Finite Model Theory
Lecture 11: Fixpoints and Infinitary Logic

Spring 2025

Announcements

- HW2 graded, posted.
- HW1 posted too (apparently it wasn't posted)
- HW3 due on Friday
- This week: fixpoints, infinitary logics, pebble games

Fixpoints

Tarski-Knaster

Recall: 2^U is the set of subsets of U.

$F : 2^U \to 2^U$ is monotone if $A \subseteq B$ implies $F(A) \subseteq F(B)$.

Tarski-Knaster

Recall: 2^U is the set of subsets of U.

$F : 2^U \to 2^U$ is monotone if $A \subseteq B$ implies $F(A) \subseteq F(B)$.

X is a fixpoint if $F(X) = X$.

X is a least fixpoint if it is a fixpoint and $X \subseteq X'$ for any other fixpoint X'.

Tarski-Knaster

Recall: 2^U is the set of subsets of U.

$F : 2^U \to 2^U$ is monotone if $A \subseteq B$ implies $F(A) \subseteq F(B)$.

X is a fixpoint if $F(X) = X$.

X is a least fixpoint if it is a fixpoint and $X \subseteq X'$ for any other fixpoint X'.

Theorem (Tarski-Knaster)

Every monotone function F has a least fixpoint, denoted $\mathrm{lfp}(F)$.

We consider only the finite case, discussed next.

The Kleene Sequence

Suppose U is finite, and $F : 2^U \to 2^U$ is monotone.

Definition

The Kleene Sequence is $X^0, X^1, \cdots, X^n, \cdots$, where $X^0 = \varnothing$ and $X^{i+1} = F(X^i)$

The Kleene Sequence

Suppose U is finite, and $F : 2^U \to 2^U$ is monotone.

Definition

The Kleene Sequence is $X^0, X^1, \cdots, X^n, \cdots$, where $X^0 = \varnothing$ and $X^{i+1} = F(X^i)$

Fact

The Kleene sequence is increasing: $X^0 \subseteq X^1 \subseteq \cdots \subseteq X^n \subseteq \cdots$

The Kleene Sequence

Suppose U is finite, and $F : 2^U \to 2^U$ is monotone.

Definition

The Kleene Sequence is $X^0, X^1, \cdots, X^n, \cdots$, where $X^0 = \varnothing$ and $X^{i+1} = F(X^i)$

Fact

The Kleene sequence is increasing: $X^0 \subseteq X^1 \subseteq \cdots \subseteq X^n \subseteq \cdots$

Proof by induction.
$\varnothing = X^0 \subseteq X^1$

The Kleene Sequence

Suppose U is finite, and $F : 2^U \to 2^U$ is monotone.

Definition

The Kleene Sequence is $X^0, X^1, \cdots, X^n, \cdots$, where $X^0 = \varnothing$ and $X^{i+1} = F(X^i)$

Fact

The Kleene sequence is increasing: $X^0 \subseteq X^1 \subseteq \cdots \subseteq X^n \subseteq \cdots$

Proof by induction.
$\varnothing = X^0 \subseteq X^1$

If $X^{i-1} \subseteq X^i$, by F-monotonicity $X^i = F(X^{i-1}) \subseteq F(X^i) = X^{i+1}$

The Kleene Sequence

Suppose U is finite, and $F : 2^U \to 2^U$ is monotone.

Definition

The Kleene Sequence is $X^0, X^1, \cdots, X^n, \cdots$, where $X^0 = \varnothing$ and $X^{i+1} = F(X^i)$

Fact

The Kleene sequence is increasing: $X^0 \subseteq X^1 \subseteq \cdots \subseteq X^n \subseteq \cdots$

Proof by induction.
$\varnothing = X^0 \subseteq X^1$

If $X^{i-1} \subseteq X^i$, by F-monotonicity $X^i = F(X^{i-1}) \subseteq F(X^i) = X^{i+1}$

Since U is finite, $\exists n \geq 0$ such that $X^n = X^{n+1}$. Then $\boxed{X^n = \mathrm{lfp}(F)}$

Datalog

Datalog

A datalog rule has the form:

$$P(\boldsymbol{x}_0) = R_1(\boldsymbol{x}_1) \wedge \cdots \wedge R_m(\boldsymbol{x}_m)$$

Every $R_i(\boldsymbol{x}_i)$ is a relational atom.

Datalog

A datalog rule has the form:

$$P(\boldsymbol{x}_0) = R_1(\boldsymbol{x}_1) \wedge \cdots \wedge R_m(\boldsymbol{x}_m)$$

Every $R_i(\boldsymbol{x}_i)$ is a relational atom.

