Mathematical and Logical Foundations of Computer Science

Revision Lecture

Vincent Rahli

University of Birmingham

What did we cover in this module?

- Propositional logic
- Predicate logic

Today

Revision

- **Propositional Logic**
  - Syntax
  - Natural Deduction proofs
  - Semantics
  - SAT
- **Predicate Logic**
  - Syntax
  - Natural Deduction proofs
  - Semantics

Propositional Logic – Syntax & Informal Semantics

Syntax:

$$P ::= a \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \neg P$$

Propositional Logic – Syntax & Informal Semantics

Syntax:

$$P ::= a \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \neg P$$

Lower-case letters are atoms: p, q, r, etc.

Upper-case letters are (meta-)variables: P, Q, R, etc.

Propositional Logic – Syntax & Informal Semantics

Syntax:

$$P ::= a \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \neg P$$

Lower-case letters are atoms: p, q, r, etc.

Upper-case letters are (meta-)variables: P, Q, R, etc.

Two special atoms:

- $\top$ which stands for True
- $\perp$ which stands for False

Propositional Logic – Syntax & Informal Semantics

Syntax:

$$P ::= a \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \neg P$$

Lower-case letters are atoms: p, q, r, etc.

Upper-case letters are (meta-)variables: P, Q, R, etc.

Two special atoms:

- $\top$ which stands for True
- $\perp$ which stands for False

We also introduced four connectives:

- $P \wedge Q$: we have a proof of both P and Q
- $P \vee Q$: we have a proof of at least one of P and Q
- $P \rightarrow Q$: if we have a proof of P then we have a proof of Q
- $\neg P$: stands for $P \rightarrow \perp$

Propositional Logic – Precedence & Associativity

Precedence: in decreasing order of precedence $\neg, \wedge, \vee, \rightarrow$.

Propositional Logic – Precedence & Associativity

Precedence: in decreasing order of precedence $\neg, \wedge, \vee, \rightarrow$.

For example:

- $\neg P \vee Q$ means $(\neg P) \vee Q$
- $P \wedge Q \vee R$ means $(P \wedge Q) \vee R$
- $P \wedge Q \rightarrow Q \wedge P$ means $(P \wedge Q) \rightarrow (Q \wedge P)$

Propositional Logic – Precedence & Associativity

Precedence: in decreasing order of precedence $\neg, \wedge, \vee, \rightarrow$.

For example:

- $\neg P \vee Q$ means $(\neg P) \vee Q$
- $P \wedge Q \vee R$ means $(P \wedge Q) \vee R$
- $P \wedge Q \rightarrow Q \wedge P$ means $(P \wedge Q) \rightarrow (Q \wedge P)$

Associativity: all operators are right associative

Propositional Logic – Precedence & Associativity

Precedence: in decreasing order of precedence $\neg, \wedge, \vee, \rightarrow$.

For example:

- $\neg P \vee Q$ means $(\neg P) \vee Q$
- $P \wedge Q \vee R$ means $(P \wedge Q) \vee R$
- $P \wedge Q \rightarrow Q \wedge P$ means $(P \wedge Q) \rightarrow (Q \wedge P)$

Associativity: all operators are right associative

For example:

- $P \vee Q \vee R$ means $P \vee (Q \vee R)$.
- $P \wedge Q \wedge R$ means $P \wedge (Q \wedge R)$.
- $P \rightarrow Q \rightarrow R$ means $P \rightarrow (Q \rightarrow R)$.

Propositional Logic – Precedence & Associativity

Precedence: in decreasing order of precedence $\neg, \wedge, \vee, \rightarrow$.

For example:

- $\neg P \vee Q$ means $(\neg P) \vee Q$
- $P \wedge Q \vee R$ means $(P \wedge Q) \vee R$
- $P \wedge Q \rightarrow Q \wedge P$ means $(P \wedge Q) \rightarrow (Q \wedge P)$

Associativity: all operators are right associative

For example:

- $P \vee Q \vee R$ means $P \vee (Q \vee R)$.
- $P \wedge Q \wedge R$ means $P \wedge (Q \wedge R)$.
- $P \rightarrow Q \rightarrow R$ means $P \rightarrow (Q \rightarrow R)$.

What are the parse trees of the above formulas?

Propositional Logic – Constructive Natural Deduction

Constructive Natural Deduction rules:

$$\frac{\bot}{A}\ [\bot E] \qquad \frac{}{\top}\ [\top I] \qquad \dfrac{\begin{array}{c}\overline{A}^{\,1} \\ \vdots \\ B\end{array}}{A \to B}\ 1\ [\to I] \qquad \frac{A \to B \quad A}{B}\ [\to E]$$

$$\dfrac{\begin{array}{c}\overline{A}^{\,1} \\ \vdots \\ \bot\end{array}}{\neg A}\ 1\ [\neg I] \qquad \frac{\neg A \quad A}{\bot}\ [\neg E]$$

$$\frac{A}{A \vee B}\ [\vee I_L] \qquad \frac{A}{B \vee A}\ [\vee I_R] \qquad \frac{A \vee B \quad A \to C \quad B \to C}{C}\ [\vee E]$$

$$\frac{A \quad B}{A \wedge B}\ [\wedge I] \qquad \frac{A \wedge B}{B}\ [\wedge E_R] \qquad \frac{A \wedge B}{A}\ [\wedge E_L]$$

