Logic(s) for computer science - Week 14
Rezoluion for FOL

The resolution for FOLis a way of proving that a formula from FOLin CSNF is unsatisfiable, similar to the ground resolution studied in the previous lecture. The advantage of using the resolution (instead of ground resolution) is that we don't need to "conveniently" choose a ground substitution, but this choice is done automatically, using a method invented by Robinson in 1960 named unification; therefore is easier to be automatized (implemented in a program).

1 Unification

Definition 1.1 (Unifier)**.** *A substitution σ is an unifier for the terms t_1 and t_2 if $\sigma^\sharp(t_1) = \sigma^\sharp(t_2)$.*

Example 1.1. *Let take the terms $t_1 = f(x, h(y))$ and $t_2 = f(h(z), z')$. An unifier for t_1 and t_2 is:*

$$\sigma = \{z \mapsto a, x \mapsto h(a), z' \mapsto h(y)\} \qquad (\sigma^\sharp(t_1) = f(h(a), h(y)) = \sigma^\sharp(t_2)).$$

Another unifier of the two terms is:

$$\sigma_1 = \{x \mapsto h(z), z' \mapsto h(y)\} \qquad (\sigma_1^\sharp(t_1) = f(h(z), h(y)) = \sigma_1^\sharp(t_2))$$

Definition 1.2 (Unifiable terms)**.** *Two terms are* unifiable *if they have at least one unifier.*

Example 1.2. *The terms $t_1 = f(x, y)$ and $t_2 = h(z)$ don't have any unifier, so they are not unifiable. Why?*

Because for any substitution σ we have $\sigma^\sharp(t_1) = \sigma^\sharp(f(x, y)) = f(\sigma^\sharp(x), \sigma^\sharp(y))$ $\neq h(\sigma^\sharp(z)) = \sigma^\sharp(h(z)) = \sigma^\sharp(t_2)$.

The terms $t_1 = x$ and $t_2 = h(x)$ don't have any unifier. Why?

Let's consider that there is an unifier σ for the. Because $\sigma^\sharp(t_1) = \sigma^\sharp(t_2)$, the abstract trees associated to the terms $\sigma^\sharp(t_1)$ and $\sigma^\sharp(t_2)$ have the same number of nodes. We note with noduri(t) the number of nodes in the abstract tree associated to the term t.

We have that $noduri(\sigma^\sharp(t_2)) = noduri(\sigma^\sharp(h(x))) = 1 + noduri(\sigma^\sharp(x)) = 1 + noduri(\sigma^\sharp(t_1)) > noduri(\sigma^\sharp(t_1))$*, which is a contradiction because we should have* $noduri(\sigma^\sharp(t_2)) = noduri(\sigma^\sharp(t_1))$*. Therefore, the supposed unifier* σ *does not exist.*

Definition 1.3 (Composition of two substitutions)**.** *Let* σ_1, σ_2 *be two substitutions. The substitution* $\sigma_2 \circ \sigma_1 : \mathcal{X} \to \mathcal{T}$*, called* the composition of substitutions σ_1 and σ_2*, is defined as:*

- $(\sigma_2 \circ \sigma_1)(x) = \sigma_2^\sharp(\sigma_1(x))$*, for any* $x \in \mathcal{X}$*.*

Exercise 1.1. *Verify that the function* $\sigma_2 \circ \sigma_1$ *is indeed a substitution (that is, the set of those variables* x *with the property that* $(\sigma_2 \circ \sigma_1)(x) \neq x$ *is finite).*

Example 1.3. *Continuing the above example, let the substitutions* $\sigma = \{z \mapsto a, x \mapsto h(a), z' \mapsto h(y)\}$*,* $\sigma_1 = \{x \mapsto h(z), z' \mapsto h(y)\}$ *and respectively* $\sigma_2 = \{z \mapsto a\}$*.*

