Teoria delle espressioni

Silvio Valentini
Dipartimento di Matematica Pura ed Applicata
Università di Padova
via G. Belzoni n.7, I–35131 Padova, Italy
silvio@math.unipd.it

April 16, 2003

1 Premessa

Prima di parlare della teoria delle espressioni vera e propria richiamiamo alcune nozioni generali riguardanti il linguaggio che utilizzeremo per poter affrontare questioni inerenti *segni*, *espressioni*, *deduzioni*, *proposizioni*, *proprietà*, *verità*, *dimostrazioni* e *dimostrabilità*. Uno dei compiti fondamentali della logica è infatti quello di riuscire a dare una definizione che sia la più appropriata possibile di questi concetti.

Diciamo che un segno è una forma (sonora, gestuale, visiva, ...) riconoscibile senza ambiguità ogni volta che compare e riproducibile a piacimento, mentre una espressione è un gruppo di segni combinati secondo delle regole sintattiche, che chiameremo regole di *formazione* delle espressioni corrette.

Le espressioni infatti, affinchè si possano leggere, devono essere sequenze di segni scritte in modo sensato, anche se non necessariamente dotate di significato. Vogliamo quindi astrarre dal significato delle espressioni stesse preoccupandoci della loro chiarezza e non ambiguità. Definiremo quindi un sistema di calcolo, il cosiddetto λ-calcolo tipato, che ci permetterà di decidere se due espressioni sono uguali o meno.

2 Il linguaggio e i suoi elementi

Introduciamo ora gli elementi che costituiscono il linguaggio simbolico della logica. Essi sono: *costanti*, *variabili*, *applicazioni* e *astrazioni*.

Le costanti sono i segni che vengono utilizzati per indicare oggetti precisi che appartengono all'insieme di cui vogliamo parlare.

Le variabili sono segni che vengono utilizzati per indicare elementi che possono essere sostituiti con altri.

Applicazioni ed astrazioni meritano invece una trattazione di maggior dettaglio.

2.1 Applicazione

L'applicazione è l'operazione che consiste nell'applicare una espressione, vista come una funzione, ad un'altra espressione, vista come un argomento per tale funzione. Affinchè l'espressione risultante continui ad avere senso è necessario fare attenzione al *tipo* dell'elemento a cui la funzione viene applicata. Il λ-calcolo tipato fornisce appunto le regole per l'utilizzo corretto dell'applicazione. Introduciamo quindi una nuova nozione per rappresentare il tipo degli argomenti a cui può essere applicata un'espressione: l'*arietà*. Essa viene definita in modo induttivo tramite le seguenti regole:

Definizione 2.1 (Arietà)

$$\textit{(base)} \qquad 0 \ \mathsf{ariety}$$

$$\textit{(freccia)} \qquad \frac{\alpha \ \mathsf{ariety} \qquad \beta \ \mathsf{ariety}}{\alpha \to \beta \ \mathsf{ariety}}$$

Ad una espressione viene assegnata l'arietà $\alpha \to \beta$ se essa si applica ad oggetti di arietà α per ottenere un risultato di arietà β mentre una espressione di arietà 0 non si può applicare a nessun argomento. Ad esempio l'espressione sin, ottenuta dalla costante sin, può essere applicata ad un argomento x di arietà 0 ottenendo l'espressione $\sin(x)$, a sua volta di arietà 0.

Formalizziamo quanto detto introducendo le regole seguenti che permettono di formare espressioni in modo corretto.

Definizione 2.2 (λ-espressioni (prima parte))

$$\textit{(costanti)} \qquad \frac{\mathsf{C} : \alpha \ \mathsf{const}}{\mathsf{C} : \alpha \ \mathsf{exp}}$$

$$\textit{(variabili)} \qquad \frac{x : \alpha \ \mathsf{var}}{x : \alpha \ \mathsf{exp}}$$

$$\textit{(applicazione)} \qquad \frac{b : \alpha \to \beta \ \mathsf{exp} \qquad a : \alpha \ \mathsf{exp}}{b(a) : \beta \ \mathsf{exp}}$$

Una costante o una variabile di arietà α è quindi una espressione della stessa arietà, mentre l'applicazione dà luogo ad una espressione sole se l'arietà dell'applicando e quella dell'argomento vanno d'accordo.

Come già accennato se una espressione di arietà $\alpha \to \beta$ viene applicata ad un argomento di arietà α otteniamo una espressione di arierà β. Ora possiamo comprendere perchè, in questo contesto, espressioni quali sin(sin) e $x(a)$, dove x è una variabile di arietà 0, sono prive di senso: sin, avendo arietà $0 \to 0$, può essere applicata solamente ad un argomento di arietà 0 e non ad un argomento di arietà $0 \to 0$, mentre x, avendo arietà 0, non si può applicare a nessun tipo di argomento.

Esempio 1
L'espressione $+$ si applica a due argomenti di arietà 0. Quindi se la applichiamo ad una espressione di arietà 0, per esempio $3 : 0 \ \mathsf{exp}$ otteniamo una

espressione di arietà $0 \to 0$, cioè $+(3) : 0 \to 0$ exp che può nuovamente essere applicata ad una espressione di arietà 0, per esempio $x : 0$ exp, ottenendo una espressione di arietà 0, cioè $+(3)(x) : 0$ exp.