Propositional Logic – Classical Reasoning

Classical Natural Deduction includes all the Constructive Natural Deduction rules, plus:

$$\frac{}{A \vee \neg A}\ [LEM] \qquad \frac{\neg\neg A}{A}\ [DNE]$$

Propositional Logic – Natural Deduction

Typical proof patterns when doing proofs bottom-up:

Propositional logic:

- $\vee$: first $[\vee E]$ then $[\vee I]$
- $\wedge$: first $[\wedge I]$ then $[\vee E]$
- $\rightarrow$: first $[\rightarrow I]$ then $[\rightarrow E]$
- $\neg$: first $[\neg I]$ then $[\neg E]$

Predicate logic:

- $\exists$: first $[\exists E]$ then $[\exists I]$
- $\forall$: first $[\forall I]$ then $[\forall E]$

Propositional Logic – Example of a Constructive Proof

Prove $(A \to C) \to \neg C \to \neg(A \wedge B)$

Propositional Logic – Example of a Constructive Proof

Prove $(A \to C) \to \neg C \to \neg(A \land B)$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \to C}\,1 \quad \dfrac{\dfrac{}{A \land B}\,3}{A}\,[\land E_L]}{C}\,[\to E] \quad \dfrac{}{\neg C}\,2}{\bot}\,[\neg E]}{\dfrac{\neg(A \land B)}{}}\,3\ [\neg I]}{}
$$

$$
\dfrac{\dfrac{\neg(A \land B)}{\neg C \to \neg(A \land B)}\,2\ [\to I]}{(A \to C) \to \neg C \to \neg(A \land B)}\,1\ [\to I]
$$

Propositional Logic – Example of a Classical Proof

Prove $(A \to (B \vee C)) \to \neg B \to \neg A \vee C$

Propositional Logic – Example of a Classical Proof

Prove $(A \to (B \vee C)) \to \neg B \to \neg A \vee C$

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \to (B \vee C)}\,{\scriptstyle 1} \quad \dfrac{}{A}\,{\scriptstyle 3}}{B \vee C}\,{\scriptstyle [\to E]} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{B}\,{\scriptstyle 5} \quad \dfrac{}{\neg B}\,{\scriptstyle 2}}{\bot}\,{\scriptstyle [\neg E]}}{C}\,{\scriptstyle [\bot\, E]}}{B \to C}\,{\scriptstyle 4\ [\to I]} \quad \dfrac{\dfrac{}{C}\,{\scriptstyle 5}}{C \to C}\,{\scriptstyle 5\ [\to I]}}{C}\,{\scriptstyle [\vee E]}}{\neg A \vee C}\,{\scriptstyle [\vee I_R]}}{A \to \neg A \vee C}\,{\scriptstyle 3\ [\to I]}
}{}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{A \vee \neg A}\,{\scriptstyle [LEM]} \quad \dfrac{\vdots}{A \to \neg A \vee C}\,{\scriptstyle 3\ [\to I]} \quad \dfrac{\dfrac{\dfrac{}{\neg A}\,{\scriptstyle 6}}{\neg A \vee C}\,{\scriptstyle [\vee I_L]}}{\neg A \to \neg A \vee C}\,{\scriptstyle 6\ [\to I]}}{\neg A \vee C}\,{\scriptstyle [\vee E]}}{\neg B \to \neg A \vee C}\,{\scriptstyle 2\ [\to I]}}{(A \to (B \vee C)) \to \neg B \to \neg A \vee C}\,{\scriptstyle 1\ [\to I]}
$$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$
- $\phi(A \vee B) = \mathbf{T}$ iff either $\phi(A) = \mathbf{T}$ or $\phi(B) = \mathbf{T}$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \textbf{T}$
- $\phi(\bot) = \textbf{F}$
- $\phi(A \vee B) = \textbf{T}$ iff either $\phi(A) = \textbf{T}$ or $\phi(B) = \textbf{T}$
- $\phi(A \wedge B) = \textbf{T}$ iff both $\phi(A) = \textbf{T}$ and $\phi(B) = \textbf{T}$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$
- $\phi(A \vee B) = \mathbf{T}$ iff either $\phi(A) = \mathbf{T}$ or $\phi(B) = \mathbf{T}$
- $\phi(A \wedge B) = \mathbf{T}$ iff both $\phi(A) = \mathbf{T}$ and $\phi(B) = \mathbf{T}$
- $\phi(A \to B) = \mathbf{T}$ iff $\phi(B) = \mathbf{T}$ whenever $\phi(A) = \mathbf{T}$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$
- $\phi(A \vee B) = \mathbf{T}$ iff either $\phi(A) = \mathbf{T}$ or $\phi(B) = \mathbf{T}$
- $\phi(A \wedge B) = \mathbf{T}$ iff both $\phi(A) = \mathbf{T}$ and $\phi(B) = \mathbf{T}$
- $\phi(A \to B) = \mathbf{T}$ iff $\phi(B) = \mathbf{T}$ whenever $\phi(A) = \mathbf{T}$
- $\phi(\neg A) = \mathbf{T}$ iff $\phi(A) = \mathbf{F}$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$
- $\phi(A \vee B) = \mathbf{T}$ iff either $\phi(A) = \mathbf{T}$ or $\phi(B) = \mathbf{T}$
- $\phi(A \wedge B) = \mathbf{T}$ iff both $\phi(A) = \mathbf{T}$ and $\phi(B) = \mathbf{T}$
- $\phi(A \rightarrow B) = \mathbf{T}$ iff $\phi(B) = \mathbf{T}$ whenever $\phi(A) = \mathbf{T}$
- $\phi(\neg A) = \mathbf{T}$ iff $\phi(A) = \mathbf{F}$

