Aspetti Avanzati dei Linguaggi di Programmazione

Esercizi

Autori:
Campese Stefano
Enrico Savoca

Contents

0.1 Il mini linguaggio funzionale (note 2)

Esercizio 1.1

La relazione di riduzione data definisce una strategia efficiente per la valutazione del ter- mine if-then-else, che permette cioè di valutare unicamente il ramo scelto dalla valutazione della guardia booleana. Ridefinire la semantica operazionale del linguaggio in modo che adotti una strategia non efficiente per il costrutto if-then-else, valutando entambi i rami del costrutto condizionale.

Svolgimento Per ottenere una semantica meno efficiente, si cambiano gli assiomi (IF-TRUE) e (IF-FALSE) e si aggiungono le regole (THEN) ed (ELSE). (IF) si mantiene tale.

Assiomi

$$\text{(IF-TRUE)}\ \frac{}{if\ TRUE\ then\ v_1\ else\ v_2 \rightarrow\ v_1}$$

$$\text{(IF-FALSE)}\ \frac{}{if\ FALSE\ then\ v_1\ else\ v_2 \rightarrow\ v_2}$$

Regole

$$\text{(THEN)}\ \frac{M_2 \rightarrow\ M_2'}{if\ v_1\ then\ M_2\ else\ M_3 \rightarrow\ if\ v_1\ then\ M_2'\ else\ M_3}$$

$$\text{(ELSE)}\ \frac{M_3 \rightarrow\ M_3'}{if\ v_1\ then\ v_2\ else\ M_3 \rightarrow\ if\ v_1\ then\ v_2\ else\ M_3'}$$

Esercizio 1.4

La valutazione, cioè l'esecuzione del programma, è deterministica. Se $M \rightarrow$ M' e $M \rightarrow$ M" allora M'=M".
Dimostrare la proposizione precedente per induzione sulla struttura del termine M.

Dimostrazione Si dimostra per induzione sulla derivazione.

Casi base:

- M=true
  Se true $\rightarrow$ M' e true $\rightarrow$ M" $\implies$ M' = M"
  Vero perchè true è un valore finale;
- M=false
  Se false $\rightarrow$ M' e false $\rightarrow$ M" $\implies$ M' = M"
  Vero perchè false è un valore finale;
- M=fn x.M
  Se fn x.M $\rightarrow$ M' e fn x.M $\rightarrow$ M" $\implies$ M' = M"
  Vero perchè fn x.M è un valore finale;
- M=n
  Se n $\rightarrow$ M' e n $\rightarrow$ M" $\implies$ M' = M"
  Vero perchè n è un valore finale.

Passo induttivo:

- Tesi: $M = M_1 + M_2 \ \rightarrow \ M'$, $M_1 + M_2 \ \rightarrow$ M" $\implies$ M'=M"
  Per ipotesi induttiva:
  - Se $M_1 \ \rightarrow \ M_1'$ e $M_1 \ \rightarrow \ M_1''$ allora $M_1' = M_1''$;
  - Se $M_2 \ \rightarrow \ M_2'$ e $M_2 \ \rightarrow \ M_2''$ allora $M_2' = M_2''$.

  3 Casi possibili:

  a) Derivazione tramite l'uso della regola (SUM-LEFT):

  $$\text{(SUM-LEFT)}\ \frac{M_1 \ \rightarrow \ M_1'}{M_1 + M_2 \ \rightarrow \ M_1' + M_2 = M'}$$

  In questo caso $M_1' = M_1''$ per ipotesi induttiva e quindi anche $M' = M''$;
  La derivazione risulta deterministica applicando questa regola.

  b) Derivazione tramite l'uso della regola (SUM-RIGHT):

  $$\text{(SUM-RIGHT)}\ \frac{M_2 \ \rightarrow \ M_2'}{v_1 + M_2 \ \rightarrow \ v_1 + M_2' = M'}$$

  In questo caso $M_2' = M_2''$ per ipotesi induttiva e quindi anche $M' = M''$;
  La derivazione risulta deterministica applicando questa regola.

  c) Derivazione tramite l'uso della regola (SUM):

  $$\text{(SUM)}\ \frac{\text{con n'}=n_1 + n_2}{n_1 + n_2 \ \rightarrow \ n'}$$

Il risultato della somma è univoco e di conseguenza la derivazione è deterministica.

