Complexity Theory

Georg Moser

Institute of Computer Science @ UIBK

Summer 2008

Outline

- Summary of Last Lecture: Problems Complete for PSPACE
- (Old) Exercises
- Definition of PH via ATMs
- Oracle Machines and Relativised Complexity Classes
- Equivalence

Definition — QBF

the quantified Boolean formula problem is the problem of determining the truth of

$$Q_1 x_1 Q_2 x_2 \dots Q_n x_n B(x_1, \dots, x_n)$$

$B(x_1, \dots, x_n)$ is a Boolean formula, $Q_i \in \{\forall, \exists\}$ with quantification over $\{0, 1\}$

Observation

- $\exists x_1 \dots \exists x_n B(x_1, \dots, x_n)$-QBF = SAT
- $\forall x_1 \dots \forall x_n B(x_1, \dots, x_n)$-QBF = VALIDITY

Theorem

QBF is $\leqslant_m^{\log}$-complete for PSPACE

Definition — information game

a 2-person perfect information game is a graph $G = (\text{BOARDS}, \text{MOVE})$ and a start state $s \in \text{BOARDS}$

- BOARDS defines the play field
- MOVE specifies the legal actions
- players alternate, Player I starts in s
- Player I chooses $s_1 \in \text{BOARDS}$, such that $\text{MOVE}(s, s_1)$
- Player II chooses $s_2 \in \text{BOARDS}$, such that $\text{MOVE}(s_1, s_2)$ ...
- a player wins, if the other cannot make a move

Definition

- $\text{CHECKMATE}(y) :\Leftrightarrow \forall z \neg \text{MOVE}(y, z)$
- $\text{WIN}(x) \Leftrightarrow \exists \text{MOVE}(x, y) \wedge \forall z(\text{MOVE}(y, z) \rightarrow \text{WIN}(z))$
- for $x \in \text{BOARDS} \wedge \text{WIN}(x)$, x is forced win

Definition — generalised geography

a triple (C, E, s) is called a generalised geography game if

- (C, E) directed graph
- $s \in C$
- Player I starts in $s = s_0$ and moves in even states from s_{2i} to adjacent vertex s_{2i+1}
- Player II moves from s_{2i+1} to s_{2i+2}
- vertexes must not be revisited

Theorem

Generalised geography is $\leqslant_m^{\log}$-complete for PSPACE

Homework

1. Miscellaneous Exercises 11
2. Miscellaneous Exercises 14
3. Miscellaneous Exercises 26

Definition of PH via ATMs

Definition — Σ_k-machine

a Σ_k-machine is an ATM for which the computation path is dividable in separate sections on any input and

1. any section consists only of $\wedge$- or $\vee$-configurations
2. at most k sections
3. the first consist of $\vee$-configurations

a Π_k-machine is defined by swapping $\vee$ and $\wedge$

Σ_0, Π_0 are defined to be deterministic TMs

Example

a Σ_1-machine is a nondeterministic TM

Definition — Σ_k^{p}, Π_k^{p}

$$\Sigma_k^{\mathrm{p}} := \{\mathrm{L}(M) \mid M \text{ is polytime bounded } \Sigma_k^{\mathrm{p}}\text{-machines}\}$$
$$\Pi_k^{\mathrm{p}} := \{\mathrm{L}(M) \mid M \text{ is polytime bounded } \Pi_k^{\mathrm{p}}\text{-machines}\}$$

Lemma

$$\Pi_k^{\mathrm{p}} = \mathrm{co} - \Sigma_k^{\mathrm{p}} = \{\sim A \mid A \in \Sigma_k^{\mathrm{p}}\}$$
$$\Sigma_k^{\mathrm{p}} \cup \Pi_k^{\mathrm{p}} \subseteq \Sigma_{k+1}^{p} \cap \Pi_{k+1}^{p}$$
$$\bigcup_{k \geqslant 1} \Sigma_k^{\mathrm{p}} = \bigcup_{k \geqslant 0} \Pi_k^{\mathrm{p}} \subseteq \mathrm{PSPACE}$$

Definition — H_k

$$H_k := \{M\#x\#^m \mid M \text{ an ATM and } M_k^m \text{ accepts x}\}$$

here M_k^m denotes the modification of M such that

1. at most k intervals of $\wedge$- and $\vee$-configurations, beginning with $\vee$
2. runtime at most m

Lemma

H_k is $\leqslant_m^{\log}$-complete for Σ_k^{p} and $\sim H_k$ is $\leqslant_m^{\log}$-complete for Π_k^{p}

Definition

- an oracle machine is a TM M^B with an extra write-only tape, the oracle tape
- M^B additionally has oracle query state and specific oracle answer states "yes" and "no"
- M^B writes y on oracle tape, oracle answers "yes" if $y \in B$ and "no" otherwise

Definition

let B be a language and $\mathcal{C}$ a complexity class

$$\mathrm{P}^B := \{\mathrm{L}(M) \mid M \text{ is a deterministic, polytime bounded oracle machine with oracle } B\}$$

$$\mathrm{NP}^B := \{\mathrm{L}(M) \mid M \text{ is a nondeterministic, polytime bounded oracle machine with oracle } B\}$$

$$\mathrm{P}^{\mathcal{C}} := \bigcup_{B \in \mathcal{C}} \mathrm{P}^B \qquad \mathrm{NP}^{\mathcal{C}} := \bigcup_{B \in \mathcal{C}} \mathrm{NP}^B$$

Lemma

if B is $\leqslant_m^{\log}$-complete for $\mathcal{C}$, then $\{\mathrm{P}, \mathrm{NP}\}^{\mathcal{C}} = \{\mathrm{P}, \mathrm{NP}\}^B$

Definition

$\leqslant_T^{\mathrm{p}}$

we defined polytime Turing reducibility $\leqslant_T^{\mathrm{p}}$ as follows:

$$A \leqslant_T^{\mathrm{p}} B \quad \text{if} \quad A \in \mathrm{P}^B$$

Lemma

if $A \leqslant_m^{\log} B$ then $A \leqslant_m^{\mathrm{p}} B$ and $A \leqslant_T^{\mathrm{p}} B$

Theorem

consider

$$\mathrm{NP} \subseteq \mathrm{NP}^{\mathrm{NP}} \subseteq \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}} \dots$$

i.e., $\mathrm{NP}_1 := \mathrm{NP}$ and $\mathrm{NP}_{k+1} := \mathrm{NP}^{\mathrm{NP}_k}$, then $\forall k \geqslant 1$: $\mathrm{NP}_k = \Sigma_k^p$

define $\exists^t x\ \varphi(x) :\Leftrightarrow \exists x |y| \leqslant t \wedge \varphi(x)$ and $\forall^t x\ \varphi(x) :\Leftrightarrow \forall x |y| \leqslant t \rightarrow \varphi(x)$

Theorem

a language L is in Σ_k^{p} iff there is a deterministic polytime computable $(k+1)$-ary predicate R and a constant c such that

$$A = \{x \mid \exists^{|x|^c} y_1 \forall^{|x|^c} y_2 \exists^{|x|^c} y_3 \dots \mathrm{Q}^{|x|^c} y_k R(x, y_1, \dots, y_k)$$

$(\mathrm{Q} \in \{\exists, \forall\})$