Satisfaction & validity:

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$
- $\phi(A \vee B) = \mathbf{T}$ iff either $\phi(A) = \mathbf{T}$ or $\phi(B) = \mathbf{T}$
- $\phi(A \wedge B) = \mathbf{T}$ iff both $\phi(A) = \mathbf{T}$ and $\phi(B) = \mathbf{T}$
- $\phi(A \rightarrow B) = \mathbf{T}$ iff $\phi(B) = \mathbf{T}$ whenever $\phi(A) = \mathbf{T}$
- $\phi(\neg A) = \mathbf{T}$ iff $\phi(A) = \mathbf{F}$

Satisfaction & validity:

- Given a valuation ϕ, we say that ϕ **satisfies** A if $\phi(A) = \mathbf{T}$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$
- $\phi(A \vee B) = \mathbf{T}$ iff either $\phi(A) = \mathbf{T}$ or $\phi(B) = \mathbf{T}$
- $\phi(A \wedge B) = \mathbf{T}$ iff both $\phi(A) = \mathbf{T}$ and $\phi(B) = \mathbf{T}$
- $\phi(A \to B) = \mathbf{T}$ iff $\phi(B) = \mathbf{T}$ whenever $\phi(A) = \mathbf{T}$
- $\phi(\neg A) = \mathbf{T}$ iff $\phi(A) = \mathbf{F}$

Satisfaction & validity:

- Given a valuation ϕ, we say that ϕ **satisfies** A if $\phi(A) = \mathbf{T}$
- A is **satisfiable** if there exists a valuation ϕ on atomic propositions such that $\phi(A) = \mathbf{T}$

Propositional Logic – Semantics

A **valuation** ϕ assigns **T** or **F** with each atom

A valuation is **extended** to all formulas as follows:

- $\phi(\top) = \mathbf{T}$
- $\phi(\bot) = \mathbf{F}$
- $\phi(A \vee B) = \mathbf{T}$ iff either $\phi(A) = \mathbf{T}$ or $\phi(B) = \mathbf{T}$
- $\phi(A \wedge B) = \mathbf{T}$ iff both $\phi(A) = \mathbf{T}$ and $\phi(B) = \mathbf{T}$
- $\phi(A \to B) = \mathbf{T}$ iff $\phi(B) = \mathbf{T}$ whenever $\phi(A) = \mathbf{T}$
- $\phi(\neg A) = \mathbf{T}$ iff $\phi(A) = \mathbf{F}$

Satisfaction & validity:

- Given a valuation ϕ, we say that ϕ **satisfies** A if $\phi(A) = \mathbf{T}$
- A is **satisfiable** if there exists a valuation ϕ on atomic propositions such that $\phi(A) = \mathbf{T}$
- A is **valid** if $\phi(A) = \mathbf{T}$ for all possible valuations ϕ

Truth Tables

We can use **truth tables** to check whether propositions are valid:

A	B	$A \vee B$
T	**T**	**T**
T	**F**	**T**
F	**T**	**T**
F	**F**	**F**

A	B	$A \wedge B$
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**F**

P	Q	$P \rightarrow Q$
T	**T**	**T**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

A	$\neg A$
T	**F**
F	**T**

A proposition is (semantically) valid if the last column in its truth table only contains **T**

Propositional Logic – Validity

These techniques can be used to prove the validity of propositions:

- syntactic: a **Natural Deduction** proof
- semantical: a **truth table** with only **T** in the last column

Propositional Logic – Validity

These techniques can be used to prove the validity of propositions:

- syntactic: a **Natural Deduction** proof
- semantical: a **truth table** with only **T** in the last column

We saw that:

- a formula A is provable in **Natural Deduction**
- **iff** A is **semantically valid**

Propositional Logic – Validity

These techniques can be used to prove the validity of propositions:

- syntactic: a **Natural Deduction** proof
- semantical: a **truth table** with only **T** in the last column

We saw that:

- a formula A is provable in **Natural Deduction**
- **iff** A is **semantically valid**

This is true about the classical versions of these deduction systems

Propositional Logic – Validity

These techniques can be used to prove the validity of propositions:

- syntactic: a **Natural Deduction** proof
- semantical: a **truth table** with only **T** in the last column

We saw that:

- a formula A is provable in **Natural Deduction**
- **iff** A is **semantically valid**

This is true about the classical versions of these deduction systems

Therefore, if you manage to prove a formula A using **Natural Deduction** then

- A is semantically valid by soundness
- A is satisfiable
- A is <u>not</u> falsifiable

Propositional Logic – Validity

These techniques can be used to prove the validity of propositions:

- syntactic: a **Natural Deduction** proof
- semantical: a **truth table** with only **T** in the last column

We saw that:

- a formula A is provable in **Natural Deduction**
- **iff** A is **semantically valid**

This is true about the classical versions of these deduction systems

Therefore, if you manage to prove a formula A using **Natural Deduction** then

- A is semantically valid by soundness
- A is satisfiable
- A is <u>not</u> falsifiable

Is $(A \rightarrow C) \rightarrow \neg C \rightarrow \neg(A \wedge B)$ valid/satisfiable?

