3.4 Resolution in Predicate Logic

Montag, 15. Mai 2017 08:30

Goal: Resolution on arbitrary clauses (with variables).

Drawback up to now: One has to instantiate variables of the clauses by ground terms in the beginning. This instantiation must be chosen such that it enables all future needed resolution steps.

Ex. 3.4.1 Clause set   Slide 15

$\{ \{p(X), \neg q(X)\}, \{\neg p(f(Y))\}, \{q(f(a))\} \}$

To allow the next resolution step between

$\{\underline{p(X)}, \neg q(X)\}$ and $\{\neg \underline{p(f(Y))}\}$

one has to <u>unify</u> $p(X)$ and $p(f(Y))$ (i.e., use an instantiation which makes them equal).

We can use the <u>unifier</u> $[X / f(Y)]$

$\{p(X), \neg q(X)\}$ $\{\neg p(f(Y))\}$ $\{q(f(a))\}$

$[X/f(Y)]$

$\{\neg q(f(Y))\}$

$[Y/a]$

$\Box$

Advantage:
If we instantiate X by a term $f(Y)$ with a variable Y, then we can decide <u>later</u> how to instantiate Y.

If we had required instantiations by <u>ground</u> terms, then we would have to instantiate X by $f(a)$ in the first resolution step, because otherwise the second res. step would not work.

Def 3.42 (Unification)

A clause $K = \{L_1, .., L_n\}$ is unifiable iff there exists a substitution σ with $\sigma(L_1) = \ldots = \sigma(L_n)$ (i.e., $|\sigma(K)| = 1$).

Such a subst. is called a unifier of K.

A unifier σ is most general unifier (mgu) of K iff for every unifier σ' of K there exists a substitution δ such that $\sigma' = \underbrace{\delta \circ \sigma}$.

Function composition: first apply σ, then apply δ.

$K = \{p(X), p(f(Y))\}$

mgu $\sigma = \{X/f(Y)\}$

other unifier $\sigma' = \{X/f(a), Y/a\}$

indeed: $\sigma' = \delta \circ \sigma$

for $\delta = \{Y/a\}$

If a clause is unifiable, then it also has a mgu.

The mgu is unique up to variable renaming.

$K = \{p(X), p(Y)\}$.

mgu $\sigma_1 = \{X/Y\}$

$\sigma_2 = \{Y/X\}$ ← equal up to variable renaming

First unification algorithm by J. Robinson (1965)

Ex 343 Illustrate Unification Algorithm

Slide 16

(a) $\{\, q(f(X,Y)),\ q(g(X,Y)) \,\}$ Clash Failure

(b) $\{\, q(X),\ q(h(X)) \,\}$ Occur Failure

X occurs in $h(X)$

(c) $\{\neg p(f(Z, g(a,Y)), h(Z)),\ \neg p(f(f(U,V), W), h(f(a,Y)))\}$

Z $\quad$ $f(U,V)$ $\quad$ $\sigma = \{Z/f(U,V)\}$

$\{\neg p(f(f(U,V), g(a,Y)), h(f(U,V))),$
$\neg p(f(f(U,V), W), h(f(a,Y)))\}$

$\sigma = \{W/g(a,Y)\} \circ \{Z/f(U,V)\}$
$= \{W/g(a,Y),\ Z/f(U,V)\}$

$\{\neg p(f(f(U,V), g(a,Y)), h(f(U,V))),$
$\neg p(f(f(U,V), g(a,Y)), h(f(a,Y)))\}$

$\sigma = \{U/a\} \circ \{W/g(a,Y), Z/f(U,V)\}$

$= \{U/a,\ W/g(a,Y),\ Z/f(a,V)\}$

$\{\neg p(f(f(a,V),g(a,Y)), h(f(a,V)),$
$\neg p(f(f(a,V),g(a,Y)), h(f(a,Y))\}$

$\sigma = \{Y/V\} \circ \{U/a, W/g(a,Y), Z/f(a,V)\}$
$= \{Y/V, U/a, W/g(a,V), Z/f(a,V)\}$

This is the mgu.

Thm 344 (Termination and Soundness of Unif. Alg)

The unification alg. terminates for every clause $K \neq \square$ and it is sound, i.e., it computes a mgu for K iff K is unifiable.

Proof: Alg. terminates because every iteration of the loop (Steps 2-6) removes one variable from $\sigma(K)$.

If alg. terminates with success, then $|\sigma(K)| = 1 \Rightarrow$ σ is a unifier of K.

Thus: if K is not unifiable $\Rightarrow$ alg. terminates with clash or occur failure ✓

It remains to show:
If K is unifiable $\Rightarrow$ alg. finds a unifier σ and σ is mgu.

