ESERCIZI LAMBDA CALCULUS - Alcune soluzioni -

Esercizio 1

```
     \z. ( ((\x. \x. y x) x) (v \z. \w. v) )
                  ^  | |  |   |         |
                  |__|_|  |   |         |
                     |    |   |         |
                    lib. lib.lib.      lib.
```

Esercizio 2

```
- Non esiste ancora oggi una macchina hardware che permetta
  l'implementazione del modello computazionale funzionale.
  Dunque i programmi scritti in linguaggi funzionali vengono
  interpretati o compilati con una conseguente poca efficienza.

- Poichè i dati non vengono modificati, ne vengono creati sempre di nuovi.
  Tale meccanismo comporta un alto numero di duplicati in memoria che
  necessita un utilizzo di garbadge collector per ripulirla.
```

Esercizio 3

```
Una possibilità è la seguente:

  (\vwx.(z\z.x))(x\y.(y(\t.z)yy))
```

Esercizio 4

```
  (\x.(z(\z.((xyz)x))zx))x(\x.((\y.yy)(\z.zz)))
    ^     ^   ||| |  ||  |       ^ ||   ^ ||
    |_ _ _|_ _|||_|_ ||  |       |_||   |_||
              ||     |   |
           libere  lib.lib.
```

Esercizio 6

```
per esempio (\xyzx.xyz)xyz
```

Esercizio 7

```
    xy\y.\z.z(xy)
```

Esercizio 8

```
     ((\x.(\y.(\z.((xy)(xz)))))(\x.(\y.x)))
```

Esercizio 9

```
 vedi testi.
```

Esercizio 23

Diamo innanzitutto la definizione informale di un lambda termine in Normal Form:
Un lambda termine è in forma normale se non esiste alcun suo sottotermine che possa essere ß-ridotto
Sappiamo anche che la ß-riduzione è possibile se e solo se nel termine c'è un sottotermine che è un ß-redex, ovvero è della forma:
(\x.M)N.

Var sottoinsieme di NF.

```
x in Var, M in NF
-------------------
(\x.M) in NF


x in Var, N in NF
-------------------
(xN) in NF


(LM) in NF, N in NF
-------------------
((LM)N) in NF
```

Esercizio 24

xx

Esercizio 28

Se siete riusciti a produrre un lambda-termine con la proprieta' richiesta, ristudiate il lambda-calcolo, c'e' qualcosa che non va (in voi o nel lambda-calcolo...)

Esercizio 29

(\w.((\v.v)(xy)))z

Esercizio 30

t(\y.t)

Esercizio 31

yyy

Esercizio 32

MAI

Esercizio 33

```
  Il termine non possiede forma normale.
  Infatti e' possibile ridurlo solo nel modo seguente:
  (\xy.yx)(\xy.x(yy))(\x.xz(\y.yy)) --> (\x.xz(\y.yy))(\xy.x(yy)) -->
  --> (\xy.x(yy))z(\y.yy) --> (\y.z(yy))(\y.yy) --> x((\y.yy)(\y.yy))
  e da qui si puo' solo continuare a ridurre all'infinito il sottotermine
(\y.yy)(\y.yy)
```

Esercizio 35

```
  (\xxx.((\xx.x)(xx)(\x.x)))x(xx) --> (\xx.((\xx.x)(xx)(\x.x)))(xx) -->
     --> \x.((\xx.x)(xx)(\x.x)) --> \x.((\x.x)(\x.x)) --> \x.(\x.x)
```

Esercizio 36

```
  Quando il termine possiede una forma normale (se ce l'ha e' unica).
```

Esercizio 37

```
  Basta ridurre il termine secondo la strategia Leftmost-Outermost.
    Se il termine ha forma normale questa verra' sicuramente raggiunta
utilizzando tale strategia.

    Se l'insieme dei termini con forma normale fosse decidibile, allora la
funzione di halt del teorema 11 delle note di Paulson sarebbe lamba definibile,
     il che e' impossibile.
```

Esercizio 38

```
  Domanda assurda, infatti (\x.y)((\x.xx)(\x.xx) ) contiene Omega ed e'
normalizzabile
    (anche se, ovviamente, non e' fortemente normalizzabile).
```

Esercizio 47

```
E' sufficiente produrre un controesempio alla transitivita' della beta-riduzione
parallela ( => ) :
(\x.xy)(\z.z) => (\z.z)y => y
(se un termine e' riducibile ad un altro con un passo di beta riduzione,
e' anche riducibile con un passo di beta-parallela).

Dalla definizione di beta-riduzione parallela si vede pero' che (\x.xy)(\z.z)
non e' riducibile a y con un singolo passo di beta-parallela.
```

