

Kapitel 5

- Pattern search ←
- Production systems
- Blackboard

? a.

Intern

1. Klausel $\neg a$
 $a \leftarrow b$ $b \rightarrow a$
 $\neg b \vee a$

$(\cancel{\neg a}) \wedge (\cancel{\neg b \vee a})$
Res
 $\neg b$

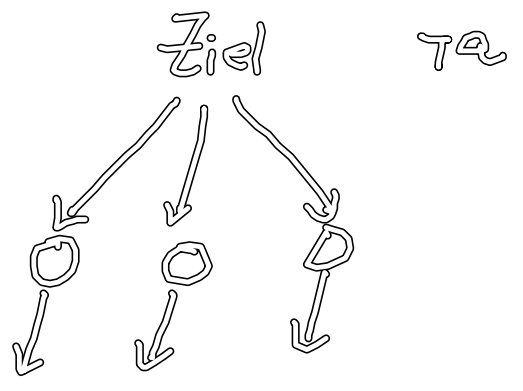
2. Klausel b . $\neg a \vee b$

$(\cancel{\neg b}) \wedge (\cancel{b})$

 Widerspruch!

$\Rightarrow \neg a$ Widerspruch
 $\Rightarrow a$ wahr

Goal directed search



Pattern search

? vater (X, adam).
 ⋮
 vater (gott, adam).
 ⋮

Metainterpreter

? solve (vater (X, adam)).
 ↑
 Prolog

1. Versuch

$\text{solve}(\text{true}) :- !.$

$\text{solve}((A, B)) :- !, \text{solve}(A),$
 $\text{solve}(B).$

$\text{solve}(A; B) :- !, (\text{solve}(A);$
 $\text{solve}(B)).$

$\text{solve}(A) :- \text{system}(A), \text{call}(A).$

$\text{solve}(A) :- \text{clause}(A, B),$
 $\text{solve}(B).$

$\text{system}(=(_, _)).$

$\text{system}(\text{fail}).$

$\text{system}(==(_, _)).$

$\text{system}(\text{write}(_)).$

\vdots

$\vdots \text{S} \rightarrow X = Y$

$\vdots \text{I} \rightarrow = (X, Y)$

$1 == 2$

$== (1, 2)$

Syntax [vater (adam, abel).

Intern [vater (adam, abel) :- true.]

Kopf

Body

? clause (vater (adam, abel), x).
X = true

? solve (vater (X, abel)).
X = adam

? solve (opa (X, isaac)).

⋮
? solve (fail).

↳ call (fail)
fail

? solve (solve (true)).

? solve (solve (solve (solve (true))))).

solve (solve (A)) :- solve (A).

— . —

$d :- a, b, c, !, d, e.$
 CP → $d :-$
 $d :-$

call(!)

2. Versuch

Cut?
 Goal ↓ nach dem Cut
 ↓

$solve(X) :- solve(X, C, R),$
 $(C == !, !, solve(R);$
 $true).$

Syntax $a, b, c, !, d, e$

Intern $(a, (b, (c, (!, (d, e))))$
 Interpretat ↓
 $C = !$
 $R = (d, e)$

$solve(true, -, -) :- !.$

$solve(!, !, true) :- !.$

$\text{solve}((A, B), x, Y) :- !,$

$\text{solve}(A, x, R),$

$(x == !, !, Y = (R, B));$

$\text{solve}(B, x, Y).$

$\text{solve}(A; B, x, Y) :- !,$

$(\text{solve}(A, x, Y), (x == !, !; \text{true});$

$\text{solve}(B, x, Y)).$

$\text{solve}(A, -, -) :- \text{system}(A), !,$

$\text{call}(A).$

$\text{solve}(A, -, -) :- \text{clause}(A, B),$

$\text{solve}(B, x, R),$

$(x == !, !, \text{solve}(R); \text{true}).$

$:- \text{dynamic} \text{ vater}/2.$

$:- \text{dynamic} \text{ solve}/3.$

\vdots

$\text{---} \bullet \text{---}$

:-dynamic solve/1
:-dynamic system/1
⋮

Unifikation

$X = [1, 2, 3]$
↑

append([], [1], R)
⋮
append([], L, L).

$x(1)$ $x(2)$ $x(100)$

unify($X, Y, \text{Sublist}$)

? unify($x(1), 2, L$).

$L = [x(1)/2]$

unify (X, Y, Sublit) :-

unify (X, Y, [], Sublit).

