

Polymorphism

- Outline
 - *Propositional logic*
 - *First-order logic*
 - *Second-order logic*
 - *System F (Second-order lambda calculus, Girard 1972)*



Propositional Logic Syntax

Propositions:

$$\begin{array}{l} e ::= \top \mid \perp \\ \mid P, Q, R, \dots \\ \mid \neg e \\ \mid e \wedge e \\ \mid e \vee e \\ \mid e \Rightarrow e \end{array}$$



Semantics

- The semantics of constants: $\top = 1, \perp = 0$
- The semantics of a propositional letter is its truth value.
- The semantics of a compound proposition is determined by truth tables.

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

A	B	$A \wedge B$
0	0	1
0	1	1
1	0	0
1	1	1



A proof (Pierce's Law)

A	B	$A \Rightarrow B$	$(A \Rightarrow B) \Rightarrow A$	$((A \Rightarrow B) \Rightarrow A) \Rightarrow A$
0	0	1	0	1
0	1	1	0	1
1	0	0	1	1
1	1	1	1	1



Sequents

- A *sequent* has the form $\Gamma \vdash \Delta$
- Δ is a list of propositions $\alpha_1, \dots, \alpha_n$
- Γ is a *context* containing a list of propositions β_1, \dots, β_n
- We can extend valuations to sequents, to get the following semantics:
 - A sequent $\beta_1, \dots, \beta_n \vdash \alpha_1, \dots, \alpha_n$ is true if some α_i is true whenever β_1, \dots, β_n are all true.



Derivations

- There are two kinds of inference rules
 - *Introduction* rules operate on the right of the turnstile
 - *Elimination* rules operate on the left of the turnstile
- The base axiom

$$\frac{}{\Gamma_1, \alpha, \Gamma_2 \vdash \Delta_1, \alpha, \Delta_2} \text{ axiom}$$



Rule table

$$\overline{\Gamma \vdash \Delta_1, \top, \Delta_2}$$

$$\overline{\Gamma_1, \perp, \Gamma_2 \vdash \Delta}$$

$$\frac{\Gamma \vdash \Delta_1, \alpha, \Delta_2 \quad \Gamma \vdash \Delta_1, \beta, \Delta_2}{\Gamma \vdash \Delta_1, \alpha \wedge \beta, \Delta_2}$$

$$\frac{\Gamma_1, \alpha, \beta, \Gamma_2 \vdash \Delta}{\Gamma_1, \alpha \wedge \beta, \Gamma_2 \vdash \Delta}$$

$$\frac{\Gamma \vdash \Delta_1, \alpha, \beta, \Delta_2}{\Gamma \vdash \Delta_1, \alpha \vee \beta, \Delta_2}$$

$$\frac{\Gamma_1, \alpha, \Gamma_2 \vdash \Delta \quad \Gamma_1, \beta, \Gamma_2 \vdash \Delta}{\Gamma_1, \alpha \vee \beta, \Gamma_2 \vdash \Delta}$$

$$\frac{\Gamma, \alpha \vdash \Delta_1, \beta, \Delta_2}{\Gamma \vdash \Delta_1, \alpha \Rightarrow \beta, \Delta_2}$$

$$\frac{\Gamma_1, \beta, \Gamma_2 \vdash \Delta \quad \Gamma_1, \Gamma_2 \vdash \alpha, \Delta}{\Gamma_1, \alpha \Rightarrow \beta, \Gamma_2 \vdash \Delta}$$



Proving the law of excluded middle

- Every proposition is either true or false

$$\frac{\frac{\overline{A \vdash A}}{\vdash A, \neg A}}{\vdash A \vee \neg A}$$

3
2
1



Pierce's law

$$\frac{\frac{\frac{\frac{\overline{A \vdash B, A} \quad 4}{\vdash A \Rightarrow B, A} \quad 3}{(A \Rightarrow B) \Rightarrow A \vdash A} \quad 2}{\vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A} \quad 1}{\vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A} \quad 1$$



Currying

$$\frac{\frac{\frac{\frac{\frac{\overline{A, B \vdash A} \quad 5}{A, B \vdash A \wedge B} \quad 4}{(A \wedge B) \Rightarrow C, A, B \vdash C} \quad 2}{(A \wedge B) \Rightarrow C \vdash A \Rightarrow B \Rightarrow C} \quad 1}{\vdash ((A \wedge B) \Rightarrow C) \Rightarrow (A \Rightarrow (B \Rightarrow C))} \quad 6}{\frac{\overline{A, B \vdash B} \quad 6}{C, A, B \vdash C} \quad 3} \quad 6$$



Soundness and completeness

Given two sets of propositions:

- $\Gamma = \beta_1, \dots, \beta_n$
- $\Delta = \alpha_1, \dots, \alpha_n$

We say $\Gamma \models \Delta$ if every valuation \mathcal{V} that makes all β_i true also makes at least one α_j true.

- Valuations are truth tables.
- The \models symbol is pronounced “models.”



