NOTE su TEORIA dei TIPI -2021

Dipartimento di Matematica
Università di Padova
via Trieste n. 63 - 35121 Padova, Italy

3 marzo 2021

Contents

1 Perchè studiare la teoria dei tipi

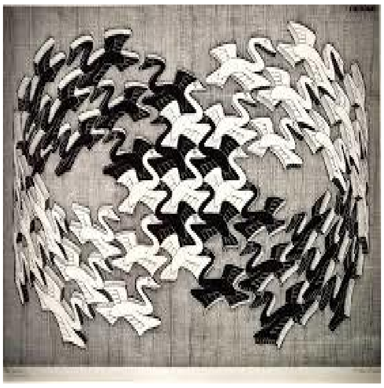
Nel seguito cercheremo di fornire un'introduzione alla teoria dei tipi dipendenti.
La principale differenza con la teoria dei tipi semplici è quella di poter rappresentare tipi di dato parametrici su elementi di altri tipi e anche interpretare le formule della logica predicativa come tipi secondo il famoso slogan *propositions-as-types*.
Per questo motivo la teoria dei tipi dipendenti è caratterizzata da una *duplice natura* perchè è nello stesso tempo:

- un linguaggio di programmazione funzionale, ovvero un lambda calcolo tipato;
- una teoria degli insiemi per formalizzare le prove in matematica.

Un'importante applicazione della teoria dei tipi dovuta alla sopra descritta duplice natura sta nel fatto che grazie alla teoria dei tipi possiamo tentare di rispondere alle seguente domande:

- Cosa significa definire una funzione *"costruttiva"* tra insiemi qualsiasi diversi dai numeri naturali?

  Si ricordi che la nozione di funzione computabile usuale è definita sui numeri naturali.
- *Quand è che una teoria degli insiemi è costruttiva?* ovvero *Quand è che tale teoria è computazionale*?
- *Quand è che una teoria degli insiemi è predicativa?* ovvero *Quand è che i suoi enti sono tutti induttivamente generabili*?



1.1 La motivazione di un matematico vincitore di una medaglia Fields

Estratto da **How I became interested in foundations of mathematics.** by **Vladimir Voedvosky**.

... Many of us do mathematics that is a little like the Rubik Cube. There is a problem. And there is the search for a solution. And when the solution is found it is certain that it is a solution.
But mathematics which earned me the Fields Medal at the International Congress of Mathematicians in Beijing in 2002 is very different.
There is a problem. And there is the search for a solution. But when the solution is found it is not certain at all that it is a solution.
The Fields Medal was awarded to me for the proof of Milnor's Conjecture.
.....

The ideas which the proof was based on turned out to be solid and the results of other people which I relied on turned out to be correct.
This is not always the case.
Let me tell you the story of another of my proofs which turned out very differently.
....

And then in the Fall of 2013, less than a year ago, some sort of a block in my mind collapsed and I suddenly understood that Carlos Simpson was correct and that the proof which Kapranov and I published in 1991 is wrong.
Not only the proof was wrong but the main theorem of that paper was false!

MORAL:
Now let us look at this story again. Kapranov and I have found a solution to the problem which we worked on - the proof of the theorem.
If the problem was to solve an equation and we would have found a solution we would have checked that it is a solution before publishing it, right? And if it were a complex equation we would probably have checked it on a computer.
So why can not we check a solution which is a proof of a theorem?
For proof verification we need to construct a *particular* **formal deduction system** and explain how it corresponds with the mathematical objects and forms of reasoning which exist in our thoughts.
....
Encoding of statements and proofs which exist in our thoughts into *symbolic expressions* is called *formalization.*
Formalization is, just like programming, first of all a tool that we can use to pass on to computers some of the mental tasks which we need to perform.

2 I giudizi della teoria dei tipi dipendenti di Martin-Löf

La teoria dei tipi, detta $\mathbf{ML}_0$, è un frammento della teoria dei tipi di Martin-Löf con un tipo ristretto di uguaglianza proposizionale.
È descritta tramite quattro tipologie diverse di giudizi

$$A\ type\ [\Gamma] \qquad A = B\ type\ [\Gamma] \qquad a \in A\ [\Gamma] \qquad a = b \in A\ [\Gamma]$$

ove

$A\ type\ [\Gamma]$ indica quando qualcosa è *un tipo possibilmente dipendente dal contesto* Γ,

$A = B\ type\ [\Gamma]$ indica quando *due tipi possibilmente dipendenti dal contesto* Γ *sono uguali*

$a \in A\ [\Gamma]$ indica quando qualcosa è *un elemento di un tipo possibilmente dipendente dal contesto* Γ

$a = b \in A\ [\Gamma]$ indica quando *due termini di un tipo possibilmente dipendente dal contesto* Γ *sono uguali* in modo **definizionale** o **computazionale**. Questo difatti si chiama **giudizio** di **uguaglianza definizionale di termini**.

Per semplificare la presentazione delle regole di formazione dei tipi e dei loro termini e rispettive uguaglianze si formula la teoria dei tipi facendo uso di un quinto giudizio ausiliario

$$\Gamma\ cont$$

che afferma che il simbolo Γ rappresenta un contesto, ovvero una lista di variabili tipate da utilizzare per la formazioni dei quattro giudizi enunciati sopra.

Nota Bene:
il giudizio dell'uguaglianza tra termini
$a = b \in A\ [\Gamma]$ NON è un tipo
e nel seguito vedremo come collegare questo giudizio con uno specifico **tipo uguaglianza**....

2.1 Regole strutturali in teoria dei tipi

Enunciamo di seguito le regole cosiddette *strutturali* della teoria dei tipi dipendenti perchè presenti in ogni teoria dei tipi dipendenti che tratteremo di seguito indipendentemente dai tipi specifici aggiunti.

2.1.1 Regole di formazione dei contesti

$$[\,]\ cont \qquad\qquad \text{F-c}\frac{A\ type\ [\Gamma]}{\Gamma, x \in A\ cont}\ ((x \in A) \text{ not in } \Gamma)$$

2.1.2 Regola di assunzione delle variabili

$$\text{var)}\ \frac{\Gamma, x \in A, \Delta \quad cont}{x \in A\ [\Gamma, x \in A, \Delta]}$$

2.1.3 Regole strutturali addizionali sull'uguaglianza

Certamente in una teoria dei tipi l'*uguaglianza tra tipi* espressa nel giudizio relativo, *deve essere una relazione d'equivalenza* e perciò nella nostra teoria dei tipi anche le seguenti regole devono essere valide

$$\text{ref)}\ \frac{A\ type\ [\Gamma]}{A = A\ type\ [\Gamma]} \qquad\qquad \text{sym)}\ \frac{A = B\ type\ [\Gamma]}{B = A\ type\ [\Gamma]}$$

$$\text{tra)}\ \frac{A = B\ type\ [\Gamma] \quad B = C\ type\ [\Gamma]}{A = C\ type\ [\Gamma]}$$

Allo stesso modo anche *l'uguaglianza definizionale (o computazionale) tra termini* espressa nel giudizio relativo, *deve essere una relazione d'equivalenza* ovvero le seguenti regole devono essere valide

$$\text{ref)}\ \frac{a \in A\ [\Gamma]}{a = a \in A\ [\Gamma]} \qquad\qquad \text{sym)}\ \frac{a = b \in A\ [\Gamma]}{b = a \in A\ [\Gamma]}$$

$$\text{tra)}\ \frac{a = b \in A\ [\Gamma] \qquad b = c \in A\ [\Gamma]}{a = c \in A\ [\Gamma]}$$

2.1.4 Regole di conversione dell'uguaglianza per tipi uguali

Per esprimere che l'appartenenza si conserva con l'uguaglianza di termini e di tipi aggiungiamo alla teoria dei tipi le seguenti regole

$$\text{conv)}\ \frac{a \in A\ [\Gamma] \qquad A = B\ type\ [\Gamma]}{a \in B\ [\Gamma]}$$

$$\text{conv-eq)}\ \frac{a = b \in A\ [\Gamma] \qquad A = B\ type\ [\Gamma]}{a = b \in B\ [\Gamma]}$$

2.2 Alcune regole ammissibili: regole di sostituzione, indebolimento e scambio

In genere nel formulare una teoria dei tipi conviene utilizzare il minimo numero possibile di regole strutturali e di formazione di tipi e termini in modo tale però che la teoria sia chiusa su alcune regole irrinunciabili quali ad esempio le regole di indebolimento e di sostituzione.
Grazie alla forma delle regole che forniremo per costruire tipi e termini non sarà necessario aggiungere esplicitamente le regole di indebolimento e sostituzioni che elenchiamo qui di seguito perchè la loro validità sarà comunque garantita (vedi osservazione 3.3) e quindi tali regole potranno essere utilizzate nelle derivazioni in ogni teoria dei tipi che presenteremo.
Ricordiamo la seguente definizione di regola ammissibile (o derivabile):

Def. 2.1 (ammissibilità di una regola) Diciamo che una regola formulata con i giudizi della teoria dei tipi è *ammissibile* (o *derivabile*) nella teoria dei tipi $\mathcal{T}$ se e solo se nel caso i suoi giudizi premessa sono derivabili in $\mathcal{T}$ allora anche il giudizio conclusione è pure derivabile in $\mathcal{T}$.

2.2.1 Le regole di indebolimento

$$\text{ind-ty)}\ \frac{A\ type\ [\Gamma] \qquad\qquad \Gamma, \Delta\ cont}{A\ type\ [\Gamma, \Delta]}$$

$$\text{ind-ty-eq)}\ \frac{A = B\ type\ [\Gamma] \qquad\qquad \Gamma, \Delta\ cont}{A = B\ type\ [\Gamma, \Delta]}$$

$$\text{ind-te)}\ \frac{a \in A\ [\Gamma] \qquad\qquad \Gamma, \Delta\ cont}{a \in A\ [\Gamma, \Delta]}$$

$$\text{ind-te-eq)}\ \frac{a = b \in A\ [\Gamma] \qquad\qquad \Gamma, \Delta\ cont}{a = b \in A\ [\Gamma, \Delta]}$$

2.2.2 Regole di sostituzione

In un calcolo di tipi dipendenti dobbiamo per forza operare sostituzioni di termini in tipi, termini e loro uguaglianza quando questi sono dipendenti da certi altri tipi. Perciò il calcolo deve essere chiuso sulle seguenti regole di sostituzione:

1.

$$\text{sub-typ)}\ \frac{\begin{array}{l} C(x_1, \ldots, x_n)\ \ type\ [\Gamma,\, x_1 \in A_1,\, \ldots,\, x_n \in A_n(x_1, \ldots, x_{n-1})\,] \\ a_1 \in A_1\ [\Gamma]\ \ldots\ a_n \in A_n(a_1, \ldots, a_{n-1})\ [\Gamma] \end{array}}{C(a_1, \ldots, a_n)\ type\ [\Gamma]}$$

2.

$$\text{sub-eq-typ)}\ \frac{\begin{array}{l} C(x_1,\ldots,x_n)\ \ type\ [\Gamma,\, x_1\in A_1,\,\ldots,\, x_n\in A_n(x_1,\ldots,x_{n-1})\,] \\ a_1=b_1\in A_1\ \ \ldots\ \ a_n=b_n\in A_n(a_1,\ldots,a_{n-1}) \end{array}}{C(a_1,\ldots,a_n)=C(b_1,\ldots,b_n)\ type\ [\Gamma]}$$

3.

$$\text{sub-Eqtyp)}\ \frac{\begin{array}{l} C(x_1,\ldots,x_n)=D(x_1,\ldots,x_n)\ \ type\ [\Gamma,\, x_1\in A_1,\,\ldots,\, x_n\in A_n(x_1,\ldots,x_{n-1})\,] \\ a_1\in A_1\ [\Gamma]\ldots\ a_n\in A_n(a_1,\ldots,a_{n-1})\ [\Gamma] \end{array}}{C(a_1,\ldots,a_n)=D(a_1,\ldots,a_n)\ type\ [\Gamma]}$$

4.

$$\text{sub-eq-Eqtyp)}\ \frac{\begin{array}{l} C(x_1,\ldots,x_n)=D(x_1,\ldots,x_n)\ \ type\ [\Gamma,\, x_1\in A_1,\,\ldots,\, x_n\in A_n(x_1,\ldots,x_{n-1})\,] \\ a_1=b_1\in A_1\ [\Gamma]\ldots\ a_n=b_n\in A_n(a_1,\ldots,a_{n-1})\ [\Gamma] \end{array}}{C(a_1,\ldots,a_n)=D(b_1,\ldots,b_n)\ type\ [\Gamma]}$$

5.

$$\text{sub-ter)}\ \frac{\begin{array}{l} c(x_1,\ldots,x_n)\in C(x_1,\ldots,x_n)\ \ [\Gamma,\, x_1\in A_1,\,\ldots,\, x_n\in A_n(x_1,\ldots,x_{n-1})\,] \\ a_1\in A_1\ [\Gamma]\ \ldots\ a_n\in A_n(a_1,\ldots,a_{n-1})\ [\Gamma] \end{array}}{c(a_1,\ldots,a_n)\in C(a_1,\ldots,a_n)\ [\Gamma]}$$

6.

$$\text{sub-eqter)}\ \frac{\begin{array}{l} c(x_1,\ldots,x_n)=d(x_1,\ldots,x_n)\in C(x_1,\ldots,x_n)\ \ [\Gamma,\, x_1\in A_1,\,\ldots,\, x_n\in A_n(x_1,\ldots,x_{n-1})\,] \\ a_1\in A_1\ [\Gamma]\ \ldots\ a_n\in A_n(a_1,\ldots,a_{n-1})\ [\Gamma] \end{array}}{c(a_1,\ldots,a_n)=d(a_1,\ldots,a_n)\in C(a_1,\ldots,a_n)\ [\Gamma]}$$

7.

$$\text{sub-eq-ter)}\ \frac{\begin{array}{l} c(x_1,\ldots,x_n)\in C(x_1,\ldots,x_n)\ \ [\Gamma,\, x_1\in A_1,\,\ldots,\, x_n\in A_n(x_1,\ldots,x_{n-1})\,] \\ a_1=b_1\in A_1\ [\Gamma]\ \ldots\ a_n=b_n\in A_n(a_1,\ldots,a_{n-1})\ [\Gamma] \end{array}}{c(a_1,\ldots,a_n)=c(b_1,\ldots,b_n)\in C(a_1,\ldots,a_n)\ [\Gamma]}$$

8.

$$\text{sub-eq-eqter)}\ \frac{\begin{array}{l} c(x_1,\ldots,x_n)=d(x_1,\ldots,x_n)\in C(x_1,\ldots,x_n)\ \ [\Gamma,\, x_1\in A_1,\,\ldots,\, x_n\in A_n(x_1,\ldots,x_{n-1})\,] \\ a_1=b_1\in A_1\ [\Gamma]\ \ldots\ a_n=b_n\in A_n(a_1,\ldots,a_{n-1})\ [\Gamma] \end{array}}{c(a_1,\ldots,a_n)=d(b_1,\ldots,b_n)\in C(a_1,\ldots,a_n)\ [\Gamma]}$$

2.2.3 Regole di scambio

La regola di scambio di assunzioni in un contesto in teoria dei tipi dipendenti NON è generalmente derivabile per la dipendenza del tipo di un'assunzione in un contesto Γ dalle assunzioni in Γ che lo precedono. Possiamo però dimostrare una forma ristretta di regola di scambio come segue

$$\text{ex-ty)}\ \frac{A\ type\ [\Gamma, x \in C, y \in D, \Delta] \qquad\qquad \Gamma, y \in D, x \in C, \Delta\ cont}{A\ type\ [\Gamma, y \in D, x \in C, \Delta]}$$

$$\text{ex-ty-eq)}\ \frac{A = B\ type\ [\Gamma, x \in C, y \in D, \Delta] \qquad\qquad \Gamma, y \in D, x \in C, \Delta\ cont}{A = B\ type\ [\Gamma, y \in D, x \in C, \Delta]}$$

$$\text{ex-te)}\ \frac{a \in A\ [\Gamma, x \in C, y \in D, \Delta] \qquad\qquad \Gamma, y \in D, x \in C, \Delta\ cont}{a \in A\ \ [\Gamma, y \in D, x \in C, \Delta]}$$

$$\text{ex-te-eq)}\ \frac{a \in A\ [\Gamma, x \in C, y \in D, \Delta] \qquad\qquad \Gamma, y \in D, x \in C, \Delta\ cont}{a \in A\ \ [\Gamma, y \in D, x \in C, \Delta]}$$

2.2.4 Regole di sanitary checks sulla correttezza del tipaggio

In una teoria dei tipi devono inoltre valere le seguenti proprietà di *sanitary check* per assicurare la correttezza del tipaggio dei termini:

1. Se $[\Gamma, \Delta]$ *cont* è derivabile allora $[\Gamma]$ *cont* è pure derivabile;
2. se A *type* $[\Gamma]$ è derivabile allora $[\Gamma]$ *cont* è pure derivabile;
3. se $a \in A$ $[\Gamma]$ è derivabile allora A *type* $[\Gamma]$ è pure derivabile;
4. se $A = B$ *type* $[\Gamma]$ è derivabile allora sia A *type* $[\Gamma]$ che B *type* $[\Gamma]$ sono derivabili;
5. se $a = b \in A$ $[\Gamma]$ è derivabile allora sia $a \in A$ $[\Gamma]$ che $b \in A$ $[\Gamma]$ sono derivabili.

2.3 Quali regole per generare tipi aggiungiamo?

L'idea nel definire le regole di un tipo

$$A(x_1, \dots, x_n)\ \mathit{type}\ [x_1 \in B_1, \dots, x_n \in B_n]$$

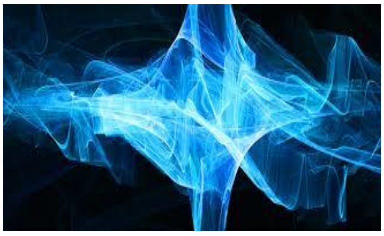
ovvero di un tipo $A(x_1, \dots, x_n)$ dipendente da

$$\Gamma \equiv x_1 \in B_1, \dots, x_n \in B_n$$

è quella di

1. descrivere la formazione del tipo o come tipo costante o come costruttore a partire da altri tipi;
2. descrivere i suoi *elementi canonici* tramite regole di **introduzione**;
3. descrivere i termini definibili su elementi del tipo $A(x_1, \dots, x_n)\ \mathit{type}\ [x_1 \in B_1, \dots, x_n \in B_n]$ ma a valori in altro tipo *dipendente*
come *estensioni* definite per **induzione** a partire da termini definiti solo sugli elementi canonici del tipo $A(x_1, \dots, x_n)\ \mathit{type}\ [x_1 \in B_1, \dots, x_n \in B_n]$ tramite le cosiddette regole di **eliminazione**;
4. descrivere le sue regole di computazione tramite regole di **conversione**;
5. garantire che i costruttori del tipo e dei suoi termini conservano l'uguaglianza tra tipi e l'uguaglianza deinizionale tra termini.

Un'importante differenza tra la teoria dei tipi dipendenti e quella semplice è che i tipi semplici sono dotati di regole di eliminazione che rappresentano **definizioni di termini per ricorsione** a partire dai loro elementi canonici, mentre gli stessi tipi semplici visti all'interno di una teoria dei tipi dipendenti vengono definiti *con le stesse regole di introduzione* ma con regole di eliminazione in genere più espressive ovvero che rappresentano **definizioni di termini per induzione** a partire dai loro elementi canonici.



3 Costruzione di tipi

Nel seguito andiamo a descrivere le regole del tipo base del singoletto e di altri tipi composti.

Nello scrivere i costruttori di eliminazione per un tipo adottiamo l'*higher-order syntax*, ovvero una sorta di lambda calcolo di ordine superiore, in
B. Nordström, K. Petersson, and J. Smith. Programming in Martin Löf's Type Theory. Clarendon Press, Oxford, 1990.
Tal notazione verrà spiegata per quanto basta ad utilizzarla nella derivazione dei giudizi.

3.1 Tipo singoletto ed esercizi

Il tipo che contiene un solo elemento (singoletto) è definito dalle seguenti regole:
Singleton

F-S) $\dfrac{\Gamma\ cont}{\mathsf{N}_1\ type\ [\Gamma]}$ I-S) $\dfrac{\Gamma\ cont}{\star \in \mathsf{N}_1\ [\Gamma]}$

E-S) $\dfrac{t \in \mathsf{N}_1\ [\Gamma] \qquad M(z)\ type\ [\Gamma, z \in \mathsf{N}_1] \qquad c \in M(\star)\ [\Gamma]}{El_{\mathsf{N}_1}(t, c) \in M(t)\ [\Gamma]}$

C-S) $\dfrac{M(z)\ type\ [\Gamma, z \in \mathsf{N}_1] \qquad c \in M(\star)\ [\Gamma]}{El_{\mathsf{N}_1}(\star, c) = c \in M(\star)\ [\Gamma]}$

eq-E-S) $\dfrac{t = s \in \mathsf{N}_1\ [\Gamma] \qquad M(z)\ type\ [\Gamma, z \in \mathsf{N}_1] \qquad c = d \in M(\star)\ [\Gamma]}{El_{\mathsf{N}_1}(t, c) = El_{\mathsf{N}_1}(s, d) \in M(t)\ [\Gamma]}$

Remark 3.1 La scrittura $\mathbf{M}(\mathbf{z})$ in

$$\mathbf{M}(\mathbf{z})\mathbf{type}\ [\boldsymbol{\Gamma}, \mathbf{z} \in \mathsf{N}_1]$$

è una *metavariabile per un tipo che può dipendere da* $\mathbf{z}$ *ma anche può NON dipendere da* $\mathbf{z}$.
Quindi la scrittura $\mathbf{M}(\mathbf{t})$ indica l'effetto della sostituzione in $\mathbf{M}(\mathbf{z})$ di $\mathbf{z}$ con $\mathbf{t}$, ovvero

$$\mathbf{M}(\mathbf{t}) \text{ sta per } \mathbf{M}(\mathbf{z})[\mathbf{z}/\mathbf{t}] \equiv \mathbf{M}(\mathbf{t})$$

Rigorosamente dovremmo scrivere l'eliminazione in tal modo:

$$\text{E-}\mathsf{N}_1)_{sub}\ \dfrac{t \in \mathsf{N}_1\ [\Gamma] \qquad D\ type\ [\Gamma, z \in \mathsf{N}_1] \qquad c \in D[z/\star]\ [\Gamma]}{El_{\mathsf{N}_1}(t, c) \in D[z/t]\ [\Gamma]}$$

Nella pratica della descrizione in testi di teoria dei tipi si usa la scrittura $\mathbf{M}(\mathbf{z})$ per agevolare la lettura e la comprensione.
È invece meglio usare la formulazione della regola E-$\mathsf{N}_1)_{sub}$ in programmazione.

Esercizi:
Sia T_1 la teoria dei tipi definita dalle regole del tipo singoletto e le regole strutturali descritte nelle sezioni precedenti incluse quelle di sostituzione e indebolimento.

1. Stabilire se i seguenti termini sono tipabili come termini del tipo singoletto secondo la teoria dei tipi T_1 e quali sono uguali definizionalmente:

$$\mathbf{El}_{N_1}(*,*) \qquad \mathbf{El}_{N_1}(x,*) \qquad \mathbf{El}_{N_1}(*,y)$$
$$\mathbf{El}_{N_1}(x,y) \qquad \mathbf{El}_{N_1}(\ \mathbf{El}_{N_1}(*,y)\ ,\ \mathbf{El}_{N_1}(x,*)\ )$$

2. Dimostrare che in T_1 è derivabile la regola

$$\text{E-}\mathsf{N1}_{prog})\ \frac{D(w)\ type\ [\Sigma, w \in \mathsf{N}_1] \qquad d \in D(*)\ [\Sigma]}{\mathsf{El}_{\mathsf{N}_1}(w,d) \in D(w)\ [\Sigma, w \in \mathsf{N}_1]}$$

ovvero dimostrare che se i giudizi premessa sono derivabili in T_1, allora lo è anche il giudizio conclusione.

3. Dimostrare che la regola E-S) è derivabile nella teoria dei tipi ottenuta da T_1 rimpiazzando la regola di eliminazione E-S) con la regola E-$\mathsf{N1}_{prog}$) e aggiungendo le regole di sostituzione e indebolimento e quelle di sanitary checks esposte nelle precedenti sezioni.

4. Si definiscano *termini untyped* con i costruttori del tipo singoletto secondo la seguente grammatica:

$$t\ \equiv\ v\ \mid\ *\ \mid\ \mathsf{El}_{\mathsf{N}_1}(t_1,t_2)$$

con $v \in \{\ x,y,w,z\ \} \bigcup \{x_i\ \mid\ i \in Nat\}$ ovvero consideriamo come variabili le ultime lettere dell'alfabeto inglese e poi tutte le variabili ottenute ponendo alla variabile x un indice con indice che varia nei numeri naturali.

Sia $\to_1$ una relazione binaria tra questi termini *untyped* definita a partire dalle seguenti regole:

$$\beta_{\mathsf{N}_1} red) \qquad \mathsf{El}_{\mathsf{N}_1}(*,c) \quad \to_1 \quad c$$

$$\text{red}_I)\ \frac{t_1\ \to_1\ t_2}{\mathsf{El}_{\mathsf{N}_1}(t_1,c)\ \to_1\ \mathsf{El}_{\mathsf{N}_1}(t_2,c)} \qquad\qquad \text{red}_{II})\ \frac{c_1\ \to_1\ c_2}{\mathsf{El}_{\mathsf{N}_1}(t,c_1)\ \to_1\ \mathsf{El}_{\mathsf{N}_1}(t,c_2)}$$

(a) Costruire l'albero dei cammini (ovvero sequenze) di passi di riduzione possibili fino a un termine *in forma normale*, ovvero *NON ulteriormente riducibile* rispetto alla relazione $\to_1$ del termine

$$\mathsf{El}_{\mathsf{N}_1}(\ \mathsf{El}_{\mathsf{N}_1}(*,*)\ ,\ \mathsf{El}_{\mathsf{N}_1}(*,x)\ )$$

(b) Produrre un'infinità di termini del tipo singoletto che non sono riducibili secondo la relazione di un passo di riduzione $\to_1$.

Dati due di questi termini, si riesce a dire che sono definizionalmente uguali secondo le regole del tipo singoletto?

