Logic (master program)

Georg Moser

Institute of Computer Science @ UIBK

Winter 2008

Summary of Last Lecture

decision problem A	set $A \subseteq \mathbb{N}^n$	function
decidable	recursive	χ_A is recursive
semi-decidable	r.e.	h_A is partial recursive
undecidable	not recursive	χ_A is not recursive

where h_A be defined as

$$h_A(x_1, \ldots, x_n) = \begin{cases} 1 & (x_1, \ldots, x_n) \in A \\ \text{undefined} & \text{otherwise} \end{cases}$$

Theorem (Rice)

let A be an index set; if A is neither $\varnothing$ or $\mathbb{N}$, then A is not recursive

The Arithmetical Hierarchy

Definition

a $\mathcal{V}_{ar}$-formula φ is

Π_1 if $\varphi \equiv \forall y \psi(\vec{x}, y)$ $\qquad$ Σ_1 if $\varphi \equiv \exists y \psi(\vec{x}, y)$ $\qquad$ $\psi \in \Delta_0$

Lemma

$\Sigma_1 = \{\text{r.e. sets}\}$

Definition

a $\mathcal{V}_{ar}$-formula φ is

- Π_{n+1} if $\varphi \equiv \forall y \psi(\vec{x}, y)$
- Σ_{n+1} if $\varphi \equiv \exists y \psi(\vec{x}, y)$
- Δ_{n+1} if $\varphi \in \Pi_{n+1} \cap \Sigma_{n+1}$

$\psi \in \Delta_n$

Lemma

$\Delta_1 = \{\text{recursive sets}\}$

$\mathsf{DTIME}(T(n)) := \{\mathrm{L}(M) \mid M$ is a deterministic multitape TM running in time $\mathrm{O}(T(n))\}$

$\mathsf{NTIME}(T(n)) := \{\mathrm{L}(M) \mid M$ is a nondeterministic multitape TM running in time $\mathrm{O}(T(n))\}$

$\mathsf{DSPACE}(S(n)) := \{\mathrm{L}(M) \mid M$ is a deterministic multitape TM running in space $\mathrm{O}(S(n))\}$

$\mathsf{NSPACE}(S(n)) := \{\mathrm{L}(M) \mid M$ is a nondeterministic multitape TM running in space $\mathrm{O}(S(n))\}$

Definition

$\mathsf{LOGSPACE} := \mathsf{DSPACE}(\log n)$ $\qquad$ $\mathsf{NP} := \mathsf{NTIME}(n^{\mathrm{O}(1)})$

$\mathsf{NLOGSPACE} := \mathsf{NSPACE}(\log n)$ $\qquad$ $\mathsf{PSPACE} := \mathsf{DSPACE}(n^{\mathrm{O}(1)})$

$\mathsf{P} := \mathsf{DTIME}(n^{\mathrm{O}(1)})$ $\qquad$ $\mathsf{NPSPACE} := \mathsf{NSPACE}(n^{\mathrm{O}(1)})$

Theorem

$$\mathsf{LOGSPACE} \subseteq \mathsf{NLOGSPACE} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \subseteq \mathsf{NPSPACE}$$

Homework

- Exercise 7.16.
- Exercise 7.25.
- Exercise 7.37.

Content

introduction, propositional logic, semantics, formal proofs, resolution (propositional)

first-order logic, semantics, structures, theories and models, formal proofs, Herbrand theory, completeness of first-order logic, properties of first-order logic, resolution (first-order)

introduction to computability, introduction to complexity, finite model theory

beyond first order: modal logics in a general setting, higher-order logics, introduction to Isabelle

Logspace Reducibility

Definition (logspace transducer)

- a total deterministic logspace-bounded TM with output is called logspace transducer
- total means: it halts on all inputs
- with output means: ∃ write-only output tape
- hence, a logspace transducer has:
  1. a two-way read-only input tape
  2. a two-way read/write logspace-bounded worktape initially blank
  3. a write-only left-to-right output tape initially blank
  4. Σ is the input alphabet
  5. Γ is the worktape alphabet
  6. Δ is the output alphabet

Lemma

output of a logspace transducer is polynomially bounded in length

Definition (logspace reducibility)

- for $A \subseteq \Sigma^*$, $B \subseteq \Delta^*$
- we write $A \leqslant_m^{\mathsf{log}} B$ if ∃ logspace-computable function $\sigma \colon \Sigma^* \to \Delta^*$ with

$$x \in A \quad \text{if and only if} \quad \sigma(x) \in B$$

- we say A is logspace reducible to B

Definition (hardness)

a set $A \subseteq \Sigma^*$ is $\leqslant_m^{\mathsf{log}}$-hard for a complexity class $\mathcal{C}$ if

