Part 2: Predicate Calculus (a)

COM2107 Logic in Computer Science

Slides by Georg Struth

Adapted by Jonni Virtema

School of Computer Science

Session code: xx-xx-xx

Limits of propositional logic

propositional logic takes propositional variables as atomic propositions

some non-trivial system properties can be encoded/analysed
(expressive enough to express NP-complete problems)

yet it's not expressive enough for many modelling/analysis tasks
let alone for formalising mathematics

even the Socrates syllogism can't be formalised!

Limits of propositional logic

internal structure of propositions can't be captured e.g.

All children love ice cream.
Some students understand implication.
No electron has positive charge.

quantifications such as "for all" / "there exists" can't be represented

Limits of propositional logic

reasoning about properties of entities and their relationships e.g.

$$\textit{If } x \leq 7 \textit{ and } 9 \leq y, \textit{ then } x \leq y.$$
$$\textit{If } y = 2 \cdot x + 6, \textit{ then } x = \frac{1}{2} \cdot y - 3.$$

is impossible

Towards predicate logic

more refined logic requires

1. more expressive language/syntax
2. new inference rules for reasoning with this language
3. more fine grained semantics beyond truth values for propositions

Towards predicate logic

syntax should capture

- terms e.g. $2 \cdot x + 6$
- relations/predicates e.g. $x \leq 7$
- quantification in sentences e.g. "*All* x *are* φ" or "*Some* x *are* φ"

semantics should

- supply objects, functions, relations that match expressions of syntax
- allow evaluating terms e.g. $2 \cdot 4 + 6$ or $9 \leq 7$ in domains e.g. $\mathbb{N}$

predicate logic satisfies these requirements

Recap of propositional logic

Propositional logic is about modelling of Boolean statements:

$$\varphi ::= \top \mid \bot \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi),$$

where p is a propositional variable.

- "$1 + 1 = 2$" (regarded as a statement)
- $(p \to q) \to (\neg q \to \neg p)$
- $\mathrm{it\_rains} \vee \mathrm{it\_shines}$

Recap of propositional logic

Propositional logic is about modelling of Boolean statements:

$$\varphi ::= \top \mid \bot \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi),$$

where p is a propositional variable.

- "$1 + 1 = 2$" (regarded as a statement)
- $(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$
- $\mathrm{it\_rains} \vee \mathrm{it\_shines}$

$$\frac{\text{I carry an umbrella} \qquad \text{If I carry an umbrella, it does not rain}}{\text{It does not rain}}$$

$$\frac{A \qquad A \rightarrow B}{B} \text{ (Modus Ponens)}$$

Predicate logic (i.e., first-order logic)

Predicate logic can describe properties of complex systems.

$$\varphi ::= \top \mid \bot \mid t_1 = t_2 \mid R(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \exists x \varphi \mid \forall x \varphi,$$

t_i are terms (denoting individual objects) and R is a relation between objects.

- Can model and deduce properties and relationships between objects.
- Formulae describe properties of data structures.
- Formulae correspond to database queries.

Predicate logic (i.e., first-order logic)

Predicate logic can describe properties of complex systems.

$$\varphi ::= \top \mid \bot \mid t_1 = t_2 \mid R(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \exists x \varphi \mid \forall x \varphi,$$

t_i are terms (denoting individual objects) and R is a relation between objects.

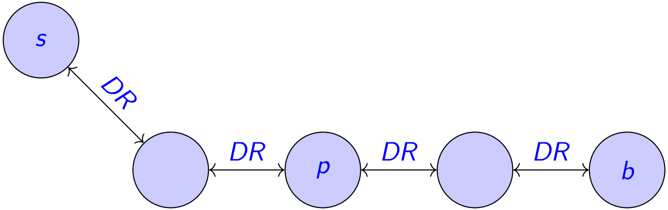
- Can model and deduce properties and relationships between objects.
- Formulae describe properties of data structures.
- Formulae correspond to database queries.
- $\forall x(\text{Morning\_star}(x) \rightarrow (x = \text{Venus} \vee x = \text{Mercury} \vee x = \text{Sirius}))$
- $\forall x \forall y \Big((\text{sister}(x, y) \wedge \text{female}(y)) \rightarrow \text{sister}(y, x) \Big)$
- $\forall x (\text{person}(x) \rightarrow \exists y \, (\text{ancestor}(y, x) \wedge y \neq \text{father\_of}(x)))$

Examples on the use of predicate logic

- Properties of mathematical structures $(\mathbb{N}, \leq, P_{\mathrm{EVEN}}, +, \cdot)$.
  - $\forall x \forall y \left(P_{\mathrm{EVEN}}(x) \to P_{\mathrm{EVEN}}(x \cdot y)\right)$
  - If $x \leq 7$ and $9 \leq y$ then $x \leq y$.