3.2 Il tipo dei Numeri Naturali

Enunciamo le regole del tipo dei Numeri Naturali e dei suoi termini e uguaglianze:

Tipo dei Numeri Naturali

F-Nat $\dfrac{\Gamma\ cont}{\mathsf{Nat}\ type\ [\Gamma]}$

I$_1$-Nat) $\dfrac{\Gamma\ cont}{0 \in \mathsf{Nat}\ [\Gamma]}$ I$_2$-Nat) $\dfrac{m \in \mathsf{Nat}\ [\Gamma]}{\mathsf{succ}(m) \in \mathsf{Nat}\ [\Gamma]}$

E-Nat) $\dfrac{t \in \mathsf{Nat}\ [\Gamma] \quad M(z)\ type\ [\Gamma, z \in \mathsf{Nat}] \quad c \in M(0)\ [\Gamma] \quad e(x,y) \in M(\mathsf{succ}(x))\ [\Gamma, x \in \mathsf{Nat}, y \in M(x)]}{\mathsf{El}_{\mathsf{Nat}}(t, c,\ (x,y).e(x,y)\ ) \in M(t)\ [\Gamma]}$

C$_1$-Nat) $\dfrac{M(z)\ type\ [\Gamma, z \in \mathsf{Nat}] \quad c \in M(0)\ [\Gamma] \quad e(x,y) \in M(\mathsf{succ}(x))\ [\Gamma, x \in \mathsf{Nat}, y \in M(x)]}{\mathsf{El}_{\mathsf{Nat}}(0, c,\ (x,y).e(x,y)\ ) = c \in M(0)\ [\Gamma]}$

C$_2$-Nat) $\dfrac{m \in \mathsf{Nat}\ [\Gamma] \quad M(z)\ type\ [\Gamma, z \in \mathsf{Nat}] \quad c \in M(0)\ [\Gamma] \quad e(x,y) \in M(\mathsf{succ}(x))\ [\Gamma, x \in \mathsf{Nat}, y \in M(x)]}{\mathsf{El}_{\mathsf{Nat}}(\mathsf{succ}(m), c,\ (x,y).e(x,y)\ ) = e(\ m\ , \mathsf{El}_{\mathsf{Nat}}(m, c,\ (x,y).e(x,y)\ )\ ) \in M(\mathsf{succ}(m))\ [\Gamma]}$

Remark 3.2 (sulla notazione nelle regole di eliminazione) Come già detto all'inizio della sezione, nello scrivere le costanti di eliminazione adottiamo la teoria delle espressioni della sintassi di ordine superiore secondo cui nel termine $\mathsf{El}_{\mathsf{Nat}}(\mathbf{w}, \mathbf{c}, (\mathbf{x}, \mathbf{y}).\mathbf{e}(\mathbf{x}, \mathbf{y})\ \ ,)$ compare il termine dipendente $e(x,y) \in M(\mathsf{succ}(x))\ [\Gamma, x \in \mathsf{Nat}, y \in M(x)]$ vincolato su x e su y e rappresentato dalla scrittura $(\mathbf{x}, \mathbf{y}).\mathbf{e}(\mathbf{x}, \mathbf{y})$ che viene abbreviata anche semplicemente alla costante $\mathbf{e}$ quando è chiaro quando le variabili dipendenti sono note.
Quindi nel seguito scriviamo semplicemente:

$$\mathsf{El}_{\mathsf{Nat}}(\mathbf{w},\ \mathbf{c},\ \mathbf{e}\ )$$

La stessa abbreviazione verrà adottata anche nella scrittura degli eliminatori degli altri tipi della teoria di Martin-Löf che verranno descritti in seguito.

Aggiungiamo pure le regole di uguaglianze che assicurano che i costruttori dei termini conservano le uguaglianze definizionali:

eq-I$_2$-Nat) $\dfrac{m_1 = m_1 \in \mathsf{Nat}\ [\Gamma]}{\mathsf{succ}(m_1) = \mathsf{succ}(m_2) \in \mathsf{Nat}\ [\Gamma]}$

eq-E-Nat) $\dfrac{\begin{array}{l} M(z)\ type\ [\Gamma, z \in \mathsf{Nat}] \qquad t_1 = t_2 \in \mathsf{Nat}\ [\Gamma] \qquad c_1 = c_2 \in M(0)\ [\Gamma] \\ e_1(x,y) = e_2(x,y) \in M(\mathsf{succ}(x))\ [\Gamma, x \in \mathsf{Nat}, y \in M(x)] \end{array}}{\mathsf{El}_{\mathsf{Nat}}(t_1, c_1, e_1) = \mathsf{El}_{\mathsf{Nat}}(t_2, c_2, e_2) \in M(t_1)\ [\Gamma]}$

Remark 3.3 (sulla formulazione delle regole) Si noti che tutte le regole di formazioni dei tipi e dei loro termini sono formulate in modo da rendere le *regole di sostituzioni per tipi e termini ammissibili.* Ad esempio, si potrebbe formulare la regola di introduzione del successore di un numero naturale come un esplicito nuovo *programma funzionale* visto come termine dipendente in tal modo:

$$\mathrm{I}_2\text{-}\mathsf{Nat}_{prog})\qquad \frac{\Gamma\ \ cont}{\mathsf{succ}(x)\in\mathsf{Nat}\ [\Gamma, x\in\mathsf{Nat}]}$$

o analogamente la regola di eliminazione in tal modo:

$$\text{E-}\mathsf{Nat}_{prog})\ \frac{M(z)\ type\ [\Gamma, z\in\mathsf{Nat}]\qquad c\in M(0)\ [\Gamma]\qquad e(x,y)\in M(\mathsf{succ}(x))\ [\Gamma, x\in\mathsf{Nat}, y\in M(x)]}{\mathsf{El}_{\mathsf{Nat}}(w,c,e)\in M(m)\ [\Gamma, w\in\mathsf{Nat}]}$$

Si lascia come esercizio al lettore di dimostrare che le regole $\mathrm{I}_2\text{-}\mathsf{Nat}_{prog})$ ed $\text{E-}\mathsf{Nat}_{prog})$ sono ammissibili nel sistema di teoria dei tipi con le regole dei numeri naturali enunciate sopra.
Inoltre si osservi come esercizio che, viceversa a quanto detto sopra, le regole $\mathrm{I}_2\text{-}\mathsf{Nat})$ ed $\text{E-}\mathsf{Nat})$ sono ammissibili in un un sistema di tipi $\mathcal{T}$ che include tutte le regole strutturali finora enunciate, le regole di indebolimento e di sostituzione e le regole del tipo dei numeri naturali fornite sopra in cui però le regole $\mathrm{I}_2\text{-}\mathsf{Nat})$ ed $\text{E-}\mathsf{Nat})$ sono rimpiazzate dalle regole $\mathrm{I}_2\text{-}\mathsf{Nat}_{prog})$ ed $\text{E-}\mathsf{Nat}_{prog})$.

Def. 3.4 Conveniamo di utilizzare l'abbreviazione

$$\mathbf{n}\ \equiv\ \underbrace{\mathsf{succ}(\mathsf{succ}\ldots(0))}_{\text{n-volte}}$$

e quindi per esempio

$$\mathbf{1}\ \equiv\ \mathsf{succ}(\mathbf{0})$$
$$\mathbf{2}\ \equiv\ \mathsf{succ}(\,\mathsf{succ}(\mathbf{0})\,)$$

Esercizi

1. Definire

$$\mathbf{w}+\mathbf{2}\in\mathsf{Nat}\ [\mathbf{w}\in\mathsf{Nat}]$$

ove $\mathbf{2}$ è l'abbreviazione del termine ottenuto applicando $\mathbf{2}\equiv\mathsf{succ}(\mathsf{succ}(\mathbf{0}))$.

2. Derivare con le regole di teoria dei tipi finora enunciate i termini che seguono:

$$\mathsf{El}_{\mathsf{Nat}}(w, y, (x, z).\mathsf{succ}(x)) \in \mathsf{Nat}\ [w \in \mathsf{Nat}, y \in \mathsf{Nat}]$$
$$\mathsf{El}_{\mathsf{Nat}}(y, w, (x, z).\mathsf{succ}(x)) \in \mathsf{Nat}\ [w \in \mathsf{Nat}, y \in \mathsf{Nat}]$$
$$\mathsf{El}_{\mathsf{Nat}}(w, y, (x, z).\mathsf{succ}(y)) \in \mathsf{Nat}\ [w \in \mathsf{Nat}, y \in \mathsf{Nat}]$$
$$\mathsf{El}_{\mathsf{Nat}}(w, y, (x, z).\mathsf{succ}(w)) \in \mathsf{Nat}\ [w \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

Spiegare inoltre che funzioni semantiche tra numeri naturali denotano tali termini, ovvero quali funzioni denotano una volta che le variabili w e y sono sostituite dai cosiddetti numerali, ovvero dai termini senza variabili di Nat ottenuti applicando le sole regole di introduzione del tipo dei numeri naturali (tali termini sono i *valori del tipo dei numeri naturali* della semantica operazionale che si può definire per i termini in teoria dei tipi).

3. Definire l'operazione di addizione usando le regole del tipo dei numeri naturali

$$x + y \in \mathsf{Nat}\ [x \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

in modo tale che valga $x + 0 = x \in \mathsf{Nat}\ [x \in \mathsf{Nat}]$.

4. Definire l'operazione di addizione usando le regole del tipo dei numeri naturali

$$x + y \in \mathsf{Nat}\ [x \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

in modo tale che valga $0 + x = x \in \mathsf{Nat}\ [x \in \mathsf{Nat}]$.

5. Definire l'operazione di moltiplicazione usando le regole del tipo dei numeri naturali

$$x \cdot y \in \mathsf{Nat}\ [x \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

in modo tale che valga $x \cdot 0 = 0 \in \mathsf{Nat}\ [x \in \mathsf{Nat}]$.

6. Si definisca la funzione predecessore

$$\mathbf{p}(x) \in \mathsf{Nat}\ [x \in \mathsf{Nat}]$$

tale che

$$\mathbf{p}(0) = 0$$
$$\mathbf{p}(\mathsf{succ}(\mathbf{n})) = \mathbf{n}$$

7. Definire un'operazione binaria $bin(x, y)$ usando le regole del tipo dei numeri naturali

$$bin(x, y) \in \mathsf{Nat}\ [x \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

in modo tale che valga $bin(x, 0) = 0 \in \mathsf{Nat}\ [x \in \mathsf{Nat}]$.

8. Si definiscano *termini untyped* con i costruttori del tipo dei numeri naturali secondo la seguente grammatica:

$$t \ \equiv\ v \ \mid\ 0 \ \mid\ \mathsf{succ}(t) \ \mid\ \mathsf{El}_{\mathsf{Nat}}(t_1, t_2, (x, y).t_3(x, y))$$

con $v \in \{\ x, y, w, z\ \} \bigcup \{x_i \ \mid\ i \in Nat\}$ ovvero consideriamo come variabili le ultime lettere dell'alfabeto inglese e poi tutte le variabili ottenute ponendo alla variabile x un indice con indice che varia nei numeri naturali.

Definiamo la *riduzione ad un passo di computazione* $\to_1$ sui termini untyped definiti sopra, convenendo di usare la seguente abbreviazione

$$\mathsf{El}_{\mathsf{Nat}}(t_1, t_2, t_3) \text{ al posto di } \mathsf{El}_{\mathsf{Nat}}(t_1, t_2,\ (x, y).t_3(x, y)\ )$$

tramite i seguenti assiomi:

$$\begin{array}{llcl} \beta_{1\mathsf{Nat}}\text{-red}) & \mathsf{El}_{\mathsf{Nat}}(0, c, e\ ) & \to_1 & c \\ \beta_{2\mathsf{Nat}}\text{-red}) & \mathsf{El}_{\mathsf{Nat}}(\mathsf{succ}(m), c, e\ ) & \to_1 & e(\ m, \mathsf{El}_{\mathsf{Nat}}(m, c, e\ )\ ) \end{array}$$

e le seguenti regole:

I-red) $\dfrac{t_1 \ \to_1\ t_2}{\mathsf{succ}(t_1)\ \to_1\ \mathsf{succ}(t_2)}$

red$_I$) $\dfrac{t_1 \ \to_1\ t_2}{\mathsf{El}_{\mathsf{Nat}}(t_1, c, e)\ \to_1\ \mathsf{El}_{\mathsf{Nat}}(t_2, c, e)}$ red$_{II}$) $\dfrac{c_1 \ \to_1\ c_2}{\mathsf{El}_{\mathsf{Nat}}(t, c_1, e)\ \to_1\ \mathsf{El}_{\mathsf{Nat}}(t, c_2, e)}$

(a) Si associ ai seguenti termini una strategia di riduzione tramite un numero finito (anche zero) di applicazioni di $\to_1$ fino ad ottenere un termine non ulteriormente riducibile:

$\mathsf{succ}(\ \mathsf{El}_{\mathsf{Nat}}(\ 0, 3, (x, y).\mathsf{succ}(y)\ )\ )$ $\mathsf{El}_{\mathsf{Nat}}(\ w, 3, (x, y).\mathsf{succ}(y)\ )$ $\mathsf{El}_{\mathsf{Nat}}(\ 2, 3, (x, y).\mathsf{succ}(y)\ )$

(b) Si associ ad ogni termine definito negli esercizi precedenti una strategia di riduzione tramite un numero finito (anche zero) di applicazioni di $\to_1$ fino ad ottenere un termine non ulteriormente riducibile.

3.2.1 Classificazione dei termini in teoria dei tipi

In questa sezione enunciamo alcune proprietà computazionali molto importanti relativi alla teoria con il tipo dei numeri naturali. Tali proprietà sono estendibili anche ad altri tipi e precisamente a tutti quelli all'interno *teoria dei tipi intensionale di Martin-Löf* in
Nordstrom, B., Petersson, K., Smith, J.: Programming in Martin-Löf's Type Theory. Clarendon Press, Oxford (1990).
Una prima distinzione tra termini tipati derivabili in teoria dei tipi è quella tra *termini canonici* e quelli *non canonici*:

Def. 3.5 Un termine tipato $a \in A\ [\Gamma]$ si dice **canonico** se si presenta in forma introduttiva ovvero il costruttore più esterno è quello di una regola di introduzione.
I termini **non canonici** sono quelli che non sono della forma descritta sopra.

Esempi: $\mathsf{succ}(x)$ è in forma canonica mentre non lo è $\mathsf{El}_{\mathsf{Nat}}(x', y', (x, y)\mathsf{succ}(x))$ oppure una qualsiasi variabile z.

Esiste anche un'altra classificazione tra termini che è completamente diversa da quella tra termini canonici e non canonici ed è la distinzione tra *termini in forma normale* e *termini non in forma normale* e di questa *diamo solo un'idea* rimandando il lettore alla letteratura per una definizione formale.
La definizione di termine in forma normale si basa usualmente sulla definizione di *riduzione ad un passo tra termini*

$$t \to_1 t'$$

tramite assiomi ottenuti trasformando tutte le regole di conversione in assiomi di riduzione della forma

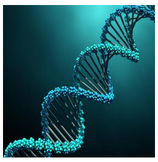
$$\mathsf{El}_{-}(\mathit{canonico}, ...) \to_1 t$$

e in modo analogo anche tutte le regole di uguaglianza che dicono che i vari costruttori di termini conservano l'uguaglianza.
Diamo ora alcuni esempi di assiomi di riduzione ad un passo di computazione $\to_1$ relativi ai termini introdotti dalle regole del tipo dei numeri naturali:

$$\begin{array}{llll} \beta_{1\mathsf{Nat}}\text{-red}) & \mathsf{El}_{\mathsf{Nat}}(0, c, e) & \to_1 & d \\ \beta_{2\mathsf{Nat}}\text{-red}) & \mathsf{El}_{\mathsf{Nat}}(\mathsf{succ}(m), c, e) & \to_1 & e(\ m\ ,\ \mathsf{El}_{\mathsf{Nat}}(m, c, e)\ ) \end{array}$$

$$\text{I-red})\ \frac{t_1 \ \to_1 \ t_2}{\mathsf{succ}(t_1) \ \to_1 \ \mathsf{succ}(t_2)}$$

$$\text{red}_I)\ \frac{t_1 \ \to_1 \ t_2}{\mathsf{El}_{\mathsf{Nat}}(t_1, c, e) \ \to_1 \ \mathsf{El}_{\mathsf{Nat}}(t_2, c, e)} \qquad \text{red}_{II})\ \frac{c_1 \ \to_1 \ c_2}{\mathsf{El}_{\mathsf{Nat}}(t, c_1, e) \ \to_1 \ \mathsf{El}_{\mathsf{Nat}}(t, c_2, e)}$$



Usando la nozione di riduzione si dice che:
Def. un termine t è *in forma normale* se non esiste termine a cui lui si riduce tramite una riduzione ad un passo ovvero per nessun termine s vale $t \to_1 s$, mentre invece un termine *non è in forma normale* se esiste un termine a cui lui si riduce tramite la riduzione ad un passo ovvero vale $t \to_1 s$ per un qualche termine s.
La distinzione tra termini in forma normale e non in forma normale è diversa da quella tra termini canonici e non canonici perchè esistono termini canonici che sono in forma normale e altri che non sono

in forma normale come pure esistono termini non canonici che sono in forma normale e altri che non sono in forma normale. Diamo ora alcuni esempi per chiarire questo.
Esempi di termini in forma normale: le variabili x sono in forma normale e sono pure termini non canonici, poi i numerali, ovvero i termini chiusi di tipo Nat della forma $\mathbf{0}$ oppure

$$\mathbf{n} \equiv \underbrace{\mathsf{succ}(\mathsf{succ}\ldots(0))}_{\text{n-volte}}$$

sono in forma normale e sono pure canonici; infine vi sono anche termini non canonici in forma normale come

$$\mathsf{El}_{\mathsf{Nat}}(x', y', (x,y).\mathsf{succ}(y))$$

Esempi di termini NON in forma normale: il termine $\mathsf{El}_{\mathsf{Nat}}(0, y', (x,y).\mathsf{succ}(y))$ è un esempio di termine in forma non normale in quanto si riduce a y' e difatti si può dedurre con le regole di riduzione sopra che

$$\mathsf{El}_{\mathsf{Nat}}(0, y', (x,y).\mathsf{succ}(y)) \ \rightarrow_1 \ y'$$

Pure il termine canonico

$$\mathsf{succ}(\ \mathsf{El}_{\mathsf{Nat}}(0, y', (x,y).\mathsf{succ}(y))\ )$$

NON è in forma normale in quanto si vede che

$$\mathsf{succ}(\ \mathsf{El}_{\mathsf{Nat}}(0, y', (x,y).\mathsf{succ}(y))\ ) \ \rightarrow_1 \ \mathsf{succ}(y')$$

Una volta definita una nozione di riduzione per i termini di una teoria dei tipi ci si può chiedere se vale il cosidetto *teorema di forma normale* il cui enunciato è il seguente:
Teorema di forma normale per una teoria dei tipi $\mathcal{T}$: Dato un termine t di una teoria dei tipi $\mathcal{T}$, tale che si derivi $t \in A\ [\Gamma]$ per qualche tipo dipendente A su Γ, allora, se t non è già in forma normale, esiste un termine s in forma normale a cui t si riduce tramite un numero finito di riduzioni. Inoltre in $\mathcal{T}$ si deriva il giudizio $s \in A\ [\Gamma]$ ed anche il giudizio

$$t = s \in A\ [\Gamma]$$

Teorema di confluenza per una teoria dei tipi $\mathcal{T}$: Dato un termine t di una teoria dei tipi $\mathcal{T}$, tale che si derivi $t \in A\ [\Gamma]$ per qualche tipo dipendente A su Γ, se esistono due termini s_1 e s_2 entrambi in forma normale a cui t si riduce tramite un numero finito di riduzioni, allora il termine s_1 è identico ad s_2 oppure uno diventa identico all'altro dopo una sostituzione di una o più variabili vincolate.

In genere il teorema di confluenza si ottiene come conseguenza del seguente teorema di Church-Rosser:
Teorema di Church-Rosser per la riduzione associata ai termini di una teoria dei tipi $\mathcal{T}$: Dati dei termini t_1, t_2, t_3 di una teoria dei tipi $\mathcal{T}$, tale che si derivi $t_i \in A\ [\Gamma]$ per $i = 1, \ldots, n$ per qualche tipo dipendente A su Γ, allora se t_1 si riduce a t_2 e pure a t_3 ovvero valgono $t_1 \rightarrow_1 t_2$ e $t_1 \rightarrow_1 t_3$ allora esiste un termine t_4 a cui sia t_2 che t_3 si riducono con un numero finito di riduzioni n indicato con $\rightarrow_n$ e si deriva pure $t_4 \in A\ [\Gamma]$.

$$\begin{tikzcd}
& t_1 \arrow[dl, "1"'] \arrow[dr, "1"] & \\
t_2 \arrow[dr, "n"'] & & t_3 \arrow[dl, "m"] \\
& t_4 &
\end{tikzcd}$$

Se in una teoria $\mathcal{T}$ valgono i teoremi di forma normale e confluenza allora si può definire una funzione che associa ad un termine t la sua forma normale $\mathsf{nf}(t)$.
Infine esiste anche una versione forte del teorema di forma normale che è la seguente:
Teorema di forma normale in versione forte per una teoria dei tipi $\mathcal{T}$: Dato un termine t di una teoria dei tipi $\mathcal{T}$, tale che si derivi $t \in A\ [\Gamma]$ per qualche tipo dipendente A su Γ, allora l'albero delle possibili sequenze di riduzioni ha profondità finita e ogni sequenza di riduzioni massimale termina in un termine s in forma normale a cui t si riduce. Inoltre in $\mathcal{T}$ si deriva il giudizio $s \in A\ [\Gamma]$ ed anche il giudizio

$$t = s \in A\ [\Gamma]$$

Inoltre si deduce che
Teorema di computabilità dell'uguaglianza definizionale: Dati due termini tipabili $\mathbf{a} \in \mathbf{A}\ [\mathbf{\Gamma}]$ e $\mathbf{b} \in \mathbf{A}\ [\mathbf{\Gamma}]$ nella teoria $\mathcal{T}$ vale che

il giudizio $\mathbf{a} = \mathbf{b} \in \mathbf{A}\ [\mathbf{\Gamma}]$ *è derivabile in* $\mathcal{T}$
se e solo se
$\mathsf{nf}(\mathbf{a}) \ \equiv\ \mathsf{nf}(\mathbf{b})$
ovvero
la forma normale $\mathsf{nf}(\mathbf{a})$ *di* $\mathbf{a}$ *è uguale sintatticamente alla forma normale* $\mathsf{nf}(\mathbf{b})$ *di* $\mathbf{b}$
(a meno di rinomina delle variabili vincolate)

ovvero

il giudizio $\mathbf{a} = \mathbf{b} \in \ [\mathbf{\Gamma}]$ *è derivabile in* $\mathcal{T}$
se e solo se
vale uno dei seguenti casi:

1. sia $\mathbf{a}$ che $\mathbf{b}$ sono in *forma normale*
e dunque $\mathbf{a}$ è uguale sintatticamente a $\mathbf{b}$
(a meno di rinomina delle variabili vincolate);
2. $\mathbf{a}$ è in *forma normale* ma $\mathbf{b}$ non lo è
e dunque $\mathbf{b}$ si riduce con un numero finito di riduzioni ad $\mathbf{a}$;
3. $\mathbf{b}$ è in *forma normale* ma $\mathbf{a}$ non lo è
e dunque $\mathbf{a}$ si riduce con un numero finito di riduzioni a $\mathbf{b}$;
4. sia $\mathbf{a}$ che $\mathbf{b}$ non sono in forma normale
e dunque esiste un termine $\mathbf{c}$ in *forma normale* a cui entrambi si riducono con un numero finito di riduzioni $\rightarrow_1$.

Se per i termini della teoria dei tipi esiste un algoritmo per calcolare la forma normale di un termine (in quanto il teorema di forma normale è dimostrato in modo costruttivo!) allora chiaramente *l'uguaglianza definizionale della teoria dei tipi risulta decidibile.*

Questo è il caso per la *teoria dei tipi intensionale di Martin-Löf* in
Nordstrom, B., Petersson, K., Smith, J.: Programming in Martin-Löf's Type Theory. Clarendon Press, Oxford (1990)
dove valgono tutti i teoremi e le proprietà enunciate in questa sezione.
Tali teoremi invece NON valgono per la sua versione estensionale in presenza di almeno un universo in
Martin-Löf, Per: Intuitionistic Type Theory. Notes by G. Sambin of a series of lectures given in Padua, June 1980. Bibliopolis, Naples (1984)
perchè la nozione di riduzione associata a tal teoria in modo che le forme normali chiuse siano solo termini in forma canonica consente di trovare dei termini con una sequenza infinita di riduzioni non terminanti in una forma normale.

Però per tale versione estensionale della teoria dei tipi di Martin-Löf vale il seguente teorema di forma canonica (che vale anche per la versione intensionale come conseguenza del teorema di forma normale):

Teorema di forma canonica per una teoria dei tipi $\mathcal{T}$: Dato un termine chiuso t di una teoria dei tipi $\mathcal{T}$ (ovvero senza variabili libere), tale che si derivi $t \in A\ [\Gamma]$ per qualche tipo dipendente A su Γ, allora esiste un termine s in forma canonica tale per cui si deriva il giudizio $s \in A\ [\Gamma]$ ed anche il giudizio

$$t = s \in A\ [\Gamma]$$

Tipo delle liste di un tipo

Il tipo delle liste di elementi di un altro tipo è definito dalle seguenti regole:

List type

F-list) $\dfrac{A\ type\ [\Gamma]}{List(A)\ type\ [\Gamma]}$ $\quad$ I$_1$-list) $\dfrac{List(A)\ type\ [\Gamma]}{\mathsf{nil} \in List(A)\ [\Gamma]}$ $\quad$ I$_2$-list) $\dfrac{s \in List(A)\ [\Gamma] \qquad a \in A\ [\Gamma]}{\mathsf{cons}(s,a) \in List(A)\ [\Gamma]}$

$$\text{E-list)}\ \frac{\begin{array}{c} M(z)\ type\ [\Gamma, z \in List(A)] \qquad t \in List(A)\ [\Gamma] \qquad c \in M(\mathsf{nil})\ [\Gamma] \\ e(x,w,y) \in M(\mathsf{cons}(x,w))\ [\Gamma, x \in List(A), w \in A, y \in M(x)] \end{array}}{El_{List}(t,c,e) \in M(t)\ [\Gamma]}$$

$$\text{C}_1\text{-list)}\ \frac{\begin{array}{c} M(z)\ type\ [\Gamma, z \in List(A)] \qquad c \in M(\mathsf{nil})\ [\Gamma] \\ e(x,w,y) \in M(\mathsf{cons}(x,w))\ [\Gamma, x \in List(A), w \in A, y \in M(x)] \end{array}}{El_{List}(\mathsf{nil},c,e) = c \in M(\mathsf{nil})\ [\Gamma]}$$

$$\text{C}_2\text{-list)}\ \frac{\begin{array}{c} M(z)\ type\ [\Gamma, z \in List(A)] \qquad s \in List(A)\ [\Gamma] \qquad a \in A\ [\Gamma] \qquad c \in M(\mathsf{nil})\ [\Gamma] \\ e(x,w,y) \in M(\mathsf{cons}(x,w))\ [\Gamma, x \in List(A), w \in A, y \in M(x)] \end{array}}{El_{List}(\mathsf{cons}(s,a),c,e) = e(s,c,El_{List}(s,c,e)) \in M(\mathsf{cons}(s,a))\ [\Gamma]}$$

Descriviamo di seguito le regole che garantiscono che i costruttori di tipo conservano l'uguaglianza tra tipi:

eq-F-Lists

$$\text{eq-list)}\ \frac{A_1 = A_2\ type\ [\Gamma]}{List(A_1) = List(A_2)\ type\ [\Gamma]}$$

Descriviamo di seguito le regole che garantiscono che i costruttori di termini conservano l'uguaglianza computazionale tra termini:

$$\text{eq-I}_2\text{ list)}\ \frac{s_1 = s_2 \in List(A)\ [\Gamma] \qquad a_1 = a_2 \in A\ [\Gamma]}{\mathsf{cons}(s_1,a_1) = \mathsf{cons}(s_2,a_2) \in List(A)\ [\Gamma]}$$

$$\text{eq-E-list)}\ \frac{\begin{array}{c} M(z)\ type\ [\Gamma, z \in List(A)] \qquad t_1 = t_2 \in List(A)\ [\Gamma] \qquad c_1 = c_2 \in M(\mathsf{nil})\ [\Gamma] \\ e_1(x,w,y) = e_2(x,w,y) \in M(\mathsf{cons}(x,w))\ [\Gamma, x \in List(A), w \in A, y \in M(x)] \end{array}}{\mathsf{El}_{List}(t_1,c_1,e_1) = \mathsf{El}_{List}(t_2,c_2,e_2) \in M(t_1)\ [\Gamma]}$$

Esercizio teorico:

Proposition 3.6 *La teoria dei tipi che include tutte le regole finora descritte eccetto quelle di sostituzione e di indebolimento rende ammissibili le regole di indebolimento 2.2.1 e sostituzione della sezione 2.2.2.*

Dim. per esercizio.
Esempio: Dato un tipo A semplice, ovvero non dipendente, per esempio $A = \mathsf{N_1}$, se vogliamo definire un'operazione

$$\mathbf{append_1}(x', y') \in List(A)\ [x' \in List(A), y' \in A]$$

tale che l'elemento y' venga posto alla fine della lista x' allora basta definire $\mathbf{append_1}$ in tal modo

$$\mathbf{append_1}(x', y') = \mathsf{cons}(x', y') \in List(A)[x' \in List(A), y' \in A]$$

Ma se vogliamo definire un'operazione

$$\mathbf{append_2}(x', y') \in List(A)\ [x' \in List(A), y' \in A]$$

tale che l'elemento y' venga posto all'*inizio* della lista x' allora occorre usare la regola di eliminazione sulla lista x'.
Dapprima si scriva equazionalmente la definizione ricorsiva di $\mathbf{append_2}$

$$\begin{array}{rcl}\mathbf{append_2}(\mathsf{nil}, y') & = & \mathsf{cons}(\mathsf{nil}, y') \\ \mathbf{append_2}(\mathsf{cons}(s, c), y') & = & \mathsf{cons}(\mathbf{append_2}(s, y'), c)\end{array}$$

poi si noti che la prima equazione definisce il termine a e la seconda il termine b nella regola di eliminazione prendendo come $B(z) = List(A)$
e quindi possiamo definire

$$\mathbf{append_2}(x', y') = \mathsf{El}_{List}(\, x'\ ,\ \mathsf{cons}(\mathsf{nil}, y')\ ,\ (x, y, z).\mathsf{cons}(z, y)\ )\ \in List(A)[x' \in List(A), y' \in A]$$

Esercizi:

1. Dati i tipi singoletto e delle liste è possibile definire un tipo dei numeri naturali Nat?

2. Definire l'operazione *append* di accostamento di una lista ad un'altra di un tipo A *type* [ ] usando le regole del tipo delle liste

   $$append(x, y) \in List(A)\ [x \in List(A), y \in List(A)]$$

   in modo tale che valga $append(x, \mathsf{nil}) = x \in List(A)\ [x \in List(A)]$.

