UNIVERSITÀ DEGLI STUDI "ROMA TRE"
IN410 - CALCOLABILITÀ E COMPLESSITÀ
A.A. 2018-2019
PROF. M. PEDICINI

SOLUZIONI DELLA SECONDA PROVA IN ITINERE DEL 7/01/2019 (DURATA 3H00)

Esercizio 1. *Sia t il λ-termine*

$$t = \lambda x((\underline{2})(\lambda y\,((y)x)x)\underline{1})x$$

dove gli interi sottolineati indicano i termini numerali di Church:

$$\underline{n} = \lambda f \lambda x \underbrace{(f)\dots(f)}_{n-volte} x$$

(1) Mostrare che

$$(t)I \simeq_\beta I$$

dove $I = \lambda xx$.

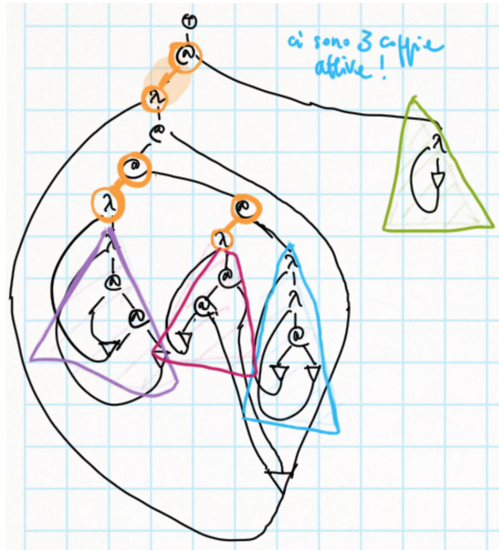
Soluzione. Per riduzione mostriamo che $(t)I \to_\beta I$ e pertanto la tesi.

$$\begin{aligned}
(\lambda x((\underline{2})(\lambda y\,((y)x)x)\underline{1})x)I &\to_\beta ((\underline{2})(\lambda y\,((y)I)I)\underline{1})I \\
&\to_\beta ((\underline{2})((\underline{1})I)I)I \\
&= ((\underline{2})((\lambda f\lambda x(f)x)I)I)I \\
&\to_\beta ((\underline{2})(\lambda x(I)x)I)I \\
&\to_\beta ((\underline{2})(I)I)I \\
&\to_\beta ((\underline{2})I)I \\
&= ((\lambda f\lambda x(f)(f)x)I)I \\
&\to_\beta (\lambda x(I)(I)x)I \\
&\to_\beta (I)(I)I \\
&= (\lambda xx)(I)I \\
&\to_\beta (I)I \\
&\to_\beta I
\end{aligned}$$

□

(2) rappresentare $(t)I$ come grafo di condivisione;

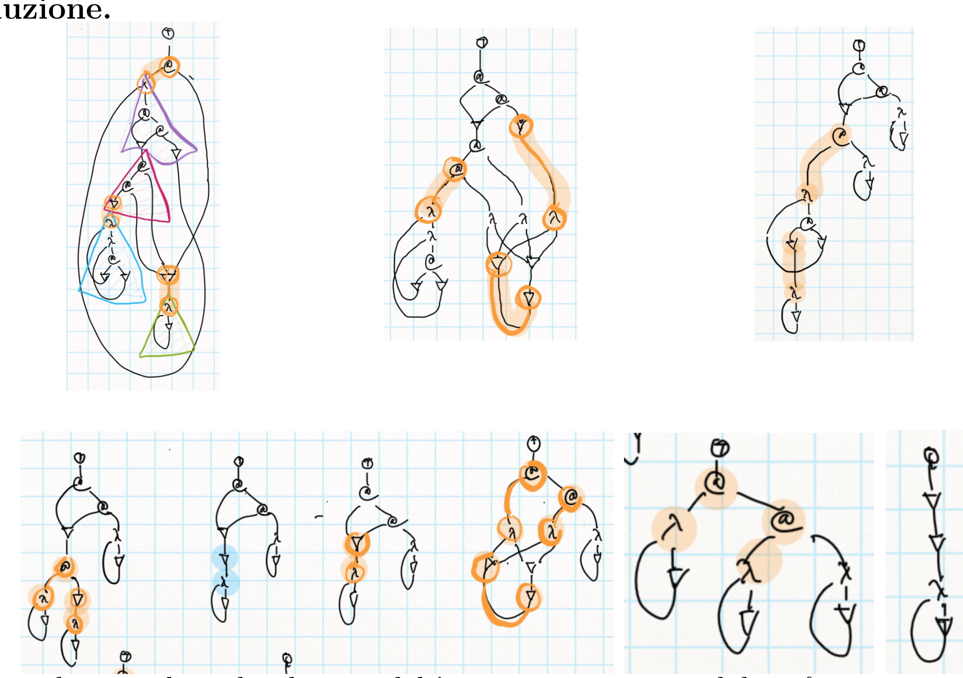
Soluzione.



