Elaborato di Fondamenti di Informatica 2006/07

Claudio Guidi
cguidi@cs.unibo.it

Perché e quando.

- Consente di comprendere come i risultati teorici della teoria dei linguaggi formali possano avere un impatto importante dal punto di vista implementativo dei compilatori/interpreti di linguaggi.
- Obbligatorio per superare l'esame
- Consegna libera durante la sessione. La scadenza per ciascuna sessione verrà esposta sulla homepage: http://www.cs.unibo.it/~cguidi
- Gli elaborati verranno corretti solo dopo la scadenza della consegna. In casi eccezionali può essere richiesto, previo accordo via e-mail, di effettuare la correzione prima di tale data.

Consegna.

- L’elaborato deve essere consegnato via email all’indirizzo cguidi@cs.unibo.it con oggetto “IT - consegna progetto 2006/07 ”
- Il testo della mail deve contenere nome, cognome e matricola dello studente.
- In allegato il file .zip (altri formati non verranno accettati) del progetto con:
  - Documentazione
  - Sorgenti progetto
  - File di esempio

Come e dove.

- Ogni studente dovrà sviluppare un elaborato
- Non sono consentiti elaborati di gruppo
- L'elaborato deve essere sviluppato in linguaggio Java.
- L'elaborato verrà corretto in laboratorio
  - L'elaborato deve essere compatibile con la versione della JVM correntemente installata in laboratorio
- L'elaborato deve essere funzionante
- Il giorno dell'elaborato deve essere consegnata una stampa cartacea della documentazione.

Documentazione.

- Deve essere fornita in due formati:
  - Formato html (a scelta può essere formato da una singola pagina oppure da più pagine navigabili)
  - Formato pdf
  - **Il giorno della discussione dell'elaborato deve essere consegnata una stampa cartacea della documentazione**
- Deve riportare nome, cognome e matricola dello studente
- Deve essere suddivisa in quattro sezioni
  - Linguaggio
  - Scanner
  - Parser
  - Interprete

Documentazione: Linguaggio

- **NB: In grassetto sono riportati i titoli delle sottosezioni**
- **Introduzione:** descrizione generica del linguaggio sviluppato illustrando velocemente i vari costrutti linguistici utilizzati
- **La grammatica:** deve riportare le produzioni della grammatica in formato BNF. Commentare le produzioni della grammatica che si ritengono più significative.
- **Esempi:** riportare almeno tre esempi di codice. Nel complesso dei tre listati si deve fare utilizzo di tutti i costrutti presenti nella grammatica. Ogni esempio deve essere corredato da una breve descrizione che ne illustra il funzionamento.
- **Osservazioni** (non obbligatoria): qualora necessario, utilizzare questa sezione per eventuali commenti od osservazioni legate allo sviluppo del linguaggio.

Documentazione: Scanner

- **Introduzione:** descrizione generica del codice che implementa lo scanner
- **I token:** deve riportare le espressioni regolari dei token utilizzati
- **Il codice**: breve descrizione di ciascuna classe utilizzata
- **Osservazioni** (non obbligatoria): qualora necessario, utilizzare questa sezione per eventuali commenti od osservazioni legate all'implementazione dello scanner

Documentazione: Parser

- **Introduzione:** descrizione generica del codice che implementa il parser, quale tipologia di parser si è scelto, motivazioni.
- **La grammatica:** deve riportare le produzioni della grammatica in formato BNF. Commentare le produzioni della grammatica che si ritengono più significative.
- **Il codice**: breve descrizione di ciascuna classe utilizzata
- **Osservazioni** (non obbligatoria): qualora necessario, utilizzare questa sezione per eventuali commenti od osservazioni legate all'implementazione del parser

Documentazione: L'interprete

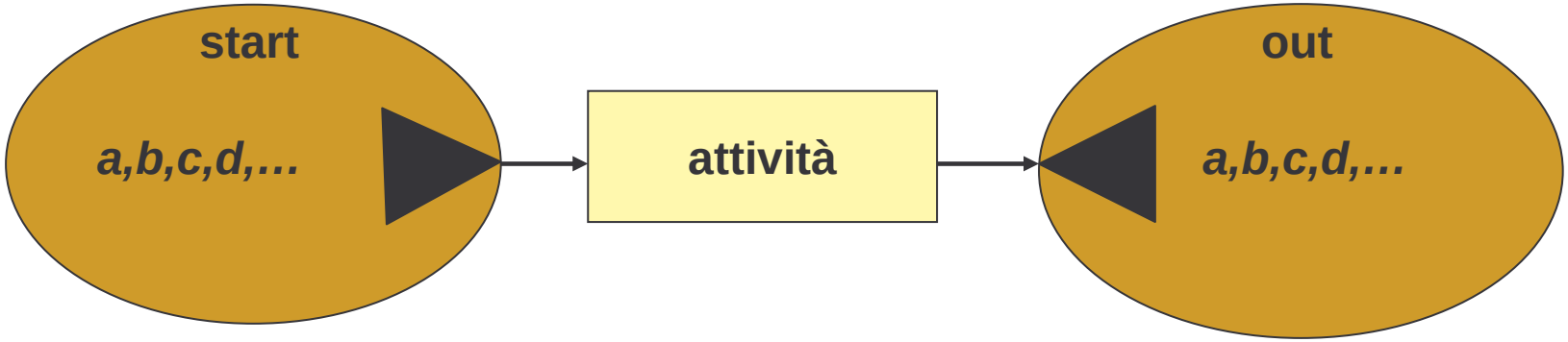
- **Introduzione:** descrizione generica del codice che implementa l'interprete
- **Architettura e strutture dati:** descrivere l'architettura dell'interprete e delle strutture dati (utilizzare una rappresentazione grafica qualora necessario)
- **Esecuzione:** descrivere tutti i passi necessari all'utente al fine di poter mettere in esecuzione l'interprete (linea di comando, eventuali parametri, ecc.)
- **Esempi:** Riportare almeno **tre** esempi di codice in cui si utilizza il costrutto di parallelo e la cui esecuzione porta a risultati diversi per alcune delle variabili utilizzate. Ogni esempio deve essere corredato di una breve descrizione circa il suo funzionamento. Tali esempi dovranno essere contenuti anche nel file di esempio e dovranno essere eseguiti durante la discussione dell'elaborato.
- **Il codice**: breve descrizione di ciascuna classe utilizzata
- **Osservazioni** (non obbligatoria): qualora necessario…

