Topics in Logic and Complexity

Handout 4

Anuj Dawar

http://www.cl.cam.ac.uk/teaching/2122/L15
Expressive Power of Logics

We have seen that the expressive power of first-order logic, in terms of computational complexity is weak. Second-order logic allows us to express all properties in the polynomial hierarchy.

Are there interesting logics intermediate between these two? We have seen one—monadic second-order logic. We now examine another—LFP—the logic of least fixed points.
LFP is a logic that formalises *inductive definitions.*

*Unlike in second-order logic, we cannot quantify over arbitrary relations, but we can build new relations inductively.*

Inductive definitions are pervasive in mathematics and computer science. The *syntax* and *semantics* of various formal languages are typically defined inductively.

*viz.* the definitions of the syntax and semantics of first-order logic seen earlier.
The transitive closure of a binary relation $E$ is the smallest relation $T$ satisfying:

- $E \subseteq T$; and
- if $(x, y) \in T$ and $(y, z) \in E$ then $(x, z) \in T$.

This constitutes an inductive definition of $T$ and, as we have already seen, there is no first-order formula that can define $T$ in terms of $E$. 

Transitive Closure
In order to introduce LFP, we briefly look at the theory of monotone operators, in our restricted context.

We write \( \text{Pow}(A) \) for the powerset of \( A \).

An operator on \( A \) is a function

\[
F : \text{Pow}(A) \rightarrow \text{Pow}(A).
\]

\( F \) is monotone if

if \( S \subseteq T \), then \( F(S) \subseteq F(T) \).
A **fixed point** of $F$ is any set $S \subseteq A$ such that $F(S) = S$.

$S$ is the **least fixed point** of $F$, if for all fixed points $T$ of $F$, $S \subseteq T$.

$S$ is the **greatest fixed point** of $F$, if for all fixed points $T$ of $F$, $T \subseteq S$. 
Least and Greatest Fixed Points

For any monotone operator $F$, define the collection of its pre-fixed points as:

$$Pre = \{ S \subseteq A \mid F(S) \subseteq S \}.$$  

*Note:* $A \in Pre$.

Taking

$$L = \bigcap Pre,$$

we can show that $L$ is a fixed point of $F$. 
Fixed Points

For any set $S \in \text{Pre}$,

- $L \subseteq S$ \hspace{1cm} \text{by definition of } L.
- $F(L) \subseteq F(S)$ \hspace{1cm} \text{by monotonicity of } F.
- $F(L) \subseteq S$ \hspace{1cm} \text{by definition of } \text{Pre}.
- $F(L) \subseteq L$ \hspace{1cm} \text{by definition of } L.
- $F(F(L)) \subseteq F(L)$ \hspace{1cm} \text{by monotonicity of } F.
- $F(L) \in \text{Pre}$ \hspace{1cm} \text{by definition of } \text{Pre}.
- $L \subseteq F(L)$ \hspace{1cm} \text{by definition of } L.
Least and Greatest Fixed Points

\( L \) is a \textit{fixed point} of \( F \).

Every fixed point \( P \) of \( F \) is in \( Pre \), and therefore \( L \subseteq P \).

Thus, \( L \) is the least fixed point of \( F \)

Similarly, the greatest fixed point is given by:

\[
G = \bigcup \{ S \subseteq A \mid S \subseteq F(S) \}.
\]
Iteration

Let $A$ be a *finite* set and $F$ be a *monotone* operator on $A$.
Define for $i \in \mathbb{N}$:

\[
\begin{align*}
F^0 &= \emptyset \\
F^{i+1} &= F(F^i).
\end{align*}
\]

For each $i$, $F^i \subseteq F^{i+1}$ (proved by induction).
Iteration

Proof by induction.

\[ \emptyset = F^0 \subseteq F^1. \]

If \( F^i \subseteq F^{i+1} \) then, by monotonicity

\[ F(F^i) \subseteq F(F^{i+1}) \]

and so \( F^{i+1} \subseteq F^{i+2} \).
Fixed-Point by Iteration

If $A$ has $n$ elements, then

$$F^n = F^{n+1} = F^m \quad \text{for all} \quad m > n$$

Thus, $F^n$ is a fixed point of $F$.