Satisfiability of CNF formulas

Problem definition: Given a CNF formula can we set **T** or **F** value to each variable/atom to satisfy the formula?

Satisfiability of CNF formulas

Problem definition: Given a CNF formula can we set **T** or **F** value to each variable/atom to satisfy the formula?

- **Example**: Consider the formula $(A \vee \neg B) \wedge (C \vee B)$

Satisfiability of CNF formulas

Problem definition: Given a CNF formula can we set **T** or **F** value to each variable/atom to satisfy the formula?

- **Example**: Consider the formula $(A \vee \neg B) \wedge (C \vee B)$
- Is it satisfiable?

Satisfiability of CNF formulas

Problem definition: Given a CNF formula can we set **T** or **F** value to each variable/atom to satisfy the formula?

- **Example**: Consider the formula $(A \vee \neg B) \wedge (C \vee B)$
- Is it satisfiable?
- Satisfiable by setting $A = \mathbf{T}$, $B = \mathbf{F}$ and $C = \mathbf{T}$

Satisfiability of CNF formulas

Problem definition: Given a CNF formula can we set **T** or **F** value to each variable/atom to satisfy the formula?

- **Example**: Consider the formula $(A \vee \neg B) \wedge (C \vee B)$
- Is it satisfiable?
- Satisfiable by setting $A = \mathbf{T}$, $B = \mathbf{F}$ and $C = \mathbf{T}$
- Known as **CNF Satisfiability** or simply **SAT**

Predicate Logic – Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

Predicate Logic – Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if f has arity n
- p ranges over predicate symbols
- $p(t_1, \ldots, t_n)$ is a well-formed formula only if p has arity n
- $\top$ and $\bot$ are predicate symbols of arity 0

Predicate Logic – Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t,\ldots,t) \\
P & ::= & p(t,\ldots,t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1,\ldots,t_n)$ is a well-formed term only if f has arity n
- p ranges over predicate symbols
- $p(t_1,\ldots,t_n)$ is a well-formed formula only if p has arity n
- $\top$ and $\bot$ are predicate symbols of arity 0

The pair of a collection of function symbols, and a collection of predicate symbols, along with their arities, is called a **signature**.

Predicate Logic – Syntax

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if f has arity n
- p ranges over predicate symbols
- $p(t_1, \ldots, t_n)$ is a well-formed formula only if p has arity n
- $\top$ and $\bot$ are predicate symbols of arity 0

The pair of a collection of function symbols, and a collection of predicate symbols, along with their arities, is called a **signature**.

The scope of a quantifier extends as far right as possible. E.g., $P \wedge \forall x.p(x) \vee q(x)$ is read as $P \wedge \forall x.(p(x) \vee q(x))$

Predicate Logic – Substitution

Substitution is defined recursively on terms and formulas:

$P[x\backslash t]$ substitute all the free occurrences of x in P with t.

Predicate Logic – Substitution

Substitution is defined recursively on terms and formulas:
$P[x\backslash t]$ substitute all the free occurrences of x in P with t.

$$
\begin{array}{lcl}
x[x\backslash t] & = & t \\
x[y\backslash t] & = & x \\
(f(t_1,\ldots,t_n))[x\backslash t] & = & f(t_1[x\backslash t],\ldots,t_n[x\backslash t]) \\
(p(t_1,\ldots,t_n))[x\backslash t] & = & p(t_1[x\backslash t],\ldots,t_n[x\backslash t]) \\
\hline
(\neg P)[x\backslash t] & = & \neg P[x\backslash t] \\
(P_1 \wedge P_2)[x\backslash t] & = & P_1[x\backslash t] \wedge P_2[x\backslash t] \\
(P_1 \vee P_2)[x\backslash t] & = & P_1[x\backslash t] \vee P_2[x\backslash t] \\
(P_1 \rightarrow P_2)[x\backslash t] & = & P_1[x\backslash t] \rightarrow P_2[x\backslash t] \\
\hline
(\forall x.P)[x\backslash t] & = & \forall x.P \\
(\exists x.P)[x\backslash t] & = & \exists x.P \\
(\forall y.P)[x\backslash t] & = & \forall y.P[x\backslash t], \text{ if } y \notin \mathtt{fv}(t) \\
(\exists y.P)[x\backslash t] & = & \exists y.P[x\backslash t], \text{ if } y \notin \mathtt{fv}(t)
\end{array}
$$

Predicate Logic – Substitution

Substitution is defined recursively on terms and formulas:

$P[x\backslash t]$ substitute all the free occurrences of x in P with t.

