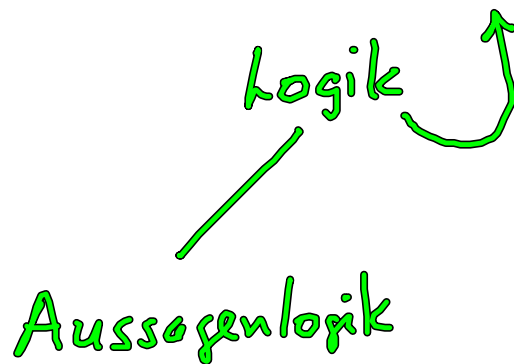


1

2

Prädikatenlogik



Atomare Aussagen

a b c

Wahrheitswert

a = "Berlin ist die Hauptstadt"

Belegung

$\mathcal{A}(a) = 1$

oder

$\mathcal{A}(a) = 0$

$a \wedge b$

"AND"

$a \vee b$

"OR"

$\neg a$

"NOT"

Metainterpreter

a	b	\wedge
0	0	0
0	1	0
1	0	0
1	1	1

$a \wedge b$

$a \vee b$

$\neg a$

— . —

Atom

Interpretation

1

True

true

True

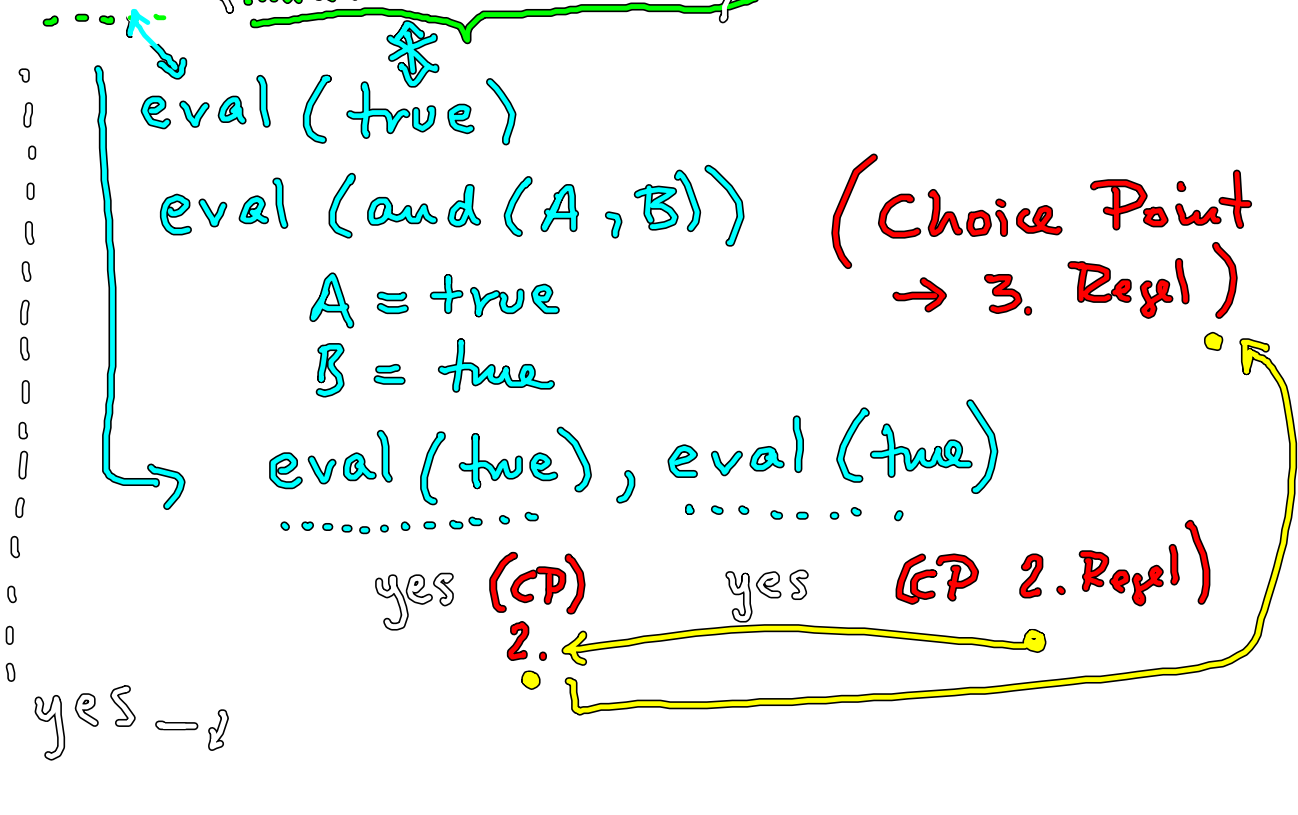
$\text{eval}(\text{true}) :- \text{true}.$

$\text{eval}(\text{and}(A, B)) :- \text{eval}(A),$
 $\text{eval}(B).$

$\text{eval}(\text{or}(A, B)) :- \text{eval}(A);$
 $\text{eval}(B).$

$\text{eval}(\text{not}(A)) :- \text{not}(\text{eval}(A)).$

? eval (and (true, true)).



