Prolog - proof search

Joanna Kołodziejczyk

20 listopada 2018

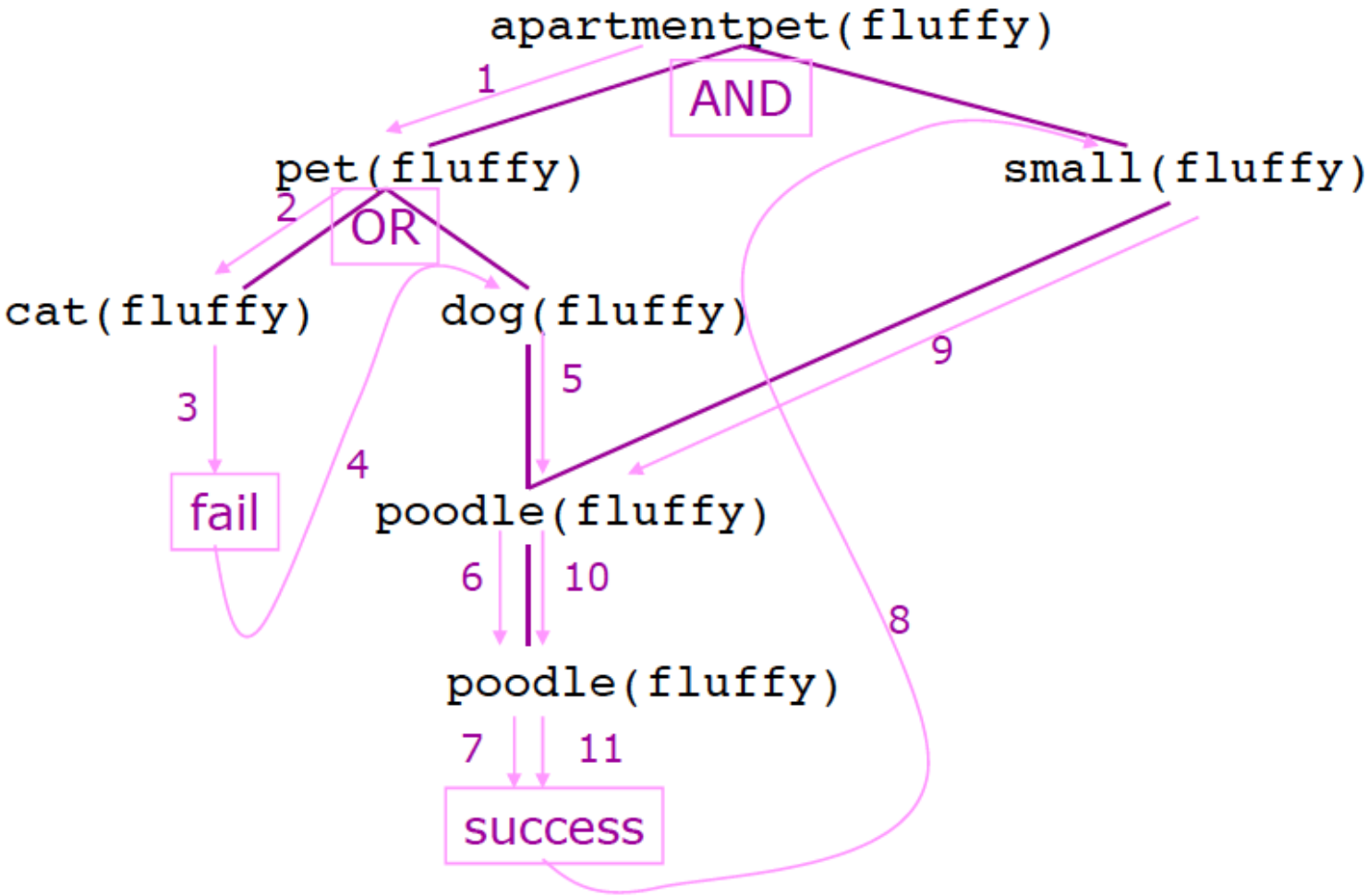
Proof search

Suppose we are working with the following knowledge base

```
apartmentpet(X) :- pet(X), small(X).
pet(X) :- cat(X).
pet(X) :- dog(X).
dog(X) :- poodle(X).
small(X) :- poodle(X).
poodle(fluffy).
```

Suppose we then pose the query `apartmentpet(fluffy).`

Graph representing the search for the prove



The order - how to find goal

- The interpreter behaviour is know and it uses DFS algorithm.
- The DFS order is important for recursive predicates (relations).
- Wrong order will lead to stack overflow and unfinished loop.

