M\|}$.

(a) Note that $\text{Pow}(M)$ has $2^{\|M\|}$ elements.
So there are $m < 2^{\|M\|}$ and $\ell \geq 1$ such that $F_m = F_{m+\ell}$.
Therefore,

$$\begin{array}{rcl} F_{m+1} & = & F(F_m) = F(F_{m+\ell}) = F_{m+1+\ell}; \\ F_{m+2} & = & F_{m+2+\ell}; \\ & \vdots & \end{array}$$

Periodicity and Fixed-Points (Cont'd)

(b) Choose $m < 2^{\|M\|}$ and $\ell \geq 1$ according to Part (a).

If $F_m = F_{m+1}$, then $F_m = F_{2^{\|M\|}-1} = F_\infty$.

If $F_m \neq F_{m+1}$ then, by Part (a), for $s \geq m$, we get

$$F_{m+s\cdot\ell} = F_m \neq F_{m+1} = F_{m+1+s\cdot\ell}.$$

Hence, F_∞ does not exist.

(c) By assumption, $F_0 \subseteq F_1 \subseteq \cdots \subseteq M$.

But M has $\|M\|$ elements.

So this sequence must get constant not later than with $F_{\|M\|}$.

Power Set Operations Induced By Formulas

- Let $\varphi(x_1,\ldots,x_k,\overline{u},X,\overline{Y})$ be a formula in the vocabulary τ, where the relation variable X has arity k.
- Let $\mathcal{A}$ be a τ-structure, $\overline{b}$ an interpretation of $\overline{u}$ in A, and $\overline{S}$ an interpretation of $\overline{Y}$ over A.
- Then $\varphi,\mathcal{A},\overline{b}$ and $\overline{S}$ give rise to an operation

$$F^{\varphi}:\mathrm{Pow}(A^k)\to\mathrm{Pow}(A^k)$$

defined by

$$F^{\varphi}(R):=\{(a_1,\ldots,a_k):\mathcal{A}\vDash\varphi[a_1,\ldots,a_k,\overline{b},R,\overline{S}]\}.$$

- Note that the notation F^{φ} does not make explicit all relevant data.

Example

- Let $\mathcal{G} = (G, E^G)$ be a graph.
  Let
  $$\varphi_0(x, y, X) := (Exy \vee \exists z(Xxz \wedge Ezy)),$$
  with xy corresponding to $\overline{x}$ above.
  Then, we have:
  $$\begin{array}{rcl} F_0^{\varphi_0} & = & \varnothing; \\ F_1^{\varphi_0} & = & F^{\varphi_0}(\varnothing) = E^G; \\ F_2^{\varphi_0} & = & F^{\varphi_0}(E^G) \\ & = & E^G \cup \{(a, b) : (E^G ac \text{ and } E^G cb) \text{ for some } c \in G\}. \end{array}$$
  By induction on n, one shows that
  $$F_n^{\varphi_0} = \{(a, b) : \text{there is a path of length} \leq n \text{ from } a \text{ to } b\}.$$
  Hence,
  $$F_\infty^{\varphi_0} = \{(a, b) : \text{there is a path from } a \text{ to } b\}.$$

A Remark on Power Set Operations Induced by Formulas

- Let $\varphi(\overline{x}, \overline{u}, X, \overline{Y})$ be a τ-formula, where:
  - The relation variable X has arity k;
  - $\overline{x}$ is of length k.
- Let $\mathcal{A}$ be a τ-structure, $\overline{b}$ an interpretation of $\overline{u}$ in A and $\overline{S}$ an interpretation of $\overline{Y}$ over A.
- Define

$$\psi := X\overline{x} \vee \varphi.$$

- Then the operation $F^{\psi} : \mathrm{Pow}(A^k) \to \mathrm{Pow}(A^k)$ is inflationary.

  We have, using the definition, that, for all $\overline{a}$ in A,

$$\begin{array}{rcl} F^{\psi} & = & \{\overline{a} : \mathcal{A} \vDash (X\overline{x} \vee \varphi)[\overline{a}, \overline{b}, R, \overline{S}]\} \\ & = & \{\overline{a} : \mathcal{A} \vDash R\overline{x}[\overline{a}]\} \cup \{\overline{a} : \mathcal{A} \vDash \varphi[\overline{a}, \overline{b}, R\overline{S}]\} \\ & \supseteq & R^{A}. \end{array}$$

Inflationary Fixed-Point Logic

- **Inflationary Fixed-Point Logic** FO(IFP) is obtained by closing first-order logic FO under inflationary fixed-points of definable operations.
- For a vocabulary τ, the class FO(IFP)$[\tau]$ of **formulas of** FO(IFP) **of vocabulary** τ is given by the following clauses:
  - φ, where φ is an atomic second-order formula over τ;
  - If φ, ψ are formulas in FO(IFP)$[\tau]$, then
  $$\neg\varphi, \quad (\varphi \vee \psi), \quad \exists x \varphi$$
  are formulas in FO(IFP)$[\tau]$;
  - If φ is a formula in FO(IFP)$[\tau]$, then
  $$[\mathrm{IFP}_{\overline{x},X}\varphi]\overline{t},$$
  where the lengths of $\overline{x}$ and $\overline{t}$ are the same and coincide with the arity of X, is a formula in FO(IFP)$[\tau]$.

Partial Fixed-Point Logic

- **Partial Fixed-Point Logic** FO(PFP) is obtained by closing first-order logic FO under arbitrary fixed-points of definable operations.
- For a vocabulary τ, the class FO(PFP)$[\tau]$ of **formulas of** FO(PFP) **of vocabulary** τ is given by the following clauses:
  - φ, where φ is an atomic second-order formula over τ;
  - If φ, ψ are formulas in FO(PFP)$[\tau]$, then
    $$\neg\varphi, \quad (\varphi \vee \psi), \quad \exists x \varphi$$
    are formulas in FO(PFP)$[\tau]$;
  - If φ is a formula in FO(PFP)$[\tau]$, then
    $$[\mathrm{PFP}_{\overline{x},X}\varphi]\overline{t},$$
    where the lengths of $\overline{x}$ and $\overline{t}$ are the same and coincide with the arity of X, is a formula in FO(PFP)$[\tau]$.

Sentences of Fixed-Point Logic

- **Sentences** are formulas without free first-order and second-order variables.
- The free occurrence of variables is defined in the standard way.
- For FO(IFP) one adds the clause

$$\mathrm{free}([\mathrm{IFP}_{\overline{x},X}\varphi]\overline{t}) := \mathrm{free}(\overline{t}) \cup (\mathrm{free}(\varphi) \backslash \{\overline{x}, X\}).$$

- Similarly, for FO(PFP) one adds the clause

$$\mathrm{free}([\mathrm{PFP}_{\overline{x},X}\varphi]\overline{t}) := \mathrm{free}(\overline{t}) \cup (\mathrm{free}(\varphi) \backslash \{\overline{x}, X\}).$$

Semantics of Fixed-Point Logic

- The semantics is defined inductively on the structure of formulas.
- The intended meanings of the fixed-point clauses are:

$$[\mathrm{IFP}_{\overline{x},X}\varphi]\overline{t} \quad \text{means that} \quad \overline{t} \in F_{\infty}^{(X\overline{x}\vee\varphi)};$$
$$\left[\mathrm{PFP}_{\overline{x},X}\varphi\right]\overline{t} \quad \text{means that} \quad \overline{t} \in F_{\infty}^{\varphi}.$$

- More precisely, if X is k-ary and if the variables free in $[\mathrm{IFP}_{\overline{x},X}\varphi]\overline{t}$ are among $\overline{u}$ and $\overline{Y}$, and $\overline{b}$ and $\overline{S}$ are interpretations in $\mathcal{A}$ of $\overline{u}$ and $\overline{Y}$, respectively, then:

$$\mathcal{A} \vDash [\mathrm{IFP}_{\overline{x},X}\varphi]\overline{t}[\overline{b},\overline{S}] \quad \text{iff} \quad (t_1[\overline{b}],\ldots,t_k[\overline{b}]) \in F_{\infty}^{(X\overline{x}\vee\varphi)};$$
$$\mathcal{A} \vDash [\mathrm{PFP}_{\overline{x},X}\varphi]\overline{t}[\overline{b},\overline{S}] \quad \text{iff} \quad (t_1[\overline{b}],\ldots,t_k[\overline{b}]) \in F_{\infty}^{\varphi}.$$

Example (Graphs)

- We work over the language of graphs.

  Consider the formula of FO(IFP)

  $$\psi_0(x,y) := [\mathrm{IFP}_{xy,X}(Exy \vee \exists z(Xxz \wedge Ezy))]xy.$$

  It expresses that x, y are connected by a path.

  Hence, the class CONN of connected graphs is axiomatizable in FO(IFP) by the set consisting of:
  - The graph axioms;
  - The sentence

    $$\forall x \forall y(\neg x = y \to \psi_0(x,y)).$$

  This class is not axiomatizable in FO.

Example (Orderings)

- We work over the vocabulary $\tau = \{<, S, \min, \max\}$.
  Consider the sentence of FO(IFP)

$$\neg[\mathrm{IFP}_{x,X}(x = \min \vee \exists y \exists z(Xy \wedge Syz \wedge Szx))] \max .$$

  The class of orderings of even cardinality is axiomatizable in FO(PFP) by the set consisting of:
  - The ordering axioms;
  - The sentence

$$\neg[\mathrm{IFP}_{x,X}(x = \min \vee \exists y \exists z(Xy \wedge Syz \wedge Szx))] \max .$$

  The same holds for the class of orderings of odd cardinality if we add instead the FO(PFP)-sentence

$$\exists x[\mathrm{PFP}_{x,X}\psi(x, X)]x,$$

  where

$$\psi(x, X) = (\forall y \neg Xy \wedge x = \min) \vee (X \max \wedge x = \max) \\ \vee \exists y(Xy \wedge \exists u(Syu \wedge Sux)).$$

Expressivity Relations

Definition

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be logics.

(a) $\mathcal{L}_1 \leq \mathcal{L}_2$ (read: $\mathcal{L}_1$ **is at most as expressive as** $\mathcal{L}_2$) if, for every τ and every sentence $\varphi \in \mathcal{L}_1[\tau]$, there is a sentence $\psi \in \mathcal{L}_2[\tau]$, such that $\text{Mod}(\varphi) = \text{Mod}(\psi)$.

(b) $\mathcal{L}_1 \equiv \mathcal{L}_2$ (read: $\mathcal{L}_1$ and $\mathcal{L}_2$ **have the same expressive power**) if $\mathcal{L}_1 \leq \mathcal{L}_2$ and $\mathcal{L}_2 \leq \mathcal{L}_1$.

(c) $\mathcal{L}_1 < \mathcal{L}_2$ if $\mathcal{L}_1 \leq \mathcal{L}_2$ and not $\mathcal{L}_2 \leq \mathcal{L}_1$.

- In most cases, a proof of $\mathcal{L}_1 \leq \mathcal{L}_2$ even yields that every formula of $\mathcal{L}_1$ is equivalent to a formula of $\mathcal{L}_2$.
- In particular, $\mathcal{L}_1 \leq \mathcal{L}_2$ implies that $\mathcal{L}_1 \leq \mathcal{L}_2$ holds for all formulas of $\mathcal{L}_1$ containing only free *individual* variables (one replaces these variables by new constants).

First-Order and First-Order with Fixed-Points

- Since $\mathcal{L}_1 \leq \mathcal{L}_2$ implies that $\mathcal{L}_1 \leq \mathcal{L}_2$ holds for all formulas of $\mathcal{L}_1$ containing only free individual variables, $\mathcal{L}_1 \leq \mathcal{L}_2$ will imply that every global relation definable in $\mathcal{L}_1$ is also definable in $\mathcal{L}_2$.
- By the preceding example, we have $\mathrm{FO} < \mathrm{FO(IFP)}$.

Proposition

$\mathrm{FO(IFP)} \leq \mathrm{FO(PFP)}$.

- Note that $[\mathrm{IFP}_{\overline{x},X}\varphi]\overline{t}$ is equivalent to $[\mathrm{PFP}_{\overline{x},X}(X\overline{x} \vee \varphi)]\overline{t}$.

Transitive Closure

- Let R be a binary relation on a set M, $R \subseteq M^2$.
- The **transitive closure** $\text{TC}(R)$ of R is defined by

$$\text{TC}(R) := \{(a,b) \in M^2 : \text{there exist } n > 0 \text{ and } e_0, \ldots, e_n \in M, \text{ such that } a = e_0, b = e_n, \text{ and, for all } i < n, (e_i, e_{i+1}) \in R\}.$$

- The **deterministic transitive closure** $\text{DTC}(R)$ is defined by

$$\text{DTC}(R) := \{(a,b) \in M^2 : \text{there exist } n > 0 \text{ and } e_0, \ldots, e_n \in M, \text{ such that } a = e_0, b = e_n, \text{ and, for all } i < n, e_{i+1} \text{ is the unique } e \text{ for which } (e_i, e) \in R\}.$$

Transitive Closure Logic

- **Transitive Closure Logic** FO(TC) is obtained by closing FO under the transitive closure of definable relations.
- For a vocabulary τ, the class FO(TC)[τ] of **formulas of** FO(TC)[τ] **of vocabulary** τ is given by the following clauses:
  - φ, where φ is an atomic first-order formula over τ;
  - If φ, ψ are in FO(TC)[τ], then
    $$\neg\varphi, \quad (\varphi \vee \psi), \quad \exists x \varphi$$
    are in FO(TC)[τ];
  - If φ is in FO(TC)[τ], then
    $$[\mathrm{TC}_{\overline{x},\overline{y}}\varphi]\overline{s}\overline{t},$$
    is a formula in FO(TC)[τ], where:
    - the variables in $\overline{x}\ \overline{y}$ are pairwise distinct;
    - the tuples $\overline{x}, \overline{y}, \overline{s}$ and $\overline{t}$ are all of the same length, $\overline{s}$ and $\overline{t}$ being tuples of terms.

Deterministic Transitive Closure Logic

- **Deterministic Transitive Closure Logic** FO(DTC) is obtained by closing FO under the deterministic transitive closure of definable relations.
- For a vocabulary τ, the class FO(DTC)[τ] of **formulas of** FO(DTC)[τ] **of vocabulary** τ is given by the following clauses:
  - φ, where φ is an atomic first-order formula over τ;
  - If φ, ψ are in FO(DTC)[τ], then
    $$\neg\varphi, \quad (\varphi \vee \psi), \quad \exists x \varphi$$
    are in FO(DTC)[τ];
  - If φ is in FO(DTC)[τ], then
    $$[\mathrm{DTC}_{\overline{x},\overline{y}}\varphi]\overline{s}\overline{t},$$
    is a formula in FO(DTC)[τ], where:
    - The variables in $\overline{x}\ \overline{y}$ are pairwise distinct;
    - The tuples $\overline{x}, \overline{y}, \overline{s}$ and $\overline{t}$ are all of the same length, $\overline{s}$ and $\overline{t}$ being tuples of terms.

Sentences of (Deterministic) Transitive Closure Logic

- **Sentences** are formulas without free variables.
- The free occurrence of variables is defined in the standard way.
- For FO(TC) one adds the clause

$$\text{free}([\text{TC}_{\overline{x},\overline{y}}\varphi]\overline{s}\overline{t}) := \text{free}(\overline{s}) \cup \text{free}(\overline{t}) \cup (\text{free}(\varphi) \backslash \{\overline{x},\overline{y}\}).$$

- Similarly, for FO(DTC) one adds the clause

$$\text{free}([\text{DTC}_{\overline{x},\overline{y}}\varphi]\overline{s}\overline{t}) := \text{free}(\overline{s}) \cup \text{free}(\overline{t}) \cup (\text{free}(\varphi) \backslash \{\overline{x},\overline{y}\}).$$

Semantics of Transitive Closure Logic

- The semantics is defined inductively on the structure of formulas.
- We consider $\{(\overline{x},\overline{y}):\varphi(\overline{x},\overline{y},\overline{u})\})$ as a binary relation on the set of length$(\overline{x})$-tuples of the universe.
- The meaning of $[\mathrm{TC}_{\overline{x},\overline{y}}\varphi(\overline{x},\overline{y},\overline{u})]\overline{s}\overline{t}$ is

$$(\overline{s},\overline{t})\in \mathrm{TC}(\{(\overline{x},\overline{y}):\varphi(\overline{x},\overline{y},\overline{u})\}).$$

- The meaning of $[\mathrm{DTC}_{\overline{x},\overline{y}}\varphi(\overline{x},\overline{y},\overline{u})]\overline{s}\overline{t}$ is

$$(\overline{s},\overline{t})\in \mathrm{DTC}(\{(\overline{x},\overline{y}):\varphi(\overline{x},\overline{y},\overline{u})\}).$$

Example (Graphs)

- Let τ be the vocabulary of graphs.