In tutti e 3 i casi, i soli possibili, la derivazione è deterministica, quindi la tesi è dimostrata.

- Tesi: $M = M_1 - M_2 \rightarrow M'$, $M_1 - M_2 \rightarrow$ M" $\implies$ M'=M"
  Per ipotesi induttiva:
  - Se $M_1 \rightarrow M_1'$ e $M_1 \rightarrow M_1''$ allora $M_1' = M_1''$;
  - Se $M_2 \rightarrow M_2'$ e $M_2 \rightarrow M_2''$ allora $M_2' = M_2''$.

  3 Casi possibili:

  a) Derivazione tramite l'uso della regola (MINUS-LEFT):

  $$\text{(MINUS-LEFT)}\,\frac{M_1 \rightarrow M_1'}{M_1 - M_2 \rightarrow M_1' - M_2 = M'}$$

  In questo caso $M_1' = M_1''$ per ipotesi induttiva e quindi anche $M' = M''$; La derivazione risulta deterministica applicando questa regola.

  b) Derivazione tramite l'uso della regola (MINUS-RIGHT):

  $$\text{(MINUS-RIGHT)}\,\frac{M_2 \rightarrow M_2'}{v_1 - M_2 \rightarrow v_1 - M_2' = M'}$$

  In questo caso $M_2' = M_2''$ per ipotesi induttiva e quindi anche $M' = M''$; La derivazione risulta deterministica applicando questa regola.

  c) Derivazione tramite l'uso della regola (MINUS):

  $$\text{(MINUS)}\,\frac{\text{con n'}=n_1 - n_2}{n_1 - n_2 \rightarrow n'}$$

  Il risultato della differenza è univoco e di conseguenza la derivazione è deterministica.

  In tutti e 3 i casi, i soli possibili, la derivazione è deterministica, quindi la tesi è dimostrata.

- Tesi: $M = if\, M_1\, then\, M_2\, else\, M_3 \rightarrow M'$, $if\, M_1\, then\, M_2\, else\, M_3 \rightarrow M'' \implies$ M'=M"
  Per ipotesi induttiva:

- Se $M_1 \rightarrow M_1'$ e $M_1 \rightarrow M_1''$ allora $M_1' = M_1''$.

3 Casi possibili:

a) Derivazione tramite l'uso della regola (IF):

$$(\text{IF})\ \frac{M_1 \rightarrow M_1'}{if\ M_1\ then\ M_2\ else\ M_3 \rightarrow if\ M_1'\ then\ M_2\ else\ M_3 = M'}$$

In questo caso $M_1' = M_1''$ per ipotesi induttiva e quindi anche $M' = M''$; La derivazione risulta deterministica applicando questa regola.

b) Derivazione tramite l'uso dell'assioma (IF-TRUE):

$$(\text{IF-TRUE})\ \frac{}{if\ true\ then\ M_2\ else\ M_3 \rightarrow M_2 = M'}$$

In questo caso $M_2 = M' = M''$ per applicazione dell'assioma e la derivazione risulta deterministica.

c) Derivazione tramite l'uso della regola (IF-FALSE):

$$(\text{IF-FALSE})\ \frac{}{if\ false\ then\ M_2\ else\ M_3 \rightarrow M_3 = M'}$$

In questo caso $M_3 = M' = M''$ per applicazione dell'assioma e la derivazione risulta deterministica.

In tutti e 3 i casi, i soli possibili, la derivazione è deterministica, quindi la tesi è dimostrata.

- Tesi: $M = M_1\, M_2 \rightarrow M'$, $M_1\, M_2 \rightarrow$ M" $\implies$ M'=M"
  Per ipotesi induttiva:
  - Se $M_1 \rightarrow M_1'$ e $M_1 \rightarrow M_1''$ allora $M_1' = M_1''$;
  - Se $M_2 \rightarrow M_2'$ e $M_2 \rightarrow M_2''$ allora $M_2' = M_2''$.

