Predicate Calculus

CS 270 Math Foundations of Computer Science

Jeremy Johnson

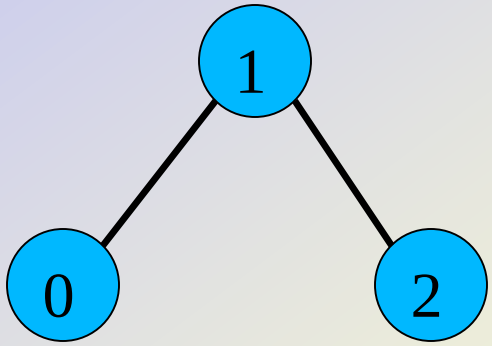
Presentation uses material from
Huth and Ryan, Logic in Computer Science: Modelling and Reasoning about Systems, 2nd Edition

Outline

1. Motivation (program specification & derivation)
    1. Variables, quantifiers and predicates
2. Syntax
    1. Terms and formulas
    2. Quantifiers, scope and substitution
3. Rules of natural deduction for quantifiers
4. Semantics
    1. Models and semantic entailment
5. Undecidability and limitations

Hamiltonian Path

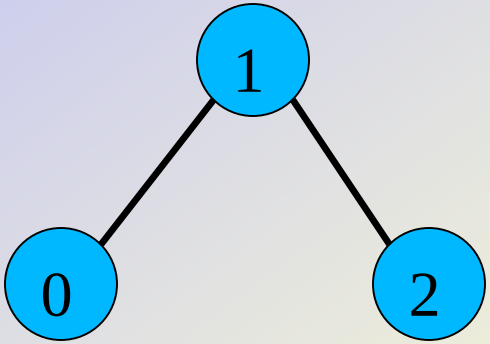
- G = (V,E) is an undirected graph with nodes V = {0,1,…,n-1} and edges E $\subseteq$ V $\times$ V



- V = {0,1,2} E = {(0,1),(1,2)}
  - edge(0,1), edge(1,0), edge(1,2), edge(2,1)

Hamiltonian Path

- A Hamiltonian path is a sequence of n nodes, where n = |V| where each node is visited exactly once (i.e. a permutation of the nodes) following edges

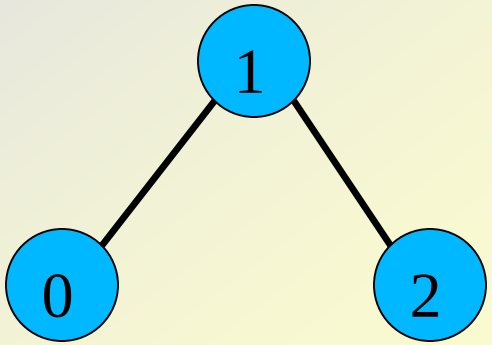


$0 \rightarrow 1 \rightarrow 2$ or $2 \rightarrow 1 \rightarrow 0$

Hamiltonian Path

- path(i,j) is true when the jth node in the path is in the ith location

  - Hamiltonian paths
    - path(0,0), path(1,1), path(2,2)
    - path(0,2), path(1,1), path(2,0)



  - Not Hamiltonian paths
    - path(0,0), path(1,2), path(2,1)
    - path(0,0), path(0,1), path(1,1), path(2,2)
    - path(0,0), path(1,1), path(1,0), path(2,2)

Hamiltonian Path Constraints

- Every node occurs exactly once in the path
  - $\forall j \exists i$ path(i,j)
  - $\forall j \forall i \forall k$ path(i,j) $\wedge$ path(k,j) $\rightarrow$ i = k
- Every location has exactly one node
  - $\forall i \exists j$ path(i,j)
  - $\forall i \forall j \forall k$ path(i,j) $\wedge$ path(i,k) $\rightarrow$ j = k
- Adjacent nodes must be connected by an edge
  - $\forall i \forall j \forall k$ (i<n-1) $\rightarrow$ (path(i,j) $\wedge$ path(i+1,k) $\rightarrow$ edge(j,k))

Simplification

- Every node occurs exactly once in the path
  - $\forall j\ \forall i\ \forall k\ path(i,j) \wedge path(k,j) \rightarrow i = k$

  $\equiv\ \forall j\ \forall i\ \forall k\ (i \neq k\ \rightarrow \neg(path(i,j) \wedge path(k,j))$

  $\equiv\ \forall j\ \forall i\ \forall k\ (i \neq k)\ \rightarrow \neg path(i,j) \vee \neg path(k,j)$

  $\equiv\ \forall j\ \forall i\ \forall k\ (i < k)\ \rightarrow \neg path(i,j) \vee \neg path(k,j)$

Simplification

- Adjacent nodes must be connected by an edge
  - $\forall i\ \forall j\ \forall k\ (i<n-1) \rightarrow ((path(i,j) \wedge path(i+1,k)) \rightarrow edge(j,k))$
  - $\equiv \forall i\ \forall j\ \forall k\ (i<n-1) \rightarrow (\neg edge(j,k) \rightarrow \neg(path(i,j) \wedge path(i+1,k)))$
  - $\equiv \forall i\ \forall j\ \forall k\ (i<n-1) \rightarrow (\neg edge(j,k) \rightarrow (\neg path(i,j) \vee \neg path(i+1,k)))$
  - $\equiv \forall i\ \forall j\ \forall k\ ((i<n-1)\ \&\ \neg edge(j,k)) \rightarrow (\neg\ path(i,j) \vee \neg\ path(i+1,k)))$