  Consider the sentence of $\text{FO(TC)}[\tau]$

  $$\forall x \forall y(\neg x = y \to [\text{TC}_{x,y} Exy]xy).$$

  A graph is connected if it is a model of this sentence.

Relation Between Transitive Closures and Fixed Points

Proposition

(a) FO(DTC) $\leq$ FO(TC).

(b) FO(TC) $\leq$ FO(IFP).

(a) The statement follows from the equivalence

$$\vDash_{\text{fin}} [\text{DTC}_{\overline{x},\overline{y}}\varphi(\overline{x},\overline{y},\overline{u})]\overline{s}\overline{t}$$
$$\leftrightarrow [\text{TC}_{\overline{x},\overline{y}}(\varphi(\overline{x},\overline{y},\overline{u}) \wedge \forall\overline{z}(\varphi(\overline{x},\overline{z},\overline{u}) \rightarrow \overline{z}=\overline{y}))]\overline{s}\overline{t}.$$

(b) The statement follows from the equivalence

$$\vDash_{\text{fin}} [\text{TC}_{\overline{x},\overline{y}}\varphi(\overline{x},\overline{y},\overline{u})]\overline{s}\overline{t}$$
$$\leftrightarrow [\text{IFP}_{\overline{x},\overline{y},X}(\varphi(\overline{x},\overline{y},\overline{u}) \vee \exists\overline{v}(X\overline{x}\overline{v} \wedge \varphi(\overline{v},\overline{y},\overline{u})))]\overline{s}\overline{t}.$$

Axiomatizability and Restriction to Large Structures

Definition

Let K be a class of τ-structures and $\mathcal{L}$ a logic. K is **axiomatizable** in $\mathcal{L}$, if there is a sentence of $\mathcal{L}$ of vocabulary τ such that $K = \text{Mod}(\varphi)$.

- When relating logics and complexity classes it may be convenient to restrict to sufficiently large structures.
- We show that this restriction does not affect problems of axiomatizability.

Axiomatizability and Large Structures (Cont'd)

- For a class K of structures and $m \geq 1$, denote by K_m the subclass of K of structures of cardinality $\geq m$,

$$K_m := \{\mathcal{A} : \mathcal{A} \in K, \|A\| \geq m\}.$$

- For every finite structure $\mathcal{A}$, there is a sentence $\varphi_{\mathcal{A}}$ of FO characterizing $\mathcal{A}$ up to isomorphism, i.e., for all $\mathcal{B}$,

$$\mathcal{B} \vDash \varphi_{\mathcal{A}} \quad \text{iff} \quad \mathcal{B} \cong \mathcal{A}.$$

- Hence, for any logic $\mathcal{L}$ with $\text{FO} \leq \mathcal{L}$,

$$K \text{ is axiomatizable in } \mathcal{L} \text{ iff } K_m \text{ is axiomatizable in } \mathcal{L}.$$

In fact, set

$$\varphi_m := \bigvee \{\varphi_{\mathcal{A}} : \mathcal{A} \in K, \|A\| < m\}.$$

Then we have:
  - $K = \text{Mod}(\varphi)$ implies $K_m = \text{Mod}(\varphi \wedge \neg \varphi_m)$;
  - $K_m = \text{Mod}(\psi)$ implies $K = \text{Mod}(\psi \vee \varphi_m)$.

Subsection 2

Turing Machines and Complexity Classes

Turing Machines: Symbols and Tape

- We fix a finite alphabet $\mathbb{A}$.
- A **Turing machine** M is a finite device that performs operations on a tape which is bounded to the left and unbounded to the right and divided into **squares** (or **cells**).
- The machine operates stepwise, each step leading from one situation to a new one.
- In any situation every square of the tape either contains a single symbol from $\mathbb{A}$ or is blank.
- In the latter case we say that it contains the symbol "blank".
- There is one exception: the leftmost or "virtual" cell always contains an endmark, the "virtual" letter α (not in $\mathbb{A}$).

Turing Machines: Head and States

- M has a **read-and-write head** which scans a single square of the tape.
- In any step of a computation:
  - It erases or replaces the scanned symbol by another one;
  - It moves one cell to the left or to the right or remains at its place.
- In every situation, M is in one of the states of a finite set $\text{State}(M)$, the **set of states** of M.
- $\text{State}(M)$ contains:
  - A special state s_0, the **initial state**;
  - A special states s_+, the **accepting state**;
  - A special state s_-, the **rejecting state**.

  We assume that s_0, s_+ and s_- are pairwise distinct.

Turing Machine: Program

- The action or behavior of M in a situation depends on the current state of M and on the symbol currently being scanned by the head.
- It is given by $\text{Instr}(M)$, the **set of instructions** of M.
- Each instruction has the form $sa \to s'bh$, where:
  - $s, s' \in \text{State}(M)$, $s \neq s_+$, $s \neq s_-$;
  - $a, b \in \mathbb{A} \cup \{\alpha, \text{blank}\}$ and ($a = \alpha$ iff $b = \alpha$);
  - $h \in \{-1, 0, 1\}$, and if $a = \alpha$ then $h \neq -1$.
- The instruction above has the following meaning:

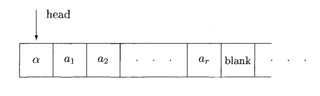
  If you are in state s and your head scans a cell with symbol a:
  - Replace a by b;
  - Move the head one cell to the left ($h = -1$), or to the right ($h = 1$), or stay put ($h = 0$);
  - Change to state s'.

Determinism and Nondeterminism

- A machine M is **deterministic** if for all $s \in S$ and $a \in \mathbb{A} \cup \{\alpha, \text{blank}\}$ there is at most one instruction of the form
$$sa \to s'bh$$
in $\text{Instr}(M)$.
- In order to emphasize that a machine is not required to be deterministic we sometimes call it **nondeterministic**.

Acceptance and Rejection

- Denote by $\mathbb{A}^*$ the set of words over $\mathbb{A}$.
- Denote by $\mathbb{A}^+$ the set of nonempty words over $\mathbb{A}$.
- Let $u \in \mathbb{A}^*$, $u = a_1 \dots a_r$ with $a_i \in \mathbb{A}$.
- M is **started** with u if M begins a computation (or **run**) in state s_0 in the situation shown on the right.



- The computation proceeds stepwise, each step corresponding to the execution of one instruction of M.
- The machine stops in a state s scanning a symbol $a \in \mathbb{A} \cup \{a, \text{blank}\}$, if there is no instruction of the form $as \to s'bh$ in $\text{Instr}(M)$.
  - If $s = s^+$ we speak of an **accepting run**;
  - If $s = s^-$ we speak of a **rejecting run**.
- M **accepts** u if there is at least one accepting run started with u.
- M **rejects** u if all runs started with u are finite and rejecting.

Decidability and Acceptability

- Subsets of $\mathbb{A}^+$ are called **languages**.
- A language $L \subseteq \mathbb{A}^+$ is **accepted** by M if, for all $u \in \mathbb{A}^+$,

$$M \text{ accepts } u \text{ iff } u \in L.$$

- L is **decided** by M if, in addition,

$$M \text{ rejects } u \text{ iff } u \notin L.$$

- Clearly, if M decides L then M accepts L.
- L is said to be **decidable** if it is decided by some deterministic Turing machine.
- L is said to be **acceptable** or **enumerable** if it is accepted by some nondeterministic Turing machine.

Time and Space Bounds

- Consider a function $f : \mathbb{N} \to \mathbb{N}$.
- We say that M is f **time-bounded**, if for all $u \in \mathbb{A}^+$ accepted by M, there is an accepting run of M started with u which has length at most $f(|u|)$ (recall that $|u|$ denotes the length of the word u).
- M is f **space-bounded**, if, for all $w \in \mathbb{A}^+$ accepted by M, there is an accepting run which uses at most $f(|u|)$ cells before stopping.

Polynomial Time and Space Bounds

- Denote by $\mathbb{N}[x]$ the set of polynomials with coefficients from $\mathbb{N}$.
- A language $L \subseteq \mathbb{A}^+$ is in PTIME ("polynomial time") if it is accepted by a deterministic machine that is p time-bounded, for some polynomial $p \in \mathbb{N}[x]$.
- A language $L \subseteq \mathbb{A}^+$ is in PSPACE ("polynomial space") if it is accepted by a deterministic machine that is p space-bounded, for some polynomial $p \in \mathbb{N}[x]$.
- The classes NPTIME ("nondeterministic polynomial time") and NPSPACE ("nondeterministic polynomial space") are defined similarly, now allowing nondeterministic machines.

Relations Between Classes

- Immediately from the definitions one gets

$$\text{PTIME} \subseteq \text{NPTIME} \quad \text{and} \quad \text{PTIME} \subseteq \text{PSPACE} \subseteq \text{NPSPACE}.$$

- One can show that

$$\text{NPTIME} \subseteq \text{PSPACE} \quad \text{and} \quad \text{PSPACE} = \text{NPSPACE}.$$

- Hence,

$$\text{PTIME} \subseteq \text{NPTIME} \subseteq \text{PSPACE}(= \text{NPSPACE}).$$

Example

- Let $\mathbb{A} = \{a, b\}$ and

$$L := \{u \in \mathbb{A}^+ : u \text{ contains an even number of } a\text{'s}\}.$$

We can easily design a machine accepting L and time-bounded by the polynomial $x + 2$.

The head just runs over the string, the state being "even" or "odd" depending on whether the number of a's already scanned is even or odd, respectively.

Essentially, this machine does not need any "working space" but only the space for the input.

Logarithmic Space

- For some reasons, it is convenient to separate the input from the working space and to introduce machines with:
  - An **input tape**;
  - A **work tape**.
- This allows, e.g., to measure only the working space.
- Measuring only the working space is critical in introducing complexity classes like LOGSPACE and NLOGSPACE, where the working space needed is smaller than the input space.

Robustness of the Turing Machine Model

- For other purposes it might be useful to introduce several **input tapes** and several **work tapes**.
- Also (if calculating a function, for example) we might use one or more **output tapes**, with a head for each tape, where the heads can move independently of each other.
- It turns out that the definition of the *usual* complexity classes does not depend on the number of tapes or on other peculiarities such as the form of the tape, i.e., whether it is unbounded to both sides or not.
- We often use this robustness, choosing, for example, the number of input and work tapes according to needs and to convenience.

Subsection 3

Trahtenbrot's Theorem

Trahtenbrot's Theorem

- Fix an alphabet $\mathbb{A}$.
- Undecidability of the Halting Problem: It is not decidable whether a deterministic Turing machine M accepts the empty word, i.e., whether M **halts**.
- We use this result to construct a vocabulary $\sigma(\mathbb{A})$ and show

Theorem (Trahtenbrot's Theorem)

Finite Satisfiability is not decidable, that is, the set

$$\text{Sat}[\sigma(\mathbb{A})] := \{\varphi : \varphi \text{ is a sentence of FO}[\sigma(\mathbb{A})] \text{ satisfiable in the finite}\}$$

is not decidable.

- We assign, in an effective way, to every deterministic machine M over $\mathbb{A}$ a sentence φ_M of $\text{FO}[\sigma(\mathbb{A})]$ such that

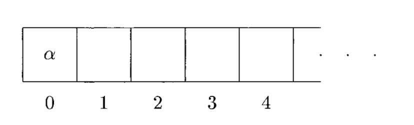
$$\varphi_M \text{ is satisfiable in the finite iff } M \text{ halts.}$$

Then use the undecidability of the Halting Problem.

Proof (Machine and Configurations)

- Without loss of generality we restrict ourselves to deterministic Turing machines M with the following features:
  - The set of states is an initial segment $\{0, \ldots, s_M\}$ of the natural numbers;
  - $s_0 := 0$ is the initial state;
  - 1 is the accepting state;
  - The machine stops only in the accepting state.

  We number the cells of the tape as indicated by



  In particular, the number 0 is given to the virtual cell.

Proof (Configurations)

- Suppose the machine M makes at least n steps.

  The n-th configuration C_n contains the following data:
  - The state;
  - The number of the cell scanned by the head;
  - The tape inscription after n steps.

Proof (Vocabulary)

- Let $\tau_0 = \{<, S, \min, \max\}$ be the vocabulary for orderings.

  In the following we sometimes write 0 instead of min.

  In addition to the symbols of τ_0 the vocabulary $\sigma(\mathbb{A})$ contains:
  - A binary relation symbol State;
  - A binary relation symbol Head;
  - For every $a \in \mathbb{A} \cup \{a, \text{blank}\}$, a binary relation symbol Letter_a.

Proof (Structure)

- Recall that $\{0,\ldots,s_M\}$ is the set of states of M.

  For every $n \geq s_M$, we define a structure $\mathcal{A}_n$, reflecting:
  - The initial segment $C_0,\ldots,C_n$ of the computation of M started with the empty word;
  - Only $C_0,\ldots,C_k$, with $k < n$, if M stops after k steps.

  The structure $\mathcal{A}_n$ has:
  - Universe $\{0,\ldots,n\}$, reflecting $C_0,\ldots,C_n$;
  - For $s,t \leq n$,

$$\begin{array}{rcl} \text{State}^{A_n}st & \text{iff} & \text{according to } C_t \text{ the state is } s; \\ \text{Head}^{A_n}it & \text{iff} & \text{according to } C_t \text{ the head is in cell } i; \\ \text{Letter}_a^{A_n}it & \text{iff} & \text{according to } C_t \text{ the letter } a \text{ is in cell } i. \end{array}$$

Proof (The Sentence φ_M)

- We construct next the sentence φ_M of $\mathrm{FO}[\sigma(\mathbb{A})]$ satisfying the following Properties (a) and (b).
  (a) If M, started with the empty word, stops after k steps (in the accepting state) and $n \geq s_M, k$, then $\mathcal{A}_n \vDash \varphi_M$;
  (b) If $\mathcal{A}$ is a finite model of φ_M and M, started with the empty word, runs at least k steps, then $\|A\| \geq k$.

  Properties (a) and (b) immediately give the equivalence

  $$\varphi_M \text{ is satisfiable in the finite iff } M \text{ halts.}$$

  As φ_M we take the conjunction of:
  - The $\{<, S, \min, \max\}$-ordering axioms;
  - The conjunction of the sentences in the following Clauses (1)-(4) (where we write 0 for min, 1 for the successor of min, etc.)

Proof (Sentences (1)-(4))

(1) "The universe has at least $s_M + 1$ elements."

(2) $\text{State}00 \wedge \text{Head}00 \wedge \text{Letter}_\alpha 00 \wedge \forall x(\neg x = 0 \to \text{Letter}_{\text{blank}} x0)$
(at time 0 the state is 0, the head scans the virtual cell, the virtual cell contains α, and all other cells are empty).

(3) For each instruction $sa \to s'bh$ a conjunct $\varphi_{sa\to s'bh}$ which describes the changes due to this instruction.

For example, if $h = 0$, then $\varphi_{sa\to s'bh}$ is the sentence

$$\begin{aligned}&\forall y \forall t((\text{State}st \wedge \text{Head}yt \wedge \text{Letter}_\alpha yt)\\ &\to \exists t'(Stt' \wedge \text{State}s't' \wedge \text{Head}yt' \wedge \text{Letter}_b yt'\\ &\wedge \forall v(\neg v = y \to \textstyle\bigwedge_{a\in\mathbb{A}\cup\{\alpha,\text{blank}\}}(\text{Letter}_a vt \to \text{Letter}_a vt')))).\end{aligned}$$

Similarly for sentences $\varphi_{sa\to s'bh}$, with $h = -1$ or $h = 1$.

(4) $\exists t \text{State}1t$ (the accepting state is reached).

Undecidability of FO in the Finite

- Coding $\sigma(\mathbb{A})$-structures by graphs, one obtains from the undecidability of $\text{Sat}[\sigma(\mathbb{A})]$ that for a binary relation symbol E the set

$$\text{Graph-Sat} := \{\varphi : \varphi \text{ is an FO}[\{E\}]\text{-sentence satisfiable in a finite graph}\}$$

  is not decidable.
- It follows that, for any vocabulary τ containing an at least binary relation symbol, the set

$$\text{Sat}[\tau] := \{\varphi : \varphi \text{ is an FO}[\tau]\text{-sentence satisfiable in the finite}\}$$

  is not decidable.

Enumerability of SAT

- There is a decision procedure that, given:
  - A finite structure $\mathcal{A}$ whose universe is a set of natural numbers;
  - A first-order sentence φ,

  checks whether $\mathcal{A} \vDash \varphi$.
- This decision procedure can be used to enumerate $\mathrm{Sat}[\tau]$.

