

Indice

1	Introduzione	3
1.1	Stato dell'arte	3
1.2	ELPI	4
1.3	Haskell	4
2	Implementazione	5
3	Conclusioni	7

Capitolo 1

Introduzione

1.1 Stato dell'arte

Ho implementato l'algoritmo di type inference di Haskell in ELPI.

Haskell è un linguaggio di programmazione che adotta il paradigma di programmazione funzionale.

La type inference è

ELPI è un linguaggio di programmazione logico. Esso è l'espansione di λ Prolog, il quale a sua volta è l'espansione di Prolog.

Prolog è un linguaggio di programmazione che adotta il paradigma di programmazione logica. Si basa sul calcolo dei predicati (logica del prim'ordine); la sintassi è composta da formule dette clausole che sono disgiunzioni di letterali del prim'ordine. Eseguire un programma Prolog può essere comparato a dimostrare un teorema mediante la regola di inferenza detta risoluzione. I concetti fondamentali di questo linguaggio sono l'unificazione, la ricorsione in coda e il backtracking.

λ Prolog è, come già detto, l'espansione di Prolog.

Essendo λ Prolog un linguaggio che non utilizza i vincoli risulta impossibile implementare la type inference; infatti tale limite si riscontra in particolare nel tentare di codificare il tipaggio per i costrutti del let-in e delle type class. L'unica strategia attuabile sarebbe quella di codificare interamente il sistema

punto per punto, il che evidentemente rende tale strategia impraticabile. È stato dunque necessario l'utilizzo di ELPI, la cui maggiore espressività permette di svolgere operazioni impossibili da codificare in λ Prolog.

1.2 ELPI

1.3 Haskell

Capitolo 2

Implementazione

Questo è il primo capitolo.

Capitolo 3

Conclusioni

Queste sono le conclusioni.

In queste conclusioni voglio fare un riferimento alla bibliografia: questo è il mio riferimento [?, ?].

Ringraziamenti

Qui possiamo ringraziare il mondo intero!!!!!!!!!!
Ovviamente solo se uno vuole, non è obbligatorio.