Esempio 2
Per il segno di integrale si può procedere in modo analogo. Consideriamo la seguente scrittura:

$$\int_a^b f(x)\mathsf{d}x \equiv \int(a)(b)(f)$$

L'espressione $\int$ si applica a tre argomenti: a, b di arietà 0 e f di arietà $0 \to 0$. Otteniamo così l'arietà della costante $\int$:

$$\int : 0 \to (0 \to ((0 \to 0) \to 0))\ \mathsf{const}$$

Se applichiamo $\int$ ad una espressione di arietà 0, ad esempio $4 : 0$ exp, otteniamo l'espressione $\int(4)$ di arietà $0 \to ((0 \to 0) \to 0)$. Applicando a sua volta questa espressione ad una nuova espressione di arietà 0, per esempio $6 : 0$ exp otteniamo una espressione di arietà $(0 \to 0) \to 0$, vale a dire $\int(4)(6)$. Quest'ultima espressione può essere applicata solamente ad una espressione di arietà $0 \to 0$, ad esempio sin, ottenendo finalmente una espressione di arietà 0, cioè $\int(4)(6)(\sin)$. La prova formale è la seguente:

$$\dfrac{\dfrac{\dfrac{\dfrac{\int : 0 \to (0 \to ((0 \to 0) \to 0))\ \mathsf{const}}{\int : 0 \to (0 \to ((0 \to 0) \to 0))\ \mathsf{exp}} \quad \dfrac{4 : 0\ \mathsf{const}}{4 : 0\ \mathsf{exp}}}{\int(4) : 0 \to ((0 \to 0) \to 0)} \quad \dfrac{6 : 0\ \mathsf{const}}{6 : 0\ \mathsf{exp}}}{\int(4)(6) : (0 \to 0) \to 0\ \mathsf{exp}} \quad \dfrac{\sin : 0 \to 0\ \mathsf{const}}{\sin : 0 \to 0\ \mathsf{exp}}}{\int(4)(6)(\sin)}$$

2.2 Astrazione

L'astrazione è l'operazione che permette di "astrarre" una variabile da una espressione trasformandola in una espressione che può essere applicata. Introduciamo la regola che la caratterizza.

Definizione 2.3 (λ-espressioni (segue))

$$\textit{(astrazione)} \qquad \dfrac{b : \beta\ \mathsf{exp} \qquad x : \alpha\ \mathsf{var}}{((x)\ b) : \alpha \to \beta\ \mathsf{exp}}$$

Cioè, se b è una espressione di arietà β ed x è una variabile di arietà α, astraendo x da b otteniamo una espressione di arietà $\alpha \to \beta$ che *non contiene più x come variabile.*

Esempio 3

$$\dfrac{+(3)(x) : 0\ \mathsf{exp} \qquad x : \alpha\ \mathsf{var}}{((x)\ +(3)(x)) : 0 \to 0\ \mathsf{exp}}$$

Esempio 4

$$\dfrac{\dfrac{3 : 0 \ \mathsf{const}}{3 : 0 \ \mathsf{exp}} \quad x : 0 \ \mathsf{exp}}{((x)\ 3) : 0 \to 0 \ \mathsf{exp}}$$

cioè $((x)\ 3)$ è una espressione che rappresenta la funzione costante che dà risultato 3 se applicata ad un qualsiasi argomento di arietà 0.

Esempio 5

$$\dfrac{\dfrac{\sin(+(x)(y)) : 0 \ \mathsf{exp} \quad x : 0 \ \mathsf{var}}{((x)\ \ \sin(+(x)(y))) : 0 \to 0 \ \mathsf{exp}} \quad y : 0 \ \mathsf{var}}{((y)((x)\ \ \sin(+(x)(y)))) : 0 \to (0 \to 0)}$$

dove le variabili x e y non appaiono in $((y)((x)\ \ \sin(+(x)(y))))$.

In generale, per indicare che una variabile x non appare in una espressione b scriveremo $x \notin \mathsf{FV}(b)$ e diremo che $\mathsf{FV}(b)$ indica l'insieme delle variabili libere nell'espressione b.

$\mathsf{FV}(b)$ si può definire formalmente per induzione sulla complessità della struttura della espressione b nel modo che segue:

$$\begin{array}{lcll} \mathsf{FV}(\mathsf{C}) & \equiv & \emptyset & \text{dove } C \text{ è una costante} \\ \mathsf{FV}(x) & \equiv & \{x\} & \text{dove } x \text{ è una variabile} \\ \mathsf{FV}(b(a)) & \equiv & \mathsf{FV}(b) \cup \mathsf{FV}(a) & \\ \mathsf{FV}(((x)\ b)) & \equiv & \mathsf{FV}(b) \backslash \{x\} & \end{array}$$

Quindi la variabile y risulta libera in $((x)\ \ \sin(x))(y)$ mentre la variabile x non appare libera perchè tutte le sue occorrenze sono astratte. Si noti tuttavia che la variabile x appare libera in $((x)\ \ \sin(x))(x)$ visto che c'è comunque una sua occorrenza libera, anche se la sua occorrenza in $\sin(x)$ è astratta.

Nel seguito cercheremo di rendere esplicito il fatto che il significato di una espressione deve dipendere solo dalle variabili che vi appaiono libere, come accade ad esempio nel caso di un integrale il cui valore non dipende dal nome della variabile di integrazione.

