Predicate logic (First-order logic)

- Syntax and semantics
- Natural deduction system
- Undecidability of predicate logic
- Expressiveness of predicate logic
- Second-order logic

Reference: Chap 2, Michael Huth and Mark Ryan, Logic in Computer Science: Modeling and Reasoning about Systems, Second Edition, Cambridge University Press, 2004.

Symbols of predicate logic

1. Logical symbols
   - connective symbols: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
   - quantifier symbols: $\forall, \exists$
   - variable symbols: $x, y, z, \ldots$
   - parentheses and commas
   - the equality symbol $=$
2. Nonlogical symbols
   - constant or individual symbols: $a, b, c, \ldots$
   - predicate or relation symbols: $P, Q, R, \ldots$, each with an arity (a 0-ary predicate symbol is a proposition)
   - function symbols: $f, g, h, \ldots$, each with an arity (a constant symbol can be treated as 0-ary function symbol)

Languages of predicate logic

- By a language, we mean a set of nonlogical symbols
- An important language: the language of arithmetic
  $L^* = [0,', +, \cdot\ ; <]$

0	constant symbol
$'$	unary function symbol
$+, \cdot$	binary function symbols
$<$	binary predicate symbol

Terms

- *Terms* are certain strings built from variables and function symbols, and are intended to represent objects in the universe of discourse.
- **Definition of terms**
  - Variables and constants are atomic terms.
  - If f is an n-ary function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.
- Examples of terms: $0''$, $x + y$, $f(g(a), x), f(x, y), g(f(a, g(a)))$

Formulas

- **Definition of formulas**:
  - $P(t_1, \cdots, t_n)$ is an *atomic* formula, where P is an n-ary predicate symbol and $t_1, \cdots, t_n$ are terms.
  - $t_1 = t_2$ is an *atomic* formula
  - If A and B are formulas, so are $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ and $(A \leftrightarrow B)$
  - If A is an formula and x is a variable, then $\forall x A$ and $\exists x A$ are formulas.
- *e.g.*, $(\neg \forall x P(x) \vee \exists x \neg P(x))$, $(\forall x \neg Q(x, y) \wedge \neg \forall z Q(f(y), z))$

Free and bound variables

- Definition. An occurrence of variable x in A is bound if it is in a subformula of A of the form $\forall x B$ or $\exists x B$. Otherwise the occurrence is free.
- *e.g.*, $x < y \wedge \neg \exists z (x < z \wedge z < y)$, $F(x) \rightarrow \forall x F(x)$
- Definition. A formula A or a term t is closed if it contains no free occurrence of variables. A closed formula is called a sentence.

Substitution

- Definition. Given a variable x, a term t, and a formula ϕ, we define $\phi[t/x]$ to be the formula obtained by replacing each free occurrence of variable x in ϕ with t.
- *e.g.*, let $\phi = \forall x(P(x) \wedge Q(x)) \rightarrow (\neg P(x) \vee Q(y))$, what is $\phi[f(x,y)/x]$?
- Undesired side effects: let $\phi = S(x) \wedge \forall y(\neg P(x) \vee Q(y))$, what is $\phi[f(y,y)/x]$? y gets "caught" by $\forall y$
- Definition. Given a variable x, a term t, and a formula ϕ, we say that t is free for x in ϕ if no free x in ϕ occurs in the scope of $\forall y$ or $\exists y$ for any variable y occurring in t.

Semantics

To know whether a formula $\forall x(P(x) \vee Q(x))$ is true, we have to specify

1. A domain D over which x ranges
2. An interpretation of the predicate P
3. An interpretation of the predicate Q

This gives us the concept of interpretation

Semantics

An interpretation $\mathcal{M}$ for a language L consists of the following:

1. A nonempty set $|\mathcal{M}|$ called the *domain* or *universe of discourse* of $\mathcal{M}$.
2. A denotation assigned to each nonlogical symbol of $\mathcal{L}$:
   - For each constant symbol c, $c^{\mathcal{M}} \in |\mathcal{M}|$;
   - For each n-ary function symbol f, $f^{\mathcal{M}}$ is an n-ary function from $|\mathcal{M}|$ to $|\mathcal{M}|$.
   - For each n-ary predicate symbol, $P^{\mathcal{M}}$ is an n-ary relation on $|\mathcal{M}|$.
3. for the equality symbol $=$, $=^{\mathcal{M}}$ is the identity relation on $|\mathcal{M}|$.