3. Specificare il tipo e definire le operazioni:

   (a) l'operazione *back* che di una lista non vuota ne estrae la lista meno l'elemento di testa;

   (b) l'operazione $front$ che data una lista non vuota ne estrae la lista meno l'elemento di inizio;

   (c) l'operazione *last* che data una lista non vuota ne estrae l'elemento in testa;

   (d) l'operazione $first$ che data una lista non vuota ne estrae l'elemento all'inizio.

4. Si definiscano *termini untyped* con i costruttori del tipo dei numeri naturali secondo la seguente grammatica:

   $$t\ \equiv\ v\ \mid\ \mathsf{nil}\ \mid\ \mathsf{cons}(t_1, t_2)\ \mid\ \mathsf{El}_{\mathsf{Nat}}(t_1, t_2, (x, w, y).t_3(x, w, y))$$

con $v \in \{\ x, y, w, z\ \} \bigcup \{x_i \quad | \quad i \in Nat\}$ ovvero consideriamo come variabili le ultime lettere dell'alfabeto inglese e poi tutte le variabili ottenute ponendo alla variabile x un indice con indice che varia nei numeri naturali.

Definiamo la *riduzione ad un passo di computazione* $\rightarrow_1$ sui termini untyped definiti sopra, convenendo di usare la seguente abbreviazione

$$\mathsf{El}_{\mathsf{Nat}}(t_1, t_2, t_3) \text{ al posto di } \mathsf{El}_{list}(t_1, t_2,\ (x, w, y).t_3(x, w, y)\ )$$

tramite i seguenti assiomi:

$$\beta_{1\mathsf{Nat}}\text{-red}) \qquad \mathsf{El}_{list}(\mathsf{nil}, c, e\ ) \quad \rightarrow_1 \quad c$$

$$\beta_{2\mathsf{Nat}}\text{-red}) \qquad \mathsf{El}_{list}(\mathsf{cons}(s, a), c, e\ ) \quad \rightarrow_1 \quad e(\ s, a, \mathsf{El}_{list}(s, c, e\ )\ )$$

e le seguenti regole:

$$\text{I-red}) \quad \frac{s_1 \ \rightarrow_1 \ s_2}{\mathsf{cons}(s_1, a) \ \rightarrow_1 \ \mathsf{cons}(s_2, a)} \qquad \text{I}_2\text{-red}) \quad \frac{a_1 \ \rightarrow_1 \ a_2}{\mathsf{cons}(s, a_1) \ \rightarrow_1 \ \mathsf{cons}(s, a_2)}$$

$$\text{red}_I) \quad \frac{t_1 \ \rightarrow_1 \ t_2}{\mathsf{El}_{list}(t_1, c, e) \ \rightarrow_1 \ \mathsf{El}_{list}(t_2, c, e)} \qquad \text{red}_{II}) \quad \frac{c_1 \ \rightarrow_1 \ c_2}{\mathsf{El}_{list}(t, c_1, e) \ \rightarrow_1 \ \mathsf{El}_{list}(t, c_2, e)}$$

(a) Si associ ai seguenti termini una strategia di riduzione tramite un numero finito (anche zero) di applicazioni di $\rightarrow_1$ fino ad ottenere un termine non ulteriormente riducibile:

$$\mathsf{succ}(\ \mathsf{El}_{\mathsf{list}}(\ \mathsf{nil}\ ,\ 3\ ,\ (x, y).\mathsf{succ}(y)\ )\ ) \qquad \mathsf{El}_{\mathsf{list}}(\ w\ ,\ \ ,\ (x, y).\mathsf{succ}(y)\ )$$

$$\mathsf{El}_{\mathsf{list}}(\ \mathsf{cons}(\ \mathsf{cons}(\mathsf{nil}, 0)\ ,\ 0\ )\ ,\ 3\ ,\ (x, y).\mathsf{succ}(y)\ )$$

(b) Si associ ad ogni termine definito negli esercizi precedenti una strategia di riduzione tramite un numero finito (anche zero) di applicazioni di $\rightarrow_1$ fino ad ottenere un termine non ulteriormente riducibile.

3.3 Tipo della somma disgiunta

Le regole del tipo della somma disgiunta:

Disjoint Sum type

F-+) $\dfrac{B\ type\ [\Gamma] \quad C\ type\ [\Gamma]}{B+C\ type\ [\Gamma]}$

I$_1$-+) $\dfrac{b \in B\ [\Gamma] \qquad B\ type\ [\Gamma] \qquad C\ type\ [\Gamma]}{\mathsf{inl}(b) \in B+C\ [\Gamma]}$ I$_2$-+) $\dfrac{c \in C\ [\Gamma] \qquad B\ type\ [\Gamma] \qquad C\ type\ [\Gamma]}{\mathsf{inr}(c) \in B+C\ [\Gamma]}$

E-+) $\dfrac{\begin{array}{l} M(z)\ type\ [\Gamma, z \in B+C] \\ t \in B+C\ [\Gamma] \qquad e_B(x_1) \in M(\mathsf{inl}(x_1))\ [\Gamma, x_1 \in B] \quad e_C(x_2) \in M(\mathsf{inr}(x_2))\ [\Gamma, x_2 \in C] \end{array}}{El_+(t, e_B, e_C) \in M(t)\ [\Gamma]}$

C$_1$-+) $\dfrac{\begin{array}{l} M(z)\ type\ [\Gamma, z \in B+C] \\ b \in B\ [\Gamma] \qquad e_B(x_1) \in M(\mathsf{inl}(x_1))\ [\Gamma, x_1 \in B] \quad e_C(x_2) \in M(\mathsf{inr}(x_2))\ [\Gamma, x_2 \in C] \end{array}}{El_+(\mathsf{inl}(b), e_B, e_C) = e_B(b) \in M(\mathsf{inl}(b))\ [\Gamma]}$

C$_2$-+) $\dfrac{\begin{array}{l} M(z)\ type\ [\Gamma, z \in B+C] \\ c \in C\ [\Gamma] \qquad e_B(x_1) \in M(\mathsf{inl}(x_1))\ [\Gamma, x_1 \in B] \quad e_C(x_2) \in M(\mathsf{inr}(x_2))\ [\Gamma, x_2 \in C] \end{array}}{El_+(\mathsf{inr}(c), e_B, e_C) = e_C(c) \in M(\mathsf{inr}(c))\ [\Gamma]}$

Descriviamo di seguito le regole che garantiscono che i costruttori di tipo conservano l'uguaglianza tra tipi:

eq-F-+ $\dfrac{B_1 = B_2\ type\ [\Gamma] \qquad C_1 = C_2\ type\ [\Gamma]}{B_1 + C_1 = B_2 + C_2\ type\ [\Gamma]}$

Diamo ora di seguito le regole che garantiscono che i costruttori di termini conservano l'uguaglianza computazionale tra termini:

eq-I$_1$-+) $\dfrac{b_1 = b_2 \in B\ [\Gamma] \qquad B+C\ type\ [\Gamma]}{\mathsf{inl}(b_1) = \mathsf{inl}(b_2) \in B+C\ [\Gamma]}$

eq-I$_2$-+) $\dfrac{c_1 = c_2 \in C\ [\Gamma] \qquad B+C\ type\ [\Gamma]}{\mathsf{inr}(c_1) = \mathsf{inr}(c_2) \in B+C\ [\Gamma]}$

eq-E-+) $\dfrac{\begin{array}{l} M(z)\ type\ [\Gamma, z \in B+C] \qquad t_1 = t_2 \in B+C\ [\Gamma] \\ (e_B)_1(x_1) = (e_B)_2(x_1) \in M(\mathsf{inl}(x_1))\ [\Gamma, x_1 \in B] \quad (e_C)_1(x_2) = (e_C)_2(x_2) \in M(\mathsf{inr}(x_2))\ [\Gamma, x_2 \in C] \end{array}}{\mathsf{El}_+(t_1, (e_B)_1, (e_C)_1) = \mathsf{El}_+(t_2, (e_B)_2, (e_C)_2) \in M(t_1)\ [\Gamma]}$

Remark 3.7 Si noti che nello scrivere il costruttore eliminatore della disgiunzione $\mathsf{El}_+(t_1, e_B, e_C)$ a partire da $e_B(x_1) \in M(\mathsf{inl}(x_1))\ [x_1 \in B]$ e $e_C(x_2) \in M(\mathsf{inr}(x_2))\ [x_2 \in C]$ abbiamo considerato questi

ultimi come semplici costanti nella notazione dell'eliminatore astraendo sulle variabili.
Secondo l'higher order syntax, ovvero la "sintassi di ordine superiore", $e_B(x) \in M(\mathsf{inl}(x_1))\ [x_1 \in B]$ dà luogo a $(x).e_1(x)$ che appartiene al tipo di ordine superiore $(x_1 \in B)\, M(\mathsf{inl}(x_1))$. Poi per η-conversione tra tipi higher order segue che $(x).e_B(x)$ è uguale a e_B.

Esercizi:

1. Si scrivano le regole del tipo booleano come tipo semplice (con regole di formazione, introduzione, eliminazioni e conversione) e si provi che è rappresentabile da $\mathsf{N_1} + \mathsf{N_1}$.

2. Si mostri che esistono termini
$$\mathbf{dec}(z) \in \mathsf{Nat}\ [z \in \mathsf{N_1} + \mathsf{Nat}] \qquad\qquad \mathbf{cod}(z) \in \mathsf{N_1} + \mathsf{Nat}\ [w \in \mathsf{Nat}]$$
tali che per ogni numerale $\mathbf{n}$
$$\begin{array}{ll} \mathbf{dec}(\mathbf{cod}(0)) = 0 & \qquad \mathbf{cod}(\mathbf{dec}(\mathsf{inl}(*))) = \mathsf{inl}(*) \\ \mathbf{dec}(\mathbf{cod}(\mathbf{n})) = \mathbf{n} & \qquad \mathbf{cod}(\mathbf{dec}(\mathsf{inr}(\mathbf{n}))) = \mathsf{inr}(\mathbf{n}) \end{array}$$

3. Dati i tipi $B\ type\ [\Gamma]$ e $C\ type\ [\Gamma]$ e $b \in B\ [\Gamma]$ allora si definisce un termine
$$\mathsf{pr}(\mathbf{z}) \in \mathbf{B}\ [\mathbf{z} \in \mathbf{B} + \mathbf{C}]$$
in modo tale che si derivi
$$\mathsf{pr}(\mathsf{inl}(\mathbf{x})) = \mathbf{x} \in \mathbf{B}\ [\mathbf{x} \in \mathbf{B}] \qquad \mathsf{pr}(\mathsf{inr}(\mathbf{y})) = \mathbf{b} \in \mathbf{B}\ [\mathbf{y} \in \mathbf{C}]$$

4. Si definiscano *termini untyped* con i costruttori del tipo della somma disgiunta secondo la seguente grammatica:
$$t \ \equiv\ v \ \mid\ \mathsf{inl}(t)\ \mid\ \mathsf{inr}(t)\ \mid\ \mathsf{El}_+(t_1,\ (x_1).e_1(x_1)\ ,\ (x_2).e_2(x_2)\ )$$
con $v \in \{\ x, y, w, z\ \} \bigcup \{x_i \ \mid\ i \in Nat\}$ ovvero consideriamo come variabili le ultime lettere dell'alfabeto inglese e poi tutte le variabili ottenute ponendo alla variabile x un indice con indice che varia nei numeri naturali.

   Definiamo la *riduzione ad un passo di computazione* $\to_1$ sui termini untyped definiti sopra, convenendo di usare la seguente abbreviazione
$$\mathsf{El}_+(t, e_1, e_2) \text{ al posto di } \mathsf{El}_+(t, (x_1).e_1(x_1),\ (x_2).e_2(x_2)\ )$$
tramite i seguenti assiomi:
$$\begin{array}{lccc} \beta_{1+}\text{-red}) & \mathsf{El}_+(\mathsf{inl}(b), e_1, e_2\ ) & \to_1 & e_1(b) \\ \beta_{2+}\text{-red}) & \mathsf{El}_+(\mathsf{inr}(c), e_1, e_2\ ) & \to_1 & e_2(c) \end{array}$$
e le seguenti regole:
$$\text{I}_1\text{-red)}\ \ \frac{b_1\ \to_1\ b_2}{\mathsf{inl}(b_1)\ \to_1\ \mathsf{inl}(b_2)} \qquad\qquad \text{I}_2\text{-red)}\ \ \frac{c_1\ \to_1\ c_2}{\mathsf{inr}(c_1)\ \to_1\ \mathsf{inr}(c_2)}$$
$$\text{red}_I)\ \ \frac{t_1\ \to_1\ t_2}{\mathsf{El}_+(t_1, e_1, e_2)\ \to_1\ \mathsf{El}_+(t_2, e_1, e_2)}$$

(a) Si associ ai seguenti termini relativi ai tipi singoletto, dei naturali e delle somme disgiunte, una strategia di riduzione tramite un numero finito (anche zero) di applicazioni di $\rightarrow_1$ (definita sopra e nelle sezioni esercizi dei tipi singoletto, dei naturali e delle liste) fino ad ottenere un termine non ulteriormente riducibile:

$\mathsf{inl}(\ \mathsf{El}_+(\mathsf{inr}(x), (x_1).\mathsf{inr}(x_1)\ , (x_2).\mathsf{inl}(x_2)\ )\ )$ $\qquad$ $\mathsf{El}_+(\mathsf{inl}(x), (x_1).\mathsf{inl}(x_1)\ , (x_2).\mathsf{inr}(x_2)\ )$

$\mathsf{El}_+(\mathsf{inl}(x), (x_1).\mathsf{inr}(\mathsf{succ}(x_1))\ , (x_2).\mathsf{inr}(x_2)\ )$

$\mathsf{El}_+(\ z\ , (x_1).\mathsf{inl}(\ \mathsf{El}_{N_1}(*, *)\ )\ ,\ (x_2).\mathsf{inr}(\ \mathsf{El}_{list}(\ \mathsf{nil},\ \mathsf{cons}(\mathsf{nil}, *)\ , (x, w, y).x\ )\ )\ )$

e si provi a vedere se sono tipabili nella teoria dei tipi con tipi somme disgiunte, naturali e singoletto.

(b) Si associ ad ogni termine definito negli esercizi precedenti una strategia di riduzione tramite un numero finito (anche zero) di applicazioni di $\rightarrow_1$ fino ad ottenere un termine non ulteriormente riducibile supponendo di includere nella riduzione $\rightarrow_1$ tutte le regole di riduzione associate ai termini dei tipi somme disgiunte, naturali e singoletto.

3.4 Tipo dell'uguaglianza proposizionale di Martin-Löf

l tipo dell'uguaglianza proposizionale intensionale è stato introdotto in teoria dei tipi da Per Martin-Löf per interpretare l'uguaglianza della logica predicativa intuizionista. Come dall'altra parte il predicato uguaglianza è un *predicato atomico primitivo* in logica predicativa (anzi l'unico), allo stesso modo il tipo uguaglianza proposizionale è un *tipo dipendente primitivo* e il solo che introdurremo che è genuinamente *dipendente*. Altri tipi dipendenti sono infatti costruiti a partire da questo attraverso altri costruttori di tipo, come il tipo delle liste o le somme disgiunte, che se utilizzati su tipi non dipendenti producono sempre tipi non dipendenti.
In verità Martin-Löf ha introdotto due versioni di tipo "uguaglianza proposizionale": una che chiamiamo qui "Intensionall Propositional Equality" e l'altra "Extensional Propositional Equality" che sarà menzionata in seguito. In letteratura il tipo "Intensionall Propositional Equality" è anche chiamato "Identity Propositional Type".
Le regole del **tipo uguaglianza proposizionale di Martin-Löf** sono le seguenti:

Martin-Löf's Intensional Propositional Equality

$$\text{F-Id)}\ \frac{A\ type\ [\Gamma] \quad a \in A\ [\Gamma] \quad b \in A\ [\Gamma]}{\mathsf{Id}(A,a,b)\ type\ [\Gamma]} \qquad \text{I-Id)}\ \frac{a \in A\ [\Gamma]}{\mathsf{id}(a) \in \mathsf{Id}(A,a,a)\ [\Gamma]}$$

$$\text{E-Id)}\ \frac{\begin{array}{c} M(z_1,z_2,z_3)\ type\ [\Gamma, z_1 \in A, z_2 \in A, z_3 \in \mathsf{Id}(A,z_1,z_2)] \\ a \in A\ [\Gamma] \quad b \in A\ [\Gamma] \quad t \in \mathsf{Id}(A,a,b)\ [\Gamma] \quad e(x) \in M(x,x,\mathsf{id}(x))\ [\Gamma, x \in A] \end{array}}{El_{\mathsf{Id}}(t,\ (x).e(x)\ ) \in M(a,b,t)\ [\Gamma]}$$

$$\text{C-Id)}\ \frac{\begin{array}{c} M(z_1,z_2,z_3)\ type\ [\Gamma, z_1 \in A, z_2 \in A, z_3 \in \mathsf{Id}(A,z_1,z_2)] \\ a \in A\ [\Gamma] \qquad e(x) \in M(x,x,\mathsf{id}(x))\ [\Gamma, x \in A] \end{array}}{El_{\mathsf{Id}}(\mathsf{id}(a),e) = e(a) \in M(a,a,\mathsf{id}(a))\ [\Gamma]}$$

Descriviamo di seguito le regole che garantiscono che i costruttori di tipo conservano l'uguaglianza tra tipi:

eq-F-Id

$$\text{eq-Id}\ \frac{A_1 = A_2\ type\ [\Gamma] \quad a_1 = a_2 \in A_1\ [\Gamma] \quad b_1 = b_2 \in A_1\ [\Gamma]}{\mathsf{Id}(A_1,a_1,b_1) = \mathsf{Id}(A_2,a_2,b_2)\ type\ [\Gamma]}$$

Diamo ora di seguito le regole che garantiscono che i costruttori di termini conservano l'uguaglianza computazionale tra termini:

$$\text{eq-I-Id)}\ \frac{a = b \in A\ [\Gamma]}{\mathsf{id}(a) = \mathsf{id}(b) \in \mathsf{Id}(A,a,b)\ [\Gamma]}$$

$$\text{eq-E-Id)}\ \frac{\begin{array}{c} M(z_1,z_2,z_3)\ type\ [\Gamma, z_1 \in A, z_2 \in A, z_3 \in \mathsf{Id}(A,z_1,z_2)] \\ a \in A\ [\Gamma] \quad b \in A\ [\Gamma] \quad t_1 = t_2 \in \mathsf{Id}(A,a,b)\ [\Gamma] \quad e_1(x) = e_2(x) \in M(x,x,\mathsf{id}(x))\ [\Gamma, x \in A] \end{array}}{El_{\mathsf{Id}}(t_1,\ (x).e_1(x)\ ) = El_{\mathsf{Id}}(t_2,\ (x).e_2(x)\ ) \in M(a,b,t_1)\ [\Gamma]}$$

Queste regole esprimono che il tipo $\mathsf{Id}(A, z_1, z_2)$ *type* $[\Gamma, z_1 \in A, z_2 \in A]$ e' il *tipo induttivamente generato dalla relazione riflessiva degli elementi di un tipo dato A type* $[\Gamma]$.

Esercizi. Si consideri la teoria dei tipi con i tipi finora descritti e in tal teoria si dimostrino i seguenti fatti:

1.

2. Sia T_{id} la teoria dei tipi definita dalle regole del tipo uguaglianza proposizionale e le regole strutturali descritte nelle sezioni precedenti incluse quelle di sostituzione e indebolimento.

   (a) Dimostrare che in T_{id} è derivabile la regola

   $$\text{E-}\mathsf{Id}_{prog})\ \frac{\begin{array}{c} M(z_1, z_2, z_3)\ type\ [\Sigma, z_1 \in A, z_2 \in A, z_3 \in \mathsf{Id}(A, z_1, z_2)] \\ d \in M(x, x, \mathsf{id}(x))\ [\Sigma, x \in A] \end{array}}{\mathsf{El_{Id}}(z_3, d) \in M(z_1, z_2, z_3)\ type\ [\Sigma, z_1 \in A, z_2 \in A, z_3 \in \mathsf{Id}(A, z_1, z_2)]}$$

   ovvero dimostrare che se i giudizi premessa sono derivabili in T_{id}, allora lo è anche il giudizio conclusione.

   (b) Dimostrare che la regola $\mathsf{E\text{-}Id}$) è derivabile nella teoria dei tipi ottenuta da T_{id} rimpiazzando la regola di eliminazione $\mathsf{E\text{-}Id}$) con la regola E-Id_{prog}) e aggiungendo le regole di sostituzione e indebolimento e quelle di sanitary checks esposte nelle precedenti sezioni.

3. Si dimostri che esiste un proof-term $\mathbf{h_1}$ della forma

   $$h_1(z_1, z_2, z_3) \in \mathsf{Id}(A, z_2, z_1)\ [\Gamma, z_1 \in A, z_2 \in A, z_3 \in \mathsf{Id}(A, z_1, z_2)]$$

4. Si dimostri che esiste un proof-term $\mathbf{h_2}$ per

   $$h_2(z_1, z_2, z_3) \in \mathsf{Id}(\mathsf{Nat}, \mathsf{succ}(z_1), \mathsf{succ}(z_2))\ [z_1 \in \mathsf{Nat}, z_2 \in \mathsf{Nat}, z_3 \in \mathsf{Id}(\mathsf{Nat}, z_1, z_2)]$$

5. Si dimostri che se $a = b \in A\ [\Gamma]$ è derivabile nella teoria dei tipi con le regole finora introdotte esiste un proof-term **pf** tale che

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

   è derivabile.

6. Si dimostri che se $\mathbf{a} = \mathbf{b} \in \mathbf{A}\ [\mathbf{\Gamma}]$ è derivabile allora $\mathsf{id}(\mathbf{b}) \in \mathsf{Id}(\mathbf{A}, \mathbf{a}, \mathbf{b})$ è derivabile.

7. Si dia un esempio di termini $a \in A\ [\Gamma]$ e $b \in A\ [\Gamma]$ per cui NON si deriva $\mathbf{a} = \mathbf{b} \in \mathbf{A}\ [\mathbf{\Gamma}]$ ma invece esiste un proof-term **pf** tale che

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

   è derivabile.

8. Dimostrare che esiste un proof-term **pf** tale che

   $$\mathbf{pf} \in \mathsf{Id}(\,\mathsf{N_1}\,, *, w\,)\ [w \in \mathsf{N_1}]$$

   è derivabile.

9. Dimostrare che esiste un proof-term **pf** tale che

   $$\mathbf{pf} \in \mathsf{Id}(\mathsf{N_1}, x, w\,)\ [x \in \mathsf{N_1}, w \in \mathsf{N_1}]$$

   è derivabile.

10. Dimostrare che esiste un proof-term che è elemento di $\mathsf{Id}(\mathbf{Nat}, \mathsf{succ}(\ \mathsf{El}_{\mathsf{Nat}}(\mathbf{0}, \mathbf{0},\ (\mathbf{x}, \mathbf{y}).\mathsf{succ}(\mathbf{y})\ )\ ), \mathsf{succ}(\mathbf{0})\ )$.

$$q \in \mathsf{Id}(\ B, f(x), f(y))\ [x \in A, y \in A, w \in \mathsf{Id}(\ A, x, y)]$$

?+

11. Si può dimostrare che esiste un proof-term **pf** del tipo

$$\mathbf{pf} \in P(y)\ [x \in A, y \in A, z \in P(x), w \in \mathsf{Id}(A, x, y)]$$

per ogni $P(y)$ *type* $[x \in A]$ derivabile?

12. Si può dimostrare che esiste un proof-term **pf** del tipo

$$\mathbf{pf} \in \mathsf{I}d(A, x, z)\ [x \in A, y \in A, z \in A, w_1 \in \mathsf{Id}(A, x, y), w_2 \in \mathsf{Id}(A, y, z)]$$

per ogni tipo A *type* [ ] derivabile?

13. Dimostrare che esiste un proof-term tale che è possibile definire l'addizione tra numeri naturali

$$x + y \in \mathsf{Nat}\ [x \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

in modo tale che esistano dei proof-term $\mathbf{pf_1}$ e $\mathbf{pf_2}$ tali che

$$\mathbf{pf_1} \in \mathsf{Id}(\mathsf{Nat}, x + 0, x)\ [x \in \mathsf{Nat}] \qquad \mathbf{pf_2} \in \mathsf{Id}(\mathsf{Nat}, 0 + x, x)\ [x \in \mathsf{Nat}]$$

14. Si dimostri che esiste un proof-term **pf** tale che

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

ove

$$x' +_1 y' \ \equiv\ \mathsf{El}_{\mathsf{Nat}}(y', x', (x, y).\mathsf{succ}(y)\ ) \qquad x' +_2 y' \ \equiv\ \mathsf{El}_{\mathsf{Nat}}(x', y', (x, y).\mathsf{succ}(y)\ )$$

15. Per esercizio si provi a definire il tipo dell'insieme "vuoto" N_0.

16. Nella teoria dei tipi dipendenti con tipo singoletto, tipo uguaglianza proposizionale alla Leibniz, tipo delle liste, tipo booleano e tipo somma disgiunta, quale tipo è definibile a partire dagli altri?

17. Si definiscano *termini untyped* con i costruttori del tipo uguaglianza proposizionale secondo la seguente grammatica:

$$t \ \equiv\ \ v \ \mid\ \mathsf{id}(t)\ \mid \mathsf{El}_{\mathsf{Id}}(t,\ (x).e(x)\ )$$

con $v \in \{\ x, y, w, z\ \} \bigcup \{x_i \ \mid\ i \in Nat\}$ ovvero consideriamo come variabili le ultime lettere dell'alfabeto inglese e poi tutte le variabili ottenute ponendo alla variabile x un indice con indice che varia nei numeri naturali.