In particolare, faremo spesso uso del fatto che data una espressione come $((x)\ b)$ la sola variabile che sicuramente *non* vi compare (libera) è x.

Abbiamo così definito gli elementi del nostro linguaggio ed introdotto le regole che costituiscono la nostra *teoria delle espressioni*. Vediamo ora quale uso possiamo farne.

2.3 La sostituzione

Il prossimo passo consiste nel definire un processo di calcolo per semplificare le scritture quali $((x)\ \ \sin(x))(\pi)$ oppure $((y)((x)\ \ +(x)(y)))(3)$ che rappresentano le consuete funzioni, rispettivamente ad una e a due variabili, applicate ad un argomento.

A tale scopo introduciamo l'operazione di *sostituzione*. Essa consiste nel sostituire, nell'espressione b, l'espressione a al posto delle occorrenze libere della variabile x (notazione $b[x := a]$).

Mentre la spiegazione intuitiva del suo significato è chiara, la presenza della operazione di astrazione rende la sua definizione formale non completamente banale.

Definizione 2.4 (Sostituzione) *Sia* $b : \beta$ exp, $a : \alpha$ exp *e* $x : \alpha$ var *allora la* sostituzione *della variabile* x *con l'espressione* a *nella espressione* b *è definita per induzione sulla complesità della struttura dell'espressione* b *come segue:*

$$
\begin{array}{ll}
\text{(cost.)} & \mathsf{C}[x := a] \equiv \mathsf{C} \\
\text{(var.)} & y[x := a] \equiv \begin{cases} a & \textit{se } x \equiv y \\ y & \textit{altrimenti} \end{cases} \\
\text{(appl.)} & b(c)[x := a] \equiv b[x := a](c[x := a]) \\
\text{(astr.)} & ((y)\ b)[x := a] \equiv \begin{cases} ((y)\ b) & \textit{se } x \equiv y \\ ((y)\ b[x := a]) & \textit{se } x \not\equiv y \textit{ e } y \notin \mathsf{FV}(a) \\ ((t)\ b[y := t][x := a]) & \textit{se } x \not\equiv y,\ y \in \mathsf{FV}(a) \\ & \textit{e } t \textit{ è una nuova va-} \\ & \textit{riabile di arietà } \alpha \end{cases}
\end{array}
$$

La maggior parte dei passi considerati nella definizione dovrebbero essere di comprensione immediata. Qualche dubbio potrebbe forse sorgere nella comprensione del primo punto relativo alla sostituzione della variabile x con l'espressione a nel caso di una astrazione $((x)\ b)$, ma è chiaro che la soluzione proposta nell'algoritmo non fa altro che rendere esplicito il fatto che la variabile x non appare in $((x)\ b)$.

Tuttavia, per comprendere meglio la ragione per cui nell'ultimo punto della definizione dell'algoritmo di sostituzione si usa una *nuova* variabile, cioè una variabile che non compare nè in b nè in a, nel caso in cui si intenda sostituire in $((y)\ b)$ la variabile x con l'espressione a e $x \not\equiv y$ e $y \in \mathsf{FV}(a)$, è forse conveniente fare un esempio. Consideriamo la seguente sostituzione

$$((x)\ \ +(x)(y))[y := x]$$

Potrebbe allora venire spontaneo pensare che il risultato della sostituzione sia:

$$((x)\ \ +(x)(x))$$

Bisogna tuttavia notare che, per la proprietà dell'astrazione, x non compare più come variabile nel risultato della sostituzione, mentre appariva chiaramente nella espressione di partenza. Il significato della scrittura assume quindi una certa importanza: $((x)\ \ +(x)(y))$ rappresenta una espressione che applicata ad un generico elemento c, della stessa arietà della variabile x, dà come risultato $+(c)(y)$. D'altra parte $((x)\ \ +(x)(x))$ rappresenta una funzione completamente diversa in quanto quando essa viene applicata ad un generico elemento c, che naturalmente deve ancora avere la stessa arietà di x, fornisce come risultato $+(c)(c)$. Pertanto la scrittura $((x)\ \ +(x)(y))[y := x]$ *deve* rappresentare una espressione che applicata ad un argomento c restituisce $c + x$, in quanto il significato inteso della sostituzione è quello di ottenere una nuova funzione in cui y viene sostituito con x ma che per il resto si comporta come la precedente.

Nella definizione dell'algoritmo di sostituzione abbiamo formalizzato quanto detto introducendo una nuova variabile non presente nelle espressioni coinvolte in modo da ottenere:

$$\begin{array}{rcl}((x)\ +(x)(y))[y := x] & \equiv & ((t)\ +(x)(y)[x := t])[y := x] \\ & \equiv & ((t)\ +(x)(y)[x := t][y := x]) \\ & \equiv & ((t)\ +(t)(y)[y := x]) \\ & \equiv & ((t)\ +(t)(x))\end{array}$$

Vogliamo ora dimostrare che l'algoritmo di sostituzione termina per qualsiasi scelta corretta delle espressioni su cui la sostituzione viene effettuata. Visto che la definizione dell'algoritmo è ricorsiva la cosa non è completamente scontata. Iniziamo osservando il seguente risultato.

Lemma 2.5 *Le espressioni b e $b[x := t]$, dove t è una nuova variabile della stessa arietà di x, hanno la stessa complessità strutturale.*

Dimostrazione. La dimostrazione procede per induzione sulla complessità della struttura dell'espressione b.