denoted $\mathcal{N}^*$

- domain: the set of natural numbers $\mathbb{N}$
- the denotation of $'$ is the successor function:
- $0, <, +, \cdot$ get their usual meanings

Object assignments

Given an interpretation, to know if $P(x)$ is true, we have to know the value of x

- Definition. An object assignment l for an interpretation $\mathcal{M}$ is a mapping from variables to the domain $|\mathcal{M}|$.
- Notation. If x is a variable and $a \in |\mathcal{M}|$, then the object assignment $l[x \mapsto a]$ is the same as l except it maps x to a.

Denotation of terms

Given an interpretation, to know if $P(t)$ is true, we have to know the denotation of t

Let $\mathcal{M}$ be an interpretation for L, l an object assignment for $\mathcal{M}$, and t a term. The denotation of t in $\mathcal{M}$ under l, denoted $t^{\mathcal{M}}[l]$, is defined as follows:

a) if t is a variable x, then $t^{\mathcal{M}}[l] = l(x)$

b) if $t = f(t_1, \ldots, t_n)$, then $t^{\mathcal{M}}[l] = f^{\mathcal{M}}(t_1^{\mathcal{M}}[l], \ldots, t_n^{\mathcal{M}}[l])$

e.g., let $l(x) = 1$, then $(0''')^{\mathcal{N}^*}[l] = 3$, $(x + 0'')^{\mathcal{N}^*}[l] = 3$

Truth for formulas

For A an L-formula, the notion $\mathcal{M} \models_l A$ ($\mathcal{M}$ *satisfies* A *under* l) is defined by structural induction on formulas A as follows:

a) $\mathcal{M} \models_l P(t_1, \cdots, t_n)$ iff $\langle t_1^{\mathcal{M}}[l], \ldots, t_n^{\mathcal{M}}[l] \rangle \in P^{\mathcal{M}}$

b) $\mathcal{M} \models_l (s = t)$ iff $s^{\mathcal{M}}[l] = t^{\mathcal{M}}[l]$

c) $\mathcal{M} \models_l \neg A$ iff $\mathcal{M} \not\models_l A$, *i.e.*, not $\mathcal{M} \models_l A$.

d) $\mathcal{M} \models_l (A \vee B)$ iff $\mathcal{M} \models_l A$ or $\mathcal{M} \models_l B$.

e) $\mathcal{M} \models_l (A \wedge B)$ iff $\mathcal{M} \models_l A$ and $\mathcal{M} \models_l B$.

f) $\mathcal{M} \models_l \forall x A$ iff $\mathcal{M} \models_{l[x \mapsto a]} A$ for all $a \in |\mathcal{M}|$

g) $\mathcal{M} \models_l \exists x A$ iff $\mathcal{M} \models_{l[x \mapsto a]} A$ for some $a \in |\mathcal{M}|$

Examples

- Let L be the language $\{R, =\}$ and let $\mathcal{M}$ be the L-interpretation whose universe $|\mathcal{M}| = \mathbb{N}$ and such that $R^{\mathcal{M}}(m, n)$ holds iff $m \leq n$. Then $\mathcal{M} \models \exists x \forall y R(x, y)$ but $\mathcal{M} \not\models \exists y \forall x R(x, y)$
- $\mathcal{N}^* \models \forall x \forall y \exists z (x + z = y \vee y + z = x)$ but $\mathcal{N}^* \not\models \forall x \exists y (y + y = x)$

Lemma: If l and l' agree on the free variables of t, then $t^{\mathcal{M}}[l] = t^{\mathcal{M}}[l']$.

Proof: Structural induction on terms t.

Lemma: If l and l' agree on the free variables of A, then $\mathcal{M} \models_l A$ iff $\mathcal{M} \models_{l'} A$.

Proof: Structural induction on formulas A.

Corollary: If A is a sentence, then for any object assignments l, l', $\mathcal{M} \models_l A$ iff $\mathcal{M} \models_{l'} A$.

