A Formal Verification of Reversible Primitive Permutations

Giacomo Maletto

Dipartimento di Matematica
Università di Torino

Tesi di Laurea Triennale, Ottobre 2021

- **Proof assistant**: software che aiuta nello sviluppo di dimostrazioni formali

- **Proof assistant**: software che aiuta nello sviluppo di dimostrazioni formali

- **NON** coincidono con gli automatic theorem prover

Un traguardo recente

Un traguardo recente

Liquid Tensor Experiment (Peter Scholze, 2021)

Strumenti

- Un proof assistant: Lean Theorem Prover (Microsoft Research, 2013)

Strumenti

- Un proof assistant: Lean Theorem Prover (Microsoft Research, 2013)
- Una libreria digitalizzata di matematica: Mathlib (2017)

Soggetto

Soggetto

- *A class of Recursive Permutations which is Primitive Recursive complete*
  Paolini, Piccolo, Roversi, Theoretical Computer Science (2020)

Soggetto

- *A class of Recursive Permutations which is Primitive Recursive complete*
  Paolini, Piccolo, Roversi, Theoretical Computer Science (2020)

- Computazione reversibile

input | programma | output

RPP

- **Reversible Primitive Permutations** (RPP): una classe di funzioni $\mathbb{Z}^n \to \mathbb{Z}^n$ **calcolabili**, **reversibili** che è PRF-**completa**

RPP

Appartengono a RPP:

RPP

Appartengono a RPP:

- L'**identità** n-aria

$$\begin{matrix} x_1 \\ \vdots \\ x_n \end{matrix} \; \boxed{\mathsf{Id}_n} \; \begin{matrix} x_1 \\ \vdots \\ x_n \end{matrix}$$

- La funzione **negazione**

$$x \; \boxed{\mathsf{Ne}} \; -x$$

- La funzione **successore** e **predecessore**

$$x \; \boxed{\mathsf{Su}} \; x+1 \qquad\qquad x \; \boxed{\mathsf{Pr}} \; x-1$$

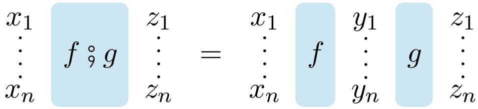
- Lo **swap**

$$\begin{matrix} x \\ y \end{matrix} \; \boxed{\mathsf{Sw}} \; \begin{matrix} y \\ x \end{matrix}$$

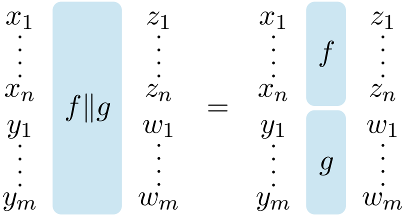
RPP

Appartengono a RPP:

- La **composizione in serie** di due $f, g \in \text{RPP}$



- La **composizione parallela** di due $f, g \in \text{RPP}$



RPP

Appartengono a RPP:

- L'**iterazione finita** di una $f \in \mathrm{RPP}$

$$\begin{array}{c} x \\ x_1 \\ \vdots \\ x_n \end{array} \boxed{\mathrm{It}[f]} \begin{array}{c} x \\ y_1 \\ \vdots \\ y_n \end{array} = \begin{array}{c} x \\ x_1 \\ \vdots \\ x_n \end{array} \underbrace{\boxed{f} \; \cdots \; \boxed{f}}_{x \text{ volte (se } x > 0)} \begin{array}{c} x \\ y_1 \\ \vdots \\ y_n \end{array}$$

- La **selezione** di tre $f, g, h \in \mathrm{RPP}$

$$\begin{array}{c} x \\ x_1 \\ \vdots \\ x_n \end{array} \boxed{\mathrm{If}[f, g, h]} \left.\begin{array}{c} x \\ y_1 \\ \vdots \\ y_n \end{array}\right\} = \begin{cases} f(x_1, \ldots, x_n) & \text{if } x > 0 \\ g(x_1, \ldots, x_n) & \text{if } x = 0 \\ h(x_1, \ldots, x_n) & \text{if } x < 0 \end{cases}$$

Proprietà delle RPP

- calcolabili
- reversibili
- PRF-completa

Proprietà delle RPP

- calcolabili
- reversibili
- PRF-completa

Possibile arrivare al risultato tramite un processo meccanizzabile finito

Proprietà delle RPP

- calcolabili
- reversibili
- PRF-completa

Proprietà delle RPP

- calcolabili
- reversibili
- PRF-completa

Ogni funzione ammette inversa:

