

3. Predicate Logic

- So far:
 - Propositional logic
 - Semantics (Truth Tables)
 - Proof (Rules)
 - Properties (sat, soundness, correctness, decidable)
- Today:
 - Introduction to Predicate Logic
- The following slides are based on the slides of J. Bowen (<http://www.cs.ucl.ac.uk/staff/J.Bowen/GS03/>)

Overview

- **The need for extra structure**
 - variables, quantifiers, terms, equality, function symbols
- **Proof**
 - well formedness
 - natural deduction
- **Semantics**
 - models
- **Properties**
 - soundness
 - completeness
 - decidability
 - expressiveness

Proposition are not enough

- Samantha is Charlie's sister
 - need to distinguish properties from the things to which they apply
- All sisters are female
 - need to express “all”
- Some people don't have sisters
 - need to express “some”
- No-one is their own sister
 - need notion of equality
- Everyone's mother is also their sisters' mother
 - useful to have functions like “mother”

Elements of predicate logic

- **Variables**
 - x, y, \dots
- **Predicates**
 - $female(x)$ x is female
 - $sister(x, y)$ x is the sister of y
 - ... any number of arguments
 - $x = y$
- **Quantifiers**
 - $\forall x$ “for all $x \dots$ ”
 - $\exists x$ “for some $x \dots$ ” or “there exists an x such that ...”
- **Logical connectives**
 - $\neg \wedge \vee \Rightarrow$
 - same as propositional logic

Extra-logical terms

- In “pure” logic, the only terms used are variables
- In any real application, predicate logic is used with some other language that is relevant to the application domain
- This language may contain for example sets, functions, relations or arithmetic
- For simplicity, we’ll start by using just one new kind of term: a function symbol
 - *Charlie* constant: function of arity 0
 - *Mother(x)* the mother of *x*
 - *Wedding(x,y)* date of the wedding of *x* and *y*
 - ... any finite number of arguments allowed

Relational calculus – for DB or Modeling

Formalising natural language

- **Samantha is Charlie’s sister**
 - *sister(Samantha, Charlie)*
- **All sisters are female**
 - $(\forall x (\forall y \text{ sister}(x, y) \Rightarrow \text{female}(x)))$
- **Some people don’t have sisters**
 - $(\exists y \neg (\exists x \text{ sister}(x, y)))$
- **No-one is their own sister**
 - $(\forall x (\forall y \text{ sister}(x, y) \Rightarrow \neg x=y))$
- **Everyone’s mother is also their sister’s mother**
 - $(\forall x (\forall y \text{ sister}(x, y) \Rightarrow \text{mother}(x) = \text{mother}(y)))$

Disambiguating natural language

- **Everyone loves a winner**
 - $(\forall x (\forall y \text{ winner } (y) \Rightarrow \text{loves } (x, y)))$
 - or
 - $(\forall x (\exists y \text{ winner } (y) \wedge \text{loves } (x, y)))$
 - or
 - $(\exists y \text{ winner } (y) \wedge (\forall x \text{ loves } (x, y)))$
 - ?

Overview

- **The need for extra structure**
 - variables, quantifiers, terms, equality, function symbols
- **Proof**
 - well formedness
 - natural deduction
- **Semantics**
 - models
- **Properties**
 - soundness
 - completeness
 - decidability
 - expressiveness

Well formed formulae

$\varphi ::= P(t_1, t_2, \dots, t_n) \mid$

$(\neg \varphi) \mid$

$(\varphi \wedge \varphi) \mid$

$(\varphi \vee \varphi) \mid$

$(\varphi \Rightarrow \varphi) \mid$

$(\forall x \varphi) \mid$

$(\exists x \varphi)$

$t ::= x \mid c \mid f(t_1, t_2, \dots, t_n)$

- Quantifiers and \neg bind most tightly

Free and bound variables

- Variables are **bound** if they are within the scope of a quantifier, otherwise they are **free**
 - $(\forall y \text{ sister}(x, y) \Rightarrow \text{female}(x))$
 - y is bound, x is free
- The same variable may be bound and free in different occurrences
 - $(\forall x (\forall y \text{ sister}(x, y) \Rightarrow \text{female}(x)) \wedge \text{loves}(x, y))$
- Terms may be substituted for free variables
- The substituted term must **not** contain any variables that are bound at that point
 - $(\forall y \text{ sister}(\text{mother}(z), y) \Rightarrow \text{female}(\text{mother}(z)))$
 - **Not** $(\forall y \text{ sister}(\text{mother}(y), y) \Rightarrow \text{female}(\text{mother}(y)))$

Rules for reasoning

- All the rules for propositional formulae carry over
- In addition there are rules for
 - equality =
 - universal quantification \forall
 - generalisation of conjunction \wedge
 - existential quantification \exists
 - generalisation of disjunction \vee
- These give rise to important equivalences which are often used in practice

Rules for equality

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

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

$$t_1 = t_2 \vdash t_2 = t_1$$

$$\begin{array}{ll} 1 & t_1 = t_2 \quad \text{premise} \\ 2 & t_1 = t_1 \quad =i \\ 3 & t_2 = t_1 \quad =e \ 1,2 \end{array}$$