Conversion to SAT

- $\forall j \exists i$ path(i,j)
  - $\bigwedge_{j=0..n-1} \bigvee_{i=0..n-1} P_{ij}$
- $\forall j \forall i \forall k$ (i < k) $\rightarrow \neg$path(i,j) $\vee \neg$path(k,j)
  - $\bigwedge_{j=0..n-1} \bigwedge_{i=0..n-1} \bigwedge_{k=i+1..n-1} \neg P_{ij} \vee \neg P_{kj}$
- $\forall i \forall j \forall k$ ((i<n-1) & $\neg$edge(j,k)) $\rightarrow$ path(i,j) $\wedge$ path(i+1,k))
  - $\bigwedge_{i=0..n-2} \bigwedge_{(j,k) \notin E} \neg P_{ij} \vee \neg P_{i+1,k}$

Example 1

- Every student is younger than some instructor

- $\forall x \, (S(x) \rightarrow \exists y (I(y) \wedge Y(x,y))$

  - S(x) : x is a student
  - I(x) : is an instructor
  - Y(x,y) : x is younger than y

Example 2

- Not all birds can fly

- $\neg \forall x (B(x) \rightarrow F(x))$
- $\exists x ((B(x) \wedge \neg F(x))$
  - B(x) : x is a bird
  - F(x) : x can fly

- Semantically equivalent formulas

Example 3

- Every child is younger than its mother

- $\forall x \, \forall y \, (C(x) \wedge M(y,x) \rightarrow Y(x,y))$
  - C(x) : x is child
  - M(x,y) : x is y's mother
  - Y(x,y) : x is younger than y
- $\forall x \, (C(x) \rightarrow Y(x,m(x))$
  - m(x) : function for mother of x

Example 4

- Andy and Paul have the same maternal grandmother
- $\forall x \, \forall y \, \forall u \, \forall v \, (M(x,y) \wedge M(y,a) \wedge M(u,v) \wedge M(v,p) \rightarrow x = u)$
- m(m(a)) = m(m(p))
- a, b : variables for Andy and Paul
- = : binary predicate

Example 5

- Everyone has a mother
- $\forall x \, \exists y \, (M(y,x))$
  - $\exists x \, \forall y \, (M(y,x))$ [not equivalent]

- Everyone has exactly one mother
- $\forall x \, \exists y \, (M(y,x) \wedge \forall z \, (M(z,x) \rightarrow z = y)$

Example 6

- Some people have more than one brother
- $\exists x \, \exists y1 \, \exists y2 \;\; (\, B(y1,x) \wedge B(y2,x) \wedge \neg(y1 = y2) \,)$

Comparison to Propositional Calculus

- Proof calculus for predicate calculus
  - $\phi_1,\ldots,\phi_n \vdash \psi$ [extend natural deduction]
- Provide semantics for predicate calculus
  - $\phi_1,\ldots,\phi_n \models \psi$ [models needed to evaluate functions and predicates – may not be finite]
- Soundness and Completeness
  - $\phi_1,\ldots,\phi_n \models \psi$ holds iff $\phi_1,\ldots,\phi_n \vdash \psi$ is valid
- Undecidable
  - No program exists that can always determine if $\models \phi$ holds

Terms

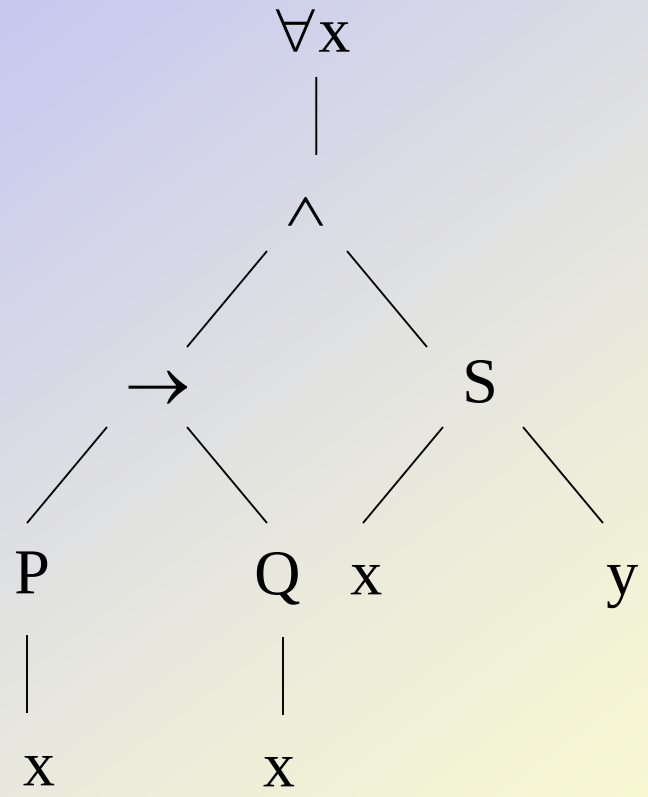
- Terms are made up of variables, constants, and functions
- Term ::=
  - Variable
  - If c is a nullary function c is a term
  - If $t_1,\ldots,t_n$ are terms and f is an n-ary function then $f(t_1,\ldots,t_n)$ is a term