- (Costante) $b \equiv \mathsf{C}$. Bisogna dimostrare che $\mathsf{C}[x := t]$ ha la stessa complessità strutturale di C. Ma questo è ovvio visto che $\mathsf{C}[x := t] \equiv \mathsf{C}$
- (Variabile) Guardando la definizione dell'algoritmo di sostituzione è chiaro che dobbiamo considerare due casi:
  - $b \equiv y \not\equiv x$. Questo caso è completamente simile al caso della costante.
  - $b \equiv x$. Allora $b[x := t] \equiv t$, ma t per ipotesi è una variabile come x e quindi le espressioni b e $b[x := t]$ hanno la stessa complessità.
- (Applicazione) $b \equiv d(a)$. Supponiamo, per ipotesi induttiva, che il risultato valga per le espressioni $d[x := t]$ e $a[x := t]$, in quanto d e a sono espressioni meno complesse di b. Allora $d[x := t]$ ha la stessa complessità di d e $a[x := t]$ ha la stessa complessità di a. Dobbiamo ora dimostrare che $d(a)[x := t]$ ha la stessa complessità di $d(a)$. Ma $d(a)[x := t] \equiv d[x := t](a[x := t])$ e quindi il risultato è ovvio.
- (astrazione) $b \equiv ((z)\ d)$. Dobbiamo dimostrare che $((z)\ d)$ e $((z)\ d)[x := t]$ hanno la stessa complessità strutturale. Per farlo distinguiamo i seguenti tre casi in conformità alla definizione dell'algoritmo di sostituzione:
  - $z \equiv x$. Secondo la definizione dell'algoritmo di sostituzione, otteniamo $((z)\ d)[x := t] \equiv ((z)\ d)$ e quindi $((z)\ d)[x := t]$ e $((z)\ d)$ hanno ovviamente la stessa complessità.
  - $z \not\equiv x$ e $z \not\equiv t$. Per ipotesi induttiva possiamo supporre che l'enunciato valga per l'espressione d, che è meno complessa dell'espressione $((z)\ d)$. Quindi d e $d[x := t]$ hanno la stessa complessità. Ora, utilizzando l'algoritmo di sostituzione otteniamo $((z)\ d)[x := t] \equiv ((z)\ d[x := t])$, pertanto $((z)\ d)[x := t]$ e $((z)\ d)$ hanno la stessa complessità.

– $z \not\equiv x$ e $z \equiv t$. Per ipotesi induttiva possiamo supporre che l'enunciato valga per l'espressione d in quanto essa è meno complessa dell'espressione $((z)\ d)$, cioè possiamo supporre che d abbia la stessa complessità di $d[t := w]$, per ogni variabile w della corretta arietà. Allora, applicando nuovamente l'ipotesi induttiva all'espressione $d[t := w]$, in quanto complessa tanto quanto l'espressione d e quindi meno complessa dell'espressione $((z)\ d)$, otteniamo che $d[t := w][x := t]$ ha la stessa complessità di $d[t := w]$ e quindi di d. Ora, per l'algoritmo di sostituzione, $((t)\ d)[x := t] \equiv ((w)\ d[t := w][x := t])$, dove w è una nuova variabile. Quindi le espressioni $((t)\ d)[x := t]$ e $((t)\ d)$ hanno la stessa complessità strutturale.

Possiamo ora dimostrare il seguente teorema.

Teorema 2.6 (Terminazione dell'algoritmo di sostituzione) *Supponiamo che* $b : \beta\ \mathsf{exp}$*,* $y : \alpha\ \mathsf{var}$ *e* $e : \alpha\ \mathsf{exp}$*. Allora l'esecuzione dell'algoritmo di sostituzione per sostituire la variabile* y *con l'espressione* e *nell'espressione* b *termina.*

Dimostrazione. Per dimostrare la terminazione dell'algoritmo di sostituzione si procede per induzione sulla complessità strutturale dell'espressione b.

- (Costante) $b \equiv \mathsf{C}$. In questo caso $b[y := e] \equiv b$ e, non essendoci chiamate ricorsive, l'algoritmo termina.
- (Variabile) $b \equiv x$. Anche in questo caso non ci sono chiamate ricorsive dell'algoritmo di sostituzione, quindi la terminazione è ovvia.
- (Applicazione) $b \equiv d(a)$. Supponiamo, per ipotesi induttiva, che l'enunciato valga per le due espressioni d e a, in quando esse sono meno complesse dell'espressione b. Quindi l'algoritmo di sostituzione termina quando viene applicato a $d(a)$.
- (Astrazione) $b \equiv ((x)\ b')$. Vediamo allora i tre casi previsti nell'algoritmo di sostituzione.
  - $x \equiv y$. In questo caso l'applicazione dell'algoritmo di sostituzione richiede che $((x)\ b')[x := e] \equiv ((x)\ b')$. Non essendoci chiamate ricorsive la terminazione è ovvia.
  - $x \not\equiv y$ e $x \notin \mathsf{FV}(e)$. Supponiamo, per ipotesi induttiva, che l'esecuzione dell'algoritmo di sostituzione termini quando esso viene applicato a b' in quanto b' è una espressione meno complessa dell'espressione b. Dobbiamo allora dimostrare che l'algoritmo termina quando viene applicato a $((x)\ b')$. Ma in questo caso $((x)\ b') \equiv ((x)\ b'[y := e])$ e quindi la terminazione è ovvia.
  - $x \not\equiv y$ e $x \in \mathsf{FV}(e)$. Per ipotesi induttiva l'algoritmo di sostituzione termina quando applicato alla espressione b' in quanto essa è meno