Let m be the number of loop iterations that the alg. performs for clause K.

For all $0 \le i \le m$, let σ_i be the value of σ after the i-th loop iteration.

We show the following for all $0 \le i \le m$:

For every unifier σ' of K, we have $\sigma' = \sigma' \circ \sigma_i$. $(\ast)$

If $(\ast)$ holds, then the alg. cannot stop with failure.

The reason is that then $\sigma_m(K)$ would not be unifiable.

But: $|\sigma'(K)| \underset{(\ast)}{=} |(\sigma' \circ \sigma_m)(K)| = |\sigma'(\sigma_m(K))| = 1$

$\sigma_m(K)$ is unifiable.

If $(\ast)$ holds, then alg. returns a mgu:

We must have $|\sigma_m(K)| = 1$, i.e., σ_m is a unifier of K.

By $(\ast)$: for every unifier σ', we have $\sigma' = \sigma' \circ \sigma_m$

$\leadsto \sigma_m$ is mgu.

It remains to show the following for all $0 \le i \le m$:

for all unifiers σ' of K, we have: $\sigma' = \sigma' \circ \sigma_i$. $(\ast)$

We prove $(\ast)$ by induction on i:

Ind. Base: $i = 0$

$\sigma_0 = \emptyset$. Thus: $\sigma' = \sigma' \circ \underset{\text{identity}}{\emptyset}$ holds for any σ'!

Ind. Step: $i > 0$

$\sigma_i = \{X/t\} \circ \sigma_{i-1}$

By the ind. hyp: $\sigma' = \sigma' \circ \sigma_{i-1}$.

Therefore, we have:

$$\begin{aligned} & \sigma' \circ \sigma_i \\ = \; & \sigma' \circ \{X/t\} \circ \sigma_{i-1} && \text{by def. of } \sigma_i \\ = \; & \sigma' \circ \sigma_{i-1} && \text{Since } \underline{\sigma' \circ \{X/t\} = \sigma'} \\ = \; & \sigma' && \text{by the ind. hyp} \end{aligned}$$

Reason for $\sigma' \circ \{X/t\} = \sigma'$:

For $Y \neq X$, we clearly have $(\sigma' \circ \{X/t\})(Y) = \sigma'(Y)$.

For X:

$(\sigma' \circ \{X/t\})(X) = \sigma'(t) = \sigma'(X)$

σ' is unifier of $\sigma_{i-1}(K)$ and every unifier of $\sigma_{i-1}(K)$ must make X and t equal

Since $\sigma' = \sigma' \circ \sigma_{i-1}$, thus

$|\sigma'(K)| = |\sigma'(\sigma_{i-1}(K))| = 1$

$\square$

Def 3.4.5 (Resolution in Pred. Logic)

Slide 17

Let K_1, K_2 be clauses. Then R is a <u>resolvent</u> of K_1 and K_2 iff

- There exist variable renamings ν_1, ν_2 such that $\nu_1(K_1)$ and $\nu_2(K_2)$ have no

$$\boxed{\begin{array}{l} \{p(X), q(X,Y)\} \\ \{p(X'), q(X',Y')\} \end{array}}$$

common variables.

- There are $L_1, \dots, L_m \in \nu_1(K_1)$ and

  $L'_1, \dots, L'_n \in \nu_2(K_2)$ such that

  $\{ \overline{L_1}, \dots, \overline{L_m}, L'_1, \dots, L'_n \}$

  is unifiable with a mgu σ.
- $R = \sigma(\nu_1(K_1) \setminus \{L_1, \dots, L_m\} \cup$

  $\nu_2(K_2) \setminus \{L'_1, \dots, L'_n\})$

For a clause set $\mathcal{K}$, we define:

$$Res(\mathcal{K}) = \mathcal{K} \cup \{ R \mid R \text{ is resolvent of two clauses from } \mathcal{K} \}$$

$$Res^0(\mathcal{K}) = \mathcal{K}$$

$$Res^{n+1}(\mathcal{K}) = Res(Res^n(\mathcal{K})) \text{ for all } n \geq 0$$

$$Res^*(\mathcal{K}) = \bigcup_{n \geq 0} Res^n(\mathcal{K})$$

For clauses without variables, this definition is equivalent to the definition of prop. resolution.

