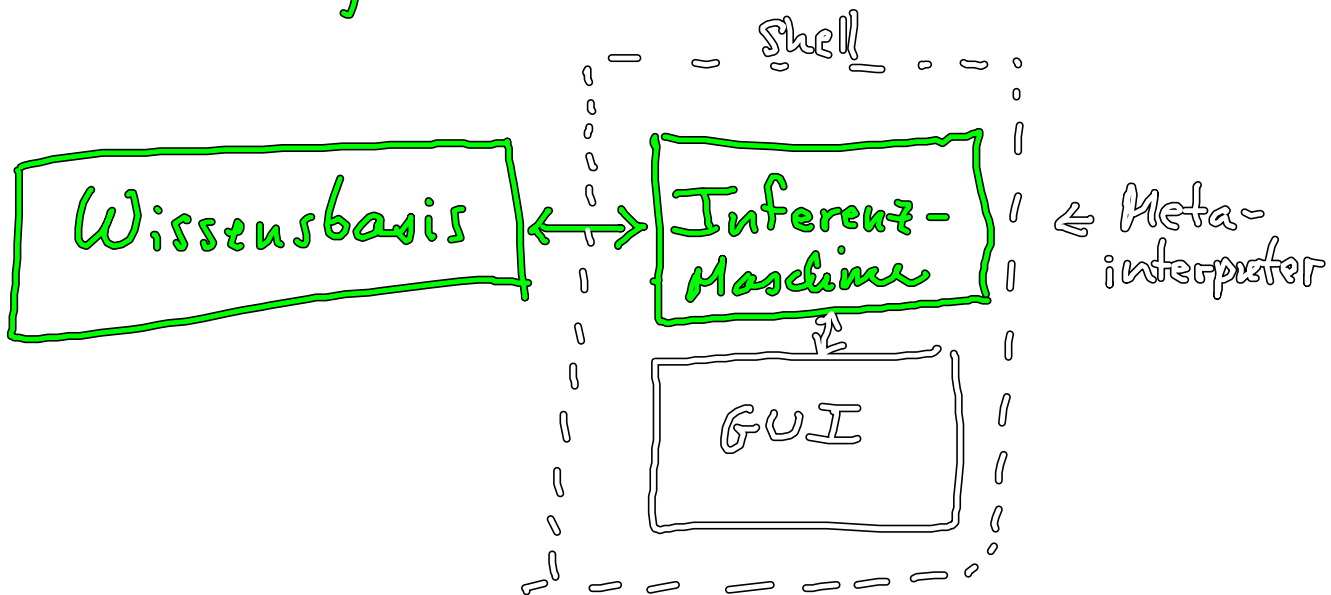


Experten-Systeme

MYCIN } Knowledge base
.....



→ Backward chaining (Prolog)

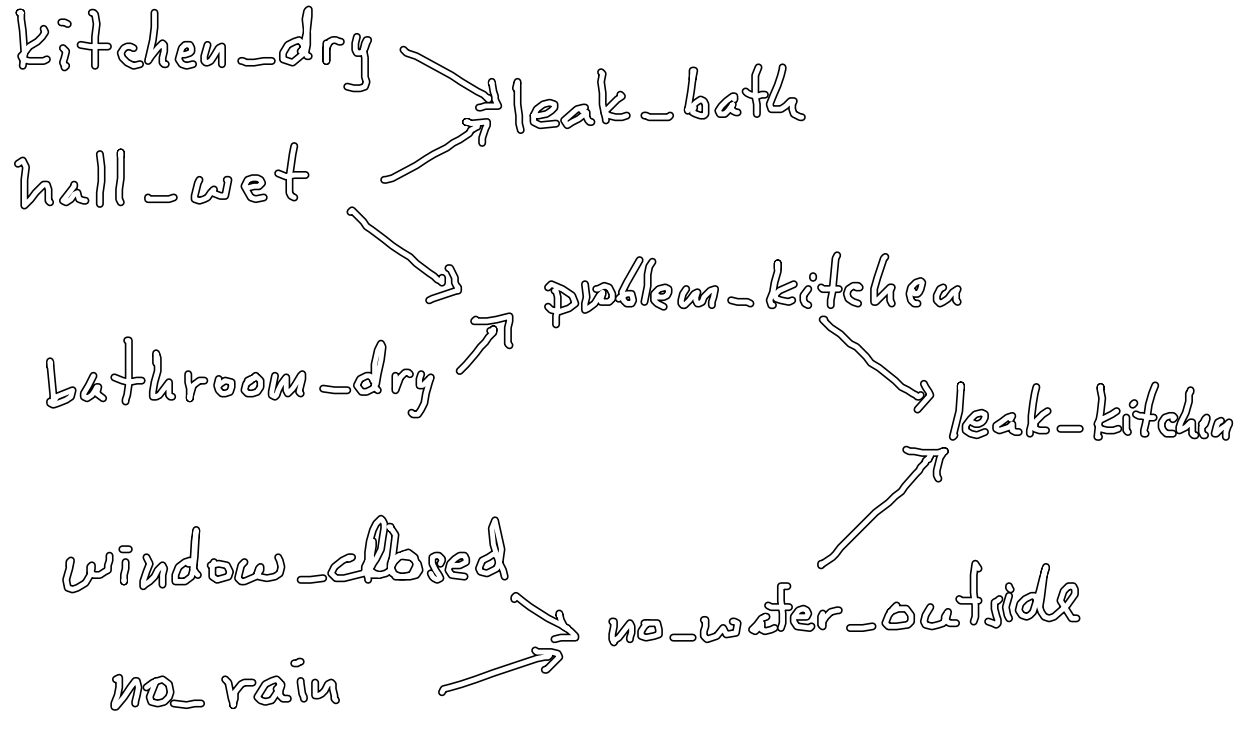
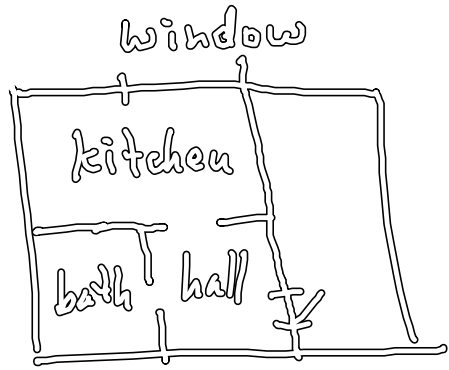
→ Forward chaining