complessa della espressione $((x)\ b')$. Inoltre, per il lemma precedente, b' e $b'[x := t]$ hanno la stessa complessità e possiamo quindi riapplicare l'ipotesi induttiva ottenendo che l'esecuzione dell'algoritmo di sostituzione termina anche nel caso in cui esso sia applicato per sostituire l'espressione e al posto della variabile y in $b'[x := t]$. Ma, per definizione dell'algoritmo di sostituzione per questo caso, sappiamo che $((x)\ b')[y := e] \equiv ((t)\ b'[x := t][y := e])$ e quindi ne segue la terminazione della sua esecuzione quando sia applicato a $((x)\ b')$.

2.4 Il processo di calcolo

Come abbiamo più volte ricordato le espressioni che abbiamo definito intendono rappresentare, in modo estremamente astratto, delle funzioni e i loro argomenti. Inoltre l'arietà è il più astratto dei modi per imporre dei vincoli sull'applicazione che ne rendano sensato l'uso. In questa sezione volgiamo vedere come si possano fare dei calcoli con queste funzioni. Introduciamo quindi alcune definizioni.

Definizione 2.7 (β-riduzione) *Sia $b : \beta$ `exp`, $a : \alpha$ `exp` e $x : \alpha$ `var`. Allora diciamo β-*riduzione *la regola seguente:*

$$((x)\ b)(a) \Rightarrow b[x := a]$$

La β-riduzione permette di *semplificare* una espressione della forma $((x)\ b)(a)$ in una espressione di forma più semplice in quanto viene eliminata una istanza di astrazione-applicazione.

Definizione 2.8 (η-riduzione) *Sia $d(x) : \delta$ `exp` una espressione tale che la variabile $x : \alpha$ `var` non appaia in d. Allora diciamo η-*riduzione *la regola seguente:*

$$((x)\ d(x)) \Rightarrow d$$

La η-riduzione è un processo di calcolo che permette di ottenere l'espressione $d : \alpha \to \delta$ astraendo la variabile x dall'espressione $d(x)$. Tale regola è stata introdotto allo scopo di garantirsi la validità nel calcolo delle espressioni della condizione di *estensionalità*: se due espressioni f e g sono tali che, per ogni argomento x della arietà corretta, $f(x) = g(x)$ allora $f = g$. L'estensionalità risulta particolarmente intuitiva nel caso si consideri una nozione insiemistica delle funzioni visto che sostiene che due funzioni con lo stesso grafico coincidono, mentre la sua validità è molto più problematica quando si considerino le funzioni come programmi: in generale è infatti un problema quello di capire quando due programmi sono uguali. Ad ogni modo, per ora non siamo in grado di fornire alcuna dimostrazione formale del fatto che la η-riduzione implica l'estensionalità visto che non abbiamo ancora introdotto una teoria dell'uguaglianza tra le espressioni; lo faremo comunque in uno degli esercizi alla fine della dispensa.

Esempio 6

Ecco un esempio di applicazione della β-riduzione:

$$((x)\ +(x)(3))(5) \Rightarrow +(x)(3)[x := 5] \equiv +(5)(3)$$

Esempio 7

Mentre il seguente è un esempio di applicazione della η-riduzione:

$$((x)\ \sin(x)) \Rightarrow \sin$$

3 Teoria dell'uguaglianza

Nei paragrafi precedenti si è visto che è sorta la necessità di sapere quando due espressioni sono uguali. Dovrebbe essere abbastanza chiaro che l'identità delle scritture è una nozione di uguaglianza troppo stretta visto che è desiderabile che due espressioni che si riducono una all'altra siano uguali, allo stesso modo in cui desideriamo in matematica che il *risultato* dell'applicazione della funzione f all'argomento a sia uguale a $f(a)$ (ad esempio sicuramente vogliamo che $3+2$ sia considerato uguale a 5 anche si si tratta chiaramente di due scritture completamente diverse). Questo è il motivo per cui introduciamo le seguenti regole che caratterizzano la minima relazione di equivalenza tra espressioni indotta dalla $\beta\eta$-riduzioni.

Definizione 3.1 (λ-uguaglianza) *La relazione di uguaglianza tra espressioni è la minima relazione di equivalenza chiusa per le seguenti regole.*

$$\textit{(costante)} \quad \frac{\mathsf{C} : \alpha\ \mathsf{const}}{\mathsf{C} = \mathsf{C} : \alpha}$$

$$\textit{(variabile)} \quad \frac{x : \alpha\ \mathsf{var}}{x = x : \alpha}$$

$$\textit{(applicazione)} \quad \frac{b_1 = b_2 : \alpha \to \beta \qquad a_1 = a_2 : \alpha}{b_1(a_1) = b_2(a_2) : \beta}$$

$$\textit{(ξ-uguaglianza)} \quad \frac{b = d : \beta \qquad x : \alpha\ \mathsf{var}}{((x)\ b) = ((x)\ d) : \alpha \to \beta}$$

$$\textit{(α-uguaglianza)} \quad \frac{b : \beta\ \mathsf{exp} \qquad x : \alpha\ \mathsf{var} \qquad y : \alpha\ \mathsf{var}}{((x)\ b) = ((y)\ b[x := y]) : \alpha \to \beta} \quad y \notin \mathsf{FV}(b)$$

$$\textit{(β-uguaglianza)} \quad \frac{b : \beta\ \mathsf{exp} \qquad a : \alpha\ \mathsf{exp}}{((x)\ b)(a) = b[x := a] : \beta}$$

$$\textit{(η-uguaglianza)} \quad \frac{b : \alpha \to \beta\ \mathsf{exp} \qquad x : \alpha\ \mathsf{var}}{((x)\ b(x)) = b : \alpha \to \beta} \quad x \notin \mathsf{FV}(b)$$