Ex. 346

$\{ \underbrace{p(f(X))}_{L_1}, \neg q(Z), \underbrace{p(Z)}_{L_2} \}$ $\quad \nu_1 = \emptyset$

$\{ \neg p(X), r(g(X)) \}$ $\quad$ Slide 17

$\nu_2 = \{ X/U, U/X \}$

results in

$\{ \neg p(U), r(g(U)) \}$

$\{\neg q(f(X)), r(g(f(X)))\}$

$\{\underline{\neg p(U)}, r(g(U))\}$

L_1'

mgu of $\{\neg p(f(X)),$
$\neg p(Z),$
$\neg p(U)\}$ is
$\sigma = \{U/f(X), Z/f(X)\}$

However: $(p \vee q) \wedge (\neg p \vee \neg q)$

$\{\underline{p}, \underline{q}\}$ $\quad$ $\{\underline{\neg p}, \underline{\neg q}\}$ $\quad$ is not sound

$\square$

Propositional resolution is sound and complete.
Now we want to prove soundness + completeness of resolution in pred. logic:

$\mathcal{K}$ is unsatisfiable iff $\square \in Res^*(\mathcal{K})$

"$\leftarrow$": soundness

"$\rightarrow$": completeness

For soundness, we need a similar resolution lemma as in prop. logic:

<u>Lemma 3.47.</u> (Resolution Lemma in Pred. Logic)

Let $\mathcal{K}$ be a set of clauses. If $K_1, K_2 \in \mathcal{K}$ and

R is resolvent of K_1 and K_2, then
$\mathcal{K}$ and $\mathcal{K} \cup \{R\}$ are equivalent.

<u>Proof</u>: similar as for prop. logic $\boxtimes$

This suffices for soundness:

$\square \in Res^*(\mathcal{K}) = \bigcup_{n \geq 0} Res^n(\mathcal{K})$ $\leadsto$ there is an n_0 with $\square \in Res^{n_0}(\mathcal{K})$.

By the resolution lemma: $\mathcal{K}$ is equivalent to $Res(\mathcal{K})$

$\leadsto$ (induction on n) $\mathcal{K}$ is equivalent to $Res^n(\mathcal{K})$ for all $n \geq 0$

Since $Res^{n_0}(\mathcal{K})$ is unsatisfiable, $\mathcal{K}$ is unsat. as well.

Now we prove completeness.

Idea: We know that propositional resolution is complete. Let us re-use this observation by "lifting" propositional resolution proofs to resolution proofs in pred. logic.

<u>Lemma 348</u> (Lifting Lemma)

Let K_1, K_2 be two clauses, let K_1', K_2' be ground instances of K_1 and K_2. If R' is a (propositional) resolvent of K_1' and K_2', then there exists a resolvent R of K_1 and K_2 such that R' is a ground instance of R.

$$\begin{tikzcd}
K_1 \arrow[d, "\text{gr. inst.}"] & & K_2 \arrow[d, "\text{gr. inst}"] \\
K_1' \arrow[dr, no head] & & K_2' \arrow[dl, no head] \\
& R' &
\end{tikzcd}
\quad \text{implies} \quad
\begin{tikzcd}
K_1 \arrow[dr, no head] & & K_2 \arrow[dl, no head] \\
& R \arrow[d, "\text{gr. inst}"] & \\
& R' &
\end{tikzcd}$$

Ex. 3.4.9

$$\underset{K_1}{\{p(f(X)), \neg q(Z), p(Z)\}} \qquad \underset{K_2}{\{\neg p(X), r(g(X))\}}$$

$$\downarrow \text{gr. inst} \qquad\qquad \downarrow \text{gr. inst}$$

$$K_1' = \{p(f(a)), \neg q(f(a))\} \qquad K_2' = \{\neg p(f(a)), r(g(f(a)))\}$$

$$R' = \{\neg q(f(a)), r(g(f(a)))\}$$

instead:

$$\underset{K_1}{\{p(f(X)), \neg q(Z), p(Z)\}} \qquad \underset{K_2}{\{\neg p(X), r(g(X))\}}$$

$$R = \{\neg q(f(X)), r(g(f(X)))\}$$

$$\downarrow \text{gr. inst}$$

$$R' = \{\neg q(f(a)), r(g(f(a)))\}$$

Proof of the lifting lemma 3.4.8:

Let ϑ_1, ϑ_2 be variable renamings such that $\vartheta_1(K_1)$ and $\vartheta_2(K_2)$ are variable-disjoint. Then we can use the same ground subst. σ to obtain

$$K_1' = \sigma(\vartheta_1(K_1)) \quad \text{and} \quad K_2' = \sigma(\vartheta_2(K_2)).$$

Since R' is resolvent of K_1' and K_2', there is a literal L with $L \in K_1'$ and $\overline{L} \in K_2'$ such that

$$R' = (K_1' \setminus \{L\}) \cup (K_2' \setminus \{\overline{L}\}).$$

Let $L_1, \dots, L_m \in \vartheta_1(K_1)$ be all literals where

$$\sigma(L_1) = \dots = \sigma(L_m) = L.$$

Let $L_1', \dots, L_n' \in \vartheta_2(K_2)$ be all literals where

$$\sigma(L_1') = \dots = \sigma(L_n') = \overline{L}.$$

So σ is a unifier of $\{\overline{L}_1, \dots, \overline{L}_m, L_1', \dots, L_n'\}$. Since this clause is unifiable, it also has a mgu σ'.
Thus, there exists a subst. δ with $\sigma = \delta \circ \sigma'$.