Let $P$ be any fixed point of $F$. We can show by induction on $i$, that $F^i \subseteq P$.

$$F^0 = \emptyset \subseteq P$$

If $F^i \subseteq P$ then

$$F^{i+1} = F(F^i) \subseteq F(P) = P.$$ 

Thus $F^n$ is the least fixed point of $F$. 
Defined Operators

Suppose $\phi$ contains a relation symbol $R$ (of arity $k$) not interpreted in the structure $A$ and let $x$ be a tuple of $k$ free variables of $\phi$.

For any relation $P \subseteq A^k$, $\phi$ defines a new relation:

$$F_P = \{ a \mid (A, P) \models \phi[a] \}.$$

The operator $F_\phi : \text{Pow}(A^k) \rightarrow \text{Pow}(A^k)$ defined by $\phi$ is given by the map

$$P \mapsto F_P.$$

Or, $F_{\phi, b}$ if we fix parameters $b$. 
**Definition**
A formula $\phi$ is *positive* in the relation symbol $R$, if every occurrence of $R$ in $\phi$ is within the scope of an even number of negation signs.

**Lemma**
For any structure $A$ not interpreting the symbol $R$, any formula $\phi$ which is positive in $R$, and any tuple $b$ of elements of $A$, the operator $F_{\phi,b} : \text{Pow}(A^k) \to \text{Pow}(A^k)$ is monotone.
Syntax of LFP

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

All occurrences of $R$ and variables in $x$ in $\text{lfp}_{R,x} \phi$ 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 \( t \) is a tuple of terms of length \( k \), then \( P(t) \) is a formula of LFP.
• If \( \phi \) and \( \psi \) are formulas of LFP, then so are \( \phi \land \psi \), and \( \neg \phi \).
• If \( \phi \) is a formula of LFP and \( x \) is a variable then, \( \exists x \phi \) is a formula of LFP.
Let $A = (A, \mathcal{I})$ be a structure with universe $A$, and an interpretation $\mathcal{I}$ of a fixed vocabulary $\sigma$.

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

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

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

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

$\iota(f(t_1, \ldots, t_n)) = \mathcal{I}(f)(\iota(t_1), \ldots, \iota(t_n))$
Semantics of LFP

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

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

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

Anuj Dawar Logic and Complexity
Semantics of LFP

• If $\phi$ is of the form $t_1 = t_2$, then $A \models \phi[i]$ if, $i(t_1) = i(t_2)$.
• If $\phi$ is of the form $R(t_1, \ldots, t_k)$, then $A \models \phi[i]$ if,

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

• If $\phi$ is of the form $\psi_1 \land \psi_2$, then $A \models \phi[i]$ if, $A \models \psi_1[i]$ and $A \models \psi_2[i]$.
• If $\phi$ is of the form $\neg \psi$ then, $A \models \phi[i]$ if, $A \not\models \psi[i]$.
• If $\phi$ is of the form $\exists x \psi$, then $A \models \phi[i]$ if there is an $a \in A$ such that $A \models \psi[i\langle x/a \rangle]$. 
Transitive Closure

The formula (with free variables $u$ and $v$)

$$
\theta \equiv \text{lfp}_{T,xy}[(x = y \lor \exists z(E(x, z) \land T(z, y)))](u, v)
$$

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 $\phi$ is a formula in which the relation symbol $R$ occurs \textit{positively}, then the \textit{greatest fixed point} of the monotone operator $F_\phi$ defined by $\phi$ can be defined by the formula:

$$\neg[lfp_{R,x} \neg \phi(R/\neg R)](x)$$

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

\textit{Exercise:} Verify!.
Simultaneous Inductions

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

The pair \( (\phi_1, \phi_2) \) can be seen as defining a map:

\[
F : \text{Pow}(A^k) \times \text{Pow}(A^l) \rightarrow \text{Pow}(A^k) \times \text{Pow}(A^l)
\]

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 \( A \).
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 $\phi_S$ and $\phi_T$ of LFP which, on any structure $A$ containing at least two elements, define the two relations that are defined on $A$ by $\phi_1$ and $\phi_2$ by simultaneous induction.
Proof

Assume $k \leq l$.