- $\mathsf{Id}_n^{-1} = \mathsf{Id}_n$
- $\mathsf{Ne}^{-1} = \mathsf{Ne}$
- $\mathsf{Su}^{-1} = \mathsf{Pr}$
- $\mathsf{Pr}^{-1} = \mathsf{Su}$
- $\mathsf{Sw}^{-1} = \mathsf{Sw}$
- $(f \fatsemi g)^{-1} = g^{-1} \fatsemi f^{-1}$
- $(f \| g)^{-1} = f^{-1} \| g^{-1}$
- $\mathsf{It}[f]^{-1} = \mathsf{It}[f^{-1}]$
- $\mathsf{If}[f, g, h]^{-1} = \mathsf{If}[f^{-1}, g^{-1}, h^{-1}]$

Proprietà delle RPP

- calcolabili
- reversibili
- PRF-completa

Per esempio, $\left(\mathsf{Sw} \,;\, (\mathsf{Ne}\|\mathsf{Su})\right)^{-1} = (\mathsf{Ne}\|\mathsf{Pr}) \,;\, \mathsf{Sw}$:

$$
\begin{array}{cccccc}
x & \boxed{\mathsf{Sw}} & y & \boxed{\mathsf{Ne}} & -y \\
y & & x & \boxed{\mathsf{Su}} & x+1
\end{array}
$$

Proprietà delle RPP

- calcolabili
- **reversibili**
- PRF-completa

Per esempio, $\big(\mathsf{Sw} \fatsemi (\mathsf{Ne} \| \mathsf{Su})\big)^{-1} = (\mathsf{Ne} \| \mathsf{Pr}) \fatsemi \mathsf{Sw}$:

$$
\begin{array}{ccccc}
x & \multirow{2}{*}{$\boxed{\mathsf{Sw}}$} & y & \boxed{\mathsf{Ne}} & -y \\
y & & x & \boxed{\mathsf{Su}} & x+1
\end{array}
$$

$$
\begin{array}{ccccc}
-y & \boxed{\mathsf{Ne}} & y & \multirow{2}{*}{$\boxed{\mathsf{Sw}}$} & x \\
x+1 & \boxed{\mathsf{Pr}} & x & & y
\end{array}
$$

Proprietà delle RPP

- calcolabili
- reversibili
- PRF-completa

Proprietà delle RPP

- calcolabili
- reversibili
- PRF-completa

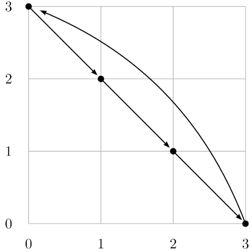
Sia $F \in \mathrm{PRF}$.
Allora, esiste $g \in \mathrm{RPP}$ che **codifica** F:

$$\begin{array}{c|c|c} z & & z + F(\boldsymbol{x}) \\ \boldsymbol{x} & g & \boldsymbol{x} \\ \mathbf{0} & & \mathbf{0} \end{array}$$

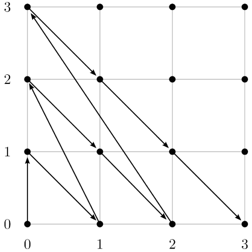
Formalizzazione in Lean

Formalizzazione in Lean

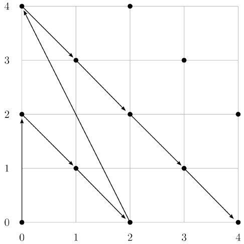
Divisione euclidea



Cantor pairing



Radice quadrata



Formalizzazione in Lean

Principali teoremi:

Formalizzazione in Lean

Principali teoremi:

- Ogni RPP è invertibile:

```
theorem inv_iff (f : RPP) (X Y : list ℤ) :
  ‹f⁻¹› X = Y ↔ ‹f› Y = X
```

Formalizzazione in Lean

Principali teoremi:

- Ogni RPP è invertibile:

```
theorem inv_iff (f : RPP) (X Y : list ℤ) :
  <f⁻¹> X = Y ↔ <f> Y = X
```

- PRF-completezza:

```
theorem completeness (F : ℕ → ℕ) :
  nat.primrec F → ∃ f : RPP, encode F f
```