Definitions

Soundness If $\Gamma \vdash \Delta$, then $\Gamma \models \Delta$

Completeness If $\Gamma \models \Delta$, then $\Gamma \vdash \Delta$



Soundness

- Assume $\Gamma \vdash \Delta$, prove $\Gamma \models \Delta$
- Use induction on the length of the proof $\Gamma \vdash \Delta$



Soundness: the axiom rule

- Suppose the last rule of the proof is the **axiom** rule:

$$\frac{}{\Gamma_1, \alpha, \Gamma_2 \vdash \Delta_1, \alpha, \Delta_2} \textit{ axiom}$$

- Consider any valuation \mathcal{V} that makes all propositions in Γ true.
- Then \mathcal{V} makes at least one proposition (α) in Δ true.
- So $\Gamma_1, \alpha, \Gamma_2 \models \Delta_1, \alpha, \Delta_2$



Soundness: the implies intro rule

- Suppose the last rule of the proof is the **implies intro** rule.

$$\frac{\Gamma, \alpha \vdash \Delta_1, \beta, \Delta_2}{\Gamma \vdash \Delta_1, \alpha \Rightarrow \beta, \Delta_2} \text{ implies intro}$$

- By (proof) induction, we know $\Gamma, \alpha \models \Delta_1, \beta, \Delta_2$.
- So, any valuation that makes Γ, α true also makes some proposition in $\Delta_1, \beta, \Delta_2$ true.



Soundness: implies intro

- Consider any valuation \mathcal{V}' that makes Γ true:
 - If α is false in \mathcal{V}' , then $\alpha \Rightarrow \beta$ is true.
 - If α is true in \mathcal{V}' , then so is some proposition in $\Delta_1, \beta, \Delta_2$.
- It follows that $\Gamma \models \Delta_1, \alpha \Rightarrow \beta, \Delta_2$



Completeness

Any unfinished branch can be extended to a valuation that is a counterexample.

$$\frac{\frac{\frac{A \vdash B, C}{A, \neg B \vdash C} \quad 4 \quad \frac{}{A, C \vdash C} \quad 5}{A, \neg B \vee C \vdash \neg A, C} \quad 3}{A, \neg B \vee C \vdash \neg A \vee C} \quad 2}{\vdash A \Rightarrow (\neg B \vee C) \Rightarrow (\neg A \vee C)} \quad 1$$

The counterexample is $A = 1, B = 0, C = 0$.



First-order logic

- Add *atomic formulas* $f(a_1, \dots, a_n)$ of various arities.
- Add *atomic predicates* $P(a_1, \dots, a_m)$ of various arities.

$$\begin{array}{l} a ::= f(a_1, \dots, a_n) \\ e ::= \top \mid \perp \\ \quad | P(a_1, \dots, a_m) \\ \quad | \neg e \\ \quad | e \wedge e \\ \quad | e \vee e \\ \quad | e \Rightarrow e \\ \quad | \forall v. e \\ \quad | \exists v. e \end{array}$$



Some examples

- $\forall x. \text{big}(x) \Rightarrow \text{heavy}(x)$
- $\forall i. (i + 1) > i$
- $\forall i. \exists j. j > i$
- $\exists i. \forall j. j \leq i$



Adding rules for FOL

$$\frac{\Gamma \vdash \Delta_1, P(t), \Delta_2}{\Gamma \vdash \Delta_1, \exists v.P(v), \Delta_2} \text{ exists intro}$$

$$\frac{\Gamma \vdash \Delta_1, P(c), \Delta_2 \text{ (new } c\text{)}}{\Gamma \vdash \Delta_1, \forall v.P(v), \Delta_2} \text{ all intro}$$

$$\frac{\Gamma_1, P(c), \Gamma_2 \vdash \Delta \text{ (new } c\text{)}}{\Gamma_1, \exists v.P(v), \Gamma_2 \vdash \Delta} \text{ exists elim}$$

$$\frac{\Gamma_1, \forall v.P(v), \Gamma_2, P(t) \vdash \Delta}{\Gamma_1, \forall v.P(v), \Gamma_2 \vdash \Delta} \text{ all elim}$$



A quantifier DeMorgan law

$$\frac{\frac{\frac{\frac{\frac{\frac{\frac{\overline{\varphi(c) \vdash \varphi(c)}^6}{\forall x.\varphi(x) \vdash \varphi(c)}^5}{\neg\varphi(c), \forall x.\varphi(x) \vdash \neg\varphi(c)}^4}{\neg\varphi(c) \vdash \neg(\forall x.\varphi(x))}^3}{\exists x.\neg\varphi(x) \vdash \neg(\forall x.\varphi(x))}^2}{\vdash (\exists x.\neg\varphi(x)) \Rightarrow \neg(\forall x.\varphi(x))}^1$$



