Topics in Logic and Complexity
Part 9

Anuj Dawar

MPhil Advanced Computer Science, Lent 2013

Syntax of LFP

- Any relation symbol of arity k is a predicate expression of arity k;
- If R is a relation symbol of arity k, $\mathbf{x}$ is a tuple of variables of length k and ϕ is a formula of LFP in which the symbol R only occurs positively, then

$$\mathbf{lfp}_{R,\mathbf{x}}\phi$$

  is a predicate expression of LFP of arity k.

All occurrences of R and variables in $\mathbf{x}$ in $\mathbf{lfp}_{R,\mathbf{x}}\phi$ are *bound*

Syntax of LFP

- If t_1 and t_2 are terms, then $t_1 = t_2$ is a formula of LFP.
- If P is a predicate expression of LFP of arity k and $\mathbf{t}$ is a tuple of terms of length k, then $P(\mathbf{t})$ is a formula of LFP.
- If ϕ and ψ are formulas of LFP, then so are $\phi \wedge \psi$, and $\neg\phi$.
- If ϕ is a formula of LFP and x is a variable then, $\exists x \phi$ is a formula of LFP.

Semantics of LFP

Let $\mathbb{A} = (A, \mathcal{I})$ be a structure with universe A, and an interpretation $\mathcal{I}$ of a fixed vocabulary σ.

Let ϕ be a formula of LFP, and $\imath$ an interpretation in A of all the free variables (*first or second* order) of ϕ.

To each individual variable x, $\imath$ associates an element of A, and to each k-ary relation symbol R in ϕ that is not in σ, $\imath$ associates a relation $\imath(R) \subseteq A^k$.

$\imath$ is extended to terms t in the usual way.

For constants c, $\imath(c) = \mathcal{I}(c)$.

$\imath(f(t_1, \ldots, t_n)) = \mathcal{I}(f)(\imath(t_1), \ldots, \imath(t_n))$

Semantics of LFP

- If R is a relation symbol in σ, then $\imath(R) = \mathcal{I}(R)$.
- If P is a predicate expression of the form $\mathbf{lfp}_{R,\mathbf{x}}\phi$, then $\imath(P)$ is the relation that is the least fixed point of the monotone operator F on A^k defined by:

$$F(X) = \{\mathbf{a} \in A^k \mid \mathbb{A} \models \phi[\imath\langle X/R, \mathbf{x}/\mathbf{a}\rangle],$$

where $\imath\langle X/R, \mathbf{x}/\mathbf{a}\rangle$ denotes the interpretation $\imath'$ which is just like $\imath$ *except* that $\imath'(R) = X$, and $\imath'(\mathbf{x}) = \mathbf{a}$.

Semantics of LFP

- If ϕ is of the form $t_1 = t_2$, then $\mathbb{A} \models \phi[\imath]$ if, $\imath(t_1) = \imath(t_2)$.
- If ϕ is of the form $R(t_1, \ldots, t_k)$, then $\mathbb{A} \models \phi[\imath]$ if,

$$(\imath(t_1), \ldots, \imath(t_k)) \in \imath(R).$$

- If ϕ is of the form $\psi_1 \wedge \psi_2$, then $\mathbb{A} \models \phi[\imath]$ if, $\mathbb{A} \models \psi_1[\imath]$ *and* $\mathbb{A} \models \psi_2[\imath]$.
- If ϕ is of the form $\neg\psi$ then, $\mathbb{A} \models \phi[\imath]$ if, $\mathbb{A} \not\models \psi[\imath]$.
- If ϕ is of the form $\exists x\psi$, then $\mathbb{A} \models \phi[\imath]$ if there is an $a \in A$ such that $\mathbb{A} \models \psi[\imath\langle x/a\rangle]$.

Transitive Closure

The formula (with free variables u and v)

$$\theta \quad \equiv \quad \mathbf{lfp}_{T,xy}[(x = y \vee \exists z(E(x,z) \wedge T(z,y)))](u,v)$$

defines the *reflexive and transitive closure* of the relation E.

Thus $\forall u \forall v\ \theta$ defines *connectedness*.

The expressive power of LFP properly extends that of first-order logic.

Greatest Fixed Points

If ϕ is a formula in which the relation symbol R occurs *positively*, then the *greatest fixed point* of the monotone operator F_ϕ defined by ϕ can be defined by the formula:

$$\neg[\mathbf{lfp}_{R,\mathbf{x}}\neg\phi(R/\neg R)](\mathbf{x})$$

where $\phi(R/\neg R)$ denotes the result of replacing all occurrences of R in ϕ by $\neg R$.

Exercise: Verify!.

Simultaneous Inductions

We are given two formulas $\phi_1(S, T, \mathbf{x})$ and $\phi_2(S, T, \mathbf{y})$,
S is k-ary, T is l-ary.

The pair (ϕ_1, ϕ_2) can be seen as defining a map:

$$F : \mathsf{Pow}(A^k) \times \mathsf{Pow}(A^l) \to \mathsf{Pow}(A^k) \times \mathsf{Pow}(A^l)$$

If both formulas are positive in both S and T, then there is a least fixed point.

$$(P_1, P_2)$$

defined by *simultaneous induction* on $\mathbb{A}$.

Simultaneous Inductions

Theorem
For any pair of formulas $\phi_1(S, T)$ and $\phi_2(S, T)$ of LFP, in which the symbols S and T appear only positively, there are formulas ϕ_S and ϕ_T of LFP which, on any structure $\mathbb{A}$ containing at least two elements, define the two relations that are defined on $\mathbb{A}$ by ϕ_1 and ϕ_2 by simultaneous induction.

Proof

Assume $k \leq l$.