$$
\begin{array}{lcl}
x[x\backslash t] & = & t \\
x[y\backslash t] & = & x \\
(f(t_1,\ldots,t_n))[x\backslash t] & = & f(t_1[x\backslash t],\ldots,t_n[x\backslash t]) \\
(p(t_1,\ldots,t_n))[x\backslash t] & = & p(t_1[x\backslash t],\ldots,t_n[x\backslash t]) \\
\hline
(\neg P)[x\backslash t] & = & \neg P[x\backslash t] \\
(P_1 \wedge P_2)[x\backslash t] & = & P_1[x\backslash t] \wedge P_2[x\backslash t] \\
(P_1 \vee P_2)[x\backslash t] & = & P_1[x\backslash t] \vee P_2[x\backslash t] \\
(P_1 \rightarrow P_2)[x\backslash t] & = & P_1[x\backslash t] \rightarrow P_2[x\backslash t] \\
\hline
(\forall x.P)[x\backslash t] & = & \forall x.P \\
(\exists x.P)[x\backslash t] & = & \exists x.P \\
(\forall y.P)[x\backslash t] & = & \forall y.P[x\backslash t], \text{ if } y \notin \mathtt{fv}(t) \\
(\exists y.P)[x\backslash t] & = & \exists y.P[x\backslash t], \text{ if } y \notin \mathtt{fv}(t)
\end{array}
$$

The additional **conditions** ensure that **free variables do not get captured**.

Predicate Logic – Substitution

Substitution is defined recursively on terms and formulas:

$P[x\backslash t]$ substitute all the free occurrences of x in P with t.

$$
\begin{array}{lcl}
x[x\backslash t] & = & t \\
x[y\backslash t] & = & x \\
(f(t_1,\ldots,t_n))[x\backslash t] & = & f(t_1[x\backslash t],\ldots,t_n[x\backslash t]) \\
(p(t_1,\ldots,t_n))[x\backslash t] & = & p(t_1[x\backslash t],\ldots,t_n[x\backslash t]) \\
\hline
(\neg P)[x\backslash t] & = & \neg P[x\backslash t] \\
(P_1 \wedge P_2)[x\backslash t] & = & P_1[x\backslash t] \wedge P_2[x\backslash t] \\
(P_1 \vee P_2)[x\backslash t] & = & P_1[x\backslash t] \vee P_2[x\backslash t] \\
(P_1 \rightarrow P_2)[x\backslash t] & = & P_1[x\backslash t] \rightarrow P_2[x\backslash t] \\
\hline
(\forall x.P)[x\backslash t] & = & \forall x.P \\
(\exists x.P)[x\backslash t] & = & \exists x.P \\
(\forall y.P)[x\backslash t] & = & \forall y.P[x\backslash t], \text{ if } y \notin \texttt{fv}(t) \\
(\exists y.P)[x\backslash t] & = & \exists y.P[x\backslash t], \text{ if } y \notin \texttt{fv}(t)
\end{array}
$$

The additional **conditions** ensure that **free variables do not get captured**.

These conditions can always be met by silently renaming bound variables before substituting.

Predicate Logic – Constructive Natural Deduction

Natural Deduction rules for the propositional connectives:

$$\frac{\bot}{A}\ [\bot E] \qquad \frac{}{\top}\ [\top I] \qquad \frac{\begin{array}{c}\overline{A}^{\,1}\\ \vdots \\ B\end{array}}{A \to B}\ 1\ [\to I] \qquad \frac{A \to B \quad A}{B}\ [\to E]$$

$$\frac{\begin{array}{c}\overline{A}^{\,1}\\ \vdots \\ \bot\end{array}}{\neg A}\ 1\ [\neg I] \qquad \frac{\neg A \quad A}{\bot}\ [\neg E]$$

$$\frac{A}{A \vee B}\ [\vee I_L] \qquad \frac{A}{B \vee A}\ [\vee I_R] \qquad \frac{A \vee B \quad A \to C \quad B \to C}{C}\ [\vee E]$$

$$\frac{A \quad B}{A \wedge B}\ [\wedge I] \qquad \frac{A \wedge B}{B}\ [\wedge E_R] \qquad \frac{A \wedge B}{A}\ [\wedge E_L]$$

Predicate Logic – Constructive Natural Deduction

Natural Deduction rules for quantifiers:

$$\frac{P[x\backslash y]}{\forall x.P}\ [\forall I] \qquad \frac{\forall x.P}{P[x\backslash t]}\ [\forall E] \qquad \frac{P[x\backslash t]}{\exists x.P}\ [\exists I] \qquad \frac{\exists x.P \qquad \begin{array}{c}\overline{P[x\backslash y]}^{\,1}\\ \vdots \\ Q\end{array}}{Q}\ 1\ [\exists E]$$

Condition:

- for $[\forall I]$: y must not be free in any not-yet-discharged hypothesis or in $\forall x.P$
- for $[\forall E]$: $\texttt{fv}(t)$ must not clash with $\texttt{bv}(P)$
- for $[\exists I]$: $\texttt{fv}(t)$ must not clash with $\texttt{bv}(P)$
- for $[\exists E]$: y must not be free in Q or in not-yet-discharged hypotheses or in $\exists x.P$

Predicate Logic – Classical Reasoning

Classical Natural Deduction for Predicate Logic

Add the rules:

$$\frac{}{A \vee \neg A}\ [LEM] \qquad\qquad \frac{\neg\neg A}{A}\ [DNE]$$

Predicate Logic – Example of a Proof

Prove $(S_1) \to (S_2) \to \forall x.\texttt{even}(x) \to \neg\texttt{odd}(\texttt{succ}(\texttt{succ}(x)))$ where
$S_1 = \forall x.\texttt{even}(x) \to \texttt{even}(\texttt{succ}(\texttt{succ}(x)))$ and
$S_2 = \forall x.\texttt{odd}(x) \to \neg\texttt{even}(x)$