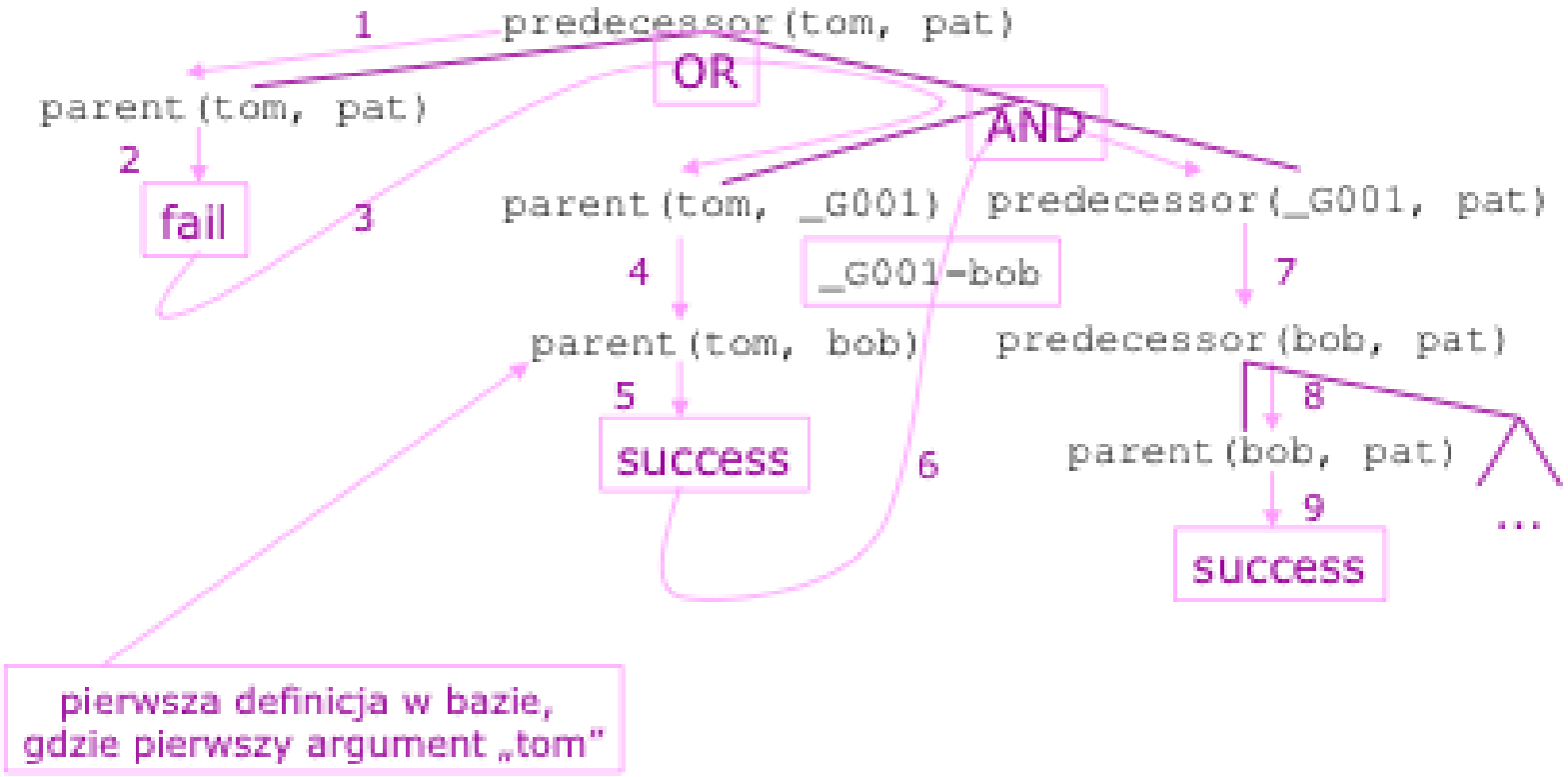
Recursion

Suppose there is a knowledge base

```
parent(pam, bob).
parent(tom, bob).
parent(tom, liz).
parent(bob, ann).
parent(bob, pat).
parent(pat, jim).
predecessor(X, Y):- parent(X, Y).
predecessor(X, Y):- parent(X, Z), predecessor(Z, Y).
```

Suppose we pose the query `predecessor(tom, pat).`.

