

Type Theory workbook

Denis Mazzucato

May 2019

Workbook for exercises given in the Type Theory (Maietti and Sambin) course at Padua in 2019

Contents

1	Equality preservation among programs	3
2	Simmetry	4
3	Transitivity, Path Induction	4
4	Transitivity, Martin-Löf's	5
5	Propositional Equality among sum operators (Basic)	6
6	Axiom of choice	7
7	Bool terms inequality	8
8	Nat terms inequality	9
9	Peano Axioms	9
10	Propositional Equality among sum operators	9

Assunzioni

Per semplificare la scrittura degli esercizi utilizzo alcune semplificazioni, l'unico esercizio svolto per intero senza omettere alcun tipo di passaggio è la preservazione dell'uguaglianza tra programmi (1)

- Il tipo delle variabili dentro il contesto deve essere specificato solo dove è necessario. Ovvero quando la variabile è inserita per la prima volta dentro al contesto oppure durante l'utilizzo della regola *var*) che richiede di abbinare il giudizio con la variabile dentro al contesto. In qualche raro caso se compare una Γ al posto del contesto è per problemi di spazio e significa che il contesto non è mutato dal passo precedente.
- Se devo derivare un giudizio del tipo $a \in A [\Gamma, a \in A, \Delta]$ e il contesto $\Gamma, a \in A, \Delta$ è *semplice* allora concludo che riesco a derivarlo. Solo in casi dove il contesto non è banale continuo con la derivazione.
- Se il contesto è *banale* allora posso indebolirlo senza dover utilizzare le regole di indebolimento.
- Posso concludere con i tre punti verticali quando la stessa parte della dimostrazione è già stata svolta in un altro ramo differente nello stesso albero.
- Quando mi riferisco ad una *Label* la trovo definita sempre sopra il punto in cui mi trovo, per problemi di spazio alcune volte ometto di ripetere la formula e la rimpiazzo con dei puntini.

1 Equality preservation among programs

$$\mathbf{pf} \in Id(B, f(a), f(b)) \ [w \in Id(A, a, b)]$$

Date le seguenti assunzioni:

$$\pi_1) \ A \ type \ [\Gamma]$$

$$\pi_2) \ a \in A \ [\Gamma]$$

$$\pi_3) \ b \in A \ [\Gamma]$$

$$\pi_4) \ B \ type \ [\Gamma]$$

$$\pi_5) \ f(x) \in B \ [\Gamma, x \in A]$$

$$\begin{array}{c}
 \frac{\pi_5}{f(x) \in B \ [w, x \in A]} \text{I-Id)} \\
 \frac{}{id(f(x)) \in Id(B, f(x), f(x)) \ [w, x \in A]} \text{(B)} \\
 \\
 \frac{\pi_1 \quad \frac{\frac{A \ type \ [w, x]}{w, x, y \in A \ cont} \text{F-c)}}{x \in A \ [w, x \in A, y]} \text{var)} \quad \frac{\pi_1 \quad \frac{A \ type \ [w, x]}{w, x, y \in A \ cont} \text{F-c)}}{y \in A \ [w, x, y \in A]} \text{var)} \quad \frac{}{Id(A, x, y) \ type \ [w, x, y]} \text{F-Id)} \\
 \frac{}{Id(A, x, y) \ type \ [w, x, y]} \text{(D)} \\
 \\
 \frac{\frac{Id(A, x, y) \ type \ [w, x, y]}{w, x, y, z \in Id(A, x, y) \ cont} \text{F-c)}}{y \in A \ [w, x, y \in A, z]} \text{var)} \quad \frac{\pi_5 \quad \frac{Id(A, x, y) \ type \ [w, x, y]}{w, x, y, z \in Id(A, x, y) \ cont} \text{F-c)}}{f(x) \in B \ [w, x \in A]} \quad \frac{}{f(x) \in B \ [w, x, y, z]} \text{sub-ter)} \\
 \frac{}{f(y) \in B \ [w, x, y, z]} \text{(E)} \\
 \\
 \frac{\pi_4 \quad \frac{\pi_5 \quad \frac{Id(A, x, y) \ type \ [w, x, y]}{w, x, y, z \in Id(A, x, y) \ cont} \text{F-c)}}{f(x) \in B \ [w, x \in A]} \quad \frac{}{f(x) \in B \ [w, x, y, z]} \text{ind-ty)} \quad \frac{}{f(y) \in B \ [w, x, y, z]} \text{(E)} \\
 \frac{}{Id(B, f(x), f(y)) \ type \ [w, x \in A, y \in A, z \in Id(A, x, y)]} \text{F-Id)} \\
 \text{(A)} \\
 \\
 \frac{\pi_1 \quad \pi_2 \quad \pi_3}{A \ type \ [\] \quad a \in A \ [\] \quad b \in A \ [\]} \\
 \frac{}{Id(A, a, b) \ type \ [\]} \text{F-c)} \\
 \frac{}{w \in Id(A, a, b) \ cont} \text{var)} \\
 \frac{\pi_2 \quad \pi_3}{a \in A \ [w] \quad b \in A \ [w]} \quad \frac{}{w \in Id(A, a, b) \ [w \in Id(A, a, b)]} \text{(B)} \\
 \frac{}{El_{Id}(w, (x).id(f(x))) \in Id(B, f(a), f(b)) \ [w \in Id(A, a, b)]} \text{E-Id)}
 \end{array}$$