Formulas

- Formula ::=
  - P is a predicate and $t_1,\ldots,t_n$ are terms then $P(t_1,\ldots,t_n)$ is a formula
  - If φ is a formula $\neg\varphi$ is a formula
  - If φ_1 and φ_2 are formulas, $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, $\varphi_1 \rightarrow \varphi_2$ are formulas
  - If φ is a formula and *x* is a variable $\forall x\, \varphi$ and $\exists x\, \varphi$ are formulas

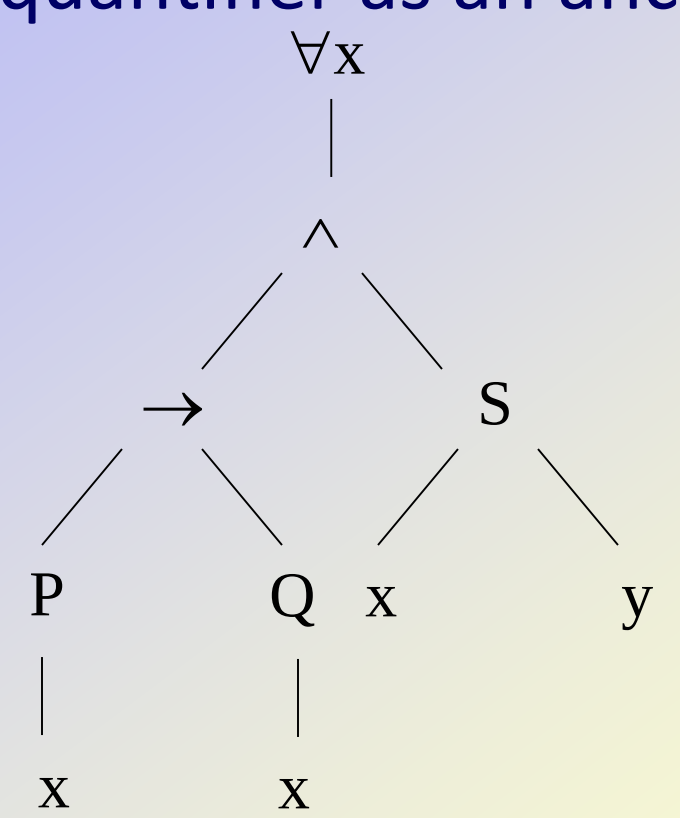
Parse Trees

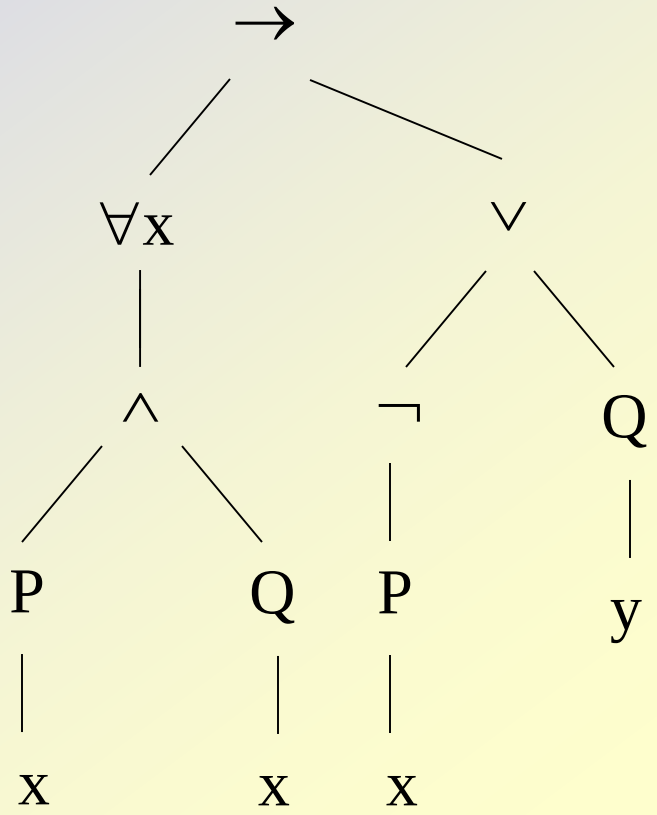
$\forall x \; (\; (\; P(x) \rightarrow Q(x) \;) \wedge S(x,y) \;)$