→ a :- b, c, d.
b.

..... N-Prolog
?a.
x → a

c.
 d.
 ? a.
 ↳ b, c, d
 ↳ true, c, d
 ↳
 yes

x, y → a
 x; y → a
 ⋮
 z → w
 w → a



$:- \text{op}(800, fx, if).$

if hall_wet and kitchen_dry
then
leak_bath.

if ('; (hall_wet, kitchen_dry), leak_bath)

$:- \text{op}(800, fx, if).$

$:- \text{op}(700, xfx, then).$

$:- \text{op}(200, xfy, and).$

$:- \text{op}(300, xfy, or).$

a and b and c

((a and b) and c)

{ if hall_wet and kitchen_dry
then
leak_bath.

R2 { if hall-wet and bathroom-dry
then
problem-in-kitchen.

R3 { if window-closed or no-rain
then
no-water-outside.

R4 { if problem-kitchen and
no-water-outside
then
leak-kitchen.

fact(hall-wet).
fact(bathroom-dry).
fact(window-closed).

— . —

Meta interpreter

is_true(P) :- fact(P).

BC { is_true(P) :-
 if Condition then P,
 is_true(Condition).

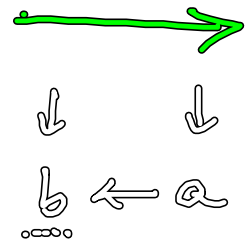
is_true(P1 and P2) :-

is_true(P1), is_true(P2).

is_true(P1 or P2) :-

is_true(P1); is_true(P2).

— . —



Forward_Chaining

if window-closed
 then
 no(~~problem~~-in-kitchen).
 asserta(nofact(problem-in-kitchen))

fact(das-ist-wahr).
 nofact(das-ist-falsch).

Exkurs:

$a: -b, c, d.$		$b \equiv 1$
$b: \text{true.}$		$c \equiv 1$
$c.$		$d \equiv 1$
$d.$		
		$b, c, d \rightarrow a$
		$\neg b \vee \neg c \vee \neg d \vee a$
		0 0 0 1
		}
		1