Examples on the use of predicate logic

- Properties of mathematical structures $(\mathbb{N}, \leq, P_{\mathrm{EVEN}}, +, \cdot)$.
  - $\forall x \forall y\, (P_{\mathrm{EVEN}}(x) \rightarrow P_{\mathrm{EVEN}}(x \cdot y))$
  - If $x \leq 7$ and $9 \leq y$ then $x \leq y$.
- Properties of graph networks $(\mathrm{Cities}, \mathrm{DirectRail}, \mathrm{sheffield}, \mathrm{paris}, \mathrm{brussels})$
  - $\exists x (DR(s, x) \wedge DR(x, p))$
  - If $DR(x, b)$ then $DR(b, x)$ (due to the axiom $\forall x \forall y\, (DR(x, y) \rightarrow DR(y, x))$)



Examples on the use of predicate logic

- Properties of mathematical structures $(\mathbb{N}, \leq, P_{\mathrm{EVEN}}, +, \cdot)$.
  - $\forall x \forall y\, (P_{\mathrm{EVEN}}(x) \rightarrow P_{\mathrm{EVEN}}(x \cdot y))$
  - If $x \leq 7$ and $9 \leq y$ then $x \leq y$.
- Properties of graph networks $(\mathrm{Cities}, \mathrm{DirectRail}, \mathrm{sheffield}, \mathrm{paris}, \mathrm{brussels})$
  - $\exists x \big(DR(s, x) \wedge DR(x, p)\big)$
  - If $DR(x, b)$ then $DR(b, x)$ (due to the axiom $\forall x \forall y\, \big(DR(x, y) \rightarrow DR(y, x)\big)$)
- Queries on databases:
  - SQL: SELECT Student, Course, Grade FROM GradeList
    WHERE Course = 'COM2107' and Grade >= 70
  - FO: $\{(x_1, x_2, x_3) \mid \exists \vec{y}\, (\mathrm{GradeList}(x_1, x_2, x_3, \vec{y}) \wedge x_2 = COM2107 \wedge x_3 \geq 70)\}$

Predicate Logic: Natural Deduction

once again we start with foundational view on natural deduction

we use sentence as synonym for proposition

formulas are possibly incomplete sentences e.g. "x is an even number."

truth/falsity of formulas may depend on binding variables by quantifiers or substituting expressions e.g. "*Socrates*" or "5" for them

we call expression that can be substituted for variables in formulas terms

Variables, terms, and atomic formulae

Variables x are placeholders for individual objects (e.g., 7, $\mathrm{Socrates}$)
(c.f. propositional variables p are placeholders for 0 and 1)

Terms are (complex) pointers for individual objects
e.g., x, $x + y$, $\mathrm{mother\_of}(x)$

Atomic formulae describe relationships between terms
e.g. $\mathrm{ancestor}\Big(\mathrm{mother\_of}(\mathrm{mother\_of}(x)), x\Big)$, $x \leq y$, and $7 \leq x$
(cf. atomic formulae of propositional logic are p, $\top$ and $\bot$)

Complex formulae are built from atomic formulae by using connectives $\wedge, \vee, \rightarrow, \neg$ and quantifiers $\exists, \forall$.

Variables, terms, and atomic formulae

Variables x are placeholders for individual objects (e.g., 7, $\mathrm{Socrates}$)
(c.f. propositional variables p are placeholders for 0 and 1)

Terms are (complex) pointers for individual objects
e.g., x, $x + y$, $\mathrm{mother\_of}(x)$

Atomic formulae describe relationships between terms
e.g. $\mathrm{ancestor}\Big(\mathrm{mother\_of}(\mathrm{mother\_of}(x)), x\Big)$, $x \leq y$, and $7 \leq x$
(cf. atomic formulae of propositional logic are p, $\top$ and $\bot$)

Complex formulae are built from atomic formulae by using connectives $\wedge, \vee, \rightarrow, \neg$
and quantifiers $\exists, \forall$.