  This follows from the fact that φ is satisfiable in the finite iff there is a model $\mathcal{A}$ of φ with $A = \{0, \ldots, n\}$ for some n.

Non-Enumerability of FO in the Finite

Theorem

If τ contains an at least binary relation symbol then the set

$$\mathsf{Val}[\tau] := \{\varphi \in \mathsf{FO}[\tau] : \varphi \text{ is a sentence valid in all finite structures}\}$$

of sentences valid in all finite structures is not enumerable.

- Suppose, to the contrary that $\mathsf{Val}[\tau]$ is enumerable.
  For any sentence φ we have
  $$\varphi \notin \mathsf{Sat}[\tau] \quad \text{iff} \quad \neg\varphi \in \mathsf{Val}[\tau].$$
  By hypothesis and this equivalence, $\mathsf{FO}[\tau] \backslash \mathsf{Sat}[\tau]$ is enumerable.
  Now consider the following procedure.
  > Given a sentence φ, start enumeration procedures for $\mathsf{Sat}[\tau]$ and $\mathsf{FO}[\tau] \backslash \mathsf{Sat}[\tau]$ until one of them yields φ.

  This is a decision procedure for $\mathsf{Sat}[\tau]$.
  This contradicts Trahtenbrot's Theorem.

Non-Existence of Proof Calculus in the Finite

Corollary

There is no complete proof calculus for FO in the finite.

- Suppose to the contrary.

  Then we can effectively enumerate all possible formal proofs.

  Hence, we can enumerate the sentences valid in the finite.

  This contradicts the preceding theorem.

Subsection 4

Structures as Inputs

Ordered Structures

Definition

Let $\{<\} \subseteq \tau_0 \subseteq \{<, S, \min, \max\}$.
Let $\tau_0 \subseteq \tau$.
A τ-structure $\mathcal{A}$ is **ordered** if the reduct $\mathcal{A}|_{\tau_0}$ is an ordering, i.e.:

- $<^A$ is an ordering;
- S, if present, is interpreted by the successor relation;
- min and max, if present, are interpreted as the least and the last element of the ordering, respectively.

$\mathcal{O}[\tau]$ is the class of ordered τ-structures.
If ψ is a sentence in the vocabulary τ, $\text{ordMod}(\psi)$ denotes the class of ordered models of ψ. Equivalently, $\text{ordMod}(\psi) = \text{Mod}(\psi \wedge \psi_0)$, where ψ_0 is the conjunction of the ordering axioms for the vocabulary τ_0.

Ordered Structures: Conventions

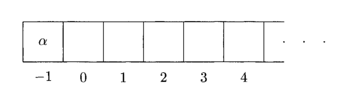
- Let $\mathcal{A} \in \mathcal{O}[\tau]$ be an ordered structure with $\|A\| = n$.
- By passing to an isomorphic copy we will assume that:
  - $A = \{0, \ldots, n-1\}$;
  - $<^A$ is the natural ordering on this set.
    I.e., we identify or "label" the least element of $<^A$ in $\mathcal{A}$ with 0, its successor with 1, etc.
- We assume that $\tau = \tau_0 \uplus \tau_1$, with, say,

$$\tau_1 = \{R_1, \ldots, R_k, c_1, \ldots, c_\ell\}.$$

  - τ_0 as in the preceding definition;
  - When writing τ_1 in this way, we tacitly assume that the symbols in τ_1 are given in the order $R_1, \ldots, R_k, c_1, \ldots, c_\ell$.

Turing Machine for Structures

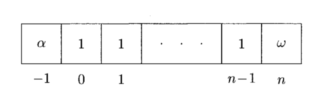
- A **Turing machine for τ-structures** will have $1 + k + \ell$ input tapes and m work tapes for some $m \geq 1$.
- All tapes are bounded to the left and unbounded to the right.
- Their cells are numbered as indicated in



- The "virtual" cell is numbered by -1 and always contains α.
- All input tapes will contain an input word followed by the virtual letter ω indicating the end of the input word.
- Each tape has its own head.
  - The heads can move independently of each other.
  - Those on input tapes are read-only heads.
  - Those on the work tapes are read-and-write heads.
- The alphabet only contains the symbol "1".
- We identify "0" with "blank".

Contents of the Input Tapes

- With an ordered τ-structure $\mathcal{A}$ we associate the following input inscriptions on the $1 + k + \ell$ input tapes (numbered from 0 to $k + \ell$).
- The 0-th tape, the "universe tape", contains a sequence of 1's of length $n := \|A\|$.



- For $1 \leq i \leq k$, the i-th input tape contains the information about $R := R_i$ coded as follows:
  - If R is r-ary, then $R^A \subseteq \{0, \ldots, n-1\}^r$.
  - Since $\|\{0, \ldots, n-1\}^r\| = n^r$, for $j < n^r$, the j-th cell will contain "1" in case the j-th r-tuple in the lexicographic ordering of $\{0, \ldots, n-1\}^r$ is in R.

Contents of the Input Tapes (Cont'd)

- For $j < n^r$, denote by $\lfloor j \rfloor_r$ be the j-th r-tuple in the lexicographic ordering of $\{0, \ldots, n-1\}^r$.
  - Consider the unique n-adic representation of j,

$$j = j_1 \cdot n^{r-1} + j_2 \cdot n^{r-2} + \cdots + j_{r-1} \cdot n + j_r, \quad 0 \leq j_i < n;$$

  - Set $\lfloor j \rfloor_r := (j_1, \ldots, j_r)$.
- To make n explicit, we sometimes write $\lfloor j \rfloor_r^n$ instead of $\lfloor j \rfloor_r$.
- Then the i-th input tape has the inscription

α	a_0	a_1	a_2	a_3	$\cdot\ \cdot\ \cdot$	a_{n^r-1}	ω
-1	0	1	2	3		n^r-1	n^r

  where $a_j = 1$ iff $R^A \lfloor j \rfloor_r$ (equivalently, $a_j = 0$ iff not $R^A \lfloor j \rfloor_r$).
- For $1 \leq i \leq \ell$, the $(k+i)$-th input tape contains the binary representation of $j := c_i^A$ without leading zeros.

Starting Configuration and States

- We say that a Turing machine M is **started with $\mathcal{A}$**, if:
  - The input tapes contain the information on $\mathcal{A}$ in the way just described;
  - The work tapes are empty;
  - Each head scans the cell numbered 0 of its tape.
- As in the case of one-tape machines:
  - M has a finite set $\text{State}(M)$ of **states**;
  - M has a finite set $\text{Instr}(M)$ of **instructions**.
- $\text{State}(M)$ contains:
  - An **initial** (or **starting**) state s_0;
  - An **accepting state** s_+;
  - A **rejecting state** s_-.

Instructions

- **Instructions** now have the form

$$sb_0 \dots b_{k+\ell} c_1 \dots c_m \to s' c'_1 \dots c'_m h_0 \dots h_{k+\ell+m}.$$

- This instruction has the following meaning.

  If:
  - You are in state s;
  - Your heads scan $b_0, \dots, b_{k+\ell}$ on the input tapes;
  - Your heads scan $c_1, \dots, c_m$ on the work tapes;

  Then:
  - Replace $c_1, \dots, c_m$ by $c'_1, \dots, c'_m$;
  - Move the i-th head according to h_i;
  - Change to state s'.

Instructions (Cont'd)

- In the instruction

$$sb_0 \dots b_{k+\ell} c_1 \dots c_m \to s' c'_1 \dots c'_m h_0 \dots h_{k+\ell+m}$$

we have:

  - $s, s' \in \mathrm{State}(M)$;
  - $b_0, \dots, b_{k+\ell} \in \{0, 1, \alpha, \omega\}$;
  - $c_1, \dots, c_m, c'_1, \dots, c'_m \in \{0, 1, \alpha\}$;
  - $h_0, \dots, h_{k+\ell+m} \in \{-1, 0, 1\}$.

Moreover:

  - if $b_j = \alpha$ then $h_j \neq -1$; (if the head scans the leftmost square it cannot move left)
  - if $b_j = \omega$ then $h_j \neq 1$;
  - if $c_j = \alpha$ then $h_{k+\ell+j} \neq -1$ and $c'_j = \alpha$;
  - if $c_j \in \{0, 1\}$ then $c'_j \in \{0, 1\}$;
  - $s \neq s_+$ and $s \neq s_-$.

Instructions (Some Terminology)

- The **base** of the instruction

$$sb_0 \dots b_{k+\ell} c_1 \dots c_m \to s' c'_1 \dots c'_m h_0 \dots h_{k+\ell+m}$$

  is given by $sb_0 \dots b_{k+\ell} c_1 \dots c_m$.
- M is said to be **deterministic** if no two distinct instructions in $\mathrm{Instr}(M)$ have the same base.
- Sometimes, to emphasize that we do not require a machine to be deterministic, we speak of a **nondeterministic machine**.

Acceptance and Rejection

- The notions of **accepting run**, **rejecting run**, and of "M accepts $\mathcal{A}$" are adapted from the preceding section in the obvious way.
- Let K he a class of ordered τ-structures.
- We say that M **accepts** K if M accepts exactly those ordered τ-structures that lie in K.

Time and Space Bounds

- Let $f : \mathbb{N} \to \mathbb{N}$ be a function.
- We say that M is f **time-bounded** if, for any $\mathcal{A}$ accepted by M, there is an accepting run of M, started with $\mathcal{A}$, of length at most $f(\|A\|)$.
- M is f **space-bounded** if, for all $\mathcal{A}$ accepted by M, there is an accepting run which uses at most $f(\|A\|)$ squares on each work tape before stopping.

Polynomial Time and Space

- A class K of structures is in PTIME ("polynomial time") iff, there is a deterministic machine M and a polynomial $p \in \mathbb{N}[x]$, such that M accepts K and M is p time-bounded.
- A class K of structures is in NPTIME ("nondeterministic polynomial time") iff, there is a nondeterministic machine M and a polynomial $p \in \mathbb{N}[x]$, such that M accepts K and M is p time-bounded.
- A class K of structures is in PSPACE ("polynomial space") iff, there is a deterministic machine M and a polynomial $p \in \mathbb{N}[x]$, such that M accepts K and M is p space-bounded.
- A class K of structures is in NPSPACE ("nondeterministic polynomial space") iff, there is a nondeterministic machine M and a polynomial $p \in \mathbb{N}[x]$, such that M accepts K and M is p space-bounded.

Logarithmic Space

- Denote by $\log n$ the least natural number $\geq \log_2 n$.
- A class K of structures is in LOGSPACE ("logarithmic space") iff, there is a deterministic machine M and $d \geq 1$, such that M accepts K and is $d \cdot \log$ space-bounded.
- A class K of structures is in NLOGSPACE, "nondeterministic logarithmic space" iff there is a nondeterministic machine M and $d \geq 1$ such that M accepts K and is $d \cdot \log$ space-bounded.

Restriction to Monic Polynomials

- We have observed that for a class K of ordered structures and $m \geq 1$ the class $K_m := \{\mathcal{A} \in K : \|A\| \geq m\}$ is axiomatizable in a logic $\mathcal{L}$ iff K is axiomatizable in $\mathcal{L}$.
- Analogously, for any of the complexity classes $\mathcal{C}$ introduced so far, we have

$$K \in \mathcal{C} \quad \text{iff} \quad K_m \in \mathcal{C}.$$

  This is a consequence of the following fact.

  > We can change a machine, without essentially affecting its time and space bounds, in such a way that it runs on a given finite set of inputs in a prescribed form.

Restriction to Monic Polynomials

- We show that we can restrict ourselves to monic polynomials $p(x) = x^d$ when considering PSPACE (or PTIME, NPTIME).

  Suppose that K is in PSPACE.

  Then K_2 is in PSPACE, too.

  Suppose K_2 is accepted by a machine M that is q space-bounded, where

  $$q(x) = a_s x^s + a_{s-1} x^{s-1} + \cdots + a_1 x + a_0.$$

  For suitable d,

  $$q(n) \leq n^d, \quad \text{for all } n \geq 2.$$

  Thus, M is x^d space-bounded.

Subsection 5

Logical Descriptions of Computations

Logic and Complexity Theory

- Let K be a class of ordered τ-structures, $K \subseteq \mathcal{O}[\tau]$.
- We write $K \in \mathrm{IFP}$ if K is axiomatizable in FO(IFP).
- We use similar notations for the other logics.
- Our main goal is to show

$$\begin{array}{rcl} K \in \mathrm{LOGSPACE} & \text{iff} & K \in \mathrm{DTC} \\ K \in \mathrm{NLOGSPACE} & \text{iff} & K \in \mathrm{TC} \\ K \in \mathrm{PTIME} & \text{iff} & K \in \mathrm{IFP} \\ K \in \mathrm{NPTIME} & \text{iff} & K \in \Sigma_1^1 \\ K \in \mathrm{PSPACE} & \text{iff} & K \in \mathrm{PFP}. \end{array}$$

  (Σ_1^1 denotes the fragment of second-order logic consisting of the sentences of the form $\exists X_1 \cdots \exists X_m \psi$, where ψ is first-order).
- These results provide the bridge between logic and complexity theory.
- In this section we prove the implications from left to right and in the next the converse implications.

The Proof Strategy

- Let $\mathcal{C}$ be one of the complexity classes listed above.
- Let $\mathcal{L}$ be the logic associated to $\mathcal{C}$ by the corresponding equivalence.
- Assume that $K \in \mathcal{C}$.
- Let M be a Turing machine witnessing that $K \in \mathcal{C}$.
- We are going to describe the behavior of M by a formula φ_M of $\mathcal{L}$ in such a way that for any ordered structure $\mathcal{A}$,

$$\mathcal{A} \vDash \varphi_M \quad \text{iff} \quad M \text{ accepts } \mathcal{A}.$$

- This will yield $K = \text{ordMod}(\varphi_M)$.

Vocabulary and Machines

- We fix a vocabulary
$$\tau = \tau_0 \uplus \tau_1,$$
where, for simplicity, we assume that:
  - $\tau_0 = \{<, S, \min, \max\}$;
  - τ_1 is relational,
$$\tau_1 = \{R_1, \ldots, R_k\},$$
with r_i-ary R_i.
- For convenience we set $r_0 = 1$.
- A Turing machine M for τ-structures has:
  - $1 + k$ input tapes;
  - A certain number m of work tapes.

Configurations

- A **configuration** of a Turing machine M started with a structure contains the following data:
  - The current state;
  - The current inscriptions of the work tapes;
  - The current position of the heads on both the input and the work tapes.
- An **accepting configuration** is a configuration with state s_+.
- A configuration CONF' is a **successor** of the configuration CONF, if an instruction of M allows M to go from CONF to CONF' in one step.
- An accepting configuration is viewed as a successor of itself.
- If M is deterministic, every configuration has at most one successor.

Assumptions on the Size of $\mathcal{A}$

- Let M be a (nondeterministic) Turing machine for τ-structures which is x^d space bounded.
- So, if M accepts an ordered structure $\mathcal{A}$, then there is an accepting run that scans at most n^d squares on each work tape, where $n := \|A\|$.
- We may assume that $r_i < d$ for $i = 1, \ldots, k$ (r_i being the arity of R_i).
- Fix a structure $\mathcal{A}$.
- When proving that a class of structures is axiomatizable in a logic or acceptable by a Turing machine of a certain complexity bound, we can restrict ourselves to sufficiently large finite structures.
- Here we look at structures $\mathcal{A}$ such that, for $n := \|A\|$:
  - $n > k + m$;
  - $n > \|\text{State}(M)\|$.
- We assume that $\text{State}(M)$, the set of states of M, is an initial segment of the natural numbers and that $s_0 = 0$ is the starting state.

Dependence of Configuration on Size of Structure

- Let CONF be a configuration, where at most the n^d first cells of each work tape are not empty and where the heads scan one of these cells.
- A first attempt to code the contents of these cells could consist in dividing the relevant part of each work tape into $\frac{n^d}{\log n} =: r$ blocks of length $\log n$ and reading each block as a natural number $< n$ in binary representation.
- This would require variables $x_1, \ldots, x_r$ for each tape.
- Then a formula bearing the information on successive configurations would contain at least the variables $x_1, \ldots, x_r$.
- So it would depend on the cardinality n of the universe.
- We overcome this difficulty for PTIME, NPTIME and PSPACE by using relation variables instead of individual variables.