The example analysis



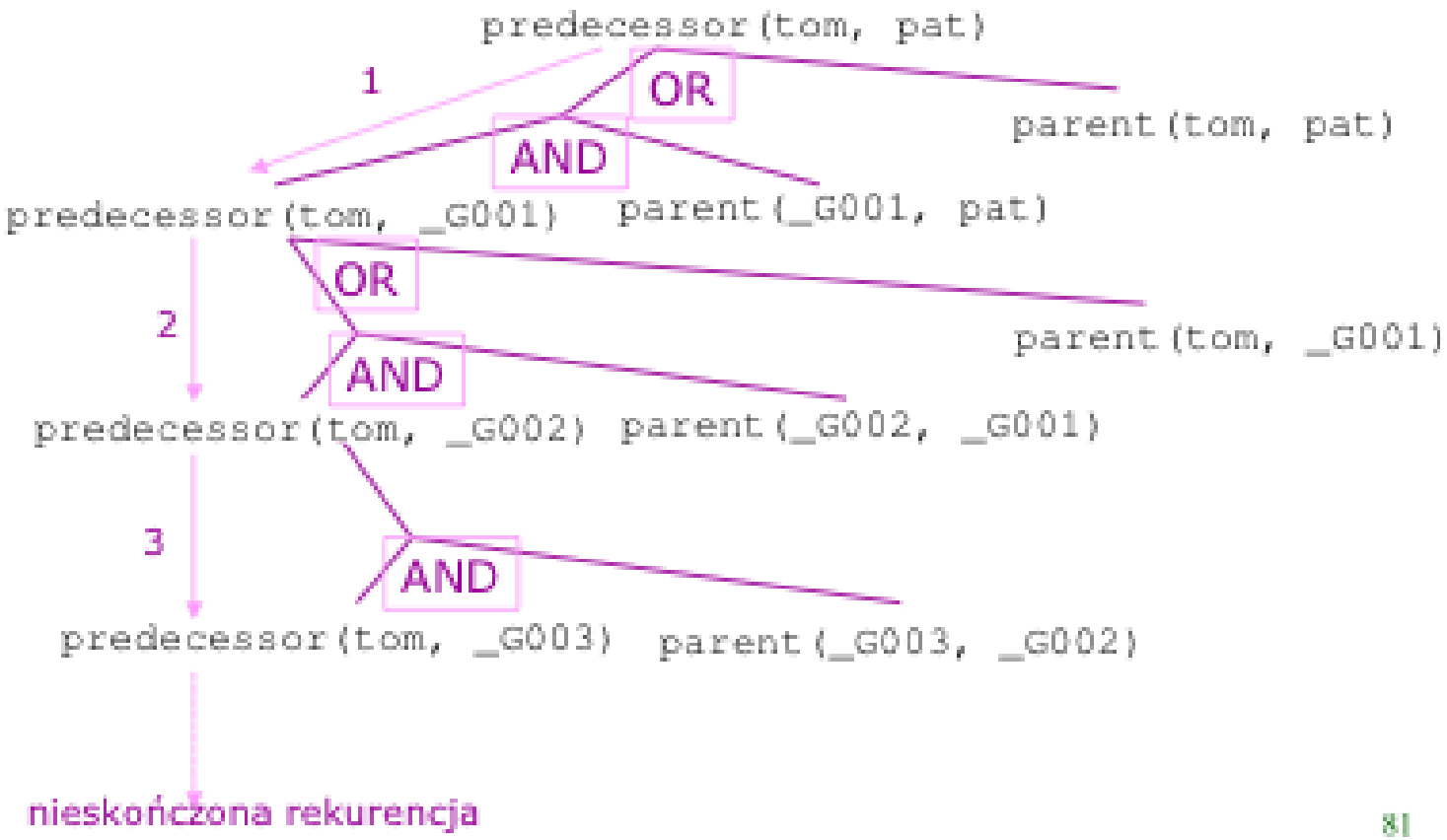
The example of wrong recursion

Suppose the code for the predecessor is changed to

```
predecessor(X, Y):- predecessor(X, Z), parent(Z, Y).
predecessor(X, Y):- parent(X, Y).
```

Suppose we pose the query `predecessor(tom, pat).`.

Recursion that fail



81