Free and Bound Variables

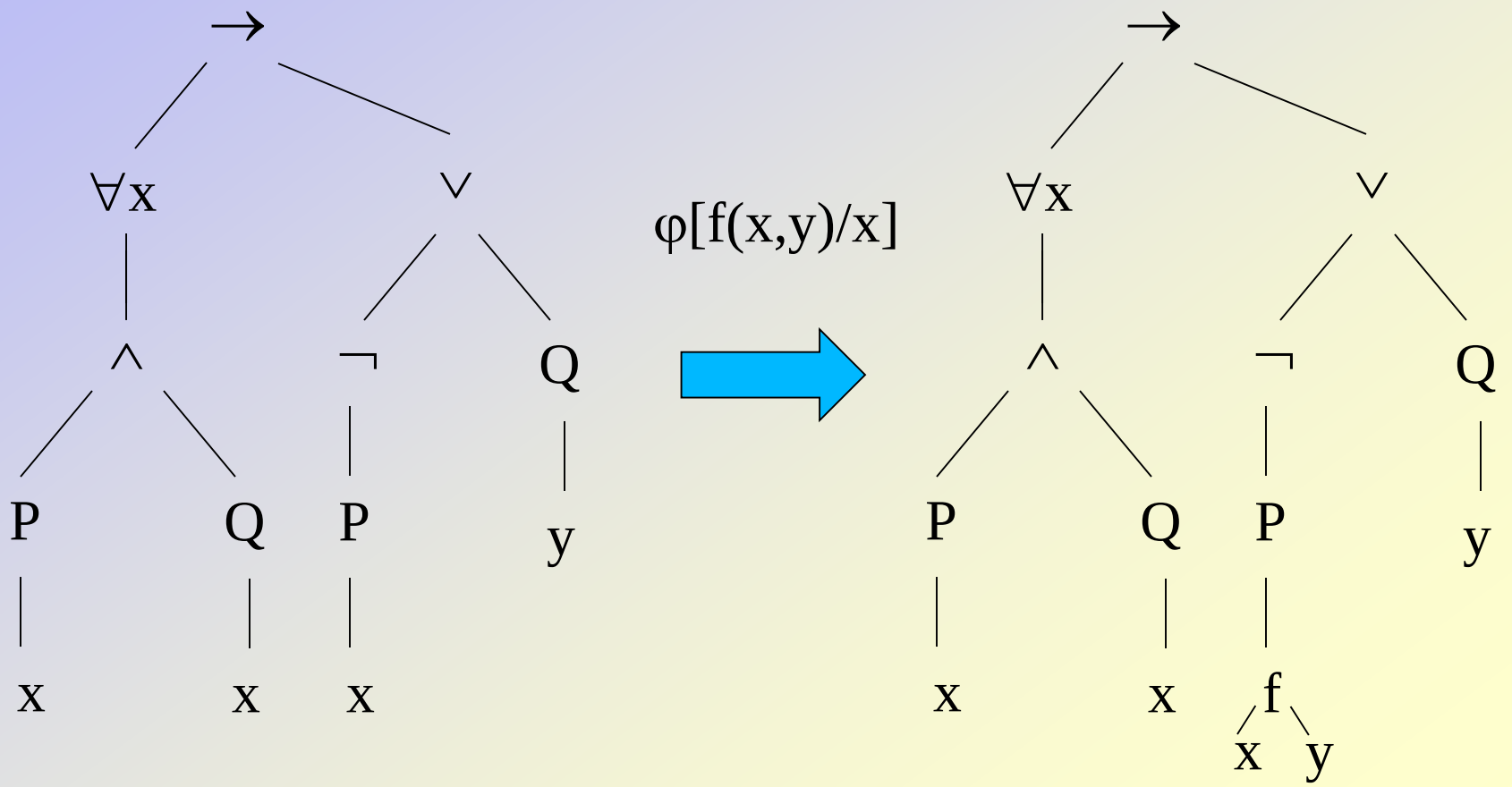
- An occurrence of x in φ is free if it is a leaf node in the parse tree for φ with no quantifier as an ancestor





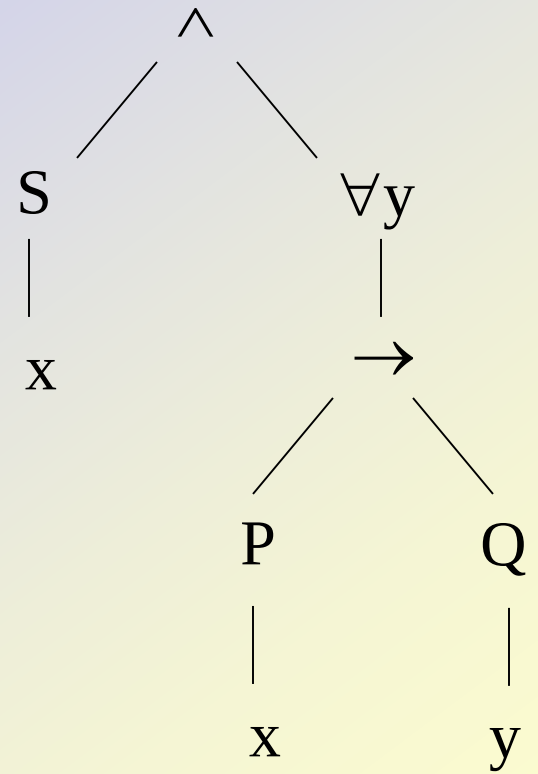
Substitution

- Given a variable x, a term t and a formula φ, $\varphi[t/x]$ is the formula obtained by replacing each free occurrence of x by t



