KI

8:30 – 10:00

- Artificial Intelligence
  Luger
  Pearson Books

1) K.I.
2) Prolog, LISP
3) Suchalgorithmen
   - Tiefensuche
   - Breitensuche
4) Effiziente Alg.
   - A* (Heuristik)
5) Logik: Prädikatenlogik
6) Sprache Prolog → Syntax
7) Wissensrepräsentation
8) Expertensysteme

Tutorium: Fr 10-12 Marco Block

Mi 14-16 E. Tapia

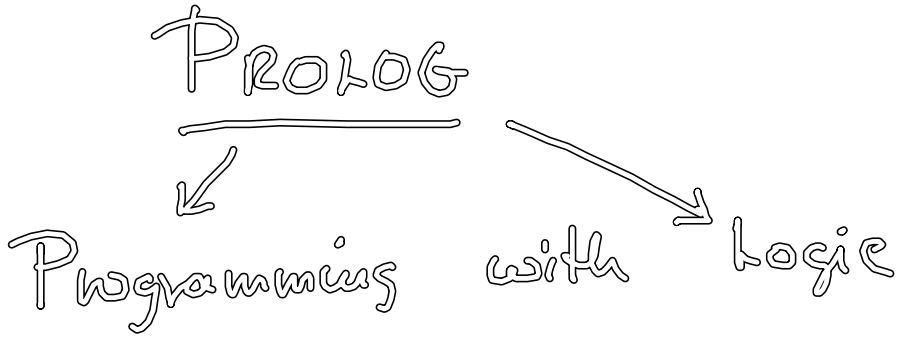
Übungen: 1 Übung

.....

60% der Punkte aus Übungen

⇒ Klausur ⇒ Schein

SWI Prolog



$a \leftarrow b \wedge c \wedge d$

$b \wedge c \wedge d \rightarrow a$

Horn-Klausel

$a \wedge b \wedge c \rightarrow (d \vee e)$

Keine Horn-Klausel

1974

LISP - 1959

Strukturen

Terme
- Konstanten
- Variablen
- Strukturen

Konstante
- Atome
- Zahlen

Bsp:

235

adam ← Atome fangen mit Kleinbuchs. faben an

eva

cain

abel

lilith 2

abel_eins

Bsp. Variablen:

X ← fangen mit Großbuchstabe an

Unbekannt

~~1adam~~

Bsp: Struktur

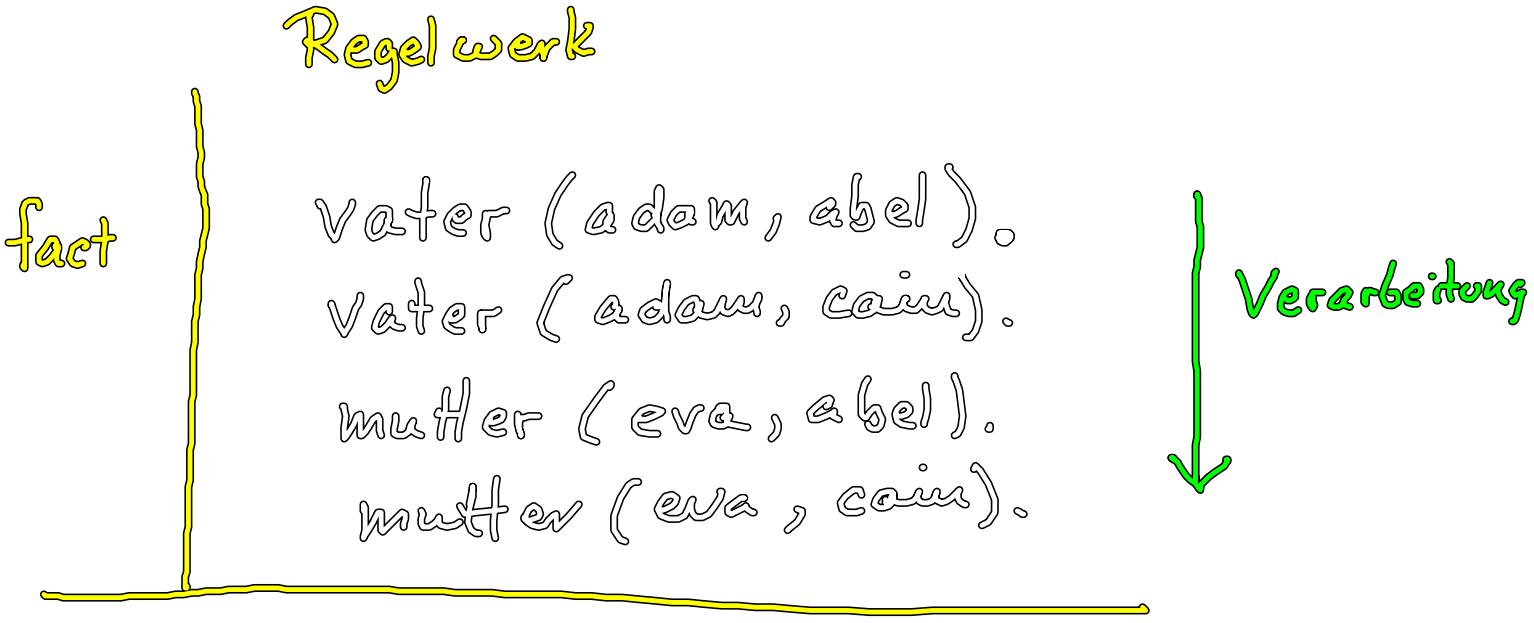
student (emil, mueller, 276107, X)

Functor

Arity

book (illuminati, dan, brown, 4)

Anfragen

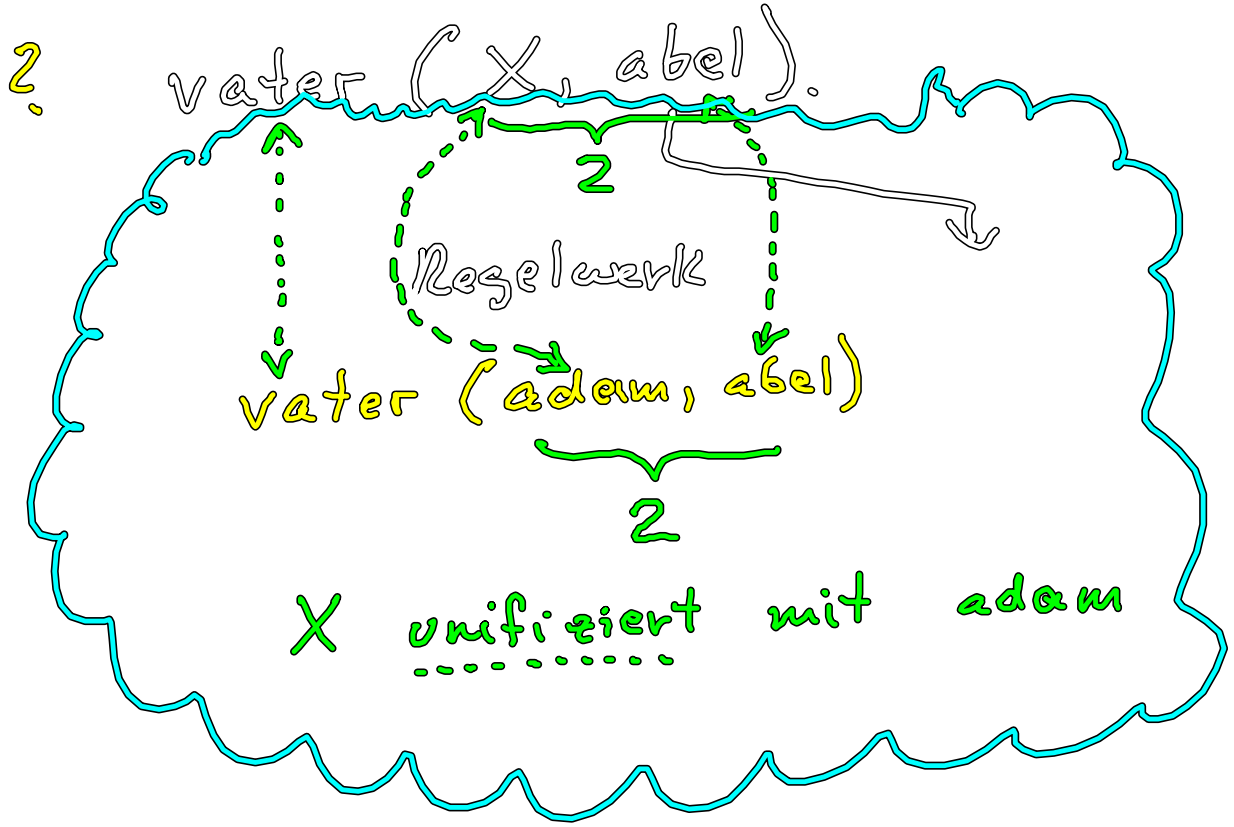


? vater (adam, abel).

yes

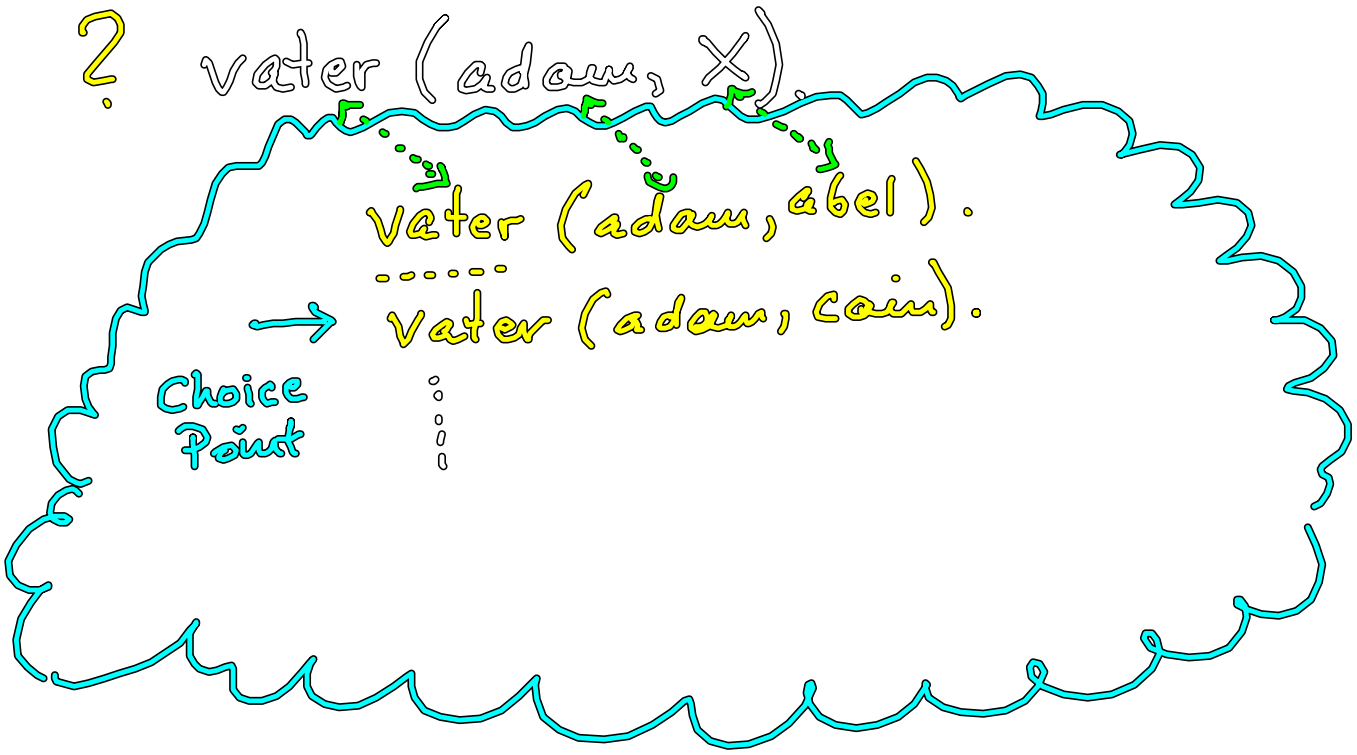
? mutter (eva, abel).

yes.


2.
vater (X, abel).
2
Regelwerk
vater (adam, abel)
2
X unifiziert mit adam

X = adam

yes


?
vater (adam, X)
vater (adam, abel).
vater (adam, cain).
Choice Point

X = abel ;
X = cain
yes

? opa(X, abel).

.

? vater(X, Y).

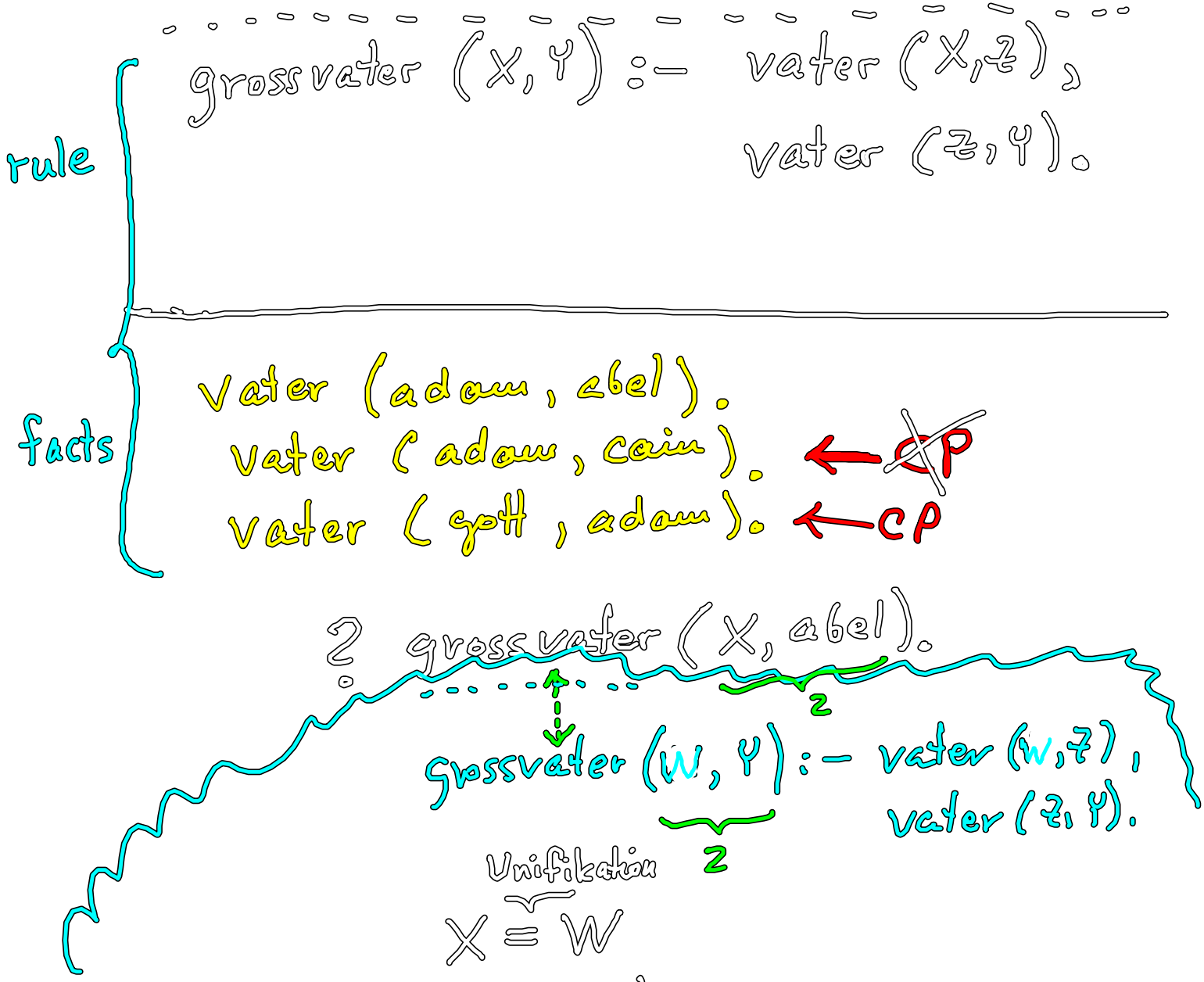
X = adam
Y = abel ; backtracking

X = adam
Y = cain

?

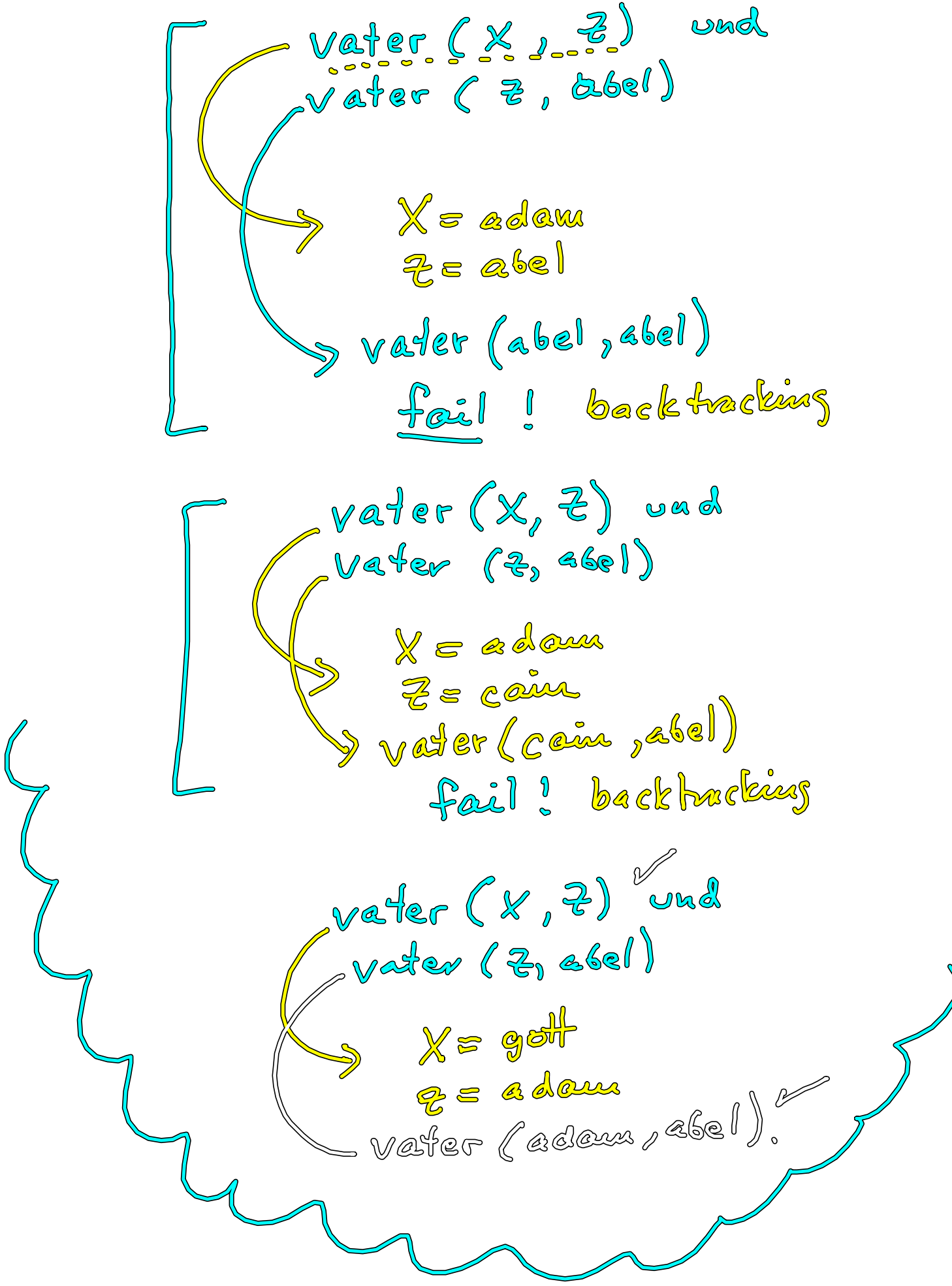
---

grossvater(X, Y) ← vater(X, Z) ∧
vater(Z, Y).

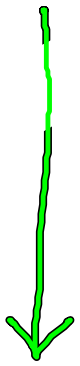


Goal: grossvater (X, abel)

↳ Zwei Subgoals

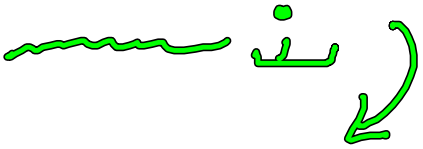

vater (X, Z) und
vater (Z, abel)
X = adam
Z = abel
vater (abel, abel)
fail ! backtracking
vater (X, Z) und
vater (Z, abel)
X = adam
Z = cain
vater (cain, abel)
fail ! backtracking
vater (X, Z) und
vater (Z, abel)
X = gott
Z = adam
vater (adam, abel).

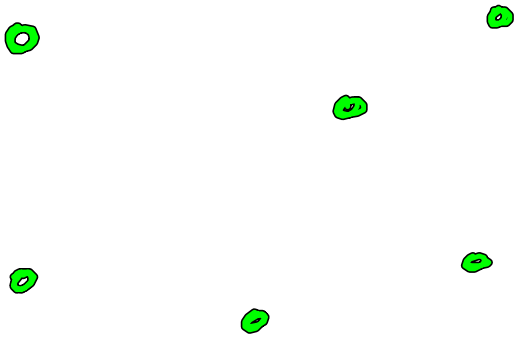
? X = sott.
yes
?



mutter (eva, abel).
vater (adam, cain).

? etwas.





1 → vater(...).

2 → vater( ).

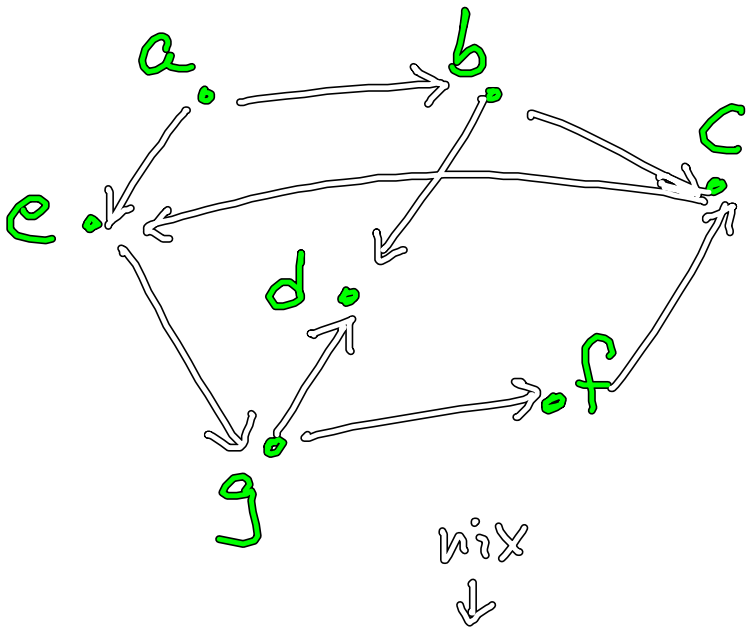
mutter( ).

grossvater(...).

mutter( )

3 → vater( ).

grossvater( ).



```
connection (a,e).
connection (e,g).
connection (g,d).
connection (g,f).
   ⋮
```

Klausel

```
weg(X,Y) :- connection(X,Y).
weg(X,Y) :- connection(X,Z),
            weg(Z,Y).
```

?- weg(a,c).

X = a
Y = c ⟶ connection(a,c)

fail!
backtracking

1. Versuch

2. Versuch { connection(a,Z), weg(Z,c)

Z = e

⋮

weg(e,c)

⋮

weg(g,c)

⋮

weg(f,c)

yes

$[1, 2, 3, 4]$

$[a, b, c, d]$

Hornformel

$$a \wedge b \wedge c \rightarrow d$$

$$\neg(a \wedge b \wedge c) \vee d$$

$$\neg a \vee \neg b \vee \neg c \vee d$$

$$a \wedge b \wedge c \rightarrow \neg d$$

$$\neg a \vee \neg b \vee \neg c \vee \neg d$$

$a \leftarrow \neg b, \neg c, d$

not(....)

? not (grossvater (gott, adam)).

Negation als Failure

$a \vee b \rightarrow c$

→ c :- a.
c :- b.

Kopf Körper