Definiamo la *riduzione ad un passo di computazione* $\to_1$ sui termini untyped definiti sopra, convenendo di usare la seguente abbreviazione

$$\mathsf{El}_{\mathsf{Id}}(t,\ e\,) \text{ al posto di } \mathsf{El}_{+}(t,\ (x).e(x)\ )$$

tramite i seguenti assiomi:

$$\beta_{\mathsf{Id}}\text{-red)} \qquad \mathsf{El}_{+}(\mathsf{id}(a),\ e\ ) \quad \to_1 \quad e(a)$$

e le seguenti regole:

$$\text{I-red)} \quad \frac{a_1 \ \to_1\ a_2}{\mathsf{id}(\mathsf{a}_1)\ \to_1\ \mathsf{id}(\mathsf{a}_2)} \qquad\qquad \text{red}_I) \quad \frac{t_1 \ \to_1\ t_2}{\mathsf{El}_{\mathsf{Id}}(t_1,\ e)\ \to_1\ \mathsf{El}_{\mathsf{Id}}(t_2,\ e)}$$

(a) Si associ ai seguenti termini relativi ai tipi singoletto, dei naturali, delle liste, delle somme disgiunte e dell'uguaglianza proposizionale una strategia di riduzione tramite un numero finito (anche zero) di applicazioni di $\rightarrow_1$ (definita sopra e nelle sezioni esercizi dei tipi singoletto, dei naturali, delle liste, delle somme disgiunte) fino ad ottenere un termine non ulteriormente riducibile:

$$\mathsf{id}(\ \mathsf{El}_+(\mathsf{inr}(x), (x_1).\mathsf{inr}(x_1)\ , (x_2).\mathsf{inl}(x_2)\ )\ ) \qquad \mathsf{El}_{\mathsf{Id}}(\mathsf{id}(z), (x).\mathsf{id}(x)\ ) \qquad \mathsf{El}_{\mathsf{N}_1}(z, \mathsf{id}(*)\ )$$

e si provi a vedere se sono tipabili nella teoria dei tipi con i tipi finora descritti.

(b) Si associ ad ogni termine definito negli esercizi precedenti una strategia di riduzione tramite un numero finito (anche zero) di applicazioni di $\rightarrow_1$ fino ad ottenere un termine non ulteriormente riducibile supponendo di includere nella riduzione $\rightarrow_1$ tutte le regole di riduzione associate ai termini dei tipi somme disgiunte, naturali e singoletto.

3.5 Altre formulazione di tipo uguaglianza: alla Gentzen/Lawvere, alla Leibniz e Propositional Equality with Path Induction

Vi sono altre formulazioni del tipo dell'uguaglianza da usare poi per interpretare l'uguaglianza predicativa.
Una di queste è il tipo dell'uguaglianza proposizionale alla Gentzen/Lawvere la cui regola di eliminazione ricorda la regola a sinistra dell'uguaglianza nel calcolo dei sequenti per la logica predicativa:

Gentzen/Lawvere Propositional Equality

$$\text{F-Id}_g)\ \frac{A\ type\ [\Gamma] \quad a \in A\ [\Gamma] \quad b \in A\ [\Gamma]}{\mathsf{Id}_g(A,a,b)\ type\ [\Gamma]} \qquad \text{I-Id}_g)\ \frac{a \in A\ [\Gamma]}{\mathsf{id}(a) \in \mathsf{Id}_g(A,a,a)\ [\Gamma]}$$

$$\text{E-Id}_g)\ \frac{\begin{array}{l} C(x,y)\ type\ [\Gamma, x \in A, y \in A] \\ a \in A\ [\Gamma] \quad b \in A\ [\Gamma] \quad p \in \mathsf{Id}_g(A,a,b)\ [\Gamma] \quad c \in C(a,a)\ [\Gamma] \end{array}}{El_{\mathsf{Id}_g}(p,c) \in C(a,b)\ [\Gamma]}$$

$$\text{C-Id}_g)\ \frac{\begin{array}{l} C(x,y)\ type\ [\Gamma, x \in A, y \in A] \\ a \in A\ [\Gamma] \quad c \in C(a,a)\ [\Gamma] \end{array}}{El_{\mathsf{Id}_g}(\mathsf{id}_A(a),c) = c \in C(a,a)\ [\Gamma]}$$

Un altro tipo uguaglianza proposizionale è quello di Leibniz la cui regola di eliminazione è una forma ristretta della regola di eliminazione del tipo uguaglianza proposizionale di Martin-Löf che verrà proposto in seguito:

Leibniz Propositional Equality

$$\text{F-Id}_l)\ \frac{A\ type\ [\Gamma] \quad a \in A\ [\Gamma] \quad b \in A\ [\Gamma]}{\mathsf{Id}_l(A,a,b)\ type\ [\Gamma]} \qquad \text{I-Id}_l)\ \frac{a \in A\ [\Gamma]}{\mathsf{id}(a) \in \mathsf{Id}_l(A,a,a)\ [\Gamma]}$$

$$\text{E-Id}_l)\ \frac{\begin{array}{l} C(x,y)\ type\ [\Gamma, x \in A, y \in A] \\ a \in A\ [\Gamma] \quad b \in A\ [\Gamma] \quad p \in \mathsf{Id}_l(A,a,b)\ [\Gamma] \quad c(x) \in C(x,x)\ [\Gamma, x \in A] \end{array}}{El_{\mathsf{Id}_l}(p,c) \in C(a,b)\ [\Gamma]}$$

$$\text{C-Id}_l)\ \frac{\begin{array}{l} C(x,y)\ type\ [\Gamma, x \in A, y \in A] \\ a \in A\ [\Gamma] \quad c(x) \in C(x,x)\ [\Gamma, x \in A] \end{array}}{El_{\mathsf{Id}_l}(\mathsf{id}(a),c) = c(a) \in C(a,a)\ [\Gamma]}$$

Esiste un'altra descrizione del tipo uguaglianza proposizionale, detto **Propositional Equality with Path Induction**, le cui regole includono una regola di eliminazione che risulta proprio la regola di eliminazione associata alla regola di introduzione:

Propositional Equality with Path Induction

$$\text{F-Id)}\ \frac{A\ type\ [\Gamma] \quad a \in A\ [\Gamma] \quad b \in A\ [\Gamma]}{\mathsf{Id}_p(A, a, b)\ type\ [\Gamma]} \qquad \text{I-Id)}\ \frac{a \in A\ [\Gamma]}{\mathsf{id}(a) \in \mathsf{Id}_p(A, a, a)\ [\Gamma]}$$

$$\text{E-Id)}\ \frac{\begin{array}{ll} a \in A\ [\Gamma] & C(y, z)\ type\ [\Gamma, y \in A, z \in \mathsf{Id}(A, a, y)] \\ b \in A\ [\Gamma] & p \in \mathsf{Id}_p(A, a, b)\ [\Gamma] \qquad c \in C(a, \mathsf{id}(a))\ [\Gamma] \end{array}}{El_{\mathsf{Id}_p}(p, c) \in C(b, p)\ [\Gamma]}$$

$$\text{C-Id)}\ \frac{\begin{array}{l} C(y, z)\ type\ [\Gamma, y \in A, z \in \mathsf{Id}_p(A, a, y)] \\ a \in A\ [\Gamma] \quad c \in C(a, \mathsf{id}(a))\ [\Gamma] \end{array}}{El_{\mathsf{Id}_p}(\mathsf{id}(a), c) = c \in C(a, \mathsf{id}(a))\ [\Gamma]}$$

Nota bene: Anticipiamo qui che negli esercizi che seguono scriveremo $\mathsf{Id}_?(A, a, b)$ per indicare un tipo uguaglianza generico e chiameremo **proof-term** gli elementi di un qualsiasi tipo uguaglianza proposizionale ovvero quei termini **pf** tali che

$$\mathbf{pf} \in \mathsf{Id}_?(A, a, b)\ [\Gamma]$$

è derivabile in una teoria dei tipi dipendente.

Esercizi:

1. In che relazione stanno i tipi di uguaglianza descritti finora?
2. Con quale tipo di uguaglianza si può dimostrare che dato un termine $f(x) \in B\ [x \in A]$ si può trovare un proof-term **q** tale che

   $$q \in \mathsf{Id}_?(B, f(x), f(y))\ [x \in A, y \in A, w \in \mathsf{Id}(A, x, y)]$$

   ?
3. Con quale tipo di uguaglianza si può dimostrare che esiste un proof-term **pf** del tipo

   $$\mathbf{pf} \in P(y)\ [x \in A, y \in A, z \in P(x), w \in \mathsf{Id}_?(A, x, y)]$$

4. Con quale tipo di uguaglianza si può dimostrare che esiste un proof-term **pf** del tipo

$$\mathbf{pf} \in \mathsf{Id}_?(A, x, z)\ [x \in A, y \in A, z \in A, w_1 \in \mathsf{Id}_?(A, x, y), w_2 \in \mathsf{Id}_?(A, y, z)]$$

5. Con quale tipo di uguaglianza si può dimostrare che esiste un proof-term **pf** del tipo

$$\mathbf{pf} \in \mathsf{Id}_?(\mathsf{N_1}, x, *)\ [x \in \mathsf{N_1}]$$

6. Con quale tipo di uguaglianza si può dimostrare che esiste un proof-term tale che è possibile definire l'addizione tra numeri naturali

$$x + y \in \mathsf{Nat}\ [x \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

in modo tale che esistano dei proof-term $\mathbf{pf_1}$ e $\mathbf{pf_2}$ tali che

$$\mathbf{pf_1} \in \mathsf{Id}_?(\mathsf{Nat}, x + 0, x)\ [x \in \mathsf{Nat}] \qquad \mathbf{pf_2} \in \mathsf{Id}_?(\mathsf{Nat}, 0 + x, x)\ [x \in \mathsf{Nat}]$$

7. Quali assiomi dell'aritmetica di Peano si riescono a derivare con i tipi finora descritti?

8. La regola di eliminazione del tipo uguaglianza proposizionale alla Leibniz rappresenta davvero una definizione per induzione sul tipo dei suoi elementi?

9. Per capire l'importanza della regola di eliminazione del tipo di uguaglianza proposizionale formulato da Martin-Löf si dimostri che si aggiunge alla teoria dei tipi con il tipo di uguaglianza proposizionale formulato da Martin-Löf le seguenti regole di *unicità dei proof-term di uguaglianza*

$$\text{I-Iduni)}\ \frac{a \in A\ [\Gamma] \qquad p \in \mathsf{Id}(A, a, a)\ [\Gamma]}{\mathit{Iduni}(a, p) \in \mathsf{Id}(\ \mathsf{Id}(A, a, a)\ , p, \mathsf{id}(a)\ )\ [\Gamma]}$$

$$\text{C-Iduni)}\ \frac{a \in A\ [\Gamma]}{\mathit{Iduni}(a, \mathsf{id}(a)) = \mathsf{id}(\mathsf{id}(a)) \in \mathsf{Id}(\ \mathsf{Id}(A, a, a)\ , \mathsf{id}(a), \mathsf{id}(a)\ )\ [\Gamma]}$$

allora si dimostra che dati $a \in A\ [\Gamma]$ e $b \in A\ [\Gamma]$ c'e' un proof-term **pf** di

$$pf \in \forall_{w_1 \in \mathsf{Id}(A,a,b)}\ \forall_{w_2 \in \mathsf{Id}(A,a,b)}\ \ \mathsf{Id}(\ \mathsf{Id}(A, a, b)\ , w_1, w_2\ )\ [\Gamma]$$

ovvero i proof-term di un tipo di uguaglianza proposizionale sono tutti uguali proposizionalmente.

(Suggerimento: si usi l'eliminazione dell'uguaglianza proposizionale di Martin-Löf sul tipo

$$C(x, y, z) \equiv \forall_{w \in \mathsf{Id}(A,x,y)}\ \mathsf{Id}(\ \mathsf{Id}(A, x, y)\ , z, w)\ [\Gamma, x \in A, y \in A,\ z \in \mathsf{Id}(A, x, y)]$$

)

10. si risolvano gli esercizi di sezione 3.4 rispetto ai nuovi tipi di uguaglianza.

11. Sia k un numero naturale. Dare le regole del tipo N_k con k elementi. Poi dimostrare che nella teoria dei tipi estesa con i tipi N_k, ognuno di questi tipi risulta isomorfo ad un tipo definibile a partire dai costrutti di tipo precedentemente descritti.

12. Si dimostri che il tipo dell'uguaglianza proposizionale di Martin-Löf è isomorfo a quello dell'uguaglianza proposizionale con path induction.

13. Si provi a stabilire se il tipo dell'uguaglianza proposizionale di Leibniz rende ammissibili le regole di tipo uguaglianza proposizionale di Martin-Löf.

14. Si stabilisca se dato un tipo A *type* $[\Gamma]$ e termini $a \in A$ $[\Gamma]$ e $b \in A$ $[\Gamma]$, il tipo uguaglianza proposizionale à la Leibniz $\mathsf{Id_l}(A, a, b)$ è equivalente a quello di Martin-Löf $\mathsf{Id}(A, a, b)$.

    I due tipi sono anche isomorfi tra loro?

15. Mostrare che per ogni tipo $List(C)$ esiste una proof-term del tipo

$$\begin{array}{l} \mathsf{Id}(List(C),\ z\ \ z'\ ) \ \leftrightarrow \\ \qquad \exists_{l \in List(R)}\ (\ \mathsf{Id}(\ List(C),\ \overline{\pi_1}(l),\ z)\ \ \&\ \ \mathsf{Id}(\ List(C),\ \overline{\pi_2}(l),\ z')\ )\ \textit{type} \\ \qquad\qquad\qquad\qquad\qquad\qquad [z \in List(C), z' \in List(C)] \end{array}$$

    ove

$$R \equiv \Sigma_{x \in C}\, \Sigma_{y \in C}\, \mathsf{Id}(\, C\,,\ x\,,\ y\,)$$

    e $\overline{\pi_1}(l)$ e $\overline{\pi_2}(l)$ sono rispettivamente la prima proiezione e la seconda proiezione di R estese alle liste su di lui.

3.6 Tipo somma indiciata

Regole del tipo somma indiciata o **Strong Indexed sum type**:
Strong Indexed sum type

F-Σ) $\dfrac{C(x)\ \ type\ [\Gamma, x \in B]}{\Sigma_{x \in B} C(x)\ type\ [\Gamma]}$ I-Σ) $\dfrac{b \in B\ [\Gamma] \qquad c \in C(b)\ [\Gamma] \qquad C(x)\ type\ [\Gamma, x \in B]}{\langle b, c \rangle \in \Sigma_{x \in B} C(x)\ [\Gamma]}$

$$\text{E-}\Sigma)\ \dfrac{\begin{array}{l} M(z)\ type\ [\Gamma, z \in \Sigma_{x \in B} C(x)] \\ t \in \Sigma_{x \in B} C(x)\ [\Gamma] \qquad e(x, y) \in M(\langle x, y \rangle)\ [\Gamma, x \in B, y \in C(x)] \end{array}}{El_{\Sigma}(t, e) \in M(t)\ [\Gamma]}$$

$$\text{C-}\Sigma)\ \dfrac{\begin{array}{l} M(z)\ type\ [\Gamma, z \in \Sigma_{x \in B} C(x)] \\ b \in B \qquad c \in C(b) \qquad e(x, y) \in M(\langle x, y \rangle)\ [\Gamma, x \in B, y \in C(x)] \end{array}}{El_{\Sigma}(\ \langle b, c \rangle,\ e\ ) = e(b, c) \in M(\langle b, c \rangle)\ [\Gamma]}$$

Descriviamo di seguito le regole che garantiscono che i costruttori di tipo conservano l'uguaglianza tra tipi:

eq-F-Id

$$\text{eq-list)}\ \dfrac{B_1 = B_2\ type\ [\Gamma] \quad C_1(x) = C_2(x)\ type\ [\Gamma, x \in B_1]}{\Sigma_{x \in B_1} C_1(x) = \Sigma_{x \in B_2} C_2(x)\ type\ [\Gamma]}$$

Diamo ora di seguito le regole che garantiscono che i costruttori di termini conservano l'uguaglianza computazionale tra termini:

$$\text{eq-I-}\Sigma)\ \dfrac{b_1 = b_2 \in B\ [\Gamma] \qquad c_1 = c_2 \in C(b_1)\ [\Gamma] \qquad C(x)\ type\ [\Gamma, x \in B]}{\langle b_1, c_1 \rangle = \langle b_2, c_2 \rangle \in \Sigma_{x \in B} C(x)\ [\Gamma]}$$

$$\text{eq-E-}\Sigma)\ \dfrac{\begin{array}{l} M(z)\ type\ [\Gamma, z \in \Sigma_{x \in B} C(x)] \\ t_1 = t_2 \in \Sigma_{x \in B} C(x)\ [\Gamma] \qquad e_1(x, y) = e_2(x, y) \in M(\langle x, y \rangle)\ [\Gamma, x \in B, y \in C(x)] \end{array}}{El_{\Sigma}(t_1, e_1) = El_{\Sigma}(t_1, e_1) \in M(t)\ [\Gamma]}$$

Esercizi:

1. Provare a scrivere le regole del tipo prodotto cartesiano $A \times B$ di un tipo A con un tipo B.
2. Come rappresentare con i tipi finora descritti il tipo dei naturali positivi? e il tipo delle liste non vuote?
3. Si definisca il tipo dei naturali positivi Nat^+ e una funzione

$$\mathbf{pred}(x) \in \mathsf{Nat}\ [x \in \mathsf{Nat}^+]$$

che calcola il predecessore.

4. Si definisca il tipo delle liste non vuote $List(\mathsf{Nat})^*$ e le funzioni

   (a) l'operazione *back* che di una lista non vuota ne estrae la lista meno l'elemento di testa;

   (b) l'operazione $front$ che data una lista non vuota ne estrae la lista meno l'elemento di inizio;

   (c) l'operazione *last* che data una lista non vuota ne estrae l'elemento in testa;

   (d) l'operazione $first$ che data una lista non vuota ne estrae l'elemento all'inizio.

5. Si definiscano *termini untyped* con i costruttori del tipo dei numeri naturali secondo la seguente grammatica:

$$t \ \equiv \ v \ \mid \ \langle t_1, t_2 \rangle \ \mid \ \mathsf{El}_\Sigma(t_1,\ (x,y).e(x,y)\ )$$

con $v \in \{\ x, y, w, z\ \} \bigcup \{x_i \ \mid \ i \in Nat\}$ ovvero consideriamo come variabili le ultime lettere dell'alfabeto inglese e poi tutte le variabili ottenute ponendo alla variabile x un indice con indice che varia nei numeri naturali.

Definiamo la *riduzione ad un passo di computazione* $\to_1$ sui termini untyped definiti sopra, convenendo di usare la seguente abbreviazione

$$\mathsf{El}_\Sigma(t, e) \text{ al posto di } \mathsf{El}_\Sigma(t, (x,y).e(x,y)\ )$$

tramite il seguente assioma:

$$\beta_\Sigma\text{-red}) \qquad \mathsf{El}_\Sigma(\langle b, c\rangle, e\ ) \qquad \to_1 \qquad e(b,c)$$

e le seguenti regole:

$$\mathrm{I}_1\text{-red}) \quad \frac{b_1 \ \to_1 \ b_2}{\langle b_1, c\rangle \ \to_1 \ \langle b_2, c\rangle} \qquad \mathrm{I}_2\text{-red}) \quad \frac{c_1 \ \to_1 \ c_2}{\langle b, c_1\rangle \ \to_1 \ \langle b, c_2\rangle}$$

$$\text{red}_I) \quad \frac{t_1 \ \to_1 \ t_2}{\mathsf{El}_{\mathsf{Nat}}(t_1, e) \ \to_1 \ \mathsf{El}_{\mathsf{Nat}}(t_2, e)}$$

   (a) Poi, rispetto alla relazione $\to_1$ si definiscano termini in forma normale (ovvero non riducibile a nessun altro termine) e termini riducibile secondo la relazione data.

   (b) Posto che ogni termine si riduca ad un numero finito di passi, anche nessuno, a forma normale, si stabilisca se si riesce a derivare l'uguaglianza

$$\lambda x.\mathsf{Ap}(z, x) = z \in \mathsf{Nat} \ \to \ \mathsf{Nat} \qquad [z \in \mathsf{Nat} \ \to \ \mathsf{Nat}]$$

3.7 Tipo prodotto dipendente

Regole del tipo dei prodotti dipendenti:

Tipo prodotto dipendente

F-Π) $\dfrac{C(x)\ type\ [\Gamma, x \in B]}{\Pi_{x\in B}C(x)\ type\ [\Gamma]}$ I-Π) $\dfrac{c(x) \in C(x)\ [\Gamma, x \in B]}{\lambda x^B.c(x) \in \Pi_{x\in B}C(x)\ [\Gamma]}$

E-Π) $\dfrac{b \in B\ [\Gamma] \qquad f \in \Pi_{x\in B}C(x)\ [\Gamma]}{\mathsf{Ap}(f, b) \in C(b)\ [\Gamma]}$

βC-Π) $\dfrac{b \in B\ [\Gamma] \qquad c(x) \in C(x)\ [\Gamma, x \in B]}{\mathsf{Ap}(\lambda x^B.c(x), b) = c(b) \in C(b)\ [\Gamma]}$

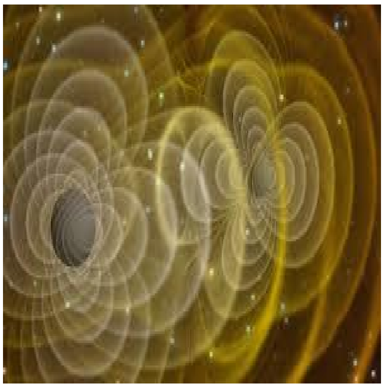
Descriviamo di seguito le regole che garantiscono che i costruttori di tipo conservano l'uguaglianza tra tipi:

eq-F-Π) $\dfrac{B_1 = B_2\ type\ [\Gamma] \qquad C_1(x) = C_2(x)\ type\ [\Gamma, x \in B_1]}{\Pi_{x\in B_1}C_1(x) = \Pi_{x\in B_2}C_2(x)\ type\ [\Gamma]}$

Diamo ora di seguito le regole che garantiscono che i costruttori di termini conservano l'uguaglianza computazionale tra termini:

ξ) $\dfrac{c_1(x) = c_2(x) \in C(x)\ [\Gamma, x \in B]}{\lambda x^B.c_1(x) = \lambda x^B.c_2(x) \in \Pi_{x\in B}C(x)\ [\Gamma]}$

eq- E-Π) $\dfrac{b_1 = b_2 \in B\ [\Gamma] \qquad f_1 = f_2 \in \Pi_{x\in B}C(x)\ [\Gamma]}{\mathsf{Ap}(f_1, b_1) = \mathsf{Ap}(f_2, b_2) \in C(b)\ [\Gamma]}$



Esercizi:

1. Provare a scrivere le regole del tipo delle funzioni $A \to B\ [\Gamma]$ da un tipo A ad un tipo B.
2. Come rappresentare on i tipi finora descritti il tipo delle funzioni tra naturali positivi?
3. Si consideri l'insieme dei termini Term ottenuti in modo induttivo dalle seguenti clausole:
   (a) ciascuna variabile x o y o z o w oppure ciascuna variabile x_i o y_j o z_k o w_l indiciata sui numeri naturali (ovvero i, j, k, l rappresentano dei numeri naturali) è un termine in Term;
   (b) se f è un termine in Term allora $\lambda x.f$ è un termine in Term per ogni variabile x in Term;
   (c) se f e b sono entrambi termini in Term allora $\mathsf{Ap}(f, b)$ è un termine in Term.

Si definisca la relazione di riduzione ad un passo $\rightarrow_1$ con le seguenti regole e assioni:

β-red $\qquad \mathsf{Ap}(\lambda x.c, b) \;\rightarrow_1\; c[x/b]$

$$\frac{\mathbf{f_1} \;\rightarrow_1\; \mathbf{f_2}}{\mathsf{Ap}(\mathbf{f_1}, \mathbf{b}) \rightarrow_1 \mathsf{Ap}(\mathbf{f_2}, \mathbf{b})} \qquad \frac{\mathbf{b_1} \;\rightarrow_1\; \mathbf{b_2}}{\mathsf{Ap}(\mathbf{f}, \mathbf{b_1}) \rightarrow_1 \mathsf{Ap}(\mathbf{f}, \mathbf{b_2})} \qquad \frac{\mathbf{c_1} \;\rightarrow_1\; \mathbf{c_2}}{\lambda \mathbf{x}.\mathbf{c_1} \rightarrow_1 \lambda \mathbf{x}.\mathbf{c_2}}$$

$$\mathsf{Ap}(\lambda x.c, b) \;\rightarrow_1\; c[x/b]$$

ove con la scrittura $c[x/b]$ si intende che in c tutte le occorrenze libere di x sono sostituite con b.

(a) Poi, rispetto alla relazione $\rightarrow_1$ si definiscano termini in forma normale (ovvero non riducibile a nessun altro termine) e termini riducibile secondo la relazione data.

(b) Posto che ogni termine si riduca ad un numero finito di passi, anche nessuno, a forma normale, si stabilisca se si riesce a derivare l'uguaglianza

$$\lambda x.\mathsf{Ap}(z, x) = z \in \mathsf{Nat} \;\rightarrow\; \mathsf{Nat} \qquad [z \in \mathsf{Nat} \;\rightarrow\; \mathsf{Nat}]$$

3.8 Rappresentazione dei tipi semplici in teoria dei tipi dipendenti

Nella teoria dei tipi dipendenti con i tipi descritti finora è possibile interpretare alcuni tipi semplici, ovvero non dipendenti, come il tipo prodotto cartesiano e il tipo delle funzioni come segue.

3.8.1 Tipo prodotto cartesiano

Per rappresentare in teoria dei tipi dipendenti il tipo prodotto $B \times C$ (si riscrivano le sue regole per esercizio!) dati due tipi semplici B *type* [ ] e C *type* [ ] basta definire

$$B \times C \;\equiv\; \Sigma_{x \in B}\, C$$

e le corrispondenti regole diventano

Tipo prodotto cartesiano

$$\text{F-}\times)\ \frac{B\ \mathit{type}\ [\Gamma] \qquad C\ \mathit{type}\ [\Gamma]}{B \times C\ \mathit{type}\ [\Gamma]} \qquad \text{I-}\times)\ \frac{b \in B\ [\Gamma] \qquad c \in C\ [\Gamma]}{\langle b, c\rangle \in B \times C\ [\Gamma]}$$

$$\text{E}_1\text{-}\times)\ \frac{d \in B \times C\ [\Gamma]}{\pi_1(d) \in B\ [\Gamma]} \qquad \text{E}_2\text{-}\times)\ \frac{d \in B \times C\ [\Gamma]}{\pi_2(d) \in C\ [\Gamma]}$$

$$\beta_1\text{-}\times)\ \frac{b \in B\ [\Gamma] \qquad c \in C\ [\Gamma]}{\pi_1(\,\langle b, c\rangle\,) = b \in B\ [\Gamma]} \qquad \beta_2\text{-}\times)\ \frac{b \in B\ [\Gamma] \qquad c \in C\ [\Gamma]}{\pi_2(\,\langle b, c\rangle\,) = c \in C\ [\Gamma]}$$

Lemma 3.8 Supposto che B *type* [ ] e C *type* [ ] siano derivabili nella teoria dei tipi con le regole finora introdotte, allora in tal teoria è pure derivabile

$$\Sigma_{x \in B}\, C\ \mathit{type}\ [\,]$$

e lo si può usare per interpretare il tipo semplice $B \times C$ con le sue regole di introduzione, eliminazione e relative β-conversioni.

dim. Lo si provi per **esercizio** dopo aver mostrato il lemma dell'indebolimento opportuno.