Predicate Logic – Example of a Proof

Prove $(S_1) \rightarrow (S_2) \rightarrow \forall x.\mathtt{even}(x) \rightarrow \neg\mathtt{odd}(\mathtt{succ}(\mathtt{succ}(x)))$ where
$S_1 = \forall x.\mathtt{even}(x) \rightarrow \mathtt{even}(\mathtt{succ}(\mathtt{succ}(x)))$ and
$S_2 = \forall x.\mathtt{odd}(x) \rightarrow \neg\mathtt{even}(x)$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{S_1}\;1}{\mathtt{E}(x) \rightarrow \mathtt{E}(\mathtt{s}(\mathtt{s}(x)))}\;[\forall E] \qquad \dfrac{}{\mathtt{E}(x)}\;3}{\mathtt{E}(\mathtt{s}(\mathtt{s}(x)))}\;[\rightarrow E] \qquad \dfrac{\dfrac{\overline{S_2}\;2}{\mathtt{O}(\mathtt{s}(\mathtt{s}(x))) \rightarrow \neg\mathtt{E}(\mathtt{s}(\mathtt{s}(x)))}\;[\forall E] \qquad \dfrac{}{\mathtt{O}(\mathtt{s}(\mathtt{s}(x)))}\;4}{\neg\mathtt{E}(\mathtt{s}(\mathtt{s}(x)))}\;[\rightarrow E]}{\bot}\;[\neg E]}{\neg\mathtt{O}(\mathtt{s}(\mathtt{s}(x)))}\;4\;[\neg I]}{\mathtt{E}(x) \rightarrow \neg\mathtt{O}(\mathtt{s}(\mathtt{s}(x)))}\;3\;[\rightarrow I]}{\forall x.\mathtt{E}(x) \rightarrow \neg\mathtt{O}(\mathtt{s}(\mathtt{s}(x)))}\;[\forall I]}{(S_2) \rightarrow \forall x.\mathtt{E}(x) \rightarrow \neg\mathtt{O}(\mathtt{s}(\mathtt{s}(x)))}\;2\;[\rightarrow I]}{(S_1) \rightarrow (S_2) \rightarrow \forall x.\mathtt{E}(x) \rightarrow \neg\mathtt{O}(\mathtt{s}(\mathtt{s}(x)))}\;1\;[\rightarrow I]
$$

Where O stands for `odd`; E stands for `even`; and s stands for `succ`

Predicate Logic – Semantics

Models: a model provides the interpretation of all symbols

Predicate Logic – Semantics

Models: a model provides the interpretation of all symbols

Given a **signature** $\langle\langle f_1^{k_1}, \ldots, f_n^{k_n}\rangle, \langle p_1^{j_1}, \ldots, p_m^{j_m}\rangle\rangle$

- of function symbols f_i of arity k_i, for $1 \leqslant i \leqslant n$
- of predicate symbols p_i of arity j_i, for $1 \leqslant i \leqslant m$

Predicate Logic – Semantics

Models: a model provides the interpretation of all symbols

Given a **signature** $\langle\langle f_1^{k_1}, \ldots, f_n^{k_n}\rangle, \langle p_1^{j_1}, \ldots, p_m^{j_m}\rangle\rangle$

- of function symbols f_i of arity k_i, for $1 \leqslant i \leqslant n$
- of predicate symbols p_i of arity j_i, for $1 \leqslant i \leqslant m$

A **model** is a structure $\langle D, \langle \mathcal{F}_{f_1}, \ldots, \mathcal{F}_{f_n}\rangle, \langle \mathcal{R}_{p_1}, \ldots, \mathcal{R}_{p_m}\rangle\rangle$

- of a non-empty domain D
- interpretations $\mathcal{F}_{f_i}$ for function symbols f_i
- interpretations $\mathcal{R}_{p_i}$ for function symbols p_i

Predicate Logic – Semantics

Models: a model provides the interpretation of all symbols

Given a **signature** $\langle\langle f_1^{k_1}, \ldots, f_n^{k_n}\rangle, \langle p_1^{j_1}, \ldots, p_m^{j_m}\rangle\rangle$

- of function symbols f_i of arity k_i, for $1 \leqslant i \leqslant n$
- of predicate symbols p_i of arity j_i, for $1 \leqslant i \leqslant m$

A **model** is a structure $\langle D, \langle \mathcal{F}_{f_1}, \ldots, \mathcal{F}_{f_n}\rangle, \langle \mathcal{R}_{p_1}, \ldots, \mathcal{R}_{p_m}\rangle\rangle$

- of a non-empty domain D
- interpretations $\mathcal{F}_{f_i}$ for function symbols f_i
- interpretations $\mathcal{R}_{p_i}$ for function symbols p_i

Models of predicate logic replace **truth assignments** for propositional logic

Predicate Logic – Semantics

Models: a model provides the interpretation of all symbols

Given a **signature** $\langle\langle f_1^{k_1}, \dots, f_n^{k_n}\rangle, \langle p_1^{j_1}, \dots, p_m^{j_m}\rangle\rangle$

- of function symbols f_i of arity k_i, for $1 \leqslant i \leqslant n$
- of predicate symbols p_i of arity j_i, for $1 \leqslant i \leqslant m$