  2 Casi possibili:

  a) Derivazione tramite l'uso della regola (APP1):

  $$(\text{APP1})\ \frac{M_1 \rightarrow M_1'}{M_1\, M_2 \rightarrow M_1'\, M_2 = M'}$$

  In questo caso $M_1' = M_1''$ per ipotesi induttiva e quindi anche $M' = M''$; La derivazione risulta deterministica applicando questa regola.

  b) Derivazione tramite l'uso della regola (APP2):

$$\text{(APP2)}\ \frac{M_2 \ \to \ M_2'}{v_1\, M_2 \ \to \ v_1\, M_2' = M'}$$

In questo caso $M_2' = M_2''$ per ipotesi induttiva e quindi anche $M' = M''$;
La derivazione risulta deterministica applicando questa regola.

In tutti e 2 i casi, i soli possibili, la derivazione è deterministica, quindi la tesi è dimostrata.

- Tesi: $M = fn\, x.M\ v \ \to \ Mx := v = M'$, $fn\, x.M\ v \ \to \ Mx := v = M'' \implies$ M'=M"
  Derivazione tramite l'uso dell'assioma (BETA):

  $$\text{(BETA)}\ \frac{}{fn\, x.M\ v \ \to \ Mx := v = M'}$$

  Tramite l'applicazione dell'assioma si ottiene Mx:=v=M'=M".
  La derivazione risulta quindi deterministica e la tesi risulta dimostrata.

La derivazione è determinstica all'applicazione di tutte le regole e gli assiomi del linguaggio, di conseguenza è deterministica l'esecuzione del programma.

□

Esercizio 1.5

Descrivere la valutazione del termine ((fn x.3) (fn y.y)) ((fn z.if z then 1 else 0) (false)). Modificare le regole di valutazione in modo tale che, mantenendo una strategia call-by-value, il termine precedente evolva in un termine stuck in meno passi di riduzione. Scrivere le regole di valutazione della strategia call-by-name e valutare il termine precedente secondo questa strategia.

Svolgimento Valutazione tramite strategia call-by-value:
passo 1

$$\text{(APP1)}\ \frac{\text{(BETA)}\ \dfrac{}{(\mathsf{fn}\ x.3)(\mathsf{fn}\ y.y) \ \to \ 3}}{((\mathsf{fn}\ x.3)(\mathsf{fn}\ y.y))\ ((\mathsf{fn}\ z.if\ z\ then\ 1\ else\ 0)(false)) \ \to \ 3\ ((\mathsf{fn}\ z.if\ z\ then\ 1\ else\ 0)(false))}$$

passo 2

$$(\text{APP2})\ \dfrac{(\text{BETA})\ \dfrac{}{(\mathsf{fn}\ z.if\ z\ then\ 1\ else\ 0)(false)\ \rightarrow\ if\ false\ then\ 1\ else\ 0}}{3\ ((\mathsf{fn}\ z.if\ z\ then\ 1\ else\ 0)(false))\ \rightarrow\ 3\ (if\ false\ then\ 1\ else\ 0)}$$

passo 3

$$(\text{APP2})\ \dfrac{(\text{IF-FALSE})\ \dfrac{}{if\ false\ then\ 1\ else\ 0\ \rightarrow\ 0}}{3\ (if\ false\ then\ 1\ else\ 0)\ \rightarrow\ 3\ 0}$$

3 0 costituisce uno stuck: non esistono regole o assiomi per continuare l'esecuzione di tale programma. Esecuzione terminata dopo 3 passi di derivazione.
Si può mantenere la strategia CBV e terminare l'esecuzione in un numero minore di passi modificando (APP2):

$$(\text{APP2})\ \dfrac{N\ \rightarrow\ N'}{(\mathsf{fn}\ x.M)\ N\ \rightarrow\ \mathsf{fn}\ x.M\ N'}$$

In questo modo l'esecuzione del programma terminerebbe dopo un solo passo. Infatti, ci si ritroverebbe in una forma in cui non è possibile applicare alcuna regola o assioma (3 non è una funzione).