2 Simmetry

$$\mathbf{pf} \in Id(A, b, a) [w \in Id(A, a, b)]$$

Date le seguenti assunzioni: $\pi_1) A \text{ type } [\Gamma]$ $\pi_2) a \in A [\Gamma]$ $\pi_3) b \in A [\Gamma]$

$$Id(A, y, x) \text{ type } [w, x \in A, y \in A, z \in Id(A, a, b)]$$

$$(A)$$

$$\frac{\begin{array}{c} (A) \\ \dots \end{array} \quad \begin{array}{c} \pi_2 \\ a \in A [w] \end{array} \quad \begin{array}{c} \pi_3 \\ b \in A [w] \end{array} \quad w \in Id(A, a, b) [w] \quad \frac{x \in A [w, x] \quad id(x) \in Id(A, x, x) [w, x \in A]}{\text{I-Id)}}}{El_{Id}(w, id) \in Id(A, b, a) [w \in Id(A, a, b)]} \text{E-Id)}$$

3 Transitivity, Path Induction

$$\mathbf{pf} \in Id_p(A, e, g) [w_1 \in Id_p(A, e, f), w_2 \in Id_p(A, f, g)]$$

Date le seguenti assunzioni: $\pi_1) A \text{ type } [\Gamma]$ $\pi_2) e \in A [\Gamma]$ $\pi_3) f \in A [\Gamma]$ $\pi_4) g \in A [\Gamma]$

$$Id_p(A, e, y) \text{ type } [w_1, w_2, y \in A, z \in Id_p(A, f, y)]$$

$$(A)$$

$$\frac{\begin{array}{c} (A) \\ \dots \end{array} \quad \begin{array}{c} \pi_3 \\ f \in A [w_1, w_2] \end{array} \quad \begin{array}{c} \pi_6 \\ g \in A [w_1, w_2] \end{array} \quad w_2 \in Id_p(A, f, g) [w_1, w_2] \quad w_1 \in Id_p(A, e, f) [w_1, w_2]}{El_{Id_p}(w_2, w_1) \in Id_p(A, e, g) [w_1 \in Id_p(A, e, f), w_2 \in Id_p(A, f, g)]} \text{E-Id}_p)$$

4 Transitivity, Martin-Löf's

$$\mathbf{pf} \in Id(A, a, c) \ [w_1 \in Id(A, a, b), w_2 \in Id(A, b, c)]$$

Date le seguenti assunzioni:

$$\pi_1) \ A \text{ type } [\Gamma]$$

$$\pi_2) \ a \in A \ [\Gamma]$$

$$\pi_3) \ b \in A \ [\Gamma]$$

$$\pi_4) \ c \in A \ [\Gamma]$$

$$\frac{w \in Id(A, a, x) [w_1, w_2, x, w \in Id(A, a, x)]}{\lambda w. w \in Id(A, a, x) \rightarrow Id(A, a, x) [w_1, w_2, x \in A]} \text{I-}\rightarrow)$$

