CS304 – CS310

Mathematical and *Logic*al Aspects of Computing

Lectures 11 and 12: Clause Form and Resolution

8^{th} and 11^{th} of October 2013

Today's topics

1. Why bother with normal forms?
2. Recall: Conjunctive Normal Form (CNF)
3. Clause form
4. Properties of clausal forms
5. The Resolution Rule
6. Resolution
7. Exercises

Why bother with normal forms?

To date we've studied both Disjunctive Normal Form, and Conjunctive Normal Form. These have several applications, including

- They can be used to prove that the set $\{\wedge, \vee, \neg\}$ is functionally complete.
- Automated reasoning software uses these forms to check if two expressions are equivalent.
- We can develop a new proof method based on CNF (and Clause Form) called *resolution*.

Recall: Conjunctive Normal Form (CNF)

Recall: a proposition is in *normal form* if it is expressed using only negation ($\neg$), disjunction ($\vee$) and conjunction ($\wedge$).
It is in Conjunctive Normal Form (**CNF**) if it is it is written as the conjunction of disjunction of literals.
We construct the CNF as follows

1. Rewrite all operators using just AND, OR and NOT.
2. Use de Morgan's laws so that any negation applies to atomic propositions, not compound propositions.
3. Eliminate double negations.
4. Use the distributive laws:

$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$$

$$(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C)$$

Recall: Conjunctive Normal Form (CNF)

Example 1: Write $(a \to b) \wedge (\neg a \to c)$ in CNF.

Example 2: Write $(a \downarrow b) \vee (\neg a \to c)$ in CNF.

Recall that with the semantic tableau, we would consider the set of propositions

$$\{A_1, A_2, \ldots, A_n\}$$

as a compound proposition

$$A_1 \wedge A_2 \wedge \cdots \wedge A_n$$

If we think of the second line as a single compound proposition, then the first line expresses this proposition is as an *implicit conjunction of proposition*. This is *Clause form*.

Writing a proposition in *Conjunctive Normal Form (CNF)* naturally leads to a representation in *Clausal Form* (the two notions are so closely related that some books use CNF to mean "Clausal Normal Form"). We now define

Unit clause: a clause that consists of a single literal;

A clause: a set of literals that is considered to be an implicit disjunction.

Clausal form: the implicit conjunction of a set of literals.

Example:

I find the notation can be quite confusing at first, and so often write clauses with an explicit disjunction. However, it is very helpful later, and let's us define resolution correctly.
In particular, we can use the operations of set theory easily.
Example:

[See Ben-Ari, Section 4.1]
If U and V are sets of clauses, we write $U \approx V$ if U is satisfiable if and only if V is satisfiable.
Note that this does *not* mean that U and V are logically equivalent.
Example:

Let us denote the compliment of a literal a as a^c.

Prop 1. Suppose a is a literal in some clause in U, but a^c is not. If we obtain V by deleting every clause containing a from U, then $U \approx V$.

Example:

Prop. 2: Suppose U contains a unit clause $\{a\}$. If V is formed by deleting every clause containing a from U, and deleting $\neg a$ from every remaining clause in U, then $U \approx V$.

Example:

Prop. 3: Suppose $C \in U$ is a clause containing both a and a^c. If V is formed by deleting C from U, then $U \approx V$. (That is, deleting all valid clauses from U does not change the satisfiablity of U)

Example:

Prop. 4: Suppose C_1 and C_2 are clauses in U and that C_1 is a *subclause* of C_2. If V is formed by deleting C_2 from U, then $U \approx V$. (That is, one can delete the larger clause).

Example:

Prop. 5: An empty clause, $\square$, is **not** satisfiable.

Prop. 6: If V is obtained from U by replacing every instance of a particular literal a by a^c, then $U \approx V$.

We can now restart formally the "resolution rule"

The resolution rule

Suppose U contains clauses C_1 and C_2 where $a \in C_1$ and $a^c \in C_2$. We call C_1 and C_2 "clashing clauses". The *resolvent* of C_1 and C_2 is the clause

$$\mathrm{Res}(C_1, C_2) = C_1/\{a\} \cup C_2/\{a^c\}.$$

Prop. 7 $\{C_1, C_2\} \approx \mathrm{Res}(C_1, C_2)$.

Example:

Resolution Procedure

Let S_0 be as set of clauses. For $i = 0, 1, 2, \ldots...$

(1) Choose a pair of clashing clauses, C_1 and C_2 that have not been chosen before;

(2) Compute $C = \mathrm{Res}(C_1, C_2)$.

(3) Let $S_{i+1} = S_i \cup C$.
STOP IF:

- $C = \square$: and S_0 is not satisfiable.
- All pairs of clashing clauses have been resolved.

The key idea in **resolution** is

$$\text{If } a \vee b = T \text{ and } b = F, \text{ then } a = T.$$

This can be used to verify that

$$\{a \vee b, \neg b\} \models a.$$

It is useful to view this as a tree:

A little more generally: $\{a \vee b, a \vee \neg b\} \models a$.
And so: $\{a \vee b, a \vee \neg b, \neg a\}$ is not consistent.

Now we'll use this idea to determine if the following set of propositions consistent as a collection:

$$\{p \vee q, \neg p \vee s, \neg(q \vee s)\}$$

Example: Apply resolution to determine if:

Given $\{\neg p \vee (q \wedge s), q \rightarrow (p \wedge \neg s\}$ can we conclude $\neg p$?

Q1. Use resolution to show that the following arguments are valid:

1. $\{\neg a \vee (b \wedge c),\ b \rightarrow (a \wedge \neg c)\} \models \neg a$
2. $\{a \vee b, \quad a \rightarrow c, \quad b \rightarrow c\} \models c.$

Q2. To construct the DNF of compound proposition, we used a simple (if slow) algorithm based on writing out the logic table for the propositions. Devise a similar algorithm (i.e., based just on the logic table, and not by using the distributive and de Morgan's Laws) for CNF. (*Hint:* think about the rows for which the proposition evaluates as false).