In view of the Corollary, if A is a sentence, then l is irrelevant, so we omit mention of l and simply write $\mathcal{M} \models_l A$.

Important definitions

a) A is *satisfiable* iff there is some $\mathcal{M}$ and l such that $\mathcal{M} \models_l A$.

b) Φ is *satisfiable* if there is some $\mathcal{M}$ and l such that $\mathcal{M} \models_l \phi$ for all $\phi \in \Phi$.

c) $\Phi \models A$ (Φ entails A) iff for all $\mathcal{M}$ and all l, if $\mathcal{M} \models_l \phi$ for all $\phi \in \Phi$ then $\mathcal{M} \models_l A$.

d) $\models A$ (A is *valid*) iff $\mathcal{M} \models_l A$ for all $\mathcal{M}$ and l.

e) $A \Longleftrightarrow B$ (A and B are *logically equivalent*) iff for all $\mathcal{M}$ and all l, $\mathcal{M} \models_l A$ iff $\mathcal{M} \models_l B$.

Examples

- $\forall x(P(x) \to Q(x)) \models \forall x P(x) \to \forall x Q(x)$
- Does $\forall x P(x) \to \forall x Q(x) \models \forall x(P(x) \to Q(x))$?
- $(\forall x A \vee \forall x B) \models \forall x(A \vee B)$
- Does $\forall x(A \vee B) \models (\forall x A \vee \forall x B)$?
- $\neg \forall x A \iff \exists x \neg A$
- $\neg \exists x A \iff \forall x \neg A$
- $(\forall x A \wedge \forall x B) \iff \forall x(A \wedge B)$
- $\exists x(A \vee B) \iff (\exists x A \vee \exists x B)$
- $\exists x(A \wedge B) \models (\exists x A \wedge \exists x B)$
- Does $(\exists x A \wedge \exists x B) \models \exists x(A \wedge B)$?

Examples

- $\forall x \forall y A \iff \forall y \forall x A$
- $\exists x \exists y A \iff \exists y \exists x A$
- $\exists y \forall x A \models \forall x \exists y A$
- Does $\forall x \exists y A \models \exists y \forall x A$?
- $\forall x A \models \exists x A$
- $\forall x \forall y (x = y \to f(x) = f(y))$ is valid.
- $\forall x \forall y (f(x) = f(y) \to x = y)$ is NOT valid.

Proof that $\forall x(P(x) \to Q(x)) \models \forall x P(x) \to \forall x Q(x)$

- Let $\mathcal{M}$ be any interpretation and let l be any object assignment.
- Suppose that $\mathcal{M} \models_l \forall x(P(x) \to Q(x))$ and $\mathcal{M} \models_l \forall x P(x)$.
- Let a be an arbitrary element of $|\mathcal{M}|$.
- Since $\mathcal{M} \models_l \forall x P(x)$, $a \in P^M$.
- Since $\mathcal{M} \models_l \forall x(P(x) \to Q(x))$, $a \in Q^M$.
- Thus for all $a \in |\mathcal{M}|$, $a \in Q^M$.
- So $\mathcal{M} \models_l \forall x Q(x)$.

Proof that $(\forall x A \vee \forall x B) \models \forall x(A \vee B)$

- Let $\mathcal{M}$ be any interpretation and let l be any object assignment.
- Suppose that $\mathcal{M} \models_l \forall x A \vee \forall x B$.
- Then $\mathcal{M} \models_l \forall x A$ or $\mathcal{M} \models_l \forall x B$.
- Say $\mathcal{M} \models_l \forall x A$.
- Then $\mathcal{M} \models_{l[x \mapsto a]} A$ for all $a \in |\mathcal{M}|$.
- Thus $\mathcal{M} \models_{l[x \mapsto a]} A \vee B$ for all $a \in |\mathcal{M}|$.
- Therefore, $\mathcal{M} \models_l \forall x(A \vee B)$.