(B)

$$\frac{\frac{\frac{\pi_1}{A \text{ type } [\Gamma]} \quad \frac{\pi_2}{a \in A \ [\Gamma]} \quad x \in A \ [w_1, w_2, x \in A, y]}{Id(A, a, x) \text{ type } [w_1, w_2, x, y]} \quad \frac{\frac{\pi_1}{A \text{ type } [\Gamma]} \quad \frac{\pi_2}{a \in A \ [\Gamma]} \quad y \in A \ [w_1, w_2, x, y \in A]}{Id(A, a, y) \text{ type } [w_1, w_2, x, y]} \text{F-to)}}{Id(A, a, x) \rightarrow Id(A, a, y) \text{ type } [w_1, w_2, x \in A, y \in A, z \in Id(A, x, y)]} \text{(A)}$$

$$\frac{w_1 \in Id(A, a, b) \ [w_1, w_2] \quad \frac{\frac{\text{(A)} \quad \dots \quad b \in A \ [w_1, w_2] \quad c \in A \ [w_1, w_2] \quad w_2 \in Id(A, b, c) \ [w_1, w_2] \quad \text{(B)} \quad \dots}{El_{Id}(w_2, \lambda w. w) \in Id(A, a, b) \rightarrow Id(A, a, c) \ [w_1, w_2]} \text{E-Id)}}{Ap(El_{Id}(w_2, \lambda w. w), w_1) \in Id(A, a, c) \ [w_1 \in Id(A, a, b), w_2 \in Id(A, b, c)]} \text{E-}\rightarrow)$$

5 Propositional Equality among sum operators (Basic)

$$\mathbf{pf} \in Id(Nat, x' +_1 0, x' +_2 0) [x' \in Nat]$$

Per questo esercizio uguaglianze composizionali e *definizionali* sono svolte *in place*, questi passaggi saranno indicati da linee tratteggiate. Date le seguenti definizioni:

- $x' +_1 y' \equiv El_{Nat}(y', x', (x, z).succ(z))$
- $x' +_2 y' \equiv El_{Nat}(x', y', (x, z).succ(z))$

La dimostrazione è basata sulla proposizione $x +_2 0 = 0$ e la si prova induttivamente. Se vale nel caso base, quindi applicata a 0 ($0 +_2 0 = 0$), e se vale nel passo induttivo, quindi applicata al successore, allora vale per un qualsiasi x . Quando la applichiamo al passo successivo ipotizziamo di avere una prova che vale al passo precedente.

Le sostituzioni composizionali e definizionali sono specificate alla fine, con l'identificativo associato per identificarle facilmente durante la prova. Possiamo eseguire queste sostituzioni formalmente utilizzando le regole di sostituzione e conversione.

$$id(succ(\hat{x})) \in Id(Nat, succ(\hat{x}), succ(\hat{x})) [x', x, z, \hat{x} \in Nat] \quad (D)$$

$$z \in Id(Nat, x, x +_2 0) [x', x, z \in Id(Nat, x, x +_2 0)] \quad (C)$$

$$Id(Nat, succ(\hat{x}), succ(\hat{y})) \text{ type } [x', x, z, \hat{x} \in Nat, \hat{y} \in Nat, \hat{z} \in Id(Nat, \hat{x}, \hat{y})] \quad (B)$$

$$\frac{\begin{array}{c} (B) \\ \dots \quad x \in Nat [x', x, z] \quad x +_2 0 \in Nat [x', x, z] \quad \dots \quad \dots \end{array} \quad \begin{array}{c} (C) \\ \dots \end{array} \quad \begin{array}{c} (D) \\ \dots \end{array}}{El_{Id}(z, (\hat{x}).id(succ(\hat{x}))) \in Id(Nat, succ(x), succ(x +_2 0)) [x', x \in Nat, z \in Id(Nat, x, x +_2 0)]} \text{ E-Id) } (3)$$

