

W przód i w tył...

24 marca 2009

## Typy indukcyjne (c.d.)

- zaawansowana destrukcja — `inversion`
- jeden krok obliczeń — `simpl`
- definicja równości — `eq`
- dowodzenie i używanie równości — `reflexivity`, `rewrite`

## Dowodzenie „w tył” (*goal-oriented*)

- `apply...`
- `constructor`

## Dowodzenie „w przód” (ukierunkowywanie wiedzy)

- `assert`, `cut`
- `generalize`
- `pose`, `set`
- `remember`

Jak to się mogło stać ?

- `inversion`  
(wnioski z (Even n) oraz discriminate i injection)
- `inversion_clear`  
(j.w. ale trochę czyści)
- `dependent inversion`  
(jeśli dane założenie występuje w „celu”)

## Typ indukcyjny: równość

- równość Leibniza `eq`
- dowodzenie: `reflexivity` lub `split` lub `constructor` lub `apply`  
`eq_refl`
- dowodzenie: `symmetry`, `transitivity`
- używanie: `rewrite`, `rewrite <-`, `subst`, `replace`

## Typy indukcyjne (c.d.)

- zaawansowana destrukcja — `inversion`
- jeden krok obliczeń — `simpl`
- definicja równości — `eq`
- dowodzenie i używanie równości — `reflexivity`, `rewrite`

## Dowodzenie „w tył” (*goal-oriented*)

- `apply...`
- `constructor`

## Dowodzenie „w przód” (ukierunkowywanie wiedzy)

- `assert`, `cut`
- `generalize`
- `pose`, `set`
- `remember`

# Dowodzenie „w tył” i „w przód”

- Dowodzenie „w tył” (goal-oriented)

Mamy udowodnić *cel*. Używamy taktyki *tac*, która dowiodłaby *cel* (gdyby spełnione były założenia).

Potem musimy udowodnić prawdziwość założeń.

**Przykład:** `apply`

Dane:  $H : A \rightarrow B$

Cel:  $B$

`apply H.`

Nowy Cel:  $A$

# Dowodzenie „w tył” i „w przód”

- Dowodzenie „w przód” (ukierunkowywanie wiedzy)

Mamy jakąś wiedzę, mamy udowodnić (odległy) *cel*. Z naszej wiedzy potrafimy udowodnić *fakt*, z którego będzie łatwiej udowodnić *cel*.

**Przykład:** assert

Dane H1: forall n, Strange n -> Even n, H2 : Strange k

Cel: divides 2 n

assert (Even k).

Nowe cele:

1) Dane: bez zmian

Cel: Even k

2) Dane: H1: ... H2: ... H3: Even k

Cel: jak poprzednio

# Taktyki „w tył”

Taktyka *apply term*.

Term ma typ postaci:  $\overrightarrow{\forall x : T}, \text{coś}$

- używając unifikacji wyższego rzędu uzyskuje odpowiednią instancję *coś*, wyznaczając podstawienie na zmienne, od których zależy *coś*
- poprzedniki nienazwane stają się nowymi celami
- poprzedniki nazwane, ale nie występujące w *celu* trzeba podać *apply term* with  $term_1 \dots term_n$   
lub  
*apply term* with  $(y := term_1) \dots$

Jest też odmiana *apply*, w której nie trzeba nic podawać: *eapply*.  
Wprowadza ona *zmiennie egzystencjalne*. Omówimy je później.

# Taktyki „w przód”

- $\text{cut } A$  — zamienia  $cel$  na dwa cele:  $A \rightarrow cel$  oraz  $A$ .
- $\text{assert } A$  — dodatnie założenia (niestety również do udowodnienia), zamienia  $cel$  na dwa cele:  $A$  oraz  $cel$  przy założeniu  $A$ .
- warianty  $\text{assert}$ :  
 $\text{assert } ( ident [ : typ ] [ := term ] )$   
 $\text{assert typ as intro\_pattern}$
- $\text{pose proof term}$  — dodaj „założenie”, którego dowodem jest  $term$
- odmiany związane z konwersją:  
 $\text{set } (ident:=term)$   
— dodaje lokalną deklarację i zastępuje  $term$  przez  $ident$   
 $\text{pose } (ident:=term)$   
— dodaje lokalną deklarację i **nie** zastępuje  $term$  przez  $ident$
- „wyciąganie” równości:  
 $\text{remember term as ident}$   
— zastępuje  $term$  przez  $ident$  w  $cel$  i zamienia go na  $ident=term \rightarrow cel'$

## Inne taktyki „w tył”

- `generalize term` — zastępuje `term` w `celu` przez nową zmienną `x` i zamienia `cel` na `forall x, cel'`  
UWAGA: jeśli `term` w ogóle nie występuje w `celu`, `generalize term` zmieni `cel` na `typ -> cel`, gdzie `typ` jest typem `termu`.
- odmiany:  
`generalize ( term [at num1...numk] [as ident] )+`  
`at...` — tylko niektóre wystąpienia,  
`as...` — określona nazwa zmiennej związanej,  
więcej `termów` — po kolei **od prawej**  
`generalize dependent term` — również hipotezy, w których występuje `term`, po czym je usuwa
- inne, podobne:  
`revert id` — jakby odwrotność `intro`  
`specialize (H t1 ... tn)`  
`specialize H with ...` —