We define P, of arity $l + 2$ such that:

$(c, d, a_1, \ldots, a_l) \in P$ if, and only if, either $c = d$ and $(a_1, \ldots, a_k) \in P_1$ or $c \neq d$ and $(a_1, \ldots, a_l) \in P_2$

For new variables x_1 and x_2 and a new $l + 2$-ary symbol R, define ϕ_1' and ϕ_2' by replacing all occurrences of $S(t_1, \ldots, t_k)$ by:

$$x_1 = x_2 \wedge \exists y_{k+1}, \ldots, \exists y_l R(x_1, x_2, t_1, \ldots, t_k, y_{k+1}, \ldots, y_l),$$

and replacing all occurrences of $T(t_1, \ldots, t_l)$ by:

$$x_1 \neq x_2 \wedge R(x_1, x_2, t_1, \ldots, t_l).$$

Proof

Define ϕ as

$$(x_1 = x_2 \wedge \phi_1') \vee (x_1 \neq x_2 \wedge \phi_2').$$

Then,

$$(\mathbf{lfp}_{R, x_1 x_2 \mathbf{y}} \phi)(x, x, \mathbf{y})$$

defines P, so

$$\phi_S \equiv \exists x \exists y_{k+1}, \ldots, \exists y_l (\mathbf{lfp}_{R, x_1 x_2 \mathbf{y}} \phi)(x, x, \mathbf{y});$$

and

$$\phi_T \equiv \exists x_1 \exists x_2 (x_1 \neq x_2 \wedge \mathbf{lfp}_{R, x_1 x_2 \mathbf{y}} \phi)(x_1, x_2, \mathbf{y}).$$

Inflationary Fixed Points

We can associate with any formula $\phi(R, \mathbf{x})$ (even one that is not *monotone* in R an *inflationary operator*

$$IF_\phi(P) = P \cup F_\phi(P),$$

On any *finite* structure $\mathbb{A}$ the sequence

$$\begin{aligned} IF^0 &= \emptyset \\ IF^{n+1} &= IF_\phi(IF^n) \end{aligned}$$

converges to a limit IF^∞.

If F_ϕ is monotone, then this fixed point is, in fact, the least fixed point of F_ϕ.

IFP

We define the logic IFP with a syntax similar to LFP except, instead of the **lfp** rule, we have

> If R is a relation symbol of arity k, $\mathbf{x}$ is a tuple of variables of length k and ϕ is any formula of IFP, then
>
> $$\mathbf{ifp}_{R,\mathbf{x}}\phi$$
>
> is a predicate expression of IFP of arity k.

Semantics: we say that the predicate expression $\mathbf{ifp}_{R,\mathbf{x}}\phi$ denotes the relation that is the limit reached by the iteration of the inflationary operator IF_ϕ.

IFP

If ϕ defines a monotone operator, the relation defined by

$$\mathbf{ifp}_{R,\mathbf{x}}\phi$$

is the least fixed point of ϕ.

Thus, the *expressive power* of IFP is at least as great as that of LFP.

In fact, it is no greater:

Theorem (Gurevich-Shelah)
For every formula of ϕ of LFP, there is a predicate expression ψ of LFP such that, on any finite structure $\mathbb{A}$, ψ defines the same relation as $\mathbf{ifp}_{R,\mathbf{x}}\phi$.

Ranks

Let $\phi(R, \mathbf{x})$ be a formula defining an operator F_ϕ and IF_ϕ be the associated *inflationary* operator given by

$$IF_\phi(S) = S \cup F_\phi(S)$$

In a structure $\mathbb{A}$, we define for each $\mathbf{a} \in A^k$ a *rank* $|\mathbf{a}|_\phi$.

> The least n such that $\mathbf{a} \in IF^\alpha$, if there is such an n and ∞ otherwise.

Stage Comparison

We define the two *stage comparison* relations $\preceq$ and $\prec$ by:

$$\mathbf{a} \preceq \mathbf{b} \Leftrightarrow \mathbf{a} \in IF_\phi^\infty \wedge |\mathbf{a}|_\phi \leq |\mathbf{b}|_\phi;$$

$$\mathbf{a} \prec \mathbf{b} \Leftrightarrow |\mathbf{a}|_\phi < |\mathbf{b}|_\phi.$$

These two relations can themselves be defined in IFP.

Stage Comparison

$$\mathbf{a} \preceq \mathbf{b} \Leftrightarrow \mathbf{a} \in IF_\phi(\{\mathbf{b}' \mid \mathbf{b}' \prec \mathbf{b}\}).$$

$$\mathbf{a} \prec \mathbf{b} \Leftrightarrow \mathbf{b} \notin \{\mathbf{a}' \mid \neg(\mathbf{a} \preceq \mathbf{a}')\}.$$

Together, these give:

$$\mathbf{a} \preceq \mathbf{b} \Leftrightarrow \mathbf{a} \in IF_\phi(\{\mathbf{b}' \mid \mathbf{b} \notin \{\mathbf{a}' \mid \neg(\mathbf{a} \preceq \mathbf{b}')\}\}).$$

This is an inductive definition of $\preceq$.

A similar inductive definition is obtained from $\prec$.

Stage Comparison in LFP

In the inductive definition of $\preceq$:

$$\mathbf{a} \preceq \mathbf{b} \Leftrightarrow \mathbf{a} \in IF_\phi(\{\mathbf{b}' \mid \mathbf{b} \notin \{\mathbf{a}' \mid \neg(\mathbf{a} \preceq \mathbf{b}')\}\}).$$

we can replace the *negative* occurrences of $\mathbf{a} \preceq \mathbf{b}$ with $\neg(\mathbf{b} \prec \mathbf{a})$, and similarly, in the definition of $\prec$ replace negative occurrences of $\prec$ with positive occurrences of $\preceq$

as long as we can define the maximal rank

Maximal Rank

There is a formula $\mu(\mathbf{y})$, which defines the set of tuples of maximal rank.

$$IF_\phi(\{\mathbf{b} \mid \mathbf{b} \preceq \mathbf{a}\}) \subseteq IF_\phi(\{\mathbf{b} \mid \mathbf{b} \prec \mathbf{a}\}).$$

Replace the negative occurrence of $\mathbf{b} \preceq \mathbf{a}$ by $\neg(\mathbf{a} \prec \mathbf{b})$.