Lemma 3.9 Per ogni tipo dipendente $C(x)$ *type* $[\Gamma, x \in B]$ è possibile definire delle proiezioni

$$\pi_1(w) \in B\ [\Gamma, w \in \Sigma_{x \in B}\, C(x)] \qquad \pi_2(w) \in C(\pi_1(w))\ [\Gamma, w \in \Sigma_{x \in B}\, C(x)]$$

tali che si derivano

$$\pi_1(\,\langle x, y\rangle\,) = x \in B\ [\Gamma, x \in B, y \in C(x)] \qquad \pi_2(\,\langle x, y\rangle\,) = y \in C(x)\ [\Gamma, x \in B, y \in C(x)]$$

3.8.2 Tipo delle funzioni (o tipo freccia)

Per rappresentare in teoria dei tipi dipendenti il tipo delle funzioni $B \to C$ (si riscrivano le sue regole per esercizio!) dati due tipi semplici B *type* [ ] e C *type* [ ] basta definire

$$B \to C \;\equiv\; \Pi_{x\in B}\, C$$

e le corrispondenti regole diventano

Tipo delle funzioni

F-$\to$) $\dfrac{B\ type\ [\Gamma] \qquad\qquad C\ type\ [\Gamma]}{B \to C\ type\ [\Gamma]}$ I-$\to$) $\dfrac{c(x) \in C\ [\Gamma, x \in B]}{\lambda x^{B}.c(x) \in B \to C\ [\Gamma]}$

E-$\to$) $\dfrac{a \in A\ [\Gamma] \qquad\qquad f \in B \to C\ [\Gamma]}{\mathsf{Ap}(f, b) \in C\ [\Gamma]}$

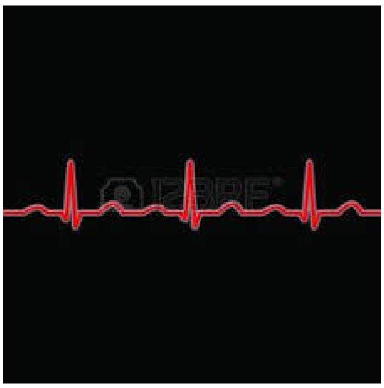
βC-$\to$) $\dfrac{b \in B\ [\Gamma] \qquad c(x) \in C\ [\Gamma, x \in B]}{\mathsf{Ap}(\lambda x^{B}.c(x), b) = c(b) \in C\ [\Gamma]}$

Lemma 3.10 Supposto che B *type* [ ] e C *type* [ ] siano derivabili nella teoria dei tipi con le regole finora introdotte, allora in tal teoria è pure derivabile

$$\Pi_{x\in B}\, C\ type\ [\ ]$$

e lo si può usare per interpretare il tipo semplice $B \to C$ con le sue regole di introduzione, eliminazione e relativa β-conversione.

dim. Lo si provi per **esercizio** dopo aver mostrato il lemma dell'indebolimento opportuno.



Esercizi.

1. La η-conversione del tipo prodotto ovvero

$$\langle \pi_1(z), \pi_2(z)\rangle = z \in A \times B\ [z \in A \times B]$$

   è derivabile ?

É derivabile la corrispondente uguaglianza proposizionale, ovvero esiste un proof-term **pf** tale che

$$\mathbf{pf} \in \mathsf{Id}(\, A \times B, \langle \pi_1(z), \pi_2(z) \rangle, z \,) \ [z \in A \times B]$$

è derivabile nella teoria dei tipi dipendenti?

2. Si dimostri che esiste un proof-term **pf** del tipo

$$\mathbf{pf} \in P(y) \ [x \in A, y \in A, z \in P(x), w \in \mathsf{Id}(A, x, y)]$$

3. Si dimostri che esiste un proof-term **pf** del tipo

$$\mathbf{pf} \in \mathsf{I}d(A, x, z) \ [x \in A, y \in A, z \in A, w_1 \in \mathsf{Id}(A, x, y), w_2 \in \mathsf{Id}(A, y, z)]$$

3.9 Proprietà di manipolazione dei contesti

I costruttori di tipo somma indiciata forte e prodotto dipendente possono essere utilizzati rispettivamente per raggruppare le assunzioni di un contesto e toglierle secondo le seguenti proprietà.

Proposition 3.11 Se $c(x, y) \in C(x, y) \ [\Gamma, x \in A, y \in B(x)]$ è derivabile allora anche $c(\pi_1(z), \pi_2(z)) \in C(x, y) \ [\Gamma, z \in \Sigma_{x \in A} B(x)]$ è derivabile.
Viceversa se $d(z) \in D(z) \ \ [\Gamma, z \in \Sigma_{x \in A} B(x)]$ è derivabile allora anche $d(\langle x, y \rangle) \in D(\langle x, y \rangle) \ [\Gamma, x \in A, y \in B(x)]$ è derivabile.

Dim. Per esercizio.

Corollary 3.12 Se $c(x, y) \in C(x, y) \ [\Gamma, x \in A, y \in B]$ è derivabile allora anche $c(\pi_1(z), \pi_2(z)) \in C(x, y) \ [\Gamma, z \in A \times B]$ è derivabile.
Viceversa se $d(z) \in D(z) \ \ [\Gamma, z \in A \times B]$ è derivabile allora anche $d(\langle x, y \rangle) \in D(\langle x, y \rangle) \ [\Gamma, x \in A, y \in B]$ è derivabile.

Dim. Per esercizio.

Proposition 3.13 Se $c(x) \in C(x) \ [\Gamma, x \in A]$ è derivabile allora anche $\lambda x.c(x) \in \Pi_{x \in A} C(x) \ [\Gamma]$ è derivabile.
Viceversa se $d \in \Pi_{x \in A} C(x) \ \ [\Gamma$ è derivabile allora anche $\mathsf{Ap}(d, x) \in C(x) \ [\Gamma, x \in A]$ è derivabile.

Dim. Per esercizio.

Corollary 3.14 Se $c(x) \in C \ [\Gamma, x \in A]$ è derivabile allora anche $\lambda x.c(x) \in A \ \rightarrow \ C \ [\Gamma]$ è derivabile.
Viceversa se $d \in A \ \rightarrow \ C \ \ [\Gamma$ è derivabile allora anche $\mathsf{Ap}(d, x) \in C \ [\Gamma, x \in A]$ è derivabile.

Dim. Per esercizio.

4 Differenze tra uguaglianza definizionale e uguaglianza proposizionale

In una teoria dei tipi come quella intensionale di Martin-Löf, che chiamiamo **MLTT** per brevità, grazie al teorema di forma normale si dimostra che:

Proposition 4.1 Per ogni tipo chiuso (ovvero senza variabili libere) A *type* [ ] derivabile in **MLTT** e per ogni coppia di termini chiusi (senza variabili libere) $a \in A$ [ ] e $b \in A$ [ ] derivabili in **MLTT**, se esiste un termine p tale che

$$p \in \mathsf{Id}(A, a, b)\ [\,]$$

è derivabile in **MLTT**, allora si deriva in **MLTT** pure

$$a = b \in A\ [\,]$$

e quindi la forma normale di a è lo stessa che quella di b ovvero

$$\mathsf{nf}(a) \equiv \mathsf{nf}(b)$$

Dim. Per il teorema di riduzione a forma normale il proof-term $\mathbf{p}$ si riduce con un numero finito di passi (anche zero) necessariamente ad un termine in forma normale $\mathsf{id}(\mathbf{c})$ per un qualche $\mathbf{c}$ e questo $\mathbf{c}$ deve essere definizionalmente uguale al termine $\mathbf{a}$ e al termine $\mathbf{b}$ e quindi deve essere la loro forma normale.

Inoltre grazie alle regole di sostituzione e della regola *conv*) si dimostra pure che:

Proposition 4.2 Per ogni insieme A *type* $[\Gamma]$ derivabile in **MLTT** e per ogni coppia di termini $a \in A\ [\Gamma]$ e $b \in A\ [\Gamma]$ derivabili in **MLTT**, se in **MLTT** si deriva

$$a = b \in A\ [\Gamma]$$

allora esiste un termine p tale che

$$p \in \mathsf{Id}(A, a, b)\ [\Gamma]$$

e possiamo pure scegliere $p \equiv \mathsf{id}(a)$ oppure $p \equiv \mathsf{id}(b)$.

Dim. lasciata al lettore come esercizio.

Da cui si deduce come caso particolare:

Corollary 4.3 Per ogni tipo chiuso A *type* [ ] derivabile in **MLTT** e per ogni coppia di termini chiusi $a \in A$ [ ] e $b \in A$ [ ] derivabili in **MLTT**, se in **MLTT** si deriva

$$a = b \in A\ [\,]$$

allora esiste un termine p tale che

$$p \in \mathsf{Id}(A, a, b)\ [\,]$$

e possiamo pure scegliere $p \equiv \mathsf{id}(a)$.

Esempio: Con le regole dei tipi finora definite possiamo definire

$$x + y \ \equiv\ \mathsf{El}_{\mathsf{Nat}}(x, y, (x', z).\mathsf{succ}(z))$$

per $x \in \mathsf{Nat}$ e $y \in \mathsf{Nat}$ lasciando al lettore come esercizio di derivare

$$\mathsf{El}_{\mathsf{Nat}}(x, y, (x', z).\mathsf{succ}(z)) \in \mathsf{Nat}\ [x \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

Inoltre si deriva pure

$$0 = 0 + 0 = 0 \in \mathsf{Nat}$$

(il lettore derivi questo giudizio per esercizio) e quindi dalla proposizione 4.5 concludiamo che si deriva pure

$$\mathsf{id}(0) \in \mathsf{Id}(\mathsf{Nat}, 0, 0+0)$$

Tutto ciò ci permette di concludere che *l'uguaglianza proposizionale è* **equivalente** *all'uguaglianza definizionale per i termini e insiemi derivabili senza ipotesi*

$$\boxed{\begin{array}{c} a = b \in A\ [\,]\ \text{è derivabile} \\ \text{sse} \\ p \in \mathsf{Id}(A, a, b)\ [\,]\ \text{è derivabile per qualche termine } p \\ \text{sse} \\ \mathsf{nf}(a)\ \equiv\ \mathsf{nf}(b) \end{array}}$$

e siccome l'uguaglianza delle forme normale è **decidibile** in **MLTT** sappiamo che l'uguaglianza proposizionale di termini di insiemi chiusi (ovvero senza variabili libere) è pure decidibile.

Esempio: Si noti che NON esiste un termine p tale che

$$p \in \mathsf{Id}(\mathsf{Nat}\ \to \mathsf{Nat},\ \lambda x.x+0\ ,\ \lambda x.x\ )$$

perchè se esistesse otterremmo che

$$\mathsf{nf}(\lambda x.x+0)\ \equiv\ \mathsf{nf}(\lambda x.x)$$

ma ciò NON vale perchè sono due programmi in forma normale diversi ovvero

$$\mathsf{nf}(\lambda x.x+0)\ \equiv\ \lambda x.x+0 \qquad \text{che è diversa da} \qquad \mathsf{nf}(\lambda x.x)\ \equiv\ \lambda x.x$$

!! Questo lo si vede perchè non si possono applicare ai termini regole di conversioni che li semplificano. Le due forme normali sono tali anche se in **MLTT**, come già specificato, vi sono le regole di conservazione dell'uguaglianza definizionale per ogni costruttore tipato, il che significa che vi è pure la regola chiamata ξ che afferma che il costruttore di binding $\lambda x.-$ conserva l'uguaglianza computazionale ovvero vale

$$\xi)\ \ \frac{c(x) = d(x) \in C(x)\ [\Gamma, x \in B]}{\lambda x^B.c(x) = \lambda x^B.d(x) \in \Pi_{x\in B}C(x)\ [\Gamma]}$$

che dà origine alla regola di riduzione

$$\xi\text{-red})\ \ \frac{c(x) \to_1 d(x)}{\lambda x^B.c(x) \to_1 \lambda x^B.d(x)}$$

la quale permette di ridurre un programma funzionale dentro al corpo di una λ-espressione. Ma la riduzione ξ-red nel nostro caso si applica a vuoto perchè i corpi delle due lambda espressioni

$$\mathsf{nf}(x+0)\ \equiv\ x+0 \qquad \text{che è diversa da} \qquad \mathsf{nf}(x)\ \equiv\ x$$

sono comunque in forma normale e sono diversi!

Questo controesempio permette di concludere il seguente fatto

Proposition 4.4 In **MLTT** NON vale il principio di estensionalità delle funzioni ovvero NON esiste un termine prova p tale che

$$p \in \Pi_{x\in A}\ \mathsf{Id}(B(x),\ c(x),\ d(x)\ ) \ \rightarrow\ \mathsf{Id}(\ \Pi_{x\in A}\ B(x)\,,\ \lambda x^{B}.c(x),\ \lambda x^{B}.d(x)\ )$$

Dim. si lascia al lettore dimostrare che non esiste p tale che

$$p \in \Pi_{x\in \mathsf{Nat}}\ \mathsf{Id}(\mathsf{Nat},\ x+0,\ x\ ) \ \rightarrow\ \mathsf{Id}(\ \Pi_{x\in \mathsf{Nat}}\ \mathsf{Nat}\,,\ \lambda x^{\mathsf{Nat}}.x+0,\ \lambda x^{\mathsf{Nat}}.x\ )$$

in quanto si dimostra che esiste un termine prova (lasciato da derivare al lettore)

$$q(x) \in \mathsf{Id}(\mathsf{Nat},\ x+0,\ x\ )\ [x \in \mathsf{Nat}]$$

applicando le regole di eliminazione dei numeri naturali. Quindi deduciamo che esiste un termine prova (lasciato da derivare al lettore)

$$\lambda x^{\mathsf{Nat}}.q(x) \in \Pi_{x\in \mathsf{Nat}}\ \mathsf{Id}(\mathsf{Nat},\ x+0,\ x\ )$$

che unito alla supposta esistenza di p permette di trovare un termine prova (lasciato da derivare al lettore per esercizio)

$$q' \in \mathsf{Id}(\ \Pi_{x\in \mathsf{Nat}}\ \mathsf{Nat}\,,\ \lambda x^{\mathsf{Nat}}.x+0,\ \lambda x^{\mathsf{Nat}}.x\ )$$

che sappiamo non esistere.

Con il controesempio nella dimostrazione precedente si dimostra pure che

Proposition 4.5 Per ogni insieme A *type* $[\Gamma]$ derivabile in **MLTT** e per ogni coppia di termini $a \in A\ [\Gamma]$ e $b \in A\ [\Gamma]$ derivabili in **MLTT**, se in **MLTT** esiste un termine p tale che si deriva

$$p \in \mathsf{Id}(A, a, b)\ [\Gamma]$$

allora NON è sempre vero che si deriva

$$a = b \in A\ [\Gamma]$$

Dim. Sappiamo già che esiste un termine prova tale che

$$q(x) \in \mathsf{Id}(\mathsf{Nat},\ x+0,\ x\ )\ [x \in \mathsf{Nat}]$$

risulta derivabile. Poi si osservi che

$$x + 0 = x \in \mathsf{Nat}\ [x \in \mathsf{Nat}]$$

NON è derivabile in **MLTT** per la proposizione 4.1 perchè $x + 0$ è in forma normale come pure la variabile x ma le forme normali sono diverse ovvero

$$x + 0 \equiv \mathsf{nf}(x + 0) \qquad \text{che è diverso da} \qquad x \equiv \mathsf{nf}(x)$$

Quindi concludiamo che

$$\boxed{\begin{array}{c} a = b \in A\ [\Gamma] \text{ è derivabile} \\ \Downarrow \\ p \in \mathsf{Id}(A, a, b)\ [\Gamma] \text{ è derivabile per qualche termine } p \\ \text{ma NON vale il viceversa } \Uparrow!! \end{array}}$$

ovvero la dimostrabilità dell'uguaglianza proposizionale sotto contesto NON implica l'uguaglianza definizionale computabile anche perchè l'uguaglianza proposizionale è in genere **indecidibile**, e lo è per **MLTT**, mentre quella definizionale è **decidibile** per teorie dei tipi intensionali come per **MLTT**.
Osservazione sull' uso delle regole di uguaglianza definizionale Si osservi che se vogliamo dimostrare che nella teoria dei tipi con tipo dei numeri naturali e identità proposizionale c'è un proof-term **pr** tale per cui si deriva

$$\mathbf{pr} \in \mathsf{Id}(Nat, x + 0, x)\ [x \in Nat]$$

Per definire **pr** è necessario usare altre regole strutturali oltre all'assunzione di variabile e alle regole di sostituzioni. Infatti posto che

$$x + y \equiv \mathsf{El}_{Nat}(y, x, (x', z).\mathsf{succ}(z)) \in \mathsf{Nat}\ [x \in \mathsf{Nat}, y \in \mathsf{Nat}]$$

ed anche

$$x + 0 \equiv \mathsf{El}_{Nat}(0, x, (x', z).\mathsf{succ}(z)) = x \in \mathsf{Nat}\ [x \in \mathsf{Nat}]$$

$$\mathsf{Id}(\mathsf{Nat}, x, x) = \mathsf{Id}(\mathsf{Nat}, x + 0, x)\ type\ [x \in \mathsf{Nat}]$$

e sappiamo anche derivare

$$\mathsf{id}(x) \in \mathsf{Id}(\mathsf{Nat}, x, x)\ [x \in \mathsf{Nat}]$$

ma poi come proseguiamo?? A questo punto ci servono le regole che dicono che *tipi uguali hanno gli stessi elementi* ovvero la regola *conv*) di sezione 2.1.4.

5 Come tradurre la logica predicativa con uguaglianza in teoria dei tipi

È possibile interpretare i connettivi logici e i quantificatori universali ed esistenziali in teoria dei tipi a patto di *restringere ad un tipo il dominio di quantificazione delle variabili* e dotare *ogni termine di un tipo a partire dalle variabili.*
Nel seguito indichiamo con le metavariabili ϕ, ψ etc... le **formule della logica predicativa** $Form(\Gamma)$ con variabili tipate che occorrono in un contesto Γ della teoria dei tipi di Martin-Löf generate come segue

1. la costante falso $\perp$ è una formula in $Form(\Gamma)$;
2. la costante vero $\mathtt{tt}$ è una formula in $Form(\Gamma)$;
3. Se ϕ e ψ sono formule in $Form(\Gamma)$ allora $\phi \vee \psi$ è una formula in $Form(\Gamma)$;
4. Se ϕ e ψ sono formule in $Form(\Gamma)$ allora $\phi \rightarrow \psi$ è una formula in $Form(\Gamma)$;
5. Se ϕ e ψ sono formule in $Form(\Gamma)$ allora $\phi \,\&\, \psi$ è una formula in $Form(\Gamma)$;
6. Se ϕ è una formula in $Form(\Gamma)$ allora $\neg\phi$ è una formula in $Form(\Gamma)$;
7. Se ϕ è una formula in $Form(\Gamma, x \in A)$ allora $\forall_{x\in A}\ \phi$ è una formula in $Form(\Gamma)$;
8. Se $t \in A\ [\Gamma]$ e $s \in A\ [\Gamma]$ sono derivabili nella teoria dei tipi di Martin-Löf allora $t =_A s$ è una formula in $Form(\Gamma)$.

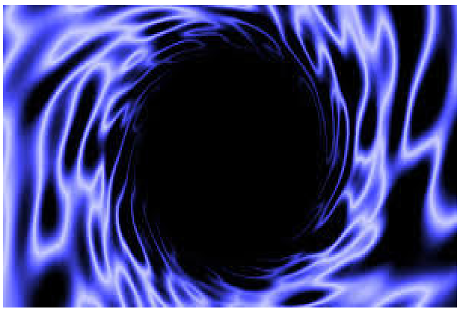
Inoltre indichiamo con **Form** l'unione di tutte le formule $Form(\Gamma)$ per ogni contesto Γ in teoria dei tipi, ovvero nella teoria dei tipi di Martin-Löf con i costrutti definiti finora.
Nel seguito infatti indicheremo con il termine "*giudizio derivabile*" un *giudizio derivabile nella teoria dei tipi di Martin-Löf.*
È possibile interpretare le formule della logica predicativa nella teoria dei tipi con i tipi finora descritti e l'aggiunta del tipo vuoto.

Il tipo vuoto è definito in modo induttivo dall'assenza di elementi come segue:

Empty type

F-N_0) $\dfrac{\Gamma\ cont}{\mathsf{N}_0\ type\ [\Gamma]}$

E-N_0) $\dfrac{t \in \mathsf{N}_0\ [\Gamma] \qquad\qquad M(z)\ type\ [\Gamma, z \in \mathsf{N}_0]}{El_{\mathsf{N}_0}(t) \in M(t)\ [\Gamma]}$



Traduzione propositions-as-sets di Curry-Howard-Martin-Löf (in breve traduzione CHM)

Def. 5.1 Definiamo di seguito **l'interpretazione di formule logiche del linguaggio del primo ordine con uguaglianza** a partire dal **predicato di uguaglianza proposizionale definito su termini tipati nella teoria dei tipi** con i tipi definiti finora:

$(\perp \in Form(\Gamma))^I$	$=$	$\mathsf{N}_0\ type\ [\Gamma]$
$(\mathtt{tt} \in Form(\Gamma))^I$	$=$	$\mathsf{N}_1\ type\ [\Gamma]$
$(\phi\ \&\ \psi \in Form(\Gamma))^I$	$=$	$\phi^I\ \times\ \psi^I\ type\ [\Gamma]$
$(\phi\ \vee\ \psi \in Form(\Gamma))^I$	$=$	$\phi^I\ +\ \psi^I\ type\ [\Gamma]$
$(\phi\ \rightarrow\ \psi \in Form(\Gamma))^I$	$=$	$\Pi_{z\in\phi^I}\ \psi^I\ type\ [\Gamma]$
$(\neg\phi)^I$	$=$	$\Pi_{z\in\phi^I}\ \mathsf{N}_0\ type\ [\Gamma]$
$(t = s)^I\ [\Gamma]$	$=$	$\mathsf{Id}(A, t, s)\ type\ [\Gamma]$
per termini $t \in A\ [\Gamma]$ ed $s \in A\ [\Gamma]$ derivabili in teoria dei tipi		
$(\forall_{x\in A}\ \phi)^I\ [\Gamma]$	$=$	$\Pi_{x\in A^I}\ \phi^I\ type\ [\Gamma]$
$(\exists_{x\in A}\ \phi)^I\ [\Gamma]$	$=$	$\Sigma_{x\in A^I}\ \phi^I\ type\ [\Gamma]$

ove si suppone che $t \in A\ [\Gamma]$ e $s \in A\ [\Gamma]$ siano derivabili in teoria dei tipi.

Invece di interpretare le formule possiamo pensarle *abbreviazioni per tipi* e quindi possiamo *aggiungere alla teoria dei tipi il linguaggio predicativo di predicati con variabili tipate* nel senso di scrivere

$$\phi\ prop\ [\Gamma] \qquad \text{come abbreviazione di} \qquad \phi^I\ type\ [\Gamma]$$

e coerentemente con tal abbreviazione scriviamo pure

$$pf \in \phi\ [\Gamma] \qquad \text{come abbreviazione di} \qquad pf \in \phi^I\ [\Gamma]$$

e

$$\Gamma, x \in \phi\ cont \qquad \text{come abbreviazione di} \qquad \Gamma, x \in \phi^I\ cont$$

Più precisamente:

Def. 5.2 *Indichiamo con il giudizio*

$$\alpha\ prop\ [\Gamma]$$

un tipo α type $[\Gamma]$ derivabile nella teoria dei tipi finora descritta, ove α è la traduzione di una formula del linguaggio predicativo con uguaglianza descritto in definizione 5.1.

Def. 5.3 Date le proposizioni derivabile in teoria dei tipi

$$\phi\ prop\ [\Gamma] \qquad \alpha_i\ prop\ [\Gamma]$$

per $i = 1, \ldots n$, introduciamo un nuovo giudizio

$$\phi\ true\ [\Gamma, \alpha_1\ true, \ldots, \alpha_n\ true]$$

e lo diciamo **derivabile nella teoria dei tipi** $\mathcal{T}$ se in $\mathcal{T}$ esiste un proof-term **pf** tale che

$$\mathsf{pf} \in \phi\ [\Gamma, x_1 \in \alpha_1, \ldots, x_n \in \alpha_n]$$

è derivabile in $\mathcal{T}$.

Si interpreta poi un sequente della logica del primo ordine con uguaglianza a partire dai predicati su termini tipati come segue:

$$(\ \Sigma\ \vdash \Delta\ )^I\ \equiv\ (\Delta^\vee)^I\ type\ [\Gamma, x \in (\Sigma^\&)^I]$$

supposto che le variabili delle varie formule siano tipate nel contesto Γ
(anche se $(\Delta^\vee)^I$ come tipo NON dipende dall'assunzione $x \in (\Sigma^\&)^I$
e che

$$\boxed{\begin{array}{ll} \Sigma^\& \equiv \mathtt{tt} \text{ (costante vero)} & \text{se } \Sigma \text{ è la lista vuota} \\ \Sigma^\& \equiv (\,\mathtt{pr_1}\&\mathtt{pr_2}\,)\ldots\&\mathtt{pr_n} & \text{se } \Sigma \equiv \mathtt{pr_1}, \mathtt{pr_2}, \ldots \mathtt{pr_n} \\ & \\ \Delta^\vee \equiv \perp \text{ (costante falso)} & \text{se } \Delta \text{ è la lista vuota} \\ \Delta^\vee \equiv (\,\mathtt{pr_1}\vee\mathtt{pr_2}\,)\ldots\vee\mathtt{pr_n} & \text{se } \Delta \equiv \mathtt{pr_1}, \mathtt{pr_2}, \ldots \mathtt{pr_n} \end{array}}$$

Inoltre si dice che

Lemma 5.4 Dati Σ e Δ sono liste di formule in $Form(\Gamma)$ con Γ contesto in teoria dei tipi, *l'interpretazione CHM di un sequente di formule predicative con uguaglianza* $\Sigma\ \vdash \Delta$ data dal giudizio

$$(\Delta^\vee)^I\ type\ [\Gamma,\ x \in (\Sigma^\&)^I]$$

risulta ben definita nel senso che tal giudizio è derivabile nella teoria dei tipi definita finora

Dim. Per esercizio.

Def. 5.5 *Dati* Σ *e* Δ *sono liste di formule in* $Form(\Gamma)$ *con* Γ *contesto in teoria dei tipi, si dice che il sequente* $\Sigma\ \vdash \Delta$ è valido in teoria dei tipi secondo la traduzione CHM *se e solo se esiste un* **proof-term** *(ovvero un* **termine-prova***)* **pf** *tale che*

$$\mathbf{p}r \in (\Delta^\vee)^I\ [\Gamma,\ x \in (\Sigma^\&)^I]$$

è derivabile in teoria dei tipi.

Enunciamo la seguente proprietà che segue dai corollari 3.12 e 3.14 e dalla proposizione 3.13:

Proposition 5.6 Date le proposizioni $\phi\ prop\ [\Gamma]$, $\alpha\ prop\ [\Gamma]$, $\beta\ prop\ [\Gamma]$, allora valgono le seguenti proprietà:

1. $\phi\ true\ [\Gamma, \alpha\ true, \beta\ true]$ è derivabile se e solo se $\phi\ true\ [\Gamma, \alpha\ \&\ \beta\ true]$ è derivabile;
2. $\phi\ true\ [\Gamma, \alpha\ true]$ è derivabile se e solo se $\phi\ \alpha\ \rightarrow\ \phi\ true\ [\Gamma]$ è derivabile;

3. $\phi(x)$ $true[\Gamma, x \in A]$ è derivabile se e solo se $\forall_{x \in A}$ $true$ $[\Gamma]$ è derivabile.

Dim. Per esercizio.

Lemma 5.7 Dati Σ e Δ sono liste di formule in $Form(\Gamma)$ con Γ contesto in teoria dei tipi, l'interpretazione CHM di un sequente di formule $\Sigma \ \vdash \Delta$ è *valida in teoria dei tipi* se e solo se esiste un **proof-term** (ovvero un **termine-prova**) **pf** tale che

$$\mathbf{p}r \in (\Sigma^{\&}) \ \rightarrow \ (\Delta^{\vee}) \ [\Gamma]$$

è derivabile in teoria dei tipi.

Dim. per esercizio usando il tipo delle funzioni.

Risulta poi che la logica valida secondo tale interpretazione **NON è la logica classica** bensì una logica che include quella intuizionista (per una descrizione delle regole della *logica intuizionista* si veda l'appendice e per approfondimenti si legga il capitolo 8 delle note all'indirizzo `http://www.math.unipd.it/~maietti/lomat20.html`) e *non rende valido il principio del terzo escluso* ovvero **non esiste un proof-term** di

$$\phi \ \vee \ \neg\phi \ true$$

per una qualsiasi proposizione ϕ senza variabili libere come pure non esiste un proof-term

$$\neg\neg\phi^{I} \ \rightarrow \ \phi^{I}$$

.

Theorem 5.8 I sequenti derivabili nella deduzione naturale $\mathbf{DNI}_{=}$ in appendice 14 sono tutti validi nella teoria dei tipi finora descritta.