The conjunction $\bigwedge_i R_i(\boldsymbol{x}_i)$ is called the body

The atom $P(\boldsymbol{x}_0)$ is called the head.

Datalog

A datalog rule has the form:

$$P(\boldsymbol{x}_0) = R_1(\boldsymbol{x}_1) \wedge \cdots \wedge R_m(\boldsymbol{x}_m)$$

Every $R_i(\boldsymbol{x}_i)$ is a relational atom.

The conjunction $\bigwedge_i R_i(\boldsymbol{x}_i)$ is called the body

The atom $P(\boldsymbol{x}_0)$ is called the head.

Safety condition: every variable in the head must occur in the body.

A datalog program is a set of datalog rules.

Classic Datalog Example

Transitive closure:

$$\boxed{\begin{array}{l} T(x,y) = E(x,y). \\ T(x,y) = E(x,z) \wedge T(z,y). \end{array}}$$

Classic Datalog Example

Transitive closure:

$$\boxed{\begin{array}{l} T(x,y) = E(x,y). \\ T(x,y) = E(x,z) \wedge T(z,y). \end{array}}$$

An equivalent, less commonly used notation:

$$T(x,y) = E(x,y) \vee \exists z(E(x,z) \wedge T(z,y))$$

Fixpoint Semantics of Datalog

Extensional Database Predicates, EDB: the input relations.

Intentional Database Predicates, IDB: the computed relations.

Fixpoint Semantics of Datalog

Extensional Database Predicates, EDB: the input relations.

Intentional Database Predicates, IDB: the computed relations.

Immediate Consequence Operator (ICO): new IDBs from current IDBs:

$$\boldsymbol{P}' = F(\boldsymbol{P})$$

Fixpoint Semantics of Datalog

Extensional Database Predicates, EDB: the input relations.

Intentional Database Predicates, IDB: the computed relations.

Immediate Consequence Operator (ICO): new IDBs from current IDBs:

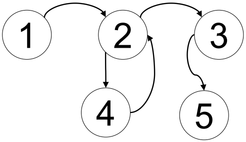
$$\boldsymbol{P}' = F(\boldsymbol{P})$$

The semantics of a datalog program is $\text{lfp}(F)$

Use Kleene sequence: $\boldsymbol{P}^0 (:= \emptyset) \subset \boldsymbol{P}^1 \subset \boldsymbol{P}^2 \subset \cdots \subset \boldsymbol{P}^n = \boldsymbol{P}^{n+1} = \text{lfp}(F)$

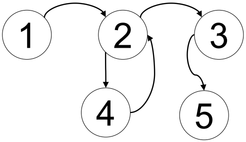
Example: Reachability

$$\boxed{\begin{array}{l} P(4). \\ P(y) = P(x) \wedge E(x,y). \end{array}}$$



Example: Reachability

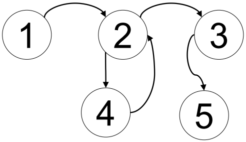
$$\boxed{\begin{array}{l} P(4). \\ P(y) = P(x) \wedge E(x,y). \end{array}}$$



$$P(y) = (y = 4) \vee \exists x (P(x) \wedge E(x,y))$$

Example: Reachability

$$\boxed{\begin{array}{l} P(4). \\ P(y) = P(x) \wedge E(x,y). \end{array}}$$

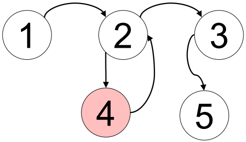


$$P(y) = (y = 4) \vee \exists x (P(x) \wedge E(x,y))$$

$$P^0 = \varnothing$$

Example: Reachability

$$\boxed{\begin{array}{l} P(4). \\ P(y) = P(x) \wedge E(x,y). \end{array}}$$



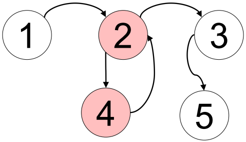
$$P(y) = (y = 4) \vee \exists x(P(x) \wedge E(x,y))$$

