CS262 Logic and Verification

Lecture 7: Resolution

Propositional resolution

Tableau proofs $\leftrightarrow$ DNF

Resolution proofs $\leftrightarrow$ CNF

Tableau proofs presented as trees: Each branch is a conjunction, tree is the disjunction of its branches

Trees are not convenient for resolution: We use the standard representation $\langle [\ldots], [\ldots], \ldots \rangle$, one disjunction on each line.

In each step, select a disjunction and a non-literal formula N in it.

If $N = \neg\top$, then append a new disjunction where N is replaced by $\bot$.
If $N = \neg\bot$, then append a new disjunction where N is replaced by $\top$.
If $N = \neg\neg Z$, then append a new disjunction where N is replaced by Z.
If N is an α-formula, then append two new disjunctions, one in which N is replaced by α_1, and one in which it is replaced by α_2 (**α-expansion**).
If N is a β-formula, then append a new disjunction where N is replaced by β_1, β_2 (**β-expansion**).

Resolution expansion rules:

$$\frac{\neg\top}{\bot} \qquad \frac{\neg\bot}{\top} \qquad \frac{\neg\neg Z}{Z} \qquad \frac{\beta}{\begin{array}{c}\beta_1\\ \beta_2\end{array}} \qquad \frac{\alpha}{\alpha_1 \mid \alpha_2}$$

Example

1. $[p \downarrow (q \wedge r)]$
2. $[\neg(q \vee (p \rightarrow q))]$

Strict resolution

A sequence of resolution expansion applications is **strict**, if every disjunction has at most one resolution expansion rule applied to it.

Intuitively: no formula reuse

As with tableau, strict version suits itself for implementation: Remove formula from list after applying an expansion rule to it.

With this, resolution expansion becomes identical to conjunctive normal form expansion

Resolution rule

For resolution, we need yet another rule of a different nature, the **resolution rule**.

Suppose D_1 and D_2 are two disjunctions, with X occuring in D_1 and $\neg X$ in D_2. Let D be the result of the following:

- delete all occurences of X from D_1
- delete all occurences of $\neg X$ from D_2
- combine the resulting disjunctions

Special case: If a disjunction contains $\perp$, delete all occurences of $\perp$, and call the resulting disjunction the **trivial resolvent**.

D is the result of **resolving D_1 and D_2 on X**. D is the **resolvent** of D_1 and D_2, and X is the formula being **resolved on**. If X is atomic, then this is an **atomic** application of the resolution rule.

Examples:

1. $[p, q \to r]$	1. $[a \wedge b]$	1. $[p, q \uparrow r, \bot]$
2. $[a \wedge b, \neg p]$	2. $[\neg(a \wedge b)]$	?
?	?	

Justification of the resolution rule:

$\langle [X, Y], [\neg X, Z] \rangle \equiv \langle [X, Y], [\neg X, Z], [Y, Z] \rangle$

Resolution proof

A resolution expansion is **closed**, if it contains the empty clause [].

A **resolution proof** for X is a closed resolution expansion for $\neg X$.

We write $\vdash_r X$ if X has a resolution proof.

Example

Resolution proof for $((p \wedge q) \vee (r \to s)) \to ((p \vee (r \to s)) \wedge (q \vee (r \to s)))$:

1. $[\neg(((p \wedge q) \vee (r \to s)) \to ((p \vee (r \to s)) \wedge (q \vee (r \to s))))]$
2. $[(p \wedge q) \vee (r \to s)]$
3. $[\neg((p \vee (r \to s)) \wedge (q \vee (r \to s)))]$
4. $[p \wedge q, r \to s]$
5. $[\neg(p \vee (r \to s)), \neg(q \vee (r \to s))]$
6. $[p, r \to s]$
7. $[q, r \to s]$
8. $[\neg p, \neg(q \vee (r \to s))]$
9. $[\neg(r \to s), \neg(q \vee (r \to s))]$
10. $[\neg p, \neg q]$
11. $[\neg p, \neg(r \to s)]$
12. $[\neg(r \to s), \neg q]$
13. $[\neg(r \to s), \neg(r \to s)]$
14. $[r \to s, \neg p]$
15. $[r \to s, r \to s]$
16. $[]$

Example

Proof steps:

- α-expansion on 1. creates 2.+3.
- β-expansion on 2. creates 4.
- β-expansion on 3. creates 5.
- α-expansion on 4. creates 6.+7.
- α-expansion on 5. creates 8.+9.
- α-expansion on 8. creates 10.+11.
- α-expansion on 9. creates 12.+13.
- resolving on q in 7. and 10. creates 14.
- resolving on p in 6. and 14. creates 15.
- resolving on $r \rightarrow s$ in 13. and 15. creates 16.

Resolution method extends to first order logic (quantifiers)

Resolution can be generalized to establish propositional consequences $S \models X$, not just tautologies $\models X$.

Resolution rules are non-deterministic: We have freedom in applying them. Different rules may produce proofs of different length.

Theorem: The resolution proof system is **sound**, i.e., if X has a resolution proof, then X is a tautology.

Theorem: The resolution proof system is **complete**, i.e., if X is a tautology, then the resolution system will terminate with a proof for it, even if if all resolution rule applications are atomic or trivial, and come after all resolution expansion steps.

Equivalently, $\vdash_r X$ if and only if $\models X$.

First theorem follows from the correctness proof of our CNF expansion algorithm given before. Also argued that resolution rule produces a semantically equivalent formula.

Proof of second theorem not given here; requires more advanced tools.

Recall the definition of propositional consequence $S \models X$.

***S*-introduction rule for tableau:** Any formula $Y \in S$ can be added to the end of any tableau branch. We write $S \vdash_t X$ if there is a closed tableau for $\neg X$ allowing the S-introduction rule for tableau.

***S*-introduction rule for resolution:** For any formula $Y \in S$, the line $[Y]$ can be added as a line to a resolution expansion. We write $S \vdash_r X$ if there is a closed resolution expansion for $\neg X$, allowing the S-introduction rule for resolution.

Theorem (Strong soundness and completeness): For any set S of propositional formulas and any formula X, we have $S \models X$ if and only if $S \vdash_t X$ if and only if $S \vdash_r X$.

Example

Prove $\{p \to q, q \to r\} \models \neg(\neg r \wedge p)$ via tableau and resolution.