Dim. per esercizio.

Theorem 5.9 Nella teoria dei tipi finora descritta valgono i seguenti principi di induzione:

1. per $\phi(z)$ $prop$ $[\Gamma, z \in \mathsf{N}_1]$ derivabile

$$\phi(*) \ \rightarrow \ \forall_{z \in \mathsf{N}_1} \ \phi(z) \ true \ [\Gamma]$$

   è derivabile;

2. per $\phi(z)$ *prop* $[\Gamma, z \in \mathsf{Nat}]$ derivabile
$$\phi(0)\ \& \forall_{x\in\mathsf{Nat}}\ (\phi(x)\ \rightarrow\ \phi(\mathsf{succ}(x))\ )\ \rightarrow\ \forall_{z\in\mathsf{Nat}}\ \phi(z)\ true\ [\Gamma]$$
è derivabile;

3. per $\phi(z)$ *prop* $[\Gamma, z \in \mathsf{List}(A)]$ derivabile
$$\phi(\mathsf{nil})\ \& \forall_{x\in\mathsf{List}(A)}\ \forall_{w\in A}\ (\phi(x)\ \rightarrow\ \phi(\mathsf{cons}(x,w))\ )\ \rightarrow\ \forall_{z\in\mathsf{List}(A)}\ \phi(z)\ true\ [\Gamma]$$
è derivabile;

4. per $\phi(z)$ *prop* $[\Gamma, z \in B + C]$ derivabile
$$\forall_{x_1\in B}\ \phi(\mathsf{inl}(x_1))\ \&\ \forall_{x_2\in C}\ \phi(\mathsf{inr}(x_2))\ \rightarrow\ \forall_{z\in B+C}\ \phi(z)\ true\ [\Gamma]$$
è derivabile;

5. per $\phi(z)$ *prop* $[\Gamma, z \in \Sigma_{x\in B}C(x)]$ derivabile
$$\forall_{x\in B}\ \forall_{y\in C(x)}\ (\phi(\langle x,y\rangle)\ \rightarrow\ \forall_{z\in\Sigma_{x\in B}C(x)}\ \phi(z)\ true\ [\Gamma]$$
è derivabile;

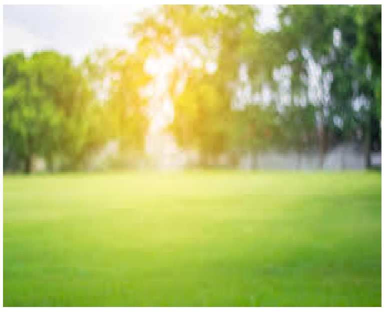
6. per $\phi(z_1,z_2,z_3)$ *prop* $[\Gamma, z_1 \in A, z_2 \in A, z_3 \in \mathsf{Id}(A,z_1,z_2)]$ derivabile
$$\forall_{x\in A}\ \phi(x,x,\mathsf{id}(x))\ \rightarrow\ \forall_{z_1\in A}\ \forall_{z_2\in A}\ \forall_{z_3\in\mathsf{Id}(A,z_1,z_2)}\ \phi(z_1,z_2,z_3)\ true\ [\Gamma]$$
è derivabile.

Dim. per esercizio.

Grazie al giudizio di verità di un tipo pensato come formula possiamo definire la nozione di **equivalenza tra tipi**:

Def. 5.10 (equivalenza di tipi) Un tipo A *type* $[\Gamma]$ si dice *equivalente* ad un tipo B *type* $[\Gamma]$ nella teoria dei tipi $\mathcal{T}$ se in $\mathcal{T}$ si possono derivare due termini $\mathbf{pf_1} \in B\ [\Gamma, x \in A]$ e $\mathbf{pf_2} \in A\ [\Gamma, y \in B]$ ovvero sono derivabili i seguenti giudizi
$$(A \to B)\ \&\ (B \to A)\ true\ [\Gamma]$$
o equivalentemente
$$A \leftrightarrow B\ true\ [\Gamma]$$
Questa nozione può essere rafforzata a quella di isomorfismo:

Def. 5.11 (isomorfismo) Un tipo A *type* $[\Gamma]$ si dice isomorfo ad un tipo B *type* $[\Gamma]$ nella teoria dei tipi $\mathcal{T}$ se in $\mathcal{T}$ si possono derivare due termini $f(x) \in B\ [\Gamma, x \in A]$ e $h(y) \in A\ [\Gamma, y \in B]$ tali per cui esistono dei proof-term $\mathbf{pf_1}$ e $\mathbf{pf_1}$ per cui in $\mathcal{T}$ si derivano
$$\mathbf{pf_1} \in \mathsf{Id}(A, x, h(f(x))\,)\ [\Gamma, x \in A] \qquad\qquad \mathbf{pf_2} \in \mathsf{Id}(B, y, f(h(y))\,)\ [\Gamma, y \in B]$$



Esercizi

1. Si mostri che il seguente principio di scelta, detto comunemente **assioma di scelta**: dati i giudizi derivabili $A type\ [\Gamma]$, $B type\ [\Gamma]$ e $R(x,y)\ prop\ [\Gamma, x \in A, y \in B]$ si mostri nella teoria dei tipi di Martin-Löf la derivabilità del giudizio

$$(\text{ AC }) \qquad \forall x \in A\ \exists y \in B\ R(x,y) \quad \longrightarrow \quad \exists f \in A \to B\ \forall x \in A\ R(x, \mathsf{Ap}(f,x))\ true\ [\Gamma]$$

2. Si dimostri che il tipo $\mathsf{N}_1 \ \rightarrow\ \mathsf{N}_0$ è isomorfo a N_0. Secondo la traduzione **propositions-as-sets** di Curry-Howard-Martin-Löf che proposizione rappresenta $\mathsf{N}_1 \ \rightarrow\ \mathsf{N}_0$??

3. Il seguente giudizio

$$\begin{array}{l} \forall_{x\in A}\ \exists_{y\in B}\ R(x,y)\ \rightarrow \\ \qquad \forall_{x_1\in A}\ \forall_{x_2\in A}\ \ \exists_{y_1\in B}\ \exists_{y_2\in B}\ (\ \ (\ R(x_1,y_1)\ \&\ R(x_2,y_2)\ )\ \&\ (x_1 =_A x_2\ \ \rightarrow\ \ y_1 =_B y_2\ )\ )\ true\ [\Gamma] \end{array}$$

   è derivabile nella teoria dei tipi con i tipi finora descritti, supposto che $A\ type\ [\Gamma]$ e $B\ type\ [\Gamma]$ e $R(x,y)\ type\ [\Gamma]$ siano derivabili in teoria dei tipi
   ?? ??

4. Supposto di rappresentare il tipo dei booleani come

$$\mathbf{Boole} \equiv \mathsf{N}_1 + \mathsf{N}_1$$

   con

$$\mathbf{0} \equiv \mathsf{inl}(*) \qquad\qquad \mathbf{1} \equiv \mathsf{inr}(*)$$

   si dimostri che esiste un proof-term $\mathbf{q}$ tale

$$\mathbf{q} \in \forall_{x\in\mathbf{Boole}}\ (\ \mathsf{Id}(\mathbf{Boole},\ x\ ,\ \mathbf{0}\ )\ \vee\ \mathsf{Id}(\mathbf{Boole},\ x\ ,\ \mathbf{1}\ )\ )$$

5. Supposto di rappresentare il tipo dei booleani come

$$\mathbf{Boole} \equiv \mathsf{N}_1 + \mathsf{N}_1$$

   con

$$\mathbf{0} \equiv \mathsf{inl}(*) \qquad\qquad \mathbf{1} \equiv \mathsf{inr}(*)$$

   e supposto che esista un proof-term

$$\mathbf{dis} \in \mathsf{N}_0\ [z \in \mathsf{Id}(\mathbf{Boole}, \mathbf{0}, \mathbf{1})]$$

   si dimostri che dati due tipi $A\ type$ e $B\ type$ esiste una funzione

$$f(z) \in \Sigma_{x\in\mathbf{Boole}}\ (\ (\mathsf{Id}(\mathbf{Boole}, x, \mathbf{0})\ \ \to A)\ \ \times\ \ (\mathsf{Id}(\mathbf{Boole}, x, \mathbf{1})\ \ \to B)\ )\ [\ z \in A + B\ ]$$

   ed anche una funzione

$$g(w) \in A + B\ type\ [w \in \Sigma_{x\in\mathbf{Boole}}\ (\ (\mathsf{Id}(\mathbf{Boole}, x, \mathbf{1})\ \ \to A)\ \ \times\ \ (\mathsf{Id}(\mathbf{Boole}, x, \mathbf{0})\ \ \to B)\ \ )]$$

6. Esistono funzioni $f(z)$ e $g(w)$ come nel punto precedente che stabiliscono una retrazione tra $A+B$ e $\Sigma_{x\in\mathbf{Boole}}\ (\ (\mathsf{Id}(\mathbf{Boole}, x, \mathbf{1})\ \ \to A)\ \ \times\ \ (\mathsf{Id}(\mathbf{Boole}, x, \mathbf{0})\ \ \to B)\ )$ ovvero si dimostra che esiste un termine prova q tale che

$$q \in \mathsf{Id}(\ A + B\ ,\ g(f(z))\ ,\ z\ )\ [z \in A + B]$$

   è derivabile nella teoria con le regole finora descritte e l'assunzione dell'esistenza di **dis**.

7. Quali regole di inferenze del calcolo dei sequenti della logica classica predicativa nelle dispense in appendice sono valide in teoria dei tipi, nel senso che conservano la validità dei sequenti coinvolti nella regola?

8. Si noti che in teoria dei tipi vale il seguente principio chiamato *existence property under a context*: Se $\exists y\ \phi(y)\ true\ [\Gamma]$ è derivabile in teoria dei tipi allora esiste un proof-term **pf** tale che

$$\phi(\mathbf{pf})\ true\ [\Gamma]$$

è derivabile in teoria dei tipi.

9. Per ogni connettivo e quantifcatore della logica predicativa con uguaglianza si descrivano le relative regole di formazione, introduzione, eliminazione e uguaglianza che risultano ammissibili in teorie dei tipi.

10. Per ogni connettivo e quantifcatore della logica predicativa con uguaglianza si descrivano le relative regole con giudizi del tipo $\phi\ prop\ [\Gamma]$ e $\phi\ true\ [\Gamma]$ di formazione, introduzione ed eliminazione che risultano ammissibili in teorie dei tipi.

11. Verificare che tutti gli assiomi di Peano sono validi in teoria dei tipi con le regole finora introdotte e interpretando le formule logiche come sopra descritto.

12. Dati i tipi $B\ type\ [\Gamma]$ e $C\ type\ [\Gamma]$ dedurre dal precedente esercizio un termine

$$\mathbf{pf_1} \in \mathsf{Id}(\mathbf{B}, \mathbf{x}, \mathbf{x'})\ [\mathbf{x} \in \mathbf{B}, \mathbf{x'} \in \mathbf{B}, \mathbf{z} \in \mathsf{Id}(\mathbf{B} + \mathbf{C}, \mathsf{inl}(\mathbf{x}), \mathsf{inl}(\mathbf{x'}))]$$

13. Dati i tipi $B\ type\ [\Gamma]$ e $C\ type\ [\Gamma]$ costruire un termine

$$\mathbf{pf_2} \in \mathsf{Id}(\mathbf{C}, \mathbf{y}, \mathbf{y'})\ [\mathbf{y} \in \mathbf{C}, \mathbf{y'} \in \mathbf{C}, \mathbf{z} \in \mathsf{Id}(\mathbf{B} + \mathbf{C}, \mathsf{inr}(\mathbf{y}), \mathsf{inr}(\mathbf{y'}))]$$

14. Mostrare che per ogni tipo $B + C$ esiste una proof-term del tipo

$$\begin{array}{l} \mathsf{Id}(\ B + C\ ,\ z\ ,\ z'\ )\ \ \leftrightarrow \\ \quad \exists_{x \in B}\, \exists_{x' \in B}\ (\ (\ \mathsf{Id}(B + C, z, \mathsf{inl}(x))\ \&\ \mathsf{Id}(B + C, z', \mathsf{inl}(x'))\ )\ \&\ \mathsf{Id}(B,\ x\ , x'\ ) \\ \qquad \vee \\ \quad \exists_{y \in C}\, \exists_{y' \in C}\ (\ (\ \mathsf{Id}(B + C, z, \mathsf{inr}(y))\ \&\ \mathsf{Id}(B + C, z', \mathsf{inr}(y'))\ \ )\ \&\ \mathsf{Id}(C,\ y\ , y'\ )\ )\ type \\ \qquad\qquad\qquad\qquad\qquad\qquad [z \in B + C, z' \in B + C] \end{array}$$

supposto di avere derivato un termine

$$\mathbf{disj} \in \mathsf{N_0}\ [x \in B, y \in C, z \in \mathsf{Id}(\mathbf{B} + \mathbf{C}, \mathsf{inl}(x), \mathsf{inr}(y))]$$

Osservazione sulla traduzione CHM della logica in teoria dei tipi: Si osservi che la traduzione CHM sopra della seguente formula logica predicativa intuizionista con variabili tipate

$$\forall_{x \in A}\, \exists_{y \in B}\, R(x, y)\ \rightarrow\ \forall_{x_1 \in A}\, \forall_{x_2 \in A}\ \exists_{y_1 \in B}\, \exists_{y_2 \in B}\ (\ (\ R(x_1, y_1)\ \&\ R(x_2, y_2)\ )\ \&\ (\ x_1 =_A x_2\ \rightarrow\ y_1 =_B y_2\ )\ )$$

è valida in teoria dei tipi mentre si dimostra che non lo è in logica intuizionista e quindi *la traduzione CHM delle formule della logica predicativa in teoria dei tipi rende vere più formule di quelle valide nella logica intuizionista* (si ricordi che la traduzione CHM invece NON rende vere tutte le formule della logica predicativa classica in quanto non vale il principio del terzo escluso o della doppia negazione). Ovviamente la traduzione data ha senso in quanto non rende vere tutte le formule poichè il falso

$$\perp^I \quad [\,]$$

essendo interpretato nel tipo vuoto NON è valido.

6 Esempi di universi predicativi: il tipo del primo universo

In questa sezione mostriamo le regole del tipo del **primo universo**, o **universo dei tipi piccoli**, nella teoria dei tipi di Martin-Löf. Tale universo è indicato con il simbolo U_0 ed è un *contenitore di tipi cosiddetti "piccoli" ovvero dei tipi costruiti nelle precedenti sezioni.* Tale universo è **predicativo** perchè lui stesso NON è piccolo ma *grande* e *non può essere contenuto in se stesso* a meno di finire in una teoria contradditoria (vedi sezione12)!!.
Si informa il lettore che nella teoria dei tipi di Martin-Löf esistono infiniti universi, uno per ogni numero naturale i indicato con il simbolo

$$\mathsf{U}_i$$

Tali universi contengono gli universi di indice inferiore e sono chiusi su tutti i costruttori di tipo presentati nelle note e per le loro regole si rimanda alla lettura del libro "*Programming in Martin-Loef's Type Theory*" B. Nordström, k. Petersson, J.M. Smith.
Si è scelto di presentare solo le regole del primo universo per la sua maggiore utilità e indispensabilità per la formalizzazione della matematica utile alle nostre applicazioni.
Si osservi che esistono due versioni di tali universi e in particolare del primo universo U_0: una versione detta *à la Tarski* in cui l'universo contiene *codici* dei tipi piccoli; una versione *à la Russell* in cui l'universo contiene i tipi piccoli direttamente senza codice.

6.1 Il tipo del primo universo à la Tarski

Di seguito descriviamo le regole di un tipo **predicativo** "universo predicativo à la Tarski" che contiene i codici di tutti i tipi precedentemente descritti. Chiaramente l'universo non contiene un codice di se stesso in quanto se così fosse la teoria diventerebbe inconsistente. Come conseguenza la presenza di un universo predicativo introduce una differenza tra tipi, ovvero tra quelli codificati nell'universo, che chiameremo *tipi piccoli*, e i tipi non codificati all'interno che possiamo chiamare *tipi grandi*.

Universe of small types

$$\text{F-Un}_0)\ \frac{\Gamma\ cont}{\mathsf{U}_0\ type\ [\Gamma]}$$

$$\text{I}_1\text{-Un}_0)\ \frac{\Gamma\ cont}{\hat{\mathbf{N}}_0 \in \mathsf{U}_0\ [\Gamma]} \qquad \text{I}_2\text{-Un}_0)\ \frac{\Gamma\ cont}{\hat{\mathbf{N}}_1 \in \mathsf{U}_0\ [\Gamma]} \qquad \text{I}_3\text{-Un}_0)\ \frac{\Gamma\ cont}{\hat{\mathsf{Nat}} \in \mathsf{U}_0\ [\Gamma]}$$

$$\text{I}_4\text{-Un}_0)\ \frac{c(x)\ \in \mathsf{U}_0\ \ [\Gamma, x \in \mathbf{T}(b)] \qquad b \in \mathsf{U}_0\ [\Gamma]}{\hat{\mathbf{\Sigma}}_{x\in b} c(x)\ \in \mathsf{U}_0\ [\Gamma]}$$

$$\text{I}_5\text{-Un}_0)\ \frac{c(x)\ \in \mathsf{U}_0\ \ [\Gamma, x \in \mathbf{T}(b)] \qquad b \in \mathsf{U}_0}{\hat{\mathbf{\Pi}}_{x\in b} c(x)\ \in \mathsf{U}_0\ [\Gamma]} \qquad \text{I}_6\text{-Un}_0)\ \frac{b \in \mathsf{U}_0\ [\Gamma] \qquad c \in \mathsf{U}_0\ [\Gamma]}{b\hat{+}c\ \in \mathsf{U}_0\ [\Gamma]}$$

$$\text{I}_7\text{-Un}_0)\ \frac{b \in \mathsf{U}_0\ [\Gamma]}{\hat{\mathbf{List}}(b)\ \in \mathsf{U}_0\ [\Gamma]} \qquad \text{I}_8\text{-Un}_0)\ \frac{b \in \mathsf{U}_0\ [\Gamma] \qquad c \in T(b)\ [\Gamma] \qquad d \in T(b)\ [\Gamma]}{\hat{\mathsf{Id}}(b,c,d)\ \in \mathsf{U}_0\ [\Gamma]}$$

$$\text{E-Un}_0)\ \frac{a\ \in \mathsf{U}_0\ [\Gamma]}{\mathbf{T}(a)\ type\ [\Gamma]}$$

$$\text{C}_1\text{-Un}_0)\ \frac{\Gamma\ cont}{\mathbf{T}(\hat{\mathbf{N}}_0) = N_0\ type\ \ [\Gamma]} \qquad \text{C}_2\text{-Un}_0)\ \frac{\Gamma\ cont}{\mathbf{T}(\hat{\mathbf{N}}_1) = N_1\ type\ \ [\Gamma]}$$

$$\text{C}_3\text{-Un}_0)\ \frac{\Gamma\ cont}{\mathbf{T}(\hat{\mathsf{Nat}}) = \mathsf{Nat}\ type\ [\Gamma]} \qquad \text{C}_4\text{-Un}_0)\ \frac{c(x)\ \in \mathsf{U}_0\ \ [\Gamma, x \in \mathbf{T}(b)] \qquad b \in \mathsf{U}_0\ [\Gamma]}{\mathbf{T}(\hat{\mathbf{\Sigma}}_{x\in b} c(x)) = (\Sigma x \in \mathbf{T}(b))\mathbf{T}(c(x))\ type}$$

$$\text{C}_5\text{-Un}_0)\ \frac{c(x)\ \in \mathsf{U}_0\ \ [\Gamma, x \in \mathbf{T}(b)] \qquad b \in \mathsf{U}_0\ [\Gamma]}{\mathbf{T}(\hat{\mathbf{\Pi}}_{x\in b} c(x)) = (\Pi x \in \mathbf{T}(b))\mathbf{T}(c(x))\ type\ [\Gamma]} \qquad \text{C}_6\text{-Un}_0)\ \frac{b \in \mathsf{U}_0\ [\Gamma] \qquad c \in \mathsf{U}_0\ [\Gamma]}{\mathbf{T}(b\hat{+}c) = \mathbf{T}(b) + \mathbf{T}(c)\ type\ [\Gamma]}$$

$$\text{C}_7\text{-Un}_0)\ \frac{b \in \mathsf{U}_0\ [\Gamma]}{\mathbf{T}(\hat{\mathbf{List}}(b)) = List(\mathbf{T}(b))\ type\ [\Gamma]} \qquad \text{C}_8\text{-Un}_0)\ \frac{b \in \mathsf{U}_0\ [\Gamma] \qquad c \in T(b)\ [\Gamma] \qquad d \in T(b)\ [\Gamma]}{\mathbf{T}(\hat{\mathsf{Id}}(b,c,d)) = \mathsf{Id}(\mathbf{T}(b),c,d)\ type\ [\Gamma]}$$

$$\text{eq-I}_4\text{-Un}_0)\ \frac{c_1(x) = c_2(x)\ \in \mathsf{U}_0\ \ [\Gamma, x \in \mathbf{T}(b_1)] \qquad b_1 = b_2 \in \mathsf{U}_0\ [\Gamma]}{\hat{\mathbf{\Sigma}}_{x\in b_1} c_1(x) = \hat{\mathbf{\Sigma}}_{x\in b_2} c_2(x)\ \in \mathsf{U}_0\ [\Gamma]}$$

$$\text{I}_5\text{-Un}_0)\ \frac{c_1(x) = c_2(x)\ \in \mathsf{U}_0\ \ [\Gamma, x \in \mathbf{T}(b_1)] \qquad b_1 = b_2 \in \mathsf{U}_0}{\hat{\mathbf{\Pi}}_{x\in b_1} c_1(x) = \hat{\mathbf{\Pi}}_{x\in b_2} c_2(x)\ \in \mathsf{U}_0\ [\Gamma]} \qquad \text{I}_6\text{-Un}_0)\ \frac{b_1 = b_2 \in \mathsf{U}_0\ [\Gamma] \qquad c_1 = c_2 \in \mathsf{U}_0\ [\Gamma]}{b_1\hat{+}c_1 = \in \mathsf{U}_0\ [\Gamma]}$$

$$\text{I}_7\text{-Un}_0)\ \frac{b \in \mathsf{U}_0\ [\Gamma]}{\hat{\mathbf{List}}(b)\ \in \mathsf{U}_0\ [\Gamma]} \qquad \text{I}_8\text{-Un}_0)\ \frac{a_1 = a_2 \in \mathsf{U}_0\ [\Gamma] \qquad c_1 = c_2 \in T({}_1)\ [\Gamma] \qquad d_1 = d_2 \in T(b)\ [\Gamma]}{\hat{\mathsf{Id}}(a_1,c_1,d_1) = \hat{\mathsf{Id}}(a_2,c_2,d_2)\ \in \mathsf{U}_0\ [\Gamma]}$$

$$\text{E-Un}_0)\ \frac{a_1 = a_2\ \in \mathsf{U}_0\ [\Gamma]}{\mathbf{T}(a_1) = T(a_2)\ type\ [\Gamma]}$$

Esercizi:

1. Dare la definizione di eliminazione induttiva dell'universo U_0.
2. Dimostrare che nella teoria dei tipi con tutti i costrutti introdotti *senza il tipo universo* U_0 si possono interpretare i tipi o nell'insieme vuoto N_0 *type* $[\Gamma]$ o nell'insieme singoletto N_1 *type* $[\Gamma]$ e i termini come elementi dell'interpretazione dei loro tipi in modo tale che

   un tipo dipendente $\mathbf{B(x)}$ **type** $[\mathbf{x} \in \mathbf{A}]$ sia interpretato come come una famiglia di insiemi

   $$B(x)^J(z)\ type\ [z \in A^I]$$

   e valgano

$(\mathsf{N}_0\ [\Gamma])^J$	$=$	$\mathsf{N}_0\ type\ [\Gamma^J]$
$(\mathsf{N}_1\ [\Gamma])^J$	$=$	$\mathsf{N}_1\ type\ [\Gamma^J]$
$(List(A)\ [\Gamma])^J$	$=$	$\mathsf{N}_1\ type\ [\Gamma^J]$
$(A\ +\ B\,[\Gamma])^J$	$=$	$\begin{cases} \mathsf{N}_0\ type\ [\Gamma^J] & \text{se } A^J = B^J = \mathsf{N}_0\ type\ [\Gamma^J] \\ \mathsf{N}_1\ type\ [\Gamma^J] & \text{altrimenti} \end{cases}$
$(\mathsf{Id}(A,t,s)\,[\Gamma])^J$	$=$	$A^J\ type\ [\Gamma^J]$
$(\Pi_{x\in A}\ B(x))^I\,[\Gamma]$	$=$	$\begin{cases} \mathsf{N}_0\ type\ [\Gamma^J] & \text{se } A^J = \mathsf{N}_1\ type\ [\Gamma^J] \text{ e } B^J = \mathsf{N}_0\ type\ [\Gamma^J] \\ \mathsf{N}_1\ type\ [\Gamma^J] & \text{altrimenti} \end{cases}$
$(\Sigma_{x\in A}\ B(x)\,[\Gamma])^J$	$=$	$\begin{cases} \mathsf{N}_1\ type\ [\Gamma^J] & \text{se } A^J = B^J = \mathsf{N}_1\ type\ [\Gamma^j] \\ \mathsf{N}_0\ type\ [\Gamma^J] & \text{altrimenti} \end{cases}$

   Dimostrare poi che questa interpretazione *rende vero*

   $$\mathsf{Id}(Nat, 0, 1)\ true\ [\ ]$$

   e quindi *rende falso*

   $$\neg\mathsf{Id}(Nat, 0, 1)\ true\ [\ ]$$

   ovvero l'interpretazione costruita è un *modello in teoria dei tipi della proposizione* $0 =_{\mathsf{Nat}} 1$ ed è un *contromodello in teoria dei tipi della proposizione* $\neg 0 =_{\mathsf{Nat}} 1$.
3. Dimostrare che nella teoria dei tipi con le regole finora descritte si deriva un termine q

   $$q \in (\ T(x)\ \to\ T(y)\ )\ \times\ (\ T(y)\ \to\ T(x)\ )\ [x \in \mathsf{U}_0, y \in \mathsf{U}_0, z \in \mathsf{Id}(\mathsf{U}_0,\ x, y\,)]$$

4. Dimostrare in teoria dei tipi *con un universo* con almeno il codice del tipo vuoto e del tipo singoletto, ad esempio con U_0, si può derivare

   $$\neg\mathsf{Id}(Nat, 0, 1)\ true\ [\ ]$$

5. Identificando il tipo dei booleani con $\mathsf{N_1} + \mathsf{N_1}$, dati $a \in U_0$ e $b \in U_o$ allora il tipo $\mathbf{T}(\mathbf{a}) \ \times \ \mathbf{T}(\mathbf{b})$ si può rappresentare utilizzando opportunamente il tipo prodotto dipendente Π, il tipo dei booleani e l'universo $\mathsf{U_0}$ senza utilizzare il tipo somma indiciata forte.

6. Identificando il tipo dei booleani con $\mathsf{N_1} + \mathsf{N_1}$, dati $a \in U_0$ e $b \in U_o$ allora il tipo $\mathbf{T}(\mathbf{a}) \ + \ \mathbf{T}(\mathbf{b})$ si può rappresentare utilizzando opportunamente il tipo somma indiciata forte Σ, il tipo dei booleani e l'universo $\mathsf{U_0}$ senza utilizzare il tipo somma disgiunta binaria.

