Topics in Logic and Complexity
Handout 5

Anuj Dawar

MPhil Advanced Computer Science, Lent 2010

Fagin's Theorem

Theorem (Fagin)
A class $\mathcal{C}$ of finite structures is definable by a sentence of *existential second-order logic* if, and only if, it is decidable by a *nondeterminisitic machine* running in polynomial time.

$$\mathsf{ESO} = \mathsf{NP}$$

One direction is easy: Given $\mathbb{A}$ and $\exists P_1 \ldots \exists P_m \phi$.

> a nondeterministic machine can guess an interpretation for $P_1, \ldots, P_m$ and then verify ϕ.

Fagin's Theorem

Given a machine M and an integer k, there is an ESO sentence ϕ such that $\mathbb{A} \models \phi$ if, and only if, M accepts $[\mathbb{A}]_<$, for some order $<$ in n^k steps.

We construct a *first-order* formula $\phi_{M,k}$ such that

$(\mathbb{A}, <, \mathbf{X}) \models \phi_{M,k} \quad \Leftrightarrow \quad$ $\mathbf{X}$ codes an accepting computation of M of length at most n^k on input $[\mathbb{A}]_<$

So, $\mathbb{A} \models \exists < \exists \mathbf{X}\, \phi_{M,k}$ if, and only if, there is some order $<$ on $\mathbb{A}$ so that M accepts $[\mathbb{A}]_<$ in time n^k.

Order

The formula $\phi_{M,k}$ is built up as the *conjunction* of a number of formulas. The first of these simply says that $<$ is a *linear order*

$$\begin{aligned}
&\forall x (\neg x < x) \wedge \\
&\forall x \forall y (x < y \rightarrow \neg y < x) \wedge \\
&\forall x \forall y (x < y \vee y < x \vee x = y) \\
&\forall x \forall y \forall z (x < y \wedge y < z \rightarrow x < z)
\end{aligned}$$

We can use a linear order on the elements of $\mathbb{A}$ to define a lexicographic order on k-tuples.

Ordering Tuples

If $\mathbf{x} = x_1, \ldots, x_k$ and $\mathbf{y} = y_1, \ldots, y_k$ are k-tuples of variables, we use $\mathbf{x} = \mathbf{y}$ as shorthand for the formula $\bigwedge_{1 \leq i \leq k} x_i = y_i$ and $\mathbf{x} < \mathbf{y}$ as shorthand for the formula

$$\bigvee_{1 \leq i \leq k} \left((\bigwedge_{j<i} x_j = y_j) \wedge x_i < y_i \right)$$

We also write $\mathbf{y} = \mathbf{x} + 1$ for the following formula:

$$\mathbf{x} < \mathbf{y} \wedge \forall \mathbf{z} \big(\mathbf{x} < \mathbf{z} \rightarrow (\mathbf{y} = \mathbf{z} \vee \mathbf{y} < \mathbf{z}) \big)$$

Constructing the Formula

Let $M = (K, \Sigma, s, \delta)$.

The tuple $\mathbf{X}$ of second-order variables appearing in $\phi_{M,k}$ contains the following:

S_q	a k-ary relation symbol for each $q \in K$
T_σ	a $2k$-ary relation symbol for each $\sigma \in \Sigma$
H	a $2k$-ary relation symbol

Intuitively, these relations are intended to capture the following:

- $S_q(\mathbf{x})$ – the state of the machine at time $\mathbf{x}$ is q.
- $T_\sigma(\mathbf{x}, \mathbf{y})$ – at time x, the symbol at position j of the tape is σ.
- $H(\mathbf{x}, \mathbf{y})$ – at time $\mathbf{x}$, the tape head is pointing at tape cell $\mathbf{y}$.

We now have to see how to write the formula $\phi_{M,k}$, so that it enforces these meanings.

Initial state is s and the head is initially at the beginning of the tape.

$$\forall \mathbf{x} \big((\forall \mathbf{y}\ \mathbf{x} \leq \mathbf{y}) \rightarrow S_s(\mathbf{x}) \wedge H(\mathbf{x}, \mathbf{x}) \big)$$

The head is never in two places at once

$$\forall \mathbf{x} \forall \mathbf{y} \big(H(\mathbf{x}, \mathbf{y}) \rightarrow (\forall \mathbf{z} (\mathbf{y} \neq \mathbf{z}) \rightarrow (\neg H(\mathbf{x}, \mathbf{z}))) \big)$$

The machine is never in two states at once

$$\forall \mathbf{x} \bigwedge_q (S_q(\mathbf{x}) \rightarrow \bigwedge_{q' \neq q} (\neg S_{q'}(\mathbf{x})))$$