Reading List for this Part

1. Immerman. Chapter 4.
2. Libkin. Sections 10.2 and 10.3
3. Grädel et al. Section 2.6.
4. Ebbinghaus and Flum Sections 8.1 and 8.2.

Topics in Logic and Complexity
Part 10

Anuj Dawar

MPhil Advanced Computer Science, Lent 2012

Complexity of LFP

Any *query* definable in LFP is decidable by a *deterministic* machine in *polynomial time*.

To be precise, we can show, by induction on the structure of the formula $\phi(\mathbf{x})$ that for each formula ϕ there is a k such that

$$\mathbb{A} \models \phi[\mathbf{a}]$$

is decidable in time $O(n^t)$ where n is the number of elements of $\mathbb{A}$.

We prove this by induction on the structure of the formula.

Complexity of LFP

- Atomic formulas by direct lookup ($O(n^a)$ time, where a is the maximum arity of any predicate symbol in σ).
- Boolean connectives are easy.

  If $\mathbb{A} \models \phi_1$ can be decided in time $O(n^{t_1})$ and $\mathbb{A} \models \phi_2$ in time $O(n^{t_2})$, then $\mathbb{A} \models \phi_1 \land \phi_2$ can be decided in time $O(n^{\max(t_1,t_2)})$
- If $\phi \equiv \exists x\, \psi$ then for each $a \in \mathbb{A}$ check whether

  $$(\mathbb{A}, c \mapsto a) \models \psi[c/x],$$

  where c is a new constant symbol. If $\mathbb{A} \models \psi$ can be decided in time $O(n^t)$, then $\mathbb{A} \models \phi$ can be decided in time $O(n^{t+1})$.

Complexity of LFP

Suppose $\phi \equiv \mathbf{lfp}_{R,\mathbf{x}}\psi(\mathbf{t})$ (R is l-ary)

To decide $\mathbb{A} \models \phi[\mathbf{a}]$:

$R := \emptyset$
for $i := 1$ **to** n^l **do**
  $R := F_\psi(R)$
end
if $\mathbf{a} \in R$ **then** accept **else** reject

Complexity of LFP

To compute $F_\psi(R)$

For every tuple $\mathbf{a} \in A^l$, determine whether $(\mathbb{A}, R) \models \psi[\mathbf{a}]$.

If deciding $(\mathbb{A}, R) \models \psi$ takes time $O(n^t)$, then each assignment to R inside the loop requires time $O(n^{l+t})$. The total time taken to execute the loop is then $O(n^{2l+t})$. Finally, the last line can be done by a search through R in time $O(n^l)$. The total running time is, therefore, $O(n^{2l+t})$.

The *space* required is $O(n^l)$.

Capturing P

For any ϕ of LFP, the language $\{[\mathbb{A}]_< \mid \mathbb{A} \models \phi\}$ is in P.

Suppose ρ is a signature that contains a *binary relation symbol* $<$, possibly along with other symbols.

Let $\mathcal{O}_\rho$ denote those structures $\mathbb{A}$ in which $<$ is a *linear order* of the universe.

For any language $L \in$ P, there is a sentence ϕ of LFP that defines the class of structures

$$\{\mathbb{A} \in \mathcal{O}_\rho \mid [\mathbb{A}]_{<^\mathbb{A}} \in L\}$$

(Immerman; Vardi 1982)

Capturing P

Recall the proof of *Fagin's Theorem*, that ESO captures NP.

Given a machine M and an integer k, there is a *first-order* formula $\phi_{M,k}$ such that

$$\mathbb{A} \models \exists < \exists T_{\sigma_1} \cdots T_{\sigma_s} \exists S_{q_1} \cdots S_{q_m} \exists H\, \phi_{M,k}$$

if, and only if, M accepts $[\mathbb{A}]_<$ in time n^k, for some order $<$.

If we *fix* the order $<$ as part of the structure $\mathbb{A}$, we do not need the outermost quantifier.

Moreover, for a *deterministic* machine M, the relations $T_{\sigma_1} \ldots T_{\sigma_s}, S_{q_1} \ldots S_{q_m}, H$ can be defined *inductively*.

Capturing P

For any ϕ of LFP, the language $\{[\mathbb{A}]_< \mid \mathbb{A} \models \phi\}$ is in P.

Suppose ρ is a signature that contains a *binary relation symbol* $<$, possibly along with other symbols.

Let $\mathcal{O}_\rho$ denote those structures $\mathbb{A}$ in which $<$ is a *linear order* of the universe.

For any language $L \in$ P, there is a sentence ϕ of LFP that defines the class of structures

$$\{\mathbb{A} \in \mathcal{O}_\rho \mid [\mathbb{A}]_{<^{\mathbb{A}}} \in L\}$$

(Immerman; Vardi 1982)

Capturing P

Recall the proof of *Fagin's Theorem*, that ESO captures NP.

Given a machine M and an integer k, there is a *first-order* formula $\phi_{M,k}$ such that

$$\mathbb{A} \models \exists < \exists T_{\sigma_1} \cdots T_{\sigma_s} \exists S_{q_1} \cdots S_{q_m} \exists H\, \phi_{M,k}$$

if, and only if, M accepts $[\mathbb{A}]_<$ in time n^k, for some order $<$.

If we *fix* the order $<$ as part of the structure $\mathbb{A}$, we do not need the outermost quantifier.