K_1 and K_2 have the resolvent

$$R = \sigma'\Big(\big(\nu_1(K_1) \setminus \{L_1, \dots, L_m\}\big) \cup \big(\nu_2(K_2) \setminus \{L_1', \dots, L_n'\}\big)\Big).$$

It remains to show that R' is a ground instance of R.

$$\begin{aligned}
R' &= (K_1' \setminus \{L\}) \cup (K_2' \setminus \{\overline{L}\}) \\
&= (\sigma(\nu_1(K_1)) \setminus \{L\}) \cup (\sigma(\nu_2(K_2)) \setminus \{\overline{L}\}) \\
&= \sigma\Big(\big(\nu_1(K_1) \setminus \{L_1, \dots, L_m\}\big) \cup \big(\nu_2(K_2) \setminus \{L_1', \dots, L_n'\}\big)\Big) \\
&= \delta\Big(\sigma'\big((\nu_1(K_1) \setminus \{L_1, \dots, L_m\}) \cup (\nu_2(K_2) \setminus \{L_1', \dots, L_n'\})\big)\Big) \\
&= \delta(R)
\end{aligned}$$

$\boxtimes$

<u>Thm 3.4.10.</u> (Soundness + Completeness for Resolution in Pred. Logic)

Let $\mathcal{K}$ be a set of clauses.
Then $\mathcal{K}$ is unsatisfiable iff $\square \in Res^*(\mathcal{K})$.

<u>Proof</u>: "$\Leftarrow$" (Soundness), see above.

"$\Rightarrow$" (Completeness)

$\mathcal{K}$ unsatisfiable

$\leadsto$ Herbrand-expansion $E(\mathcal{K})$ is unsatisfiable (Thm 3.2.7) i.e.: there is a finite set of ground instances of the clauses in $\mathcal{K}$ that is unsatisfiable.

By completeness of propositional resolution (Thm 3.3.7) one can derive $\Box$ from these ground instances.

Thus, there is a sequence of ground clauses

$K_1', \dots, K_m'$ with $K_m' = \Box$,

where for all $1 \le i \le m$ we have:

- K_i' is a ground instance of a clause $K \in \mathcal{K}$ or
- K_i' is a resolvent of K_j' and K_k' for $j, k < i$

With the lifting lemma 3.4.8 we now construct a sequence of clauses $K_1, \dots, K_m$ where K_i' is a ground instance of K_i (for all $1 \le i \le m$) and all $K_i \in Res^*(\mathcal{K})$:

- if K_i' is a ground instance of some $K \in \mathcal{K}$, then we choose $K_i := K$
- if K_i' is a resolvent of K_j' and K_k':
  we have already constructed K_j, K_k such that K_j' and K_k' are ground instances of K_j, K_k

$$\begin{array}{ccc} K_j & & K_k \\ \downarrow \text{gr. in} & & \downarrow \text{gr. in} \\ K_j' & & K_k' \\ & \searrow \swarrow & \\ & K_i' & \end{array} \quad \overset{\text{lifting lemma}}{\leadsto} \quad \begin{array}{ccc} K_j & & K_k \\ & \searrow \swarrow & \\ & K_i & \\ & \downarrow \text{gr. inst.} & \\ & K_i' & \end{array}$$

$\leftarrow$ we choose K_i to be the resolvent of K_j, K_k such that K_i' is a ground inst. of K_i

Since $K_m' = \Box$ is a ground instance of K_m,
we have $K_m = \Box$.

Thus: $K_1, \dots, K_m = \Box$ is a resolution proof in
pred. logic which shows $\Box \in \mathrm{Res}^*(\mathfrak{M})$. $\blacksquare$

Now we can improve the algorithm to check
$\{\varphi_1, \dots, \varphi_k\} \models \varphi$.

1. Attempt Gilmore's Alg. — 2 Drawbacks:
   (a) How to instantiate variables?
   (b) How to check unsat. in prop. logic?

2. Attempt Ground Resolution Alg — solves (b)

3. Attempt Resolution Alg. — also solves (a) — Slide 18

Alg. is again a semi-decision procedure:
if $\{\varphi_1, \dots, \varphi_k\} \not\models \varphi$,
then the alg. could be non-terminating.

Problem: Alg computes all possible resolution steps.
To improve efficiency, it would be desirable to
restrict the possible resolution steps

without losing completeness.