$$El_{Id}(z, (\hat{x}).id(succ(\hat{x}))) \in Id(Nat, succ(x), succ(x) +_2 0) [x', x \in Nat, z \in Id(Nat, x, x +_2 0)] \quad (A)$$

$$\frac{x' \in Nat [x' \in Nat] \quad Id(Nat, x', x' +_2 0) \text{ type } [x'] \quad \frac{id(0) \in Id(Nat, 0, 0) [x']}{id(0) \in Id(Nat, 0, 0 +_2 0) [x']} (2) \quad (A)}{El_{Nat}(x', id(0), (x, z).El_{Id}(z, (\hat{x}).id(succ(\hat{x})))) \in Id(Nat, x', x' +_2 0) [x' \in Nat]} \dots \text{ Nat-e) } (1)$$

$$El_{Nat}(x', id(0), (x, z).El_{Id}(z, (\hat{x}).id(succ(\hat{x})))) \in Id(Nat, x' +_1 0, x' +_2 0) [x' \in Nat] \quad (1)$$

$$(1) \quad x' +_1 0 \equiv El_{Nat}(0, x', (x, z).succ(z)) = x'$$

$$(2) \quad 0 +_2 0 \equiv El_{Nat}(0, 0, (x, z).succ(z)) = 0$$

$$(3) \quad succ(x) +_2 0 \equiv El_{Nat}(succ(x), 0, (x, z).succ(z)) = succ(El_{Nat}(x, 0, (x, z).succ(z))) \equiv succ(x +_2 0)$$

6 Axiom of choice

$$(\forall x \in A)(\exists y \in B(x))C(x, y) \rightarrow (\exists f \in A \rightarrow B)(\forall x \in A)C(x, Ap(f, x)) \text{ true}$$

Per questo esercizio sono state svolte alcune semplificazioni:

- Nessun tipo è stato provato essere derivabile in quanto essendo derviabili

$$A \text{ type } [\Gamma], B(x) \text{ type } [\Gamma, x \in A] \text{ and } C(x, y) \text{ type } [\Gamma, x \in A, y \in B(x)]$$

saranno derivabili pure loro combinazioni tra somme indiciate e prodotti dipendenti

- Per non inquinare eccessivamente lo spazio delle variabili utilizzo queste convenzioni:

$$x \in A, y \in B(x), f \in \Pi_{x \in A} B(x) \text{ and } z \in \Pi_{x \in A} \Sigma_{y \in B(x)} C(x, y)$$

- La precedente assunzione può creare confusione nello *scoping* delle variabili, per questo definisco una metrica di priorità per identificare univocamente lo *scope* di una variabile:

1. *Abstraction* (la più forte)
2. *Indexed sum type* oppure *Dependent product type*
3. *Context* (il più debole)

La dimostrazione è basata nell'interpretazione intuizionistica delle costanti logiche con le sostituzioni $\forall = \Pi$ e $\exists = \Sigma$ (*propositions-as-sets* - *Curry-Howard*). Iniziamo supponendo di avere una prova della prima parte $(\forall x)(\exists y)C(x, y)$, significa che abbiamo un metodo che quando applicato ad x tiene una prova di $(\exists y)C(x, y)$. Prendiamo f come metodo che dato una arbitraria x assegna la prima componente. Quindi sia $C(x, f(x))$ che segua con la seconda componente. Abbiamo così (ri)composto l'operatore.

