Most general unifier (MGU):

matching logical sentences is the first step in MP
e.g., Cat(x) $\Rightarrow$ Mammal(x), Cat(Felix) |– Mammal(Felix)

UNIFY[P(A, B, y, z), P(x, y, z, B)]

same predicate *P()*, same arity of 4 arguments

unifier: A $\equiv$ x, B $\equiv$ y, y $\equiv$ z, z $\equiv$ B

θ = { x / A, y / B, z / B }

UNIFY[Knows(x, x), Knows(Father(y), y)]

same predicate *Knows()*, same arity of 2 arguments

unifier: x $\equiv$ Father(y), x $\equiv$ y

θ = { x / Father(y), x / y }

implies Father(y) = y *?!?* *F(y)=y ?*

→ impossible *if assuming* the most obvious semantics i.e., “fatherhood”; then θ = { }

→ possible *if assuming* another semantics e.g., “provider” etc. (F not defined!)

Backward chaining inference:

knowledge base (Horn clauses):

(1) Sick(John, 2006)
(2) Sick(Mary, y)
(3) SitsNextTo(Ginger, Mary, 2006)
(4) SitsNextTo(a, b, y) $\wedge$ Sick(b, y) $\Rightarrow$ Sick(a, y)
(5) Parent(a, b) $\wedge$ HasCold(b, y) $\Rightarrow$ HasCold(a, y)
(6) HasCold(a, y) $\Rightarrow$ Sick(a, y)

backward chaining MP: *? Sick(x, 2006)*

(1) under x=John |– Sick(John, 2006)

(2) under x=Mary, y=2006 |– Sick(Mary, 2006)

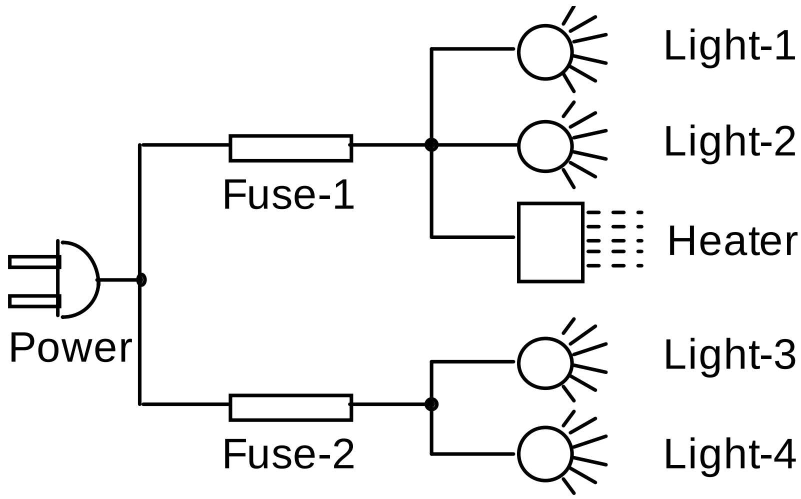
(3) *fails*

(4) prove Sick(a, y) under x=a, y=2006
→ SitsNextTo(a, b, 2006) and Sick(b, 2006)
→ (3) under a=Ginger, b=Mary
(2) b=Mary |– Sick(Ginger, 2006)

(5) *fails*

(6) prove Sick(a, y) under x=a, y=2006
→ HasCold(a, 2006)
→ (5) Parent(a, b) $\wedge$ HasCold(b, 2006)
fails (no matching clause…)

Expert System for an electric network



constants: P for Power; F1, F2 for Fuse-1, Fuse-2

L1, L2 etc. for Light-1 … , H for Heater

predicates:

On(d)	"device d is turned on"
Working(d)	"device d is working"
Broken(d)	"device f is broken"
Connected(e, f)	"devices e and f are connected"
Intact(f)	"device f is intact"
Fuse(f)	"device f is a fuse"
Hot(d)	"device d is hot"

Device(d) ? → *not needed (everything is…)*

Knowledge base rules in FOL:

"All devices are on if there is power and off otherwise."