Qualche commento può essere utile allo scopo di capire meglio il significato delle regole di uguaglianza. Le prime quattro regole hanno il solo scopo di garantire che l'uguaglianza rispetta le operazioni che abbiamo deciso di usare per costruire le espressioni, vale a dire costanti, variabili, applicazioni e astrazioni.

La regola di *α-uguaglianza* garantisce che due espressioni che differiscono solo per i nomi delle variabili astratte sono uguali. Abbiamo già detto che desideriamo la validità di questa condizione per evitare che le variabili astrette usate nel definire una funzione nel influenzino il significato. La condizione a lato della regola serve per assicurarsi di non legare con l'astrazione variabili diverse da quella voluta. Si consideri ad esempio la seguente espressione $((x) + (x)(y))$ che denota la funzione che, per ogni argomento c, fornisce $+(c)(y)$. E' chiaro che essa è diversa dalla funzione $((y) + (y)(y))$, che otterremo dalla sostituzione $+(x)(y)[y := x]$. Infatti, quest'ultima è la funzione che per ogni argomento c fornisce $+(c)(c)$.

Le ultime due regole si limitano a garantire che una espressione è comunque uguale ad una espressione a cui è possibile ridurla utilizzando una *β-riduzione* o una *η-riduzione*.

3.1 Decidibilità dell'uguaglianza tra λ-espressioni

Nel seguito sarà per noi essenziale poter decidere quando due espressioni sono uguali. Fortunatamente l'arietà che accompagna le espressioni, ci permette di dimostrare in modo effettivo quando due espressioni sono uguali. Diremo quindi che l'uguaglianza tra espressioni è effettivamente decidibile.

La dimostrazione di questo teorema non è completamente banale e non la inseriremo in queste note. Tuttavia è possibile avere un'idea di come funziona tale dimostrazione dopo aver introdotto la nozione di *forma normale* di una espressione.

Definizione 3.2 (Forma normale) *Una espressione è in* forma normale *se non può essere applicata una β-riduzione a nessuna delle sue sotto-espressioni.*

Data una qualunque espressione e, il seguente algoritmo permette di ottenere una espressione in forma normale, che indicheremo con $\mathsf{nf}(e)$, equivalente ad e.

$$\mathsf{nf}(e) \equiv \begin{cases} ((x)\ \mathsf{nf}(e(x))) & \text{se } e : \alpha \to \beta \text{ } \mathsf{exp} \text{ e } x \\ & \text{è una nuova variabile di arietà } \alpha \\ x(\mathsf{nf}(a_1))\ldots(\mathsf{nf}(a_n)) & \text{se } e \equiv x(a_1)\ldots(a_n) : 0 \text{ } \mathsf{exp} \\ & x : \alpha_1 \to (\ldots(\alpha_n \to 0)\ldots) \text{ } \mathsf{var} \\ & a_1 : \alpha_1 \text{ } \mathsf{exp}, \ldots, a_n : \alpha_n \text{ } \mathsf{exp} \\ \mathsf{C}(\mathsf{nf}(a_1))\ldots(\mathsf{nf}(a_n)) & \text{se } e \equiv \mathsf{C}(a_1)\ldots(a_n) : 0 \text{ } \mathsf{exp} \\ & \mathsf{C} : \alpha_1 \to (\ldots(\alpha_n \to 0)\ldots) \text{ } \mathsf{var} \\ & a_1 : \alpha_1 \text{ } \mathsf{exp}, \ldots, a_n : \alpha_n \text{ } \mathsf{exp} \\ \mathsf{nf}(d[y := a_0](a_1)\ldots(a_n)) & \text{se } e \equiv ((y)\ d)(a_0)(a_1)\ldots(a_n) : 0 \text{ } \mathsf{exp} \\ & d : \alpha_1 \to (\ldots(\alpha_n \to 0)\ldots) \text{ } \mathsf{exp} \\ & a_0 : \alpha_0 \text{ } \mathsf{exp}, \ldots, a_n : \alpha_n \text{ } \mathsf{exp} \end{cases}$$

Per provare la parziale correttezza dell'algoritmo di conversione in forma normale si dimostra il seguente teorema (si noti che per dimostrare che l'algoritmo è totalmente corretto bisognerebbe anche dimostrare che la sua esecuzione termina, ma tale dimostrazione eccede i limiti che ci siamo posti nella stesura di questi appunti).