Where φ is $x=t_1$

$$t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$$

$$\begin{array}{ll} 1 & t_2 = t_3 \quad \text{premise} \\ 2 & t_1 = t_2 \quad \text{premise} \\ 3 & t_1 = t_3 \quad =e \ 1,2 \end{array}$$

Where φ is $t_1=x$

Rules for universal quantification

$$\frac{\phi [x_0/x] \text{ where } x_0 \text{ is a fresh variable}}{(\forall x \phi)} \quad \forall i$$

$$\frac{(\forall x \phi)}{\phi [t/x] \text{ where } t}$$

To understand this, think of the following analogy. If you want to prove to someone that you can, say, split a tennis ball in your hand by squashing it, you might say 'OK, give me a tennis ball and I'll split it.' So we give you one and you do it. But how can we be sure that you could split *any* tennis ball in this way? Of course, we can't give you *all of them*, so how could we be sure that you could split *any one*? Well, we assume that the one you did split was an arbitrary, or 'random,' one, i.e. that it wasn't special in any way – like a ball which you may have 'prepared' beforehand; and that is enough to convince us that you could split *any* tennis ball. Our rule says that if you can prove ϕ about an x_0 that isn't special in any way, then you could prove it for any x whatsoever.

1	$P(t)$	premise
2	$\forall x (P(x) \rightarrow \neg Q(x))$	premise
3	$P(t) \rightarrow \neg Q(t)$	$\forall x e 2$
4	$\neg Q(t)$	$\rightarrow e 3, 1$

Rules for existential quantification

$$\frac{\phi [t/x] \text{ where } t \text{ free for } x \text{ in } \phi}{(\exists x \phi)} \quad \exists i$$

$$\frac{(\exists x \phi) \quad \phi [x_0/x] \vdash \chi}{\chi} \quad \exists e$$

Proving the validity of the sequent $\forall x (P(x) \rightarrow Q(x)), \exists x P(x) \vdash \exists x Q(x)$ is more complicated:

1	$\forall x (P(x) \rightarrow Q(x))$	premise
2	$\exists x P(x)$	premise
3	$x_0 \quad P(x_0)$	assumption
4	$P(x_0) \rightarrow Q(x_0)$	$\forall x e 1$
5	$Q(x_0)$	$\rightarrow e 4, 3$
6	$\exists x Q(x)$	$\exists x i 5$
7	$\exists x Q(x)$	$\exists x e 2, 3-6$

There are many useful equivalences

Example: generalisation of De Morgan's laws

$$\neg \forall x \phi \dashv\vdash \exists x \neg \phi$$

$$\neg \exists x \phi \dashv\vdash \forall x \neg \phi$$

Sample proof: $\exists x \neg \phi \vdash \neg \forall x \phi$

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

Overview

- The need for extra structure
 - variables, quantifiers, terms, equality, function symbols
- Proof
 - well formedness
 - natural deduction
- Semantics
 - models
- Properties
 - soundness
 - completeness
 - decidability
 - expressiveness

Models

- In propositional logic, truth depends only on the truth of the (uninterpreted) propositions
- In predicate logic, we need to consider what the variables represent
- So every formula is interpreted relative to some *Universe of Discourse* (UoD)
 - “All men are mortal; Socrates is a man; therefore Socrates is mortal”
 - UoD is perhaps all living things
 - “All states with successors are live; all reachable states have successors; therefore all reachable states are live”
 - UoD is states of some system

What is a model?

- **A non-empty, possibly infinite, set A**
 - living things, states, transactions, aircraft, ...
- **For each predicate symbol P with arity n , a set P^M of n -tuples of A**
 - the set of tuples of values for which P is true
- **For each function symbol f with arity n , a function $f^M : A^n \rightarrow A$**
 - the real-world value of the function f
 - if n is 0 , this is just a constant element of A

Relational calculus – for
DB or Modeling

A model for a state transition system

- Let $P \cong \{R, Final\}$; $F \cong \{i\}$
 - R is a two-place predicate: the transition relation
 - $Final$ is a one-place predicate: the final states
 - i is a constant: the initial state

Then

- A is the set of all states
 - e.g., $\{a, b, c\}$
- i^M is the initial state
 - e.g., a
- R^M is the transition relation
 - e.g., $\{(a,a), (a,b), (a,c), (b,c), (c,c)\}$
- $Final^M$ is the set of final states
 - e.g., $\{b,c\}$

Questions

**What do the following mean?
Which are true in the model?**

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

Environments and free variables

- If a formula contains free variables, we need to know what values they have
- An environment (look-up table) for \mathcal{A} is a function $l: var \rightarrow \mathcal{A}$, where var is the set of all variable names
- An environment can be overwritten: $l[x \mapsto a]$ is the environment where x has the value a and all other variables y have the value $l(y)$

