COMP2411 Lecture 8: Resolution, Horn formulas

Reading: Huth and Ryan, Section 1.5.3

Note: some of the material in this lecture is not covered in the book

In general, conversion to CNF leads to an exponential blowup in the size of the formula.

However, many problems can be represented directly as a succinct CNF formula.

For such problems, it makes sense to develop specialised techniques for dealing with CNF formulas.

Example: Graph (Map) Colouring

Let V be a set of vertices and $E \subseteq V \times V$ be a set of edges. The graph $G = (V, E)$ is *three-colourable* if there exists a function $f : V \to \{R, B, G\}$ such that for all edges $(x, y) \in E$, we have $f(x) \neq f(y)$. (i.e adjacent nodes get different colours.)

We can translate this problem to a set of clauses $S(G)$ containing, for each vertex $v \in V$:

- $R_v \vee B_v \vee G_v$ (v is red, blue or green)
- $\neg R_v \vee \neg G_v$ (v is not both red and green)
- $\neg R_v \vee \neg B_v$
- $\neg G_v \vee \neg B_v$

and for each edge $(v, w) \in E$:

- $\neg R_v \vee \neg R_w$ (v and w are not both red)
- $\neg B_v \vee \neg B_w$
- $\neg G_v \vee \neg G_w$

G is three colourable iff $S(G)$ is satisfiable.

Resolution Theorem Proving

The following rule of inference which can be applied to clauses is called *resolution*

$$\frac{p \vee \phi_1 \quad \neg p \vee \phi_2}{\phi_1 \vee \phi_2} Res$$

Observe that if $p \vee \phi_1$ and $\neg p \vee \phi_2$ are both clauses then so is $\phi_1 \vee \phi_2$.

Soundness of resolution is established by the following argument:

$1:$	$p \vee \phi_1$	Premise
$2:$	$\neg p \vee \phi_2$	Premise
$3:$	p	Asmtn
$4:$	$\neg p$	Asmtn
$5:$	$\bot$	$3, 4, \ \neg e$
$6:$	$\phi_1 \vee \phi_2$	$5, \ \bot e$
$4:$	ϕ_2	Asmtn
$5:$	$\phi_1 \vee \phi_2$	$4, \ \vee i_2$
$7:$	$\phi_1 \vee \phi_2$	$2, 4-5, 4'-5', \ \vee e$
$8:$	ϕ_1	Asmtn
$9:$	$\phi_1 \vee \phi_2$	$8, \vee i_1$
$10:$	$\phi_1 \vee \phi_2$	$1, 3-7, 8-9, \ \vee e$

Remark: If we view $\perp$ as a clause containing zero literals, then the $\neg e$ rule becomes a special case of resolution:

$$\frac{p \quad \neg p}{\perp}$$

Using resolution to prove validity

Lemma: $\phi_1, \ldots, \phi_n \models \psi$ if and only if $\phi_1 \wedge \ldots \wedge \phi_n \wedge \neg\psi$ is NOT satisfiable.

Lemma: ϕ is not satisfiable if $\phi \vdash \perp$

This leads to the following approach to showing $\phi_1, \ldots, \phi_n \models \psi$:

1. convert $\phi_1 \wedge \ldots \wedge \phi_n \wedge \neg\psi$ into a set of clauses (i.e. CNF form)
2. apply resolution to derive $\perp$.

Example: $P \longrightarrow Q,\ Q \longrightarrow R \models P \longrightarrow R$

Proof: converting to CNF, we get 4 clauses:

- $P \longrightarrow Q = \neg P \vee Q$
- $Q \longrightarrow R = \neg Q \vee R$
- $\neg(P \longrightarrow R) = P \wedge \neg R$

This gives the resolution derivation:

$1:$	$\neg P \vee Q$	Premise
$2:$	$\neg Q \vee R$	Premise
$3:$	P	Premise
$4:$	$\neg R$	Premise
$5:$	Q	$1, 3, Res$
$6:$	R	$2, 5, Res$
$7:$	$\bot$	$4, 6, Res$

Another example: we reconsider the alarm example from lecture 4:

Premises:

1. $S \longrightarrow (A \vee F)$
2. $(A \wedge D) \longrightarrow S$
3. $F \longrightarrow S$

Conclusion: $D \longrightarrow (S \longleftrightarrow (A \vee F))$

Converting to CNF:

1. $S \longrightarrow (A \vee F) = \neg S \vee A \vee F$
2. $(A \wedge D) \longrightarrow S = \neg A \vee \neg D \vee S$
3. $F \longrightarrow S = \neg F \vee S$

