CS 511, Fall 2018, Handout 31

First-Order Resolution

Assaf Kfoury

November 18, 2018

REVIEW and PRELIMINARIES

- This handout continues Handout 11, which introduced resolution for propositional logic .
- This handout also depends on Handout 29, which is a presentation of unification , limited to the kind we use in **first-order resolution**.

REVIEW and PRELIMINARIES

- First-order resolution starts from a Skolemized sentence whose matrix is in CNF.
- So, let φ be such a Skolemized first-order sentence:

  $$\varphi \triangleq \forall x_1 \cdots \forall x_k (C_1 \wedge C_2 \wedge \cdots C_n)$$

  where each C_i is a disjunction of literals (atomic and negated atomic WFF's).
- Standard practice is to write each **disjunct** (or **clause**) C_i as a set of literals, *i.e.*, if $C_i \triangleq (L_1 \vee L_2 \vee \cdots L_p)$, we may write instead $C_i' \triangleq \{L_1, L_2, \ldots, L_p\}$.
- The **clausal form** of φ is the set of clauses $\{C_1', C_2', \ldots, C_n'\}$ where C_i' is the set representation of C_i.

  The clausal form of φ is therefore a set of sets of literals.[1]

[1] As written, each C_i' may be a multiset, not a set, because some literals in C_i may be duplicates. One simplifying advantage of the set representation is to disallow duplicated literals as well as duplicated clauses. C_i' and $\{C_1', C_2', \ldots, C_n'\}$ have to be adjusted accordingly (left to you).

REVIEW and PRELIMINARIES

- We can assume that each of the clauses in $\{C_1, C_2, \ldots, C_n\}$, or in its set representation $\{C'_1, C'_2, \ldots, C'_n\}$, is universally quantified over all its variables – because "$\forall$" distributes over "$\wedge$".
- Because each clause is implicitly universally closed, we can assume that for all distinct clauses C_i and C_j, it holds that $\mathsf{FV}(C_i) \cap \mathsf{FV}(C_j) = \varnothing$ (why?).

  This is useful when we unify one literal C_i and one literal in C_j.

FIRST-ORDER RESOLUTION

- We need two rules for carrying out first-order resolution, both using unification: one for **resolution** proper and one for what is called **factoring**.
- The ***resolution rule*** has two clauses, D_1 and D_2, as ***antecedents*** with:
  - $P(\vec{s}) \triangleq P(s_1, \ldots, s_k) \in D_1$ and $\neg P(\vec{t}) \triangleq \neg P(t_1, \ldots, t_k) \in D_2$, *i.e.*, clauses D_1 and D_2 contain conflicting literals $P(\vec{s})$ and $\neg P(\vec{t})$, modulo a unification of $\vec{s}$ and $\vec{t}$, where P is a k-ary predicate symbol,
  - we may assume $\mathsf{FV}(\vec{s}) \cap \mathsf{FV}(\vec{t}) = \varnothing$ for a simpler unification,
  - a most general unifier of $P(\vec{s})$ and $P(\vec{t})$ exists, $\sigma \triangleq \mathsf{MGU}\big(P(\vec{s}), P(\vec{t})\big)$,

  and one conclusion (or ***resolvent*** clause) D:
  - $D \triangleq \Big(\sigma\,(D_1) - \Big\{\sigma\,\big(P(\vec{s})\big)\Big\}\Big) \cup \Big(\sigma\,(D_2) - \Big\{\sigma\,\big(\neg P(\vec{t})\big)\Big\}\Big)$
- More succinctly, the resolution rule is written:

$$\frac{D_1 \qquad\qquad D_2}{\Big(\sigma\,(D_1) - \Big\{\sigma\,\big(P(\vec{s})\big)\Big\}\Big) \cup \Big(\sigma\,(D_2) - \Big\{\sigma\,\big(\neg P(\vec{t})\big)\Big\}\Big)}$$

where $P(\vec{s}) \in D_1$ and $\neg P(\vec{t}) \in D_2$ and $\sigma \triangleq \mathsf{MGU}\big(P(\vec{s}), P(\vec{t})\big)$

EXAMPLE OF HOW THE RESOLUTION RULE IS USED

$$\frac{\{\ \underline{P(x,y)},\ P(y,x),\ P(x,a)\ \} \qquad \{\ \underline{\neg P(f(z),f(z))},\ Q(z)\ \}}{\{\ P(f(z),a),\ Q(z)\ \}} \qquad (\sigma)$$

where $\sigma \triangleq \{x \mapsto f(z), y \mapsto f(z)\}$ and the members of the resolution pair are underlined.

FIRST-ORDER RESOLUTION

- The ***factoring rule*** has one clause, D_1, as an ***antecedent*** with:
  - $P(\vec{s}) \triangleq P(s_1, \ldots, s_k) \in D_1$ and $P(\vec{t}) \triangleq P(t_1, \ldots, t_k) \in D_1$,
    i.e., clause D_1 contains two non-conflicting literals $P(\vec{s})$ and $P(\vec{t})$, modulo a unification of $\vec{s}$ and $\vec{t}$, where P is a k-ary predicate symbol,
  - a most general unifier of $P(\vec{s})$ and $P(\vec{t})$ exists, $\sigma \triangleq \text{MGU}\big(P(\vec{s}), P(\vec{t})\big)$,

  and one conclusion (or ***resolvent*** clause) D:

  - $D \triangleq \sigma(D_1)$

  With D_1 in set representation, $\sigma\big(P(\vec{s})\big)$ and $\sigma\big(P(\vec{t})\big)$ are the same literal in $\sigma(D_1)$.

- More succinctly, the factoring rule is written:[2]

$$\boxed{\begin{array}{l} \dfrac{D_1}{\sigma(D_1)} \\[2ex] \text{where } P(\vec{s}) \in D_1 \text{ and } P(\vec{t}) \in D_1 \text{ and } \sigma \triangleq \text{MGU}\big(P(\vec{s}), P(\vec{t})\big) \end{array}}$$

[2] There is no need for a factoring rule in *propositional resolution*. Do you see why?

EXAMPLE OF HOW THE FACTORING RULE IS USED

$$\frac{\{\ \underline{P(x,y)},\ P(y,x),\ \underline{P(x,a)}\ \}}{\{\ P(x,a),\ P(a,x)\ \}} \qquad (\sigma)$$

where $\sigma \triangleq \{y \mapsto a\}$ and the members of the unification pair are underlined.

SOUNDNESS and COMPLETENESS

Theorem

Let $\Psi_0 = \{C_1, C_2, \ldots, C_n\}$ be the clausal form of a Skolemized first-order sentence φ. We then have that:

1. *Applying the* **resolution rule** *and* **factoring rule** *repeatedly in any order, we obtain a sequence of clausal forms that is bound to terminate:*

   $$\Psi_0 \quad \Psi_1 \quad \Psi_2 \quad \cdots \quad \Psi_p \qquad \text{for some } p \geqslant 1$$

2. *If $\perp \in \Psi_p$ then φ is unsatisfiable (soundness).*
3. *If φ is unsatisfiable then $\perp \in \Psi_p$ (completeness).*

Exercises

In the exercises below, keep in mind that First-Order Resolution starts from a Skolemized sentence whose matrix is in CNF. In general, this requires that the initial input set of first-order sentences must be transformed accordingly in a pre-processing phase.

1. **Exercise**. Redo Exercise 1 on the last slide of Handout 27, now using First-Order Resolution.
2. **Exercise**. Redo Exercise 2 on the last slide of Handout 27, now using First-Order Resolution.

(THIS PAGE INTENTIONALLY LEFT BLANK)