Moreover, for a *deterministic* machine M, the relations $T_{\sigma_1} \ldots T_{\sigma_s}, S_{q_1} \ldots S_{q_m}, H$ can be defined *inductively*.

Capturing P

$$\begin{aligned}
&\mathsf{Tape}_a(\mathbf{x},\mathbf{y}) \Leftrightarrow \\
&(\mathbf{x} = \mathbf{1} \wedge \mathrm{Init}_a(\mathbf{y}))\vee \\
&\;\exists\mathbf{t}\exists\mathbf{h} \bigvee\nolimits_q \quad \big(\mathbf{x} = \mathbf{t}+1 \wedge \mathsf{State}_q(\mathbf{t},\mathbf{h})\wedge \\
&\qquad\qquad [(\mathbf{h} = \mathbf{y} \wedge \bigvee\nolimits_{\{b,d,q' \mid \Delta(q,b,q',a,d)\}} \mathsf{Tape}_b(\mathbf{t},\mathbf{y})\vee \\
&\qquad\qquad \mathbf{h} \neq \mathbf{y} \wedge \mathsf{Tape}_a(\mathbf{t},\mathbf{y})]\big);
\end{aligned}$$

where $\mathrm{Init}_a(\mathbf{y})$ is the formula that defines the positions in which the symbol a appears in the input.

Capturing P

$$\begin{aligned}
&\mathsf{State}_q(\mathbf{x},\mathbf{y}) \Leftrightarrow \\
&(\mathbf{x} = \mathbf{1} \wedge \mathbf{y} = \mathbf{1} \wedge q = q_0)\vee \\
&\;\exists\mathbf{t}\exists\mathbf{h} \quad \bigvee\nolimits_{\{a,b,q' \mid \Delta(q',a,q,b,R)\}} \quad \big(\mathbf{x} = \mathbf{t}+1 \wedge \mathsf{State}_{q'}(\mathbf{t},\mathbf{h})\wedge \\
&\qquad\qquad\qquad\qquad\qquad \mathsf{Tape}_a(\mathbf{t},\mathbf{h}) \wedge \mathbf{y} = \mathbf{h}+1)\big) \\
&\qquad\quad \bigvee\nolimits_{\{a,b,q' \mid \Delta(q',a,q,b,L)\}} \quad \big(\mathbf{x} = \mathbf{t}+1 \wedge \mathsf{State}'_q(\mathbf{t},\mathbf{h})\wedge \\
&\qquad\qquad\qquad\qquad\qquad \mathsf{Tape}_a(\mathbf{t},\mathbf{h}) \wedge \mathbf{h} = \mathbf{y}+1)\big).
\end{aligned}$$

Unordered Structures

In the absence of an *order relation*, there are properties in P that are not definable in LFP.

There is no sentence of LFP which defines the structures with an *even* number of elements.

Evenness

Let $\mathcal{E}$ be the collection of all structures in the empty signature.

In order to prove that *evenness* is not defined by any LFP sentence, we show the following.

Lemma
For every LFP formula ϕ there is a first order formula ψ, such that for all structures $\mathbb{A}$ in $\mathcal{E}$, $\mathbb{A} \models (\phi \leftrightarrow \psi)$.

Unordered Structures

Let $\psi(\mathbf{x}, \mathbf{y})$ be a first order formula.

$\mathbf{lfp}_{R,\mathbf{x}}\psi$ defines the relation

$$F^{\infty}_{\psi,\mathbf{b}} = \bigcup_{i \in \mathbb{N}} F^{i}_{\psi,\mathbf{b}}$$

for a fixed interpretation of the variables $\mathbf{y}$ by the tuple of parameters $\mathbf{b}$.

For each i, there is a first order formula ψ^i such that on any structure $\mathbb{A}$,

$$F^{i}_{\psi,\mathbf{b}} = \{\mathbf{a} \mid \mathbb{A} \models \psi^i[\mathbf{a}, \mathbf{b}]\}.$$

Defining the Stages

These formulas are obtained by *induction*.

ψ^1 is obtained from ψ by replacing all occurrences of subformulas of the form $R(\mathbf{t})$ by $t \neq t$.

ψ^{i+1} is obtained by replacing in ψ, all subformulas of the form $R(\mathbf{t})$ by $\psi^i(\mathbf{t}, \mathbf{y})$

Let $\mathbf{b}$ be an l-tuple, and $\mathbf{a}$ and $\mathbf{c}$ two k-tuples in a structure $\mathbb{A}$ such that

there is an automorphism $\imath$ *of* $\mathbb{A}$ (i.e. an *isomorphism* from $\mathbb{A}$ to itself) such that

- $\imath(\mathbf{b}) = \mathbf{b}$
- $\imath(\mathbf{a}) = \mathbf{c}$

Then,

$$\mathbf{a} \in F^i_{\psi,\mathbf{b}} \quad \text{if, and only if,} \quad \mathbf{c} \in F^i_{\psi,\mathbf{b}}.$$

Bounding the Induction

This defines an *equivalence relation* $\mathbf{a} \sim_{\mathbf{b}} \mathbf{c}$.

If there are p distinct equivalence classes, then

$$F^\infty_{\psi,\mathbf{b}} = F^p_{\psi,\mathbf{b}}$$

In $\mathcal{E}$ there is a uniform bound p, that does not depend on the size of the structure.

Reading List for this Part

1. Libkin. Chapter 10.
2. Grädel et al. Section 3.3.

Topics in Logic and Complexity
Part 11

Anuj Dawar

MPhil Advanced Computer Science, Lent 2013

Complexity of First-Order Logic

The problem of deciding whether $\mathbb{A} \models \phi$ for first-order ϕ is in time $O(ln^m)$ and $O(m \log n)$ space.

where n is the size of $\mathbb{A}$, l is the length of ϕ and m is the quantifier rank of ϕ.