Interpretation will give a meaning to variables, term constructors, and relationships
(cf. In propositional logic an assignment interprets propositional variables)

Notation

we write

- $\varphi(x)$ for formula φ that is parametric in variable x
- $r, s, t \ldots$ for terms
- $\varphi[t/x]$ for result of substituting t for x in φ
- $\varphi(t)$ to indicate that the term t occurs in φ

example:

- substituting term 5 for variable x in formula $x \leq 7$ yields sentence $5 \leq 7$
- $(x \leq 7)[5/x]$ indicates variable/term relationship in this substitution

we'll study formal syntax later

Parametric Judgments

we need a new kind of judgment for formulas of predicate logic

a parametric judgment

$$J \textit{ (is true) for an arbitrary } x$$

is judgment that depends on variable x

evidence for it is a deduction of J that is parametric in x

it is therefore uniform for any term we can substitute for x
and hence for all possible instances

Parametric Judgments

evidence for a parametric judgment

$$\varphi(x) \ \textit{(is true) for an arbitrary } x$$

is proof template for $\varphi(x)$ that can be instantiated to
a proof of $\varphi(x)[t/x]$, for any term t

multiple parameters must be kept distinct across parametric derivations

we write $a, b, c, \ldots$ to distinguish parameters from ordinary variables

Proofs for universal statements

The most common mathematical statement:

$$\text{Every } A \text{ is a } B$$

The sum of two odd numbers is an even number.

The complement of any finite language is a regular language.

How are statements of this kind proven?

Proofs for universal statements

The most common mathematical statement:

Every A is a B

The sum of two odd numbers is an even number.

The complement of any finite language is a regular language.

How are statements of this kind proven?

- Pick an arbitrary object satisfying A and show that it satisfies B!

Universal Quantification

if φ is formula and x variable, then $\forall x.\ \varphi$ is formula

we often write $\forall x.\ \varphi(x)$ to indicate that φ depends on x

we read $\forall x.\ \varphi$ as "*Every x satisfies φ*" or "*Every x is a φ*"

we can write $\forall x.\ (\varphi(x) \to \psi(x))$ for
"*Every x satisfies $\varphi(x) \to \psi(x)$*" or "*Every $\varphi(x)$ is a $\psi(x)$*"

Inference Rules

inference rules for universal quantification are

$$\dfrac{\begin{array}{c}\overline{}\ a\\ \vdots\\ \varphi[a/x]\end{array}}{\forall x.\ \varphi}\ \forall I \qquad\qquad \dfrac{\forall x.\ \varphi}{\varphi[t/x]}\ \forall E$$

where t is term and a fresh parameter

evidence for $\forall x.\ \varphi$ is evidence for parametric judgment $\varphi[a/x]$ (parametric proof of $\varphi[a/x]$)

a must be discharged in $(\forall I)$ once this proof is done

Inference Rules

$(\forall I)$ is standard proof strategy in mathematics

examples:

- prove that for every natural number n, either n or $n+1$ is even.
- prove that if $x \in X$ and $X \subseteq Y \subseteq Z$, then $x \in Z$

typical approach:

- fix arbitrary object a, and show the property for a.

Universal Quantification

example: $\forall x.\ (P(x) \to Q(x)), \forall x.\ P(x) \vdash \forall x.\ Q(x)$

1. How to start the proof?

see lecture notes for additional examples

Universal Quantification

example: $\forall x.\ (P(x) \to Q(x)), \forall x.\ P(x) \vdash \forall x.\ Q(x)$

1.	$\forall x.\ (P(x) \to Q(x))$	hyp
2.	$\forall x.\ P(x)$	hyp
3.		

see lecture notes for additional examples

Universal Quantification

example: $\forall x.\ (P(x) \to Q(x)), \forall x.\ P(x) \vdash \forall x.\ Q(x)$

1.	$\forall x.\ (P(x) \to Q(x))$	hyp
2.	$\forall x.\ P(x)$	hyp
3.	a	
4.	subproof here	
5.	$Q(a)$	
6.	$\forall x.\ Q(x)$	$\forall I,$

see lecture notes for additional examples

Universal Quantification

example: $\forall x.\ (P(x) \rightarrow Q(x)), \forall x.\ P(x) \vdash \forall x.\ Q(x)$

1.	$\forall x.\ (P(x) \rightarrow Q(x))$	hyp
2.	$\forall x.\ P(x)$	hyp
3.	a	
4.	$P(a) \rightarrow Q(a)$	$\forall E, 1$
5.	$P(a)$	$\forall E, 2$
6.	$Q(a)$	$\rightarrow E, 5, 4$
7.	$\forall x.\ Q(x)$	$\forall I$, 3–6

proof wouldn't work if we applied $(\forall E)$ before $(\forall I)$

see lecture notes for additional examples

Universal quantifier and implication

$$\dfrac{\begin{array}{c}\overline{}\; a \\ \vdots \\ \varphi[a/x]\end{array}}{\forall x.\ \varphi}\ \forall I \qquad\qquad \dfrac{\begin{array}{c}[\varphi] \\ \vdots \\ \psi\end{array}}{\varphi \to \psi} \to I$$

a is a fresh parameter.