7. Mostrare che in presenza dell'universo $\mathsf{U_0}$ si può dimostrare che dati due termini di tipi qualsiasi, ovvero giudizi derivabili $b \in B\ [\Gamma]$ e $c \in C\ [\Gamma]$, allora la somma disgiunta $\mathbf{B}+\mathbf{C}$ è proprio *disgiunta* nel senso che esiste un termine

$$\mathbf{dis} \in \mathsf{N_0}\ [z \in \mathsf{Id}(\mathbf{B}+\mathbf{C}, \mathsf{inl}(b), \mathsf{inr}(c))]$$

6.2 Il tipo del primo universo à la Russell

Diamo qui le regole per la costruzione dell'universo i cui elementi sono tipi precedentemente definiti e non codici che li rappresentano come nella versione alla Tarski e la grande differenza è che non c'è bisogno di aggiungere regole di uguaglianza computazionale come invece accade nella versione à la Tarski.. Tale versione di universo di dice *à la Russell.*

Universe of small types à la Russell

F-rUn$_0$) $\dfrac{\Gamma\ cont}{\mathsf{U}_0\ type\ [\Gamma]}$ I$_1$-rUn$_0$) $\dfrac{\Gamma\ cont}{\mathsf{N}_0 \in \mathsf{U}_0\ [\Gamma]}$

I$_2$-rUn$_0$) $\dfrac{\Gamma\ cont}{\mathsf{N}_1 \in \mathsf{U}_0}$ I$_3$-rUn$_0$) $\dfrac{\Gamma\ cont}{\mathsf{Nat} \in \mathsf{U}_0}$

I$_4$-rUn$_0$) $\dfrac{C(x)\ \in \mathsf{U}_0\ \ [x \in B] \qquad B \in \mathsf{U}_0}{\Sigma_{x\in B} C(x)\ \in \mathsf{U}_0}$

I$_5$-rUn$_0$) $\dfrac{C(x)\ \in \mathsf{U}_0\ \ [x \in B] \qquad B \in \mathsf{U}_0}{\Pi_{x\in B} C(x)\ \in \mathsf{U}_0}$ I$_6$-rUn$_0$) $\dfrac{B \in \mathsf{U}_0 \qquad C \in \mathsf{U}_0}{B + C\ \in \mathsf{U}_0}$

I$_7$-rUn$_0$) $\dfrac{B \in \mathsf{U}_0}{\hat{\mathbf{List}}(B)\ \in \mathsf{U}_0}$ I$_8$-rUn$_0$) $\dfrac{B \in \mathsf{U}_0 \qquad c \in B \qquad d \in B}{\mathsf{Id}(B, c, d)\ \in \mathsf{U}_0}$

E-rUn$_0$) $\dfrac{A\ \in \mathsf{U}_0\ [\Gamma]}{A\ type\ [\Gamma]}$

Esercizi:

1. È possibile aggiungere un'infinità di universi ove ogni universo successivo contiene un codice dell'universo precedente ed è chiuso sugli stessi costruttori di tipo del primo universo U_0. Provare a formularne le regole.

2. Sia $\mathcal{T}_t$ la teoria dei tipi con tutte i tipi finora descritti ma con il solo universo dei tipi piccoli alla Tarski. Sia $\mathcal{T}_r$ la teoria dei tipi con tutte i tipi finora descritti ma con il solo universo dei tipi piccoli alla Russell. Dimostrare che le teorie $\mathcal{T}_t$ e $\mathcal{T}_r$ sono equivalenti.

7 L'universo impredicativo delle proposizioni

In taluni teorie dei tipi, come il Calcolo delle Costruzioni di T. Coquand and G. Huet, le proposizioni *NON sono identificate con i tipi* ma sono definite primitivamente. Tale calcolo esteso con i tipi induttivi si chiama **CIC** per *Calculus of Inductive Constructions.* In **CIC** in particolare le proposizione sono definite a partire da un **universo delle proposizioni** pensato come un tipo definito impredicativamente e **chiuso** soltanto **sulla quantificazione universale**. Tale universo è impredicativo in quanto si può dimostrare che NON gode di un principio di induzione e quindi NON può definito predicativamente se con tipo predicativo intendiamo un tipo che può essere induttivamente generato.
Inoltre è possibile dimostrare che tale universo è chiuso su tutti gli usuali connettivi proposizionali e sulla quantificazione esistenziale.

Universe of primitive propositions

F-prop) $\dfrac{\Gamma\ cont}{\mathsf{Prop}\ type\ [\Gamma]}$ $\qquad$ I-prop) $\dfrac{c(x)\ \in \mathsf{Prop}\ \ [\Gamma, x \in A]}{\forall_{x\in A} c(x)\ \in \mathsf{Prop}\ [\Gamma]}$

E-prop) $\dfrac{a\ \in \mathsf{Prop}\ [\Gamma]}{\mathbf{T}(a)\ type\ [\Gamma]}$

eq-prop) $\dfrac{c(x)\ \in \mathsf{Prop}\ \ [\Gamma, x \in A]}{\mathbf{T}(\forall_{x\in A} c(x)) = \Pi x \in A\ \mathbf{T}(c(x))\ type\ [\Gamma]}$

Def. 7.1 (implicazione) *Date due proposizioni* $a \in \mathsf{Prop}\ [\Gamma]$ *e* $b \in \mathsf{Prop}\ [\Gamma]$ *definiamo il connettivo* implicazione *in tal modo*

$$a \to b\ \equiv\ \forall_{x\in T(a)}\ b$$

Def. 7.2 *Date due proposizioni* $a \in \mathsf{Prop}\ [\Gamma]$ *e* $b \in \mathsf{Prop}\ [\Gamma]$ *il connettivo* congiunzione *è definito in tal modo*

$$a \,\&\, b\ \equiv\ \forall_{p\in \mathsf{Prop}}\ \ (\ \forall_{x\in T(a)}\ \forall_{x\in T(b)}\ p\ ) \to p$$

Def. 7.3 *Date due proposizioni* $a \in \mathsf{Prop}\ [\Gamma]$ *e* $b \in \mathsf{Prop}\ [\Gamma]$ *il connettivo* disgiunzione *è definito in tal modo*

$$a \vee b\ \equiv\ \forall_{p\in \mathsf{Prop}}\ \ (\ (\ a\ \to p\ )\ \&\ (\ b\ \to p\ )\ \ \to\ p\ )$$

Def. 7.4 *Date due proposizioni* $a \in \mathsf{Prop}\ [\Gamma]$ *e* $b \in \mathsf{Prop}\ [\Gamma]$ *la* quantificazione esistenziale *è definito in tal modo*

$$\exists_{x\in A}\ c(x)\ \equiv\ \forall_{p\in \mathsf{Prop}}\ \ (\ (\forall_{x\in A}\ c(x)\ \to p\ )\ \ \to\ p\ )$$

Def. 7.5 *Dato un tipo* $A\ [\Gamma]$ *e due termini* $a \in A\ [\Gamma]$ *e* $b \in A\ [\Gamma]$ *l'uguaglianza proposizionale è definita in tal modo*

$$\mathsf{Id}_{CIC}(A, a, b)\ \equiv\ \forall_{p\in A\to\mathsf{Prop}}\ (\ p(a)\ \ \to\ \ p(b)\ )$$

Esercizi

1. Si dimostri che in tal teoria si deriva che un termine

   $$q \in T(\alpha\ \&\ \beta)\ \ \leftrightarrow\ \ T(\alpha) \times T(\beta)\ [\alpha \in \mathsf{Prop}, \beta \in \mathsf{Prop}]$$

   ove $A\ \leftrightarrow\ B\ \equiv\ (A \to B)\ \times\ (B \to A)$.

2. Si dimostri che i connettivi proposizionali definiti sopra sono effettivamente delle proposizioni tipate in Prop e mostrare che rendono valide le regole dei sequenti della logica intuizionista proposizionale.

3. Si dimostri che le quantificazioni universali ed esistenziale definite sopra sono effettivamente delle proposizioni tipate in Prop e mostrare che rendono valide le regole dei sequenti della logica intuizionista predicativa.

4. Che tipo di uguaglianza è definita con l'universo delle proposizioni nella teoria sopra?

5. In che relazione stanno le costruzioni delle proposizioni sopra viste come tipi rispetto alle proposizioni definite secondo l'isomorfismo "proposizioni come tipi" nella teoria di Martin-Löf?

8 Versione ESTENSIONALE della teoria dei tipi

Esiste un'altro tipo uguaglianza proposizionale detto **estensionale** in quanto rende l'uguaglianza computazionale $a = b \in A\ [\Gamma]$ *indecidibile.*

Extensional Propositional Equality

$$\text{F-Eq)}\ \frac{C\ \ type\ [\Gamma] \qquad c \in C\ [\Gamma] \qquad d \in C\ [\Gamma]}{\mathsf{Eq}(C, c, d)\ type\ \ [\Gamma]}$$

$$\text{I-Eq)}\ \frac{c \in C\ [\Gamma]}{\mathsf{eq_C} \in \mathsf{Eq}(C, c, c)\ [\Gamma]} \qquad\qquad \text{E-Eq)}\ \frac{p \in \mathsf{Eq}(C, c, d)\ [\Gamma]}{c = d \in C\ [\Gamma]}$$

$$\text{C-Eq)}\ \frac{p \in \mathsf{Eq}(C, c, d)\ [\Gamma]}{p = \mathsf{eq_C} \in \mathsf{Eq}(C, c, d)\ [\Gamma]}$$

In particolare in presenza di un universo e dell'ugualgianza proposizionale estensionale i termini tipati della teoria dei tipi NON sono normalizzabili:

Proposition 8.1 La teoria dei tipi con un universo e uguaglianza proposizionale estensionale NON è normalizzante.

Dim. Si osservi che il seguente giudizio

$$\lambda x \in T(w).x(x) \in T(w)\ [w \in \mathsf{U_0}, y \in \mathsf{Eq}(\mathsf{U_0},\ \widehat{\Pi}_{x\in T(w)}\ w\ ,\ w\ )]$$

è derivabile in teoria dei tipi e NON è fortemente normalizzante rispetto alla riduzione della β-riduzione del lambda calcolo puro.

Esercizio.

1. Si provi che in presenza dell'uguaglianza proposizionale estensionale vale l'estensionalità delle funzioni ovvero, dati due termini derivabili $f(x) \in B(x)\ [x \in A]$ e $g(x) \in B(x)\ [x \in A]$ si può derivare

$$\mathsf{Eq}(\ \Pi_{x\in A}B(x),\ \lambda x \in A.f(x)\ ,\ \lambda x \in A.g(x)\ )\ [w \in \Pi_{x\in A}\ \mathsf{Eq}(\ B(x),\ f(x), g(x)\ )]$$

2. Si provi che in presenza dell'uguaglianza proposizionale estensionale vale l'estensionalità delle funzioni ovvero, dato un termine derivabile $f \in \pi_{x\in A}\ B(x)\ [\Gamma]$ si può derivare

$$\lambda x.f(x) = f \in \pi_{x\in A}\ B(x)\ [\Gamma]$$

3. Dimostrare che il tipo somma indiciata è isomorfo al seguente tipo di somma indiciata indebolita una volta aggiunto alla teoria dei tipi

Howard's Weak Indexed Sum type

$$\text{w}\Sigma)\ \frac{C(x)\ type\ \ [\Gamma, x \in B]}{\Sigma^w_{x\in B}C(x)\ type\ [\Gamma]} \qquad\qquad \text{I-w}\Sigma)\ \frac{b \in B\ [\Gamma] \qquad c \in C(b) \qquad \Sigma^w_{x\in B}C(x)\ type\ [\Gamma]}{< b, c >\in \Sigma^w_{x\in B}C(x)\ [\Gamma]}$$

$$\text{E-w}\Sigma)\ \frac{\begin{array}{l} M\ type\ [\Gamma] \\ d \in \Sigma^{w}_{x\in B} C(x)\ [\Gamma] \qquad\qquad m(x,y) \in M\ [\Gamma, x \in B, y \in C(x)] \end{array}}{El_{\Sigma w}(d, m) \in M\ [\Gamma]}$$

$$\text{C-w}\Sigma)\ \frac{\begin{array}{l} M\ type\ [\Gamma] \\ b \in B \quad c \in C(b) \qquad m(x,y) \in M\ [\Gamma, x \in B, y \in C(x)] \end{array}}{El_{\Sigma w}(< b, c >, m) = m(b, c) \in M\ [\Gamma]}$$

4. In presenza dell'uguaglianza proposizionale estensionale mostrare che il tipo delle liste è isomorfo ad un tipo lista detto *non dipendente* le cui regole da aggiungere alla teoria dei tipi sono quelle del tipo lista già introdotto per quanto riguarda le regole di introduzione mentre la loro regola di eliminazione è ristretta a valere per tipi non dipendenti dal tipo lista stesso e le regole di computazioni includono l'unicità dell'eliminatore oltre alle regole di computazione del tipo lista dipendente adattate al caso dell'eliminazione verso tipi non dipendenti.

Non-dependent List type

$$\text{F-nd-list)}\ \frac{C\ type\ [\Gamma]}{ndList(C)\ type\ [\Gamma]}$$

$$\text{I}_1\text{-nd-list)}\ \frac{ndList(C)\ type\ [\Gamma]}{\mathsf{nil} \in ndList(C)\ [\Gamma]} \qquad \text{I}_2\text{-nd-list)}\ \frac{s \in ndList(C)\ [\Gamma] \qquad c \in C\ [\Gamma]}{\mathsf{cons}(s, c) \in ndList(C)\ [\Gamma]}$$

$$\text{E-nd-list)}\ \frac{\begin{array}{l} L\ type\ [\Gamma] \\ s \in List(C)\ [\Gamma] \qquad a \in L\ [\Gamma] \qquad l(x,z) \in L\ [\Gamma, x \in C, z \in L] \end{array}}{\mathsf{El}_{\mathsf{ndList}}(a, l, s) \in L\ [\Gamma]}$$

$$\text{C}_1\text{-nd-list)}\ \frac{\begin{array}{l} L\ type\ [\Gamma] \\ a \in L\ [\Gamma] \qquad l(x,z) \in L\ [\Gamma, x \in C, z \in L] \end{array}}{\mathsf{El}_{\mathsf{ndList}}(\mathsf{nil}, a, l) = a \in L\ [\Gamma]}$$

$$\text{C}_2\text{-ndlist)}\ \frac{\begin{array}{l} L\ type\ [\Gamma] \\ s \in ndList(C)\ [\Gamma] \qquad c \in C\ [\Gamma] \qquad a \in L\ [\Gamma] \qquad l(x,z) \in L\ [\Gamma, x \in C, z \in L] \end{array}}{\mathsf{El}_{\mathsf{ndList}}(\mathsf{cons}(s, c), a, l) = l(c, \mathsf{El}_{\mathsf{ndList}}(s, a, l)) \in L\ [\Gamma]}$$

$$\eta_s\text{C-ndlist)}\ \frac{\begin{array}{l} L\ type\ [\Gamma] \qquad a \in L\ [\Gamma] \qquad l(x,z) \in L\ [\Gamma, x \in C, z \in L] \\ t(y) \in L\ [\Gamma, y \in ndList(C)] \qquad s \in ndList(C)\ [\Gamma] \\ t(\epsilon) = a \in L\ [\Gamma] \qquad t(\mathsf{cons}(z, x)) = l(t(z), x) \in L\ [\Gamma, x \in C, z \in ndList(C)] \end{array}}{\mathsf{El}_{\mathsf{ndList}}(a, a, l) = t(s) \in L\ [\Gamma]}$$

5. Scrivere le regole del tipo dei numeri naturali con eliminazione verso tipi non dipendenti e mostrare

che il tipo dei naturali ottenuto, indicato con $ndNat$, una volta aggiunto alla teoria dei tipi con uguaglianza proposizionale estensionale risulta isomorfo al tipo dei naturali Nat precedentemente definito.

6. Mostrare che all'interno della teoria dei tipi con uguaglianza proposizionale estensionale il tipo somma indiciata risulta isomorfo al seguente tipo somma indiciata con proiezioni

   Extensional Indexed Sum type (with projections)

$$\text{F-}\Sigma)\ \frac{C(x)\ type\quad [\Gamma, x \in B]}{\Sigma_{x\in B}C(x)\ type\ [\Gamma]} \qquad \text{I-}\Sigma)\ \frac{b \in B\ [\Gamma] \qquad c \in C(b)\ [\Gamma] \qquad \Sigma_{x\in B}C(x)\ type\ [\Gamma]}{< b, c > \in \Sigma_{x\in B}C(x)\ [\Gamma]}$$

$$\text{E}_{\mathbf{1}}\text{-}\Sigma)\ \frac{d \in \Sigma_{x\in B}C(x)\ [\Gamma]}{\pi_1^B(d) \in B\ [\Gamma]} \qquad \text{E}_{\mathbf{2}}\text{-}\Sigma)\ \frac{d \in \Sigma_{x\in B}C(x)\ [\Gamma]}{\pi_2^{C(\pi_1(d))}(d) \in C(\pi_1(d))\ [\Gamma]}$$

$$\beta_1\ \text{C-}\Sigma)\ \frac{b \in B\ [\Gamma] \qquad c \in C(b)\ [\Gamma]}{\pi_1^B(< b, c >) = b \in B\ [\Gamma]} \qquad \beta_2\ \text{C-}\Sigma)\ \frac{b \in B\ [\Gamma] \qquad c \in C(b)\ [\Gamma]}{\pi_2^{C(b)}(< b, c >) = c \in C(b)\ [\Gamma]}$$

$$\eta\ \text{C-}\Sigma)\ \frac{d \in \Sigma_{x\in B}C(x)\ [\Gamma]}{< \pi_1^B(d), \pi_2^{C(\pi_1(d))}(d) >= d \in \Sigma_{x\in B}C(x)\ [\Gamma]}$$

7. Mostrare che all'interno della teoria dei tipi con uguaglianza proposizionale estensionale le regole di eliminazione e conversione del tipo somma binaria possono essere sostituite dalle seguenti:

$$\text{E}_s\text{-+})\ \frac{\begin{array}{l} A\ type\ [\Gamma] \\ p \in C + D\ [\Gamma] \qquad a_C(x) \in A\ [\Gamma, x \in C] \qquad a_D(y) \in A\ [\Gamma, y \in D] \end{array}}{El_{+s}(p, a_C, a_D) \in A}$$

$$\text{C}_{1s}\text{-+})\ \frac{\begin{array}{l} A\ type\ [\Gamma] \\ c \in C \qquad a_C(x) \in A\ [\Gamma, x \in C] \qquad a_D(y) \in A\ [\Gamma, y \in D] \end{array}}{El_{+s}(\mathsf{inl}(c), a_C, a_D) = a_C(c) \in A\ [\Gamma]}$$

$$\text{C}_{2s}\text{-+})\ \frac{\begin{array}{l} A\ type\ [\Gamma] \\ d \in D \qquad a_C(x) \in A\ [\Gamma, x \in C] \quad a_D(y) \in A\ [\Gamma, y \in D] \end{array}}{\mathsf{El}_{+s}(\mathsf{inr}(d), a_C, a_D) = a_D(d) \in A\ [\Gamma]}$$

$$\eta\text{-+})\ \frac{p \in C + D\ [\Gamma] \qquad t(z) \in A\ [\Gamma, z \in C + D]}{\mathsf{El}_{+s}(p,\ (x)\,t(\mathsf{inl}(x))\,,\ (y)\,t(\mathsf{inr}(x))\,) = t(p) \in A\ [\Gamma]}$$

9 Teoria dei tipi di Martin-Löf intensionale MLTT (alla Russell/Tarski)

La teoria dei tipi introdotta da Per Martin-Löf è quella con i tipi singoletto, liste, unioni disgiunte, somme indiciate, prodotto dipendente, uguaglianza proposizionale $\mathsf{Id}(A, a, b)$ e universo dei tipi piccoli alla Russell/Tarski.

10 Teoria dei tipi di Martin-Löf estensionale

La teoria dei tipi di Martin-Löf in versione estensionale è quella con i tipi singoletto, liste, unioni disgiunte, somme indiciate, prodotto dipendente, uguaglianza proposizionale $\mathsf{Eq}(A, a, b)$ e universo dei tipi piccoli alla Russell.

11 Calcolo delle costruzioni di Coquand CIC

Il calcolo delle Costruzioni introdotto Thierry Coquand è la teoria dei tipi di Martin-Löf intensionale alla Russell con l'aggiunta del tipo impredicativo delle proposizioni in sezione 7 (e definizione induttive di altri possibili tipi induttivi).

12 Inconsistenza della teoria dei tipi di Martin-Löf con $\mathsf{U}_0 \in \mathsf{U}_0$

Consideriamo qui la teoria dei tipi di Martin-Löf intensionale con l'aggiunta dell'assioma

$$\mathsf{U}_0 \in \mathsf{U}_0$$

Chiamiamo tale teoria, che diventa impredicativa, *MLim.*
A tal scopo diamo una breve descrizione di come formalizzare in tale teoria una versione del paradosso di Mirimanoff formulato in teoria degli insiemi classica.
Ricordiamo che un *insieme* V *si dice* ben fondato *se* NON esiste *una catena infinita discendente di insiemi* $(x_i)_{i \in Nat}$

$$\ldots x_n \in x_{n-1} \ldots x_2 \in x_1 \in V$$

Tale paradosso afferma che *la collezione* W *degli insiemi ben fondati NON è un insieme* perchè se lo fosse si dimostra che W stesso sarebbe ben fondato in quanto altrimenti una catena infinita

$$\ldots x_n \in x_{n-1} \ldots x_2 \in x_1 \in W$$

dà luogo ad una sottocatena

$$\ldots x_n \in x_{n-1} \ldots x_2 \in x_1$$

di x_1 che per ipotesi è ben fondato essendo in W e dunque ciò non è possibile. Concludiamo quindi che W è ben fondato. Però in tal caso si avrebbe anche che $W \in W$ e quindi ci sarebbe una catena discendente infinita ove ogni $x_i \equiv W$ ovvero

$$\ldots W \in W \in W \in W$$

contro il fatto che se W è un insieme allora è un insieme ben fondato. La conclusione è dunque che *la collezione* W *degli insiemi ben fondati NON è un insieme ma solo una collezione anche se ben fondata e quindi non può appartenere a se stesso.*

Ora cerchiamo di rappresentare questo paradosso nella teoria *MLim*.
Si procede come segue.
Si definisce

$$\Omega \equiv \Sigma_{A\in \mathsf{U}_o} \Sigma_{<_A \in A\times A\to \mathsf{U}_0} \mathsf{WF}(A,<_A)$$

ove $<$ è qui usato *come variabile che rappresenta una relazione su w* che è *ben fondata* in quanto, usando l'abbreviazione $a <_A b \equiv \mathsf{Ap}(\langle a,b\rangle)$ poniamo

$$\mathsf{WF}(A,<_A) \equiv \forall_{P\in A\to \mathsf{U}_0} \quad \exists_{x\in A} \mathsf{Ap}(P,x) \quad \to \quad \neg(\forall_{x\in A}(\mathsf{Ap}(P,x) \to \exists_{y\in A} \mathsf{Ap}(P,y) \,\&\, y <_A x))$$

ove P rappresenta una funzione proposizionale su A, ovvero un suo sottoinsieme, e quindi la scrittura sopra dice che
A è ben fondato se per ogni sottoinsieme P non vuoto NON è vero che per ogni suo elemento x ne esiste un altro y tale che $y < x$.
Nel seguito scriviamo semplicemente

$$P(x) \qquad \text{al posto di} \qquad \mathsf{Ap}(P,x)$$

12.1 Ω è un tipo ben fondato

Ora mostriamo che Ω è un tipo ben fondato definendo una opportuna relazione su di esso.
A tal scopo introduciamo le seguenti abbreviazioni:

dato $W\in\Omega$ chiamiamo

$$\overline{W} \equiv \pi_1(W) \qquad\qquad <_W \equiv \pi_1(\pi_2(W))$$

Poi definiamo per $W,Y\in\Omega$

$$W <_\Omega Y \equiv \exists_{f\in \overline{W}\to\overline{Y}} \; ((\mathsf{Mon})(f) \,\&\, \exists_{m\in\overline{Y}} f(\overline{W}) <_{mag} m)$$

ove $\mathsf{Mon}(f)$ esprime che f è una funzione monotona:

$$(\mathsf{Mon})(f) \equiv \forall_{x_1\in\overline{W}} \forall_{x_2\in\overline{W}} (x_1 <_W x_2 \to f(x_1) <_Y f(x_2))$$

e l'abbreviazione

$$f(\overline{W}) <_{mag} m \equiv \forall_{x\in\overline{W}} f(x) <_Y m$$

dice m è un maggiorante di ogni elemento di $\overline{Y}$ nell'immagine di f.

Si dimostra che questa relazione è ben ordinata (farlo per bene per esercizio). L'idea è che dato un predicato $P\in\Omega\times\Omega \to \mathsf{U}_0$ tale che $z\in\exists_{Y\in\Omega} P(y)$ chiamiamo

$$B \equiv \pi_1(\pi_1(z))$$

che sappiamo essere un tipo ben fondato con relazione $<_{\overline{B}} \equiv \pi_1(\pi_2(\pi_1(z)))$.
Usiamo ora il fatto che B è ben fondato per dimostrare che in Ω

$$\neg(\forall_{X\in\Omega}(P(X) \to \exists_{Y\in\Omega} (P(Y) \,\&\, Y <_\Omega X)))$$

A tal scopo supponiamo che sia vero che

$$\forall_{X\in\Omega}(P(X) \to \exists_{Y\in\Omega} P(Y) \,\&\, Y <_\Omega X)$$

e definiamo un predicato $Q\in\overline{B}\to\mathsf{U}_0$ su B in tal modo

$$\lambda x\in B.Q(x)$$

ove

$$Q(x) \equiv \exists_{Y\in\Omega}\ P(Y)\ \&\ (\ \exists_{g\in\overline{Y}\to\overline{B}}\ (\,(\mathsf{Mon})(g)\ \&\ g(\overline{Y}) <_{mag} x\,)\ )$$

Utilizzando l'ipotesi su $P(B)$ *true* si ha che esiste un $X \in \Omega$ tale che $P(X)$ *true* e $\overline{X} <_\Omega B$ e da questo che esiste un $m \in B$ ed una $g \in \overline{X} \to B$ tale che $g(\overline{X}) <_{mag} m$. Dunque senz'altro vale

$$Q(m)\ true$$

Poi si dimostra che per ogni $m' \in B$ tale che $Q(m')$ *true* si trova un suo minorante che soddisfa Q. Infatti dal fatto che $Q(m')$ *true* si ha che esiste un $X' \in \Omega$ per cui $P(X')$ *true* con una $g' \in \overline{X'} \to B$ tale che $g(\overline{X'}) <_{mag} m'$. Quindi a sua volta sfruttando l'ipotesi su Ω si ha che esiste $Y \in \Omega$ tale che $P(Y)$ *true* e $\overline{Y} <_\Omega \overline{X'}$ tramite una $f \in \overline{Y} \to \overline{X'}$ con un $m'' \in \overline{X'}$ tale che $f(\overline{Y}) <_{mag} m''$.
Ora considerando che m' è un maggiorante dell'immagine di g' si conclude

$$g'(m'') <_B m'$$

e pure che

$$Q(g'(m''))\ true$$

in quanto $\lambda x \in \overline{Y}.\mathsf{Ap}(\,g'\,,(\mathsf{Ap}(f,x)\,)\in \overline{Y} \to B$ è una funzione monotona da $\overline{Y}$ a B (lo si dimostri per bene!) la cui immagine ha come maggiorante proprio $g'(m'')$ per monotonia di f e g'.
Dato che B è ben fondato abbiamo trovato una contraddizione.

12.2 $\Omega <_\Omega \Omega$

Si noti che $\Omega <_\Omega \Omega$ vale.
A tal scopo dato $X \in \Omega$ definiamo $f \in \Omega \to \Omega$ in modo tale che

$$\overline{f}(X) \equiv \pi_1(f(X)) \equiv \Sigma_{Y\in\Omega}\ \overline{Y} <_\Omega \overline{X}$$

ove $\overline{f} \in \Omega\ \to \mathsf{U_0}$

$$W_1 <_{\overline{f}(X)} W_2 \equiv \pi_1(W_1) <_\Omega \pi_1(W_2)$$

Si osservi che è necessario dimostrare che la relazione $<_\Omega$ è transitiva per dimostrare che la funzione f indotta da $\overline{f}$ è monotona.