$(T \wedge T) \wedge T$

? eval (and (and (true, true) true)).

↳ eval (and (true, true)),
eval (true)

↓ ↳ yes und yes

yes -

? eval (or (true, true)).

↳ eval (true); eval (true)

~
ORDER

yes ; eval(true)

? yes \Leftarrow

? eval(or(untue, untue)).

\hookrightarrow eval(untue); eval(untue)
fail ODER fail

? fail

eval(untue)

R₁
R₂
R₃
R₄

Metapredikat

? not(true).

fail

? eval(not(true)).

R₄ eval(not(A)) :- not(eval(A)).
.....

A=true

not(eval(true))

fail

?

? not (fail).

yes

?

.....

Meta-Interpreter für not

$\left. \begin{array}{l} \text{not}(A) :- \text{call}(A), !, \text{fail}. \\ \text{not}(A) :- \text{true}. \end{array} \right\}$

? not (true).

A = true ~~(CP 2. Regel)~~

call (true), !, fail.

yes ↑
 cut

eval (nicht (A)) :-
not (eval (A)).

fail

? eval (eval (true)).

fail

?

R5: $\text{eval}(A) :- \text{call}(A)$.

? $\text{eval}(\text{eval}(\text{true}))$.

↳ $\text{eval}(\text{true})$

yes

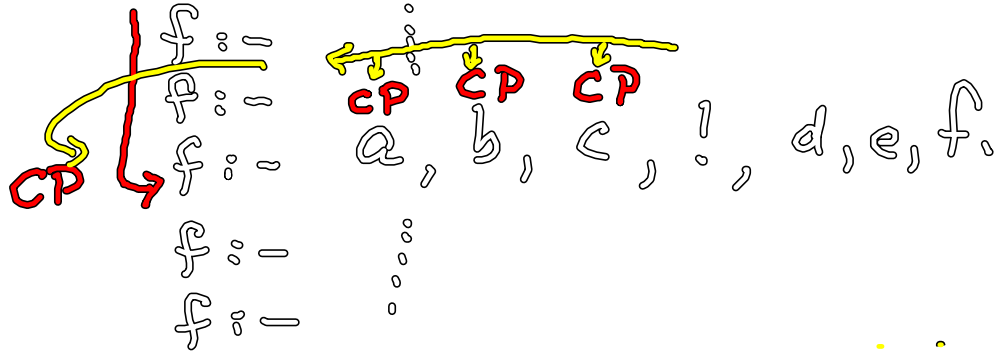
? $\text{eval}(\text{irgendwas}(\text{true}))$.

$\text{and}(\text{and}(\text{true}, \text{untue}), \text{untue})$

$\text{belubigen}(\text{sofo}(\text{a}, \text{b}), \text{dunns})$

R5: $\text{eval}(\text{untue}) :- !, \text{fail}$.

R6: $\text{eval}(A) :- \text{write}(\text{'error'})$.



a: - i, j, k.
a: - true.

? not (fail). CP 2. Regel von not

↳ call (fail), !, fail.

fail

↳ true

yes

? not (fail).

R1: not (A): - call (A), !, fail.

R2: not (A): - true.

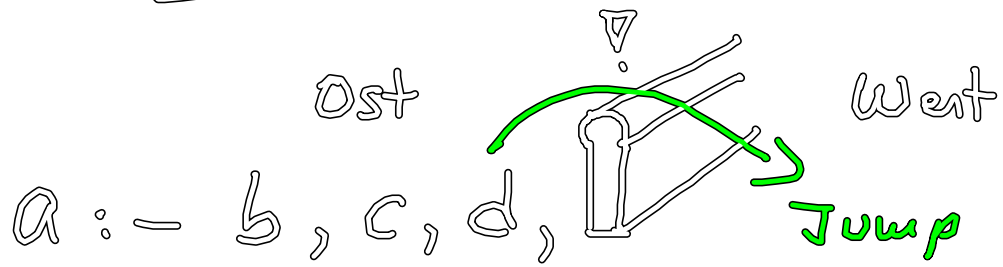
↳ call (fail), !, fail.

fail

fail

yes

GOTO



? eval (and (value, value)).

↳ eval (value), eval (value)

fail

Hausaufgabe: auf Konsistenz prüfen!

? eval (true, true).

eval (true) :-
()

eval/2 (true, true)

eval/1 ()

()
()
? blablab.

Prädikatenlogik = Aussagenlogik
+ Quantoren Variablen
+ Funktionen
0 1 \wedge \vee \neg

$\forall x$ {adam, eva, cain}

$\exists x$

S(adam)
vater von Adam

forall (x, Aussage)

exists (x, Aussage)

$x \in \{0, 1\}$

$x \vee \neg x \equiv 1$

forall (X, or (X, not(X)))
exists (X, and (X, X))

→ Konjunktive Normal Form