Existential Quantification

if φ is formula and x variable, then $\exists x.\ \varphi$ is formula

we read $\exists x.\ \varphi$ as "Some x satisfies φ" or "There exists an x such that $\varphi(x)$"

Inference Rules

inference rules for existential quantification are

$$\frac{\varphi[t/x]}{\exists x.\ \varphi}\ \exists I \qquad\qquad \frac{\exists x.\ \varphi \qquad \begin{array}{c}[\varphi[a/x]\ ]\\ \vdots\\ \psi\end{array}}{\psi}\ \exists E$$

where t is term and a fresh parameter

evidence for $\exists x.\ \varphi$ is evidence for judgment $\varphi[t/x]$ for some witness t

when using $\exists x.\ \varphi$ as hypothesis, we need not know a particular witness

Existential quantifier and disjunction

$$\dfrac{\exists x.\ \varphi \qquad \begin{matrix}[\varphi[a/x]] \\ \vdots \\ \psi\end{matrix}}{\psi}\ \exists E \qquad\qquad \dfrac{\varphi \vee \psi \qquad \begin{matrix}[\varphi] \\ \vdots \\ \chi\end{matrix} \qquad \begin{matrix}[\psi] \\ \vdots \\ \chi\end{matrix}}{\chi}\ \vee E$$

a is a fresh parameter.

Existential quantifier is like an infinite disjunction!

Example

example: $\forall x.\ (P(x) \to Q(x)), \exists x.\ P(x) \vdash \exists x.\ Q(x)$

1.	$\forall x.\ (P(x) \to Q(x)))$	hyp
2.	$\exists x.\ P(x)$	hyp
3.	a	
4.	$P(a)$	hyp
5.	$P(a) \to Q(a)$	$\forall E, 1$
6.	$Q(a)$	$\to E, 4, 5$
7.	$\exists x.\ Q(x)$	$\exists I, 6$
8.	$\exists x.\ Q(x)$	$\exists E, 2, 3–7$

once again it's important to start applying $(\exists E)$ before $(\exists I)$ and $(\forall E)$

Equality

equality matters

it has special inference rules, too

$$\frac{}{t = t} =I \qquad\qquad \frac{t_1 = t_2 \qquad \varphi[t_1/x]}{\varphi[t_2/x]} =E$$

where t, t_1, t_2 are terms

evidence for $s = t$ is proof that s equals t

Examples

example: equality is symmetric: $t_1 = t_2 \vdash t_2 = t_1$

1.	$t_1 = t_2$	hyp
2.	$t_1 = t_1$	$=I$
3.	$(x = t_1)[t_1/x]$	rewriting 2
4.	$(x = t_1)[t_2/x]$	$=E, 1, 3$
5.	$t_2 = t_1$	rewriting 4

Examples

example: equality is transitive: $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$

1.	$t_1 = t_2$	hyp
2.	$t_2 = t_3$	hyp
3.	$(t_1 = x)[t_2/x]$	rewriting 1
4.	$(t_1 = x)[t_3/x]$	$=E, 2, 3$
5.	$t_1 = t_3$	rewriting 4

Derived Rules

the following derived rules for equality are helpful

$$\frac{t_1 = t_2}{t_2 = t_1}\,\text{sym} \qquad \frac{t_1 = t_2 \qquad t_2 = t_3}{t_1 = t_3}\,\text{trans}$$

$$\frac{t_1 = t_2}{s[t_1/x] = s[t_2/x]}\,\text{subst}$$

see lecture notes for additional equivalences/proofs for predicate logic

Recap: Functions and relations

- A set $A = \{a_1, \ldots, a_n, \ldots\}$ is a collection of objects $a_1, a_2, \ldots$.
- A^n is the set of all n-tuples $(b_1, \ldots, b_n)$, where each $b_i \in A$.
- An n-ary function f over A maps elements of A^n to elements of A.
  - Addition $(2, 5) \mapsto 7$ is a binary function.
  - We can write $f_+(a, b)$, but also the infix $a + b$ is used.
  - $\mathrm{Mother\_of}$ is a unary fuction.
  - $\mathrm{Mother\_of(Bob)}$ equals Alice, if Alice is the mother of Bob.
- An n-ary relation R over A is a subset of A^n.