We have seen that the problem is PSPACE-complete, even for fixed $\mathbb{A}$.

For each fixed ϕ, the problem is in L.

Is FO contained in an initial segment of P?

> Is there a fixed c such that for every first-order ϕ, $\mathrm{Mod}(\phi)$ is decidable in time $O(n^c)$?

If P = PSPACE, then the answer is yes, as the satisfaction relation is then itself decidable in time $O(n^c)$.

> Thus, though we expect the answer is no, this would be difficult to prove.

A more uniform version of the question is:

> Is there a constant c and a computable function f so that the satisfaction relation for first-order logic is decidable in time $O(f(l)n^c)$?

In this case we say that the satisfaction problem is *fixed-parameter tractable* (FPT) with the formula length as parameter.

Parameterized Problems

Some problems are given a graph G and a positive integer k

Independent Set: does G contain k vertices that are pairwise distinct and non-adjacent?

Dominating Set: does G contain k vertices such that every vertex is among them or adjacent to one of them?

Vertex Cover: does G contain k vertices such that every edge is incident on one of them?

For each fixed value of k, there is a first-order sentence ϕ_k such that $G \models \phi_k$ if, and only if, G contains an independent set of k vertices.

Similarly for dominating set and vertex cover.

Parameterized Complexity

FPT—the class of problems of input size n and *parameter* l which can be solved in time $O(f(l)n^c)$ for some computable function f and constant c.

There is a hierarchy of *intractable* classes.

$$\mathsf{FPT} \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq \mathsf{AW}[\star]$$

Vertex Cover is FPT.

Independent Set is $W[1]$-complete.

Dominating Set is $W[2]$-complete.

Parameterized Complexity of First-Order Satisfaction

Writing Π_t for those formulas which, in prenex normal form have t alternating blocks of quantifiers starting with a universal block:

> The satisfaction problem restricted to Π_t formulas (parameterized by the length of the formula) is hard for the class $W[t]$.

The satisfaction relation for first-order logic ($\mathbb{A} \models \phi$), parameterized by the length of ϕ is AW[$\star$]-complete.

Thus, if the satisfaction problem for first-order logic were FPT, this would collapse the edifice of parameterized complexity theory.

Restricted Classes

One way to get a handle on the complexity of first-order satisfaction is to consider restricted classes of structures.

> Given: a first-order formula ϕ and a structure $\mathbb{A} \in \mathcal{C}$
> Decide: if $\mathbb{A} \models \phi$

For many interesting classes $\mathcal{C}$, this problem has been shown to be FPT, even for formulas of MSO.

We say that satisfaction of FO (or MSO) is *fixed-paramter tractable on $\mathcal{C}$*.

Words as Relational Structures

For an alphabet $\Sigma = \{a_1, \ldots, a_s\}$ let

$$\sigma_\Sigma = (<, P_{a_1}, \ldots, P_{a_s})$$

where

> $<$ is binary; and $P_{a_1}, \ldots, P_{a_s}$ are unary.

With each $w \in \Sigma^*$ we associate the canonical structure

$$S_w = (\{1, \ldots, n\}, <, P_{a_1}, \ldots, P_{a_s})$$

where

- n is the length of w
- $<$ is the natural linear order on $\{1, \ldots, n\}$.
- $i \in P_a$ if, and only if, the ith symbol in w is a.

Languages Defined by Formulas

The formula ϕ in the signature σ_Σ defines:

$$\{w \mid S_w \models \phi\}.$$

The class of structures isomorphic to word models is given by:

$$lo(<) \wedge \forall x \bigvee_{a \in A} P_a(x) \wedge \forall x \bigwedge_{a,b \in A, a \neq b} (P_a(x) \rightarrow \neg P_b(x)),$$

where

> $lo(<)$ is the formula that states that $<$ is a linear order

Examples

The set of strings of length 3 or more:

$$\exists x \exists y \exists z (x \neq y \wedge y \neq z \wedge z \neq z).$$

The set of strings which begin with an a:

$$\exists x (P_a(x) \wedge \forall y y \geq x)$$

The set of strings of even length:

$$\begin{array}{lll} \exists X & \forall x(\forall y & y \leq x) \rightarrow X(x) \wedge \\ & \forall x \forall y & (x < y \wedge \forall z (z \leq x \vee y \leq z)) \\ & & \rightarrow (X(x) \leftrightarrow \neg X(y)) \wedge \\ & \forall x(\forall y & x \leq y) \rightarrow \neg X(x). \end{array}$$

Examples

$(ab)^*$:

$$\begin{array}{ll} \forall x(\forall y & y \leq x) \rightarrow P_a(x) \wedge \\ \forall x \forall y & (x < y \wedge \forall z (z \leq x \vee y \leq z)) \\ & \rightarrow (P_a(x) \leftrightarrow P_b(y)) \wedge \\ \forall x(\forall y & x \leq y) \rightarrow P_b(x). \end{array}$$

MSO on Words

Theorem (Büchi-Elgot-Trakhtenbrot)
A language L is defined by a sentence of MSO if, and only if, L is regular.

Recall that a language L is *regular* if:

- it is the set of words matching a *regular expression*; or equivalently
- it is the set of words accepted by some *nondeterministic finite automaton*; or equivalently
- it is the set of words accepted by some *deterministic finite automaton*.

Myhill-Nerode Theorem

Let $\sim$ be an equivalence relation on Σ^*.

We say $\sim$ is *right invariant* if, for all $u, v \in \Sigma^*$,

if $u \sim v$, then for all $w \in \Sigma^*$, $uw \sim vw$.

Theorem (Myhill-Nerode)
The following are equivalent for any language $L \subseteq \Sigma^*$:

- L is regular;
- L is the union of equivalence classes of a right invariant equivalence relation of finite index on Σ^*.