$$P^0 = \emptyset$$

$$P^1 = \{4\}$$

Example: Reachability

$$\boxed{\begin{array}{l} P(4). \\ P(y) = P(x) \wedge E(x,y). \end{array}}$$

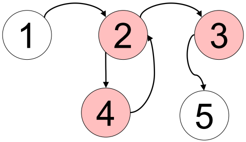


$$P(y) = (y = 4) \vee \exists x(P(x) \wedge E(x,y))$$

$$P^0 = \varnothing$$
$$P^1 = \{4\}$$
$$P^2 = \{4, 2\}$$

Example: Reachability

$$\boxed{\begin{array}{l} P(4). \\ P(y) = P(x) \wedge E(x,y). \end{array}}$$

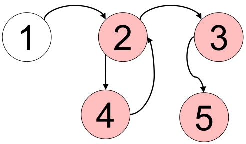


$$P(y) = (y = 4) \vee \exists x (P(x) \wedge E(x,y))$$

$$P^0 = \emptyset$$
$$P^1 = \{4\}$$
$$P^2 = \{4, 2\}$$
$$P^3 = \{4, 2, 3\}$$

Example: Reachability

$$\boxed{\begin{array}{l} P(4). \\ P(y) = P(x) \wedge E(x,y). \end{array}}$$

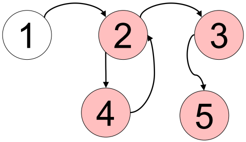


$$P(y) = (y = 4) \vee \exists x(P(x) \wedge E(x,y))$$

$$\begin{aligned} P^0 &= \emptyset \\ P^1 &= \{4\} \\ P^2 &= \{4,2\} \\ P^3 &= \{4,2,3\} \\ P^4 &= \{4,2,3,5\} \end{aligned}$$

Example: Reachability

$$\boxed{\begin{array}{l} P(4). \\ P(y) = P(x) \wedge E(x,y). \end{array}}$$



$$P(y) = (y = 4) \vee \exists x (P(x) \wedge E(x,y))$$

$$\begin{aligned} P^0 &= \varnothing \\ P^1 &= \{4\} \\ P^2 &= \{4,2\} \\ P^3 &= \{4,2,3\} \\ P^4 &= \{4,2,3,5\} \\ P^5 &= \{4,2,3,5\} = \mathrm{lfp} \end{aligned}$$

Example: Path of Even Length

Is there a path of even length from a to b?

Example: Path of Even Length

Is there a path of even length from a to b?

$$\boxed{\begin{array}{l} P(x,y) = E(x,z) \wedge E(z,y) \\ P(x,y) = P(x,z) \wedge E(z,u) \wedge E(u,y) \\ \text{Answer}() = P(a,b) \end{array}}$$

Example: Paths of Equal Length

Is there a path $a \to b$ of the same length as a path from $c \to d$?

Example: Paths of Equal Length

Is there a path $a \to b$ of the same length as a path from $c \to d$?

Product graph $G \times G$:

- Nodes: $V \times V$
- Edges: $\{((x_1, x_2), (y_1, y_2)) \mid (x_1, y_1), (x_2, y_2) \in E\}$

Example: Paths of Equal Length

Is there a path $a \to b$ of the same length as a path from $c \to d$?

Product graph $G \times G$:

- Nodes: $V \times V$
- Edges: $\{((x_1, x_2), (y_1, y_2)) \mid (x_1, y_1), (x_2, y_2) \in E\}$

Path $(a, c) \to^* (b, d)$ is a pair of paths $a \to^* b$, $c \to^* d$ of equal length.

Example: Paths of Equal Length

Is there a path $a \to b$ of the same length as a path from $c \to d$?

Product graph $G \times G$:

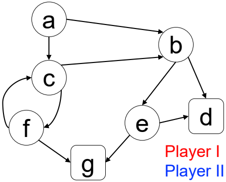
- Nodes: $V \times V$
- Edges: $\{((x_1, x_2), (y_1, y_2)) \mid (x_1, y_1), (x_2, y_2) \in E\}$

Path $(a, c) \to^* (b, d)$ is a pair of paths $a \to^* b$, $c \to^* d$ of equal length.

$$
\boxed{\begin{array}{l}
P(x1, x2, y1, y2) = E(x1, y1) \wedge E(x2, y2) \\
P(x1, x2, y1, y2) = P(x1, x2, z1, z2) \wedge E(z1, y1) \wedge E(z2, y2)) \\
\text{Answer}() = P(a, c, b, d)
\end{array}}
$$

Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

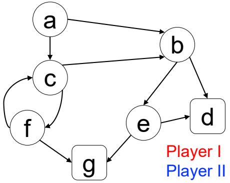


Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

A pebble is placed on a starting node:

- Player I moves to a child,
- Player II moves to a child, etc
- Player who cannot move loses.

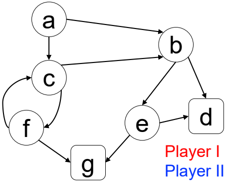


Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

A pebble is placed on a starting node:

- Player I moves to a child,
- Player II moves to a child, etc
- Player who cannot move loses.