Recap: Functions and relations

- A set $A = \{a_1, \dots, a_n, \dots\}$ is a collection of objects $a_1, a_2, \dots$.
- A^n is the set of all n-tuples $(b_1, \dots, b_n)$, where each $b_i \in A$.
- An n-ary function f over A maps elements of A^n to elements of A.
- An n-ary relation R over A is a subset of A^n.
  - Friend_of is a binary relation.
  - $\text{Friend\_of}(\text{Alice}, \text{Bob})$ denotes that Alice is a friend of Bob.
  - Size comparison relation between numbers is a binary relation.
  - We can write $R_{\leq}(a, b)$, but also the infix $a \leq b$ is used.
  - The betweenness $\beta(a, b, c)$ relation of geometry is a ternary relation.

Recap: Functions and relations

- A set $A = \{a_1, \ldots, a_n, \ldots\}$ is a collection of objects $a_1, a_2, \ldots$.
- A^n is the set of all n-tuples $(b_1, \ldots, b_n)$, where each $b_i \in A$.
- An n-ary function f over A maps elements of A^n to elements of A.
- An n-ary relation R over A is a subset of A^n.
- n-ary function and relation symbols are placeholders for n-ary functions and relations.

Alphabet of Predicate Logic

alphabet for predicate logic consists of

1. (countably infinite) set $\mathcal{V}$ of logical variables $x, y, z, x_1, x_2, \ldots$
2. set $\mathcal{F} = \{f_1, f_2, \ldots\}$ of function symbols of fixed arity
3. set $\mathcal{P} = \{P_1, P_2, \ldots\}$ of predicate/relation symbols of fixed arity
4. predicate symbol $=$
5. connectives $\bot$, $\top$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\forall$, $\exists$
6. auxiliary symbols (,)

Signature (often Σ) is a collection of function symbols and relation symbols.

for more information and notation for alphabets see lecture notes

Alphabet of Predicate Logic

examples:

1. signature of arithmetic: $\Sigma_A = \{+, \cdot, 0, 1\}$
2. signature of boolean algebras: $\Sigma_{BA} = \{\sqcap, \sqcup, -, 0, 1\}$
3. signature of graphs: $\Sigma_G = \{E\}$

first two signatures are algebraic (only function symbols), third one is relational (only relation symbols)

Terms

set $\mathcal{T}_\Sigma$ of Σ-terms over signature Σ is defined by grammar

$$t ::= x \mid f(t_1, \ldots, t_n)$$

x is variable, f function symbol of some arity n and $t_1, \ldots, t_n$ are Σ-terms.

In particular

1. Every variable x and a constant symbol c is a term.
2. Application $f(t_1, \ldots, t_n)$ to terms $t_1, \ldots, t_n$ is a term.
3. There are no other terms.

terms without variables are ground terms

Terms

example: arithmetic

extend Σ_A with constant symbols $2, 3, 4, \ldots$ for each element of $\mathbb{N}$ and let $x, y \in \mathcal{V}$ be variables, then

$$3 + 5 \cdot 7 \qquad\qquad x + 5 \cdot y$$

are terms of arithmetic

Terms

example: boolean algebra

extend Σ_{BA} with constant symbols a and b, then

$$a \sqcap -a \qquad a \qquad a \sqcup (b \sqcap -a)$$

are terms of boolean algebra

Terms

example:

consider signature $\{MotherOf, Alice, Betty\}$, then

$$MotherOf(Alice) \qquad \text{and} \qquad Betty$$

are terms

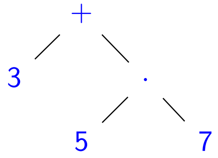
terms don't evaluate to true or false, but to other values e.g. numbers

variables are simple pointers to values, while terms are complex pointers to values

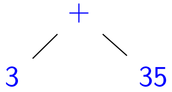
$MotherOf(Alice)$ could, e.g., evaluate to $Betty$

Terms as Trees

term $3 + 5 \cdot 7$ corresponds to (abstract syntax) tree



evaluation in $\mathbb{N}$ yields



and then 38

Formulas

Recall:

- signature Σ determines the available function and relation symbols.
- Σ-terms t_i are constructed using variables and function symbols from Σ

The set of predicate logic formulae of signature Σ are generated via the grammar:

$$\begin{aligned}\Phi ::= \bot \mid \top \mid t_1 = t_2 \mid P(t_1, \ldots, t_n) \mid \\ (\neg\Phi) \mid (\Phi \wedge \Phi) \mid (\Phi \vee \Phi) \mid (\Phi \to \Phi) \mid (\forall x.\ \Phi) \mid (\exists x.\ \Phi),\end{aligned}$$

where $t_1, \ldots t_n$ are Σ-terms in $\mathcal{T}_\Sigma$ and P is a relation symbol in Σ of arity n.

- $\bot$, $\top$, $t_1 = t_2$ and $P(t_1, \ldots, t_n)$ are atomic formulas
- all others are composite

Formulas

example:

$$\forall P.\ (P(0) \wedge \forall k.\ P(k) \to P(k+1)) \to \forall n.\ P(n))$$

operator precedences help minimising the number of brackets
you can read about them in the lecture notes

Formulas

example:

$$\forall P.\ (P(0) \land \forall k.\ P(k) \to P(k+1)) \to \forall n.\ P(n))$$

is not a formula of predicate logic — it quantifies over a predicate

operator precedences help minimising the number of brackets
you can read about them in the lecture notes

Formulas

examples:

$$\mathit{Loves}(\mathit{Alice}, \mathit{Bob}) \qquad \mathit{Loves}(x, \mathit{Bob}) \qquad \mathit{Loves}(\mathit{Alice}, \mathrm{MotherOf}(x))$$

Formulas

examples:

1. $\exists x.\ \mathit{Loves}(x, \mathit{Bob})$ (somebody loves Bob)
2. $\exists x.\ \mathit{Loves}(\mathit{Alice}, x)$ (Alice loves someone)
3. $\neg\exists x.\ \mathit{Loves}(x, \mathit{Bob})$ (nobody loves Bob)
4. $\forall x.\ \mathit{Loves}(x, \mathit{Alice})$ (everybody loves Alice)
5. $\neg\forall x.\ \mathit{Loves}(x, \mathit{Bob})$ (not everybody loves Bob)
6. $\forall x \exists y.\ \mathit{Loves}(x, y)$ (everybody loves somebody)
7. $\exists x \forall y.\ \mathit{Loves}(x, y)$ (somebody loves everybody)

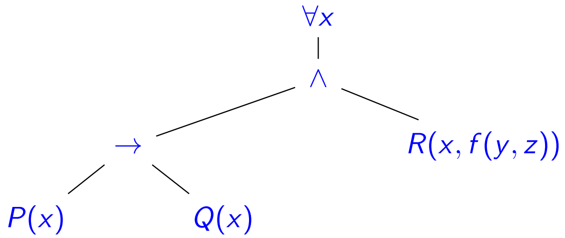
Formulas

example (arithmetic formulas):

$$3 + 5 = 8 \qquad 7 + 2 \leq 3 \qquad x + 4 \cdot y = 21 \qquad \exists x.\ 2 \cdot x = 7$$

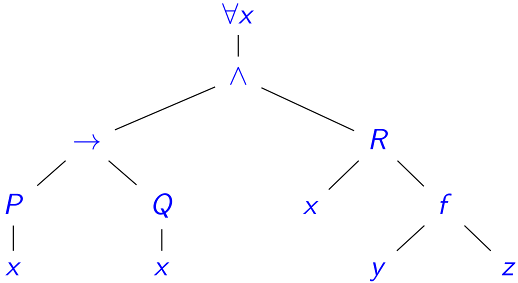
Formulas as Trees

$\forall x. \Big((P(x) \to Q(x)) \wedge R(x, f(y,z)) \Big)$ has abstract syntax tree



Formulas as Trees

expanding its term structure yields



Function vs Predicate Symbols

"Every son of my father is my brother." translates as

$$\forall x \forall y.\ (F(x, m) \wedge S(y, x) \rightarrow B(y, m))$$
$$\forall x.\ (S(x, f(m)) \rightarrow B(x, m))$$

- both formulas are correct
- but the second one is more concise

What did you learn?

- Propositional logic doesn't suffice for expressing relationships between objects.
- We familiarised with predicate logic and learned new rules of natural deduction.
- In the end, we got a bit more formal exposition to basics of predicate logic.
- Next time: add some precision to the proof rules and cover semantics.