Each tape cell contains only one symbol

$$\forall \mathbf{x} \forall \mathbf{y} \bigwedge_\sigma (T_\sigma(\mathbf{x}, \mathbf{y}) \rightarrow \bigwedge_{\sigma' \neq \sigma} (\neg T_{\sigma'}(\mathbf{x}, \mathbf{y})))$$

Initial Tape Contents

The initial contents of the tape are $[\mathbb{A}]_<$.

$$\begin{aligned}\forall \mathbf{x} \quad & \mathbf{x} \leq n \rightarrow T_1(\mathbf{1}, \mathbf{x}) \wedge \\ & \mathbf{x} \leq n^a \rightarrow (T_1(\mathbf{1}, \mathbf{x} + n + 1) \leftrightarrow R_1(\mathbf{x}|_a)) \\ \ldots & \end{aligned}$$

where,

$$\mathbf{x} < n^a \quad : \quad \bigwedge_{i \leq (k-a)} x_i = 0$$

The tape does not change except under the head

$$\forall \mathbf{x} \forall \mathbf{y} \forall \mathbf{z} (\mathbf{y} \neq \mathbf{z} \rightarrow (\bigwedge_{\sigma} (H(\mathbf{x}, \mathbf{y}) \wedge T_\sigma(\mathbf{x}, \mathbf{z}) \rightarrow T_\sigma(\mathbf{x} + 1, \mathbf{z})))$$

Each step is according to δ.

$$\begin{aligned}\forall \mathbf{x} \forall \mathbf{y} \bigwedge_{\sigma} \bigwedge_{q} & (H(\mathbf{x}, \mathbf{y}) \wedge S_q(\mathbf{x}) \wedge T_\sigma(\mathbf{x}, \mathbf{y})) \\ & \rightarrow \bigvee_{\Delta} (H(\mathbf{x} + 1, \mathbf{y}') \wedge S_{q'}(\mathbf{x} + 1) \wedge T_{\sigma'}(\mathbf{x} + 1, \mathbf{y}))\end{aligned}$$

where Δ is the set of all triples (q', σ', D) such that $((q, \sigma), (q', \sigma', D)) \in \delta$ and

$$\mathbf{y}' = \begin{cases} \mathbf{y} & \text{if } D = S \\ \mathbf{y} - 1 & \text{if } D = L \\ \mathbf{y} + 1 & \text{if } D = R \end{cases}$$

Finally, some accepting state is reached

$$\exists \mathbf{x}\, S_{\text{acc}}(\mathbf{x})$$

NP

Recall that a languge L is in NP if, and only if,

$$L = \{x \mid \exists y R(x, y)\}$$

where R is *polynomial-time decidable* and *polynomially-balanced*.

Fagin's theorem tells us that polynomial-time decidability can, in some sense, be replaced by *first-order definability*.

co-NP

USO—*universal second-order logic* consists of those formulas of second-order logic of the form:

$$\forall X_1 \cdots \forall X_k \, \phi$$

where ϕ is a first-order formula.

A corollary of Fagin's theorem is that a class $\mathcal{C}$ of finite structures is definable by a sentence of *existential second-order logic* if, and only if, it is decidable by a *nondeterminisitic machine* running in polynomial time.

$$\mathsf{USO} = \mathsf{co\text{-}NP}$$

Second-Order Alternation Hierarchy

We can define further classes by allowing other second-order *quantifier prefixes*.

$\Sigma^1_1 = \mathsf{ESO}$

$\Pi^1_1 = \mathsf{USO}$

Σ^1_{n+1} is the collection of properties definable by a sentence of the form: $\exists X_1 \cdots \exists X_k \, \phi$ where ϕ is a Π^1_n formula.

Π^1_{n+1} is the collection of properties definable by a sentence of the form: $\forall X_1 \cdots \forall X_k \, \phi$ where ϕ is a Σ^1_n formula.

Note: every formula of second-order logic is Σ^1_n and Π^1_n for some n.

Polynomial Hierarchy

We have, for each n:

$$\Sigma^1_n \cup \Pi^1_n \quad \subseteq \quad \Sigma^1_{n+1} \cap \Pi^1_{n+1}$$

The classes together form the *polynomial hierarchy* or PH.

$\mathsf{NP} \subseteq \mathsf{PH} \subseteq \mathsf{PSPACE}$

$\mathsf{P} = \mathsf{NP}$ if, and only if, $\mathsf{P} = \mathsf{PH}$

Reading List for this Handout

1. Grädel et al. Section 3.2
2. Libkin. Chapter 9.
3. Ebbinghaus and Flum. Chapter 7.