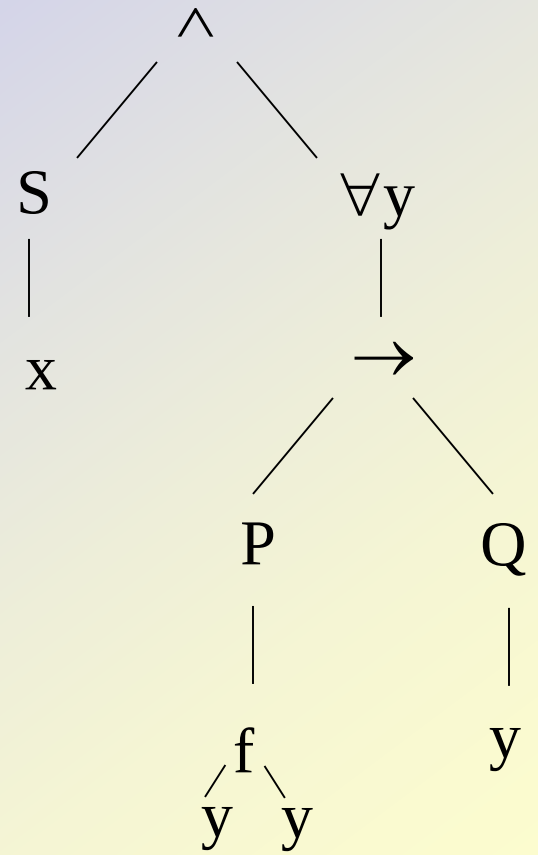
Variable Capture

- t is free for x in φ if no free x occurs in φ in the scope of any quantifier for any variable y occurring in t.



Variable Capture

- t is free for x in φ if no free x occurs in φ in the scope of any quantifier for any variable y occurring in t.



Equality Rules

- Introduction Rule

$$\frac{}{t = t} = i$$

- Elimination Rule

$$\frac{t_1 = t_2 \quad \phi[t_1/x]}{\phi[t_2/x]} = e$$

Equivalence Relation

- Symmetry: $t_1 = t_2 \vdash t_2 = t_1$

1	$t_1 = t_2$	premise
2	$t_1 = t_1$	=i
3	$t_2 = t_1$	=e 1,2

- Transitivity: $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$

1	$t_1 = t_2$	premise
2	$t_2 = t_3$	premise
3	$t_1 = t_3$	=e 2,1

Conjunction Rules

- Introduction Rule

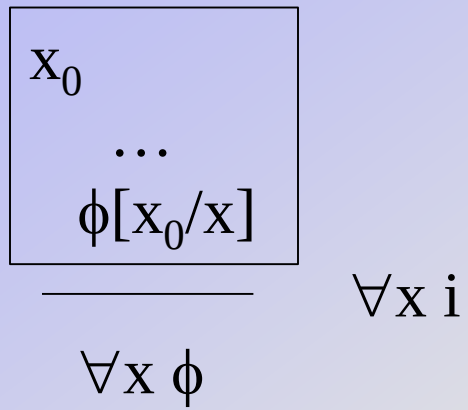
$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge \mathrm{i}$$

- Elimination Rule

$$\frac{\phi \wedge \psi}{\phi} \wedge \mathrm{e1} \qquad \frac{\phi \wedge \psi}{\psi} \wedge \mathrm{e2}$$

Universal Quantification Rules

- Introduction Rule



- Elimination Rule

$$\frac{\forall x\ \phi}{\phi[t/x]}\ \forall x\ e$$

Illegal Substitution Leads to False Reasoning

$\forall x\, \varphi$

$\varphi = \exists y\, (x < y)$

$\varphi[y/x] = \exists y\, (y < y)$

y is not free for x in φ

Example Proof

- $\forall x \left(\mathrm{P}(x) \rightarrow \mathrm{Q}(x)\right), \forall x \mathrm{P}(x) \vdash \forall x \mathrm{Q}(x)$

1	$\forall x \left(\mathrm{P}(x) \rightarrow \mathrm{Q}(x)\right),$	premise
2	$\forall x P(x)$	premise
3	x_0 $\mathrm{P}(x_0) \rightarrow \mathrm{Q}(x_0)$	$\forall x$ e1
4	$\mathrm{P}(x_0)$	$\forall x$ e2
5	$\mathrm{Q}(x_0)$	$\rightarrow$e3,4
6	$\forall x \mathrm{Q}(x)$	$\forall x\ i3-5$

Disjunction Rules

- Introduction Rule

$$\frac{\phi}{\phi \vee \psi} \vee \text{i1} \qquad \frac{\psi}{\phi \vee \psi} \vee \text{i2}$$

- Elimination Rule (proof by case analysis)

$$\frac{\phi \vee \psi \qquad \begin{array}{|l} \hline \phi \\ \dots \\ \chi \\ \hline \end{array} \qquad \begin{array}{|l} \hline \psi \\ \dots \\ \chi \\ \hline \end{array}}{\chi} \vee \text{e}$$

Existential Quantification Rules

- Introduction Rule

$$\frac{\phi[t/x]}{\exists x\, \phi}\ \exists x\ \mathrm{i}$$

- Elimination Rule (proof by case analysis)

$$\frac{\exists x\, \phi \qquad \boxed{\begin{array}{ll} x_0 & \phi[x_0/x] \\ & \vdots \\ & \chi \end{array}}}{\chi}\ \exists\ \mathrm{e}$$

Example Proof

- $\forall x\,(\mathrm{Q}(x)\rightarrow \mathrm{R}(x)), \exists x(\mathrm{P}(x)\wedge \mathrm{Q}(x)) \vdash \exists x(\mathrm{P}(x)\wedge \mathrm{R}(x))$