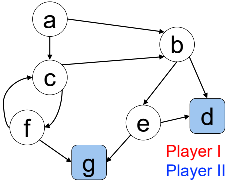
Compute all nodes x where Player I has a winning strategy

Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

A pebble is placed on a starting node:

- Player I moves to a child,
- Player II moves to a child, etc
- Player who cannot move loses.



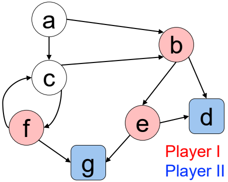
Compute all nodes x where Player I has a winning strategy

Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

A pebble is placed on a starting node:

- Player I moves to a child,
- Player II moves to a child, etc
- Player who cannot move loses.



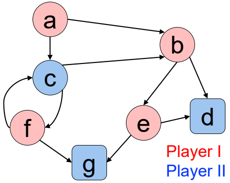
Compute all nodes x where Player I has a winning strategy

Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

A pebble is placed on a starting node:

- Player I moves to a child,
- Player II moves to a child, etc
- Player who cannot move loses.



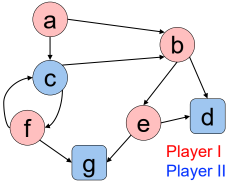
Compute all nodes x where Player I has a winning strategy

Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

A pebble is placed on a starting node:

- Player I moves to a child,
- Player II moves to a child, etc
- Player who cannot move loses.



Compute all nodes x where Player I has a winning strategy

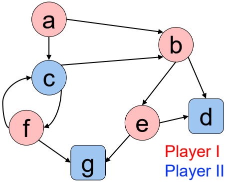
$$P2(x) = L(x)$$

Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

A pebble is placed on a starting node:

- Player I moves to a child,
- Player II moves to a child, etc
- Player who cannot move loses.



Compute all nodes x where Player I has a winning strategy

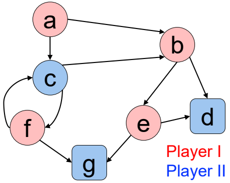
$$P2(x) = L(x)$$
$$P1(x) = T(x, y, z) \wedge P2(y)$$
$$P1(x) = T(x, y, z) \wedge P2(z)$$

Win-Move Game

$T(x, y, z)$: x has children y, z; $L(x)$: x is a leaf.

A pebble is placed on a starting node:

- Player I moves to a child,
- Player II moves to a child, etc
- Player who cannot move loses.



Compute all nodes x where Player I has a winning strategy

$$
\boxed{\begin{aligned}
P2(x) &= L(x)\\
P1(x) &= T(x, y, z) \wedge P2(y)\\
P1(x) &= T(x, y, z) \wedge P2(z)\\
P2(x) &= T(x, y, z) \wedge P1(y) \wedge P1(z)
\end{aligned}}
$$

Semi-positive Datalog

The Immediate Consequence Operator must be monotone in IDBs

Semi-positive Datalog allows EDBs to be negated.

Semi-positive Datalog

The Immediate Consequence Operator must be monotone in IDBs

Semi-positive Datalog allows EDBs to be negated.

Example: transitive closure of the complement graph:

$$\boxed{\begin{array}{l} T(x,y) = V(x) \wedge V(y) \wedge \neg E(x,y). \\ T(x,y) = T(x,z) \wedge \neg E(z,y). \end{array}}$$

Stratified Datalog

Stratified Datalog is a program partitioned into strata, such that the rules in stratum i may use:

- Positive IDBs defined in stratum i,
- Positive or negated IDBs defined in strata $< i$.

Stratified Datalog

Stratified Datalog is a program partitioned into strata, such that the rules in stratum i may use:

- Positive IDBs defined in stratum i,
- Positive or negated IDBs defined in strata $< i$.

Example: the complement of the transitive closure:

$$\boxed{\begin{array}{l} T(x,y) = E(x,y). \\ T(x,y) = T(x,z) \wedge E(z,y). \\ \text{Answer}(x,y) = V(x) \wedge V(y) \wedge \neg T(x,y). \end{array}}$$

Discussion

- Datalog can express some cool queries. Try at home:
  - Same generation: if $G = (V, E)$ is a tree, find pairs of nodes x, y in the same generation (same distance to the root)
  - Check if G is a totally balanced tree.
- But it cannot express some trivial queries:
  - Is $|E|$ even?
  - Is $|A| \leq |B|$? (Homework)
- To prove inexpressibility results for datalog we will show that it is a subset of a much more powerful logic, $L^{\omega}_{\infty\omega}$, then describe a game for $L^{\omega}_{\infty\omega}$.
- First, we discuss Fixpoint Logic

Fixpoint Logic

Definition

Fixpoint Logic, LFP extends FO with a least fixpoint predicate:

$$\mathrm{lfp}_{P,\boldsymbol{x}}\varphi$$

where P is a fresh relational symbol that occurs positively, and φ is a formula that uses P has free variables $\boldsymbol{x}$.