Teorema 3.3 (Correttezza parziale dell'algoritmo di forma normale) *Sia e una espressione di arietà α. Allora, se l'algoritmo di forma normale termina su e, $\mathsf{nf}(e)$ è una espressione in forma normale uguale ad e.*

Dimostrazione. Prima di tutto si noti che stiamo lavorando nell'ipotesi che l'esecuzione dell'algoritmo termini e quindi esiste un numero m di volte in cui esso deve essere ancora applicato all'espressione e per ottenere la sua forma normale $\mathsf{nf}(e)$. è chiaro che è allora possibile dimostrare una proprietà per induzione su m.

Prima di tutto non è difficile vedere che se l'algoritmo termina quando viene applicato ad e allora il suo risultato, vale a dire $\mathsf{nf}(e)$, è una espressione in forma normale. Consideriamo infatti i vari casi previsti dall'algoritmo:

1. Dobbiamo dimostrare che $((x)\ \mathsf{nf}(e(x)))$ è una espressione in forma normale, ma questo risultato è immediato, visto che se c è una espressione in forma normale allora anche $((x)\ c)$ è in forma normale.

2. Dobbiamo dimostrare che $x(\mathsf{nf}(a_1))\dots(\mathsf{nf}(a_n))$ è una espressione in forma normale, ma anche questo risultato è immediato, visto che se $c_1, \dots, c_n$ sono espressioni in forma normale allora anche $x(c_1)\dots(c_n)$ è in forma normale.

3. Dobbiamo dimostrare che $\mathsf{C}(\mathsf{nf}(a_1))\dots(\mathsf{nf}(a_n))$ è una espressione in forma normale, ma per ottenere questo risultato possiamo procedere esattamente come nel punto precedente.

4. Ovvio.

Notare che una dimostrazione formale richiede di procede per induzione su m.

Anche la dimostrazione che $e = \mathsf{nf}(e)$ si sviluppa per induzione sul numero m. Analizziamo anche in questo caso i vari possibili passi dell'algoritmo di forma normale:

1. Supponiamo che e sia una espressione di arietà $\alpha \to \beta$. Quindi secondo l'algoritmo di normalizzazione $\mathsf{nf}(e) \equiv ((x)\ \mathsf{nf}(e(x)))$. Supponiamo allora per ipotesi induttiva che il risultato valga per l'espressione $e(x)$, vale a dire $e(x) = \mathsf{nf}(e(x)) : \beta$. Quindi, usando una istanza di ξ-*uguaglianza*, otteniamo $((x)\ e(x)) = ((x)\ \mathsf{nf}(e(x)) : \alpha \to \beta$. Ma ora per la η-*uguaglianza*, otteniamo $e = ((x)\ e(x)) : \alpha \to \beta$ (si ricordi che $x : \alpha$ è una variabile nuova e non appare quindi in e). Perciò

$$\mathsf{nf}(e) \equiv ((x)\ \mathsf{nf}(e(x))) = ((x)\ e(x)) = e : \alpha \to \beta$$

2. Supponiamo che $e \equiv x(a_1)\dots(a_n) : 0\ \mathsf{exp}$. Quindi, secondo l'algoritmo di normalizzazione, $\mathsf{nf}(e) \equiv x(\mathsf{nf}(a_1))\dots(\mathsf{nf}(a_n))$. Supponiamo, per ipotesi induttiva, che il risultato valga per le espressioni $a_1, \dots, a_n$. Allora, per ogni $i = 1, \dots, n$, $a_i = \mathsf{nf}(a_i)$. Utilizzando ora l'uguaglianza tra variabili e n volte l'uguaglianza tra applicazioni otteniamo

$$x(a_1)\dots(a_n) = x(\mathsf{nf}(a_1))\dots(\mathsf{nf}(a_n)) : 0$$

quindi

$$\mathsf{nf}(e) \equiv x(\mathsf{nf}(a_1))\dots(\mathsf{nf}(a_n)) = x(a_1)\dots(a_n) : 0$$

3. La dimostrazione del caso in cui $e \equiv \mathsf{C}(a_1)\ldots(a_n) : 0\ \mathsf{exp}$ è completamente analoga a quella del caso precedente.

4. Supponiamo che $e \equiv ((y)\ d)(a_0)(a_1)\ldots(a_n) : 0\ \mathsf{exp}$. Quindi, per l'algoritmo di normalizzazione, $\mathsf{nf}(e) \equiv \mathsf{nf}(d[y := a_0](a_1)\ldots(a_n))$. Supponiamo ora, per ipotesi induttiva, che i risultato valga per l'espressione

$$d[y := a_0](a_1)\ldots(a_n)$$

Allora

$$d[y := a_0](a_1)\ldots(a_n) = \mathsf{nf}(d[y := a_0](a_1)\ldots(a_n)) : 0$$

Ma, per la *β-uguaglianza*,

$$((y)\ d)(a_0)(a_1)\ldots(a_n) = d[y := a_0](a_1)\ldots(a_n) : 0$$

e quindi

$$\begin{array}{rcl}\mathsf{nf}(((y)\ d)(a_0)(a_1)\ldots(a_n)) & \equiv & \mathsf{nf}(d[y := a_0](a_1)\ldots(a_n)) \\ & = & d[y := a_0](a_1)\ldots(a_n) \\ & = & ((y)\ d)(a_0)(a_1)\ldots(a_n)\end{array}$$

Il teorema di forma normale stabilisce una forma *canonica* per le espressioni, cioè la forma più semplice.