1	$\forall x\,(\mathrm{Q}(x)\rightarrow \mathrm{R}(x)),$	premise
2	$\exists x(\mathrm{P}(x)\wedge \mathrm{Q}(x))$	premise
3	x_0 $\quad \mathrm{P}(x_0) \wedge \mathrm{Q}(x_0)$	assumption
4	$\mathrm{Q}(x_0) \rightarrow \mathrm{R}(x_0)$	$\forall x$ e1
5	$\mathrm{Q}(x_0)$	$\wedge e_2$ 3
6	$\mathrm{R}(x_0)$	$\rightarrow e$ 4,5
7	$\mathrm{P}(x_0)$	$\wedge e_1$ 3
8	$\mathrm{P}(x_0) \wedge \mathrm{R}(x_0)$	$\wedge$i7,6
9	$\exists x(\mathrm{P}(x)\wedge \mathrm{R}(x))$	$\exists x$i 8
10	$\exists x(\mathrm{P}(x)\wedge \mathrm{R}(x))$	$\exists x$e 2,3-9

Quantifier Equivalences

1. $\neg \forall x\, \varphi \dashv\vdash \exists x \neg \varphi$
2. $\neg \exists x\, \varphi \dashv\vdash \forall x \neg \varphi$
3. $\forall x\, \varphi \wedge \psi \dashv\vdash \forall x\, (\varphi \wedge \psi)$ [x not free in ψ]
4. $\forall x\, \varphi \vee \psi \dashv\vdash \forall x\, (\varphi \vee \psi)$
5. $\exists x\, \varphi \wedge \psi \dashv\vdash \exists x\, (\varphi \wedge \psi)$
6. $\exists x\, \varphi \vee \psi \dashv\vdash \exists x\, (\varphi \vee \psi)$
7. $\forall x\, (\psi \rightarrow \varphi) \dashv\vdash \psi \rightarrow \forall x\, \varphi$
8. $\exists x\, (\varphi \rightarrow \psi) \dashv\vdash \forall x\, \varphi \rightarrow \psi$
9. $\forall x\, (\varphi \rightarrow \psi) \dashv\vdash \exists x\, \varphi \rightarrow \psi$
10. $\exists x\, (\psi \rightarrow \varphi) \dashv\vdash \psi \rightarrow \exists x\, \varphi$

Quantifier Equivalences

1. $\forall x\, \varphi \wedge \forall x\, \psi \dashv\vdash \forall x\, (\varphi \wedge \psi)$
2. $\exists\, x\, \varphi \vee \exists\, x\, \psi \dashv\vdash \exists x\, (\varphi \vee \psi)$
3. $\forall x\, \forall y\, \varphi \dashv\vdash \forall y\, \forall x\, \varphi$
4. $\exists\, x\, \exists\, y\, \varphi \dashv\vdash \exists\, y\, \exists\, x\, \varphi$

De Morgan's Law

$$\neg(p_1 \wedge p_2) \vdash (\neg p_1 \vee \neg p_2)$$

1	$\neg(p_1 \wedge p_2)$	premise
2	$\neg(\neg p_1 \vee \neg p_2)$	assumption
3	$\neg p_1$	assumption
4	$\neg p_1 \vee \neg p_2$	$\vee i_1$ 3
5	$\bot$	$\neg$e4,2
6	p_1	$\neg$e 3-5
7	$\neg p_2$	assumption
8	$\neg p_1 \vee \neg p_2$	$\vee i_2$ 7
9	$\bot$	$\bot$i 4,2
10	p_2	$\neg$e 7-9
11	$(p_1 \wedge p_2)$	$\wedge$i 6,10
12	$\bot$	$\bot$i 11,1
13	$(\neg p_1 \vee \neg p_2)$	$\neg$e 2-12

Generalized De Morgan's Law

$$\neg\forall x\, P(x) \vdash \exists x\, \neg P(x)$$

1	$\neg\forall x\, P(x)$	premise
2	$\neg\exists x\, \neg P(x)$	assumption
3	x_0	
4	$\neg P(x_0)$	assumption
5	$\exists x\, \neg P(x)$	$\exists$x i 4
6	$\bot$	$\bot$i 5,2
7	$P(x_0)$	$\neg$e 4-6
8	$\forall x\, P(x)$	$\forall$x i 3-7
9	$\bot$	$\bot$i 8,1
10	$\exists x\, \neg P(x)$	$\neg$e 2-9

Generalized De Morgan's Law

$$\neg\forall x\, \varphi \vdash \exists x\, \neg\varphi$$

1	$\neg\forall x\varphi$	premise
2	$\neg\exists x$	

Exercise

Prove the reverse $\exists x \neg\varphi \vdash \neg\forall x\, \varphi$

Models

- Let *F* be a set of functions and *P* a set of predicates. A model *M* for (*F*,*P*) consists of
  - A non-empty set A [universe] of concrete values
  - For each nullary $f \in F$ an element of $A = f_M$
  - For each n-ary $f \in F$ a function $f^M : A^n \rightarrow A$
  - For each n-ary $P \in P$ a subset $P^M \subseteq A^n$

Example 1

- F = {i} and P = {R,F}
- i a constant function, R binary and F unary predicates
- Model – A set of states, initial state i, state transitions R, final states F
- A = {a,b,c}
- i^M = a
- R^M = {(a,a),(a,b),(a,c),(b,c), (c,c)}
- F^M = {b,c}