□

(3) ridurre il grafo alla forma normale;

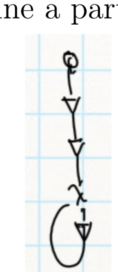
Soluzione.



□

(4) mostrare la procedura di rilettura del λ-termine a partire dal grafo ottenuto.

Soluzione. La rilettura del termine a partire dal grafo



Si effettua attraversando i nodi secondo le regole a partire dal nodo radice: i questo caso solo il nodo lambda che la prima volta che viene incontrato 'emette' un λx le volte che viene visitato da arco corrispondente al parametro viene scritta un'occorrenza della variabile e quindi dal terzo passaggio in poi non si tiene conto della variabile (risalendo dal corpo del termine non si emette niente). Dunque nel nostro caso il termine riletto è proprio l'identità $\lambda x\, x$. □

Esercizio 2. *(1) Data una funzione $f : \mathbb{N} \to \mathbb{N}$, ricordare cosa si intende che f è costruibile in tempo.*

Soluzione. Una funzione $f : \mathbb{N} \to \mathbb{N}$ si dice che è *costruibile in tempo* se esiste una macchina di Turing a k nastri μ di alfabeto A tale che $\{0, 1\} \subset A$ e per ogni input w, il tempo di arresto $T_\mu(w) = f(|w|)$.

Si noti che nella definizione si fa riferimento esplicito al fatto che il modello è quello multi-nastro. Questo elimina molto overhead nella copia e nei risultati di simulazione.

Ricordiamo che se una funzione è costruibile in tempo allora è ricorsiva; la dimostrazione si basa sul fatto che la macchina μ che esiste per definizione e termina in esattamente $f(|w|)$ passi può essere modificata per contare il numero di passi (su un nastro aggiuntivo) e dare in output il valore del contatore.

□

(2) Mostrare che se f è costruibile in tempo allora anche la funzione $g(x) = f(x)/2$ è costruibile in tempo.

Soluzione. Se f è costruibile in tempo esiste una macchina μ il cui tempo di arresto su ogni input binario di lunghezza n è esattamente $f(n)$. Grazie al teorema di speedup è possibile (espandendo l'alfabeto) accelerare di un fattore c qualsiasi la computazione, pertanto preso $c = 1/2$ avremo che la macchina μ' – costruita in modo analogo a quella della dimostrazione del teorema per simulare la computazione della macchina μ in modo accelerato – eseguirà il calcolo in tempo $cf(n) = g(n)$.

□

Esercizio 3. *Si consideri la rappresentazione[1] di sequenze di bit (di lunghezza qualunque) nel λ-calcolo nel modo seguente:*

$$\begin{aligned}
[] &= \lambda x_0 \lambda x_1 \lambda z z \\
[b] &= \lambda x_0 \lambda x_1 \lambda z (z) x_b \\
[b_1 b_2] &= \lambda x_0 \lambda x_1 \lambda z ((z) x_{b_1}) x_{b_2} \\
&\vdots \\
[b_1 \ldots b_n] &= \lambda x_0 \lambda x_1 \lambda z (\ldots ((z) x_{b_1}) x_{b_2} \ldots) x_{b_n}
\end{aligned}$$

così ad esempio la sequenza

$$[01101] = \lambda x_0 \lambda x_1 \lambda z (((((z) x_0) x_1) x_1) x_0) x_1$$

(1) Definire una rappresentazione nel λ-calcolo per la funzione complemento C (che complementa una sequenza: ad esempio $(C)[01101] \to_\beta [10010]$);

Soluzione. Definiamo $C = \lambda w \lambda a \lambda b \lambda c (((w) b) a) c$.