MSO Equivalence

We write $\mathbb{A} \equiv_m^{\mathsf{MSO}} \mathbb{B}$ to denote that, for all MSO sentences ϕ with $\mathrm{qr}(\phi) \leq m$,

$$\mathbb{A} \models \phi \quad \text{if, and only if,} \quad \mathbb{B} \models \phi.$$

We count both first and second order quantifiers towards the rank.

The relation $\equiv_m^{\mathsf{MSO}}$ has finite index for every m.

For any m, there are up to logical equivalence, only finitely many formulas with quantifier rank at most m, with at most k free variables.

Invariance

Suppose u_1, u_2, v_1, v_2 are words over an alphabet Σ such that

$$u_1 \equiv_m^{\mathsf{MSO}} u_2 \quad \text{and} \quad v_1 \equiv_m^{\mathsf{MSO}} v_2$$

then $u_1 \cdot v_1 \equiv_m^{\mathsf{MSO}} u_2 \cdot v_2$.

Dulpicator has a winning strategy on the game played on the pair of words $u_1 \cdot v_1, u_2 \cdot v_2$ that is obtained as a composition of its strategies in the games on u_1, u_2 and v_1, v_2.

It follows that $\equiv_m^{\mathsf{MSO}}$ is *right invariant*.

For any MSO sentence ϕ, the language defiend by ϕ is the union of equivalence classes of $\equiv_m^{\mathsf{MSO}}$ where m is the quantifier rank of ϕ.

Regular Expressions to MSO

For the converse, we translate a regular expression r to an MSO sentence ϕ_r.

$r = \emptyset$: $\phi_r = \exists x(x \neq x)$.

$r = \varepsilon$: $\phi_r = \neg\exists x(x = x)$.

$r = a$: $\phi_r = \exists x \forall y(y = x \wedge P_a(x))$.

$r = s + t$: $\phi_r = \phi_s \vee \psi_t$.

$r = st$: $\phi_r = \exists x(\phi_s^{<x} \wedge \phi_t^{\geq x})$,

where $\phi_s^{<x}$ and $\phi_t^{\geq x}$ are obtained from ϕ_s and ϕ_t by relativising the first order quantifiers.

That is, every subformula of ϕ_s of the form $\exists y \psi$ is replaced by $\exists y(y < x \wedge \psi^{<x})$,

and similarly every subformula $\exists y \psi$ of ϕ_t by $\exists y(y \geq x \wedge \psi^{\geq x})$

Kleene Star

$r = s^*$:

$$\begin{aligned}
\phi_r = \quad & \phi_\varepsilon \vee \\
& \exists X \ \forall x(X(x) \wedge \forall y(y < x \to \neg X(y)) \to \phi_s^{<x}) \wedge \\
& \qquad \forall x(X(x) \wedge \forall y(y \geq x \to \neg X(y)) \to \phi_s^{\geq x}) \wedge \\
& \qquad \forall x \forall y \ (X < y \wedge X(x) \wedge X(y) \wedge \\
& \qquad\qquad \forall z(x < z \wedge z < y \to \neg X(z)) \\
& \qquad\qquad \to \phi_s^{\geq x, <y}),
\end{aligned}$$

where $\phi_s^{\geq x, <y}$ is obtained from ϕ_s by relativising all first order quantifiers simultaneously with $< y$ and $\geq x$.

First-Order Languages

The class of *star-free* regular expressions is defined by:

- the strings $\emptyset$ and ε are star-free regular expressions;
- for any element $a \in A$, the string a is a star-free regular expression;
- if r and s are star-free regular expressions, then so are (rs), $(r+s)$ and $(\bar{r})$.

A language is defined by a first order sentence *if, and only if,* it is denoted by a star-free regular expression.

Applications

A class of linear orders is definable by a sentence of MSO if, and only if, its set of cardinalities is *eventually periodic*.

Some results on graphs:

The class of balanced bipartite graphs is not definable in MSO.

The class of Hamiltonian graphs is not definable by a sentence of MSO.

Reading List for this Handout

1. Libkin. Sections 7.4 and 7.5
2. Ebbinghaus, Flum Chapter 6

Topics in Logic and Complexity
Part 12

Anuj Dawar

MPhil Advanced Computer Science, Lent 2013

MSO is FPT on Words

There is a computable function f such that the problem of deciding, given a word w and an MSO sentence ϕ whether,

$$S_w \models \phi$$

can be decided in time $O(f(l)n)$ where l is the length of ϕ and n is the length of w.

The algorithm proceeds by constructing, from ϕ an *automaton* $\mathcal{A}_\phi$ such that the language recognized by $\mathcal{A}_\phi$ is

$$\{w \mid S_w \models \phi\}$$

then running $\mathcal{A}_\phi$ on w.

The automaton $\mathcal{A}_\phi$

The states of $\mathcal{A}_\phi$ are the equivalence classes of $\equiv_m^{\mathsf{MSO}}$ where m is the quantifier rank of ϕ.

We write $\mathsf{Type}_m^{\mathsf{MSO}}(\mathbb{A})$ for the set of all formulas ϕ with $\mathrm{qr}(\phi) \leq m$ such that $\mathbb{A} \models \phi$.

$\mathbb{A} \equiv_m^{\mathsf{MSO}} \mathbb{B}$ is equivalent to

$$\mathsf{Type}_m^{\mathsf{MSO}}(\mathbb{A}) = \mathsf{Type}_m^{\mathsf{MSO}}(\mathbb{B})$$

There is a single formula $\theta_{\mathbb{A}}$ that characterizes $\mathsf{Type}_m^{\mathsf{MSO}}(\mathbb{A})$.