Definition

Fixpoint Logic, LFP extends FO with a least fixpoint predicate:

$$\mathrm{lfp}_{P,\boldsymbol{x}}\varphi$$

where P is a fresh relational symbol that occurs positively, and φ is a formula that uses P has free variables $\boldsymbol{x}$.

Example: transitive closure $T(u, v)$ becomes:

$$\boxed{\mathrm{lfp}_{T,x,y}(E(x, y) \vee \exists z(E(x, z) \wedge T(z, y)))(u, v)}$$

Definition

Fixpoint Logic, LFP extends FO with a least fixpoint predicate:

$$\mathrm{lfp}_{P,\boldsymbol{x}}\varphi$$

where P is a fresh relational symbol that occurs positively, and φ is a formula that uses P has free variables $\boldsymbol{x}$.

Example: transitive closure $T(u, v)$ becomes:

$$\boxed{\mathrm{lfp}_{T,x,y}(E(x,y) \vee \exists z(E(x,z) \wedge T(z,y)))(u,v)}$$

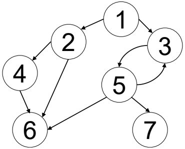
We will continue to use the more friendly notation, example:

$$\boxed{T(u,v)\ \texttt{where}\ T(x,y) = E(x,y) \vee \exists z(E(x,z) \wedge T(z,y))}$$

Example

What doses this formula compute?

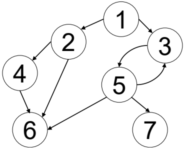
$$\boxed{S(x) = (\forall y(E(x,y) \rightarrow S(y)))}$$



Example

What doses this formula compute?

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$



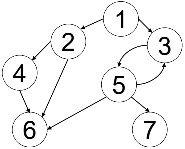
The Kleene sequence:

$$S^0 = \emptyset$$
$$S^1 =$$
$$S^2 =$$
$$S^3 =$$
$$S^4 =$$

Example

What doses this formula compute?

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$



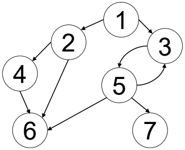
The Kleene sequence:

$$S^0 = \emptyset$$
$$S^1 = \{6, 7\}$$
$$S^2 =$$
$$S^3 =$$
$$S^4 =$$

Example

What doses this formula compute?

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$



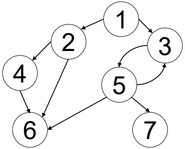
The Kleene sequence:

$$S^0 = \emptyset$$
$$S^1 = \{6,7\}$$
$$S^2 = \{6,7,4\}$$
$$S^3 =$$
$$S^4 =$$

Example

What doses this formula compute?

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$



The Kleene sequence:

$$S^0 = \emptyset$$

$$S^1 = \{6,7\}$$

$$S^2 = \{6,7,4\}$$

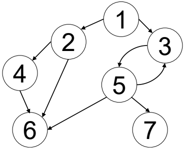
$$S^3 = \{6,7,4,2\}$$

$$S^4 =$$

Example

What doses this formula compute?

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$



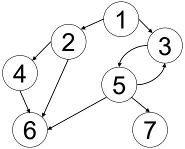
The Kleene sequence:

$$
\begin{aligned}
S^0 &= \emptyset \\
S^1 &= \{6,7\} \\
S^2 &= \{6,7,4\} \\
S^3 &= \{6,7,4,2\} \\
S^4 &= \{6,7,4,2\} = S^3 = \text{lfp}
\end{aligned}
$$

Example

What doses this formula compute?

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$



The Kleene sequence:

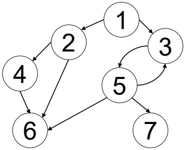
$$S^0 = \emptyset$$
$$S^1 = \{6,7\}$$
$$S^2 = \{6,7,4\}$$
$$S^3 = \{6,7,4,2\}$$
$$S^4 = \{6,7,4,2\} = S^3 = \text{lfp}$$

Nodes $1,3,5$ never show up in S^n. WHY?

Example

What doses this formula compute?

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$



The Kleene sequence:

$$S^0 = \emptyset$$
$$S^1 = \{6,7\}$$
$$S^2 = \{6,7,4\}$$
$$S^3 = \{6,7,4,2\}$$
$$S^4 = \{6,7,4,2\} = S^3 = \text{lfp}$$

Nodes $1,3,5$ never show up in S^n. WHY?