Encoding the Data of a Configuration

- The "state relation" $\mathrm{ST}^{\mathrm{CONF}}$ is defined by

$$\mathrm{ST}^{\mathrm{CONF}} := \{s\},$$

where s is the state of CONF;
- The "end-of-tape relations" $\mathrm{E}_j^{\mathrm{CONF}}$ are defined, for $0 \leq j \leq k + m$, by

$$\mathrm{E}_j^{\mathrm{CONF}} = \begin{cases} \{0\}, & \text{if the } j\text{-th head faces } \alpha \\ \{n-1\}, & \text{if the } j\text{-th head faces } \omega \\ \varnothing, & \text{otherwise} \end{cases};$$

- The "head relation" $\mathrm{H}_j^{\mathrm{CONF}}$ is defined, for $0 \leq j \leq k$, as the r_j-ary relation

$$\mathrm{H}_j^{\mathrm{CONF}} := \{|e|_{r_j} : 0 \leq e, \text{the } j\text{-th head scans the } e\text{-th square and this does not contain } \omega\};$$

Encoding the Data of a Configuration (Cont'd)

- The "head relation" H_j^{CONF} is defined , for $k+1 \leq j \leq k+m$, as the d-ary relations
$$H_j^{CONF} := \{|e|_d : 0 \leq e, \text{the } j\text{-th head scans the } e\text{-th square}\};$$
- The "inscription relations" I_j^{CONF} are defined, for $k+1 \leq j \leq k+m$, as the d-ary relations
$$I_j^{CONF} : = \{|e|_d : 0 \leq e < n^d \text{ and the } e\text{-th square of the } j\text{-th work tape contains the symbol } 1\}.$$
- Note that the latter are only introduced for the work tapes, since the inscriptions of input tapes are given by the input structure and kept fixed during the whole computation.
- Obviously, CONF is uniquely determined by the preceding relations.

Example

- The starting configuration CONF_0 is given by

$$\begin{array}{rcl} \mathsf{ST}^{\mathsf{CONF}_0} & = & \{0\}, \\ \mathsf{E}^{\mathsf{CONF}_0} & = & \varnothing, \\ \mathsf{H}^{\mathsf{CONF}_0} & = & \{(0,\ldots,0)\}, \\ \mathsf{I}_j^{\mathsf{CONF}_0} & = & \varnothing. \end{array}$$

The Relation C^{CONF}

- For technical convenience we encode CONF in a single $(d+2)$-ary relation $C^{\mathrm{CONF}} \subseteq \{0,\ldots,n-1\}^{d+2}$ by joining the preceding relations.
- We add two first coordinates to distinguish these relations and fill up with zeroes in the middle to get arity $(d+2)$.
- We denote by $\widetilde{0}$ the constant sequences $0\ldots0$ of appropriate length.

$$\begin{array}{rcl} C^{\mathrm{CONF}} & := & \{(0,0)\} \times \{\widetilde{0}\} \times \mathrm{ST}^{\mathrm{CONF}} \\ & & \cup \bigcup_{0\le j\le k+m}\{(1,j)\} \times \{\widetilde{0}\} \times \mathrm{E}_j^{\mathrm{CONF}} \\ & & \cup \bigcup_{0\le j\le k+m}\{(2,j)\} \times \{\widetilde{0}\} \times \mathrm{H}_j^{\mathrm{CONF}} \\ & & \cup \bigcup_{k+1\le j\le k+m}\{(3,j)\} \times \mathrm{I}_J^{\mathrm{CONF}}. \end{array}$$

- Clearly, given $C \subseteq \{0,\ldots,n-1\}^{d+2}$, we can easily decide whether there is a configuration CONF of M, where only the first n^d cells of each work tape are relevant, such that $C = C^{\mathrm{CONF}}$.
- We call such a configuration C an n^d-**bounded configuration**.

Formulas for Starting and Successor Configurations

Lemma

Let M be a Turing machine which is x^d space-hounded. There is a first-order formula $\varphi_{\text{start}}(\overline{x})$ and there are first-order formulas $\varphi_{\text{succ}}(\overline{x}, X)$ and $\psi_{\text{succ}}(X, Y)$ (more precisely, second-order formulas without second-order quantifiers) such that for all sufficiently large $\mathcal{A} \in \mathcal{O}[\tau]$ and $\overline{a} \in A^{d+2}$ we have:

(a) "$\varphi_{\text{start}}(\overline{x})$ describes the starting configuration": If C_0 denotes the starting configuration of M started with A then $\mathcal{A} \vDash \varphi_{\text{start}}[\overline{a}]$ iff $\overline{a} \in C_0$.

(b) "$\varphi_{\text{succ}}(\overline{x}, X)$ describes the successor of X": If M is deterministic and C is an n^d-bounded configuration of M (where $n := \|A\|$) then $\mathcal{A} \vDash \varphi_{\text{succ}}[\overline{a}, C]$ iff C has an n^d-bounded successor C' and $\overline{a} \in C'$.

(c) "$\psi_{\text{succ}}(X, Y)$ expresses that Y is a successor of X": If C_1 is an n^d-bounded configuration of M and C_2 a further $(d+2)$-ary relation on A then $\mathcal{A} \vDash \psi_{\text{succ}}[C_1, C_2]$ iff C_2 is an n^d-hounded configuration of M which is a successor of C_1.

Proof of the Lemma Introduction

- Let M be an x^d space-bounded machine for τ-structures.

  Recall the encoding of an n^d-bounded configuration CONF in a single relation C^{CONF} comprising relations:
  - ST^{CONF}, containing the information on the state;
  - E_j^{CONF}, containing the information on the endmarks;
  - H_j^{CONF}, containing the information on the head positions;
  - I_j^{CONF}, containing the information on the inscription of the work tapes.

  Let $\overline{x}$ be the sequence of variables $xyx_1 \ldots x_d$.

Proof of the Lemma Part (a)

(a) We can set

$$\varphi_{\text{start}}(\overline{x}) := \overline{x} = \widetilde{0} \vee (x = 2 \wedge 0 \leq y \leq k + m \wedge x_1 \ldots x_d = \widetilde{0}).$$

This formula asserts that

"the state is s_0" or "the heads scan the 0-th cell".

Note that, in writing this formula, we used:

- $0 \leq y \leq k + m$ to mean that y is equal to or less than the $(k + m)$-th element in the ordering $<$;
- 0 to stand for min.

In the following we shall use similar self-explanatory abbreviations.

Proof of the Lemma Parts (b) and (c)

- For parts (b) and (c) we consider instructions $\text{instr} \in \text{Instr}(M)$ of the form

$$\text{instr} = sb_0 \dots b_{k+1} c_1 \dots c_m \to s' c'_1 \dots c'_m h_0 \dots h_{k+\ell+m}.$$

  For every such instruction, we first introduce:
  - A formula $\varphi_{\text{instr}}(\overline{x}, X)$ which for n^d-bounded configurations X expresses "X has base $s\overline{b}\overline{c}$, and:
    - If the successor configuration according to instr is not n^d-bounded, then $\{\overline{x} : \varphi_{\text{instr}}(\overline{x}, X)\} = \varnothing$;
    - Otherwise, $\{\overline{x} : \varphi_{\text{instr}}(\overline{x}, X)\}$ is this successor configuration".
  - A formula $\varphi_{\text{acc}}(X)$ which for n^d-bounded configurations X expresses "X is an accepting configuration".

Proof of the Lemma Parts (b) and (c) (Cont'd)

- Now, recall that we agreed to set $C_{m+1} = C_m$ for accepting configurations C_m.

  Using $\varphi_{\text{instr}}(\overline{x}, X)$ and $\varphi_{\text{acc}}(X)$, we get the desired formulas φ_{succ} and ψ_{succ} of Parts (b) and (c).

$$\varphi_{\text{succ}}(\overline{x}, X) := (\varphi_{\text{acc}}(X) \wedge X\overline{x}) \vee \bigvee_{\text{insrt} \in \text{Instr(M)}} \varphi_{\text{instr}}(\overline{x}, X);$$

$$\psi_{\text{succ}}(X, Y) := (\varphi_{\text{acc}}(X) \wedge \forall \overline{x}(Y\overline{x} \leftrightarrow X\overline{x})) \vee \bigvee_{\text{instr} \in \text{Instr}(M)} (\exists \overline{x} \varphi_{\text{instr}}(\overline{x}, X) \wedge \forall \overline{x}(Y\overline{x} \leftrightarrow \varphi_{\text{instr}}(\overline{x}, X))).$$

Proof of the Lemma ($\varphi_{\text{instr}}(\overline{x}, X)$ and $\varphi_{\text{acc}}(X)$)

- It remains to give $\varphi_{\text{instr}}(\overline{x}, X)$ and $\varphi_{\text{acc}}(X)$.
  We set
  $$\varphi_{\text{acc}}(X) := X00\widetilde{0}s_+.$$
  The formula $\varphi_{\text{instr}}(\overline{x}, X)$ has the form
  $$\varphi_{\text{instr}}(\overline{x}, X) := \varphi_{a,\overline{b},\overline{c}}(X) \wedge \varphi_{s',\overline{c'},\overline{h}}(\overline{x}, X),$$
  where, for an n^d-bounded configuration X:
  - The formula
    $$\varphi_{s,\overline{b},\overline{c}}(X)$$
    expresses "X has base $s, \overline{b}, \overline{c}$";
  - The formula
    $$\varphi_{s',\overline{c}',\overline{h}}(\overline{x}, X)$$
    expresses
    "if the successor Y of X according to $s', \overline{c}', \overline{h}$ is not n^d-bounded then $\{x : \varphi_{s',\overline{c}',\overline{h}}(\overline{x}, X)\} = \emptyset$, else $\{\overline{x} : \varphi_{s',\overline{c}',\overline{h}}(\overline{x}, X)\} = Y$".

Proof of the Lemma (Notation)

- For easier reading of the formulas below we introduce the following abbreviations:
  - The formula
$$\text{ENDMARK}yz := X1y\widetilde{0}z,$$
    expressing that "the y-th head faces the endmark z";
  - The formula
$$\text{HEAD}y\overline{z} := X2y\widetilde{0}\overline{z},$$
    expressing that "the y-th head is on position $|\overline{z}|$";
  - The formula
$$\text{ONE}y\overline{z} := X3y\overline{z},$$
    expressing that "the y-th work tape contains 1 on position $|\overline{z}|$".

Proof of the Lemma ($\varphi_{s,\overline{b},\overline{c}}(X)$)

- We take as $\varphi_{s,\overline{b},\overline{c}}(X)$ the conjunction of the following formulas:
  - $X00\widetilde{0}s$,
    "s is the state";
  - $\bigwedge_{b_j=\alpha} \text{ENDMARK} j \min \wedge \bigwedge_{c_j=\alpha} \text{ENDMARK}(k+j) \min$,
    "heads at the left end of a tape";
  - $\bigwedge_{b_j=\omega} \text{ENDMARK} j \max$,
    "heads at the right end on input tapes";
  - $\bigwedge_{b_j=1} \exists x_1 \cdots \exists x_{r_j} (\text{HEAD} j \widetilde{0} x_1 \ldots x_{r_j} \wedge R_j x_1 \ldots x_{r_j})$,
    "heads of input tapes facing a 1";
  - $\bigwedge_{b_j=0} \exists x_1 \cdots \exists x_{r_j} (\text{HEAD} j \widetilde{0} x_1 \ldots x_{r_j} \wedge \neg R_j x_1 \ldots x_{r_j})$,
    "heads of input tapes facing a 0";
  - $\bigwedge_{c_j=1} \exists x_1 \cdots \exists x_d (\text{HEAD}(k+j) x_1 \ldots x_d \wedge \text{ONE}(k+j) x_1 \ldots x_d)$,
    "heads of work tapes facing a 1";
  - $\bigwedge_{c_j=0} \exists x_1 \cdots \exists x_d (\text{HEAD}(k+j) x_1 \cdots x_d \wedge \neg \text{ONE}(k+j) x_1 \ldots x_d)$,
    "heads of work tapes facing a 0".

Proof of the Lemma ($\varphi_{s',\overline{c}',\overline{h}}(\overline{x}, X)$)

- Finally, we take as $\varphi_{s',\overline{c}',\overline{h}}(\overline{x}, X)$ the conjunction $(\varphi_1 \wedge \varphi_2)$:
- φ_1 is $\bigwedge_{\substack{h_j=1 \\ k+1\leq j\leq k+m}} \neg \mathrm{HEAD}j\widetilde{\max}$,

  "heads of work tapes moving to the right do not face the $(n^d - 1)$-th square".
- φ_2 is the disjunction of the following formulas:
  - $(x = y = 0 \wedge x_1 \ldots x_{d-1} = \widetilde{0} \wedge x_d = s')$,
    "s' is the new state";
  - $\bigvee_{k+1\leq j\leq k+m}(\neg \mathrm{HEAD}jx_1 \ldots x_d \wedge \mathrm{ONE}jx_1 \ldots x_d \wedge x = 3 \wedge y = j)$,
    "work tape content unchanged on squares not scanned";
  - $\bigvee_{c_j'=1}(\mathrm{HEAD}jx_1 \ldots x_d \wedge x = 3 \wedge y = k + j)$,
    "new content 1 on scanned squares of a work tape";
  - $\bigvee_{h_j=1}(\mathrm{ENDMARK}j0 \wedge x = 2 \wedge y = j \wedge x_1 \ldots x_d = \widetilde{0})$,
    "heads scanning α and moving to the right come to position 0";
  - $\bigvee_{h_j=0}(\mathrm{ENDMARK}j0 \wedge x = 1 \wedge y = j \wedge x_1 \ldots x_d = \widetilde{0})$,
    "unchanged position of heads facing α";

Proof of the Lemma ($\varphi_{s',\overline{c}',\overline{h}}(\overline{x}, X)$ Cont'd)

- List of φ_2 disjuncts continued:
  - $\bigvee_{h_j=-1}(\text{ENDMARK}j\max \wedge x = 2 \wedge y = j$
    $\wedge\, x_1 \ldots x_{d-r_j} = \widetilde{0} \wedge x_{d-r_j+1} \ldots x_d = \widetilde{\max})$,
    "heads scanning ω and moving to the left come to position $n^{r_j} - 1$";
  - $\bigvee_{h_j=0}(\text{ENDMARK}j\max \wedge x = 1 \wedge y = j \wedge x_1 \ldots x_{d-1} = \widetilde{0} \wedge x_d = \max)$,
    "unchanged position of heads facing ω";
  - $\bigvee_{h_j=-1}(\text{HEAD}j\widetilde{0} \wedge x = 1 \wedge y = j \wedge x_1 \ldots x_d = \widetilde{0})$,
    "heads scanning α from their new position";
  - $\bigvee_{\substack{j\leq k\\ h_j=1}}(\text{HEAD}j\widetilde{0}\underbrace{\max\ldots\max}_{r_j\text{-times}} \wedge x = 1 \wedge y = j \wedge x_1 \ldots X_{d-1} = \widetilde{0} \wedge x_d = \max)$,
    "heads of input tapes scanning ω from their new position";
  - $\bigvee_{j\leq k+m} \exists u_1 \cdots \exists u_d($"$x_1 \ldots x_d = u_1 \ldots u_d + h''_j$
    $\wedge\, \text{HEAD}ju_1 \ldots u_d \wedge x = 2 \wedge y = j)$,
    "new head position of heads on "interior" squares".

PSPACE and FO(PFP)

Theorem

Let $K \subseteq \mathcal{O}[\tau]$ be a class of ordered structures. If K is in PSPACE then K is axiomatizable in FO(PFP).

- Let M be a deterministic machine witnessing $K \in$ PSPACE.
  By previous remarks, assume M is x^d space bounded for some d.
  We set

  $$\varphi(\overline{x}, X) := (\neg\exists\overline{y}X\overline{y} \wedge \varphi_{\text{start}}(\overline{x})) \vee (\exists\overline{y}X\overline{y} \wedge \varphi_{\text{succ}}(\overline{x}, X)),$$

  where φ_{start} and φ_{succ} are the formulas in the preceding lemma.
  Let $\mathcal{A}$ be an ordered structure and $n := \|A\|$.
  By the lemma, F_0^φ, F_1^φ, F_2^φ, ... is the sequence $\varnothing, C_0, C_1, \ldots$ where:
  - C_0 is the starting configuration;
  - If C_i is an n^d-bounded configuration of M with an n^d-bounded successor configuration C then $C_{i+1} = C$.
    In particular, if C_i is accepting then $C_i = C_{i+1} = C_{i+2} = \cdots$.

PSPACE and FO(PFP) (Cont'd)

- If C_i is an n^d-bounded configuration without a successor configuration or with a successor configuration which is not n^d-bounded, then
  $C_{i+1} = 0,\ C_{i+2} = C_0,\ C_{i+3} = C_1,\ \ldots.$
  So the sequence has no fixed-point.