$$\begin{array}{c}
\text{E-II)} \frac{x \in A [z, x] \quad z \in \Pi_{x \in A} \Sigma_{y \in B(x)} C(x, y) [z, x]}{\frac{Ap(z, x) \in \Sigma_{y \in B(x)} C(x, y) [z, x]}{\pi_1(Ap(z, x)) \in B(x) [z, x]} \pi_1)} \quad \frac{x \in A [z, x] \quad z \in \Pi_{x \in A} \Sigma_{y \in B(x)} C(x, y) [z, x]}{\frac{Ap(z, x) \in \Sigma_{y \in B(x)} C(x, y) [z, x]}{\pi_2(Ap(z, x)) \in C(x, \pi_1(Ap(z, x))) [z, x]} \pi_2)} \text{E-II)} \\
\text{(A)} \qquad \qquad \qquad \text{(C)} \\
\text{(A)} \\
\frac{\frac{x \in A [z, x] \quad \pi_1(Ap(z, x)) \in B(x) [z, x]}{Ap(\lambda x. \pi_1(Ap(z, x)), x) = \pi_1(Ap(z, x)) \in B(x) [z, x]} \beta\text{C-II)} \quad \text{(C)}}{C(x, Ap(\lambda x. \pi_1(Ap(z, x)), x)) = C(x, \pi_1(Ap(z, x))) \text{ type } [z, x]} \text{sub)} \\
\frac{\pi_2(Ap(z, x)) \in C(x, \pi_1(Ap(z, x))) [z, x]}{\pi_2(Ap(z, x)) \in C(x, Ap(\lambda x. \pi_1(Ap(z, x)), x)) [z, x \in A]} \text{conv)} \\
\frac{\pi_2(Ap(z, x)) \in C(x, Ap(\lambda x. \pi_1(Ap(z, x)), x)) [z, x \in A]}{\lambda x. \pi_2(Ap(z, x)) \in \Pi_{x \in A} C(x, Ap(\lambda x. \pi_1(Ap(z, x)), x)) [z]} \text{I-II)} \\
\text{(B)} \\
\text{(A)} \\
\frac{\pi_1(Ap(z, x)) \in B(x) [z, x \in A]}{\lambda x. \pi_1(Ap(z, x)) \in \Pi_{x \in A} B(x) [z]} \text{I-II)} \quad \text{(B)} \\
\frac{\lambda x. \pi_2(Ap(z, x)) \in \Pi_{x \in A} C(x, Ap(\lambda x. \pi_1(Ap(z, x)), x)) [z]}{\langle \lambda x. \pi_1(Ap(z, x)), \lambda x. \pi_2(Ap(z, x)) \rangle \in \Sigma_{f \in \Pi_{x \in A} B(x)} \Pi_{x \in A} C(x, Ap(f, x)) [z \in \Pi_{x \in A} \Sigma_{y \in B(x)} C(x, y)]} \text{I-}\Sigma) \\
\frac{\langle \lambda x. \pi_1(Ap(z, x)), \lambda x. \pi_2(Ap(z, x)) \rangle \in \Pi_{z \in \Pi_{x \in A} \Sigma_{y \in B(x)} C(x, y)} \Sigma_{f \in \Pi_{x \in A} B(x)} \Pi_{x \in A} C(x, Ap(f, x)) [z]}{\lambda z. \langle \lambda x. \pi_1(Ap(z, x)), \lambda x. \pi_2(Ap(z, x)) \rangle \in \Pi_{z \in \Pi_{x \in A} \Sigma_{y \in B(x)} C(x, y)} \Sigma_{f \in \Pi_{x \in A} B(x)} \Pi_{x \in A} C(x, Ap(f, x)) [z]} \text{I-II)}
\end{array}$$

7 Bool terms inequality

$$\mathbf{pf} \in Id(Bool, true, false) \rightarrow N_0 [] \quad \equiv \quad \neg(true = false)$$

Date le seguenti definizioni:

$$\begin{aligned} f &\equiv El_{Bool}(z, (x).\hat{N}_1, (x).\hat{N}_0) \in U_0 [z \in Bool] \\ p_f &\equiv El_{Id}(w, (x).id(f(x))) \in Id(U_0, f(true), f(false)) [w \in Id(Bool, true, false)] \\ k &\equiv El_{Id}(w, (x).\langle \lambda y.y, \lambda y.y \rangle) \in T(x) \leftrightarrow T(y) [x \in U_0, y \in U_0, w \in Id(U_0, x, y)] \end{aligned}$$