Proof that $\exists y \forall x A \models \forall x \exists y A$

- Let $\mathcal{M}$ be any interpretation and let l be any object assignment.
- Suppose that $\mathcal{M} \models_l \exists y \forall x A$.
- Then $\mathcal{M} \models_{l[x \mapsto b]} \forall x A$ for some b in $|\mathcal{M}|$.
- Call this b b_0. Then $\mathcal{M} \models_{l[y \mapsto b_0]} \forall x A$.
- Thus $\mathcal{M} \models_{l[y \mapsto b_0][x \mapsto a]} A$ for all a in $|\mathcal{M}|$.
- So $\mathcal{M} \models_{l[x \mapsto a]} \exists y A$ for all a in $|\mathcal{M}|$.
- Therefore, $\mathcal{M} \models_l \forall x \exists y A$.

Now we would like to prove change of bound variables preserves logical equivalence, *e.g.*, $\forall x(P(x) \vee Q(x)) \iff \forall x(P(x) \vee Q(x))$

We need to make some preparations.

Lemma For each interpretation $\mathcal{M}$ and each object assignment l,

$$(t[s/x])^{\mathcal{M}}[l] = t^{\mathcal{M}}[l[x \mapsto s^{\mathcal{M}}[l]]].$$

Example: Consider $\mathcal{N}^*$. Let $l(x) = 5$ and $l(y) = 7$. Let t be the term $x + y$ and let s be the term $0''$.

Proof of the Lemma: Structural induction on t.

Question: Does the above lemma apply to formulas A? I.e. can we say $\mathcal{M} \models_l A(t/x)$ iff $\mathcal{M} \models_{l[x \mapsto a]} A$, where $a = t^{\mathcal{M}}[l]$? Something can go wrong.

Example: Suppose A is $\forall y \neg(x = y + y)$. This says "$x$ is odd". But $A(x + y/x)$ is $\forall y \neg(x + y = y + y)$, which does not say "$x + y$ is odd" as desired, but instead it is always false. The problem is that y in the term $x + y$ got "caught" by the quantifier $\forall y$.

Substitution Theorem: If t is free for x in A then for all interpretations $\mathcal{M}$ and all object assignments l, $\mathcal{M} \models_l A(t/x)$ iff $\mathcal{M} \models_{l[x \mapsto a]} A$, where $a = t^{\mathcal{M}}[l]$.

Proof: Structural induction on A. The interesting case is when A is $\forall yB$. (The case when A is $\exists yB$ is similar). Then we are to prove

$$\mathcal{M} \models_l (\forall yB)(t/x) \text{ iff } \mathcal{M} \models_{l[x \mapsto a]} \forall yB \tag{1}$$

where $a = t^{\mathcal{M}}[l]$.

Change of Bound Variable

If a term t is not free for x in A, it is because some variable y in t gets caught by a quantifier $\forall y$ or $\exists y$ in A. One way to fix this is simply rename the bound variable y in A to some new variable z.

Definition: $\forall z A(z/y)$ results from $\forall y A$ by *change of bound variable* provided z does not occur in A. Similarly for $\exists z A(z/y)$.

Lemma: If z does not occur in A, then $\forall z A(z/y)$ and $\forall y A$ are logically equivalent. Also $\exists z A(z/y)$ and $\exists y A$ are equivalent.

Proof: This follows from the Substitution Theorem.

Definition A' is a *variant* of A if A' results by a sequence of changes of bound variables to subformulas of A.

Theorem: If A' is a variant of A then A and A' are equivalent.

This follows from the preceding Lemma and the following theorem:

Replacement Theorem: If B and B' are equivalent formulas and A' results from A by replacing some occurrence of B in A by B', then A and A' are equivalent.

Proof: By structural induction on A (relative to B). The base case is when A and B coincide.

Example: B is $\neg\forall x P(x, y)$, B' is $\exists z \neg P(z, y)$, A is $\forall y(\neg\forall x P(x, y) \supset Q(y))$.