Con una strategia CBN (Call-by-name) non ho (APP2) e la funzione (BETA) è la seguente:

$$(\text{BETA})\ \dfrac{\checkmark}{(\mathsf{fn}\ x.M)\ N\ \rightarrow\ Mx := N}$$

Con questa regola mostriamo l'esecuzione del programma:

passo 1

$$(\text{APP1})\ \dfrac{(\text{BETA})\ \dfrac{}{(\mathsf{fn}\ x.3)(\mathsf{fn}\ y.y)\ \rightarrow\ 3}}{((\mathsf{fn}\ x.3)(\mathsf{fn}\ y.y))\ ((\mathsf{fn}\ z.if\ z\ then\ 1\ else\ 0)(false))\ \rightarrow\ 3\ ((\mathsf{fn}\ z.if\ z\ then\ 1\ else\ 0)(false))}$$

Stuck! Non esistono regole o assiomi applicabili: APP1 non è applicabile in quanto 3 non è funzione.

Esercizio 1.6

Consideriamo le seguenti definizioni in Scala: def square(x:Int):Int = x*x def sumOfSquare(x:Int,y:Int):Int = square(x)+square(y) Descrivere i passi di riduzione dell'espressione sumOfSquare(3,4) secondo una strategia call- by-value analoga a quella definita nella sezione precedente. Descrivere inoltre la riduzione della stesa espressione secondo la strategia call-by-name. Descrivere i passi di riduzione dell'espressione sumOfSquare(3,2+2) secondo le strategie call-by-value e call-by-name.

Svolgimento Caso sumOfSquare(3,4).
CBV si comporta come CBN.

passo 1

$$(\text{BETA})\ \frac{}{\text{sumOfSquare}(3,\ 4) \to \text{square}(3) + \text{square}(4)}$$

passo 2

$$(\text{SUM-L})\ \frac{(\text{BETA})\ \frac{}{\text{square}(3) \to 3*3}}{\text{square}(3) + \text{square}(4) \to 3*3 + \text{square}(4)}$$

passo 3

$$(\text{SUM-L})\ \frac{(\text{MULT})\ \frac{}{3*3 \to 9}}{3*3 + \text{square}(4) \to 9 + \text{square}(4)}$$

passo 4

$$(\text{SUM-R})\ \frac{(\text{BETA})\ \frac{}{\text{square}(4) \to 4*4}}{9 + \text{square}(4) \to 9 + 4*4}$$

passo 5

$$(\text{SUM-R})\ \frac{(\text{MULT})\ \frac{}{4*4 \to 16}}{9 + 4*4 \to 9 + 16}$$

passo 6

$$(\text{SUM})\ \frac{}{9 + 16 \to 25}$$

Caso sumOfSquare(3,2+2)
CBV:
passo 1

$$(\text{APP2})\ \frac{(\text{SUM})\ \frac{}{2+2 \to 4}}{\text{sumOfSquare}(3,\ 2+2) \to \text{sumOfSquare}(3,\ 4)}$$

I passi successivi sono gli stessi del caso precedente, quindi i passi di esecuzione sono 7.

CBN:
passo 1

$$(\text{BETA})\ \frac{}{\text{sumOfSquare}(3,\ 2+2) \rightarrow \text{square}(3) + \text{square}(2+2)}$$

passo 2

$$(\text{SUM-L})\ \frac{(\text{BETA})\ \frac{}{\text{square}(3) \rightarrow 3*3}}{\text{square}(3) + \text{square}(2+2) \rightarrow 3*3 + \text{square}(2+2)}$$

passo 3

$$(\text{SUM-L})\ \frac{(\text{MULT})\ \frac{}{3*3 \rightarrow 9}}{3*3 + \text{square}(2+2) \rightarrow 9 + \text{square}(2+2)}$$

passo 4

$$(\text{SUM-R})\ \frac{(\text{BETA})\ \frac{}{\text{square}(2+2) \rightarrow (2+2)*(2+2)}}{9 + \text{square}(2+2) \rightarrow 9 + (2+2)*(2+2)}$$

passo 5

$$(\text{SUM-R})\ \frac{(\text{MULT-L})\ \frac{(\text{SUM})\ \frac{}{2+2 \rightarrow 4}}{(2+2)*(2+2) \rightarrow (4)*(2+2)}}{9 + (2+2)*(2+2) \rightarrow 9 + (4)*(2+2)}$$