[1] in questo esercizio, per rappresentazione NON intendiamo rappresentazione forte

Infatti

$$
\begin{aligned}
(C)[010] &\to_\beta \lambda a\lambda b\lambda c((([010])b)a)c \\
&\to_\beta \lambda a\lambda b\lambda c(((\lambda x_0\lambda x_1\lambda z(((z)x_0)x_1)x_0)b)a)c \\
&\to_\beta \lambda a\lambda b\lambda c((\lambda z(((z)b)a)b)c \\
&\to_\beta \lambda a\lambda b\lambda c(((c)b)a)b = [101].
\end{aligned}
$$

□

(2) *Definire una rappresentazione nel λ-calcolo per l'operazione J di concatenazione di due sequenze, ad esempio $((J)[01101])[1010] \to_\beta [011011010]$;*
Soluzione. Definiamo $J = \lambda w_1\lambda w_2\lambda a\lambda b\lambda c(((w_2)a)b)(((w_1)a)b)c$.
Infatti

$$
\begin{aligned}
(J)[010] &\to_\beta \lambda a\lambda b\lambda c((([010])b)a)c \\
&\to_\beta \lambda a\lambda b\lambda c(((\lambda x_0\lambda x_1\lambda z(((z)x_0)x_1)x_0)b)a)c \\
&\to_\beta \lambda a\lambda b\lambda c((\lambda z(((z)b)a)b)c \\
&\to_\beta \lambda a\lambda b\lambda c(((c)b)a)b = [101].
\end{aligned}
$$

□

(3) *Definire una rappresentazione nel λ-calcolo per le operazioni A_b con $b \in \{0,1\}$ che aggiungono b alla sequenza, ad esempio $(A_0)[01101] \to_\beta [011010]$ e $(A_1)[01101] \to_\beta [011011]$);*
Soluzione. Utilizzando il termine J dell'esercizio precedente è possibile definire

$$A_0 = \lambda w((J)w)[0]$$

e

$$A_1 = \lambda w((J)w)[1].$$

In alternativa, definiamo esplicitamente

$$A_0 = \lambda w\lambda a\lambda b\lambda c((((w)a)b)c)a$$

e

$$A_1 = \lambda w\lambda a\lambda b\lambda c((((w)a)b)c)b.$$

Infatti

$$
\begin{aligned}
(A_0)[010] &\to_\beta \lambda a\lambda b\lambda c(((([010])a)b)c)a \\
&\to_\beta \lambda a\lambda b\lambda c(((\lambda x_0\lambda x_1\lambda z((((z)x_0)x_1)x_0)a)b)c)a \\
&\to_\beta \lambda a\lambda b\lambda c((\lambda z(((z)a)b)a)c)a \\
&\to_\beta \lambda a\lambda b\lambda c((((c)a)b)a)a = [0100].
\end{aligned}
$$

□

(4) *Definire un λ-termine M per il quale*

$$(\langle M,u\rangle)\langle M,v\rangle \to_\beta \langle M,(v)u\rangle$$

Soluzione. Poichè la coppia è definita come $\langle u,v\rangle = \lambda p((p)u)v$ allora

$$(\langle M,u\rangle)\langle M,v\rangle \to_\beta ((\langle M,v\rangle)M)u\rangle \to_\beta (((M)M)v)u$$

e dunque basta scegliere M in modo tale che sia soddisfatta

$$(((M)M)v)u \to_\beta \langle M, (v)u\rangle$$

dunque una funzione di tre argomenti che costruisce la coppia $\langle M, (v)u\rangle = \lambda p((p)M)(v)u$ per cui

$$M = \lambda m\lambda a\lambda b\lambda p((p)m)(a)b.$$

□

(5) *Con l'ausilio del termine M trovato al punto precedente definire un termine L che calcola la lunghezza di una sequenza.*

Soluzione. Definiamo

$$L = \lambda w(((w)\langle M, \underline{succ}\rangle)\langle M, \underline{succ}\rangle)\langle M, \underline{0}\rangle$$

con $\underline{succ} = \lambda n\lambda f\lambda x(f)((n)f)x$.