Null quantification (1)

When x does not occur as a free variable in A

- $\forall x P(x) \vee A \equiv \forall x (P(x) \vee A)$
- $\exists x P(x) \vee A \equiv \exists x (P(x) \vee A)$
- $\forall x P(x) \wedge A \equiv \forall x (P(x) \wedge A)$
- $\exists x P(x) \wedge A \equiv \exists x (P(x) \wedge A)$

Null quantification (2)

When x does not occur as a free variable in A

- $\forall x P(x) \to A \equiv \exists x (P(x) \to A)$
- $\exists x P(x) \to A \equiv \forall x (P(x) \to A)$
- $A \to \forall x P(x) \equiv \forall x (A \to P(x))$
- $A \to \exists x P(x) \equiv \exists x (A \to P(x))$

The size of models

- Let n be a positive integer. Write a sentence I_n using identity but no nonlogical symbols such that I_n is true in M iff there are $\geq n$ distinct individuals in M
- Write J_n to express "there are at most n individuals" and K_n to express "there are exactly n individuals"

Natural deduction rules for quantifiers and equality

Whenever we write $\phi[t/x]$, we assume that t is free for x in ϕ.

$$\frac{}{t = t} = i, \qquad \frac{t_1 = t_2, \phi[t_1/x]}{\phi[t_2/x]} = e$$

$$\frac{\forall x \phi}{\phi[t/x]} \forall e, \qquad \frac{\phi[t/x]}{\exists x \phi} \exists i$$

$$\dfrac{\boxed{\begin{array}{l} x_0 \\ \qquad \vdots \\ \qquad \phi[x_0/x] \end{array}}}{\forall x\, \phi}\ \forall x\, \mathrm{i}. \qquad \dfrac{\exists x\, \phi \quad \boxed{\begin{array}{l} x_0\ \phi[x_0/x] \\ \qquad \vdots \\ \qquad \chi \end{array}}}{\chi}\ \exists \mathrm{e}.$$

What does the box mean? x_0 is a new variable which does not appear anywhere outside its box.

Why the proviso for $\phi[t/x]$?

1. Consider $\forall e$
2. Let ϕ be $\exists y R(x, y)$
3. $\forall x \exists y R(x, y) \not\models \exists y R(y, y)$

Examples

1. $t_1 = t_2 \vdash t_2 = t_1$
2. $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$
3. $\forall x(P(x) \to Q(x)), \forall x P(x) \vdash \forall x Q(x)$
4. $\forall x \phi \vdash \exists x \phi$
5. $\forall x(P(x) \to Q(x)), \exists x P(x) \vdash \exists x Q(x)$
6. $\forall x(Q(x) \to R(x)), \exists x(P(x) \land Q(x)) \vdash \exists x(P(x) \land R(x))$
7. $\exists x P(x), \forall x \forall y(P(x) \to Q(y)) \vdash \forall y Q(y)$

Why cannot x_0 appear outside its box?

$$\exists x P(x), \forall x(P(x) \to Q(x)) \not\models \forall y Q(y)$$

What goes wrong in the following proof?

1		$\exists x\, P(x)$	premise
2		$\forall x\,(P(x) \to Q(x))$	premise
3	x_0		
4	x_0	$P(x_0)$	assumption
5		$P(x_0) \to Q(x_0)$	$\forall x$ e 2
6		$Q(x_0)$	$\to$e 5, 4
7		$Q(x_0)$	$\exists x$ e 1, 4–6
8		$\forall y\, Q(y)$	$\forall y$ i 3–7

The soundness and completeness theorem

Theorem. $\Gamma \vdash D$ iff $\Gamma \models D$.

We do not prove it in this course.

Proof via natural deduction

1. (a) $\neg\forall x\phi \iff \exists x\neg\phi$

   (b) $\neg\exists x\phi \iff \forall x\neg\phi$

2. (a) $\forall x\forall y\phi \iff \forall y\forall x\phi$

   (b) $\exists x\exists y\phi \iff \exists y\exists x\phi$

3. (a) $\forall x\phi \wedge \forall x\psi \iff \forall x(\phi \wedge \psi)$

   (b) $\exists x\phi \vee \exists x\psi \iff \exists x(\phi \vee \psi)$

4. Assuming that x is not free in ψ:

   (a) $\diamond x\phi \odot \psi \iff \diamond x(\phi \odot \psi)$, where $\diamond \in \{\forall, \exists\}$, and $\odot \in \{\wedge, \vee\}$