Esercizio 48 (by A. Nicolosi)

```
Per il Teorema di Church-Rosser, se i due termini fossero b-convertibili,
esisterebbe un altro
 \-termine L al quale entrambi si b-riducono.
Ma (\x.xx)(\x.xx) non si riduce ad altro che a se stesso: pertanto deve accadere
che:
(\x.xx)(\x.xxx) ->> (\x.xx)(\x.xx).     (*)
```

```
Dopo il primo passo di b-riduzione, (\x.xx)(\x.xxx) si riduce a (\x.xxx)
(\x.xxx).
Sopra è stato provato che tale termine si riduce solo a termini del tipo p...p,
dove p == (\x.xxx).
Pertanto non potrà mai accadere che da (\x.xx)(\x.xxx) si pervenga a (\x.xx)
(\x.xx), per cui la (*) è assurda.
L'assurdo deriva dall'aver supposto che i due termini fossero b-convertibili.
```

Esercizio 54_wrong

```
Dim.:
sia N un \-termine che non contiene x.
Allora vale sempre (\x.Mx)N = Mx [N/x] = MN
 Leggendo il primo e l' ultimo membro dell' uguaglianza otteniamo
(\x.Mx)N=MN
ovvero
\x.Mx=M
c.v.d.
```

Esercizio 55 (by N.Fazio)

```
- Y
Y == \f.(\x.f(xx))(\x.f(xx)) -> \f. f((\x.f(xx))(\x.f(xx)))
Quindi Y è definito.

- Y not
Y not == (\f.(\x.f(xx))(\x.f(xx))) (\p. if p false true) ->>
->> (\f.(\x.f(xx))(\x.f(xx))) (\p. p false true) ->
-> (\x.(\p. p false true)(xx))(\x.(\p. p false true)(xx)) ->>
->> (\x.(xx) false true))(\x. (xx) false true)) ->
-> ((\x. (xx) false true))(\x. (xx) false true))) false true
A partire da questo momento in poi l'unica beta-riduzione possibile è quella che
sostituisce a
(xx) il termine ((\x. (xx) false true)(\x. (xx) false true)):
l'effetto non è altro che quello di aggiungere alla fine una coppia false true.
Pertanto la testa non cambierà e non otterremo alcuna hnf: quindi Y not è non-
definito.

- K == \xy.x (Quindi K non è altro che true!)
K è già in hnf: quindi K è definito.

- Y I
Y I == (\f.(\x.f(xx))(\x.f(xx))) (\x.x) -> (\x.(\x.x)(xx))(\x.(\x.x)(xx)) ->
-> (\x.(xx))(\x.(xx)) -> (\x.(xx))(\x.(xx))
Da questo momento in poi la beta riduzione porta sempre allo stesso termine:
quindi Y I è non-definito.

- x omega
x omega == x(\x.xx)(\x.xx)
x omega è già in hnf: quindi x omega è definito.

- Y(K x)
Y(K x) == (\f.(\x.f(xx))(\x.f(xx)))((\xy.x)x) -> (\f.(\x.f(xx))(\x.f(xx)))(\y.x)
->
-> (\x.(\y.x)(xx))(\x.(\y.x)(xx)) ->> (\x.x)(\x.x) -> \x.x == I
Siamo pervenuti ad un termine in hnf: pertanto Y(K x) è definito.
```

[[FB: E' vero, il termine e' definito, pero' nelle riduzioni che mostrano che si riduce ad una head normal form, c'e' un errore che riguarda variabili legate e libere, quale?]]

- n
n == \fx.f(f(...(f x) ...)
n è già in hnf.

Esercizio 59

La prima e' vera; si riducono entrambi i redessi.
La seconda e' falsa: anche riducendo entrambi i redessi contemporaneamente non si otterebbe I.
La terza e' vera: basta ridurre l'unico redesso presente.
La quarta e' falsa: c'e' un unico redesso presente e quindi quello che al piu' puo' fare la beta riduzione
parallela e' ridurlo.
Per ottenere I andrebbero poi fatte altre riduzioni.