12.3 La contraddizione

Per ottenere una contraddizione si definisca il predicato

$$P \in \Omega\ \to\ \mathsf{U_0}$$

in tal modo

$$P \equiv \lambda Z \in \Omega.\ \Omega\ <_\Omega Z$$

Chiaramente vale

$$P(\Omega)$$

e pure che dato un $X \in \Omega$ tale che $P(X)$ *true* si trova che

$$\Omega \ <_{\Omega} X$$

che contraddice il fatto che Ω è un tipo ben fondato, ovvero la teoria è *inconsistente*.

Esercizio: Dopo aver formalizzato la dimostrazione in tutti i dettagli si provi ad enucleare i punti in cui $\mathsf{U}_0 \in \mathsf{U}_0$ è necessario per ottenere la contraddizione.

13 Esercizio riassuntivo:

Formulare per ogni tipo $\mathcal{T}$ della teoria dei tipi di Martin-Löf, escluso il tipo prodotto dipendente, il corrispondente principio $\rho(w)$ *prop* $[w \in \mathcal{T}]$ che stabilisce che ogni suo elemento w è un *elemento canonico a meno di uguaglianza proposizionale* e stabilire in che relazione sta il tipo $\Sigma_{w\in\mathcal{T}}\ \rho(w)$ con il tipo $\mathcal{T}$ stesso sia nella versione intensionale che nella versione estensionale, ovvero stabilire se sono isomorfi come tipi o equivalenti nel caso siano pensati entrambi come proposizioni.
Ad esempio si stabilisca la relazione tra Nat e il tipo

$$\Sigma_{x\in\mathsf{Nat}}\ (\ \mathsf{Id}(\mathsf{Nat},0,x)\ \vee\ \exists_{y\in\mathsf{Nat}}\ \mathsf{Id}(\mathsf{Nat},\mathsf{succ}(y),x)\ )$$

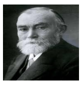
e tra il tipo $\mathsf{N_1}$ e il tipo

$$\Sigma_{x\in\mathsf{N_1}}\ \mathsf{Id}(\mathsf{Nat},*,x)$$

Esercizi *facoltativi* su *naive set theory* con assioma di comprensione

Sia $\mathcal{L}_1$ il linguaggio del primo ordine, ovvero un linguaggio contenente connettivi $\&, \vee, \rightarrow$, costante del vero tt, costante del falso $\perp$, quantificatori $\forall, \exists$ e predicato atomico di uguaglianza $=$, con sole variabili $\mathbf{x}, \mathbf{y}, \mathbf{z} \ldots$ in numero numerabile e il simbolo di relazione binaria ϵ di *appartenza* che utilizzato tra due variabili ha questo significato inteso

$$\mathbf{x}\epsilon\mathbf{y} \quad \equiv \quad \text{“l'insieme } x \text{ appartiene all'insieme } y\text{”}$$

Definiamo poi le seguenti teorie del primo ordine:

Def. 13.1 Sia $\mathcal{T}_1$ la teoria del primo ordine nel linguaggio $\mathcal{L}_1$ ottenuta estendendo il calcolo della logica classica del primo ordine con il seguente **assioma di comprensione**:

per ogni formula ϕ di $\mathcal{L}$ ove $\mathbf{y} \notin \mathbf{VL}(\phi)$

$$\mathbf{ax-compre} \qquad \exists\mathbf{y}\ \ \forall\mathbf{z}\ (\ \mathbf{z}\epsilon\mathbf{y}\ \leftrightarrow\ \phi\ )$$

ovvero *informalmente* con la notazione insiemistica della pratica usuale

$$\textbf{ax-compre} \qquad \exists\mathbf{y}\ \ \mathbf{y} \equiv \{\ \mathbf{z}\ \mid\ \phi\ \}$$

Def. 13.2 Sia $\mathcal{L}_\mathbf{2}$ il linguaggio ottenuto estendendo $\mathcal{L}_\mathbf{1}$ con una costante $\mathbf{V}$ che indica l'*universo di tutti gli insiemi*. Sia $\mathcal{T}_2$ la teoria del primo ordine nel linguaggio ottenuta estendendo il calcolo della logica classica del primo ordine con il seguente assioma

per ogni formula ϕ di $\mathcal{L}$ ove $\mathbf{y} \notin \mathbf{VL}(\phi)$

$$\textbf{ax-compre}_V \qquad \exists\mathbf{y}\epsilon\mathbf{V}\ \forall\mathbf{z}\ (\ \ \mathbf{z}\epsilon\mathbf{V}\ \rightarrow\ (\ \mathbf{z}\epsilon\mathbf{y}\ \leftrightarrow\ \phi\ )\ \ )$$

ovvero *informalmente* con la notazione insiemistica della pratica usuale

$$\textbf{ax-compre}_V \qquad \exists\mathbf{y}\ \ (\ \ \mathbf{y}\epsilon\mathbf{V}\ \&\ \mathbf{y} \equiv \{\ \mathbf{z}\epsilon\mathbf{V}\ \mid\ \phi\ \}\ \ )$$

Def. 13.3 Sia $\mathcal{T}_3$ la teoria del primo ordine nel linguaggio $\mathcal{L}_2$ ottenuta estendendo il calcolo della logica classica del primo ordine con il seguente assioma

per ogni formula ϕ di $\mathcal{L}$ ove $\mathbf{y} \notin \mathbf{VL}(\phi)$

$$\textbf{ax-sep}_V \qquad \forall\mathbf{w}\ \mathbf{w}\epsilon\mathbf{V}\ \ \exists\mathbf{y}\epsilon\mathbf{V}\ \forall\mathbf{z}\ (\ \ \mathbf{z}\epsilon\mathbf{w}\ \rightarrow\ (\ \mathbf{z}\epsilon\mathbf{y}\ \leftrightarrow\ \phi\ )\ \ )$$

ovvero *informalmente* con la notazione insiemistica della pratica usuale dato un *insieme* $\mathbf{w}$ *in* $\mathbf{V}$ ovvero tale che valga $\mathbf{w}\epsilon\mathbf{V}$ allora

$$\textbf{ax-sep}_V \qquad \exists\mathbf{y}\ \ (\ \ \mathbf{y}\epsilon\mathbf{V}\ \&\ \mathbf{y} \equiv \{\ \mathbf{z}\epsilon\mathbf{w}\ \mid\ \phi\ \}\ \ )$$

Esercizi
Dire se sono valide o meno le seguenti affermazioni producendo un'argomentazione (informale) nella teoria proposta:

1. **Teorema 1** La teoria $\mathcal{T}_\mathbf{1}$ è contraddittoria.
2. **Teorema 2** Nella teoria $\mathcal{T}_\mathbf{1}$ esiste un insieme $\mathbf{V} \equiv \{\mathbf{x}\ \mid\ \mathbf{x} = \mathbf{x}\ \}$.
3. **Teorema 3** La teoria $\mathcal{T}_\mathbf{2}$ è contraddittoria.

4. **Teorema 4** Se $\mathbf{V}\epsilon\mathbf{V}$ allora la teoria T_3 è contraddittoria.

5. **Teorema 5** Se la teoria $\mathcal{T}_\mathbf{3}$ NON è contraddittoria allora l'universo $\mathbf{V}$ di tutti gli insiemi NON è un insieme ovvero $\mathbf{V} \not\epsilon \mathbf{V}$ vale in $\mathcal{T}_\mathbf{3}$.

6. Basta tipare le variabili nell'assioma di comprensione per ottenere una teoria degli insiemi non contraddittoria? (si pensi agli esempi di teorie date sopra).

Esercizi *facoltativi* **su λ-calcolo** (detto anche *lambda-calcolo*) puro

Ricordiamo in breve la sintassi del λ-calcolo puro di Church consistente di termini $\mathbf{t}$ definiti secondo la grammatica:

$$\mathbf{t} \ \equiv \ \mathbf{x} \mid \mathbf{t_1(t_2)} \mid \lambda\mathbf{x.t}$$

Definiamo poi la seguente relazione $\rightarrow_1$ di **un passo di riduzione** tra termini $\mathbf{t}$ ottenuta utilizzando le seguenti regole:

1. per ogni termine $\mathbf{t}$ e $\mathbf{b}$
$$(\lambda\mathbf{x.t})(\mathbf{b}) \rightarrow_\mathbf{1} \mathbf{t}[\mathbf{x}/\mathbf{b}] \qquad (\beta\text{-}\mathbf{riduzione})$$

2. per ogni termine $\mathbf{b}, \mathbf{b_1}, \mathbf{b_2}$ e $\mathbf{a}, \mathbf{a_1}, \mathbf{a_2}$
$$\frac{\mathbf{a_1} \ \rightarrow_\mathbf{1} \mathbf{a_2}}{\mathbf{a_1(b)} \rightarrow_\mathbf{1} \mathbf{a_2(b)}} \qquad \frac{\mathbf{b_1} \ \rightarrow_\mathbf{1} \mathbf{b_2}}{\mathbf{a(b_1)} \rightarrow_\mathbf{1} \mathbf{a(b_2)}} \qquad \frac{\mathbf{t_1} \ \rightarrow_\mathbf{1} \mathbf{t_2}}{\lambda\mathbf{x.t_1} \rightarrow_\mathbf{1} \lambda\mathbf{x.t_2}}$$

Si osservi che secondo la relazione di riduzione enunciata un termine $\mathbf{s}$ del λ-calcolo può essere ridotto a nessuno o a uno o a più lambda-termini (ad esempio una variabile $\mathbf{x}$ NON si riduce ad alcun termine, il lambda termine $(\lambda\mathbf{x.x})(\mathbf{y})$ si riduce per β-riduzione alla variabile $\mathbf{y}$ mentre si osservi che il lambda termine $(\lambda\mathbf{x}.(\ (\lambda\mathbf{y.y})(\mathbf{x})\ )\ )(\mathbf{z})$ può essere ridotto a due termini diversi, quali?).
Per enunciare il concetto di convergenza ci limitiamo alle sue sottorelazioni deterministiche (che scelgono come ridurre un termine in modo univoco) denominate *strategie di riduzione deterministica*:

Def. 13.4 Una *strategia di riduzione deterministica* per i termini del λ-calcolo descritto sopra è una sottorelazione *deterministica* della relazione di riduzione descritta sopra, ovvero una relazione di riduzione tra termini rispetto a cui

- se un termine $\mathbf{t}$ è riducibile ad un termine $\mathbf{s}$ allora $\mathbf{t} \rightarrow_\mathbf{1} \mathbf{s}$;
- un termine $\mathbf{t}$ è riducibile ad al più un'unico termine $\mathbf{s}$, ovvero se $\mathbf{t} \rightarrow_\mathbf{1} \mathbf{s_1}$ e $\mathbf{t} \rightarrow_\mathbf{1} \mathbf{s_2}$ allora $\mathbf{s_1}$ è identico come termine ad $\mathbf{s_2}$ (a meno di rinomina delle variabili vincolate).

Ad esempio una strategia deterministica è determinata dalla sola β-riduzione.

Def. 13.5 Un termine $\mathbf{t}$ del λ-calcolo descritto sopra si dice *convergente rispetto ad una strategia di riduzione deterministica* se:

- esiste un numero finito $\mathbf{n} \geq \mathbf{1}$ di termini $\mathbf{s_1}, \ldots, \mathbf{s_n}$ tale che $\mathbf{s_1} \ \equiv \ \mathbf{t}$ e $\mathbf{s_i} \rightarrow_\mathbf{1} \mathbf{s_{i+1}}$ per $\mathbf{i} = \mathbf{1}, \ldots, \mathbf{n}{-}\mathbf{1}$ e $\mathbf{s_n}$ NON è riducibile ad alcun termine.

Si noti che nella definizione sopra viene incluso il caso in cui $\mathbf{t}$ NON si riduce ad alcun termine.

Quindi si deduce che

Def. 13.6 Un termine $\mathbf{t}$ del λ-calcolo descritto sopra si dice *NON convergente rispetto ad una strategia di riduzione deterministica* se:

- esiste una quantità numerabile di termini $\mathbf{s_i}$ al variare di $\mathbf{i} \in \mathbf{Nat}$ tale che $\mathbf{s_1} \ \equiv \ \mathbf{t}$ e $\mathbf{s_i} \rightarrow_\mathbf{1} \mathbf{s_{i+1}}$ per ogni $\mathbf{i} \in \mathbf{Nat}$ (ossia esiste una lista infinita di passi computazioni a partire da $\mathbf{t}$).

Esercizi

1. Descrivere un termine del λ-calcolo sopra che è convergente con almeno un passo di riduzione rispetto ad una specifica strategia di riduzione deterministica.

2. Descrivere due termini diversi del λ-calcolo sopra che NON sono convergenti sempre rispetto ad una strategia di riduzione deterministica.

3. Che relazione c'è tra il lambda-calcolo puro con le regole di riduzione date sopra rispetto a quello in cui adottando la stessa sintassi di termini imponiamo la seguente definizione di riduzione $\rightarrow_1^*$

   (a) per ogni termine $\mathbf{t}$ e $\mathbf{b}$

   $$(\lambda \mathbf{x}.\mathbf{t})(\mathbf{b}) \rightarrow_{\mathbf{1}}^{*} \mathbf{t}[\mathbf{x}/\mathbf{b}]$$

   (b) per ogni termine $\mathbf{b}, \mathbf{b_1}, \mathbf{b_2}$ e $\mathbf{a}, \mathbf{a_1}, \mathbf{a_2}$

   $$\frac{\mathbf{a_1} \ \rightarrow_{\mathbf{1}}^{*} \mathbf{a_2} \qquad \mathbf{b_1} \ \rightarrow_{\mathbf{1}}^{*} \mathbf{b_2}}{\mathbf{a_1}(\mathbf{b_1}) \rightarrow_{\mathbf{1}}^{*} \mathbf{a_2}(\mathbf{b_2})} \qquad\qquad \frac{\mathbf{t_1} \ \rightarrow_{\mathbf{1}}^{*} \mathbf{t_2}}{\lambda \mathbf{x}.\mathbf{t_1} \rightarrow_{\mathbf{1}}^{*} \lambda \mathbf{x}.\mathbf{t_2}}$$

14 APPENDICE: Calcolo dei sequenti per la deduzione naturale intuizionista con uguaglianza DNI$_=$ (con meta-variabili per formule)

Il calcolo della deduzione naturale **DNI$_=$**è determinato dagli assiomi e regole presentate sotto ricordando che

- i simboli $\mathtt{fr}$, $\mathtt{fr}_1$e $\mathtt{fr}_2$, ψ sono META-variabili che indicano formule complesse arbitrarie come le lettere mentre le lettere greche maiuscole come Γ sono meta-variabili che indicano liste arbitrarie di formule;
- la scrittura $\mathtt{fr}[\mathtt{x/t}]$ indica la formula ottenuta sostituendo TUTTE le occorrenze libere della variabile $\mathbf{x}$ in $\mathtt{fr}$ con il termine $\mathbf{t}$ (si veda la definizione formale di sostituzione in sezione **??**);
- Il simbolo $\mathbf{t}$ è una META-variabile che indica un termine qualsiasi del linguaggio che può essere una delle variabili $\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots$ oppure una delle costanti $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$;
- le regole di quantificazioni sotto si intendono chiuse sulla sostituzione della variabili $\mathbf{x}, \mathbf{w}$ che appaiono sotto con QUALSIASI altra variabile purchè vengano rispettate le condizioni indicate.

$$\begin{array}{c}\text{ax-id}\\ \Gamma, \mathtt{fr}, \Gamma' \vdash \mathtt{fr}\end{array} \qquad \begin{array}{c}\text{ax-}\mathtt{tt}\\ \Gamma \vdash \mathtt{tt}\end{array}$$

$$\frac{\Gamma \vdash \perp}{\Gamma \vdash \mathtt{fr}}\ \text{ex-f-q} \qquad \frac{\Sigma, \Gamma, \Theta, \Gamma', \Delta \vdash \mathtt{fr}}{\Sigma, \Gamma', \Theta, \Gamma, \Delta \vdash \mathtt{fr}}\ \mathrm{sc}_{\mathrm{sx}}$$

$$\frac{\Gamma \vdash \mathtt{fr}_1 \& \mathtt{fr}_2}{\Gamma \vdash \mathtt{fr}_1}\ \&-\mathrm{Sn}_1 \qquad \frac{\Gamma \vdash \mathtt{fr}_1 \& \mathtt{fr}_2}{\Gamma \vdash \mathtt{fr}_2}\ \&-\mathrm{Sn}_2 \qquad \frac{\Gamma \vdash \mathtt{fr}_1 \quad \Gamma \vdash \mathtt{fr}_2}{\Gamma \vdash \mathtt{fr}_1 \& \mathtt{fr}_2}\ \&-\mathrm{D}$$

$$\frac{\Gamma \vdash \mathtt{fr}_1 \vee \mathtt{fr}_2 \quad \Gamma, \mathtt{fr}_1 \vdash \psi \quad \Gamma, \mathtt{fr}_2 \vdash \psi}{\Gamma \vdash \psi}\ \vee-\mathrm{Sn} \qquad \frac{\Gamma \vdash \mathtt{fr}_1}{\Gamma \vdash \mathtt{fr}_1 \vee \mathtt{fr}_2}\ \vee-\mathrm{Dn}_1 \qquad \frac{\Gamma \vdash \mathtt{fr}_2}{\Gamma \vdash \mathtt{fr}_1 \vee \mathtt{fr}_2}\ \vee-\mathrm{Dn}_2$$

$$\frac{\Gamma \vdash \mathtt{fr}_1 \quad \Gamma \vdash \neg \mathtt{fr}_1}{\Gamma \vdash \perp}\ \neg-\mathrm{Sn} \qquad \frac{\Gamma, \mathtt{fr}_1 \vdash \perp}{\Gamma \vdash \neg \mathtt{fr}_1}\ \neg-\mathrm{D}$$

$$\frac{\Gamma \vdash \mathtt{fr}_1 \to \mathtt{fr}_2 \quad \Gamma \vdash \mathtt{fr}_1}{\Gamma \vdash \mathtt{fr}_2}\ \to-\mathrm{Sn} \qquad \frac{\Gamma, \mathtt{fr}_1 \vdash \mathtt{fr}_2}{\Gamma \vdash \mathtt{fr}_1 \to \mathtt{fr}_2}\ \to-\mathrm{D}$$

$$\frac{\Gamma \vdash \forall x\ \mathtt{fr}}{\Gamma \vdash \mathtt{fr}[x/t]}\ \forall-\mathrm{Sn} \qquad \frac{\Gamma \vdash \mathtt{fr}[x/w]}{\Gamma \vdash \forall x\ \mathtt{fr}}\ \forall-\mathrm{D}\ (w \notin VL(\Gamma, \forall x\ \mathtt{fr}))$$

$$\frac{\Gamma \vdash \exists x\ \mathtt{fr} \quad \Gamma, \mathtt{fr}[x/w] \vdash \psi}{\Gamma \vdash \psi}\ \exists-\mathrm{Sn}\ (w \notin VL(\Gamma, \exists x\ \mathtt{fr}, \psi)) \qquad \frac{\Gamma \vdash \mathtt{fr}[x/t]}{\Gamma \vdash \exists x\ \mathtt{fr}}\ \exists-\mathrm{Dn}$$

$$\frac{\Gamma \vdash t = s \quad \Gamma \vdash \mathtt{fr}_1[x/t]}{\Gamma \vdash \mathtt{fr}[x/s]}\ =-\mathrm{Sn} \qquad \begin{array}{c}=-\mathrm{ax}\\ \Gamma \vdash t = t\end{array}$$

15 APPENDICE: Calcolo dei sequenti per la deduzione naturale classica con uguaglianza DNCla$_=$

Regole di **DNI$_=$**

$+$

$$\begin{array}{c}\text{EM-ax}\\ \boldsymbol{\Gamma} \vdash \mathtt{fr} \vee \neg \mathtt{fr}\end{array}$$

16 APPENDICE: Assiomi dell'aritmetica di Peano

Gli assiomi dell'aritmetica di Peano sono formulati in un linguaggio del primo ordine con uguaglianza e l'aggiunta della costante *zero* **0** e della funzione successore $\mathsf{succ}(x)$ e sono i seguenti:

$$Ax1. \vdash \forall x \ \mathsf{succ}(x) \neq 0$$
$$Ax2. \vdash \forall x \ \forall y \ (\ \mathsf{succ}(x) = \mathsf{succ}(y) \rightarrow x = y \ )$$
$$Ax3. \vdash \forall x \ \ x + 0 = x$$
$$Ax4. \vdash \forall x \ \forall y \ x + \mathsf{succ}(y) = \mathsf{succ}(x + y)$$
$$Ax5. \vdash \forall x \ x \cdot 0 = 0$$
$$Ax6. \vdash \forall x \ \forall y \ x \cdot \mathsf{succ}(y) = x \cdot y + x$$
$$Ax7. \vdash \phi(0) \ \& \ \forall x \ (\ \phi(x) \ \rightarrow \ \phi(\mathsf{succ}(x)) \ ) \ \rightarrow \ \forall x \ \phi(x)$$

ove il numerale **n** si rappresenta in tal modo

$$\mathbf{n} \equiv \underbrace{\mathsf{succ}(\mathsf{succ}\ldots(0))}_{\text{n-volte}}$$

e quindi per esempio

$$1 \equiv \mathsf{succ}(0)$$
$$2 \equiv \mathsf{succ}(\ \mathsf{succ}(0)\ )$$

Poi si definisce

Aritmetica classica di Peano (in breve **PA**)= il calcolo ottenuto aggiungendo gli assiomi sopra a $\mathbf{DNCla}_=$
Aritmetica intuizionista di Heyting (in breve **HA**) = il calcolo ottenuto aggiungendo gli assiomi sopra a $\mathbf{DNI}_=$

17 APPENDICE: Calcolo dei sequenti $\mathbf{LC}_=$ per la logica classica predicativa con uguaglianza

$$\overset{\text{ax-id}}{\mathbf{\Gamma}, \mathbf{A}, \mathbf{\Gamma}' \vdash \mathbf{\Delta}, \mathbf{A}, \mathbf{\Delta}'} \qquad \overset{\text{ax-}\bot}{\mathbf{\Gamma}, \bot, \mathbf{\Gamma}' \vdash \nabla} \qquad \overset{\text{ax-}\texttt{tt}}{\mathbf{\Gamma} \vdash \nabla, \texttt{tt}, \nabla'}$$

$$\frac{\mathbf{\Sigma}, \mathbf{\Gamma}, \mathbf{\Theta}, \mathbf{\Gamma}', \mathbf{\Delta} \vdash \mathbf{\Sigma}'}{\mathbf{\Sigma}, \mathbf{\Gamma}', \mathbf{\Theta}, \mathbf{\Gamma}, \mathbf{\Delta} \vdash \mathbf{\Sigma}'}\ \text{sc}_{\text{sx}} \qquad \frac{\mathbf{\Gamma} \vdash \mathbf{\Sigma}, \mathbf{\Delta}, \mathbf{\Theta}, \mathbf{\Delta}', \nabla}{\mathbf{\Gamma} \vdash \mathbf{\Sigma}, \mathbf{\Delta}', \mathbf{\Theta}, \mathbf{\Delta}, \nabla}\ \text{sc}_{\text{dx}}$$

$$\frac{\mathbf{\Gamma}, \mathbf{A}, \mathbf{B} \vdash \mathbf{\Delta}}{\mathbf{\Gamma}, \mathbf{A}\&\mathbf{B} \vdash \mathbf{\Delta}}\ \&\text{–S} \qquad \frac{\mathbf{\Gamma} \vdash \mathbf{A}, \mathbf{\Delta} \quad \mathbf{\Gamma} \vdash \mathbf{B}, \mathbf{\Delta}}{\mathbf{\Gamma} \vdash \mathbf{A}\&\mathbf{B}, \mathbf{\Delta}}\ \&\text{–D}$$

$$\frac{\mathbf{\Gamma}, \mathbf{A} \vdash \mathbf{\Delta} \quad \mathbf{\Gamma}, \mathbf{B} \vdash \mathbf{\Delta}}{\mathbf{\Gamma}, \mathbf{A} \vee \mathbf{B} \vdash \mathbf{\Delta}}\ \vee\text{–S} \qquad \frac{\mathbf{\Gamma} \vdash \mathbf{A}, \mathbf{B}, \mathbf{\Delta}}{\mathbf{\Gamma} \vdash \mathbf{A} \vee \mathbf{B}, \mathbf{\Delta}}\ \vee\text{D}$$

$$\frac{\mathbf{\Gamma} \vdash \mathbf{A}, \mathbf{\Delta}}{\mathbf{\Gamma}, \neg\mathbf{A} \vdash \mathbf{\Delta}}\ \neg\text{–S} \qquad \frac{\mathbf{\Gamma}, \mathbf{A} \vdash \mathbf{\Delta}}{\mathbf{\Gamma} \vdash \neg\mathbf{A}, \mathbf{\Delta}}\ \neg\text{–D}$$

$$\frac{\mathbf{\Gamma} \vdash \mathbf{A}, \mathbf{\Delta} \quad \mathbf{\Gamma}, \mathbf{B} \vdash \mathbf{\Delta}}{\mathbf{\Gamma}, \mathbf{A} \rightarrow \mathbf{B} \vdash \mathbf{\Delta}}\ \rightarrow\text{–S} \qquad \frac{\mathbf{\Gamma}, \mathbf{A} \vdash \mathbf{B}, \mathbf{\Delta}}{\mathbf{\Gamma} \vdash \mathbf{A} \rightarrow \mathbf{B}, \mathbf{\Delta}}\ \rightarrow\text{–D}$$

$$\frac{\mathbf{\Gamma}, \forall \mathbf{x}\ \mathbf{A}(\mathbf{x}), \mathbf{A}(\mathbf{t}) \vdash \nabla}{\mathbf{\Gamma}, \forall \mathbf{x}\ \mathbf{A}(\mathbf{x}) \vdash \nabla}\ \forall\text{–S} \qquad \frac{\mathbf{\Gamma} \vdash \mathbf{A}(\mathbf{w}), \nabla}{\mathbf{\Gamma} \vdash \forall \mathbf{x}\mathbf{A}(\mathbf{x}), \nabla}\ \forall\text{–D}\ (w \notin VL(\mathbf{\Gamma}, \forall \mathbf{x}\mathbf{A}(\mathbf{x}), \nabla))$$

$$\frac{\mathbf{\Gamma}, \mathbf{A}(\mathbf{w}) \vdash \nabla}{\mathbf{\Gamma}, \exists \mathbf{x}\ \mathbf{A}(\mathbf{x}) \vdash \nabla}\ \exists\text{–S}\ (w \notin VL(\mathbf{\Gamma}, \exists \mathbf{x}\ \mathbf{A}(\mathbf{x}), \nabla)) \qquad \frac{\mathbf{\Gamma} \vdash \mathbf{A}(\mathbf{t}), \exists \mathbf{x}\ \mathbf{A}(\mathbf{x}), \nabla}{\mathbf{\Gamma} \vdash \exists \mathbf{x}\ \mathbf{A}(\mathbf{x}), \nabla}\ \exists\text{–D}$$

$$\frac{\mathbf{\Sigma}, \mathbf{t} = \mathbf{s}, \mathbf{\Gamma}(\mathbf{t}) \vdash \mathbf{\Delta}(\mathbf{t}), \nabla}{\mathbf{\Sigma}, \mathbf{\Gamma}(\mathbf{s}), \mathbf{t} = \mathbf{s} \vdash \mathbf{\Delta}(\mathbf{s}), \nabla}\ =\text{–S}_1 \qquad \frac{\mathbf{\Sigma}, \mathbf{s} = \mathbf{t}, \mathbf{\Gamma}(\mathbf{t}) \vdash \mathbf{\Delta}(\mathbf{t}), \nabla}{\mathbf{\Sigma}, \mathbf{\Gamma}(\mathbf{s}), \mathbf{s} = \mathbf{t} \vdash \mathbf{\Delta}(\mathbf{s}), \nabla}\ =\text{–S}_2$$

$$\overset{=\text{–ax}}{\mathbf{\Gamma} \vdash \mathbf{t} = \mathbf{t}, \mathbf{\Delta}}$$