Künstliche Intelligenz

Hausaufgabe 5

N. Lehmann, T. Gleißner, A. Zubarev

27.05.2015

Aufgabe 1: Unifikation

Um die Disagreement Sets und den Most General Unifier zu finden, sucht man sich immer das linkste Symbol, bei dem sich die Formeln unterscheiden und bildet daraus ein Disagreement Set S_i sowie Ersetzungsregeln σ_i. Anschließend wendet man die Ersetzungsregeln σ_i auf die Formeln S_i an, um die neuen Formeln S_{i+1} zu erhalten bis entweder die Formeln gleich sind oder man feststellt, dass Unifikation nicht möglich ist.

$$
\begin{aligned}
S_0 &= \{p(X, f(Y), Z), p(T, T, g(cat)), p(f(dog), S, g(W))\}\\
D_0 &= \{X, T, f(dog)\}\\
\sigma_0 &= \{X/f(dog), T/f(dog)\}\\
S_1 &= \{p(f(dog), f(Y), Z),\ p(f(dog), f(dog), g(cat)),\ p(f(dog), S, g(W))\}\\
D_1 &= \{f(Y), f(dog), S\}\\
\sigma_1 &= \{X/f(dog), T/f(dog), Y/dog, S/f(dog)\}\\
S_2 &= \{p(f(dog), f(dog), Z),\ p(f(dog), f(dog), g(cat)),\ p(f(dog), f(dog), g(W))\}\\
D_2 &= \{Z, g(cat), g(W)\}\\
\sigma_2 &= \{X/f(dog), T/f(dog), Y/dog, S/f(dog), Z/g(cat), W/cat\}\\
S_3 &= \{p(f(dog), f(dog), g(cat)),\ p(f(dog), f(dog), g(cat)),\ p(f(dog), f(dog), g(cat))\}
\end{aligned}
$$

Alle Terme in S_3 sind gleich, das Most General Unifier ist damit

$$\sigma_2 = \{X/f(dog), T/f(dog), Y/dog, S/f(dog), Z/g(cat), W/cat\}$$

Aufgabe 2)

Hier ist eine recht gute Quelle zur binären- sowie vollständigen-Resolution:

- http://www.cs.miami.edu/~geoff/Courses/TPTPSYS/FirstOrder/Resolution.shtml

a)

Es folgt nun die binäre Resolution, jedoch ohne der Zwischenschritte wie in den Beispielen der Quelle gezeigt wurde:

S = $\{\{p(X) \vee q(X)\}, \{\neg p(Z) \vee q(Z)\}, \{p(Y) \vee \neg q(Y)\}, \{\neg p(U) \vee \neg q(U)\}\}$

1. $\{p(X) \vee q(X)\}$
2. $\{\neg p(Z) \vee q(Z)\}$
3. $\{p(Y) \vee \neg q(Y)\}$
4. $\{\neg p(U) \vee \neg q(U)\}$

$$\frac{1.\ \ 2.}{5.} = \frac{\{p(X) \vee q(X)\} \quad \{\neg p(Z) \vee q(Z)\}}{\{q(X) \vee q(X)\} \equiv \{q(X)\}}$$ Ersetzungsregel: θ = {Z / X}

$$\frac{3.\ \ 4.}{6.} = \frac{\{p(Y) \vee \neg q(Y)\} \quad \{\neg p(U) \vee \neg q(U)\}}{\{\neg q(Y) \vee \neg q(Y)\} \equiv \{\neg q(Y)\}}$$ Ersetzungsregel: θ = {U / Y}

$$\frac{5.\ \ 6.}{7.} = \frac{\{q(X)\} \quad \{\neg q(Y)\}}{\textit{nicht resolvierbar}}$$

Es heißt die binäre-Resolution kann durch das zusätzliche Anwenden der „Factoring“-Regel zu einer vollständigen-Resolution erweitert werden.

Aus dem Buch „A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning“ von Larry Wos und Gail W. Pieper (**siehe Google Books**) haben wir folgendes recherchiert:

„***The object of an application of binary resolution is to produce a new clause from to existing clauses, each of which happens to contain an appropriate literal.*** *[…] Binary resolution is a generalization of the inference rule that yields (Q or R) from (P or Q) and ((not P) or R). Equivalently, binary resolution is a generalization of the inference rule that yields (Q or R) from (if P then Q) and (if (not P) then R).* ***The generalization allows Q or R (or both, see Section 3.5) to be empty. The generalization also allows for unification to be required to produce a required P and ~P.***“ (Siehe Abschnitt 3.4.2 Binary Resolution.)

„*Contradiction is found usually when two unit clauses are obtained that are alike in predicate, are opposite in sign, and are unifiable.*“ (Siehe Abschnitt 3.5 The Empty Clause.)

Daher gehen wir davon aus, dass unsere Lösung nicht weiter resolvierbar ist und damit zu keiner Lösung, also Kontradiktion, (ohne der Anwendung der „Factoring“-Regel) führt.

