Kurt Rose
Steven Bo...

(1) convert $A \wedge ((B \wedge C) \vee G) \wedge \neg A$ into CNF:

$(B \wedge C) \vee G \equiv (B \vee G) \wedge (C \vee G)$

$A \wedge ((B \wedge C) \vee G) \wedge \neg A \equiv A \wedge (B \vee G) \wedge (C \vee G) \wedge \neg$

(2)

$$A \wedge (B \cup G) \wedge (C \cup G) \wedge \neg A$$

$$\Downarrow$$

$$\{A\}, \{B, G\}, \{C, G\}, \{\neg A\}$$

$$\Downarrow$$

$$\{\}$$

3) ① Negate the goal $\{\bar{B}\}$

② Choose 1st rule with B in head

$$\frac{\{B, \bar{A}, \bar{C}\}, \{\bar{B}\}}{\{\bar{A}, \bar{C}\}}$$

③ Choose 1st rule with A in head

$$\frac{\{A\}, \{\bar{A}, \bar{C}\}}{\{\bar{C}\}}$$

④ Choose 1st rule with C in head

$$\frac{\{C, \bar{E}\}, \{\bar{C}\}}{\{\bar{E}\}}$$

⑤ Choose 1st rule with E in head

$$\frac{\{E, \bar{A}\}, \{\bar{E}\}}{\{\bar{A}\}}$$

⑥ Choose 1st rule with A in head

$$\frac{\{A\}, \{\bar{A}\}}{\{\}}$$

Done.

④ Forward chaining would start w/ an empty agenda then add facts to get $\{A, C\}$. It would then see w/c consequences were derivable from these facts in one step using $B :- A, C.$ and $E :- A$ we could derive $\{A, C, B, E\}$. This contains B so done

⑤ $\forall x \exists y (\text{Minute}(x) \Rightarrow \text{BornWhen}(y, x) \wedge \text{Such}(y))$

6 Want to compute

Unify(t(s(h(x),0),z), t(d,g(s(n,m))), ())

↳ compound? ck of unify satisfied so compute

Unify((s(h(x),0),z), (d g(s(n,m)))),
Unify(t(slot1,slot2), t(slot1,slot2), ())

since 1st two parameters same this returns third parameter. i.e., ()

So want to compute

unify((s(h(x),0),z), (d g(s(n,m))), ())

↳ List? ck holds so compute

unify((z), (g(s(n,m))), Unify(s(h(x),0), d, ()))

d is a variable so compute

Unify-Var(d, s(h(x),0), ())

↓

now get substitution list

(d/s(h(x),0))

So need to compute

Unify((z), (g(s(n,m))), (d/s(h(x),0)))

↳ List? ck holds so compute

Unify((), (), Unify(z, g(s(n,m)), (d/s(h(x),0))))

after compute Unify-var get (z/g(s(n,m)), (d/s(h(x),0)))

↳ as

() == ()

this last unify returns θ =

(z/g(s(n,m)), d/s(h(x),0))

So mgu would be

t(s(h(x),0), g(s(n,m)))

7. s --> [].
s --> "(", s, ")", s.

Corresponds to

```
s(Input, Input).
s(Input, AfterParen) :- Input=[40, After1
                         s(After1, After2),
                         After2=[41, After
                         s(After3, AfterPare
```

Thomas
Twain.

(8) Effect Axiom : changes that result ~~from~~ taking actions.

~~possibility Axiom~~ : when it possible to execute the action

~~Frame problem~~ : Representing the things ~~that~~ stays the sam

(8)

Effect Axiom : An axiom the situation calculus that Specifies the results of some action.

possibility ax : An axiom in the situation calc. of the form : if certain preconditions are met then some action is possible.

Frame prob : Representing the things that stays the same after you have done an action in the situation calculus.

9

S': On(A,B) & On(B,C) This is goal.

1. Move(A, Table, B)
-Remove On(A,B), Clear(Table) from S'. Add precondition literals.

S2: On(B,C) & On(A,Table) & Clear(A) & Clear(B) & Block(A)

2. Move(B, Table,C)
-Remove On(B,C), Clear(Table) from S2. Add precondition literals.

S1: On(A,Table) & On(B,Table) & Clear(A) & Clear(B) & Clear(C) & Block(A)
& Block(B)

This contains only things which are true in our initial goal state so planner would
terminate with the plan: Move(B, Table,C), Move(A, Table, B)