↑
bis
hierher

↑
nach
X=Y

unify (x(N), x(N),

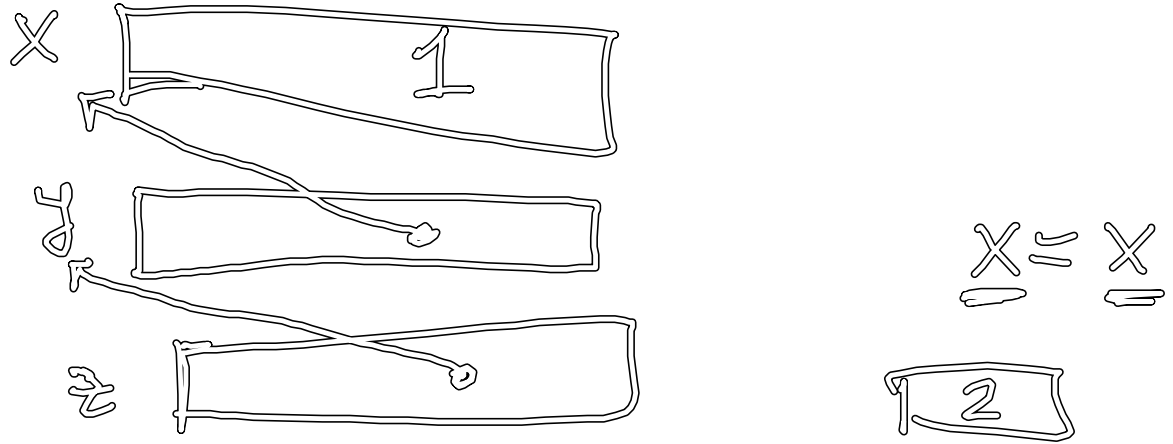
$X = X$

? $\underbrace{y = X}$, $\underbrace{z = Y}$, $\underbrace{x = z}$.

x und
y sind dasselbe

$X = X$?

true.



$$z = x, y = 1$$

$$x = y, y = z, z = 1$$

$\text{unify}(x(N), x(N), \text{Sublist}, \text{Sublist}) :- !.$

$\text{unify}(x(N), x(M), \text{Sublist}, \text{Sublist}) :-$

$\text{member}(x(N)/x(M), \text{Sublist}), !;$

$\text{member}(x(M)/x(N), \text{Sublist}), !.$

? $\text{unify}(x(N), x(M), \text{Sublist},$
 $[x(N)/x(M) | \text{Sublist}]) :- !.$

$x=1, y=2, x=y, \dots$

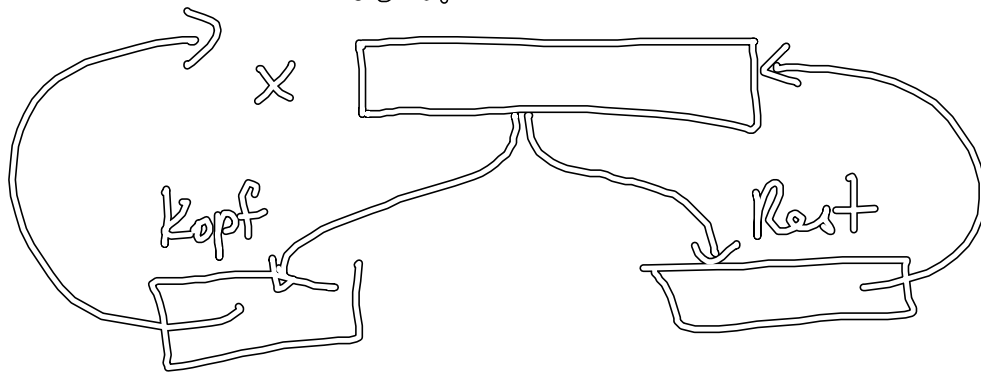
$\text{unify}(x(N), y, \text{Sublist}, \text{Sublist}) :-$

$\text{member}(x(N)/Y, \text{Sublist}), !.$
 $\text{unify}(x(N), Y, \text{Sublist}, _):-$
 $\text{occurs_in}(x(N), Y), !, \text{fail}.$
 \vdots

Interpreter

occurs check

? $X = [X|X].$ unendliche Struktur
 fail



$\text{unify}(Y, x(N), \text{Sublist}, \text{Sublist}):-$
 $(\text{member}(x(N)/Y, \text{Sublist})$
 $\text{member}(Y/x(N), \text{Sublist}), !.$
 \vdots

$\text{unify}(X, Y, \text{Sublist}, \text{Sublist}):-$

atomic(x), (atomic(y), !,
x==y; !, fail).

unify (y, x, lsublist, rsublist):-

atomic(x), (atomic(y), !,
x==y; !, fail).

⋮

Listen

Strukturen

⋮

{ x(N) schon instanziiert