Esercizio 61

Sono beta-convertibili,
basta vedere che (modulo alfa-conversione)
(DDD)((\x.xxx)D) <-- ((\x.xxx)D)((\x.xxx)D) --> ((\y.yyy)D)(DDD)
oppure che
(DDD)((\x.xxx)D) --> (DDD)(DDD) <-- ((\y.yyy)D)(DDD)

Esercizio 62

Se J fosse definibile allora avremmo nel lambda-calcolo che
false <-- (\x.false)y <-- (\x.Jxy)y --> Jyy --> true
cioe' true = false, cioe' due forme normali beta convertibili, impossibile.

Esercizio 63

No.
Il primo e' beta convertibile a (DDDx)(DDDy), mentre il secondo a (DDDy)(DDDx).
Se i due termini dati fossero beta-convertibili, allora, per il teorema di Church-Rosser dovrebbero avere un ridotto comune,
il che e' impossibile, visto che sia in (DDDx)(DDDy) che in (DDDy)(DDDx) possono solamente ridursi a se stessi
ed hanno variabili libere distinte nelle stesse posizioni.

Esercizio 64

Se modifichiamo la definizione come mostrato, la proprieta' del diamante si perde.
Il termine fornito pero' non e' un controesempio per la proprieta' del diamante.

Un possibile controesempio e' il seguente: (\x.x(Ix))(Ia) infatti (\x.xx)a <= (\x.x(Ix))(Ia) => (Ia)(I(Ia))
Ora, con la definizione di "=>" modificata, non e' possibile far confluire sullo stesso termine (\x.xx)a e (Ia)(I(Ia))
con un solo passo di riduzione =>.

Notare come il controesempio sia un adattamento di quello del Paulson alla proprieta' del diamante per la beta-riduzione ad un passo.

Esercizio 65

L'insieme e' ricorsivo.
E' l'insieme dei termini N a cui posso arrivare, partendo da M, con due passi di riduzione =>.
E' ricorsivo perche' e' finito.
Infatti, preso un lambda termine questo ha un numero finito di sottinsiemi di suoi possibili redessi,
ovviamente, e quindi il numero di P a cui posso arrivare con un passo di riduzione => e' finito.
Per lo stesso motivo sono finiti gli N a cui posso arrivare partendo dai possibili P.
Qualcuno potrebbe essere stato tratto in inganno credendo che l'insieme sia chiuso per beta-conversione.

Basta prendere il termine (\xy.y(xz))III come M.
Ora, M e' beta-convertibile a z, ma non riusciamo ad arrivare a z con due sole riduzioni =>.

Esercizio 66

E' ricorsivo.
Per vedere se un termine appartiene all'insieme dato basta ridurre tutti i beta redessi in esso contenuti
e controllare se quella ottenuta e' una forma normale.

Esercizio 67

Entrambe calcolano la stessa funzione: il successore.
Infatti, consideriamo il numerale di Church per il numero m : \gy.g(..-mvolte-..(g(y))...)
Abbiamo che (\nfx.f(nfx))(\gy.g(..-mvolte-..(g(y))...)) -->> \fx.f(f(..-mvolte-..(f(x))...))
e vediamo che \fx.f(f(..-mvolte-..(f(x))...)) non e' altro che \fx.f(..-m+1volte-..(f(x))...)),
cioe' il numerale di Church per il numero m+1.
Per l'altro termine (\nfx.nf(fx))(\gy.g(..-mvolte-..(g(y))...)) -->> \fx.f(..-mvolte-..(f(f(x)))...)
e vediamo che \fx.f(..-mvolte-..(f(f(x)))...) non e' altro che \fx.f(..-m+1volte-..(f(x))...)),
cioe' il numerale di Church per il numero m+1.
I due termini sono in forma normale e, non essendo identici (modulo alfa conversione)
non possono essere beta convertibili per il teorema di Church-Rosser.
Per far si che due termini che rappresentano la stessa funzione siano convertibili
occorrerebbe aggiungere l'assioma di eta-conversione alla teoria della beta-conversione
(occorrerebbe cioe' essere nella teoria delle beta-eta-conversione).

Non e' comunque da ritenersi sconveniente la cosa perche' a noi puo' bastare benissimo

una teoria che uguaglia i risultati dell'applicazione di argomenti a termini che rappresentano
la stessa funzione matematica.
La teoria della beta-conversione vuole essere una teoria di funzioni come algoritmi e
non di funzioni in termini strettamente matematici.

I nostri due termini quindi in tale teoria devono essere diversi poiche',
pur calcolando la stessa funzione sono, come algoritmi, differenti
(i due termini vengono solitamente chiamati successore destro e successore sinistro).