Summarizing, we have

$$\begin{array}{lll} M \text{ accepts } \mathcal{A} & \text{iff} & F_\infty^\varphi \text{ is an accepting configuration} \\ & \text{iff} & F_\infty^\varphi \text{ is a configuration with state } s_+. \end{array}$$

"F_∞^φ is a configuration with state s_+" is expressed by the formula

$$\exists y (\text{"}y \text{ is the } s_+\text{-th element of } <\text{"} \wedge [\mathrm{PFP}_{\overline{x},X_\varphi}] \min \min \widetilde{\min} y).$$

We abbreviate it by $[\mathrm{PFP}_{\overline{x}X_\varphi}] \min \min \widetilde{\min} s_+$.

Then, $\mathcal{A} \in K$ iff M accepts $\mathcal{A}$ iff $\mathcal{A} \vDash [\mathrm{PFP}_{\overline{x}X_\varphi}] \min \min \widetilde{\min} s_+$.

I.e., $K = \mathrm{ordMod}([\mathrm{PFP}_{\overline{x},X_\varphi}] \min \min \widetilde{\min} s_+)$.

PSPACE and Single Occurrence of PFP

Corollary

Let $K \subseteq \mathcal{O}[\tau]$ be in PSPACE. Then K is axiomatizable by a sentence of FO(PFP) with only one occurrence of PFP.

- Immediate from the proof of the preceding theorem.

PSPACE to PTIME: PFP to IFP

- For PTIME on the logical side we therefore can replace PFP by IFP.
- Consider a (finite or infinite) run $C_0, C_1, \ldots$ of an x^d time-bounded (and hence, x^d space-bounded) deterministic machine started with a structure of cardinality n.
- If the run accepts the structure, C_{n^d-1} must be an accepting configuration.
- The inflationary process indicated above is given by a formula $\varphi(\overline{v}, \overline{x}, Z)$ with:
  - $F_i^{(Z\overline{v}\overline{x} \vee \varphi)} = \bigcup_{\substack{m<i \\ C_m \text{ defined}}} \{|m|_d\} \times C_m$;
  - $F_\infty^{(Z\overline{v}\overline{x} \vee \varphi)} = \bigcup_{\substack{m<n^d \\ C_m \text{ defined}}} \{|m|_d\} \times C_m$.

  That is, we use the first d coordinates as time stamps when coding the run in one relation (as above, $|m|_d$ denotes the m-th d-tuple in $\{0, \ldots, n-1\}^d$ in the lexicographic ordering).

PTIME and FO(IFP)

Theorem

Let $K \subseteq \mathcal{O}[\tau]$ be a class of ordered structures. If K is in PTIME then K is axiomatizable in FO(IFP).

- Let M be a deterministic machine witnessing $K \in$ PTIME.

  We can assume that, for suitable d, M is x^d time-bounded.

  For $\overline{v} = v_0 \dots v_{d-1}$ we set

  $$\varphi(\overline{v}, \overline{x}, Z) := (\overline{v} = \widetilde{\min} \wedge \varphi_{\text{start}}(\overline{x})) \vee \exists \overline{u}(S^d \overline{uv} \wedge \varphi_{\text{succ}}(\overline{x}, Z\overline{u}\_)),$$

  where:
  - $\overline{v} = \widetilde{\min}$ abbreviates $v_0 = \min \wedge \cdots \wedge v_{d-1} = \min$;
  - $S^d \overline{uv}$ stands for "$\overline{v}$ is the successor of $\overline{u}$ in the lexicographic ordering";
  - $\varphi_{\text{succ}}(\overline{x}, Z\overline{u}\_)$ is obtained from $\varphi_{\text{succ}}(\overline{x}, X)$ by replacing subformulas $X\overline{t}$ by $Z\overline{u}\overline{t}$.

PTIME and FO(IFP) (Cont'd)

- Then we have, for $\mathcal{A} \in \mathcal{O}[\tau]$, with $n := \|A\|$,

$$\begin{array}{lll} \mathcal{A} \in K & \text{iff} & M \text{ accepts } \mathcal{A} \\ & \text{iff} & \text{the } (n^d - 1)\text{-th configuration of } M\text{, started} \\ & & \text{with } \mathcal{A}\text{, is defined and has state } s_+ \\ & \text{iff} & \mathcal{A} \vDash [\mathrm{IFP}_{\overline{vx},Z}\varphi]\widetilde{\max} \min \min \widetilde{\min} s_+. \end{array}$$

That is, K is the class of ordered models of a sentence of FO(IFP).

Corollary

Let $K \subseteq \mathcal{O}[\tau]$ be in PTIME. Then K is axiomatizable in FO(IFP) by a sentence with only one occurrence of IFP.

NPTIME and Second Order Logic

Theorem

Let $K \subseteq \mathcal{O}[\tau]$ be a class of ordered structures. If K is in NPTIME then K is axiomatizable in SO by a Σ_1^1-sentence.

- Choose M witnessing $K \in$ NPTIME.

  Assume that M is x^d time bounded.

  Then, for $\mathcal{A} \in \mathcal{O}[\tau]$ with $n := \|A\|$,

  $\mathcal{A} \in K$ iff there is a run of M, started with $\mathcal{A}$, of length $\leq n^d$ that accepts $\mathcal{A}$

  iff there is a sequence $C_0, \ldots, C_{n^d-1}$ of n^d-bounded configurations of M, started with $\mathcal{A}$, such that C_0 is the starting configuration, C_{i+1} is a successor configuration of C_i and s_+ is the state of C_{n^d-1}.

NPTIME and Second Order Logic (Cont'd)

- Equivalently,

$$\mathcal{A} \vDash \varphi,$$

where

$$\begin{aligned}\varphi \quad := \quad & \exists Z(\forall \overline{x}(Z\widetilde{\min}\overline{x} \leftrightarrow \varphi_{\text{start}}(\overline{x})) \\ & \wedge \forall \overline{u}\forall \overline{v}(S^d\overline{uv} \to \psi_{\text{succ}}(Z\overline{u}_{-}, Z\overline{v}_{-})) \\ & \wedge Z\widetilde{\max}\min\min\widetilde{\min s_{+}}).\end{aligned}$$

Here, the intended meaning of Z is $\bigcup_{m<n^d}\{|m|_d\} \times C_m$.

LOGSPACE and NLOGSPACE: Configuration Description

- $K \in$ NLOGSPACE means that there is a (nondeterministic) machine M and some $d \geq 1$ such that:
  - M accepts K;
  - M is $d \cdot \log$ space bounded.
- Every natural number $i < n$ codes a word over $\{0, 1\}$ of length $\log n$, namely, its binary representation $|i|^2_{\log n}$ of length $\log n$.
- Thus, using d variables, we can represent the relevant contents of a work tape.
- Moreover, by restricting ourselves to sufficiently large structures $\mathcal{A}$, we can assume that $d \cdot \log n < n$ (where $n := \|A\|$).
- Hence, each head position can be represented by a single number $< n$.
- Altogether, we can describe the data of a configuration by a sequence of natural numbers $< n$ of length independent of n, where we agree to use the first number to represent the state.

Existence of Formulas for Start and Successor

Lemma

Let M be $d \cdot \log$ space bounded. Then there are formulas $\chi_{\text{start}}(\overline{x})$ of FO and $\chi_{\text{succ}}(\overline{x}, \overline{x}')$ of FO(DTC), such that, for all sufficiently large $\mathcal{A} \in \mathcal{O}[\tau]$ and $\overline{a}$ in A,

(a) $\mathcal{A} \vDash \chi_{\text{start}}[\overline{a}]$ iff $\overline{a}$ is the (description of the) starting configuration;

(b) For any $(d \cdot \log \|A\|)$-bounded configuration $\overline{a}$ and any $\overline{b}$,

$$\mathcal{A} \vDash \chi_{\text{succ}}[\overline{a}, \overline{b}] \quad \text{iff} \quad \overline{b} \text{ is a } (d \cdot \log \|A\|)\text{-bounded successor configuration of } \overline{a}.$$

- Before we give a proof we derive some consequences.

LOGSPACE and FO(DTC)

Theorem

Let $K \subseteq \mathcal{O}[\tau]$ be a class of ordered structures. If $K \in$ LOGSPACE then K is axiomatizable in FO(DTC).

- Let M be a deterministic machine witnessing $K \in$ LOGSPACE.

  Suppose M is $d \cdot \log$ space bounded.

  Let χ_{start} and χ_{succ} be the formulas of the lemma.

  By Parts (a) and (b) of the lemma, we have for $\mathcal{A} \in \mathcal{O}[\tau]$,

  M accepts $\mathcal{A}$

  iff there is a sequence $\overline{a}_0, \ldots, \overline{a}_k$ of $(d \cdot \log \|A\|)$-bounded configs such that $\overline{a}_0$ is the starting configuration, $\overline{a}_{i+1}$ is the successor configuration of $\overline{a}_i$, and $\overline{a}_k$ is an accepting configuration

  iff $\mathcal{A} \vDash \exists \overline{x}(\chi_{\text{start}}(\overline{x}) \wedge \exists \overline{x}'([\text{DTC}_{\overline{x},\overline{x}'} \chi_{\text{succ}}(\overline{x}, \overline{x}')]\overline{x}, \overline{x}' \wedge x_1' = s_+))$.

  Hence, K is the class of ordered models of a sentence of FO(DTC).

NLOGSPACE and FO(TC)

Theorem

Let $K \subseteq \mathcal{O}[\tau]$ be a class of ordered structures. If $K \in$ NLOGSPACE then K is axiomatizable in FO(TC).

- Let M be a machine witnessing $K \in$ NLOGSPACE.

  Here, M is nondeterministic.

  So, we just have to replace DTC by TC in the last proof.

  In fact, we have for $\mathcal{A} \in \mathcal{O}[\tau]$,

  $$\begin{array}{ll} & M \text{ accepts } \mathcal{A} \\ \text{iff} & \mathcal{A} \vDash \exists \overline{x}(\chi_{\text{start}}(\overline{x}) \wedge \exists \overline{x}'([\text{TC}_{\overline{x},\overline{x}'}\chi_{\text{succ}}(\overline{x},\overline{x}')]\overline{x}\overline{x}' \wedge x_1' = s_+)). \end{array}$$

  Recall that, by the lemma, $\chi_{\text{start}}(\overline{x})$ and $\chi_{\text{succ}}(\overline{x},\overline{x}')$ are in FO(DTC).

  Since FO(DTC) $\leq$ FO(TC), we have obtained a sentence of FO(TC) axiomatizing K.

NLOGSPACE and FO(posTC)

- Denote by FO(posTC) the class of formulas of FO(TC) which only contain positive occurrences of TC.
- That is, in formulas of FO(posTC), each occurrence of TC is in the scope of an even number of negation symbols.
- It can be shown that FO(DTC) $\leq$ FO(posTC).
- Thus the preceding proof yields

Corollary

If a class of ordered structures is in NLOGSPACE then it is axiomatizable by a sentence of FO(posTC).

Formulas for Arithmetical Functions

Lemma

There are FO(DTC) formulas $\varphi_+(x,y,z)$, $\varphi_\cdot(x,y,z)$, $\varphi_2(x,y)$ and $\varphi_{\log(\text{universe})}(x)$, such that, for any ordered structure $\mathcal{A}$, with $A = \{0,\ldots,\|A\|-1\}$, and any $a,b,c \in A$:

$$\begin{array}{lll} \mathcal{A} \vDash \varphi_+[a,b,c] & \text{iff} & a+b=c; \\ \mathcal{A} \vDash \varphi_\cdot[a,b,c] & \text{iff} & a \cdot b = c; \\ \mathcal{A} \vDash \varphi_2[a,b] & \text{iff} & 2^a = b; \\ \mathcal{A} \vDash \varphi_{\log(\text{universe})}[a] & \text{iff} & a = \log\|A\|. \end{array}$$

- For better readability, instead of describing the natural numbers in terms of the ordering, we use constants

$$1, 2, \ldots.$$

Formulas for Arithmetical Functions (Cont'd)

- Given numbers x and y, consider the path

$$(0,x) \to (1,x+1) \to (2,x+2) \to \cdots \to (y,x+y)$$

from $(0,x)$ to $(y,x+y)$.

This path shows that as $\varphi_+(x,y,z)$ we can take the formula

$$(y = \min \wedge z = x) \vee [\mathrm{DTC}_{uv,u'v'}(Suu' \wedge Svv')] \min xyz.$$

Consider, next, the path

$$(0,0) \to (1,x) \to (2,2\cdot x) \to (3,3\cdot x) \to \cdots \to (y,y\cdot x).$$

This path shows that we can set

$$\begin{aligned} \varphi_{\cdot}(x,y,z) \quad := \quad & (y = \min \wedge z = \min) \\ & \vee [\mathrm{DTC}_{uv,u'v'}(Suu' \wedge \varphi_+(v,x,v'))] \min \min yz. \end{aligned}$$

Formulas for Arithmetical Functions (Cont'd)

- As before, consider the path

$$(0,1) \to (1,2) \to (2,4) \to \cdots \to (x, 2^x).$$

It shows that we can set

$$\begin{aligned} \varphi_2(x,y) \quad := \quad & (x = \min \wedge y = 1) \\ & \vee [\mathrm{DTC}_{uv,u'v'}(Suu' \wedge \varphi_{\cdot}(v, 2, v'))] \min 1xy. \end{aligned}$$

Finally, for $\varphi_{\log(\text{universe})}(x)$, we define

$$\varphi_{\log(\text{universe})}(x) := \neg \exists y \varphi_2(x,y) \wedge \forall z (z < x \to \exists y \varphi_2(z,y)).$$

Notation

- Let $\ell := \log n - 1$. Then $2^\ell < n$.
- Recall that for:
  - m, with $m < 2^\ell$;
  - $m_0, \ldots, m_{\ell-1} \in \{0,1\}$,

  we have

$$|m|_\ell^2 = m_0 \ldots m_{\ell-1} \quad \text{iff } m = m_0 \cdot 2^{\ell-1} + m_1 \cdot 2^{\ell-2} + \cdots + m_{\ell-2} \cdot 2 + m_{\ell-1}.$$

- We then say that m_k is the k**-th digit** of $|m|_\ell^2$.
- If n, and hence ℓ, is clear from the context, we denote $|m|_\ell^2$ by $[m]$.
- We write:
  - $\overline{u}$ for $u_0 \ldots u_d$;
  - $\overline{u} < 2^\ell$ for $u_0 < 2^\ell \wedge \cdots \wedge u_d < 2^\ell$.
- Similar conventions are used for $\overline{u}', \overline{x}$ and $\overline{x}'$.

Some Definable Relations

Lemma

There are formulas of FO(DTC) which, in ordered structures $\mathcal{A}$, define the following relations (where $n = \|A\|$, $\ell = \log n - 1$):

$\text{One}mk$	iff	$m < 2^\ell, k < \ell$, and the k-th digit of $[m]$ is 1;
$\text{Zero}mk$	iff	$m < 2^\ell, k < \ell$, and the k-th digit of $[m]$ is 0;
$\text{One}_d\overline{u}k$	iff	$\overline{u} < 2^\ell, k < (d+1)\cdot\ell$, and the k-th digit of the concatenation $[u_0]\ldots[u_d]$ is 1;
$\text{Zero}_d\overline{u}k$	iff	$\overline{u} < 2^\ell, k < (d+1)\cdot\ell$, and the k-th digit of the concatenation $[u_0]\ldots[u_d]$ is 0;
$\text{Equal}_d\overline{u}k\overline{u}'$	iff	$\overline{u}, \overline{u}' < 2^\ell, k < (d+1)\cdot\ell$, and the words $[u_0]\ldots[u_d]$ and $[u_0']\ldots[u_d']$ differ at most at the k-th position.

- We denote the corresponding formulas by $\varphi_{\text{one}}(x,z)$, $\varphi_{\text{zero}}(x,z)$, $\varphi_{\text{d-one}}(\overline{x},z)$, $\varphi_{\text{d-zero}}(\overline{x},z)$, and $\varphi_{\text{d-equal}}(\overline{x},z,\overline{x}')$, respectively.