We have that $\sigma = \sigma_2 \circ \sigma_1$*.*

Exercise 1.2. *Show that indeed the substitutions* σ *and respectively* $\sigma_2 \circ \sigma_1$ *are equal (they have the same result for any variable).*

Definition 1.4 (More general substitution)**.** *A substitution* σ_1 *is more general then the substitution* σ *if* σ *can be obtained by composing the substitution* σ_1 *with another substitution* σ_2*:* $\sigma = \sigma_2 \circ \sigma_1$*.*

Example 1.4. *For example,* $\sigma_1 = \{x \mapsto h(z), z' \mapsto h(y)\}$ *is more general then* $\{z \mapsto a, x \mapsto h(a), z' \mapsto h(y)\}$*, because* $\sigma = \sigma_2 \circ \sigma_1$*, where* σ_2 *is defined in the above example.*

Definition 1.5 (The most general unifier)**.** *A substitution* σ *is* the most general unifier of the terms t_1 and t_2 *if:*

1. σ *is unifier for the terms* t_1, t_2 *and*
2. σ *is a more general substitution then any other unifier of* t_1, t_2*.*

Example 1.5. *Let consider* $t_1 = f(x, a)$ *and* $t_2 = f(y, a)$*. The unifier* $\{y \mapsto x\}$ *is more general then* $\{x \mapsto a, y \mapsto a\}$*.*

Example 1.6. *The substitution* $\sigma_1 = \{x \mapsto h(z), z' \mapsto h(y)\}$ *is more general unifier of the terms* $t_1 = f(x, h(y))$ *and* $t_2 = f(h(z), z')$*.*

Theorem 1.1 (Theorem of existence of the most general unifier)**.** *Any two unifiable terms have a most general unifier.*

Remark 1.1. *In general, the most general unifier is NOT unique.*

Example 1.7. *An unifier for the terms* $h(x)$ *and* $h(y)$ *is the substitution* $\{x \mapsto a, y \mapsto a\}$ *(but it is not the most general unifier).*

A most general unifier is $\{x \mapsto y\}$*. Another most general unifier is* $\{y \mapsto x\}$*.*

In the following, we present an algorithm to compute a most general unifier.

In order to present this algotithm, we need the generalization of the notion of unification for more then one pair of terms.

Definition 1.6 (Unification problem)**.** *An* unification problem P *is:*

- *either a set*
$$P = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$$
of n pairs of terms
- *or the special symbol*
$$P = \bot \text{ (citit bottom)}.$$

Definition 1.7 (Solution for an unification problem)**.** *A substitution σ is a* solution *of a unification problem P if:*

1. *the problem has the form $P = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$ and*
2. *σ is an unifier for t_i and t'_i, for any $i \in \{1, \ldots, n\}$.*

Definition 1.8 (The set of solutions for an unification problem)**.** *We note by $unif(P)$ the set of solutions of the unification problem P:*
$$unif(P) = \{\sigma \mid \sigma \text{ is solution for the problem } P\}.$$

Remark 1.2. *By the definition of the solution of an unification problem, if $P = \bot$, then $unif(P) = \emptyset$.*

Example 1.8. *Let $P = \{f(x, a) \doteq f(y, a)\}$. We have that $unif(P) = \{\{x \mapsto z, y \mapsto z\}, \{x \mapsto y\}, \ldots\}$.*

Definition 1.9 (The most general solution)**.** *The substitution σ is the most general solution for the unification problem $P = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$ if:*

1. *σ is solution for P: $\sigma^\sharp(t_i) = \sigma^\sharp(t'_i)$, for any $1 \leq i \leq n$;*
2. *σ is more general then any other solution for P.*

Remark 1.3. *Note that in case $unif(P) \neq \emptyset$ (the unification problem has solutions), then there is at least one most general solution for P.*

Notation 1.1. *By $mgu(P)$ we note the most general solution of the unification problem P (if the problem P has solutions).*

By $mgu(t_1, t_2)$ we note the most general unifier of the terms t_1 and t_2 (if the terms are unifiable).

