

5.5 Input and Output

Freitag, 19. Juni 2015 08:30

Up to now:

Input: only via queries

Output: only via answer substitutions

Now: extra-logical predicates for "real" input + output

Write / 1

• write(t)

- proof always succeeds

- side-effect: t is printed on the current output-stream (by default: screen)

?- X is $2+3$, write(X).

5 ← printed on the screen

$X=5$

?- write('Hello World').

Hello World

true

fd. symbol of arity 0

← write omits quotes in the output

Prog:

mult(X, Y):- Result is $X * Y$, write($X * Y$),
write(' = '), write(Result).

? - mult(3, 4).

$$3 * 4 = 12$$

true

mult(3, 4)

Res is 3 * 4, write (3 * 4)

write ('= '), write (Result)

write (3 * 4), write ('=')

write (12) !

Side-Effects cannot be undone when backtracking:

q(a).

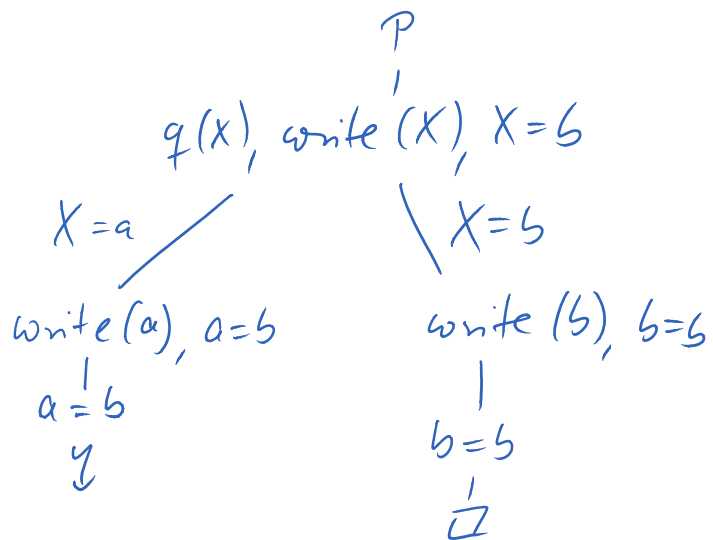
q(b).

P :- q(X), write(X), X = b.

? - P.

a b

true



nl/o

new line predicate

• always succeeds

• creates a new line in the output stream

? - write(a), nl, write(b), nl, write(c).

a

b

c

true

read/1

• read(t)

• reads a term s from the standard input stream (by default: keyboard)

End of term must be marked by .

• succeeds iff t and s unify

(can be used to check which input was given by the user)

Example: `sgv`

Input + Output can also be done with files.

→ change input/output stream.

see/1 and tell/1, seen/0 and told/0

see(t)

sets the input stream to the file t

told(t)

sets output stream to file t

tell(k)

sets output stream to file t

seen

told

} close the current i/o
stream and set it back to
the default

Sqr-example:

Input file should contain 3. -4.

If the end of file is reached, then read(X)
returns the answer $X = \text{end_of_file}$.

Afterwards, Output file contains

The square of 3 is 9

The square of -4 is 16