Aufgabe 2b: Full Resolution

$S = \{p(X)|q(X), \sim p(Z)|q(Z), p(Y)| \sim q(Y), \sim p(U)| \sim q(U)\}$

$CNF = p(X) \vee q(X) \wedge \neg p(Z) \vee q(Z) \wedge p(Y) \vee \neg q(Y) \wedge \neg p(U) \vee \neg q(U)$

$\neg CNF = \neg p(X) \wedge \neg q(X) \vee p(Z) \wedge \neg q(Z) \vee \neg p(Y) \wedge q(Y) \vee p(U) \wedge q(U)$

$\neg CNF = \underbrace{(\neg p \wedge \neg q)}_{1} \vee \underbrace{(p \wedge \neg q)}_{2} \vee \underbrace{(\neg p \wedge q)}_{3} \vee \underbrace{(p \wedge q)}_{4}$

Vollständige Resolution:

$1 + 2 + 4 = 5 = \neg q \wedge q = \emptyset$

Aufgabe 3: CNF-Resoluton

Umwandeln der Klausel $(A1 \wedge A2 \wedge A3) \Rightarrow B$

Simplify

$$\begin{aligned}\neg((\forall X(\neg p(X) \vee (\forall Y(\neg k(Y) \vee \neg m(X,Y)))))\wedge \\ (\forall X(\neg p(X) \vee (\exists Y(f(Y) \wedge m(X,Y)))))\wedge \\ (\exists X(p(X))))\vee \\ (\exists X(f(X) \wedge \neg k(X)))\end{aligned}$$

Move negations in

$$\begin{aligned}(\neg(\forall X(\neg p(X) \vee (\forall Y(\neg k(Y) \vee \neg m(X,Y)))))\vee \\ \neg(\forall X(\neg p(X) \vee (\exists Y(f(Y) \wedge m(X,Y)))))\vee \\ \neg(\exists X(p(X))))\vee \\ (\exists X(f(X) \wedge \neg k(X)))\end{aligned}$$

$$\begin{aligned}(\exists X\neg(\neg p(X) \vee (\forall Y(\neg k(Y) \vee \neg m(X,Y)))))\vee \\ \exists X\neg(\neg p(X) \vee (\exists Y(f(Y) \wedge m(X,Y)))))\vee \\ \forall X\neg(p(X)))\vee \\ (\exists X(f(X) \wedge \neg k(X)))\end{aligned}$$

$$\begin{aligned}(\exists X(\neg\neg p(X) \wedge \neg(\forall Y(\neg k(Y) \vee \neg m(X,Y)))))\vee \\ \exists X(\neg\neg p(X) \wedge \neg(\exists Y(f(Y) \wedge m(X,Y)))))\vee \\ \forall X\neg(p(X)))\vee \\ (\exists X(f(X) \wedge \neg k(X)))\end{aligned}$$

$$\begin{aligned}(\exists X(p(X) \wedge (\exists Y\neg(\neg k(Y) \vee \neg m(X,Y)))))\vee \\ \exists X(p(X) \wedge (\forall Y\neg(f(Y) \wedge m(X,Y)))))\vee \\ \forall X\neg(p(X)))\vee \\ (\exists X(f(X) \wedge \neg k(X)))\end{aligned}$$

$$\begin{aligned}(\exists X(p(X) \wedge (\exists Y(k(Y) \wedge m(X,Y)))))\vee \\ \exists X(p(X) \wedge (\forall Y(\neg f(Y) \vee \neg m(X,Y)))))\vee \\ \forall X\neg(p(X)))\vee \\ (\exists X(f(X) \wedge \neg k(X)))\end{aligned}$$

Move quantifiers out

$$\begin{aligned}(\exists A\exists B(p(A) \wedge (k(B) \wedge m(A,B)))\vee \\ \exists C\forall D(p(D) \wedge (\neg f(D) \vee \neg m(C,D)))\vee \\ \forall E\neg(p(E)))\vee \\ (\exists F(f(F) \wedge \neg k(F)))\end{aligned}$$

$$\begin{aligned}\exists A\exists B\exists C\forall D\forall E\exists F \\ ((p(A) \wedge (k(B) \wedge m(A,B)))\vee \\ (p(D) \wedge (\neg f(D) \vee \neg m(C,D)))\vee \\ \neg(p(E))\vee \\ (f(F) \wedge \neg k(F)))\end{aligned}$$

Skolemize

$$\begin{aligned}
\gamma &= \{\} \\
\exists A &: A/sk_A(\{\}) \\
\exists B &: B/sk_B(\{\}) \\
\exists C &: C/sk_C(\{\}) \\
\forall D &: \gamma = \{D\} \\
\forall E &: \gamma = \{D, E\} \\
\exists F &: F/sk_F(\{D, E\})
\end{aligned}$$

$$\begin{aligned}
((p(sk_A(\{\})) \wedge (k(sk_B(\{\})) \wedge m(sk_A(\{\}), sk_B(\{\}))))\vee \\
(p(D) \wedge (\neg f(D) \vee \neg m(sk_C(\{\}), D)))\vee \\
\neg(p(E))\vee \\
(f(sk_F(\{D, E\})) \wedge \neg k(sk_F(\{D, E\}))))
\end{aligned}$$

Distribute disjunctions and Convert to CNF

Zu viele Zeilen, wahrscheinlich ist auch irgendwo ein Fehler drinnen, Teilaufgabe b ist damit dann auch schlecht zu lösen...

Aufgabe 4: Occures Check

Der Occurs Check in SWI Prolog wird mit der Funktion $unifiy\_with\_occurs\_check(operand_1, operand_2)$ aktiviert.

Ohne Occurs Check:

```
?- A=f(A).
A = f(A).
?- A.
% ... 1,000,000 ............ 10,000,000 years later
%
%       >> 42 << (last release gives the question)
```

Mit Occurs Check:

```
?- unify_with_occurs_check(A,f(A)).
false.
```