

# Jeszcze o rekursji i indukcji

19 maja 2009

# Plan

- Functional Scheme i functional induction
- Polecenie Function
- „Język” Russell i polecenie Program

## Schemat wg definicji funkcji

- polecenie `Functional Scheme` — wygenerowanie schematu indukcji  
`Functional Scheme ident1 := Induction for ident'1 Sort sort1`  
`with`  
`...`  
`with identm := Induction for ident'm Sort sortm`
- taktyka `functional induction (ident' x1 ... xn)` — użycie wygenerowanego schematu indukcji `ident_ind` :)

## Wspomaganie definiowania funkcji

- 1 Function  $ident\ binder_1 \dots binder_n : type_0 := term_0$   
Defines the not recursive function  $ident$  as if declared with Definition. Moreover the following are defined:
  - $ident\_rect$ ,  $ident\_rec$  and  $ident\_ind$ , which reflect the pattern matching structure of  $term_0$
  - The inductive  $R\_ident$  corresponding to the graph of  $ident$  (silently);
  - $ident\_complete$  and  $ident\_correct$  which are inversion information linking the function and its graph.
- 2 Function  $ident\ binder_1 \dots binder_n \{struct\ ident_0\} : type_0 := term_0$   
Defines the structural recursive function  $ident$  as if declared with Fixpoint. Moreover the following are defined:
  - The same objects as above;
  - The fixpoint equation of  $ident$ :  $ident\_equation$ .

# Polecnie Function (c.d)

- 3 Function  $ident\ binder_1 \dots binder_n\ \{\text{measure } f\ ident_0\} : type_0 := term_0$
- 4 Function  $ident\ binder_1 \dots binder_n\ \{\text{wf } R\ ident_0\} : type_0 := term_0$

## Require Recdef.

The  $\{\}$  annotation must be one of the following:

- $\{\text{measure } f\ ident_0\}$  with  $ident_0$  being the decreasing argument and  $f : T_{ident_0} \rightarrow \text{nat}$  such that for each recursive call we have  $\text{lt } (f\ i')\ (f\ i)$ .
- $\{\text{wf } R\ ident_0\}$  with  $ident_0$  being the decreasing argument and  $R : T_{ident_0} \rightarrow T_{ident_0} \rightarrow \text{Prop}$  such that for each recursive call we have  $R\ i'\ i$   
 $R$  must be well-founded.

Once proof obligations are discharged, following objects are defined:

- The same objects as with the `struct`;
- The lemma  $ident\_tcc$  which collects all proof obligations;
- The lemmas  $ident\_terminate$  and  $ident\_F$ .

Wsparcie dla typów postaci  $\{n:\text{nat} \mid P\ n\}$

- definiujemy funkcję (również rekurencyjną) tak jakby jej wynikiem był `nat`
- rozwiązujemy obligacje dowodowe, korzystając z założenia indukcyjnego