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.
Inductive Definitions

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.
Transitive Closure

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 $E$. 
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 $F(S) \subseteq F(T)$. 
Least and Greatest Fixed Points

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 $F$. 

Anuj Dawar  
Logic and Complexity
Fixed Points

For any set \( S \in \text{Pre} \),

\[
\begin{align*}
L & \subseteq S \\
F(L) & \subseteq F(S) & \text{by definition of } F. \\
F(L) & \subseteq S & \text{by definition of } \text{Pre}. \\
F(L) & \subseteq L & \text{by definition of } L. \\
F(F(L)) & \subseteq F(L) & \text{by monotonicity of } F. \\
F(L) & \in \text{Pre} & \text{by definition of } \text{Pre}. \\
L & \subseteq F(L) & \text{by definition of } L.
\end{align*}
\]
Least and Greatest Fixed Points

$L$ is a 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:

$$G = \bigcup \{S \subseteq A \mid S \subseteq F(S)\}.$$
Let $A$ be a \textit{finite} set and $F$ be a \textit{monotone} operator on $A$.

Define for $i \in \mathbb{N}$:

\[
F^0 = \emptyset \\
F^{i+1} = F(F^i).
\]

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

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

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 $F$. 

Anuj Dawar  
Logic and Complexity
Defined Operators

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

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

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

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

$$P \mapsto F_P.$$ 

Or, $F_{\phi,b}$ if we fix parameters $b$. 

Anuj Dawar

Logic and Complexity
Positive Formulas

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 $F_{\phi,b} : \text{Pow}(A^k) \rightarrow \text{Pow}(A^k)$ 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.
Semantics 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 \( \mathcal{I} \) an interpretation in \( A \) of all the free variables (first or second order) of \( \phi \).

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

\( \mathcal{I} \) is extended to terms \( t \) in the usual way.

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

\( \mathcal{I}(f(t_1, \ldots, t_n)) = \mathcal{I}(f)(\mathcal{I}(t_1), \ldots, \mathcal{I}(t_n)) \)
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 $\text{lfp}_{R,x} \phi$, then $\iota(P)$ is the relation that is the least fixed point of the monotone operator $F$ on $A^k$ defined by:

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

where $\iota(X/R, x/a)$ 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 $\mathcal{A} \models \phi[\iota]$ if, $\iota(t_1) = \iota(t_2)$.

• If $\phi$ is of the form $R(t_1, \ldots, t_k)$, then $\mathcal{A} \models \phi[\iota]$ if,

$$\langle \iota(t_1), \ldots, \iota(t_k) \rangle \in \iota(R).$$

• If $\phi$ is of the form $\psi_1 \land \psi_2$, then $\mathcal{A} \models \phi[\iota]$ if, $\mathcal{A} \models \psi_1[\iota]$ and $\mathcal{A} \models \psi_2[\iota]$.

• If $\phi$ is of the form $\neg \psi$ then, $\mathcal{A} \models \phi[\iota]$ if, $\mathcal{A} \not\models \psi[\iota]$.

• If $\phi$ is of the form $\exists x \psi$, then $\mathcal{A} \models \phi[\iota]$ if there is an $a \in A$ such that $\mathcal{A} \models \psi[\iota(x/a)]$. 
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:

$$\neg[\text{Ifp}_{R,x} \neg \phi(R/\neg R)](x)$$

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

\textbf{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 $\mathcal{A}$ containing at least two elements, define the two relations that are defined on $\mathcal{A}$ 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:

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

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

$$\exists x_1 \exists x_2 x_1 \neq x_2 \land R(x_1, x_2, t_1, \ldots, t_i).$$
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

$$\phi_T \equiv \exists x_1 \exists x_2 (x_1 \neq x_2 \land \text{lfp}_{R, x_1 x_2 y} \phi)(x_1, x_2, y).$$
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 $O(n^{t+1})$. 

Anuj Dawar

Logic and Complexity
Complexity of LFP

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

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

$R := \emptyset$

for $i := 1$ to $n'$ do

$R := F_{\psi}(R)$

end

if $a \in R$ then accept else reject
Complexity of LFP

To compute $F_\psi(R)$

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

If deciding $(\Delta, R) \models \psi$ 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 \( \{ [A] < | A \models \phi \} \) is in \( P \).

Suppose \( \rho \) is a signature that contains a binary relation symbol \( < \), possibly along with other symbols.

Let \( O_\rho \) denote those structures \( A \) in which \( < \) 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

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

(Immerman; Vardi 1982)
Recall the proof of \textit{Fagin’s Theorem}, that ESO captures NP. Given a machine $M$ and an integer $k$, there is a \textit{first-order} formula $\phi_{M,k}$ such that

\[
\mathcal{A} \models \exists < \exists T_{\sigma_1} \cdots T_{\sigma_s} \exists S_{q_1} \cdots S_{q_m} \exists H \phi_{M,k}
\]

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

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

Capturing P

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

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

\[
\text{State}_q(x, y) \iff (x = 1 \land y = 1 \land q = q_0) \lor \\
\exists t \exists h \bigvee \{a, b, q' | \Delta(q', a, q, b, R)\} \\
\bigvee \{a, b, q' | \Delta(q', a, q, b, L)\} \\
(x = t + 1 \land \text{State}_{q'}(t, h) \land \text{Tape}_a(t, h) \land y = h + 1) \\
(x = t + 1 \land \text{State}_{q'}(t, h) \land \text{Tape}_a(t, h) \land h = y + 1).
\]
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

$$F_{\psi, b}^\infty = \bigcup_{i \in \mathbb{N}} F_{\psi, b}^i$$

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 $A$,

$$F_{\psi, b}^i = \{ a \mid A \models \psi^i[a, b] \}.$$
Defining the Stages

These formulas are obtained by *induction*.