   (b) $\diamond x(\phi \to \psi) \iff \bar{\diamond} x\phi \to \psi$, where $\diamond \in \{\forall, \exists\}$, $\bar{\forall} = \exists$, $\bar{\exists} = \forall$

   (c) $\diamond x(\psi \to \phi) \iff \psi \to \diamond x\phi$, where $\diamond \in \{\forall, \exists\}$

Proof that $\neg\forall x\phi \vdash \exists x\neg\phi$

1	$\neg\forall x\, \phi$	premise
2	$\neg\exists x\, \neg\phi$	assumption
3	x_0	
4	$\neg\phi[x_0/x]$	assumption
5	$\exists x\neg\phi$	$\exists x$ i 4
6	$\perp$	$\neg$e 5, 2
7	$\phi[x_0/x]$	PBC 4–6
8	$\forall x\, \phi$	$\forall x$ i 3–7
9	$\perp$	$\neg$e 8, 1
10	$\exists x\, \neg\phi$	PBC 2–9

Undecidability of predicate logic

- Recall $\models \phi$ (ϕ is valid)
- The problem of deciding if a formula is valid is an example of a decision problem.
- A solution to a decision problem is a program that takes an instance of the problem as input and always terminates, producing a correct 'yes' or 'no' output.
- Validity in propositional logic is solvable.
- However, validity in predicate logic is unsolvable.
- We prove this by the technique of problem reduction: take a problem known to be unsolvable, and show that the solvability of our problem would entail that of this problem.

The Post correspondence problem (PCP)

- Given a finite sequence of pairs (s_1, t_1), (s_2, t_2), $\ldots$, (s_k, t_k) such that all s_i and t_i are binary strings of positive length, is there a sequence of indices $i_1, i_2, \ldots, i_n$ with $n \geq 1$ such that the concatenation of strings $s_{i_1} s_{i_2} \ldots s_{i_n}$ equals $t_{i_1} t_{i_2} \ldots t_{i_n}$?
- Note: An index can appear multiple times in the sequence.
- An instance: $(1, 101), (10, 00), (011, 11)$

The Post correspondence problem (PCP)

- Given a finite sequence of pairs (s_1, t_1), (s_2, t_2), $\ldots$, (s_k, t_k) such that all s_i and t_i are binary strings of positive length, is there a sequence of indices $i_1, i_2, \ldots, i_n$ with $n \geq 1$ such that the concatenation of strings $s_{i_1} s_{i_2} \ldots s_{i_n}$ equals $t_{i_1} t_{i_2} \ldots t_{i_n}$?
- Note: An index can appear multiple times in the sequence.
- An instance: $(1, 101), (10, 00), (011, 11)$ A solution: $1, 3, 2, 3$
- Another instance: $(001, 0), (01, 011), (01, 101), (10, 001)$ Solution?
- The Post correspondence problem is unsolvable.
- A rough explanation: the search space is infinite.

Theorem. Validity in predicate logic is undecidable:
no program exists which, given any ϕ, decides whether $\models \phi$.

Idea of proof:

- Reduce the Post correspondence problem to this problem.
- *i.e.*, give a program which takes a PCP instance C as input and constructs a formula ϕ such that ϕ is valid iff C has a solution.

The language of predicate logic we use in the proof

- A constant symbol e with intended meaning: the empty string
- Two unary function symbols f_0 and f_1: $f_b(x)$ means the string xb
  - so the binary string $b_1 b_2 \ldots b_l$ can be represented as $f_{b_l}(\ldots(f_{b_2}(f_{b_1}(e)))\ldots)$
  - we abbreviate $f_{b_l}(\ldots(f_{b_2}(f_{b_1}(t)))\ldots)$ as $f_{b_1 b_2 \ldots b_l}(t)$
- A binary predicate symbol P
  - $P(s,t)$ intends to mean: there is a sequence of indices $i_1, i_2, \ldots, i_n$ such that $s = s_{i_1} s_{i_2} \ldots s_{i_n}$ and $t = t_{i_1} t_{i_2} \ldots t_{i_n}$

