Models
Sequent Calculus
and
Tableau Proofs
for first order logic

Models

- To give semantics to formulas in first order logic we need a model.
- A model for $L(C,F,P)$ is a mathematical structure $M = <D,I>$ where
  - D is a **non empty** set called the domain of M
  - I is a mapping, called an interpretation, that associates
    - $c^I \in D$ for every $c \in C$
    - $f^I \in D^n \rightarrow D$ for every n-ary function symbol $f \in F$
    - $p^I \subseteq D^n$ for every n-ary predicate symbol $p \in P$

Meanings

- We will use a model to give meanings to sentences (closed formula)
- We will need an assignment to give meanings to variables (sometimes called a valuation)
- An assignment A for a model M =<D,I> is a mapping from the set of variables to the set D
- The image of a variable v under a valuation A we denote v^A
  - $v^I = v^A$

Exercise

- Given a model and an assignment design a function mapping quantifier free formula to their meanings.

- Next time we will think about quantifers and meanings

Exercise

- Work out the details of a logic language, and a model for it, for the integers modulo 3 under addition. Start with a few constants and function symbols (at least + and -)
- Add a few predicates
- Pick a formula you expect to be true, show that it is true in your model.

Definitions

- Let A be a formula with no free variables
- Let I be an interpretation
- We say
  - I satisfies a formula A if $|=_I$ A holds
  - A set of formula S is valid if every interpretation of S satisfies every formula in S
  - A set of formula is satisfiable (or consistent) if there is some interpretation of S that satisifes every formula in S
  - A set of formual is unsatisfiable (or inconsistent) if it is not satisfiable (i.e. every interpretation falsifies some formula of S)
  - A model M =<D,I> of a set S is an interpretation I that satisifies every formula of S

Sequent calculus

- A sequent A, B, C => D, E, F
  - provided A B and C are true
  - We can show at least one of D E F are true
- Similar to natural deduction but makes the assumptions explicit
- A sequent is true if it is true in all valuations
- The basic step for proving a sequent is for the same term to be on both sides
  - A B => C B

$$\textit{basic sequent}:\ A, \Gamma \Rightarrow A, \Delta$$

Negation rules:

$$\frac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta}\ (\neg l) \qquad \frac{A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A}\ (\neg r)$$

Conjunction rules:

$$\frac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\ (\wedge l) \qquad \frac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}\ (\wedge r)$$

Disjunction rules:

$$\frac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}\ (\vee l) \qquad \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B}\ (\vee r)$$

Implication rules:

$$\frac{\Gamma \Rightarrow \Delta, A \quad B, \Gamma \Rightarrow \Delta}{A \rightarrow B, \Gamma \Rightarrow \Delta}\ (\rightarrow l) \qquad \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \rightarrow B}\ (\rightarrow r)$$

Rules for FO logic

- Recall some properties of the sequent calculus
  - Weakening (throw things away on the left or add things on the right)
  - Exchange (duplicate things)
- While we don’t usually need these rules, in FO logic they are necessary to deal with quantifies
- We need new rules for quantifiers

Here are the sequent rules for $\forall$:

$$\frac{A[t/x], \Gamma \Rightarrow \Delta}{\forall x\ A, \Gamma \Rightarrow \Delta}\ (\forall l) \qquad \frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, \forall x\ A}\ (\forall r)$$

Notes

1. The rule $\forall r$ only holds if (the now free) variable x is not free in Γ or Δ

2. The rule $\forall l$ lets one create many instances of $\forall$x A

Here are the sequent rules for $\exists$:

$$\frac{A, \Gamma \Rightarrow \Delta}{\exists x\, A, \Gamma \Rightarrow \Delta}\ (\exists l) \qquad \frac{\Gamma \Rightarrow \Delta, A[t/x]}{\Gamma \Rightarrow \Delta, \exists x\, A}\ (\exists r)$$

Notes

1. The rule $\exists l$ only holds if (the now free) variable x is not free in Γ or Δ
2. The rule $\exists r$ lets one create many instances of $\forall$x A
3. One may always rename a bound variable if one needs to.

Implementing the Sequent Calculus

- We will implement the sequent calculus in Haskell
- We will make it an interactive program
- The user will choose rules to transform one sequent into an equivalent one
- We will use monads to deal with possible mistakes by the user
- If a rule doesn't apply the computation will fail in the monad.
- By using Monad Plus we can also deal with more than one choice.

Discriminating formulas in FO logic

- As in predicate logic we can discriminate formulas into certain sets (e.g Alpha, Beta, Lit)