Example 1

- $\exists y\ R(i,y)$
- $\neg F(i)$
- $\forall x\ \forall y\ \forall z\ (R(x,y) \wedge R(x,z) \rightarrow y = z\ )$
- $\forall x\ \exists y\ R(x,y)$

Example 2

- $F = \{e, \cdot\}$ and $P = \{\leq\}$
- e a constant function, $\cdot$ a binary function, $\leq$ a binary predicate
- Model – string prefix
- A = {binary strings}
- $e^M = \varepsilon$, $\cdot^M$ concatenation, $\leq^M$ prefix ordering [011 is a prefix of 011001

Example 2

- $\forall x\,((x \leq x \cdot e) \wedge x \cdot e \leq x))$
- $\forall x\,\exists y\,(y \leq x)$
- $\exists y\,\forall x\,(y \leq x)$
- $\forall x\,\forall y\,\forall z\,((x \leq y) \wedge (y \leq z) \rightarrow (x \leq z))$
- $\forall x\,\forall y\,\forall z\,((x \leq y) \rightarrow (x \cdot z \leq y \cdot z))$

Satisfaction

- Given a model M for (F,P) and given an environment I : var $\mapsto$ A the satisfaction relation $M \vDash_I \varphi$
  - $P(t_1,\ldots,t_n) \mapsto (a_1,\ldots,a_n)$ and $M \vDash I \varphi$ iff $(a_1,\ldots,a_n) \in R^M$
  - $M \vDash_I \forall x\, \psi$ iff $M \vDash_I [x \mapsto a]\psi$ holds for all $a \in A$
  - $M \vDash_I \exists x\, \psi$ iff $M \vDash_I [x \mapsto a]\psi$ holds for some $a \in A$

Satisfaction

- Given a model M for (F,P) and given an environment I : var $\mapsto$ A the satisfaction relation $M \models_I \varphi$
  - $M \models_I \neg\psi$ iff $M \models_I \psi$ does not hold
  - $M \models_I \psi_1 \wedge \psi_2$ iff $M \models_I \psi_1$ and $M \models_I \psi_2$ holds
  - $M \models_I \psi_1 \vee \psi_2$ iff $M \models_I \psi_1$ or $M \models_I \psi_2$ holds
  - $M \models_I \psi_1 \rightarrow \psi_2$ iff $M \models_I \psi_2$ holds whenever $M \models_I \psi_1$ holds

Semantic Entailment

- For propositional calculus: If for all valuations (assignments of variables to truth values) for which all $\phi_1,\ldots,\phi_n$ evaluate to true, ψ also evaluates to true then the semantic entailment relation $\phi_1,\ldots,\phi_n \models \psi$ holds
- Decidable using truth tables

Semantic Entailment

- Let Γ be a set of formulas (possibly infinite) and ψ be a formula from predicate calculus
  - $\Gamma \models \psi$ holds iff for all models *M* and lookup tables *l*, whenever $M \models_l \varphi$ holds for all $\varphi \in \Gamma$ then $M \models_l \psi$ holds as well
  - ψ is satisfiable iff there is some model *M* and lookup table *l* such that $M \models_l \psi$ holds
  - ψ is valid iff $M \models_l \psi$ holds for all models *M* and lookup tables *l*

Soundness and Completeness

- $\phi_1,\ldots,\phi_n \vDash \psi$ holds iff $\phi_1,\ldots,\phi_n \vdash \psi$ is valid

- In particular, $\vDash \psi$, a tautology, $\vdash \psi$ is valid. I.E. ψ is a tautology iff ψ is provable

  - **Soundness** – you can not prove things that are not true in the truth table sense
  - **Completeness** – you can prove anything that is true in the truth table sense

Post Correspondence

- Given a finite sequence $(s_1,t_1),\ldots,(s_k,t_k)$ of pairs of binary strings. Is there a sequence of indices $i_1,i_2,\ldots,i_n$ such that $s_{i1}\cdots s_{in} = t_{i1}\cdots t_{in}$
- Example
  - $s_1 = 1$, $s_2 = 10$, $s_3 = 011$
  - $t_1 = 101$, $t_2 = 00$, $t_3 = 11$
- Solution (1,3,2,3)
  - 101110011
  - 101110011

Undecidability

- Theorem. The decision problem of validity for predicate calculus is undecidable: no program exists which, given any φ, can determine in a finite amount of time if $\models \varphi$
- Proof reduce to Post Correspondence problem. I.E. show that if the decision problem is solvable, we could solve the Post Correspondence problem. This is a contradiction.

Consequences of Undecidability

- From the soundness and completeness of predicate logic, which states that $\models \varphi$ iff $\vdash \varphi$, we infer that we can not decide provability either

- Since φ is satisfiable iff $\neg\varphi$ is valid, we infer that satisfiability for predicate calculus is also undecidable.

Proof

- Construct a formula φ such that $\vDash \varphi$ holds iff the corresponding Post correspondence problem has a solution.