Given $C = (s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$

Our ϕ is $\phi_1 \wedge \phi_2 \to \exists z P(z, z)$, where

$$\phi_1 = \bigwedge_{i=1}^{k} P(f_{s_i}(e), f_{t_i}(e)),$$

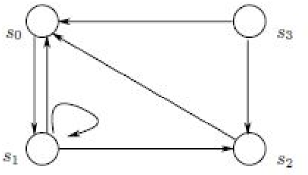
$$\phi_2 = \forall v \forall w [P(v, w) \to \bigwedge_{i=1}^{k} P(f_{s_i}(v), f_{t_i}(w))]$$

Two more negative results

- Satisfiability is not decidable.
- Provability is not decidable.
  - so there is no prefect theorem prover which can mechanically produce a proof of a given formula
  - machines still need human help

Expressiveness of predicate logic

- Software models are often described in terms of directed graphs.
- Such directed graphs can be treated as interpretations of a binary predicate symbol R.



- The validation of many applications requires to show that a 'bad' state cannot be reached from a 'good' state.

The reachability problem

- Given nodes n and n' in a directed graph, is there a finite path of transitions from n to n'.
- *e.g.*, s_2 is reachable from s_0, but s_3 is not
- Question: can we express reachability in predicate logic?
- *i.e.*, can we find a formula $\phi(u, v)$ such that it holds in a directed graph iff there is a path in the graph from the node associated to u to the node associated to v?
- For each $k \geq 0$, we can find a formula $\phi_k(u, v)$ such that it holds in a directed graph iff there is a path of k transitions ...
- However, the answer to the question is 'no'.

Compactness theorem

Theorem. Let Γ be a set of sentences of predicate logic.
If all finite subsets of Γ are satisfiable, then so is Γ.

Proof:

Löwenheim-Skolem Theorem

Theorem. Let ψ be a sentence of predicate logic such that for any natural number $n \geq 1$, there is a model of ψ with at least n elements. Then ψ has a model with infinitely many elements.

Proof:

- Write a formula ϕ_n to express there are at least n elements.
- Let $\Gamma = \{\psi\} \cup \{\phi_n \mid n \geq 1\}$.

Theorem. Reachability is not expressible in predicate logic.

Proof:

- Assume that there is such a formula $\phi(u, v)$.
- Let c and c' be two constants.
- Let $\phi_n(u, v)$ be the formula stating that there is a path of length n from u to v.
- Let $\Gamma = \{\phi[c/u][c'/v]\} \cup \{\neg\phi_n[c/u][c'/v] \mid n \geq 1\}$.

Symbols of predicate logic

1. Logical symbols
   - connective symbols: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
   - quantifier symbols: $\forall, \exists$
   - variable symbols: $x, y, z, \ldots$
   - parentheses and commas
   - the equality symbol $=$
2. Nonlogical symbols
   - constant or individual symbols: $a, b, c, \ldots$
   - predicate or relation symbols: $P, Q, R, \ldots$, each with an arity
   - function symbols: $f, g, h, \ldots$, each with an arity

Terms and formulas of predicate logic

- **Definition of terms**
  - Variables and constants are atomic terms.
  - If f is an n-ary function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.
- **Definition of formulas**:
  - $P(t_1, \cdots, t_n)$ is an *atomic* formula, where P is an n-ary predicate symbol and $t_1, \cdots, t_n$ are terms.
  - $t_1 = t_2$ is an *atomic* formula
  - If A and B are formulas, so are $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ and $(A \leftrightarrow B)$
  - If A is an formula and x is a variable, then $\forall x A$ and $\exists x A$ are formulas.

Second-order logic: syntax

- Predicate variable symbols: X, Y, Z, ..., each with an arity
- Function variable symbols: F, G, H, ..., each with an arity
- If F is an n-ary function variable symbol and $t_1, \ldots, t_n$ are terms, then $F(t_1, \ldots, t_n)$ is a term.
- If X is an n-ary predicate variable symbol and $t_1, \cdots, t_n$ are terms, then $X(t_1, \cdots, t_n)$ is an *atomic* formula.
- If A is a formula, X a predicate variable and F a function variable, then $\forall X A$, $\exists X A$, $\forall F A$ and $\exists F A$ are formulas.