G is acyclic iff $\forall x S(x)$

Example: The Win-Move Game

$G = (V, E)$ no more restriction to two children

$$\boxed{P1(x) = \exists y(E(x,y) \wedge \forall z(E(y,z) \to P1(z)))}$$

Example: The Win-Move Game

$G = (V, E)$ no more restriction to two children

$$\boxed{P1(x) = \exists y(E(x,y) \wedge \forall z(E(y,z) \to P1(z)))}$$

Theorem (Kolaitis)

The win-move game is not expressible in stratified datalog.

Contrast with our previous solution on graphs with degree 2.

Unfolding the Recursion

$$\boxed{S(x) = (\forall y(E(x, y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x, y) \to S(y))\}$

Unfolding the Recursion

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

Unfolding the Recursion

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

$$S^0 = \varnothing \qquad\qquad \varphi^0(x) = \texttt{FALSE}$$

Unfolding the Recursion

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

$$\begin{aligned} S^0 &= \emptyset & \qquad \varphi^0(x) &= \mathtt{FALSE} \\ S^1 &= F(S^0) & \qquad \varphi^1(x) &= \forall y(E(x,y) \to \varphi^0(y)) \end{aligned}$$

Unfolding the Recursion

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

$$\begin{array}{ll} S^0 = \emptyset & \varphi^0(x) = \texttt{FALSE} \\ S^1 = F(S^0) & \varphi^1(x) = \forall y(E(x,y) \to \varphi^0(y)) \\ S^2 = F(S^1) & \varphi^2(x) = \forall y(E(x,y) \to \varphi^1(y)) \end{array}$$

Unfolding the Recursion

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

$$\begin{array}{ll}
S^0 = \varnothing & \varphi^0(x) = \mathtt{FALSE} \\
S^1 = F(S^0) & \varphi^1(x) = \forall y(E(x,y) \to \varphi^0(y)) \\
S^2 = F(S^1) & \varphi^2(x) = \forall y(E(x,y) \to \varphi^1(y)) \\
 & \dots \\
S^{n+1} = F(S^n) & \varphi^{n+1}(x) = \forall y(E(x,y) \to \varphi^n(y)) \\
 & \dots
\end{array}$$

Unfolding the Recursion

$$\boxed{S(x) = (\forall y(E(x,y) \to S(y)))}$$

The ICO: $F(S) = \{x \mid \forall y(E(x,y) \to S(y))\}$

Elements of the Kleene sequence are computed by unfolded formulas:

$$\begin{array}{ll}
S^0 = \varnothing & \varphi^0(x) = \mathtt{FALSE} \\
S^1 = F(S^0) & \varphi^1(x) = \forall y(E(x,y) \to \varphi^0(y)) \\
S^2 = F(S^1) & \varphi^2(x) = \forall y(E(x,y) \to \varphi^1(y)) \\
 & \ldots \\
S^{n+1} = F(S^n) & \varphi^{n+1}(x) = \forall y(E(x,y) \to \varphi^n(y)) \\
 & \ldots
\end{array}$$

$$\boxed{\mathrm{lfp}(S)(x) = \varphi^0(x) \vee \varphi^1(x) \vee \cdots \vee \varphi^n(x) \vee \cdots}$$

Discussion

- Datalog, semi-positive Datalog, stratified Datalog
- Least Fixpoint Logic (LFP)

Discussion

- Datalog, semi-positive Datalog, stratified Datalog
- Least Fixpoint Logic (LFP)
- LFP strictly more powerful than Datalog

Discussion

- Datalog, semi-positive Datalog, stratified Datalog
- Least Fixpoint Logic (LFP)
- LFP strictly more powerful than Datalog
- There are other more powerful (and more ugly) forms of fixpoints that we wont discuss

Discussion

- Datalog, semi-positive Datalog, stratified Datalog
- Least Fixpoint Logic (LFP)
- LFP strictly more powerful than Datalog
- There are other more powerful (and more ugly) forms of fixpoints that we wont discuss
- Infinitary logic, $L^{\omega}_{\infty\omega}$ allows us to write formulas like:

$$\varphi^0(x) \vee \varphi^1(x) \vee \cdots \vee \varphi^n(x) \vee \cdots$$

Discussion

- Datalog, semi-positive Datalog, stratified Datalog
- Least Fixpoint Logic (LFP)
- LFP strictly more powerful than Datalog
- There are other more powerful (and more ugly) forms of fixpoints that we wont discuss
- Infinitary logic, $L^{\omega}_{\infty\omega}$ allows us to write formulas like:

$$\varphi^0(x) \vee \varphi^1(x) \vee \cdots \vee \varphi^n(x) \vee \cdots$$

- What are ∞ and ω? Will discuss next.

