

Corso: Fondamenti, Linguaggi e Traduttori

Paola Giannini

Introduzione al compilatore da **ac** a **dc**

Programma Sorgente ac

```
int tempa;  
tempa = 5;  
float tempb = tempa + 3.2;  
tempb = tempb + 7;  
print tempb;
```

Codice dc

```
5 sa 0 k  
la 5 k 3.2 + sb 0 k  
lb 7 5 k + sb 0 k  
lb p P
```

Definizione informale di ac

In ac ci sono

- 2 **tipi** di dato interi e floating point. Un letterale
 - **intero** è una sequenza di cifre;
 - **floating point** è una sequenza (non vuota) di cifre seguita da “.” seguita da almeno 1 cifra e al più 5 cifre; se volete usare usare una definizione più restrittiva fatelo pure!
- **variabili** che sono stringhe contenenti solo i 26 caratteri dell'alfabeto inglese minuscoli. Una variabile deve essere dichiarata prima di poter essere usata (in una espressione);
- **dichiarazioni**: **float** o **int** seguiti da una variabile
- 2 **istruzioni**: *assegnamento* e *stampa*.
 - sintassi dell'assegnamento: variabile “=” espressione
 - sintassi della stampa: **print** variabile
- Le **espressioni** possono essere letterali (interi o floating point), variabili oppure somma e sottrazione di espressioni.
- Una espressione di tipo **int** può essere **convertita automaticamente** a **float** (se necessario) e nessun altra conversione è possibile.

Il linguaggio dc

- Il linguaggio target dc è un calcolatore “stack based” (cioè usa la **notazione polacca inversa**) cross-platform. È una delle prime applicazioni scritte per Unix.
- Si usa, da terminale, digitando **dc** e poi a capo. A questo punto si scrivono le espressioni da valutare. Per stampare si digita **p** (e **q** per uscire).
- Gli operatori sono: **+** (addizione) e **-** (sottrazione) e molti altri.....
- Si può specificare la precisione delle operazioni, dicendo quante cifre decimali considerare. Il default e' 0. Per cui, **5 4 /** valuta a **0**.
- Vedere: https://www.gnu.org/software/bc/manual/dc-1.05/html_mono/dc.html

Costruzione del compilatore

Definiremo le classi necessarie per fare:

- l'**Analisi Lessicale** di ac
- l'**Analisi Sintattica** di ac e la costruzione dell' **Abstract Syntax Tree** (il nostro sarà un compilatore in 2 fasi)
- e poi quelle per fare
 - il **Type Checking**, cioè l'analisi semantica del AST e
 - la **Generazione del Codice**, anche questa a partire dal AST.