

DA MATITA A DEDUKTI E RITORNO

SOTTOTITOLO CHE NON SO ANCORA COSA

Mattia Girolimetto

Relazione per il corso 85001 - Metodi logici per la Filosofia
Alma Mater Studiorum, Università di Bologna

13 luglio 2023

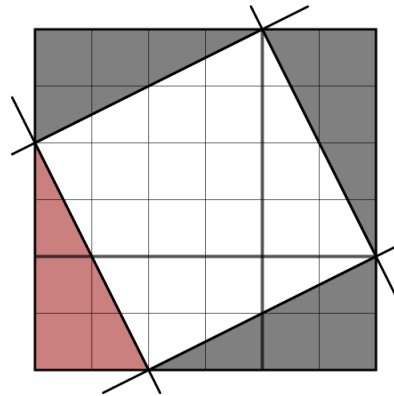
INDICE

1	I proof assistant	3
2	Dedukti	5
3	Matita	6
4	Krajono	7
5	Esportazione	8
6	Importazione	9
7	Importazione: problemi	10
8	Conclusioni	14

I PROOF ASSISTANT



matita



ISOMORFISMO DI CURRY-HOWARD

LOGICA

Proposizioni

Dimostrazioni

Verifica di una dimostrazione

TEORIA DEI TIPI

Tipi

Programmi

Verifica di tipo

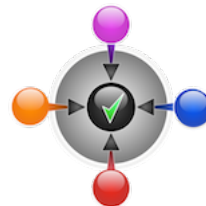
\Leftrightarrow

\Leftrightarrow

\Leftrightarrow

DEDUKTI

- Framework logico
- Implementa logiche e teoremi
- Basato sul $\lambda\Pi$ -calcolo modulo

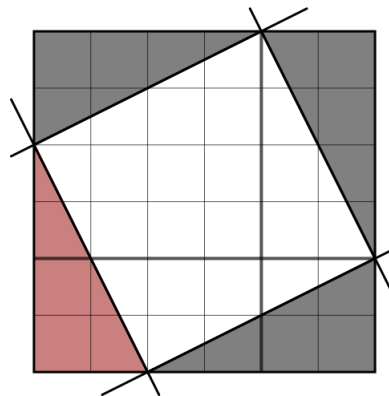


dedukti

MATITA

- ▶ Proof assistant sviluppato all'Università di Bologna
- ▶ Basato sul calcolo delle *costruzioni* *(co)induttive*

matita



KRAJONO

- ▶ *Fork* di Matita
- ▶ Esportazione verso Dedukti
- ▶ Non più supportato



ESPORTAZIONE

```
let rec plus n m on n &  
  match n with  
  [ 0 => m  
  | S x => S (plus x m) ].
```



```
def plus :  
  cic.Term (cic.type cic.z)  
  (cic.prod (cic.type cic.z) (cic.type cic.z)  
   matita_esperimento_other.nat  
   (___ : cic.Term (cic.type cic.z) matita_esperimento_other.nat =>  
    cic.prod (cic.type cic.z) (cic.type cic.z)  
    matita_esperimento_other.nat  
    (___1 :  
     cic.Term (cic.type cic.z) matita_esperimento_other.nat =>  
     matita_esperimento_other.nat))))).  
  
def plus_body :  
  cic.Term (cic.type cic.z)  
  (cic.prod (cic.type cic.z) (cic.type cic.z)  
   matita_esperimento_other.nat  
   (___ : cic.Term (cic.type cic.z) matita_esperimento_other.nat =>  
    cic.prod (cic.type cic.z) (cic.type cic.z)  
    matita_esperimento_other.nat  
    (___1 :  
     cic.Term (cic.type cic.z) matita_esperimento_other.nat =>  
     matita_esperimento_other.nat))))).
```


IMPORTAZIONE

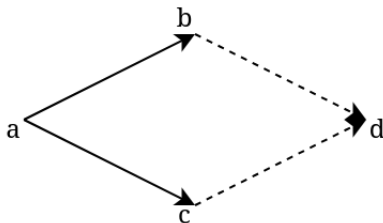
IMPORTAZIONE: PROBLEMI

- **Problema** In Dedukti le proprietà di *confluenza* e *normalizzazione* non sono garantite.

CONFLUENZA E NORMALIZZAZIONE

Definizione (Confluenza)

Dato un termine a , se esistono due regole di riscrittura $a \hookrightarrow^* b$ e $a \hookrightarrow^* c$, allora esiste un termine d tale che $b \hookrightarrow^* d$ e $c \hookrightarrow^* d$.



Definizione (Normalizzazione)

Dato un termine, questo può essere ridotto al più un numero finito di volte.

IMPORTAZIONE: PROBLEMI

- ▶ **Problema** In Dedukti le proprietà di *confluenza* e *normalizzazione* non sono garantite.
- ▶ **Problema** Durante la codifica vengono perse informazioni necessarie alla ricostruzione dei termini originali.

PRAGMA

Per preservare tali informazioni vengono usate le *pragma*:

```
1 #PRAMGA BEGIN INDUCTIVE NAME=nat LEFTNO=0 CONS:nat=O CONS:nat=S.
2
3 nat : cic.Univ (cic.type cic.z).
4
5 O : cic.Term (cic.type cic.z) matita_test_nat.nat.
6
7 S : cic.Term (cic.type cic.z) (cic.prod (cic.type cic.z) (cic.type cic.z)
8     matita_test_nat.nat(__ : cic.Term (cic.type cic.z)
9     matita_esperimento_nat.nat => matita_test_nat.nat)).
10
11 #PRAMGA END INDUCTIVE.
12
```

LINK UTILI

- ▶ **Matita**: <https://github.com/sacerdot/matita>
- ▶ **Dedukti**: <https://deducteam.github.io/>
- ▶ **Krajono**: <https://github.com/Deducteam/Krajono>