A **model** is a structure $\langle D, \langle \mathcal{F}_{f_1}, \dots, \mathcal{F}_{f_n}\rangle, \langle \mathcal{R}_{p_1}, \dots, \mathcal{R}_{p_m}\rangle\rangle$

- of a non-empty domain D
- interpretations $\mathcal{F}_{f_i}$ for function symbols f_i
- interpretations $\mathcal{R}_{p_i}$ for function symbols p_i

Models of predicate logic replace **truth assignments** for propositional logic

Variable valuations:

Predicate Logic – Semantics

Models: a model provides the interpretation of all symbols

Given a **signature** $\langle\langle f_1^{k_1}, \ldots, f_n^{k_n}\rangle, \langle p_1^{j_1}, \ldots, p_m^{j_m}\rangle\rangle$

- of function symbols f_i of arity k_i, for $1 \leqslant i \leqslant n$
- of predicate symbols p_i of arity j_i, for $1 \leqslant i \leqslant m$

A **model** is a structure $\langle D, \langle \mathcal{F}_{f_1}, \ldots, \mathcal{F}_{f_n}\rangle, \langle \mathcal{R}_{p_1}, \ldots, \mathcal{R}_{p_m}\rangle\rangle$

- of a non-empty domain D
- interpretations $\mathcal{F}_{f_i}$ for function symbols f_i
- interpretations $\mathcal{R}_{p_i}$ for function symbols p_i

Models of predicate logic replace **truth assignments** for propositional logic

Variable valuations:

- a partial function v

Predicate Logic – Semantics

Models: a model provides the interpretation of all symbols

Given a **signature** $\langle\langle f_1^{k_1}, \ldots, f_n^{k_n}\rangle, \langle p_1^{j_1}, \ldots, p_m^{j_m}\rangle\rangle$

- of function symbols f_i of arity k_i, for $1 \leqslant i \leqslant n$
- of predicate symbols p_i of arity j_i, for $1 \leqslant i \leqslant m$

A **model** is a structure $\langle D, \langle \mathcal{F}_{f_1}, \ldots, \mathcal{F}_{f_n}\rangle, \langle \mathcal{R}_{p_1}, \ldots, \mathcal{R}_{p_m}\rangle\rangle$

- of a non-empty domain D
- interpretations $\mathcal{F}_{f_i}$ for function symbols f_i
- interpretations $\mathcal{R}_{p_i}$ for function symbols p_i

Models of predicate logic replace **truth assignments** for propositional logic

Variable valuations:

- a partial function v
- that maps variables to D

Predicate Logic – Semantics

Models: a model provides the interpretation of all symbols

Given a **signature** $\langle\langle f_1^{k_1}, \ldots, f_n^{k_n}\rangle, \langle p_1^{j_1}, \ldots, p_m^{j_m}\rangle\rangle$

- of function symbols f_i of arity k_i, for $1 \leqslant i \leqslant n$
- of predicate symbols p_i of arity j_i, for $1 \leqslant i \leqslant m$

A **model** is a structure $\langle D, \langle \mathcal{F}_{f_1}, \ldots, \mathcal{F}_{f_n}\rangle, \langle \mathcal{R}_{p_1}, \ldots, \mathcal{R}_{p_m}\rangle\rangle$

- of a non-empty domain D
- interpretations $\mathcal{F}_{f_i}$ for function symbols f_i
- interpretations $\mathcal{R}_{p_i}$ for function symbols p_i

Models of predicate logic replace **truth assignments** for propositional logic

Variable valuations:

- a partial function v
- that maps variables to D
- i.e., a mapping of the form $x_1 \mapsto d_1, \ldots, x_n \mapsto d_n$

Predicate Logic – Semantics

Given a **model** M with domain D and a **variable valuation** v:

- $[\![t]\!]_v^M$ gives meaning to the term t w.r.t. M and v
- $\models_{M,v} P$ gives meaning to the formula P w.r.t. M and v

Predicate Logic – Semantics

Given a **model** M with domain D and a **variable valuation** v:

- $[\![t]\!]_v^M$ gives meaning to the term t w.r.t. M and v
- $\models_{M,v} P$ gives meaning to the formula P w.r.t. M and v

Meaning of terms:

- $[\![x]\!]_v^M = v(x)$
- $[\![f(t_1, \ldots, t_n)]\!]_v^M = \mathcal{F}_f(\langle [\![t_1]\!]_v^M, \ldots, [\![t_n]\!]_v^M \rangle)$

Predicate Logic – Semantics

Given a **model** M with domain D and a **variable valuation** v:

- $[\![t]\!]_v^M$ gives meaning to the term t w.r.t. M and v
- $\vDash_{M,v} P$ gives meaning to the formula P w.r.t. M and v

Meaning of terms:

- $[\![x]\!]_v^M = v(x)$
- $[\![f(t_1,\ldots,t_n)]\!]_v^M = \mathcal{F}_f(\langle [\![t_1]\!]_v^M,\ldots,[\![t_n]\!]_v^M \rangle)$

Meaning of formulas:

- $\vDash_{M,v} \top$ and $\neg\vDash_{M,v} \bot$
- $\vDash_{M,v} p(t_1,\ldots,t_n)$ iff $\langle [\![t_1]\!]_v^M,\ldots,[\![t_n]\!]_v^M \rangle \in \mathcal{R}_p$
- $\vDash_{M,v} \neg P$ iff $\neg\vDash_{M,v} P$
- $\vDash_{M,v} P \land Q$ iff $\vDash_{M,v} P$ and $\vDash_{M,v} Q$
- $\vDash_{M,v} P \lor Q$ iff $\vDash_{M,v} P$ or $\vDash_{M,v} Q$
- $\vDash_{M,v} P \rightarrow Q$ iff $\vDash_{M,v} Q$ whenever $\vDash_{M,v} P$
- $\vDash_{M,v} \forall x.P$ iff for every $d \in D$ we have $\vDash_{M,(v,x\mapsto d)} P$
- $\vDash_{M,v} \exists x.P$ iff there exists a $d \in D$ such that $\vDash_{M,(v,x\mapsto d)} P$

Predicate Logic – Soundness & Completeness

Natural Deduction for Predicate Logic is

- **sound**, i.e., if $\vdash A$ then $\models A$, and
- **complete**, i.e., if $\models A$ then $\vdash A$

w.r.t. the **model semantics of Predicate Logic**

Predicate Logic – Semantics

Consider the signature: $\langle\langle\rangle, \langle \texttt{even}^1, \texttt{odd}^1 \rangle\rangle$

Predicate Logic – Semantics

Consider the signature: $\langle\langle\rangle, \langle \mathtt{even}^1, \mathtt{odd}^1\rangle\rangle$

Let P be the formula $\forall x.\mathtt{even}(x) \rightarrow (\mathtt{even}(x) \wedge \mathtt{odd}(x))$

Predicate Logic – Semantics

Consider the signature: $\langle\langle\rangle, \langle \mathtt{even}^1, \mathtt{odd}^1\rangle\rangle$

Let P be the formula $\forall x.\mathtt{even}(x) \rightarrow (\mathtt{even}(x) \wedge \mathtt{odd}(x))$

Examples of models of P?

Predicate Logic – Semantics

Consider the signature: $\langle\langle\rangle, \langle \texttt{even}^1, \texttt{odd}^1\rangle\rangle$

Let P be the formula $\forall x.\texttt{even}(x) \rightarrow (\texttt{even}(x) \wedge \texttt{odd}(x))$

Examples of models of P?

- $\langle\mathbb{N}, \langle\rangle, \langle\{\langle n\rangle \mid n \text{ is even}\}, \{\langle n\rangle \mid n \text{ is even}\}\rangle\rangle$
- $\langle\mathbb{N}, \langle\rangle, \langle\{\langle n\rangle \mid n \text{ is odd}\}, \{\langle n\rangle \mid n \text{ is odd}\}\rangle\rangle$
- $\langle\mathbb{N}, \langle\rangle, \langle\{\langle n\rangle \mid \text{True}\}, \{\langle n\rangle \mid \text{True}\}\rangle\rangle$
- $\langle\mathbb{N}, \langle\rangle, \langle\{\langle n\rangle \mid n \text{ is odd}\}, \{\langle n\rangle \mid \text{True}\}\rangle\rangle$
- $\langle\mathbb{N}, \langle\rangle, \langle\{\langle 1\rangle\}, \{\langle 1\rangle\}\rangle\rangle$
- $\langle\mathbb{N}, \langle\rangle, \langle\emptyset, \{\langle 0\rangle\}\rangle\rangle$

Predicate Logic – Semantics

Consider the signature: $\langle\langle\rangle, \langle \texttt{even}^1, \texttt{odd}^1\rangle\rangle$

Let P be the formula $\forall x.\texttt{even}(x) \rightarrow (\texttt{even}(x) \wedge \texttt{odd}(x))$

Examples of models of P?

- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is even}\}, \{\langle n\rangle \mid n \text{ is even}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is odd}\}, \{\langle n\rangle \mid n \text{ is odd}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid \text{True}\}, \{\langle n\rangle \mid \text{True}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is odd}\}, \{\langle n\rangle \mid \text{True}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle 1\rangle\}, \{\langle 1\rangle\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \emptyset, \{\langle 0\rangle\}\rangle\rangle$

Examples of models that are not models of P (i.e., models of $\neg P$)?

Predicate Logic – Semantics

Consider the signature: $\langle\langle\rangle, \langle \texttt{even}^1, \texttt{odd}^1\rangle\rangle$

Let P be the formula $\forall x.\texttt{even}(x) \rightarrow (\texttt{even}(x) \wedge \texttt{odd}(x))$

Examples of models of P?

- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is even}\}, \{\langle n\rangle \mid n \text{ is even}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is odd}\}, \{\langle n\rangle \mid n \text{ is odd}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid \text{True}\}, \{\langle n\rangle \mid \text{True}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is odd}\}, \{\langle n\rangle \mid \text{True}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle 1\rangle\}, \{\langle 1\rangle\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \emptyset, \{\langle 0\rangle\}\rangle\rangle$

Examples of models that are not models of P (i.e., models of $\neg P$)?

- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is even}\}, \{\langle n\rangle \mid n \text{ is odd}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is odd}\}, \{\langle n\rangle \mid n \text{ is even}\}\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle n\rangle \mid n \text{ is odd}\}, \emptyset\rangle\rangle$
- $\langle \mathbb{N}, \langle\rangle, \langle \{\langle 0\rangle\}, \emptyset\rangle\rangle$