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

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

For new variables $x_1$ and $x_2$ and a new $l + 2$-ary symbol $R$, define $\phi'_1$ and $\phi'_2$ by replacing all occurrences of $S(t_1, \ldots, t_k)$ by:

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

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

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

Define $\phi$ as

$$(x_1 = x_2 \land \phi_1') \lor (x_1 \neq x_2 \land \phi_2').$$

Then,

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

defines $P$, so

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

and

$$\phi_T \equiv \exists x_1 \exists x_2 (x_1 \neq x_2 \land \text{lfp}_{R, x_1 x_2 y} \phi)(x_1, x_2, y).$$
Complexity of LFP

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

To be precise, we can show that for each formula $\phi$ there is a $t$ such that

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

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 $\sigma$).

- 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 $\phi \equiv [\text{lfp}_{R,x}\psi](t)$ ($R$ is $l$-ary)
To decide $\mathbb{A} \models \phi[a]$:

$$
R := \emptyset \\
\text{for } i := 1 \text{ to } n' \text{ do} \\
\quad R := F_{\psi}(R) \\
\text{end} \\
\text{if } a \in R \text{ then accept else reject}
$$
Complexity of LFP

To compute $F_\psi(R)$

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

If deciding $(A, R) \models \psi$ 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 $O(n^l)$. 
For any $\phi$ of LFP, the language $\{[A]_\prec | A \models \phi\}$ is in P.

Suppose $\rho$ is a signature that contains a binary relation symbol $\prec$, possibly along with other symbols.

Let $O_\rho$ denote those structures $A$ in which $\prec$ is a linear order of the universe.

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

$$\{A \in O_\rho | [A]_\prec A \in L\}$$

(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

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

if, and only if, $M$ accepts $[\Delta]_<$ in time $n^k$, for some order $<$. If we *fix* the order $<$ as part of the structure $\Delta$, we do not need the outermost quantifier.

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

\[
\text{Tape}_a(x, y) \iff (x = 1 \land \text{Init}_a(y)) \lor \\
\exists t \exists h \bigvee_q (x = t + 1 \land \text{State}_q(t, h) \land \\
[h = y \land \bigvee \{b, d, q' \mid \Delta(q, b, q', a, d)\} \text{Tape}_b(t, y) \lor \\
h \neq y \land \text{Tape}_a(t, y)])
\]

where \(\text{Init}_a(y)\) is the formula that defines the positions in which the symbol \(a\) appears in the input.
Capturing P

\[ \text{State}_q(x, y) \Leftrightarrow (x = 1 \land y = 1 \land q = q_0) \lor \exists t \exists h \bigvee \{a, b, q' \mid \Delta(q', a, q, b, R) \} \]

\[ \bigvee \{a, b, q' \mid \Delta(q', a, q, b, L) \} \]

\[ (x = t + 1 \land \text{State}_{q'}(t, h) \land \text{Tape}_a(t, h) \land y = h + 1) \]

\[ (x = t + 1 \land \text{State'}_{q}(t, h) \land \text{Tape}_a(t, h) \land h = y + 1) \].
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.
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 $\phi$ there is a first order formula $\psi$, such that for all structures $A$ in $\mathcal{E}$, $A \models (\phi \leftrightarrow \psi)$. 
Unordered Structures

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

\[ \text{lfp}_{R, x} \psi \] defines the relation

\[ F^\infty_{\psi, b} = \bigcup_{i \in \mathbb{N}} F^i_{\psi, b} \]

for a fixed interpretation of the variables $y$ by the tuple of parameters $b$.

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

\[ F^i_{\psi, b} = \{ a \mid \mathbb{A} \models \psi^i[a, b] \}. \]
These formulas are obtained by \textit{induction}.

\[ \psi^1 \text{ is obtained from } \psi \text{ by replacing all occurrences of subformulas of the form } R(t) \text{ by } t \neq t. \]

\[ \psi^{i+1} \text{ is obtained by replacing in } \psi, \text{ all subformulas of the form } R(t) \text{ by } \psi^i(t, y) \]
Let \( b \) be an \( l \)-tuple, and \( a \) and \( c \) two \( k \)-tuples in a structure \( \mathbb{A} \) such that there is an automorphism \( \iota \) of \( \mathbb{A} \) (i.e. an isomorphism from \( \mathbb{A} \) to itself) such that

1. \( \iota(b) = b \)
2. \( \iota(a) = c \)

Then,

\[
a \in F_{\psi,b}^i \quad \text{if, and only if,} \quad c \in F_{\psi,b}^i.
\]
Bounding the Induction

This defines an *equivalence relation* $a \sim_b c$.

If there are $p$ distinct equivalence classes, then

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

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