HELP - Linguaggi 2019

Fabio Chiarani

January 29, 2019

1 Alcune cose...

Scope: Lo scope di una variabile è il range di comandi nella quale è visibile. Le regole di scope determinano come i riferimenti ad un nome sono associati alle variabili.
Lo scope è *statico* se il nome non locale è risolto nel blocco che lo racchiude.
Lo scope è *dinamico* se + risolto nel blocco attivato più di recente, e NON ancora disattivato.

Dichiarazioni: sono una categoria sintattica nelle quali gli elementi sono elaborati per produrre legami.
Espressioni: sono una categoria sintattica dove le espressioni vengono valutate per ottenere valori esprimibili *Eval*.
Comandi: sono una categoria sintattica nella quale gli elementi sono eseguiti per aggiornare la memoria della macchina astratta che supporta il linguaggio.

Passaggio per valore: il parametro attuale viene utilizzato per inizializzare il valore del formare. Le modifiche del formale non passano all'attuale. Può essere implementato per copia.
Passaggio per risultato: nessun valore è trasmesso al sottoprogramma. Il valore del parametro formale viene copiato nell'attuale solo alla fine del sottoprogramma.
Passaggio per valore-risultato: sopra uniti
Passaggio per riferimento: passa per un cammino d'accesso. Non c'è copia o duplicazione, l'accesso è rallentato per l'indirizzamento indiretto. Crea alias che può causare side-effect.
Passaggio per nome: esegue la sostituzione testuale. I parametri fomrale sono legati al metodo d'accesso al momento della chiamata, i parametri attuali sono legati al valore durante il loro accesso. Permette il late binding e porta al conflitto dei nomi.

2 Domande e Risposte

Q: Cosa significa implementare un linguaggio L? Definire esplicitamente e dettagliatamente tutti i concetti utilizzati.
A: Un linguaggio di programmazione è un insieme di costrutti e regole per descrivere algoritmi e dati. Implementare un linguaggio significa costrutire una macchina ML che ne esegua i programmi $P \in PROG^L$. Questa macchina dovrà girare su una macchina ospite M_0 avente un linguaggio L_0. Ci sono due possibili approci: tramite interprete o tramite compilatore.

Interprete: Il primo approcio è la traduzione implicita o *interpretazione*. Un interprete $INT^{L,L_0} : (PROG^L \times D) \to D$ è un programma scritto in L_0 tale che per ogni input $d \in D$ si ha: $INT(P, d) = P(d)$ per ogni $P \in PROG^L$

Compilatore: Il secondo approcio è la traduzione esplicita o *compilazione*. Traduco un programma scritto in L in un programma scritto in L_0. Un compilatore è un programma che realizza la funzione $COMP^{L,L_0} : PROG^L \to PROG_0^L$ tale che per ogni input $d \in D$ si ha: $COMP(P)(d) = P(d)$ per ogni $P \in PROG^L$

Confronto: L'interprete è lento, ma è facile da implementare e è portabile. La compilazione rende l'esecuzione più veloce dell'interpretazione, ma è più difficile da realizzare per la distanza dei linguaggi. Inoltre si perde la struttura del programma originale.

Q: Descrivere intuitivamente cosa è una dichiarazione. Definire formalmente le dichiarazioni di IMP e dare semantica operazionale statica e dinamica alla dichiarazione di procedura.
A: Le dichiarazioni sono una categoria sintattica i quali elementi sono elaborati per produrre associazioni tra un identificatore e un tipo denotabile.

Q: Definire cosa è un interprete: dare definizione semantica e descrivere la struttura.
A: Un interprete per programmi di inguaggio S permette di eseguire tali programmi su macchine astratte il cui linguaggio è L. Consideriamo due linguaggi turing completi L, S. E' garantita l'esistenza dell'interprete, cioè di un programma $int \in L$ tale che per ogni programma $P \in S$ e per ogni input $d \in D$ si ha: $[[int]]^L(P, d) = [[P]]^S(d)$

Q: Definire cosa è il passaggio di parametri, dando in particolare la definizione di passaggio per valore e per riferimento.

A: Il passaggio di parametri è il binding tra un parametro locale e il valore denotato ottenuto dalla valutazione dell'argomento attuale. **Passaggio per valore**: il valore del parametro attuale viene utilizzato per inizializzare il valore del formale. Le modifiche del formale non passano all'attuale. Può essere implementato tramite copia o cammino d'accesso. **Passaggio per riferimento**: passa in cammino d'accesso, non c'è copia o duplicazione, ma l'accesso è rallentato dall'indirizzamento indiretto. Crea alias che causano side-effects.

Q: Definire intuitivamente e formalmente il concetto di specializzazione e specializzatore.
A: La specializzazione è la valutazione parziale di un programma su un input fissato. Formalmente, se L, S, T sono linguaggi turing completi sull'insieme di input D allora un programma $spec \in L$ è uno specializzatore se per ogni ...formula impossibile...

Q: Definire cosa è una macchina astratta: dare la struttura e la definizione di linguaggio macchina
A: Il supporto di un linguaggio è la macchina astratta. Una macchina astratta di un linguaggio L è un insieme di strutture dati ed algoritmi per memorizare ede eseguire programmi L. Viene denotata con M_L. Data la macchina astratta M definiamo ML linguaggio macchina, il linguaggio che ha come elementi tutte le stringhe interpretabili da M. Una macchina astratta è formata da memoria che immagazzina dati e programmi, ed interprete che esegue le istruzioni dei programmi. Esistono dipologie di operazioni e modalità di esecuzione indipendenti dal linguaggio (elaborazione dati primitivi, controllo sequenza di esecuzione, controllo dei dati, controllo della memoria).

Q: Definire cosa sono le politiche di binding. Spiegare in particolare la differenza tra shallow e depep, facendo attenzione a come si combinano con lo scoping statico e/o dinamico.
A: Le politiche di binding intervengono solo quando una procedure è passata come parametro ad un altra procedura. Il binding è deep se vale l'ambiente alla creazione del legame $h \to f$, oppure shallow se vale l'ambiente alla chiamata di f attraverso h. Con scoping statico si utilizza sempre il deep.

Q: Definire il concetto di ricorsione.
A: La ricorsione è un modo alternativo all'iterazione per ottenre il potere espressivo di una MdT. Una funzione è ricorsiva se è definita in termini di se stessa.