pos. literal Resolution

$(\neg b \vee \neg c \vee \neg d \vee a)$ $(\neg \text{true} \vee b)$

Resolution

$$\neg c \vee \neg d \vee a$$
$$c, d \rightarrow a$$

$$b \rightarrow \neg a$$
$$(\neg b \vee \neg a)$$

— ENDE DES EXKURSES —

$$a \rightarrow b ; c$$
$$\neg a \vee \underset{=}{b} \vee \underset{=}{c}$$

Horn-Klausel

Metainterpreter

forward :-

new_derived_fact(P),

!,

write('Derived:'), write(P), nl,

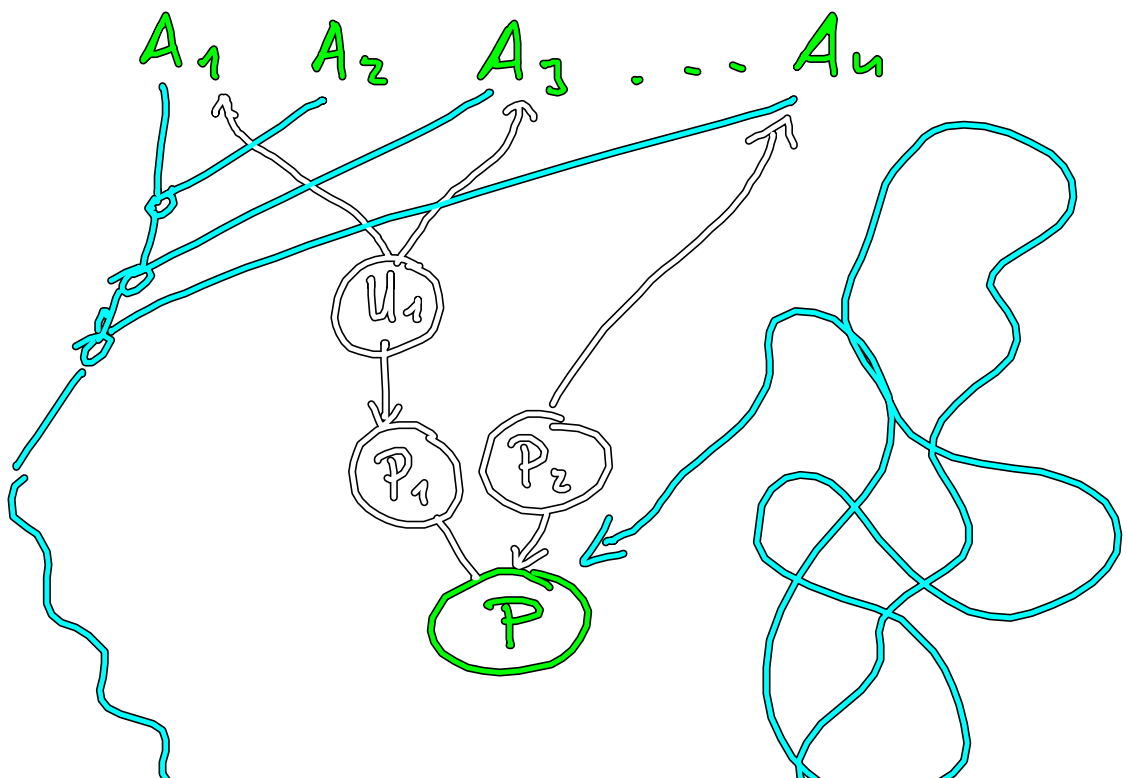
asserta(fact(P)),

forward ;
write ('no more truths').

→ asserta

? asserta (b).
? asserta (c).
? asserta (d).
? b.
yes
?

⋮ c.
b.
d.



new_derived_fact (Concl) :-

if Cond then Concl,
not (fact(Concl)),
composed_fact (Cond).

composed_fact (Cond) :-
fact (Cond).

composed_fact (Cond1 and Cond2) :-
composed_fact (Cond1), composed_fact (Cond2).

.....
composed_fact (Cond1 or Cond2) :-
..... " ; "

? forward.

====
====
====
====
====

Meta-3

Erklärungskomponente



$:- \text{op}(\text{foo}, \text{xfx}, \text{<=})$

$\text{is_true}(P, P) :- \text{fact}(P).$

$\text{is_true}(P, \text{axiom}(P)) :- \text{fact}(P).$

$\text{is_true}(P, P \text{<= Proof}) :-$

if Cond then P,

$\text{is_true}(\text{Cond}, \text{Proof}).$

$\text{is_true}(P_1 \text{ and } P_2, \text{Proof}_1 \text{ and } \text{Proof}_2) :-$

$\text{is_true}(P_1, \text{Proof}_1),$

$\text{is_true}(P_2, \text{Proof}_2).$

$\text{is_true}(P_1 \text{ or } P_2, \text{Proof}) :-$

$\text{is_true}(P_1, \text{Proof});$

$\text{is_true}(P_2, \text{Proof}).$

— . —

$\text{fact}(\text{hall-wet}).$

$\text{fact}(\text{bathroom-dry}).$

fact (window-closed).

? is-true (leak-kitchen ~~is~~, Beweis).

if hall-wet and kitchen-dry
| ~~then~~

if problem-in-kitchen and no-water-
then
leak-in-kitchen

egalwas (1, a, b).

kitchen-dry \Leftarrow and

Probs

? solve (a). } Expertensystem
 { yes

? solve (a, Proof). }
yes no
 $b \wedge c \rightarrow d$
 $d \rightarrow e$
 $e, f \rightarrow a$

if a und b dann c : vielleicht
0,8

Verisches Wissen

0 \longleftrightarrow 100%
... ..