passo 6

$$(\text{SUM-R})\ \frac{(\text{MULT-R})\ \frac{(\text{SUM})\ \frac{}{2+2 \rightarrow 4}}{4*(2+2) \rightarrow 4*4}}{9 + 4*(2+2) \rightarrow 9 + 4*4}$$

passo 7

$$(\text{SUM-R})\ \frac{(\text{MULT})\ \frac{}{4*4 \rightarrow 16}}{9 + 4*4 \rightarrow 9 + 16}$$

passo 8

$$(\text{SUM})\ \frac{}{9 + 16 \rightarrow 25}$$

I passi di esecuzione sono 8 in questo caso (CBN).

Esercizio 1.7

Si consideri la seguente definizione in Scala: def test(x:Int, y:Int):Int = x*x confrontare la velocità (cioè il numero di passi) di riduzione delle seguenti espressioni secondo le strategie CBV e CBN, indicando quale delle due è piú veloce

1. test(2,3)
2. test(3+4,8)
3. test(7,2*4)
4. test(3+4,2*4)

Svolgimento

1. CBV=CBN: entrambi terminano l'esecuzione in 2 passi;
2. CBV<CBN: CBV termina prima perchè valuta l'espressione 3+4 subito (una sola volta). CBN invece invocherà prima la funzione, sostituendo ad x l'espressione. Essendo x presente 2 volte nella funzione, la valutazione dovrà essere effettuata 2 volte.
3. CBV>CBN: CBN termina prima perchè non valuta l'espressione 2*4 che non è utilizzata nella funzione test.
4. CBV=CBN: entrambi terminano l'esecuzione in 4 passi; la differenza sta nel fatto che con CBV si svolgeranno prima le due operazioni (somma e moltiplicazione) e poi si chiamerà la funzione. Con CBN invece, si eviterà di calcolare 2*4, tuttavia si calcolerà 3+4 due volte.

Esercizio 1.8

Definire in Scala una funzione and che si comporta come il costrutto logico $x \&\& y$.

Svolgimento

```
// and logico. b1 e b2 passati by-name
def and(b1:=> Boolean, b2:=> Boolean): Boolean =
   if(!b1) false
   else if(b2) true
       else false;
```

0.2 Il mini linguaggio funzionale (note 3)

Esempi di derivazione svolti

a) $\emptyset \vdash \text{if true then } 5 + 7 \text{ else } 2 : \text{Nat}$

b) $\emptyset \vdash (\mathsf{fn}\ \ x{:}T.x)\ x : T \to T$

c) $\emptyset \vdash (\mathsf{fn}\ \ x{:}\text{Bool}.x)\ \text{true} : \text{Bool}$

d) $f{:}\text{Bool} \to \text{Bool} \vdash f\ (\text{if false then true else false}) : \text{Bool}$

e) $f{:}\text{Bool} \to \text{Bool} \vdash \mathsf{fn}\ \ x{:}\text{Bool}.f(\text{if } x \text{ then false else } x) : \text{Bool} \to \text{Bool}$

Svolgimento

a)

$$(\text{IF-THEN-ELSE})\ \dfrac{(\text{TRUE})\ \dfrac{\checkmark}{\emptyset \vdash \text{true: Bool}} \qquad (\text{SUM})\ \dfrac{(\text{NAT})\ \dfrac{\checkmark}{\emptyset \vdash 5 : \text{Nat}} \quad (\text{NAT})\ \dfrac{\checkmark}{\emptyset \vdash 7 : \text{Nat}}}{\emptyset \vdash 5 + 7 : \text{Nat}} \qquad (\text{NAT}) -}{\emptyset \vdash \text{if true then } 5 + 7 \text{ else } 2 : \text{Nat}}$$

b)

$$(\text{FUN})\ \dfrac{(\text{VAR})\ \dfrac{x{:}T \in x{:}T}{x{:}T \vdash x : T}}{\emptyset \vdash (\mathsf{fn}\ \ x{:}T.x)\ x : T \to T}$$

c)