- $\varphi = \varphi_1 \wedge \varphi_2 \rightarrow \varphi_3$
  - $\varphi_1 = \wedge_{i=1..k} P(f_{si}(e),f_{ti}(e))$
  - $\varphi_2 = \forall v \forall w\ (P(v,w) \rightarrow \wedge_{i=1..k} P(f_{si}(v),f_{ti}(w)))$
  - $\varphi_3 = \exists z\ P(z,z)$

Proof

- Find model which tells us the Post Correspondence problem has a solution
- e^M = empty string
- $f_0(s) = s0$, $f_1(s) = s1$
- $P^M = \{(s,t) \mid$ there is a sequence of indices $i_1,i_2,\ldots,i_n$ such that $s_{i1} \cdots s_{in} = t_{i1} \cdots t_{in} \}$
- Since $\vDash \varphi$ holds $M \vDash \varphi$ holds
- Both $M \vDash \varphi_1$ and $M \vDash \varphi_2$ holds so it follows that $M \vDash \varphi_3$ holds. This tells us there is a solution to the given PC problem

Proof

- Conversely assume that the PCP C has a solution $i_1,i_2,\ldots,i_n$
- Show that if *M* is any model having a constant e^M, two unary functions $f0^M$ and $f1^M$, and a binary predicate P^M, then the model has to satisfy φ
- Need to show if $M \vDash \varphi_1 \wedge \varphi_2$ then $M \vDash \varphi_3$

Proof

- Encode strings in A
- interpret(ε) = e^M
- interpret(s0) = f_0^M(interpret(s))
- interpret(s1) = f_1^M(interpret(s))
  - Interpret(011) = f_1^M (f_1^M (f_0^M(e^M)))
- Since $M \vDash \varphi_1$, (interpret(s_i),interpret(t_i)) $\in P^M$
- Since $M \vDash \varphi_2$, If (s,t) $\in P^M$ then (interpret(ss_i),interpret(tt_i)) $\in P^M$

Proof

- Therefore,

$(\text{interpret}(s_{i1}\ldots s_{in}),\text{interpret}(t_{i1}\ldots t_{in})) \in P^M$

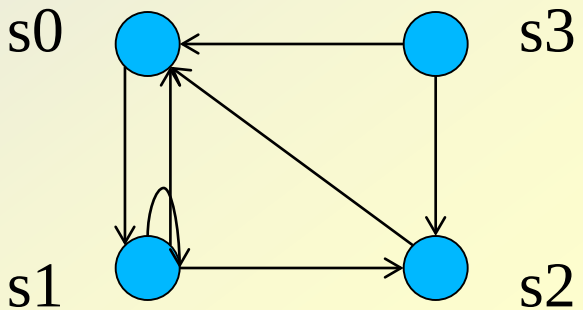
- Since $i_1,i_2,\ldots,i_n$ is a solution to the PCP $\text{interpret}(s_{i1}\ldots s_{in}) = \text{interpret}(t_{i1}\ldots t_{in})$ and $\exists z$ P(z,z) in *M* and thus $M \vDash \varphi_3$

Reachabilty

- When modeling systems via states and state transitions, we want to show that a “bad” state can not be reached from a “good” state.
- Given nodes n and n’ in a directed graph, is there a finite path of transitions from n to n’.

A = {s0,s1,s2,s3}

R^M = {(s0,s1), (s1,s0), (s1,s1),(s1,s2),
(s2,s0),(s3,s0),(s3,s2)}



Compactness Theorem

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

- Proof – uses soundness and completeness and finite length of proofs.

Reachability is Not Expressible

- Can reachability be expressed in predicate calculus?
  - $u=v \vee \exists x\,(R(u,x) \wedge R(x,v)) \vee$
  - $\exists x_1 \exists x_2\,(R(u,x_1) \wedge R(x_1,x_2) \wedge R(x_2,v)) \vee \ldots$
  - This is infinite
- The answer is no! Proof follows from compactness theorem.

Reachability is Not Expressible

- Theorem. There is no predicate-logic formula φ with *u* and *v* as its only free variables and R its only predicate such that φ holds in directed graphs iff there is a path from *u* to *v*.

- Proof. By contradiction. Suppose there is such a formula. Let φ_n be the formula expressing that there is a path from c to c'
  $\varphi_n = \exists x_1 \ldots \exists x_{n-1} (R(c,x_1) \wedge \ldots \wedge R(x_{n-1},c))$.

Reachability is Not Expressible

- Proof. By contradiction. Suppose there is such a formula φ. Let φ_n be the formula expressing that there is a path from c to c'
  $\varphi_n = \exists x_1 \ldots \exists x_{n-1} (R(c,x_1) \wedge \ldots \wedge R(x_{n-1},c))$.
- $\Delta = \{\neg \varphi_i \mid I \geq 0\} \cup \varphi[c/u][c'/v]$ is unsatisfiable, but any finite subset is satisfiable. By compactness this leads to a contradiction and hence there is no such φ.

Reachability via HOL

- Encode Transitive closure

- $\exists P \forall x \forall y \forall z\ (C_1 \wedge C_2 \wedge C_3 \wedge C_4)$
  - $C_1 = P(x,x)$
  - $C_2 = P(x,y) \wedge P(y,z) \rightarrow P(x,y)$
  - $C_3 = P(u,v) \rightarrow \bot$
  - $C_4 = R(x,y) \rightarrow P(x,y)$
- Note quantifier applied to predicate P (Existential second order logic)

Reachability via HOL

- Obtain formula for the existence of a path from u to v by negating previous formula (use DeMorgan's law)

- $\forall P \exists x \exists y \exists z \, (\neg C_1 \wedge \neg C_2 \wedge \neg C_3 \wedge \neg C_4)$

- If both $\forall$ and $\exists$ can range over predicates then second order logic.