Introduction to Artificial Intelligence

Practice Sheet 3

Gruppe: Ole Elija Dziewas, Marius Ludwig Bachmeier, Meike Valentina Bauer

Task 1.

Iteration 0 (initial set of formulas):

S = {g(D,h(x)), g(v,y), g(w,h(k(v))))}

$|S| > 1$

θ <- {}

D <- {D, v , w}

Iteration 1:

S = {g(D,h(x)), g(D,y), g(w,h(k(D))))}

$|S| > 1$

θ = {v <- D}

D <- {D, w}

Iteration 2:

S = {g(D,h(x)), g(D,y), g(D,h(k(D)))}

$|S| > 1$

θ = {w <- D}

D <- {h(x), y, h(k(D))}

Iteration 3:

S = {g(D,h(x)), g(D,h(x)), g(D,h(k(D)))}

θ = {y <- h(x)}

D <- {x, k(D)}

Iteration 4:

S = {g(D,h(k(D))), g(D,h(k(D))), g(D,h(k(D)))}

$|S| \ngtr 1$

θ = {x <- k(D)}

D = {}

MGU = {v <- D, w <- D, y <- h(x), x <- k(D)}

Task 2.

a)

Given background knowledge: ϕ = edge(A,B) ∧ edge(B,C) ∧ ∀x.∀y. edge(x,y)-> reachable(x,y) ∧ ∀x.∀y.∀z. (edge(x,z) ∧ reachable(z,y)) -> reachable(x,y)

Step 1: remove implications

edge(A,B) ∧ edge(B,C) ∧ ∀x.∀y.edge(x,y) -> reachable(x,y) ∧ ∀x.∀y.∀z. (edge(x,z) ∧ reachable(z,y)) -> reachable(x,y)

becomes

edge(A,B) ∧ edge(B,C) ∧ ¬(∀x.∀y.edge(x,y)) ∨ reachable(x,y) ∧ ∀x.∀y.∀z. ¬(edge(x,z) ∧ reachable(z,y)) ∨ reachable(x,y)

Step 2: reduce scopes of negation

edge(A,B) ∧ edge(B,C) ∧ ¬(∀x.∀y.edge(x,y)) ∨ reachable(x,y) ∧ ∀x.∀y.∀z. ¬(edge(x,z) ∧ reachable(z,y)) ∨ reachable(x,y)

becomes

edge(A,B) ∧ edge(B,C) ∧ ¬(∀x.∀y.edge(x,y)) ∨ reachable(x,y) ∧ ∀x.∀y.∀z. (¬edge(x,z) ∨ ¬reachable(z,y)) ∨ reachable(x,y) (DeMorgan)

Step 3: Skolemization

Due to no existential quantifiers being present in this formula, there are no transformations to be made in the step of Skolemization

Step 4: Standardize variables

edge(A,B) ∧ edge(B,C) ∧ ¬(∀x.∀y.edge(x,y)) ∨ reachable(x,y) ∧ ∀x.∀y.∀z.(¬edge(x,z) ∨ ¬reachable(z,y)) ∨ reachable(x,y)

becomes

edge(A,B) ∧ edge(B,C) ∧ ¬(∀x.∀y.edge(x,y)) ∨ reachable(x,y) ∧ ∀u.∀v.∀z.(¬edge(u,z) ∨ ¬reachable(z,v)) ∨ reachable(u,v)

Step 5: Prenex Form

edge(A,B) ∧ edge(B,C) ∧ ¬(∀x.∀y.edge(x,y)) ∨ reachable(x,y) ∧ ∀u.∀v.∀z.(¬edge(u,z) ∨ ¬reachable(z,v)) ∨ reachable(u,v)

becomes

∀x.∀y.∀u.∀v.∀z.[edge(A,B) ∧ edge(B,C) ∧ (¬edge(x,y) ∨ reachable(x,y)) ∧ (¬edge(u,z) ∨ ¬reachable(z,v)) ∨ reachable(u,v)]

Step 6: Conjunctive Normal Form

∀x.∀y.∀u.∀v.∀z.[edge(A,B) ∧ edge(B,C) ∧ (¬edge(x,y) ∨ reachable(x,y)) ∧ (¬edge(u,z) ∨ ¬reachable(z,v)) ∨ reachable(u,v)]

becomes

∀x.∀y.∀u.∀v.∀z.[edge(A,B) ∧ edge(B,C) ∧ (¬edge(x,y) ∨ reachable(x,y)) ∧ (¬edge(u,z) ∨ ¬reachable(z,v) ∨ reachable(u,v))] (Associativity)

Step 7: Eliminate Conjunctions

edge(A,B),

edge(B,C),

∀x.∀y.[¬edge(x,y) ∨ (reachable(x,y)],

∀u.∀v.∀z.[¬edge(u,z) ∨ ¬reachable(z,v) ∨ reachable(u,v)]