Remark 1.4. *$mgu(t_1, t_2) = mgu(\{t_1 \doteq t_2\})$.*

Definition 1.10 (Solved form)**.** *An unification problem P* is in solved form *if $P = \bot$ or $P = \{x_1 \doteq t'_1, \ldots, x_n \doteq t'_n\}$ and $x_i \notin vars(t_j)$ for any $i, j \in \{1, \ldots, n\}$.*

Why is it useful the solved form of an unification problem?

Lemma 1.1. *If $P = \{x_1 \doteq t'_1, \ldots, x_n \doteq t'_n\}$ is in solved form, then $\{x_1 \mapsto t'_1, \ldots, x_n \mapsto t'_n\}$ is the most general solution for the problem P.*

The following rules can be used to bring an unification problem in solved form:

DELETE	$P \cup \{t \doteq t\} \Rightarrow P$
DECOMPOSITION	$P \cup \{f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n)\} \Rightarrow$ $P \cup \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$
ORIENTATION	$P \cup \{f(t_1, \ldots, t_n) \doteq x\} \Rightarrow P \cup \{x \doteq f(t_1, \ldots, t_n)\}$
ELIMINATION	$P \cup \{x \doteq t\} \Rightarrow \sigma^\sharp(P) \cup \{x \doteq t\}$ if $x \notin vars(t), x \in vars(P)$(where $\sigma = \{x \mapsto t\}$)
CONFLICT	$P \cup \{f(t_1, \ldots, t_n) \doteq g(t'_1, \ldots, t'_m)\} \Rightarrow \bot$
OCCURS CHECK	$P \cup \{x \doteq f(t_1, \ldots, t_n)\} \Rightarrow \bot$ dacă $x \in vars(f(t_1, \ldots, t_n))$

The transformations from above have the following properties:

Lemma 1.2 (Progress)**.** *If P is not in solved form, then there exists P' such that $P \Rightarrow P'$.*

Lemma 1.3 (Preserving solutions)**.** *If $P \Rightarrow P'$, then $unif(P) = unif(P')$.*

Lemma 1.4 (Termination)**.** *There is not an infinite sequence $P \Rightarrow P_1 \Rightarrow P_2 \Rightarrow \ldots \Rightarrow P_i \Rightarrow \ldots$.*

Corollary 1.1. *The above rules form an algorithm to compute a most general solution for an unification problem, if it exists.*

Example 1.9.

$$P = \{f(g(x_1, a), x_2) \doteq x_3, f(x_2, x_2) \doteq f(a, x_1)\} \overset{\text{Decomposition}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_2 \doteq a, x_2 \doteq x_1\} \overset{\text{Elimination}}{\Rightarrow}$$

$$\{f(g(x_1, a), a) \doteq x_3, x_2 \doteq a, a \doteq x_1\} \overset{\text{Orientation}}{\Rightarrow}$$

$$\{f(g(x_1, a), a) \doteq x_3, x_2 \doteq a, x_1 \doteq a\} \overset{\text{Elimination}}{\Rightarrow}$$

$$\{f(g(a, a), a) \doteq x_3, x_2 \doteq a, x_1 \doteq a\} \overset{\text{Orientation}}{\Rightarrow}$$

$$\{x_3 \doteq f(g(a, a), a), x_2 \doteq a, x_1 \doteq a\}.$$

Conclusion: $\{x_3 \mapsto f(g(a, a), a), x_2 \mapsto a, x_1 \mapsto a\}$ *is the most general solution of the initial problem.*

Example 1.10.

$$P = \{f(g(x_1, a), x_2) \doteq x_3, f'(x_2) \doteq f'(x_3)\} \overset{\text{Decomposition}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_2 \doteq x_3\} \overset{\text{Orientation}}{\Rightarrow}$$

$$\{x_3 \doteq f(g(x_1, a), x_2), x_2 \doteq x_3\} \overset{\text{Elimination}}{\Rightarrow}$$

Explain why we cannot apply orientation

$$\{x_3 \doteq f(g(x_1, a), x_3), x_2 \doteq x_3\} \overset{\text{Occurs check}}{\Rightarrow}$$

$$\bot.$$

Conclusion: $unif(P) = \emptyset$.