□

(6) *Con l'ausilio del termine M trovato al punto (4) definire un termine E che trasforma una sequenza binaria nella lista invertita dei corrispondenti booleani, ad esempio*

$$(E)[01101] \to_\beta \lambda f\lambda k((f)\mathsf{true})((f)\mathsf{false})((f)\mathsf{true})((f)\mathsf{true})((f)\mathsf{false})k,$$

dove $\mathsf{true} = \lambda x\lambda y\, x$ *e* $\mathsf{false} = \lambda x\lambda y\, y$ *e la rappresentazione delle liste è quella analoga ai numerali di Church:*

$$(t_1, \dots, t_n) = \lambda f\lambda k((f)t_1)\dots((f)t_n)k.$$

Soluzione. Definiamo

$$E = \lambda w\lambda f\lambda k((((w)\langle M, (f)\mathsf{false}\rangle)\langle M, (f)\mathsf{true}\rangle)\langle M, k\rangle)\mathsf{false}$$

Dunque se consideriamo la sequenza di esempio:

$$\begin{aligned}
(E)[01101] &\to_\beta \lambda f\lambda k(((([01101])\langle M,(f)\mathsf{false}\rangle)\langle M,(f)\mathsf{true}\rangle)\langle M,k\rangle)\mathsf{false}\\
&\to_\beta \lambda f\lambda k(((((((\langle M,k\rangle)\langle M,(f)\mathsf{false}\rangle)\langle M,(f)\mathsf{true}\rangle)\langle M,(f)\mathsf{true}\rangle)\langle M,(f)\mathsf{false}\rangle)\langle M,(f)\mathsf{true}\rangle)\mathsf{false}\\
&\to_\beta \lambda f\lambda k((((((\langle M,((f)\mathsf{false})k\rangle)\langle M,(f)\mathsf{true}\rangle)\langle M,(f)\mathsf{true}\rangle)\langle M,(f)\mathsf{false}\rangle)\langle M,(f)\mathsf{true}\rangle)\mathsf{false}\\
&\to_\beta \lambda f\lambda k(((((\langle M,((f)\mathsf{true})((f)\mathsf{false})k\rangle)\langle M,(f)\mathsf{true}\rangle)\langle M,(f)\mathsf{false}\rangle)\langle M,(f)\mathsf{true}\rangle)\mathsf{false}\\
&\to_\beta \lambda f\lambda k((((\langle M,((f)\mathsf{true})((f)\mathsf{true})((f)\mathsf{false})k\rangle)\langle M,(f)\mathsf{false}\rangle)\langle M,(f)\mathsf{true}\rangle)\mathsf{false}\\
&\to_\beta \lambda f\lambda k(((\langle M,((f)\mathsf{false})((f)\mathsf{true})((f)\mathsf{true})((f)\mathsf{false})k\rangle)\langle M,(f)\mathsf{true}\rangle)\mathsf{false}\\
&\to_\beta \lambda f\lambda k((\langle M,((f)\mathsf{true})((f)\mathsf{false})((f)\mathsf{true})((f)\mathsf{true})((f)\mathsf{false})k\rangle)\mathsf{false}\\
&\to_\beta \lambda f\lambda k((\mathsf{false})M)((f)\mathsf{true})((f)\mathsf{false})((f)\mathsf{true})((f)\mathsf{true})((f)\mathsf{false})k\\
&\to_\beta \lambda f\lambda k((f)\mathsf{true})((f)\mathsf{false})((f)\mathsf{true})((f)\mathsf{true})((f)\mathsf{false})k
\end{aligned}$$

□

(7) *(facoltativo) Dare il termine E_2 che associa ad una sequenza binaria, la lista dei booleani corrispondenti, presi nello stesso ordine.*

Soluzione.

È sufficiente applicare il termine R che inverte la lista di booleani ottenuta. A questo scopo utilizziamo la struttura di iteratore delle liste di Church, il caso base è la lista vuota $\mathsf{base}_R = x$. Il passo di iterazione è dato da una funzione che prende la lista parziale e aggiunge in coda l'elemento corrente: $\mathsf{step}_R = \lambda b\lambda s\lambda z(s)((f)b)z$ e dunque

$$R = \lambda l\lambda f\lambda x((l)\mathsf{step}_R)\mathsf{base}_R$$

di conseguenza

$$E_2 = \lambda x(R)(E)x.$$

□