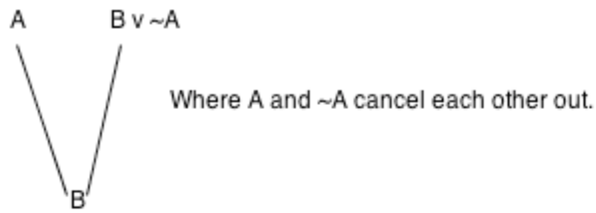
**PL-RESOLUTION AND FIRST-ORDER LOGIC**

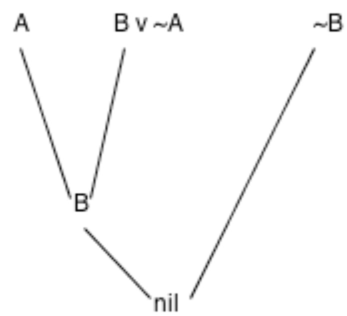
In this lecture and the next, we'll review **first-order logic (FOL)**. But first we'll take an initial look at Robinson's **resolution method** of inference.

To prove **B** from **A** and **A → B**, we can use modus ponens.

But there is another way, in which we first rewrite the if-then wff as **B v ~A**. (Question: what is meant by "rewrite as"?) Then we proceed as indicated in this diagram:




Or here is another variation of the above: we add **~B** as another "axiom" and then proceed:




Here ~B is introduced to allow a proof of B by contradiction, aka an indirect or **refutation** proof.

The conclusion *nil* means that the assumption **~B** must be rejected, hence **B** has been proven to follow from **A** and **B v ~A**.

This is a simple example of the **resolution-refutation** method introduced by Robinson in 1963. We have shown an example in PL (but it has an extension to FOL as well). *Robinson showed that it gives the same results (the same things can be proven) as in regular PL with MP and the Lukasiewicz tautologies.* Notice that *no* tautologies are needed at all! And the only inference rule is that of canceling terms such as **A** and **~A**. There is some extra work needed in preparing the wffs, by rewriting some of them in equivalent form, for instance to eliminate **→à**in terms of **v** and **~**.

Definition: A wff is in **conjunctive normal form** (CNF) if it has the form of a conjunction of disjunctions of literals. (A conjunction consists of **&**-separated expressions, and a disjunction of **v**-separated expressions. A literal is either a sentential variable/letter or its negation.)

Thus the following are in conjunctive normal form:

A, A v ~B, A & (A v ~B), B&A, (BvA)&(~BvC)&(Cv~A). [Note these are not bold schema-letters; they are actual PL letters.]

And these are not: A → B, ~(AvB).

A disjunction of literals is called a **clause**. Thus a wff in CNF is a conjunction of clauses.

It turns out that *every* wff is equivalent to a wff in CNF. [If we want to give examples of arbitrary CNF wffs, we can use letter-schema-variables such as **J, K, L** which are understood to refer to individual (but unknown) PL letters (not arbitrary wffs as in the case of **A, B, C**, etc). So this example is in CNF form even though it is not an actual wff since we don't know which letters are really involved: **L & (~L v K) & (L v ~K).** All we know is that **L, K** each refers to *some* letter such as A or B or P or Q.]

The general form of resolution for propositional logic uses two clauses, one including the letter **L** and the other **~L**, and the result is to infer a new clause that is a disjunction of all the *other* literals in both clauses. Thus from **Lv~KvJ** and **~Kv~LvM** one may infer **~KvJv~KvM** which is equivalent to **~KvJvM**. (We say two wffs are *equivalent* if they have the same truth-table values in all interpretations. This answers the above question about rewriting a wff. This is *not* the same as saying they are *equal*. Two different wffs are never equal.)

We will come back to resolution refutation after we have reviewed FOL.

**Review of FOL:**

1. **Language:**
   - the same connectives as for PL, parentheses, and a comma
   - variable symbols x,y,z, etc
   - constant symbols a,b,c, etc
   - predicate symbols P, Q, R, etc
   - function symbols f,g,h, etc
   - quantifiers ForAll and Exists
   - wffs are formed according to the familiar rules, where each predicate symbol takes the appropriate number of arguments

   Examples: P(c), Exists y (Q(y,b)), ForAll x (P(x) à ~Q(x,b))

2. **Inference rules**
   These usually include a generalized version of MP: from ForAll x [Px à Qx] and Pc, infer Qc; and also some additional rules such as: from P(a) infer Exists x P(x), and from ForAll x P(x) infer P(a).

3. **Axioms**
   Here there are special sets of wffs that play a role like that of the Lukasiewicz axioms in PL. But when we come to FOL resolution we will not need these (nor will we need the MP rule).

4. **Semantics**
   Here we have a great deal beyond what PL has to offer. An interpretation for FOL actually assigns entities to the constant symbols, meanings to the predicate symbols, functions to the function symbols. We don't simply *say* a wff is true or false, we give it so much meaning that it really is true or false in a given interpretation. To do this, we first specify the *domain* D, consisting of all the objects that we wish the constants and variables to refer to. Then we specify actual members of D to be what the constant symbols refer to, actual functions for the function symbols, and actual relations on D for the predicate symbols.

Example: Let interpretation I1 be as follows: D = N (the set of natural numbers: 0,1,2,…), let c stand for 0, and d for 1, f for the successor (add 1) function, = for equality, and G for greater than.

Then the wff

$G(f(c),c)\ \&\ =(f(c),d))$

is true in I1. We'd more normally write it as

$(f(c) > c)\ \&\ (f(c)=d)$

and it means

$(0+1 > 0)\ \&\ (0+1 = 1)$,

which is indeed true.

But now consider interpretation I2: D = {all living people }, c means Judy, d means Tom (Judy's father), f means father-of, = still means equality, and G means older than. Now the very same wff

$G(f(c),c)\ \&\ =(f(c),d))$

means in interpretation I2 that Judy's father is older than Judy, and Judy's father is Tom, which is still true.

Finally, if we let I3 be the interpretation that is just like I2 except that now the meaning of G is younger-than, then the same wff now means Judy's father is younger than Judy, and Judy's father is Tom, which is false (the conjunction is false, even though part of it is true).

As before, we use KB |- **W** to indicate provability of **W** from KB, and KB |= **W** to indicate truth of **W** in all interpretations in which the wffs in KB are true. We call such an interpretation a *model* of the axiom set KB. We also refer to any set of axioms that we wish to consider a theory.

In FOL, we still have a **soundness and completeness theorem** as in PL: Given any axiom set KB, then for every wff **W**,

KB |= **W** if and only iff KB |- **W**,

where the proof methods are suitable ones to make this work (for instance, refutation resolution). Another way to state soundness and completeness is that a wff is provable in a theory (KB |- **W**) if and only if it is true in all models of the theory (KB |= **W**).

We are almost ready to write down the KB for the hungry monkey, but not quite. The monkey's world changes as it performs actions. So we need a way to indicate different states of the world. (This is true in any planning domain, or even any search problem.) Thus we need our domain D to be one that includes not only the monkey, the bananas, and the box, but also *situations* the world can be in. For instance, locations can change; so we need constants to indicate locations as well as entities to indicate *when* the box is at one location and when at another, etc.

So we introduce a *situation* variable, s. This is not an addition to FOL. It just means s will be one of the variable letters, and *we* keep in mind what we intend by it. We also will need a start-state (or start-situation), which will be a constant, say s_0. Finally, we'll need function symbols for the possible actions: push, climb, grasp.

There are other details we will need as well. But we will leave that for another lecture. Next we will turn to the resolution refutation method in FOL.