4. $\neg(D \longrightarrow (S \longleftrightarrow (A \vee F)))$

$$
\begin{aligned}
&= \neg(D \longrightarrow ((S \longrightarrow (A \vee F)) \wedge ((A \vee F) \longrightarrow S))) \\
&= \neg(\neg D \vee ((\neg S \vee (A \vee F)) \wedge (\neg(A \vee F) \vee S))) \\
&= \neg(\neg D \vee ((\neg S \vee (A \vee F)) \wedge (\neg(A \vee F) \vee S))) \\
&= D \wedge (\neg(\neg S \vee (A \vee F)) \vee \neg(\neg(A \vee F) \vee S))) \\
&= D \wedge ((S \wedge \neg A \wedge \neg F) \vee ((A \vee F) \wedge \neg S)) \\
&= D \wedge \left(\begin{array}{l} ((S \wedge \neg A \wedge \neg F) \vee (A \vee F)) \wedge \\ ((S \wedge \neg A \wedge \neg F) \vee \neg S)) \end{array} \right) \\
&= D \wedge \quad (S \vee A \vee F) \wedge (\neg A \vee A \vee F) \wedge (\neg F \vee A \vee F) \wedge \\
&\qquad\qquad (S \vee \neg S) \wedge (\neg A \vee \neg S) \wedge (\neg F \vee \neg S)
\end{aligned}
$$

Clauses like $(\neg A \vee A \vee F)$, containing a literal and its negation, are equivalent to $\top$, and can be eliminated.

Thus the clausal premises are:

$$
\begin{aligned}
1: &\quad \neg S \vee A \vee F \\
2: &\quad \neg A \vee \neg D \vee S \\
3: &\quad \neg F \vee S \\
4: &\quad D \\
5: &\quad S \vee A \vee F \\
6: &\quad \neg A \vee \neg S \\
7: &\quad \neg F \vee \neg S
\end{aligned}
$$

The following resolution derivation shows these clauses are unsatisfiable.

$$
\begin{array}{rll}
8: & \neg A \vee S & 2, 4, \ Res \\
9: & S \vee F & 8, 5, \ Res \\
10: & S & 3, 9, \ Res \\
11: & \neg A & 6, 10, \ Res \\
12: & \neg S \vee F & 1, 11, \ Res \\
13: & F & 9, 12, \ Res \\
14: & \neg F & 7, 10, \ Res \\
15: & \bot & 13, 14, \ Res
\end{array}
$$

Note: using a clause more than once is permitted. Cf steps 10 & 13, or steps 11 & 14.

Completeness of resolution

If $c_1, \ldots, c_n$ and c are clauses, write $c_1, \ldots, c_n \vdash_{Res} c$ if c can be derived from $c_1, \ldots, c_n$ by using *only* the resolution rule.

Theorem: Resolution is sound and complete for proving unsatisfiability of clauses, i.e.,

1. If $c_1, \ldots, c_n \vdash_{Res} \bot$ then the set of clauses $c_1, \ldots, c_n$ is not satisfiable.
2. If $c_1, \ldots, c_n$ is not satisfiable then $c_1, \ldots, c_n \vdash_{Res} \bot$.

Observations on the Complexity of Resolution

Suppose C is a set of clauses in the prpositional letters $\{p_1, \ldots, p_n\}$.

Note $\alpha \vee \beta \equiv \beta \vee \alpha$. Thus we can represent the non-equivalent clauses as subsets of $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$.

There are 2^{2n} of these. Thus resolution requires at most 2^{2n} steps to produce $\perp$ (if at all).

Fact: an exponential number of steps is sometimes necessary.

Question: Does there exist an efficient (non-exponential) satisfiablity test for CNF formulas?

Answer: Nobody Knows. Given that many smart people have tried to find one and failed, it is generally believed there is none.

However, this does not prevent the existence of special classes of clauses which do have efficient satisfability tests ...

Horn Clauses

A *Horn clause* is a clause containing at most one non-negated atom.

Examples:

- $\bot$
- p
- $\neg p$
- $\neg q \vee p \vee \neg r \vee \neg s$

Not a Horn clause:

- $p \vee q$
- $p \vee \neg q \vee r \vee \neg s$

An alternative way of writing Horn clauses, used in the text:

- $p_1 \wedge \ldots \wedge p_n \longrightarrow p$ for $\neg p_1 \vee \ldots \vee \neg p_n \vee p$
- $\top \longrightarrow p$ for p
- $p \longrightarrow \bot$ for $\neg p$
- $p_1 \wedge \ldots \wedge p_n \longrightarrow \bot$ for $\neg p_1 \vee \ldots \vee \neg p_n$

(These variants can be seen to be equivalent using implication and De Morgan's laws.)