Esercizio 68

Se P -->beta Q allora esiste un sottotermine di P della forma (\x.M)N che viene contratto a M[N/x].
Per dimostrare che FV(Q) e' un sottoinsieme di FV(P) basta dimostrare quindi che FV(M[N/x]) e' un sottinsieme di FV((\x.M)N).
Distinguiamo due casi:
-- x appartiene a FV(M)
In questo caso in FV(M[N/x]) ci sono tutte le variabili libere di M
(la x a cui sostituiamo N e' legata in P) piu' tutte le variabili libere di N
(poiche' per definizione di sostituzione nessuna variabile libera di N
diventera' legata dopo la sostituzione.
Quindi abbiamo che FV(M[N/x]) e' un sottinsieme (in particolare e' uguale a) FV((\x.M)N)

-- x non appartiene a FV(M) In questo caso, per definizione di sostituzione, FV(M[N/x]) = FV(M).
Siccome FV((\x.M)N) = FV(\x.M) unito FV(N) abbiamo che FV(M[N/x]) e' un sottinsieme di FV((\x.M)N)
(in particolare e' uguale se FV(N) = empty.

La dimostrazione data si potrebbe formalizzare meglio, ma per il livello del nostro corso puo' andare.

Esercizio 69

L'insieme dato non e' chiuso per beta conversione.
Prendiamo infatti un M che appartiene all'insieme dato.
Prendiamo x che non appartiene alle variabili libere di M.
Allora (\x.M)((\y.yy)(\y.yy)) e' beta convertibile ad M (in particolare si riduce ad M in un passo).
E' chiaro che (\x.M)((\y.yy)(\y.yy)) e' normalizzabile,
ma ci vogliono ora almeno k+1 passi per arrivare alla forma normale.

Esercizio 73 (By A. Nicolosi)

La dimostrazione si basa sul teorema di Standardizzazione:
Se un \-termine ha forma normale, si perverrà a tale
forma seguendo la strategia di riduzione leftmost-outermost.
In tutti i \-termini ottenibili per b-riduzione dal \-termine di partenza,
c'è solo il b-redex (\x.xxx)(\x.xxx); dunque tutte le strategie sono
equivalenti, e dovrebbero portare ad una forma normale. Ma poichè tale
strategia riduce (\x.xxx)(\x.xxx) -> (\x.xxx)(\x.xxx)(\x.xxx), essa introduce
sempre un b-redex e non terminerà mai di b-ridurre: pertanto (\x.xxx)(\x.xxx)

non è normalizzabile.
Resta da provare che nei \-termini ottenibili per b-riduziuone dal \-termine di partenza c'è sempre uno ed un solo b-redex. Proviamo per induzione che, posto p == (\x.xxx), dopo n b-riduzioni otteniamo p...p (n+2 volte).

Caso Base: n = 0
Dopo 0 b-riduzioni siamo all'inizio dove il \-termine e'
(\x.xxx)(\x.xxx) == pp.

Passo Induttivo:
Hp: Dopo n b-riduzioni otteniamo pp...p (n+2 volte).
Ts: Dopo n+1 b-riduzioni otteniamo (pp)p...p (n+3 volte).
Dim.
L'unico b-redex in pp...p e' l'applicazione pp all'inizio:b-riducendola otteniamo (pp)p...p.
Cio' prova che i \-termini che otteniamo sono tutti della forma pp...p:
e' chiaro che in tali \-termini c'e' sempre uno ed un solo b-redex, cioe' pp.

Esercizio 74

I due termini si possono ridurre allo stesso termine (y).
Dunque sono beta convertibili poiche' la relazione di beta conversione coincide con la chiusura transitiva e simmetrica della beta-riduzione a piu' passi.

Esercizio 75 (By A. Nicolosi)

La dimostrazione si basa sulla definizione di punti fisso e di operatore di punto fisso.

Punto fisso
Dato un \-termine F, si dice che X è un punto fisso di F se accade che
F X = X.

Operatore di punto fisso
Un combinatore Q è un operatore di punto fisso se per ogni \-termine F accade che (Q F) è un punto fisso di F, cioè
F(Q F) = (Q F).
Nel caso dell'operatore di punto fisso Y, la b-equivalenza di sopra si scrive:
F(Y F) = (Y F). (*)

Applichiamo due volte la (*) al secondo membro della equivalenza da provare, scegliendo (YY) come F:

```
(YY)((YY)(Y(YY))) = (YY)(Y(YY)) = (Y(YY)).
     -----------    --------
```

Applichiamo una volta la (*) al primo membro della equivalenza, scegliendo Y come F:

```
(Y(Y(YY))) = (Y(YY)).
   ----
```

Sono stati sottolineati i sottotermini ai quali è stata applicata la (*).