Another proof

$$\frac{\frac{\frac{\frac{\frac{\varphi(c) \vdash \varphi(c)}{\varphi(c) \vdash \exists x.\varphi(x)}{(\exists x.\varphi(x)) \Rightarrow \psi, \varphi(c) \vdash \psi}}{(\exists x.\varphi(x)) \Rightarrow \psi, \varphi(c) \Rightarrow \psi}}{(\exists x.\varphi(x)) \Rightarrow \psi \vdash \forall x.\varphi(x) \Rightarrow \psi}}{\vdash ((\exists x.\varphi(x)) \Rightarrow \psi) \Rightarrow (\forall x.\varphi(x) \Rightarrow \psi))}$$

5 4
3 2
1



Soundness and completeness

- Soundness and completeness are similar to propositional logic
 - *The semantics uses “first-order structures”*
 - *Each function symbol is interpreted as a real function*
 - *Each predicate symbol is interpreted as a real predicate*
 - *Quantification is over the set defined by the structure*



Second-order logic

- In FOL, the variables range over the elements of a structure
- Sometimes it is useful to consider all subsets of a structure
 - *“Each bounded non-empty set of reals has a supremum”*
 - *“Each non-empty set of natural numbers has a minimal element”*
- Second-order logic allows quantification over relations



SOL language definition

The language contains:

- individual variables x_0, x_1, \dots
- constants c_0, c_1, \dots and n -ary function symbols f_0^n, f_1^n, \dots
- n -ary predicate variables X_0^n, X_1^n, \dots
- n -ary predicate symbols P_0^n, P_1^n, \dots
- the usual connectives $\neg, \wedge, \vee, \Rightarrow, \exists, \forall$



Rules for predicate quantification

$$\frac{\Gamma \vdash \Delta_1, \varphi[\sigma/X], \Delta_2 \quad (\text{some predicate } \sigma)}{\Gamma \vdash \Delta_1, \exists X.\varphi, \Delta_2} \exists^2\text{intro}$$

$$\frac{\Gamma_1, \varphi[C/X], \Gamma_2 \vdash \Delta \quad (\text{new symbol } C)}{\Gamma_1, \exists X.\varphi, \Gamma_2 \vdash \Delta} \exists^2\text{elim}$$

$$\frac{\Gamma \vdash \Delta_1, \varphi[C/X], \Delta_2 \quad (\text{new symbol } C)}{\Gamma \vdash \Delta_1, \forall X.\varphi, \Delta_2} \forall^2\text{intro}$$

$$\frac{\Gamma_1, \forall X.\varphi, \Gamma_2, \varphi[\sigma/X] \vdash \Delta \quad (\text{some predicate } \sigma)}{\Gamma_1, \forall X.\varphi, \Gamma_2 \vdash \Delta} \forall^2\text{elim}$$



Notation

The substitution $\varphi[\sigma/X]$ means replace each occurrence of $X(t_1, \dots, t_n)$ in φ with $\sigma(t_1, \dots, t_n)$



Negation is definable

$$\frac{\frac{\perp \quad \vdash \quad \perp}{\forall X.X \vdash \perp}}{\vdash \perp \Leftrightarrow \forall X.X}$$



Conjunction is definable

$$\frac{\frac{\frac{\varphi, \psi \vdash \varphi \quad \frac{\dots}{\varphi, \psi, \psi \Rightarrow C \vdash C}}{\varphi, \psi, \varphi \Rightarrow (\psi \Rightarrow C) \vdash C}}{\varphi, \psi \vdash (\varphi \Rightarrow (\psi \Rightarrow C)) \Rightarrow C}}{\varphi, \psi \vdash \forall X. ((\varphi \Rightarrow (\psi \Rightarrow X)) \Rightarrow X)}}{\vdash (\varphi \wedge \psi) \Rightarrow (\forall X. ((\varphi \Rightarrow (\psi \Rightarrow X)) \Rightarrow X))}$$



Operator definitions in SOL

All operators can be defined in terms of \forall and \Rightarrow

$$\begin{aligned}\perp &\Leftrightarrow \forall X.X \\ \varphi \wedge \psi &\Leftrightarrow \forall X.(\varphi \Rightarrow (\psi \Rightarrow X)) \Rightarrow X \\ \varphi \vee \psi &\Leftrightarrow \forall X.((\psi \Rightarrow X) \wedge (\varphi \Rightarrow X)) \Rightarrow X \\ \exists x.\varphi(x) &\Leftrightarrow \forall X.(\forall x.\varphi(x) \Rightarrow X) \Rightarrow X \\ \exists X.\varphi(X) &\Leftrightarrow \forall Y.(\forall X.(\varphi(X) \Rightarrow Y) \Rightarrow Y)\end{aligned}$$



The polymorphic lambda calculus

e	$::=$	v	variables
		$\lambda x:t.e$	abstraction
		$e_1 e_2$	application
		$\Lambda X.e$	type abstraction
		$e[t]$	type application
t	$::=$	X	type variables
		$t_1 \rightarrow t_2$	function types
		$\forall X.t$	type quantification