Step 8: Eliminate universal quantifiers

M = { edge(A,B), edge(B,C), ¬edge(x,y) ∨ reachable(x,y), ¬edge(u,z) ∨ ¬reachable(z,v) ∨ reachable(u,v) } b)

The background knowledge: $\phi = edge(A,B) \land edge(B,C) \land \forall x.\forall y.\ edge(x,y) \rightarrow reachable(x,y) \land \forall x.\forall y.\forall z.\ (edge(x,z) \land reachable(z,y)) \rightarrow reachable(x,y)$

The formula ψ that is to be proven: $\psi = reachable(A,C)$

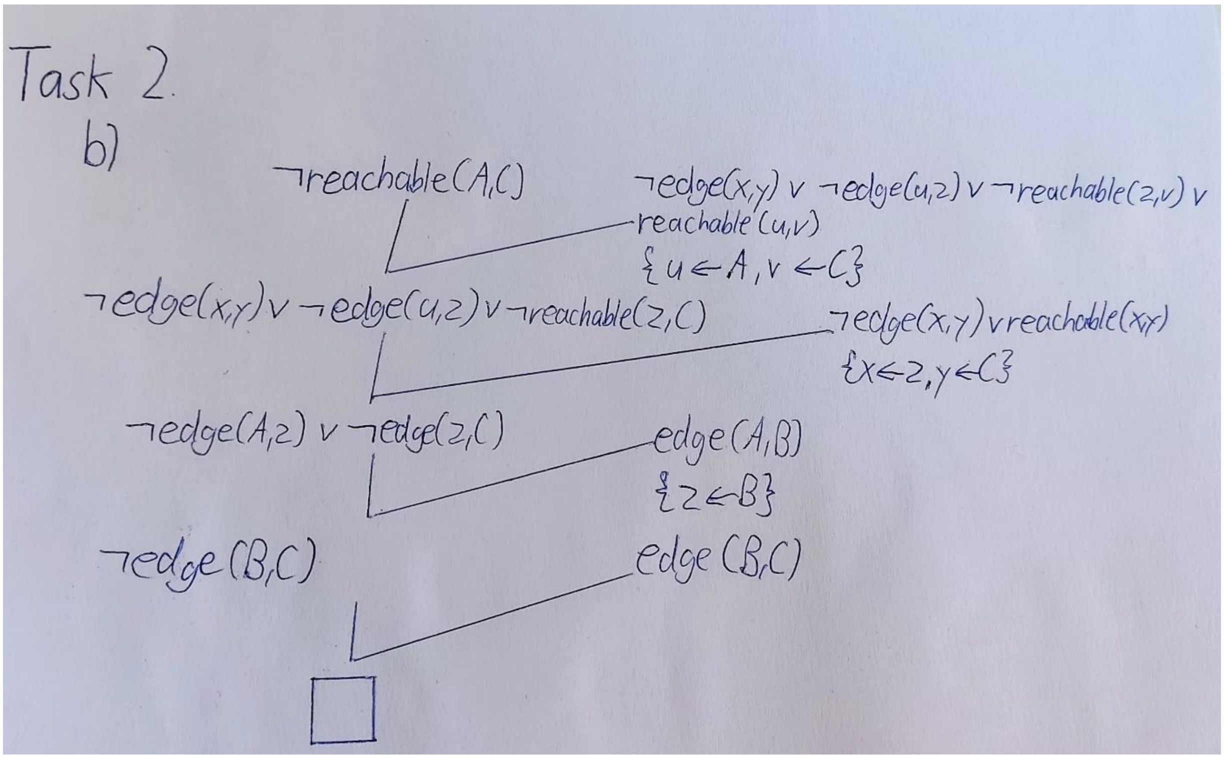
$\neg\psi = \neg reachable(A,C)$

Clause Form of the background knowledge and the negated formula that is about to be proven:

$M = \{\ edge(A,B),\ edge(B,C),\ \neg edge(x,y) \lor (reachable(x,y),\ \neg edge(x,y) \lor \neg edge(u,z) \lor \neg reachable(z,v) \lor reachable(u,v)\ \}$

$C = \{\neg reachable(A,C)\}$

Refutation Tree



Via the refutation tree from above: $\phi \vDash \psi$

Task 3.

$\forall x. [ordered(cons(x, Nil)) \rightarrow T] \vee$

$\forall x. \forall y. \forall l. [ordered(cons(y, cons(x, l)) \rightarrow leq(x,y) \wedge ordered(cons(x,l))]$

Task 4.

a) Prolog fact base:

```
edge(a,b).

edge(b,c).

reachable(X,Y) :- edge(X,Y).

reachable(X,Y) :- edge(X,Z), reachable(Z,Y).
```

→ Query: reachable(a,c). results in true

b) Due to the properties of SLD resolution an infinite recursion occurs.