Dall'identità dei due lati destri segue (per transitivita di =) che:
(Y(Y(YY))) = (YY)((YY)(Y(YY))).

Osservazione
Dalla dimostrazione si deduce anche un fatto interessante: il \-termine YY è punto fisso di se stesso!
Ciò segue sempre per la (*) scritta per Y:
Y(YY) = (YY). (**)
Infatti la b-equivalenza (**) ci dice che Y(YY), che per definizione di operatore di punto fisso è un punto fisso di (YY),
è b-equivalente a (YY): quindi (YY) è punto fisso di se stesso.

Esercizio 77(By A. Nicolosi)

Per il Teorema di Church-Rosser, se i due termin fossero b-convertibili, esisterebbe un altro \-termine L al quale entrambi si b-riducono.
Ma (\x.xx)(\x.xx) non si riduce ad altro che a se stesso: pertanto deve accadere che:

(\x.xx)(\x.xxx) ->> (\x.xx)(\x.xx). (*)

Dopo il primo passo di b-riduzione, (\x.xx)(\x.xxx) si riduce a (\x.xxx)(\x.xxx).
Sopra è stato provato che tale termine si riduce solo a termini del tipo p...p, dove p == (\x.xxx).
Pertanto non potrà mai accadere che da (\x.xx)(\x.xxx) si pervenga a (\x.xx)(\x.xx), per cui la (*) è assurda.
L'assurdo deriva dall'aver supposto che i due termini fossero b-convertibili.

Esercizio 78

Se esistesse un tale F, cosa succederebbe se considerassimo come M ed N, rispettivamente \z.z e (xy) ?

Ci sono ovviamente tanti altri modi per dimostrarlo.

Esercizio 81 (By N. Fazio)

Scriviamo preliminarmente la definizione ricorsiva del prodotto in \-calcolo:

mult == \xy.if (iszero x) 0 (add y (mult (pre x) y)).

dove add è un \-termine che rappresenta la somma curryficata.

Lambda-astraendo dalla definizione ricorsiva, otteniamo il funzionale:

F == \fxy.if (iszero x) 0 (add y (f (pre x) y)).

Per ottenere la rappresentazione eslicita di mult, basta applicare l'operatore di punto fisso:

Y == \f.((\x.f(xx))(\x.f(xx)))

ottenendo:

Y F == (\f.((\x.f(xx))(\x.f(xx)))) (\fxy.if (iszero x) 0 (add y (f (pre x) y))).

Sfruttando la beta-equivalenza che caratterizza gli operatori di punto fisso:

```
Y F =b F(Y F)
```

otteniamo:

```
Y F =b F(Y F) == ( \fxy.if (iszero x) 0 (add y (f (pre x) y)))(Y F) ->
-> (\xy.if (iszero x) 0 (add y ((Y F) (pre x) y))).
```

Questa è in una forma migliore, perché possiamo subito "applicare i due argomenti".

Esercizio 82

La funzione costante 1.

Esercizio 85

La funzione massimo tra due numeri naturali.

Esercizio 86

```
Y\l.(cons x l)

Y(\f.\n.(cons (sqr n) (f (succ n))))zero

Y\l.(cons x l) -->> (\l.(cons x l))(Y(\l.(cons x l))) --> (cons x (Y\l.(cons x
l))) -->...
```

Indicando con F il termine \f.\n.(cons (sqr n) (f (succ n))))

```
Y F zero -->> (\f.\n.(cons (sqr n) (f (succ n)))) (Y F) zero -->> cons (sqr
zero) ((Y F) (succ zero)) -->....
```

Esercizio 87

Il termine \xy.((\yx.x(yy))z(\z.zz)) non e' in head-normal form,
pero' possiede una head-normal form: \xy.zz(zz).
Un lambda termine puo' avere piu' di una head-normal form.
Per esempio per il termine \x.(Ix)(Ia)(Ib) \x.x(Ia)(Ib) e' una head-normal, come anche \x.x(Ia)b.
Ovviamente le prime astrazioni e la variabile di testa sono le stesse per tutte le head-normal forms
di un termine che possiede head-normal form e questo si dimostra utilizzando il teorema di Church-Rosser
e osservando che in un termine in head-normal form le possibili riduzioni che possiamo
fare su di esso non altereranno mai ne' le prime astrazioni, ne' la variabile di testa.