- $\forall B \in \mathcal{C}$ we have $B \leqslant_m^{\mathsf{log}} A$

Definition (completeness)

a set $A \subseteq \Sigma^*$ is complete for $\mathcal{C}$ with respect to $\leqslant_m^{\mathsf{log}}$ if

1. A is $\leqslant_m^{\mathsf{log}}$-hard for $\mathcal{C}$
2. $A \in \mathcal{C}$

Second-Order Logic

Definition — second-order logic formulas

a formula of second-order logic is an expression that is either an atomic formula or generated through the following rules

- if φ is a formula, so is $\neg\varphi$
- if φ and ψ are formulas, so is $\varphi \wedge \psi$
- if φ is a formula, then so is $\exists x \varphi$
- if φ is a formula, then so is $\exists R^n \varphi$ for any n-ary relation R

Definition

a second-order sentence is a formula having no free (individual) variables

Example

let $\mathcal{V} = \{P, Q\}$ and consider the following $\mathcal{V}$-sentence:

$$\exists R^2 (\forall x \forall y (R(x,y) \to (P(x) \wedge Q(y))) \wedge \forall x (P(x) \to \exists y R(x,y)) \wedge \\ \wedge \forall x \forall y \forall z (R(x,y) \wedge R(x,z) \to y = z))$$

Definition — $\mathcal{M} \models \varphi$

let φ be a $\mathcal{V}$-sentence, we define $\mathcal{M} \models \varphi$ by structural induction

$\mathcal{M} \models \varphi$	as in the first-order case if φ is first-order
$\mathcal{M} \models \exists R^n \psi$	if $\exists$ expansion $\mathcal{M}'$ of $\mathcal{M}$ to $\mathcal{V} \cup \{R\}$ such that $\mathcal{M}' \models \psi$

Lemma

second-order logic is more expressive than first-order logic

Example

the following sentence expresses that there exists a path from node s to node t in graph G:

$$\exists P^2 (\forall x \forall y \forall z (\neg P(x,x) \wedge (P(x,y) \wedge P(y,z) \to P(x,z))) \\ \forall x \forall y ((P(x,y) \wedge \forall z (\neg P(x,z) \vee \neg P(z,y))) \to G(x,y)) \wedge P(s,t))$$

Theorem

- second-order logic is neither complete nor compact
- the set of valid (second-order) sentences is not recursive enumerable

Complexity Theory via Logic

re-interpretation

- usually a problem in complexity theory is simply a language $\mathrm{L} \subseteq \Sigma^*$
- now a problem is a subset of a set of finite structures

Definition

let $\mathcal{S}$ a set of finite $\mathcal{V}$-structures for finite $\mathcal{V}$

- φ a $\mathcal{V}$-sentence
- $\mathcal{M}$ a $\mathcal{V}$-structure in $\mathcal{S}$
- the $\varphi - \mathcal{S}$ problem asks: $\mathcal{M} \models \varphi$

Definition

a second-order sentence φ is existential ($\exists$SO for short) if

$$\varphi \equiv \exists R_1^{n_1} \exists R_2^{n_2} \dots \exists R_k^{n_k} \psi \qquad \psi \text{ is first-order}$$

Lemma

if φ is $\exists$SO, then the $\varphi - \mathcal{S}$ problem is in NP

Proof

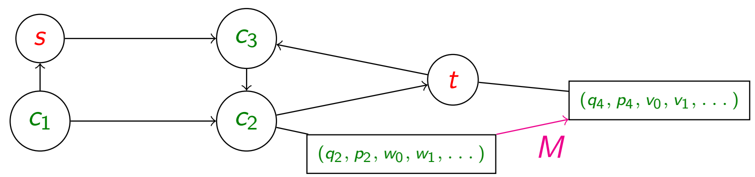
by structural induction ∎

Lemma

let $\mathcal{S}$ a set of finite $\mathcal{V}$-structures, φ a $\mathcal{V}$-sentence such that $\varphi - \mathcal{S}$ is decidable by a TM M that runs in time n^k, where n is the length of the description of $\mathcal{M}$; then φ is equivalent to an existential second-order sentence

Proof Sketch

- as TM M is terminating, we can build a configuration graph G



- reachability is expressible as existential second-order sentence ∎

Theorem

Fagin

let $\mathcal{V}$ be a relational vocabulary, $\mathcal{S}$ a set of finite $\mathcal{V}$-structure

- a $\mathcal{V}$-sentence φ is equivalent to a sentence in ∃SO if and only if $\varphi - \mathcal{S} \in \mathsf{NP}$
- moreover the first-order part of φ is universal

Proof

straightforward adaption and analysis of the lemmata ∎

Corollary

the satisfiability problem (of propositional logic) (SAT) is complete for NP with respect to $\leqslant_m^{\mathsf{log}}$

Corollary

the following is equivalent

- $\mathsf{NP} = \mathsf{co} - \mathsf{NP}$
- ∃SO is equivalent to (full) second-order logic