It turns out that we can compute $\theta_{S_{w \cdot a}}$ from θ_{S_w}.

Trees

An (undirected) *forest* is an *acyclic* graph and a *tree* is a connected forest.

We next aim to show that there is an algorithm that decides, given a tree T and an MSO sentence ϕ whether

$$T \models \phi$$

and runs in time $O(f(l)n$ where l is the length of ϕ and n is the size of T.

Rooted Directed Trees

A *rooted, directed tree* (T, a) is a directed graph with a distinguished vertex a such that for every vertex v there is a *unique* directed path from a to v.

We will actually see that MSO satisfaction is FPT for rooted, directed trees.

The result for undirected trees follows, as any undirected tree can be turned into a rooted directed one by choosing any vertex as a root and directing edges away from it.

Sums of Rooted Trees

Given rooted, directed trees (T, a) and (S, b) we define the sum

$$(T, a) \oplus (S, b)$$

to be the rooted directed tree which is obtained by taking the *disjoint union* of the two trees while *identifying* the roots.

That is,

- the set of vertices of $(T, a) \oplus (S, b)$ is $V(T) \uplus V(S) \setminus \{b\}$.
- the set of edges is $E(T) \cup E(S) \cup \{(a, v) \mid (b, v) \in E(S)\}$.

Congruence

If $(T_1, a_1) \equiv_m^{\mathsf{MSO}} (T_2, a_2)$ and $(S_1, b_1) \equiv_m^{\mathsf{MSO}} (S_2, b_2)$, then

$$(T_1, a_1) \oplus (S_1, b_1) \equiv_m^{\mathsf{MSO}} (T_2, a_2) \oplus (S_2, b_2).$$

This can be proved by an application of Ehrenfeucht games.

Moreover (though we skip the proof), $\mathsf{Type}_m^{\mathsf{MSO}}((T, a) \oplus (S, b))$ can be computed from $\mathsf{Type}_m^{\mathsf{MSO}}((T, a))$ and $\mathsf{Type}_m^{\mathsf{MSO}}((S, b))$.

Add Root

For any rooted, directed tree (T, a) define $r(T, a)$ to be rooted directed tree obtained by adding to (T, a) a new vertex, which is the root and whose only child is a.

That is,

- the vertices of $r(T, a)$ are $V(T) \cup \{a'\}$ where a' is not in $V(T)$;
- the root of $r(T, a)$ is a'; and
- the edges of $r(T, a)$ are $E(T) \cup \{(a', a)\}$.

Again, $\mathsf{Type}_m^{\mathsf{MSO}}(r(T, a))$ can be computed from $\mathsf{Type}_m^{\mathsf{MSO}}(T, a)$.

MSO satisfaction is FPT on Trees

Any *rooted, directed tree* (T, a) can be obtained from *singleton trees* by a sequence of applications of $\oplus$ and r.

The length of the sequence is linear in the size of T.

We can compute $\mathsf{Type}_m^{\mathsf{MSO}}(T, a)$ in linear time.

The Method of Decompositions

Suppose $\mathcal{C}$ is a class of graphs such that there is a finite class $\mathcal{B}$ and a finite collection Op of operations such that:

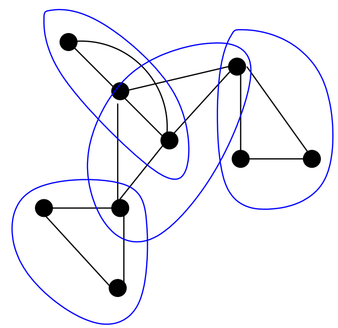
- $\mathcal{C}$ is contained in the closure of $\mathcal{B}$ under the operations in Op;
- there is a polynomial-time algorithm which computes, for any $G \in \mathcal{C}$, an Op-decomposition of G over $\mathcal{B}$; and
- for each m, the equivalence class $\equiv_m^{\mathsf{MSO}}$ is an *effective* congruence with respect to to all operations $o \in \mathsf{Op}$ (i.e., the $\equiv_m^{\mathsf{MSO}}$-type of $o(G_1, \ldots, G_s)$ can be computed from the $\equiv_m^{\mathsf{MSO}}$-types of $G_1, \ldots, G_s$).

Then, MSO satisfaction is fixed-parameter tractable on $\mathcal{C}$.

Treewidth

The *treewidth* of an undirected graph is a measure of how tree-like the graph is.

A graph has treewidth k if it can be covered by subgraphs of at most $k+1$ nodes in a tree-like fashion.



This gives a *tree decomposition* of the graph.

Treewidth

Treewidth is a measure of how *tree-like* a graph is.

For a graph $G = (V, E)$, a *tree decomposition* of G is a relation $D \subset V \times T$ with a tree T such that:

- for each $v \in V$, the set $\{t \mid (v, t) \in D\}$ forms a connected subtree of T; and
- for each edge $(u, v) \in E$, there is a $t \in T$ such that $(u, t), (v, t) \in D$.

The *treewidth* of G is the least k such that there is a tree T and a tree decomposition $D \subset V \times T$ such that for each $t \in T$,

$$|\{v \in V \mid (v, t) \in D\}| \leq k + 1.$$

Dynamic Programming

It has long been known that graphs of small treewidth admit efficient *dynamic programming* algorithms for intractable problems.

In general, these algorithms proceed bottom-up along a tree decomposition of G.

At any stage, a small set of vertices form the "*interface*" to the rest of the graph.

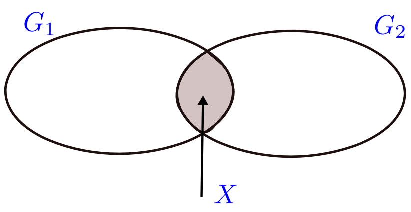
This allows a recursive decomposition of the problem.

