Da Matita a Dedukti e ritorno

sottotitolo che non so ancora cosa

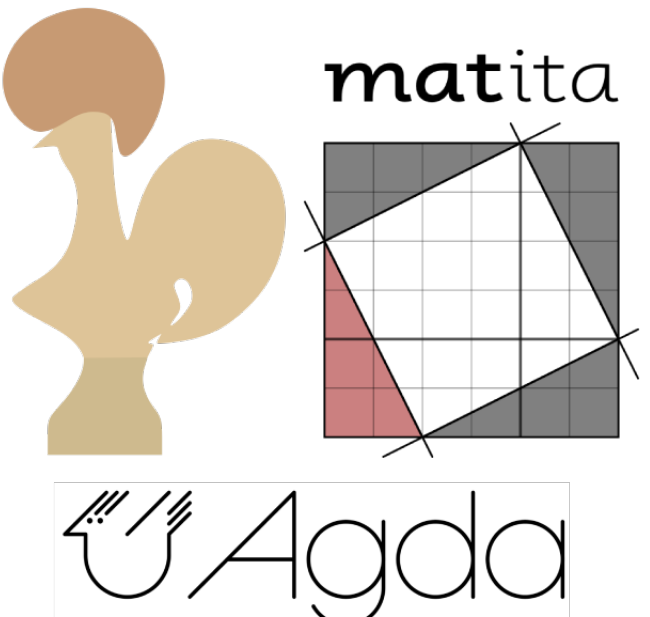
Mattia Girolimetto

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

14 luglio 2023

Indice

I PROOF ASSISTANT

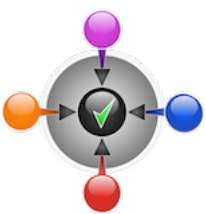


Isomorfismo di Curry-Howard

LOGICA		TEORIA DEI TIPI
Proposizioni	$\Leftrightarrow$	Tipi
Dimostrazioni	$\Leftrightarrow$	Programmi
Verifica di una dimostrazione	$\Leftrightarrow$	Verifica di tipo

Dedukti

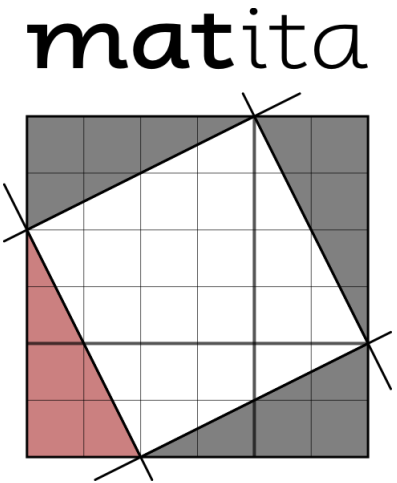
- Framework logico
- Implementa logiche e teoremi
- Basato sul $\lambda\Pi$-calcolo modulo



dedukti

MATITA

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



Krajono

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

La codifica

```
let rec plus n m on n :=
 match n with
 [ O ⇒ 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))).
```

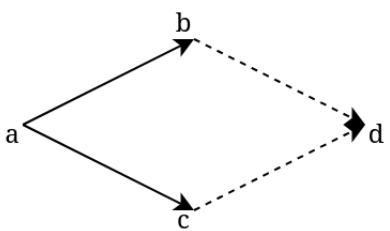
Importazione

Problemi

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

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.

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*:

```
#PRAMGA BEGIN INDUCTIVE NAME=nat LEFTNO=0 CONS:nat=O CONS:nat=S.

nat : cic.Univ (cic.type cic.z).

O : cic.Term (cic.type cic.z) matita_test_nat.nat.

S : cic.Term (cic.type cic.z) (cic.prod (cic.type cic.z) (cic.type cic.z)
      matita_test_nat.nat(__ : cic.Term (cic.type cic.z)
        matita_esperimento_nat.nat => matita_test_nat.nat)).

#PRAMGA END INDUCTIVE.

```

Conclusioni

Link utili

- Matita: https://github.com/sacerdot/matita
- Dedukti: https://deducteam.github.io/
- Krajono: https://github.com/Deducteam/Krajono