Some Definable Relations (Proof)

- Note that, for $m < 2^\ell$ and $k < \ell$, the k-th digit of $[m]$ is 1 iff

$$\exists y \in \mathbb{N} \exists z \in \mathbb{N}(m = y \cdot 2^{\ell-k} + z \wedge 2^{\ell-k-1} \leq z < 2^{\ell-k}).$$

Using this one shows, e.g., the following equivalences:

$$\begin{array}{rll} \mathrm{One}mk & \text{iff} & m < 2^\ell \wedge k < \ell \wedge \\ & & \exists y \exists z(m = y \cdot 2^{\ell-k} + z \wedge 2^{\ell-k-1} \leq z < 2^{\ell-k}); \\ \mathrm{One}_1 u_0 u_1 k & \text{iff} & (u_0 < 2^\ell \wedge u_1 < 2^\ell \wedge k < \ell \wedge \mathrm{One}u_0 k) \vee \\ & & (\ell \leq k < 2 \cdot \ell \wedge \mathrm{One}u_1(k - \ell)); \\ \mathrm{Equal}_d \overline{u} k \overline{u}' & \text{iff} & \overline{u} < 2^\ell \wedge \overline{u}' < 2^\ell \wedge k < (d+1) \cdot \ell \wedge \\ & & \forall i((i < (d+1) \cdot \ell \wedge i \neq k) \rightarrow (\mathrm{One}_d \overline{u} i \leftrightarrow \mathrm{One}_d \overline{u}' i)). \end{array}$$

These can be formalized using the formulas of the preceding lemma.

Proof: Formulas for Start and Successor (Setup)

- Let M be a log space bounded machine for τ-structures.

  Suppose M is $d \cdot \log$ space bounded.

  For simplicity we assume that:
  - $\tau = \{R\}$, with binary R;
  - M only has one work tape.

  Recall that:
  - $\ell := \log n - 1$;
  - $\{0, \ldots, s_M\}$ is the set of states of M.

  We restrict ourselves to structures of cardinality n, with:
  - $n > d \cdot \log n$;
  - $(d + 1) \cdot \ell \geq d \cdot \log n$;
  - $n > s_M + 1$.

Proof: Formulas for Start and Successor (Configuration)

- When M is started with a structure $\mathcal{A}$, where $A = \{0, \ldots, n-1\}$, we can code the data of a resulting configuration by a tuple

$$(z, u_\alpha, u_\omega, u, v_\alpha, v_\omega, v_0, v_1, w_\alpha, w, y_0, \ldots, y_d)$$

where:

  - z is the state;
  - u_α, u_ω, u code the position of the head on the 0-th input tape (the "universe tape") as follows:

$$u_\alpha = \begin{cases} 0, & \text{if the head does not face } \alpha \\ n-1, & \text{if the head faces } \alpha \end{cases};$$

$$u_\omega = \begin{cases} 0, & \text{if the head does not face } \omega \\ n-1, & \text{if the head faces } \omega \end{cases};$$

    u is the number of the cell faced by the head, if it is an interior one; otherwise, $u = 0$.

Proof: Formulas for Start and Successor (Cont'd)

- Similarly, $v_\alpha, v_\omega, v_0, v_1$ code the position of the head on the first input tape, i.e., the tape for the binary relation R.
  Here, the variables v_0, v_1 represent the head position $v_0 \cdot n + v_1$.
- w_α, w code the position of the head on the work tape ($d \cdot \log n < n$).
- The concatenation $[y_0] \ldots [y_d]$ is the inscription of the first $(d+1) \cdot \ell$ cells of the work tape.

Sometimes, for notational simplicity, we write

$$\overline{x} := z u_\alpha u_\omega \ldots y_d.$$

Proof: Formulas for Start and Successor (Instructions)

- For Part (a) of the lemma, we can set

$$\chi_{\text{start}}(\overline{x}) := \overline{x} = \widetilde{0}.$$

- For Part (b), we define

$$\chi_{\text{succ}}(\overline{x}, \overline{x}') := \chi_{\text{acc}}(\overline{x}, \overline{x}') \vee \bigvee_{\text{instr} \in \text{Instr}(M)} \chi_{\text{instr}}(\overline{x}, \overline{x}'),$$

where:

  - $\chi_{\text{acc}}(\overline{x}, \overline{x}') := (x_1 = s_+ \wedge \overline{x}' = \overline{x})$,
    expressing, in case $\overline{x}$ is a configuration, that $\overline{x}$ is accepting and $\overline{x}' = \overline{x}$;
  - $\chi_{\text{instr}}(\overline{x}, \overline{x}')$, for an instruction

    $$\text{instr} = sb_0 b_1 c_1 \to s' c_1' h_0 h_1 h_2,$$

    is a formula which, in case $\overline{x}$ is a $(d \cdot \log)$-bounded configuration, expresses that:
    - $\overline{x}$ has base $sb_0 b_1 c_1$;
    - The successor configuration of $\overline{x}$ according to instr is $(d \cdot \log)$-bounded and is $\overline{x}'$.

Proof: Instructions (Example)

- We explicitly give $\chi_{\text{instr}}(\overline{x}, \overline{x}')$ for

$$\text{instr} = s1\alpha 1 \to s'0(-1)11.$$

  This is the conjunction of the following:
  - $z = s$,
    "s is the state";
  - $u_\alpha = \min \wedge u_\omega = \min$,
    "the head of the 0-th input tape faces an interior cell";
  - $v_\alpha = \max \wedge v_\omega = \min \wedge v_0 = \min \wedge v_1 = \min$,
    "the head of the first input tape faces α";
  - $w_\alpha = \min \wedge \text{One}_d y_0 \ldots y_d w$,
    "the head of the work tape faces a 1";

Proof: Instructions (Example Cont'd)

- We continue the list of conjuncts of $\chi_{\text{instr}}(\overline{x}, \overline{x}')$ for instr $= s1\alpha 1 \to s'0(-1)11$:
  - $z' = s'$,
    "s' is the new state";
  - $u'_\omega = \min \wedge ((u > 0 \wedge Su'u \wedge u'_\alpha = \min) \vee (u = 0 \wedge u' = 0 \wedge u'_\alpha = \max))$,
    "new head position of the 0-th input tape";
  - $v'_\alpha = \min \wedge v'_0 = \min \wedge v'_1 = \min \wedge v'_\omega = \min$,
    "new head position of the first input tape is cell 0";
  - $w'_\alpha = \min \wedge Sww' \wedge w' < d \cdot \log n$, i.e.,
    $w'_\alpha = \min \wedge Sww' \wedge \exists x(\varphi_{\log(\text{universe})}(x) \wedge w' < d \cdot x)$,
    "new head position of the work tape is within the bounds";
  - $\text{Zero}_d y'_0 \ldots y'_d w$,
    "new content of cell scanned on the work tape";
  - $\text{Equal}_d y_0 \ldots y_d w y'_0 \ldots y'_d$,
    "work tape content unchanged on cells not scanned".

Subsection 6

The Complexity of the Satisfaction Relation

Introduction

- Suppose, e.g., that the class K of ordered structures is axiomatizable by the FO(IFP)-sentence φ,
$$K = \{\mathcal{A} \in \mathcal{O}[\tau] : \mathcal{A} \vDash \varphi\}.$$
- We aim at showing that $K \in \mathrm{PTIME}$.
- I.e., for fixed φ, we want to prove that the satisfaction relation $\mathcal{A} \vDash \varphi$ can be decided in time polynomially bounded in $\|A\|$.
- One also says that φ has a polynomial time **model-checker**.

Algorithmic Manipulations

- The following manipulations in algorithms do not destroy polynomial time and logarithmic space bounds.
  (1) Using an additional work tape W', it is possible at any time of a computation to move the head of a given work tape W to the rightmost square which the head of the tape has scanned so far.
  (We change the given program so that the head on W' moves in the same way as the head on W, but always prints the symbol 1.)
  (2) By (1) it is possible at any time of a computation to erase the content of a work tape (note that the additional work tape used in (1) can be cleared in a trivial way).
  In particular, one can change a program - without changing the accepted class - such that all work tapes are empty whenever the program stops.

Algorithmic Manipulations (Cont'd)

(3) The content of a worktape W can be copied to an empty tape W_1. (Using (1), bring the corresponding heads H and H_1 to the rightmost cell scanned by H and copy the content cell by cell).

(4) In our applications the 0-th input tape has the inscription $\underbrace{1\ldots1}_{n \text{ digits}}$, where n is the cardinality of the structure we consider.
One can write the binary representation of n (of length $\le \log n$) on a work tape. We say "a counter is set to n".
Similarly, a counter can be set to n^d for any fixed $d \ge 1$.

The Goal

- Let $\mathcal{L}$ be one of the logics FO(DTC), FO(TC), FO(IFP), Σ_1^1, FO(PFP), considered in the preceding section.
- Let $\mathcal{C}$ the corresponding complexity class, i.e., one of LOGSPACE, NLOGSPACE, PTIME, NPTIME, PSPACE, respectively.
- We want to show that for any sentence of $\mathcal{L}$ the class K of its ordered models is in $\mathcal{C}$.
- We even show that there is a machine M **strongly witnessing** $K \in \mathcal{C}$, that is:
  - M accepts K;
  - For any $\mathcal{A} \in \mathcal{O}[\tau]$, every run of M, started with $\mathcal{A}$, stops at s_+ or s_-; In particular, if M is deterministic then M decides K;
  - For any $\mathcal{A} \in \mathcal{O}[\tau]$ every run of M satisfies the time or space bounds characteristic for $\mathcal{C}$.

Method of Proof

- The proof that the class of ordered models of a sentence φ of $\mathcal{L}$ is in $\mathcal{C}$ proceeds by induction on φ.
- In dealing with formulas, we introduce the following notation.
- For a formula $\varphi(x_1,\ldots,x_\ell,Y_1,\ldots,Y_r)$ we let

$$\mathrm{ordMod}(\varphi) := \{(\mathcal{A},a_1,\ldots,a_\ell,P_1,\ldots,P_r) : \mathcal{A}\in\mathcal{O}[\tau],\mathcal{A}\vDash\varphi[\overline{a},\overline{P}]\}.$$

- That is, we consider the ordered models of the sentence

$$\varphi(c_1,\ldots,c_\ell,P_1,\ldots,P_r)$$

in an enlarged vocabulary.

DTC, posTC, LOGSPACE and NLOGSPACE

Theorem

Let $K \subseteq \mathcal{O}[\tau]$ be a class of ordered structures.

(a) If $K \in$ DTC then $K \in$ LOGSPACE.

(b) If $K \in$ posTC then $K \in$ NLOGSPACE.

- By induction on the corresponding formulas, we show that:
  - The class of ordered models of φ is in LOGSPACE and NLOGSPACE, respectively;
  - There exists a machine strongly witnessing this fact.

  We handle both cases simultaneously.

  By passing to an equivalent formula, we can assume that, in formulas of FO(posTC), the TC operation does not occur in the scope of any negation symbol (the new formula may also contain $\wedge, \forall$).

Proof for Atomic Formulas

- Suppose that φ is atomic, say for simplicity, $\varphi = Rxy$.

  We show that there is a machine M strongly witnessing that

  $$\{(A, i, j) : \mathcal{A} \in \mathcal{O}[\tau], R^A ij\} \in \text{LOGSPACE}.$$

  Let $(\mathcal{A}, i, j) \in \mathcal{O}[\tau \cup \{c, d\}]$, with $A = \{0, 1, \dots, n-1\}$ be given.

  Note that the information whether $R^A ij$ holds is to be found in the $(i \cdot n + j)$-th square of the input tape corresponding to R.

  The binary representations of i and j are available on the input tapes corresponding to c and d.

  Now it should be clear how a machine strongly witnessing that $\text{OrdMod}(Rxy) \in \text{LOGSPACE}$ can be designed.

Proof for Boolean Connectives

- $\varphi = \neg\psi$: By the remarks above, ψ does not contain TC. Hence, ψ is in FO(DTC). By the induction hypothesis, there is a machine M strongly witnessing that $\text{ordMod}(\psi) \in \text{LOGSPACE}$.

  For φ just interchange the roles of s_+ and s_- in M.
- $\varphi(x_1, \ldots, x_\ell) = (\psi \vee \chi)$: By the induction hypothesis, there are machines M_ψ for $\psi(x_1, \ldots, x_\ell)$ and M_χ for $\chi(x_1, \ldots, x_\ell)$.

  Let M be a machine that:
  - Carries out the computation of M_ψ;
  - Erases the work tapes;
  - Carries out the computation of M_χ;
  - Accepts in case at least one, M_ψ or M_χ, accepts, and rejects otherwise.
- $\varphi = (\psi \wedge \chi)$: Similarly.

Proof for Quantifiers

- $\varphi(x_1, \ldots, x_\ell) = \exists x \psi$: By the induction hypothesis, there is a corresponding machine M_0 for $\psi(x_1, \ldots, x_\ell, x)$.

  A machine M for φ operates as follows: Suppose M is started with an ordered structure $(\mathcal{A}, a_1, \ldots, a_\ell)$, where $A = \{0, \ldots, n-1\}$.

  Then, for $i = 0, \ldots, n-1$,

  - M writes the binary representation of i on a work tape;
  - M checks, using M_0, whether $\mathcal{A} \vDash \psi[a_1, \ldots, a_\ell, i]$.

  If the answer is positive at least once, M stops in s_+, otherwise in s_-.

  Here, the binary representation of i on the work tape does not carry an endmark ω as required on the corresponding input tape of M_0.

  To detect the end of the representation of i, we use Technique (1).
- $\varphi = \forall x \psi$: Similarly.

Proof for DTC

- $\varphi = [\text{DTC}_{\overline{x},\overline{y}}\psi]\overline{s}\overline{t}$, where ψ is a formula of FO(DTC):

  For simplicity, we assume that the free variables of ψ are among $\overline{x},\overline{y}$ and that $\overline{x} = x, \overline{y} = y$, $\overline{s} = s$ and $\overline{t} = t$.

  Let M_0 be a machine strongly witnessing $\text{ordMod}(\psi) \in \text{LOGSPACE}$.

  Given $\mathcal{A}$, with $A = \{0,\ldots,n-1\}$, if there is a ψ-path from s to t, there is one of length $\leq n$.

  So the machine M, we aim at, can be organized as follows.

  - It writes $i := s$ on a work tape and sets a counter to n;
  - M rejects in case the counter becomes negative;
  - Using M_0, the subroutine checks for $j = 0,\ldots,n-1$ whether $\mathcal{A} \vDash \psi[i,j]$ holds for exactly one j;
    - If not, M rejects;
    - Otherwise, M checks whether j equals t.

      In the affirmative case M accepts;

      In the negative case, M sets $i := j$ and reduces the counter by one.

Proof for TC

- $\varphi = [\mathrm{TC}_{\overline{x},\overline{y}}\psi]\overline{s}\overline{t}$: Once more, for simplicity, we assume that the free variables of ψ are among $\overline{x},\overline{y}$ and that $\overline{x} = x, \overline{y} = y$, $\overline{s} = s$ and $\overline{t} = t$.

  Choose a machine M_0 strongly witnessing $\mathrm{ordMod}(\psi) \in \mathrm{NLOGSPACE}$.

  We give the basic idea underlying the construction of M for φ.

  Suppose given $\mathcal{A}$, with $A = \{0, 1, \ldots, n-1\}$.

  A counter is set to n and is used to carry out a subroutine at most n times. M will stop in s_- in case the counter becomes negative.

  M writes $i := s$ on a work tape.

  The subroutine chooses $j \in \{0, \ldots, n-1\}$ nondeterministically (it uses a counter to randomly write a $\{0,1\}$ word of length $\log n$ on a tape).

  It checks, using M_0, whether $\psi(i, j)$ holds.
  - If not, M stops in state s_-.
  - Otherwise, if $j = t$, M stops in s_+, and if $j \neq t$, it sets $i := j$.

IFP,PFP, PTIME and PSPACE

Theorem

Let $K \subseteq \mathcal{O}[\tau]$ be a class of ordered structures.

(a) If $K \in$ IFP then $K \in$ PTIME.

(b) If $K \in$ PFP, then $K \in$ PSPACE.

- The proof is by induction on the formula φ axiomatizing the class K.

  The cases where φ is atomic, $\neg\psi$, $(\psi \vee \chi)$ or $\exists x \psi$ are handled as in the preceding proof.

  The corresponding machines are polynomially time-bounded or space-bounded if the machines used in the induction hypotheses are.

Proof for IFP

- Suppose that $\varphi = [\mathrm{IFP}_{\overline{x},X}\psi(\overline{x},X)]\overline{t}$, where X is r-ary.

  For simplicity, assume that the free variables of ψ are among $\overline{x},X$.

  Let M_0 be a machine strongly witnessing that

  $$\{(\mathcal{A},\overline{a},R) : \mathcal{A} \in \mathcal{O}[\tau], \mathcal{A} \vDash \psi[\overline{a},R]\} \in \mathrm{PTIME}.$$

  The machine M has a subroutine that uses work tapes W and W'.
  - Suppose the subroutine is started with a word of length n^r on W, viewed as the code of an r-ary relation R, and an empty W'.
  - It writes, using M_0, the code of $R' := \{\overline{a} : \mathcal{A} \vDash (X\overline{x} \vee \psi)[\overline{a},R]\}$ on the tape W' without changing the content of W.