La dimostrazione è basata sull'utilizzo degli universi, altrimenti non potremmo mai distinguere termini diversi all'interno della nostra teoria dei tipi. La funzione f propaga elementi diversi all'interno di universi distinti per poterne poi effettuare l'uguaglianza. La funzione k invece crea l'*isomorfismo* necessario per distinguere \hat{N}_1 da \hat{N}_0 .

dovrei chiarire quando le procedure sono astrazioni oppure no per essere formale

$$\frac{\frac{\frac{\frac{El_{Id}(El_{Id}(w, (x).id(El_{Bool}(x, (x).\hat{N}_1, (x).\hat{N}_0))), (x).\langle \lambda y.y, \lambda y.y \rangle) \in T(\hat{N}_1) \leftrightarrow T(\hat{N}_0) [w]}{El_{Id}(El_{Id}(w, (x).id(f(x))), (x).\langle \lambda y.y, \lambda y.y \rangle) \in T(\hat{N}_1) \leftrightarrow T(\hat{N}_0) [w]} \quad \frac{El_{Id}(p_f(w), (x).\langle \lambda y.y, \lambda y.y \rangle) \in T(\hat{N}_1) \leftrightarrow T(\hat{N}_0) [w]}{k(\hat{N}_1, \hat{N}_0, p_f(w)) \in N_1 \leftrightarrow N_0 [w]} \quad \frac{\star \in N_1 [w]}{\pi_1(k(\hat{N}_1, \hat{N}_0, p_f(w))) \in N_1 \rightarrow N_0 [w]} \quad \frac{Ap(\pi_1(k(\hat{N}_1, \hat{N}_0, p_f(w))), \star) \in N_0 [w \in Id(Bool, true, false)]}{\lambda w.Ap(\pi_1(k(\hat{N}_1, \hat{N}_0, p_f(w))), \star) \in Id(Bool, true, false) \rightarrow N_0 []} \quad \frac{E_1 \times}{E \rightarrow}$$

$$\frac{\frac{y \in T(x') [\Gamma, y \in T(x')]}{\lambda y.y \in T(x') \rightarrow T(x') [\Gamma]} \quad \frac{y \in T(x') [\Gamma, y \in T(x')]}{\lambda y.y \in T(x') \rightarrow T(x') [\Gamma]} \quad \frac{\langle \lambda y.y, \lambda y.y \rangle \in T(x') \leftrightarrow T(x') [x, y, w, x' \in U_0]}{\lambda y.y \in T(x') \rightarrow T(x') [\Gamma]} \quad \frac{I \rightarrow}{I \times}$$

(B)

$$\frac{\frac{x' \in U_0 [x, y, w, x' \in U_0, y', z]}{T(x') \text{ type } [x, y, w, x', y', z]} \quad \frac{y' \in U_0 [x, y, w, x', y' \in U_0, z]}{T(y') \text{ type } [x, y, w, x', y', z]} \quad \frac{E\text{-Un}_0}{F \times}$$

(A)

$$\frac{\frac{\dots \quad x \in U_0 [x, y, w] \quad y \in U_0 [x, y, w] \quad w \in Id(U_0, x, y) [x, y, w] \quad \dots}{El_{Id}(w, (x).\langle \lambda y.y, \lambda y.y \rangle) \in T(x) \leftrightarrow T(y) [x \in U_0, y \in U_0, w \in Id(U_0, x, y)]} \quad \frac{(A) \quad (B)}{E\text{-Id}}$$

(k)

$$\frac{z \in Bool [z \in Bool] \quad \hat{N}_1 \in U_0 [z, x \in Bool] \quad \hat{N}_0 \in U_0 [z, x \in Bool]}{El_{Bool}(z, (x).\hat{N}_1, (x).\hat{N}_0) \in U_0 [z \in Bool]}$$

(f)

$$\frac{\text{(Equality preservation among programs)}}{El_{Id}(w, (x).id(f(x))) \in Id(U_0, f(true), f(false)) [w \in Id(Bool, true, false)]}$$

(p_f)

8 Nat terms inequality

9 Peano Axioms

10 Propositional Equality among sum operators