Interpretations for predicate logic

An interpretation $\mathcal{M}$ for a language L consists of the following:

1. A nonempty set $|\mathcal{M}|$ called the *domain* or *universe of discourse* of $\mathcal{M}$.
2. A denotation assigned to each nonlogical symbol of $\mathcal{L}$:
   - For each constant symbol c, $c^{\mathcal{M}} \in |\mathcal{M}|$;
   - For each n-ary function symbol f, $f^{\mathcal{M}}$ is an n-ary function from $|\mathcal{M}|$ to $|\mathcal{M}|$.
   - For each n-ary predicate symbol, $P^{\mathcal{M}}$ is an n-ary relation on $|\mathcal{M}|$.
3. for the equality symbol $=$, $=^{\mathcal{M}}$ is the identity relation on $|\mathcal{M}|$.

Object assignments

- Definition. An object assignment l for an interpretation $\mathcal{M}$ is a mapping from variables such that
  - For each individual variable x, $l(x)$ is an element of $|\mathcal{M}|$
  - For each n-ary predicate variable symbol X, $l(X)$ is an n-ary relation on $|\mathcal{M}|$
  - For each n-ary function variable symbol F, $l(F)$ is an n-ary function from $|\mathcal{M}|$ to $|\mathcal{M}|$.
- Notation. If X is an n-ary predicate variable and R an n-ary relation on $|\mathcal{M}|$, then the object assignment $l[X \mapsto R]$ is the same as l except it maps X to R.
- Similarly, we have the notation $l[F \mapsto h]$ where F is a function variable.

Denotation of terms

Let $\mathcal{M}$ be an interpretation for L, l an object assignment for $\mathcal{M}$, and t a term. The denotation of t in $\mathcal{M}$ under l, denoted $t^{\mathcal{M}}[l]$, is defined as follows:

a) if t is a variable x, then $t^{\mathcal{M}}[l] = l(x)$

b) if $t = f(t_1, \ldots, t_n)$, then $t^{\mathcal{M}}[l] = f^{\mathcal{M}}(t_1^{\mathcal{M}}[l], \ldots, t_n^{\mathcal{M}}[l])$

We add an item:

c) if $t = F(t_1, \ldots, t_n)$ where F is a function variable, then $t^{\mathcal{M}}[l] = l(F)(t_1^{\mathcal{M}}[l], \ldots, t_n^{\mathcal{M}}[l])$

Truth for formulas

We add 5 items:

- $\mathcal{M} \models_l X(t_1, \cdots, t_n)$ where X is a predicate variable iff $\langle t_1^{\mathcal{M}}[l], \ldots, t_n^{\mathcal{M}}[l] \rangle \in l(X)$
- $\mathcal{M} \models_l \forall X A$ iff $\mathcal{M} \models_{l[X \mapsto R]} A$ for all n-ary relation R on $|\mathcal{M}|$
- $\mathcal{M} \models_l \exists X A$ iff $\mathcal{M} \models_{l[X \mapsto R]} A$ for some n-ary relation R on $|\mathcal{M}|$
- $\mathcal{M} \models_l \forall F A$ iff $\mathcal{M} \models_{l[F \mapsto h]} A$ for all n-ary function h from $|\mathcal{M}|$ to $|\mathcal{M}|$
- $\mathcal{M} \models_l \exists F A$ iff $\mathcal{M} \models_{l[F \mapsto h]} A$ for some n-ary function h from $|\mathcal{M}|$ to $|\mathcal{M}|$

Express reachability in second-order logic

Let $\phi(u, v)$ be $\forall P[\phi_1 \wedge \phi_2 \wedge \phi_3 \to P(u, v)]$, where P is a binary predicate variable, and

1. $\phi_1 = \forall x P(x, x)$
2. $\phi_2 = \forall x \forall y \forall z[P(x, y) \wedge P(y, z) \to P(x, z)]$
3. $\phi_3 = \forall x \forall y[R(x, y) \to P(x, y)]$

Theorem. Let $\mathcal{M}$ be an interpretation for the language $[R]$ and l an object assignment for $\mathcal{M}$. Then $\mathcal{M} \models_l \phi(u, v)$ iff $l(v)$ is reachable from $l(u)$ in $\mathcal{M}$.

Proof: