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

\[ \text{write} / 1 \]
- \[ \text{write}(t) \]
  - proof always succeeds
  - side-effect: it is printed on the current output-stream (by default: screen)

?- \( X \) is \( 2 + 3 \), write \((X)\).
5 \leftarrow \text{printed on the screen}
\( X = 5 \)

?- write(\text{'Hello World'}).
Hello World \leftarrow \text{write omits quotes in the output}

Prog:

\[ \text{mult}(X, Y) :- \text{Result} \text{ is } X \times Y, \text{write}(X \times Y), \]
\[ \text{write}(\text{'} = \text{'}), \text{write}(\text{Result}). \]
?- mult(3, 4).
3 * 4 = 12
two

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
two

\[ \begin{align*}
q(X), \text{write}(X), X &= 6 \\
X &= a \\
\text{write}(a), a &= b \\
a &= b \\
g \text{write}(b), b &= 6 \\
b &= 5 \\
\boxed{0}
\end{align*} \]

\underline{nl/0}

- new line predicate
  - always succeeds
  - creates a new line in the output stream
?- write(a), nl, write(b), nl, write(c).

a
b
c
tme

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: sqv

Input + Output can also be done with files.
  - change input/output stream.

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

see(t) sets the input stream to
the file t

tell(t) sets output stream to file t
tell(x)  sets output stream to file x
seen
fold   } 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 read, then read(X) returns the answer X = end of file.
Afterwards, Output file contains:
The square of 3 is 9
The square of -4 is 16