(1) $\forall d \; On(Power) \Rightarrow On(d)$

(2) $\forall d \; \neg On(Power) \Rightarrow \neg On(d)$

or $\forall d \; On(d) \Rightarrow On(Power)$

"If the room is hot, the heater is working."

(3) $Hot(R) \Rightarrow Working(H)$

"If a device is on, connected to a fuse and the fuse is intact, but the device is not working, then it is broken."

(4) $\forall d,f \; On(d) \wedge Connected(d, f) \wedge Fuse(f) \wedge Intact(f) \wedge \neg Working(d) \Rightarrow Broken(d)$

"A fuse is intact if a device connected to it is working."

(5) $\forall d,f \; Fuse(f) \wedge Connected(d, f) \wedge Working(d) \Rightarrow Intact(f)$

"If two different devices connected to the same fuse are on but not working, the fuse is not intact."

(6) $\forall d,e,f \; Connected(d, f) \wedge Connected(e, f) \wedge \neg d = e \wedge Fuse(f) \wedge On(d) \wedge On(e) \wedge \neg Working(d) \wedge \neg Working(e) \Rightarrow \neg Intact(f)$

"A working device is clearly on."

(7) $\forall d \; Working(d) \Rightarrow On(d)$

Knowledge base rules in CNF:

all $\forall$ already on the left, no $\exists$… need convert all $\Rightarrow$

replace $P \Rightarrow Q$ by $\neg P \vee Q$, distribute $\neg$, etc.

$P_1 \wedge P_2 \wedge \dots \wedge P_N \Rightarrow Q \quad \Leftrightarrow \quad \neg P_1 \vee \neg P_2 \vee \dots \vee \neg P_N \vee Q$

(1) $\neg$ On(Power) $\vee$ On(d)

(2) On(Power) $\vee$ $\neg$ On(d)

(3) $\neg$ Hot(R) $\vee$ Working(H)

(4) $\neg$ On(d) $\vee$ $\neg$ Connected(d, f) $\vee$ $\neg$ Fuse(f) $\vee$ $\neg$ Intact(f) $\vee$ Working(d) $\vee$ Broken(d)

(5) $\neg$ Fuse(f) $\vee$ $\neg$ Connected(d, f) $\vee$ $\neg$ Working(d) $\vee$ Intact(f)

(6) $\neg$ Connected(d, f) $\vee$ $\neg$ Connected(e, f) $\vee$ d = e $\vee$ $\neg$ Fuse(f) $\vee$ $\neg$ On(d) $\vee$ $\neg$ On(e) $\vee$ Working(d) $\vee$ Working(e) $\vee$ $\neg$ Intact(f)

(7) $\neg$ Working(d) $\vee$ On(d)

Knowledge base facts:

electric network information:

(8)	Connected(L1, F1)
(9)	Connected(L2, F1)
(10)	Connected(H, F1)
(11)	Connected(L3, F2)
(12)	Connected(L4, F2)
(13)	Connected(F1, P)
(14)	Connected(F2, P)
(15)	Fuse(F1)
(16)	Fuse(F2)

additional information:

“All lights except Light-2 are working …”

(17)	Working(L1)
(18)	¬ Working(L2)
(19)	Working(L3)
(20)	Working(L4)

“… and the room is hot.”

(21)	Hot(R)

c) Proof by resolution: “Is Light-2 broken?”

resolving (4) with (9), (15), and (18) under d=L2, f=F1 :

(22) $\neg$ On(L2) $\vee$ $\neg$ Intact(F1) $\vee$ Broken(L2)

resolving (5) with (8), (15) and (17) under d=L1, f=F1 :

(23) Intact(F1)

resolving (22) with (23):

(24) $\neg$ On(L2) $\vee$ Broken(L2)

resolving (1) with (24) under d=L2 :

(25) $\neg$ On(P) $\vee$ Broken(L2)

resolving (3) with (7) and (21) under d=H :

(26) On(H)

resolving (2) with (26) under d=H :

(27) On(P)

resolving (25) and (27) :

(28) <u>Broken(L2)</u> → *Light-2 is broken*