Example 1.11.

$$P = \{f(g(x_1, a), x_2) \doteq x_3, f(g(x_4, x_5)) \doteq f(x_3)\} \overset{\text{Decomposition}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, g(x_4, x_5) \doteq x_3\} \overset{\text{Orientation}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_3 \doteq g(x_4, x_5)\} \overset{\text{Elimination}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq g(x_4, x_5), x_3 \doteq g(x_4, x_5)\} \overset{\text{Conflict}}{\Rightarrow}$$

$$\bot.$$

Conclusion: $unif(P) = \emptyset$.

2 Resolution for FOL

Resolution for first order logic is a deductive system formed by the following two inference rules:

$$\text{Binary Resolution } \frac{P(t_1, \ldots, t_n) \vee C_1 \quad \neg P(t'_1, \ldots, t'_n) \vee C_2}{\sigma^{\flat}(C_1 \vee C_2)} \quad \begin{array}{l} V_1 \cap V_2 = \emptyset \\ \sigma = mgu(\{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}) \end{array}$$

where $V_1 = vars(P(t_1, \ldots, t_n) \vee C_1)$ and $V_2 = vars(\neg P(t'_1, \ldots, t'_n) \vee C_2)$.

$$\text{Positive Factorization } \frac{P(t_1, \ldots, t_n) \vee P(t'_1, \ldots, t'_n) \vee C}{\sigma^{\flat}(P(t_1, \ldots, t_n) \vee C)} \; \sigma = mgu(\{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\})$$

Remark 2.1.

- *In the case the clauses that represent the hypothesis for the rule* Binary Resolution *have common variables ($V_1 \cap V_2 \neq \emptyset$), the variables of one clause have to be renamed before applying the rule (see the below example);*
- *In the case the unification problem that appears in the resolution rule does not have solution, the rule cannot be applied.*
- *The positive factorization rule has only one hypothesis.*
- *In the case when the unification problem that appears in the factorization problem has no solution, the rule cannot be applied.*
- *The positive factorization rule is needed for the completeness property.*

Theorem 2.1 (Resolution theorem)**.** *A formula $\varphi = \forall x_1 \ldots . \forall x_n . (C_1 \wedge C_2 \wedge \ldots C_m)$, in CSNF, is unsatisfiable if and only if $\square$ scan be obtained from the clauses $C_1, \ldots, C_m$, by applying the rules* Binary Resolution *and* Positive Factorization.

Example 2.1. *Let prove that $\forall x.(P(x) \wedge (\neg P(h(x)) \vee Q(f(x))) \wedge (\neg Q(f(g(a)))))$ is unsatisfiable, by resolution for FOL:*

1. $P(x)$
2. $\neg P(h(x)) \vee Q(f(x))$
3. $\neg Q(f(g(a)))$
4. $Q(f(x))$ *binary resolution between* 1 *and* 2*:*

$$\frac{P(x') \qquad \neg P(h(x)) \vee Q(f(x))}{\sigma^{\flat}(Q(f(x)))} \; \sigma = \{x' \mapsto h(x)\} = mgu(\{x' \doteq h(x)\})$$

5. $\square$ *resolution between* 3 *and* 4*:*

$$\frac{Q(f(g(a))) \qquad Q(f(x))}{\sigma^{\flat}(\square)}\ \sigma = \{x \mapsto g(a)\} = mgu(\{f(g(a)) \doteq f(x)\})$$

Exercise 2.1. *Prove that* $\Big(\forall x.(P(x) \to Q(x))\Big) \wedge P(s) \to Q(s)$ *is valid, using the resolution for* `FOL`*.*