La dimostrazione della decidibilità, che procede usando il teorema di forma normale, prova che due espressioni sono uguali se e solo se le loro forme normali, che esistono sempre e sono univocamente determinate, differiscono solo per il nome delle variabili astratte che, come conseguenza della *α-uguaglianza* possono essere sempre scelte liberamente.

Esempio 8

Consideriamo l'espressione $\sin : 0 \to 0\ \mathsf{exp}$. Allora

$$\mathsf{nf}(\sin) \equiv ((x)\ \mathsf{nf}(\sin(x))) \equiv ((x)\ \ \sin(\mathsf{nf}(x))) \equiv ((x)\ \ \sin(x)) : 0 \to 0$$

Esempio 9

Consideriamo l'espressione $+(x) : 0 \to 0\ \mathsf{exp}$. Allora

$$\begin{array}{rcl}\mathsf{nf}(+(x)) & \equiv & ((y)\ \mathsf{nf}(+(x)(y))) \\ & \equiv & ((y)\ \ +(\mathsf{nf}(x))(\mathsf{nf}(y))) \\ & \equiv & ((y)\ \ +(x)(y)) : 0 \to 0\end{array}$$

Esempio 10

Consideriamo l'espressione $\int(3)(5)(\sin) : 0\ \mathsf{exp}$. Allora

$$\mathsf{nf}(\int(3)(5)(\sin)) = \int(\mathsf{nf}(3))(\mathsf{nf}(5))(\mathsf{nf}(\sin)) = \int(3)(5)((x)\ \ \sin(x))$$

Esempio 11

Consideriamo l'espressione $((x)\ +(x)(3))(((x)\ \sin(x))(\pi)) : 0\ \mathsf{exp}$. Allora

$$\begin{array}{rcl} \mathsf{nf}(((x)\ +(x)(3))(((x)\ \sin(x))(\pi))) & \equiv & \mathsf{nf}(+(x)(3)[x := ((x)\ \sin(x))(\pi)]) \\ & \equiv & \mathsf{nf}(+(((x)\ \sin(x))(\pi))(3)) \\ & \equiv & \mathsf{nf}(+(\sin(x)[x := \pi])(3)) \\ & \equiv & \mathsf{nf}(+(\sin(\pi))(3)) \\ & \equiv & +(\mathsf{nf}(\sin(\pi))(\mathsf{nf}(3)) \\ & \equiv & +(\sin(\pi))(3) \end{array}$$

Naturalmente non possiamo spingere oltre il processo di semplificazione perchè la nostra teoria è molto astratta e non sa nulla di aritmetica elementare e trigonometria.

4 Esercizi risolti

Esercizio 1

Siano $c : \beta\ \mathsf{exp}$ e $d : \beta\ \mathsf{exp}$ due espressioni tali che $c = d : \beta$, x una variabile di arietà α e a e b due espressioni di arietà α tali che $a = b : \alpha$. Dimostrare che

$$c[x := a] = d[x := b]$$

Soluzione. Dal fatto che $c = d : \beta$ otteniamo, utilizzando una *ξ-uguaglianza*, $((x)\ c) = ((x)\ d) : \alpha \to \beta$. Quindi, essendo per ipotesi $a = b : \alpha$, per la uguaglianza tra applicazioni otteniamo $((x)\ c)(a) = ((x)\ d)(b) : \beta$. Ma ora, per *$\beta$-uguaglianza*, $((x)\ c)(a) = c[x := a] : \beta$ e $((x)\ d)(b) = d[x := b] : \beta$ e quindi otteniamo la tesi per *transitività*.

Esercizio 2

Siano c e b due espressioni e x e y due variabili. Dimostrare che se $x \notin \mathsf{FV}(b)$ e $x \not\equiv y$ allora

$$c[x := a][y := b] = c[y := b][x := a[y := b]]$$

Soluzione. Per la *β-uguaglianza* abbiamo che $c[x := a] = ((x)\ c)(a)$. Se effettuiamo su ambo i membri la stessa sostituzione $[y := b]$ otteniamo (vedi esercizio precedente) $c[x := a][y := b] = ((x)\ c)(a)[y := b]$. Ora, per la definizione di sostituzione nelle ipotesi dell'esercizio, $((x)\ c)(a)[y := b] \equiv ((x)\ c[y := b])(a[y := b])$. Ma, per la *beta-uguaglianza*, $((x)\ c[y := b])(a[y := b]) = c[y := b][x := a[y := b]]$ e quindi otteniamo la tesi per *transitività*.

Esercizio 3

Dimostrare che la relazione di uguaglianza tra espressioni è estensionale, vale a dire che, supponendo $f : \alpha \to \beta\ \mathsf{exp}$ e $g : \alpha \to \beta\ \mathsf{exp}$ siano due espressione e x sia una variabile di arietà α che non compare ne in f ne in g, se $f(x) = g(x) : \beta$ allora $f = g : \alpha \to \beta$.

Soluzione. Visto che per ipotesi $f(x) = g(x) : \beta$, per la *ξ-uguaglianza*, otteniamo $((x)\ f(x)) = ((x)\ g(x)) : \alpha \to \beta$. Ma ora, per la *$\eta$-uguaglianza* che possiamo applicare visto che $x \notin \mathsf{FV}(f)$ e $x \notin \mathsf{FV}(g)$, otteniamo $((x)\ f(x)) = f : \alpha \to \beta$ e $((x)\ g(x)) = g : \alpha \to \beta$ e quindi la tesi per *transitività*.