$$(\text{APP})\ \dfrac{(\text{FUN})\ \dfrac{(\text{VAR})\ \dfrac{x{:}\text{Bool} \in x{:}\text{Bool}}{x{:}\text{Bool} \vdash x : \text{Bool}}}{\emptyset \vdash \mathsf{fn}\ \ x{:}\text{Bool}.x : \text{Bool}/rightarrow\text{Bool}} \qquad (\text{TRUE})\ \dfrac{\checkmark}{\emptyset \vdash \text{true} : \text{Bool}}}{\emptyset \vdash (\mathsf{fn}\ \ x{:}\text{Bool}.x)\ \text{true} : \text{Bool}}$$

d) Sia $\Gamma = f{:}\text{Bool} \to \text{Bool}$

$$(\text{APP})\ \dfrac{(\text{VAR})\ \dfrac{f{:}\text{Bool} \to \text{Bool} \in \Gamma}{\Gamma \vdash f : \text{Bool} \to \text{Bool}} \qquad (\text{IF-THEN-ELSE})\ \dfrac{(\text{FALSE})\ \dfrac{\checkmark}{\Gamma \vdash \text{false} : \text{Bool}} \quad (\text{TRUE})\ \dfrac{\checkmark}{\Gamma \vdash \text{true} : \text{Bo}}}{\Gamma \vdash \text{if false then true el}}}{\Gamma \vdash f\ (\text{if false then true else false}) : \text{Bool}}$$

e) Sia $\Gamma = f{:}\text{Bool} \to \text{Bool}$

$$
\text{(APP)}\ \dfrac{\text{(FUN)}\ \dfrac{\text{(VAR)}\ \dfrac{\text{f:Bool} \to \text{Bool} \in \Gamma,\ \text{x:Bool}}{\Gamma,\ \text{x:Bool} \vdash \text{f: Bool} \to \text{Bool}}}{\Gamma \vdash \mathsf{fn}\ \ \text{x:Bool.f: ?} \to \text{Bool} \to \text{Bool}} \qquad \text{(IF-THEN-ELSE)}\ \dfrac{\text{(VAR)}\ \dfrac{\text{x:Bool} \in \Gamma}{\Gamma \vdash \text{x :?}} \quad \text{(TRUE)}\ \dfrac{}{\Gamma}}{\Gamma \vdash \text{(if x t}}}{\Gamma \vdash \mathsf{fn}\ \ \text{x:Bool.f(if x then false else x) :Bool}\to\text{Bool}}
$$

Quindi ?=Bool.

Esercizio 2.1

Trovare un contesto Γ tale che Γ' f x y : Bool sia derivabile.

Svolgimento

$$
\text{(APP)}\ \dfrac{\text{(APP)}\ \dfrac{\text{(VAR)}\ \dfrac{\text{f:}T_2 \to T_1 \to \text{Bool} \in \Gamma}{\Gamma \vdash \text{f :}T_2 \to T_1 \to \text{Bool}} \qquad \text{(VAR)}\ \dfrac{\text{x:}T_2 \in \text{x:}T_2}{\Gamma \vdash \text{x :}T_2}}{\Gamma \vdash \text{f x :}T_1 \to \text{Bool}} \qquad \text{(VAR)}\ \dfrac{\text{y :}T_1 \in \Gamma}{\Gamma \vdash \text{y :}T_1}}{\Gamma \vdash \text{f x y :Bool}}
$$

Esiste un contesto per questo programma, in quanto il programma si puo' ottenere applicando le regole e costruendo l'albero di derivazione.
Fanno parte del contesto le seguenti regole di tipo:

- f:$T_2 \to T_1 \to$ Bool
- x :T_2
- y :T_1

Caso f (x y):

$$
\text{(APP)}\ \dfrac{\text{(VAR)}\ \dfrac{\text{f :}T_1 \to \text{Bool} \in \Gamma}{\Gamma \vdash \text{f :}T_1 \to \text{Bool}} \qquad \text{(APP)}\ \dfrac{\text{(VAR)}\ \dfrac{\text{x:}T_2 \to T_1 \in \Gamma}{\Gamma \vdash \text{x :}T_2} \quad \text{(VAR)}\ \dfrac{\text{y:}T_2 \in \Gamma}{\Gamma \vdash \text{y :}T_2}}{\Gamma \vdash \text{x y :}T_1}}{\Gamma \vdash \text{f (x y) :Bool}}
$$

Esiste un contesto anche per questo programma.
Fanno parte del contesto le seguenti regole di tipo:

- f:$T_1 \to$ Bool
- x :$T_2 \to T_1$
- y :T_2

Esercizio 2.2

Il giudizio Γ' x x : T e' derivabile? Se si', trovare una derivazione per qualche Γ, T, altrimenti provare che non e' derivabile.