Satisfaction

- $M \models_l \varphi$ means that model M satisfies formula φ in environment l
 - If the formula φ has no free variables it is called a *sentence*, and we don't need an environment, so can just write $M \models \varphi$
- φ is *satisfiable* iff $M \models_l \varphi$ for *some* M and l
- A set Γ of formulae is *satisfiable* (*consistent*) if there is some M and l for which $M \models_l \varphi$ for all $\varphi \in \Gamma$
- φ is *valid* if $M \models_l \varphi$ for *all* M and l
- **Semantic entailment** $\varphi_1, \varphi_2, \dots, \varphi_n \models \psi$ means that for every M and l such that $M \models_l \varphi_i$ for all φ_i , $M \models_l \psi$

Definition of satisfaction

- $M \models_l P(t_1, t_2, \dots, t_n)$ iff $(a_1, a_2, \dots, a_n) \in P^M$
 - where a_i is the result of evaluating t_i in M and l
- $M \models_l \forall x \varphi$ iff $M \models_{l[x \mapsto a]} \varphi$ holds for all $a \in A$
- $M \models_l \exists x \varphi$ iff $M \models_{l[x \mapsto a]} \varphi$ holds for some $a \in A$
- $M \models_l \neg \varphi$ iff $M \models_l \varphi$ does not hold
- $M \models_l \varphi_1 \wedge \varphi_2$ iff $M \models_l \varphi_1$ holds and $M \models_l \varphi_2$ holds
- $M \models_l \varphi_1 \vee \varphi_2$ iff $M \models_l \varphi_1$ holds or $M \models_l \varphi_2$ holds
- $M \models_l \varphi_1 \Rightarrow \varphi_2$ iff $M \models_l \varphi_2$ holds whenever $M \models_l \varphi_1$ holds

Computability

- **It is in general infeasible to compute satisfaction**
 - it requires examination of every member of possibly infinite sets
- **It is even more infeasible to compute validity or entailment**
 - it requires examination of every possible model
- **It can be feasible to compute satisfiability**
 - it only requires a single witness
- **Satisfiability of $\neg \varphi$ is a *refutation* of φ**
- **So proof theory and model semantics play complementary roles**

Overview

- **The need for extra structure**
 - variables, quantifiers, terms , equality, function symbols
- **Proof**
 - well formedness
 - natural deduction
- **Semantics**
 - models
- **Properties**
 - soundness
 - completeness
 - decidability
 - expressiveness

Predicate logic is sound and complete

- **If** $\varphi_1, \varphi_2, \dots, \varphi_n \vdash \psi$
then $\varphi_1, \varphi_2, \dots, \varphi_n \models \psi$
- **If** $\varphi_1, \varphi_2, \dots, \varphi_n \models \psi$
then $\varphi_1, \varphi_2, \dots, \varphi_n \vdash \psi$
- **Proofs: too difficult for this course**
- **Consequence**
 - We can use proofs interchangeably with model checking to investigate the validity of formulae

Predicate logic is **not** decidable

- **There is no uniform mechanical procedure for determining the validity of a formula φ**
- **Proof**
 - show that some other problem that is known to be insoluble can be expressed in predicate logic
 - Post's correspondence problem
 - halting problem
- **Consequence**
 - analysis is hard
 - there is a difficult trade-off between expressiveness and tractability

Undecidability is bad –but not catastrophic

- **It does not mean that we can never prove *anything* mechanically**
 - but we can never prove *everything* mechanically
- **It does not mean that we cannot establish the correctness of a proof**
 - if we have a proof, we can check it – but we have no uniform method of finding it in the first place
- **It does not mean that model checking is useless**
 - if we can find a refutation, our conjecture is certainly false
 - if we have a finite model, we can (in principle) check any property of that model

How expressive is predicate logic?

- **Not completely**
 - does not include the whole of arithmetic
 - cannot express closure of relations (reachability in graphs)
- **Proof**
 - see textbook
- **Consequences**
 - In practice, we never use pure predicate logic. Always have some extra-logical theory that defines the set of terms
 - Set theory (Z, VDM, B,...)
 - Relations (Alloy)
 - Arithmetic (e.g., Presburger – consistent, complete, decidable)
 - ...
 - For some purposes, need higher-order logics (used in theorem provers like HOL, Isabelle and PVS)

Summary

- **First order predicate logic is the foundation of most reasoning in software engineering**
- **It is *sound and complete***
 - We can use proof (which most easily shows positive results) interchangeably with model checking (which most easily shows negative results)
- **It is *undecidable***
 - So we always have to make compromises and trade-offs between being expressive and having full tool support
- **It normally depends on having domain-specific extra-logical models**

Conclusion

- So far
 - Introduction
 - Propositional logic
 - Predicate logic
- Next week
 - Alloy

Reading and Exercises

- Read Chapter 2 of Huth and Ryan apart from section 2.7 (especially 2.1–2.4; 2.5–2.6 optional)
- Do selected (starred) exercises with online solutions, depending on your previous experience. E.g.:
 - 2.1, 1(a), 3(a); 2.2 1(a) ii & iii, 3(a), 4
 - 2.3 3 (a),(b), 7(a), 9(a); 2.4 1, 5, 8, 11(b), 12(b)
 - If you know the correct answers do less
 - If you have difficulty do more starred questions