  The machine M for φ operates as follows:
  - It sets $R := \varnothing$ and uses the subroutine to calculate R'.
  - If $R = R'$ it checks whether $R\overline{t}$ or not $R\overline{t}$ and accepts or rejects.
  - Otherwise, it sets $R := R'$, erases W', and starts the subroutine.

  Note that $R = R'$ after at most n^d calls to the subroutine.

Proof for PFP

- Assume that $\varphi = [\text{PFP}_{\overline{x},X}\psi(\overline{x},X)]\overline{t}$, with r-ary X.

  Let M_0 be a machine strongly witnessing

  $$\{(\mathcal{A},\overline{a},R) : \mathcal{A} \in \mathcal{O}[\tau], \mathcal{A} \vDash \psi[\overline{a},R]\} \in \text{PSPACE}.$$

  Given a structure $\mathcal{A}$, the operation F^{ψ} satisfies one of:
  - $F^{\psi}_{2^{n^r}-1} = F^{\psi}_{2^{n^r}}$ (and this set is the fixed-point F^{ψ}_{∞});
  - $F^{\psi}_{\infty} = \varnothing$.

  The machine M for φ, on input $\mathcal{A}$, works as follows:
  - It sets a counter to $2^r - 1$ (writes $1\ldots1$ of length n^r on a work tape).
  - Then it proceeds as in the IFP case, but now using the counter to ensure that the subroutine which here evaluates

    $$R' := \{\overline{a} : \mathcal{A} \vDash \psi[\overline{a},R]\},$$

    is invoked at most 2^{n^r} times.
  - When the counter gets negative, it checks whether $R = R'$ and Rt.
  - If both hold it accepts, otherwise it rejects.

Σ_1^1, SO, NPTIME and PSPACE

Theorem

Let $K \subseteq \mathcal{O}[\tau]$ be a class of ordered structures.

(a) If $K \in \Sigma_1^1$ then $K \in$ NPTIME.

(b) If $K \in$ SO then $K \in$ PSPACE.

(a) Let $K = \text{Mod}(\varphi)$ where $\varphi = \exists X_1 \cdots \exists X_\ell \psi$, ψ is first-order, and the arity of X_i is r_i. By a previous theorem, there is a machine M_0 strongly witnessing that $\text{Mod}(\psi(X_1, \ldots, X_\ell))$ is in (LOGSPACE $\subseteq$) PTIME.

The machine M for φ, on input $\mathcal{A} \in \mathcal{O}[\tau]$, works as follows:

- It nondeterministically writes words over $\{0, 1\}$ of length $n^{r_1}, \ldots, n^{r_\ell}$ on different work tapes, which are intended as codes of interpretations $P_1, \ldots, P_\ell$ of $X_1, \ldots, X_\ell$.
- Using M_0, it checks whether $\mathcal{A} \models \psi[P_1, \ldots, P_\ell]$ or not.
- It stops in an accepting or rejecting state, respectively.

Σ_1^1, SO, NPTIME and PSPACE (Part (b))

(b) Let $K = \text{Mod}(\varphi)$ for a formula φ of SO.

To gain a machine M witnessing $K \in \text{PSPACE}$ we proceed by induction on φ.

For φ atomic or of the form $\neg\psi$, $(\psi \vee \chi)$ or $\exists x \psi$ we argue as in the proof of the theorem for DTC.

For $\varphi = \exists X \psi$ with r-ary X the machine M:

- Writes the word $1 \ldots 1$ of length n^r on a work tape W.
- It systematically decreases this word, checking in each case with a polynomially space-bounded machine for ψ, whether ψ holds, if the interpretation of X is given by the tape W.

Subsection 7

The Main Theorem and Some Consequences

Logics and Complexity Classes

Definition

A logic $\mathcal{L}$ **captures** a complexity class $\mathcal{C}$ if, for all τ, with $< \in \tau$, and $K \subseteq \mathcal{O}[\tau]$, we have

$$K \in \mathcal{C} \quad \text{iff} \quad K \text{ is axiomatizable in } \mathcal{L}.$$

Summary of Results

Theorem (Main Theorem)

(a) FO(DTC) captures LOGSPACE.
(b) FO(TC) captures NLOGSPACE.
(c) FO(IFP) captures PTIME.
(d) Σ_1^1 captures NPTIME.
(e) FO(PFP) captures PSPACE.

- Note that we have proved the theorem except for Part (b).
- We only have shown that FO(posTC) captures NLOGSPACE.
- At the end of the section we show that FO(posTC) = FO(TC) on ordered structures.
- This will complete the proof.

Types of Complexity

- The study of the complexity of evaluating a formula φ of a logic $\mathcal{L}$ in a structure $\mathcal{A}$ arises in various contexts.
  - $\mathcal{A}$ may be a database instance and φ a corresponding query;
  - $\mathcal{A}$ may represent the state space of a program and φ a desired property.
- When considering such evaluations the following kinds of complexities have been treated (the first one being the subject of this chapter):
  - **Data complexity** of $\mathcal{C}$: For a fixed sentence, we measure the complexity as a function of the size of the structure;
  - **Expression complexity** of $\mathcal{L}$: For a fixed structure, we measure the complexity as a function of the length of the formula;
  - **Combined complexity** of $\mathcal{L}$: It is measured as a function of both the size of the structure and the length of the formula.

Importance of the Main Theorem

- The descriptive characterizations of complexity classes given by the main theorem are of importance in various respects:
  - They may help to recognize that a concrete problem is in a given complexity class (by expressing it in the corresponding logic).
  - They allow to view the logics involved as higher programming languages for problems of the corresponding complexity class.
    (Note that the proofs of the preceding section show how to convert a sentence φ into an algorithm accepting the class of models of φ and satisfying the required resource restrictions.)
  - Characteristic features of the logic may be seen as characteristic features of the complexity class described by it and may add to a better understanding.
    (For instance, the result about FO(IFP) and PTIME shows us that inflationary inductions are an essential ingredient of PTIME algorithms.)
  - The descriptive characterizations allow to convert problems, methods, and results of complexity theory into logic and vice versa.

IFP and PFP May Be Used Only Once

- Sentences φ and ψ in a vocabulary τ with $< \in \tau$ are said to be **equivalent on ordered structures** if, for all ordered τ-structures $\mathcal{A}$,

$$\mathcal{A} \vDash \varphi \quad \text{iff} \quad \mathcal{A} \vDash \psi.$$

Corollary

On ordered structures, every FO(IFP)-sentence is equivalent to an FO(IFP)-sentence in which IFP occurs at most once.
The same applies to FO(PFP) and PFP.

- Suppose $\varphi \in \mathrm{FO(IFP)}[\tau]$ with $< \in \tau$.

  Then $\mathrm{ordMod}(\varphi) \in$ PTIME.

  Now the claim follows from a previous theorem.

  For FO(PFP) a previous theorem also applies.

The Main Theorem for Ordered Structures

Corollary

Let $\mathcal{C}$ he one of the complexity classes mentioned in the Main Theorem. If K is a class of ordered structures in $\mathcal{C}$, then there is a Turing machine M strongly witnessing $K \in \mathcal{C}$, that is:

- M accepts K;
- Every run stops in the accepting or in the rejecting state;
- Every run fulfills the time or space hounds characteristic for $\mathcal{C}$.

- Let $\mathcal{L}$ be the logic capturing $\mathcal{C}$.

  Then there is a sentence of $\mathcal{L}$ axiomatizing K.

  By the results of the preceding section we know that, for every class K axiomatizable in $\mathcal{C}$, there is a machine strongly witnessing $K \in \mathcal{C}$.

PTIME and PSPACE

- An immediate consequence of the Main Theorem is the equivalence of the following Clauses (i) and (ii):
  - (i) PTIME = PSPACE;
  - (ii) FO(IFP) $\equiv$ FO(PFP) on ordered structures.
- Note, however, that here PTIME and PSPACE are understood as classes of ordered structures and not as languages over alphabets.
- Does (i) mean the same as PTIME = PSPACE in complexity theory?

  We want to show this by making clear that here and in complexity theory we deal only with different presentations of a complexity class.

Classes of Languages vs Classes of Structures

- If $\mathcal{C}$ is a complexity class of complexity theory, we denote by $\mathcal{C}'$ the corresponding complexity class of structures.

  Example: For PTIME we have:
  - $\mathcal{C}$ consists of all languages L, $L \subseteq \mathbb{A}^+$ for some alphabet $\mathbb{A}$, such that there exists a deterministic Turing machine accepting L in polynomial time;
  - $\mathcal{C}'$ consists of all classes K, $K \subseteq \mathcal{O}[\tau]$, for some τ with $< \in \tau$, such that there is a deterministic Turing machine M accepting K in polynomial time.
- In the following let $\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2$ range over the complexity classes LOGSPACE, NLOGSPACE, PTIME, NPTIME and PSPACE of complexity theory.

$\mathcal{C} \subseteq \mathcal{C}'$

- Fix an alphabet $\mathbb{A}$.

  Let $\tau(\mathbb{A})$ be the vocabulary $\{<\} \cup \{P_a : a \in \mathbb{A}\}$, with unary P_a.

  If $u \in \mathbb{A}^+$, denote by K_u the class of structures of the form

  $$(B, <, (P_a)_{a \in A}),$$

  where:
  - The cardinality of B equals the length of w;
  - $<$ is an ordering of B;
  - P_a corresponds to the positions in u carrying an a.

  For $L \subseteq \mathbb{A}^+$, set

  $$K(L) := \bigcup_{u \in L} K_u.$$

  Clearly, $K(L) \subseteq \mathcal{O}[\tau(\mathbb{A})]$.

$\mathcal{C} \subseteq \mathcal{C}'$ (Cont'd)

- We have

$$K(\mathbb{A}^+) = \text{ordMod}\left(\forall x\left(\bigvee_{a\in\mathbb{A}}\left(P_a x \wedge \bigwedge_{\substack{b\in\mathbb{A}\\ b\neq a}} \neg P_b x\right)\right)\right).$$

It follows that

$$K(\mathbb{A}^+) \in \text{LOGSPACE}.$$

One can easily show that for $L \subseteq \mathbb{A}^+$,

$$L \in \mathcal{C} \quad \text{iff} \quad K(L) \in \mathcal{C}'.$$

Thus, we obtain "$\mathcal{C} \subseteq \mathcal{C}'$ up to transitions" (from words to ordered structures).

$\mathcal{C}' \subseteq \mathcal{C}$

- We show $\mathcal{C}' \subseteq \mathcal{C}$ up to transitions from ordered structures to words.

  Let τ, with $< \in \tau$, be given.

  Set $\mathbb{A}_0 := \{0, 1, \alpha, \omega\}$.

  For $\mathcal{A} \in \mathcal{O}[\tau]$, let

  $$u_{\mathcal{A}}$$

  be the word in $\mathbb{A}_0^+$ obtained by concatenating the inscriptions on all input tapes of a Turing machine started with input $\mathcal{A}$, including the "virtual letters" α and ω.

  For a class $K \subseteq \mathcal{O}[\tau]$, set

  $$L(K) := \{u_{\mathcal{A}} : \mathcal{A} \in K\}.$$

$\mathcal{C}' \subseteq \mathcal{C}$ (Cont'd)

- Clearly, given τ, there is a polynomial $p \in \mathbb{N}[x]$, such that, for all $\mathcal{A} \in \mathcal{O}[\tau]$, we have
$$\|A\| \le |u_{\mathcal{A}}| \le p(\|A\|).$$
In particular, for $p(x) := x^d$ we have that
$$\log \|A\| \le \log |u_{\mathcal{A}}| \le d \cdot \log \|A\|.$$
Invoking these relations one shows that:
  - $L(\mathcal{O}[\tau]) \in$ LOGSPACE;
  - For $K \subseteq \mathcal{O}[\tau]$,
$$K \in \mathcal{C}' \quad \text{iff} \quad L(K) \in \mathcal{C}.$$

Thus, "$\mathcal{C}' \subseteq \mathcal{C}$ up to transitions".

Ordered Structures and Complexity Theory

Proposition

$\mathcal{C}_1 \subseteq \mathcal{C}_2$ iff $\mathcal{C}_1' \subseteq \mathcal{C}_2'$.

- First, suppose $C_1 \subseteq \mathcal{C}_2$. Let $K \in \mathcal{C}_1'$, where $K \subseteq \mathcal{O}[\tau]$. Then, by the preceding equivalence, $L(K) \in \mathcal{C}_1$. By hypothesis, $L(K) \in \mathcal{C}_2$. Therefore, using once more the same equivalence, $K \in \mathcal{C}_2'$.

  Now assume $\mathcal{C}_1' \subseteq \mathcal{C}_2'$ and let $L \in \mathcal{C}_1$. Then, $K(L) \in \mathcal{C}_1'$ by a previous equivalence. Hence, $K(L) \in \mathcal{C}_2'$. Therefore, $L \in \mathcal{C}_2$.

Corollary

(a) FO(IFP) $\equiv$ FO(PFP) on ordered structures iff PTIME = PSPACE (in complexity theory).

(b) FO(IFP) $\equiv \Sigma_1^1$ on ordered structures iff PTIME = NPTIME (in complexity theory).

PTIME, NPTIME, FO(IFP) and SO

Corollary

The following are equivalent:

(i) PTIME = NPTIME;

(ii) FO(IFP) $\equiv$ SO on ordered structures.

- If (ii) holds then $\Sigma_1^1 \leq$ FO(IFP) on ordered structures.

  Thus NPTIME $\leq$ PTIME.

  Conversely, suppose NPTIME = PTIME.

  Then, on ordered structures, $\Sigma_1^1 \equiv$ FO(IFP).

  Now note that:

  - Σ_1^1 is closed under existential quantifications;
  - FO(IFP) is closed under boolean operations.

  Using induction, we get SO $\equiv$ FO(IFP).

Results from Complexity to Logic

- In complexity theory one shows

$$\text{LOGSPACE} \subseteq \text{NLOGSPACE} \subseteq \text{PTIME} \subseteq \text{NPTIME} \subseteq \text{PSPACE}$$

  and

$$\text{LOGSPACE} \neq \text{PSPACE}.$$

- Hence, by the Main Theorem, we get:

Corollary

On ordered structures,

(a) $\text{FO(DTC)} \leq \text{FO(TC)} \leq \text{FO(IFP)} \leq \Sigma_1^1 \leq \text{FO(PFP)}$.

(b) $\text{FO(DTC)} \not\equiv \text{FO(PFP)}$.

- Note that most of the $\leq$-relations in (a) are immediate.
- We omit the purely model-theoretic proofs of the remaining ones.

Classes of Complements

- For any of the complexity classes $\mathcal{C}$ introduced so far, in complexity theory one defines the class co-$\mathcal{C}$ to be the class of complements of languages in $\mathcal{C}$, that is, for any alphabet $\mathbb{A}$ and $L \subseteq \mathbb{A}^+$,

$$L \in \text{co-}\mathcal{C} \quad \text{iff} \quad (\mathbb{A}^+ \backslash L) \in \mathcal{C}.$$

- Clearly, any deterministic class $\mathcal{C}$ is closed under complements, that is, $\mathcal{C} = \text{co-}\mathcal{C}$.
- Similarly, we define the class co-$\mathcal{C}'$ as the class of complements of classes of structures in $\mathcal{C}'$.
- More precisely, for τ, with $< \in \tau$, and $K \subseteq \mathcal{O}[\tau]$, we set

$$K \in \text{co-}\mathcal{C}' \quad \text{iff} \quad (\mathcal{O}[\tau] \backslash K) \in \mathcal{C}'.$$

Questions Independent of Orderings

- We discuss the role of order and get information as to whether and to what extent orderings can be avoided.
- Let $\mathcal{A}$ be a not necessarily ordered structure.
- We saw that in order to consider $\mathcal{A}$ as an input for a Turing machine, we have to represent it as a string (or a sequence of strings).
- This may be done by labeling the elements of $\mathcal{A}$ in some way.
- Taking, say, the lexicographic ordering of the labels, we get an ordering on A and, hence, an ordered structure.
- If $\mathcal{A}$ is a graph, we can state questions such as

  "Is there a path from the 5-th to the 28-th element?"

- The answer depends on the ordering and is senseless for $\mathcal{A}$ itself.
- We develop a framework that enables us to concentrate on questions intrinsic to $\mathcal{A}$.