Ordinal Numbers

Review: Cardinal Numbers

A cardinal number is an equivalence class $|A|$ under bijection.

$\aleph_0 = |\mathbb{N}|$ is the smallest infinite cardinal number

$\mathfrak{c} = |\mathbb{R}|$ is the cardinal of the continuum.

Weird arithmetic: $\aleph_0 + \mathfrak{c} = \mathfrak{c}$, $\aleph_0 \times \mathfrak{c} = \mathfrak{c}$, $\mathfrak{c} \times \mathfrak{c} = \mathfrak{c}$, ...

Much larger cardinal numbers exists: $\aleph_0 < 2^{\aleph_0} < 2^{2^{\aleph_0}} < 2^{2^{2^{\aleph_0}}} < \cdots$

Well Ordered Sets

A partial order is a pair $(S, \leq)$,
where $\leq$ is reflexive, antisymmetric, and transitive.

Well Ordered Sets

A partial order is a pair $(S, \leq)$,
where $\leq$ is reflexive, antisymmetric, and transitive.

Definition

$(S, \leq)$ is a well order if every subset $A \subseteq S$ has a minimal element.

Well Ordered Sets

A partial order is a pair $(S, \leq)$,
where $\leq$ is reflexive, antisymmetric, and transitive.

Definition

$(S, \leq)$ is a well order if every subset $A \subseteq S$ has a minimal element.

Fact

If $\leq$ is a well order then it is also a total order.

Well Ordered Sets

A partial order is a pair $(S, \leq)$,
where $\leq$ is reflexive, antisymmetric, and transitive.

Definition

$(S, \leq)$ is a well order if every subset $A \subseteq S$ has a minimal element.

Fact

If $\leq$ is a well order then it is also a total order.

Proof: given x, y, set $A = \{x, y\}$.
$\min(A) \in A$: either $\min(x, y) = x$ and $x \leq y$, or $\min(x, y) = y$ and $y \leq x$.

Well Ordered Sets

A partial order is a pair $(S, \leq)$,
where $\leq$ is reflexive, antisymmetric, and transitive.

Definition

$(S, \leq)$ is a well order if every subset $A \subseteq S$ has a minimal element.

Fact

If $\leq$ is a well order then it is also a total order.

Proof: given x, y, set $A = \{x, y\}$.
$\min(A) \in A$: either $\min(x, y) = x$ and $x \leq y$, or $\min(x, y) = y$ and $y \leq x$.

Fact

$\leq$ is a well order iff its total and has no infinite descending chains:

$$x_0 > x_1 > x_2 > \dots$$

Ordinal Numbers

Definition

A cardinal number is an equivalence class (under bijection) of a set.

Definition

An ordinal number is an isomorphism class of a well ordered set.

Examples

Every finite number is an ordinal: $([n], \leq)$.

Examples

Every finite number is an ordinal: $([n], \leq)$.

$(\mathbb{N}, \leq)$ is a well ordered set. Its ordinal number is denoted ω:

$$\omega : 0 < 1 < 2 < 3 < \cdots$$

Examples

Every finite number is an ordinal: $([n], \leq)$.

$(\mathbb{N}, \leq)$ is a well ordered set. Its ordinal number is denoted ω:

$$\omega : 0 < 1 < 2 < 3 < \cdots$$

$$\omega + 1 : 0 < 1 < 2 < 3 < \cdots < 0^1$$

Examples

Every finite number is an ordinal: $([n], \leq)$.

$(\mathbb{N}, \leq)$ is a well ordered set. Its ordinal number is denoted ω:

$$\omega : 0 < 1 < 2 < 3 < \cdots$$

$$\omega + 1 : 0 < 1 < 2 < 3 < \cdots < 0^1$$

$$\omega + 2 : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1$$

Examples

Every finite number is an ordinal: $([n], \leq)$.

$(\mathbb{N}, \leq)$ is a well ordered set. Its ordinal number is denoted ω:

$$\omega : 0 < 1 < 2 < 3 < \cdots$$

$$\omega + 1 : 0 < 1 < 2 < 3 < \cdots < 0^1$$

$$\omega + 2 : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1$$

$$\ldots$$

$$\omega + \omega : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1 < 2^1 < \cdots$$

Examples

Every finite number is an ordinal: $([n], \leq)$.

$(\mathbb{N}, \leq)$ is a well ordered set. Its ordinal number is denoted ω:

$$\omega : 0 < 1 < 2 < 3 < \cdots$$

$$\omega + 1 : 0 < 1 < 2 < 3 < \cdots < 0^1$$

$$\omega + 2 : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1$$

$$\ldots$$

$$\omega + \omega : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1 < 2^1 < \cdots$$

$$\omega + \omega + \omega : \ldots$$

$$\ldots$$

Examples

Every finite number is an ordinal: $([n], \leq)$.