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

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

- $\iota(b) = b$
- $\iota(a) = c$

Then,

$$a \in F_{\psi,b}^i \text{ if, and only if, } c \in F_{\psi,b}^i.$$
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.
Capturing \( P \)

The \textit{expressive power} of \textsc{LFP} is strictly weaker than \( P \).

On the other hand, \textsc{LFP} can express all queries in \( P \) on \textit{ordered structures}. Thus, every query in \( P \) can be defined by a sentence of the form

\[
\exists < \ (\text{lo}(<) \land \phi)
\]

where \( \text{lo}(<) \) is the first-order formula that says that \( < \) is a linear order and \( \phi \) is a sentence of \textsc{LFP}. 
Capturing P

With a sentence of the form $\exists < (\text{lo}(<) \land \phi)$, we can also define $\text{NP}$-complete problems.

$\exists < (\text{lo}(<) \land \forall xy[(y = x+1 \rightarrow E(x, y)) \land (x = \text{max} \land y = \text{min} \rightarrow E(x, y))]$.

defines the graphs that contain a Hamiltonian cycle.
Finite Variable Logic

We write $L^k$ for the first order formulas using only the variables $x_1, \ldots, x_k$.

$\mathbb{A} \equiv^k \mathbb{B}$

denotes that $\mathbb{A}$ and $\mathbb{B}$ agree on all sentences of $L^k$.

$(\mathbb{A}, a) \equiv^k (\mathbb{B}, b)$

denotes that there is no formula $\phi$ of $L^k$ such that $\mathbb{A} \models \phi[a]$ and $\mathbb{B} \not\models \phi[b]$.
Finite Variable Logic

For any $k$, 

$$A \equiv^k B \Rightarrow A \equiv_k B$$

However, for any $q$, there are $A$ and $B$ such that 

$$A \equiv_q B \text{ and } A \not\equiv^2 B.$$ 

Take $A$ and $B$ to be linear orders longer than $2^q$. 
For every formula $\phi$ of LFP, there is a $k$ such that the query defined by $\phi$ is closed under $\equiv^k$.

Consider a formula $\psi(R, x)$ defining an operator. Let the variables occurring in $\psi$ be $x_1, \ldots, x_k$, with $x = (x_1, \ldots, x_l)$, and $y_1, \ldots, y_l$ be new.
Define, by induction, the formulas $\psi^m$.

$$\psi^0 = \exists x \ x \neq x$$

$\psi^{m+1}$ is obtained from $\psi(R, x)$ by replacing all sub-formulas $R(t_1, \ldots, t_l)$ with

$$\exists y_1 \ldots \exists y_l \left( \bigwedge_{1 \leq i \leq l} y_i = t_i \right) \land \phi^m(y)$$

Note that each $\psi^m$ has at most $k + l$ variables.
LFP and $L^k$

If $(A, a) \equiv^{k+l} (B, b)$, then for all $m$:

$$A \models \psi^m[a] \text{ if, and only if, } B \models \psi^m[b].$$

So, $(A, a)$ and $(B, b)$ are not distinguished by $\text{Lfp}_{R, x} \psi$. 

Anuj Dawar
Logic and Complexity
Pebble Games

The $k$-pebble game is played on two structures $A$ and $B$, by two players—*Spoiler* and *Duplicator*—using $k$ pairs of pebbles $\{(a_1, b_1), \ldots, (a_k, b_k)\}$.

*Spoiler* moves by picking a pebble and placing it on an element ($a_i$ on an element of $A$ or $b_i$ on an element of $B$).

*Duplicator* responds by picking the matching pebble and placing it on an element of the other structure.

*Spoiler* wins at any stage if the partial map from $A$ to $B$ defined by the pebble pairs is not a partial isomorphism.

If *Duplicator* has a winning strategy for $q$ moves, then $A$ and $B$ agree on all sentences of $L^k$ of quantifier rank at most $q$. *(Barwise)*
Using Pebble Games

To show that a class of structures $S$ is not definable in first-order logic:

$$\forall k \forall q \exists A, B (A \in S \land B \not\in S \land A \equiv_q^k B)$$

Since $A \equiv_q^k B \Rightarrow A \equiv_q B$, we can ignore the parameter $k$

To show that $S$ is not closed under any $\equiv^k$ (and hence not definable in LFP):

$$\forall k \exists A, B \forall q (A \in S \land B \not\in S \land A \equiv_q^k B)$$

If $A \equiv_q^k B$ holds for all $q$, then **Duplicator** actually wins an *infinite* game. That is, it has a strategy to play forever.
Evenness

To show that *Evenness* is not definable in PFP, it suffices to show that:

*for every* \( k \), there are structures \( A_k \) and \( B_k \) such that \( A_k \) has an even number of elements, \( B_k \) has an odd number of elements and

\[
A \equiv^k B.
\]

It is easily seen that *Duplicator* has a strategy to play forever when one structure is a set containing \( k \) elements (and no other relations) and the other structure has \( k + 1 \) elements.
Hamiltonicity

Take $K_{k,k}$—the complete bipartite graph on two sets of $k$ vertices.
and $K_{k,k+1}$—the complete bipartite graph on two sets, one of $k$ vertices,
the other of $k + 1$.

These two graphs are $\equiv^k$ equivalent, yet one has a Hamiltonian cycle,
and the other does not.