Ordered Presentations

Definition

Let K be a class of (unordered) τ-structures. Set $\tau_< := \tau \cup \{<\}$. The class $K_<$ of ordered representations of structures in K is given by

$$K_< := \{(\mathcal{A}, <) : \mathcal{A} \in K, < \text{ an ordering of } A\}.$$

- If $\mathcal{L}$ is a logic capturing the complexity class $\mathcal{C}$, we have

$$K_< \in \mathcal{C} \quad \text{iff} \quad \text{there is } \varphi \in \mathcal{L}[\tau_<], \text{ such that } K_< = \text{Mod}(\varphi).$$

- The sentence φ on the right side is order-invariant in the finite. In fact, for every $\mathcal{A}$ and any orderings $<_1$ and $<_2$ of A, we have

$$(\mathcal{A}, <_1) \in K_< \quad \text{iff} \quad (\mathcal{A}, <_2) \in K_<.$$

  Therefore, $(\mathcal{A}, <_1) \vDash \varphi$ iff $(\mathcal{A}, <_2) \vDash \varphi$.

Failure of Ordered Invariance

- If $\mathcal{L}$ would be closed under order-invariant sentences in the finite (by a previous result, FO does not have this property), we would have

$$K_< \in \mathcal{C} \quad \text{iff} \quad \text{there is } \psi \in \mathcal{L}[\tau] \text{ such that } K = \text{Mod}^\tau(\psi).$$

  In general, this does not hold.

  We give a counterexample for FO(DTC).

  Let $K = \text{EVEN}[\tau]$ with $\tau = \varnothing$ be the class of sets of even cardinality.

  We know that $K_< \in \text{LOGSPACE}$.

  So there is a sentence φ of $\text{FO(DTC)}[\tau_<]$ such that $K_< = \text{Mod}(\varphi)$.

  For example, as φ we can take the sentence

$$\neg[\text{DTC}_{x,y}y = x + 2]\min\max,$$

  where we use self-explanatory abbreviations.

Failure of Ordered Invariance (Cont'd)

- To axiomatize $K_<$ in $\mathrm{FO}(\mathrm{DTC})[\tau_<]$, we may use the sentence

$$\neg[\mathrm{DTC}_{x,y} y = x + 2] \min \max .$$

This sentence is order-invariant in the finite.

The evaluation of φ in a structure $(A, <^A)$ makes use of the ordering $<^A$, but the outcome of this evaluation does not depend on the specific ordering $<^A$ chosen.

In general, for no sentence ψ of $\mathrm{FO}(\mathrm{DTC})[\tau]$, even of $\mathrm{FO}(\mathrm{PFP})[\tau]$, is it the case that

$$K = \mathrm{Mod}(\psi).$$

Logic Strongly Capturing a Class

Definition

Let $\mathcal{L}$ be a logic and $\mathcal{C}$ a complexity class. $\mathcal{L}$ **strongly captures** $\mathcal{C}$ if, for all vocabularies τ and all classes K of τ-structures,

$$K_< \in \mathcal{C} \quad \text{iff} \quad K \text{ is axiomatizable in } \mathcal{L}.$$

- The following proposition holds for all complexity classes $\mathcal{C}$ considered so far; essentially one needs that $\mathcal{C}$ contains LOGSPACE.

Proposition

If $\mathcal{L}$ strongly captures $\mathcal{C}$ then $\mathcal{L}$ captures $\mathcal{C}$.

Strongly Capturing is Stronger than Capturing

- The converse is false.
- The counterexample given before the definition shows that FO(PFP) does not capture PSPACE strongly.

  For the class EVEN$[\tau]$ we used as a counterexample, we have:
  - EVEN$[\tau]_< \in$ LOGSPACE;
  - EVEN$[\tau]$ is not axiomatizable in FO(PFP).
- Now, we have, for arbitrary structures,

$$\text{FO(DTC)} \leq \text{FO(TC)} \leq \text{FO(IFP)} \leq \text{FO(PFP)}.$$

- So none of these logics strongly captures the complexity class corresponding to it by the Main Theorem.
- However, we know $\Sigma_1^1 \leq$ FO(PFP) only on ordered structures.
- So the result cannot be extended to Σ_1^1 and NPTIME.

Σ_1^1 and NPTIME

Theorem

Σ_1^1 strongly captures NPTIME.

- Let τ be arbitrary and K be a class of τ-structures.

  Assume $K = \mathrm{Mod}^{\tau}(\varphi)$, for some $\Sigma_1^1[\tau]$-sentence φ.

  Let $\varphi = \exists X_1 \cdots \exists X_m \psi$, with first order ψ.

  Set

  $$\chi := \exists X_1 \cdots \exists X_m (\psi \wedge \text{“} < \text{ is an ordering”}).$$

  Then, $\chi \in \Sigma_1^1[\tau_<]$ and $\mathrm{Mod}(\chi) = K_<$.

  Hence, $K_< \in$ NPTIME, by the Main Theorem.

Σ_1^1 and NPTIME (Cont'd)

- Conversely, suppose $K_< \in$ NPTIME.

  By the Main Theorem, there is a sentence $\varphi \in \Sigma_1^1[\tau_<]$, such that

  $$K_< = \text{Mod}^{\tau_<}(\varphi).$$

  Set $\psi := \exists {<} \varphi$. Then $\psi \in \Sigma_1^1[\tau]$.

  Moreover, for any τ-structure $\mathcal{A}$, we have

  $$\begin{array}{lll} \mathcal{A} \vDash \psi & \text{iff} & \text{there is } <^A \text{ with } (\mathcal{A}, <^A) \vDash \varphi \\ & \text{iff} & \text{there is } <^A \text{ with } (\mathcal{A}, <^A) \in K_< \\ & \text{iff} & \mathcal{A} \in K. \end{array}$$

  So $K = \text{Mod}^{\tau}(\psi)$.

Π^1_1 and co-NPTIME

Theorem

Π^1_1 strongly captures co-NPTIME.

- Let τ be arbitrary and K be a class of τ-structures. If $\mathrm{Str}[\tau]$ denotes the class of all τ-structures, then $(\mathrm{Str}[\tau]\backslash K)_< = \mathcal{O}[\tau_<]\backslash K_<$.

  Therefore, we have

$$\begin{array}{lll} K \in \Pi^1_1 & \text{iff} & \text{there is } \psi(X_1,\ldots,X_m) \in \mathrm{FO}[\tau], \text{ such that} \\ & & K = \mathrm{Mod}(\forall X_1 \cdots \forall X_m \psi) \\ & \text{iff} & \text{there is } \chi(X_1,\ldots,X_m) \in \mathrm{FO}[\tau], \text{ such that} \\ & & \mathrm{Str}[\tau]\backslash K = \mathrm{Mod}(\exists X_1 \cdots \exists X_m \chi) \\ & \text{iff} & (\mathrm{Str}[\tau]\backslash K)_< \in \mathrm{NPTIME} \\ & \text{iff} & (\mathcal{O}[\tau_<]\backslash K_<) \in \mathrm{NPTIME} \\ & \text{iff} & K_< \in \text{co-NPTIME}. \end{array}$$

NPTIME and co-NPTIME

Corollary

NPTIME = co-NPTIME iff $\Sigma^1_1 \equiv \Pi^1_1$.

- As an example we show the implication from left to right.

  Let K be a class of structures.

  Then

$$\begin{array}{lll} K \in \Sigma^1_1 & \text{iff} & K_< \in \text{NPTIME} \\ & \text{iff} & K_< \in \text{co-NPTIME} \quad \text{(by hypothesis)} \\ & \text{iff} & K \in \Pi^1_1. \end{array}$$

NPTIME and co-NPTIME (Cont'd)

Corollary

NPTIME = co-NPTIME iff $\mathrm{SO} \equiv \Sigma^1_1$.

- By the preceding corollary it suffices to prove

$$\Pi^1_1 \equiv \Sigma^1_1 \quad \text{iff} \quad \mathrm{SO} \equiv \Sigma^1_1.$$

For a logic $\mathcal{L}$ we write $\varphi \underset{\sim}{\in} \mathcal{L}$ to express that the sentence φ is equivalent to an $\mathcal{L}$-sentence. Clearly, we have:

$$\begin{aligned} \varphi \in \Sigma^1_1 &\quad \text{implies} \quad \neg\varphi \underset{\sim}{\in} \Pi^1_1; \\ \varphi \in \Pi^1_1 &\quad \text{implies} \quad \neg\varphi \underset{\sim}{\in} \Sigma^1_1. \end{aligned}$$

NPTIME and co-NPTIME (Cont'd)

- Now suppose that $\mathrm{SO} \equiv \Sigma_1^1$. Then $\Pi_1^1 \leq \Sigma_1^1$.

  Let $\varphi \in \Sigma_1^1$. Then, $\neg\varphi \underset{\sim}{\in} \Pi_1^1$.

  Hence, $\neg\varphi \underset{\sim}{\in} \Sigma_1^1$. Therefore, $\varphi \underset{\sim}{\in} \Pi_1^1$.

  Conversely, assume that $\Pi_1^1 \equiv \Sigma_1^1$.

  One easily shows that the class Σ_1^1 is closed - up to equivalence - under $\vee$ and existential first-order and second-order quantifications.

  For closure under $\neg$ argue as follows:

  Suppose $\varphi = \neg\psi$ and $\psi \in \Sigma_1^1$.

  Then $\neg\psi \underset{\sim}{\in} \Pi_1^1 \equiv \Sigma_1^1$.

FO(posTC) and FO(TC)

Theorem

On ordered structures, FO(posTC) $\equiv$ FO(TC).

- We make use of the fact that, on ordered structures, every formula of FO(posTC) is equivalent to a formula of the form $[\mathrm{TC}_{\overline{x},\overline{y}}\psi]\widetilde{\min}\widetilde{\max}$ with first-order ψ.

  The proof of the theorem proceeds by induction on FO(TC)-formulas, the only nontrivial case being the negation step.

  By the induction hypothesis and the fact just mentioned, we may assume that $\varphi = \neg[\mathrm{TC}_{\overline{x},\overline{y}}\psi]\overline{s}\overline{t}$, with first-order ψ.

  For simplicity, we assume that $\psi = \psi(\overline{x},\overline{y})$.

  By the Main Theorem, there is a Turing machine M_0 strongly witnessing that $\mathrm{ordMod}(\psi) \in$ LOGSPACE.

FO(posTC) and FO(TC) (Cont'd)

- Suppose $\overline{x} = x_1 \ldots x_r$. Given a structure $\mathcal{A}$ and $\overline{a}, \overline{b} \in A^r$, let $d_\psi(\overline{a}, \overline{b})$ be the length of the shortest ψ-path connecting $\overline{a}$ and $\overline{b}$,

$$d_\psi(\overline{a}, \overline{b}) \quad := \quad \min\{k > 0 : \text{there exist } \overline{a}_0 = \overline{a}, \overline{a}_1, \ldots, \overline{a}_k = \overline{b} \text{ such that } \mathcal{A} \vDash \psi[\overline{a}_i, \overline{a}_{i+1}] \text{ for } i < k\},$$

  where $d_\psi(\overline{a}, \overline{b}) := \infty$ in case the set on the right side is empty.

  Note that:
  - If $d_\psi(\overline{a}, \overline{b}) < \infty$, then $0 < d_\psi(\overline{a}, \overline{b}) \leq \|A\|^r$;
  - $d_\psi(\overline{a}, \overline{c}) \leq d_\psi(\overline{a}, \overline{b}) + d_\psi(\overline{b}, \overline{c})$.

  Moreover, $\neg[\mathrm{TC}_{\overline{x},\overline{y}}]\overline{s}\overline{t}$ is equivalent to

$$\text{“}\|\{\overline{v} : d_\psi(\overline{s}, \overline{v}) < \infty\}\| = \|\{\overline{v} : d_{(\psi(\overline{x},\overline{y}) \wedge \neg \overline{y} = \overline{t})}(\overline{s}, \overline{v}) < \infty\}\|\text{”}.$$

FO(posTC) and FO(TC) (The Machine M)

- We first show that there is a nondeterministic log space bounded machine M, such that, for any ordered structure $(\mathcal{A},\overline{a},\overline{\ell},\overline{w},\overline{w}')$,

  $$\text{if } \|\{\overline{e}: d_\psi(\overline{a},\overline{e}) \leq \overline{ell}\}\| = \overline{w}, \text{ then}$$
  $$M \text{ accepts } (\mathcal{A},\overline{a},\overline{\ell},\overline{w},\overline{w}') \text{ iff } \|\{\overline{e}: d_\psi(\overline{a},\overline{e}) \leq \overline{\ell}+1\}\| = \overline{w}',$$

  where the corresponding natural numbers $\leq \|A\|^r$ are given by their $\|A\|$-adic representations $\overline{\ell} = \ell_0 \ldots \ell_r$, $\overline{w} = w_0 \ldots w_r$, $\overline{w}' = w_0' \ldots w_r'$.

  We present the machine M.

  When during its computation M checks whether $\mathcal{A} \models \psi[\overline{c},\overline{d}]$ holds or not, this is done by invoking machine M_0.

FO(posTC) and FO(TC) (The Machine M Cont'd)

- Suppose M is Started with $(\mathcal{A}, \overline{a}, \overline{\ell}, \overline{w}, \overline{w}')$.

  M first sets a counter to $\overline{w}'$.

  Then, for every $\overline{b} \in A^r$, it carries out either (1) or (2), the choice being done nondeterministically.

  (1) M nondeterministically guesses a path witnessing $d_\psi(\overline{a}, \overline{b}) \leq \overline{\ell} + 1$ and decreases the counter by one.
  In case the counter is zero it rejects.

  (2) Using an additional counter, initialized at $\overline{w}$, M nondeterministically guesses $\overline{w}$ many distinct tuples $\overline{c} \in A^r$ together with a proof that $d_\psi(\overline{a}, \overline{c}) \leq \overline{\ell}$.
  For each such $\overline{c}$, it shows that $\overline{c} \neq \overline{b}$ and $\neg\psi[\overline{c}, \overline{b}]$ (thus, in case $\|\{\overline{e} : d_\psi(\overline{a}, \overline{e}) \leq \overline{\ell}\}\| = \overline{w}$, proving that $d_\psi(\overline{a}, \overline{b}) > \overline{\ell} + 1$).
  In case $\overline{\ell} = 0$, it shows $\neg\psi[\overline{a}, \overline{b}]$.

  Finally, if all $\overline{b} \in A^r$ have been dealt with and the counter is 0, M accepts.

FO(posTC) and FO(TC) (The Bound)

- Machine M is log space-bounded.

  So, by the Main Theorem, there exists a formula

  $$\chi_\psi(\overline{u}, \overline{y}, \overline{w}, \overline{w}') \in \text{FO(posTC)}$$

  axiomatizing the class accepted by M.

  Note that $1\underbrace{0\ldots0}_{r \text{ times}}$ is (the representation of) $\|A\|^r$.

  Now, we get a formula $\rho_\psi(\overline{u}, \overline{w}) \in \text{FO(posTC)}$ by setting

  $$\rho_\psi := [\text{TC}_{\overline{y}\overline{w}, \overline{y}'\overline{w}'}(\chi_\psi(\overline{u}, \overline{y}, \overline{w}, \overline{w}') \wedge \overline{y}' = \overline{y} + 1)]\widetilde{\min}\widetilde{\min}1\underbrace{0\ldots0}_{r \text{ times}}\overline{w}.$$

  $\rho_\psi(\overline{u}, \overline{w})$ has the meaning "$\|\{\overline{v} : d_\psi(\overline{u}, \overline{v}) < \infty\}\| = \overline{w}$".

  Equivalently, $\rho_\psi(\overline{u}, \overline{w})$'s meaning is "$\|\{\overline{v} : d_\psi(\overline{u}, \overline{v}) \leq \|A\|^r\}\| = \overline{w}$".

FO(posTC) and FO(TC) (The Bound Cont'd)

- We saw above that the FO(TC)- formula $\varphi = \neg[\mathrm{TC}_{\overline{x},\overline{y}}\psi(\overline{x},\overline{y})]\overline{s}\overline{t}$ is equivalent to

$$\text{“}\|\{\overline{v} : d_\psi(\overline{s},\overline{v}) < \infty\}\| = \|\{\overline{v} : d_{(\psi(\overline{x},\overline{y})\wedge\neg\overline{y}=\overline{t})}(\overline{s},\overline{v}) < \infty\}\|\text{”}.$$

  So we obtain that φ is equivalent to the FO(posTC)-formula

$$\exists\overline{z}(\rho_\psi(\overline{s},\overline{z}) \wedge \rho_{(\psi\wedge\neg\overline{y}=\overline{t})}(\overline{s},\overline{z})).$$

- As a consequence of Part (b) of the Main Theorem, whose proof we have just completed, we get, since FO(TC) is closed under negation:

Corollary

NLOGSPACE = co-NLOGSPACE.