Svolgimento

$$(\text{APP})\,\frac{\Gamma \vdash \text{x} : T_1 \to \text{T} \qquad \Gamma \vdash \text{x} : T_1}{\Gamma \vdash \text{x x} : \text{T}}$$

No, non e' derivabile. Sarebbe necessario avere un sistema di tipi che ammetta il tipo ricorsivo.

Esercizio 3.1

Provare che ogni sottotermine di un termine ben tipato e ben tipato.

Svolgimento Per la risoluzione di questo esericizio e necessario utilizzare il metodo dell'induzione sui passi di derivazione, ovvero sull'altezza dell'albero di derivazione.

Caso 0 A questo livello il termine e congruo con il sottotermine di conseguenza se il termine e bentipato lo e anche il sottotermine

Caso $n+1$ A questo livello il sottotermine e dato dai sottotermini dei passi $0...n$ e di conseguenza se questi sottotermini sono ben tipati allora lo sara anche questo. Per induzione infatti sappiamo che se applico:

SUM fdfdsf

Esercizio 3.2

Dimostrare subject reduction per induzione sulla derivazione di $\Gamma \vdash \mathrm{M} : \mathrm{T}$.

Esercizio 3.4

Vale l'opposto di subject reduction, i.e. se $\Gamma \vdash M'$:T e $M \rightarrow$ M', allora $\Gamma \vdash M : T$ (detto subject expansion)? Dimostrarlo oppure dare un controesempio.

Esercizio 3.5

Se al posto della regola (APP) si definisse la regola seguente:

$$\text{(APP')}\ \frac{\Gamma \vdash M\ :\mathrm{T} \rightarrow \mathrm{T} \qquad \Gamma \vdash N\ :\mathrm{T}}{\Gamma \vdash M\ N : T}$$

sarebbe ancora vero il teorema di safety?

Esercizio 3.6

Se al posto delle regole (APP) e (FUN) si definissero le regole seguenti regole:

$$(\text{APP'})\ \frac{\Gamma\ \vdash M\ :\text{T}\rightarrow\text{T} \qquad \Gamma\ \vdash N\ :\text{T}}{\Gamma\ \vdash M\ N : T}$$

e

$$(\text{FUN'})\ \frac{\Gamma\ ,\ \text{x} :\ T1 \vdash M\ :\text{T}}{\Gamma\ \vdash fn\ \text{x} :\ T1.M :\rightarrow \text{T}}$$

sarebbe ancora vero il teorema di safety?

Esercizio 3.6

Se si aggiungessero al sistema di tipi i seguenti due assiomi

$$(\text{TRUE'})\ \frac{}{\Gamma\ \vdash \text{true} :\ Nat}$$

e

$$(\text{FALSE'})\ \frac{}{\Gamma\ \vdash \text{false} :\ Nat}$$

sarebbe ancora vero il teorema di safety?

Esercizio 3.7

Dimostrare il seguente fatto: se $\Gamma\ \vdash M\ :\text{T}$ e derivabile allora *fv(M)* $\subseteq$ *Dom(*Γ*)*

Esercizio 3.8

Ricostruire il tipo dei seguenti termini:

- $fn\ x{:}T1.fn\ y{:}T2.if\ y$ then x else true
- $fn\ x : Nat :\rightarrow$ Bool.x
- $fn\ f : T.fn\ x : T'.f$(if true then x else $f\ x$)
- $fn\ f : T1.fn\ g : T2.$if $(f\ (g$ true))then $f\ (fn\ x$:T3.true) else $f(fn\ x : T4.x)$