Treewidth

Looking at the decomposition *bottom-up*, a graph of treewidth k is obtained from graphs with at most $k+1$ nodes through a finite sequence of applications of the operation of taking *sums over sets* of at most k elements.

$G_1 \oplus_X G_2$

$|X| \leq k$



We let $\mathcal{T}_k$ denote the class of graphs G such that $\mathrm{tw}(G) \leq k$.

Treewidth

More formally,

Consider graphs with up to $k+1$ distinguished vertices $C = \{c_0, \ldots, c_k\}$.

Define a *merge* operation $(G \oplus_C H)$ that forms the union of G and H *disjointly apart from* C.

Also define $\mathsf{erase}_i(G)$ that erases the name c_i.

Then a graph G is in $\mathcal{T}_k$ if it can be formed from graphs with at most $k+1$ vertices through a sequence of such operations.

Congruence

- Any $G \in \mathcal{T}_k$ is obtained from $\mathcal{B}_k$ by finitely many applications of the operations erase_i and $\oplus_C$.
- If $G_1, \rho_1 \equiv_m^{\mathsf{MSO}} G_2, \rho_2$, then

$$\mathsf{erase}_i(G_1, \rho_1) \equiv_m^{\mathsf{MSO}} \mathsf{erase}_i(G_2, \rho_2)$$

- If $G_1, \rho_1 \equiv_m^{\mathsf{MSO}} G_2, \rho_2$, and $H_1, \sigma_1 \equiv_m^{\mathsf{MSO}} H_2, \sigma_2$ then

$$(G_1, \rho_1) \oplus_C (H_1, \sigma_1) \equiv_m^{\mathsf{MSO}} (G_2, \rho_2) \oplus_C (H_2, \sigma_2)$$

Note: a special case of this is that $\equiv_m^{\mathsf{MSO}}$ is a congruence for *disjoint union* of graphs.

Courcelle's Theorem

Theorem (Courcelle)
For any MSO sentence ϕ and any k there is a linear time algorithm that decides, given $G \in \mathcal{T}_k$ whether $G \models \phi$.

Given $G \in \mathcal{T}_k$ and ϕ, compute:

- from G a labelled tree T; and
- from ϕ a bottom-up tree automaton $\mathcal{A}$

such that $\mathcal{A}$ accepts T if, and only if, $G \models \phi$.

Bounded Degree Graphs

In a graph $G = (V, E)$ the *degree* of a vertex $v \in V$ is the number of neighbours of v, i.e.

$$|\{u \in V \mid (u, v) \in E\}|.$$

We write $\delta(G)$ for the *smallest* degree of any vertex in G.

We write $\Delta(G)$ for the *largest* degree of any vertex in G.

$\mathcal{D}_k$—the class of graphs G with $\Delta(G) \leq k$.

Bounded Degree Graphs

Theorem (Seese)
For every sentence ϕ of FO and every k there is a linear time algorithm which, given a graph $G \in \mathcal{D}_k$ determines whether $G \models \phi$.

A proof is based on *locality* of first-order logic.

To be precise a strengthening of *Hanf's theorem*.

Note: this is not true for MSO unless P = NP.
Construct, for any graph G, a graph G' such that $\Delta(G') \leq 5$ and G' is 3-colourable iff G is, and the map $G \mapsto G'$ is polynomial-time computable.

Hanf Types

For an element a in a structure $\mathbb{A}$, define

$N_r^{\mathbb{A}}(a)$—the substructure of $\mathbb{A}$ generated by the elements whose distance from a (in $G\mathbb{A}$) is at most r.

We say $\mathbb{A}$ and $\mathbb{B}$ are *Hanf equivalent* with radius r and threshold q ($\mathbb{A} \simeq_{r,q} \mathbb{B}$) if, for every $a \in A$ the two sets

$$\{a' \in a \mid N_r^{\mathbb{A}}(a) \cong N_r^{\mathbb{A}}(a')\} \quad \text{and} \quad \{b \in B \mid N_r^{\mathbb{A}}(a) \cong N_r^{\mathbb{B}}(b)\}$$

either have the same size or both have size greater than q;

and, similarly for every $b \in B$.

Hanf Locality Theorem

Theorem (Hanf)
For every vocabulary σ and every m there are $r \leq 3^m$ and $q \leq m$ such that for any σ-structures $\mathbb{A}$ and $\mathbb{B}$: if $\mathbb{A} \simeq_{r,q} \mathbb{B}$ then $\mathbb{A} \equiv_m \mathbb{B}$.

In other words, if $r \geq 3^m$, the equivalence relation $\simeq_{r,m}$ is a refinement of $\equiv_m$.

For $\mathbb{A} \in \mathcal{D}_k$:

$N_r^{\mathbb{A}}(a)$ has at most $k^r + 1$ elements

each $\simeq_{r,m}$ has finite index.

Each $\simeq_{r,m}$-class t can be characterised by a finite table, I_t, giving isomorphism types of neighbourhoods and numbers of their occurrences up to threshold m.

Satisfaction on $\mathcal{D}_k$

For a sentence ϕ of FO, we can compute a set of tables $\{I_1, \ldots, I_s\}$ describing $\simeq_{r,m}$-classes consistent with it.

This computation is independent of any structure $\mathbb{A}$.

Given a structure $\mathbb{A} \in \mathcal{D}_k$,

for each a, determine the isomorphism type of $N_r^{\mathbb{A}}(a)$

construct the table describing the $\simeq_{r,m}$-class of $\mathbb{A}$.

compare against $\{I_1, \ldots, I_s\}$ to determine whether $\mathbb{A} \models \phi$.

For fixed k, r, m, this requires time *linear* in the size of $\mathbb{A}$.

Note: satisfaction for FO is in $O(f(l,k)n)$.

Reading List for this Handout

1. Libkin. Sections 7.6 and 7.7