$(\mathbb{N}, \leq)$ is a well ordered set. Its ordinal number is denoted ω:

$$\omega : 0 < 1 < 2 < 3 < \cdots$$

$$\omega + 1 : 0 < 1 < 2 < 3 < \cdots < 0^1$$

$$\omega + 2 : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1$$

$$\ldots$$

$$\omega + \omega : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1 < 2^1 < \cdots$$

$$\omega + \omega + \omega : \ldots$$

$$\ldots$$

$$\omega \times \omega : 0 < 1 < 2 < \cdots < 0^1 < 1^1 < \cdots < 0^2 < 1^2 < \cdots < \cdots$$

Examples

Every finite number is an ordinal: $([n], \leq)$.

$(\mathbb{N}, \leq)$ is a well ordered set. Its ordinal number is denoted ω:

$$\omega : 0 < 1 < 2 < 3 < \cdots$$

$$\omega + 1 : 0 < 1 < 2 < 3 < \cdots < 0^1$$

$$\omega + 2 : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1$$

$$\ldots$$

$$\omega + \omega : 0 < 1 < 2 < 3 < \cdots < 0^1 < 1^1 < 2^1 < \cdots$$

$$\omega + \omega + \omega : \ldots$$

$$\ldots$$

$$\omega \times \omega : 0 < 1 < 2 < \cdots < 0^1 < 1^1 < \cdots < 0^2 < 1^2 < \cdots < \cdots$$

All these are countable ordinals, i.e. their cardinal number is $\aleph_0$

Infinitely Many Ordinals

Is $(\mathbb{R}, \leq)$ well ordered?

Infinitely Many Ordinals

Is $(\mathbb{R}, \leq)$ well ordered?

NO! The set $\{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\}$ has no minimal element.

Infinitely Many Ordinals

Is $(\mathbb{R}, \leq)$ well ordered?

NO! The set $\{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\}$ has no minimal element.

Theorem

Every set can be well ordered.

Infinitely Many Ordinals

Is $(\mathbb{R}, \leq)$ well ordered?

NO! The set $\{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\}$ has no minimal element.

Theorem

Every set can be well ordered.

Every ordinal number has a unique cardinal number.

A cardinal number can have many ordinal numbers.

$L_{\infty\omega}$

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i\in I} \varphi_i$, and $\underbrace{(\ldots \exists x_j \ldots)}_{j\in J}\varphi$, and $\neg\varphi$,for $|I| < \alpha$, $|J| < \beta$.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i\in I} \varphi_i$, and $(\underbrace{\ldots \exists x_j \ldots}_{j\in J})\varphi$, and $\neg\varphi$,for $|I| < \alpha$, $|J| < \beta$.
- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i \in I} \varphi_i$, and $(\underbrace{\ldots \exists x_j \ldots}_{j \in J})\varphi$, and $\neg\varphi$,for $|I| < \alpha$, $|J| < \beta$.
- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i \in I} \varphi_i$, and $\underbrace{(\ldots \exists x_j \ldots)}_{j \in J}\varphi$, and $\neg\varphi$,for $|I| < \alpha$, $|J| < \beta$.
- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence.
- $L^{k}_{\infty\omega}$ = the restriction to k variables.

$L^{\omega}_{\infty\omega}$

- Let α, β be ordinals. The infinitary logic $L_{\alpha\beta}$ is:
  - Atoms: $x_i = x_j$, $R(\cdots)$
  - Formulas: $\bigvee_{i \in I} \varphi_i$, and $\underbrace{(\ldots \exists x_j \ldots)}_{j \in J}\varphi$, and $\neg\varphi$,for $|I| < \alpha$, $|J| < \beta$.
- $L_{\omega\omega} = FO$; finite disjunctions, finite quantifier sequence.
- $L_{\infty\omega}$ = infinite disjunction (no bound!), finite quantifier sequence.
- $L^k_{\infty\omega}$ = the restriction to k variables.
- $L^{\omega}_{\infty\omega} = \bigcup_{k \geq 0} L^k_{\infty\omega}$.

What is $\bigcup_{k \geq 0} FO^k$?

Discussion

Next time:

- The logic LFP is a subset of $L^{\omega}_{\infty\omega}$.
- Pebble games, which generalize EF games, characterize expressibility in $L^{\omega}_{\infty\omega}$.
- As an example, will show that EVEN is not expressible in $L^{\omega}_{\infty\omega}$, hence neither in LFP or Datalog.