Il progetto.

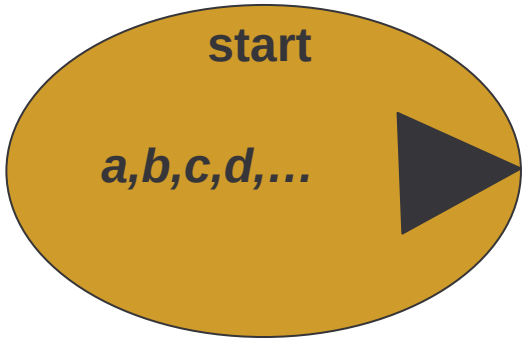
- Il progetto consiste nello sviluppo di un linguaggio in grado di rappresentare tutti i possibili diagrammi ottenibili componendo i moduli grafici descritti più avanti. Per ciascun modulo deve essere fornito un corrispondente costrutto linguistico
- Di tale linguaggio si deve sviluppare uno scanner ed un parser che ne riconoscano la grammatica
- Si deve sviluppare un interprete che dia rappresentazione del codice inserito in input mediante transition system.
  - Delle variabili specificate come output devono essere riportati tutti i valori ottenuti nei diversi rami di esecuzione del transition system.

Regole di costruzione dei diagrammi

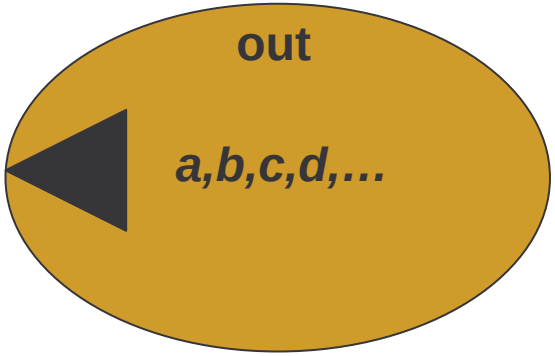
- Un diagramma ha la seguente struttura:



Descrizione blocchi principali



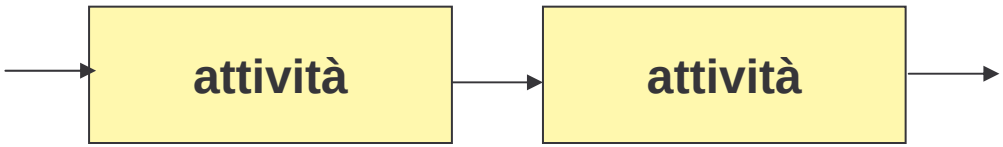
Blocco di start: un diagramma di flusso inizia sempre con questo blocco. ***a,b,c,…*** sono le variabili che verranno utilizzate all'interno del diagramma



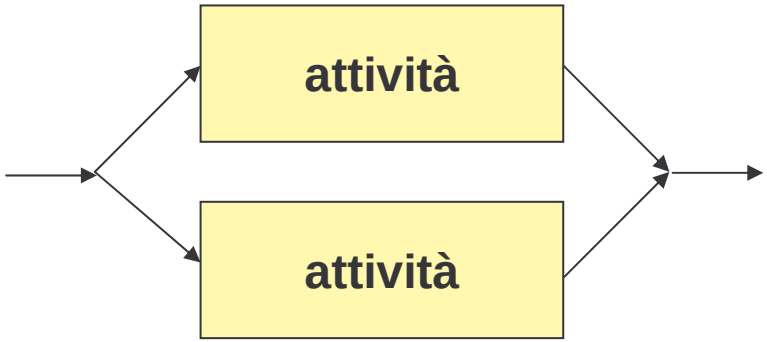
Blocco di out: in questo blocco il diagramma termina. ***a,b,c*****,…** sono le variabili delle quali si deve riportare il valore finale.

Blocco attività: può essere un blocco di composizione o un'attività base.

Blocchi di composizione

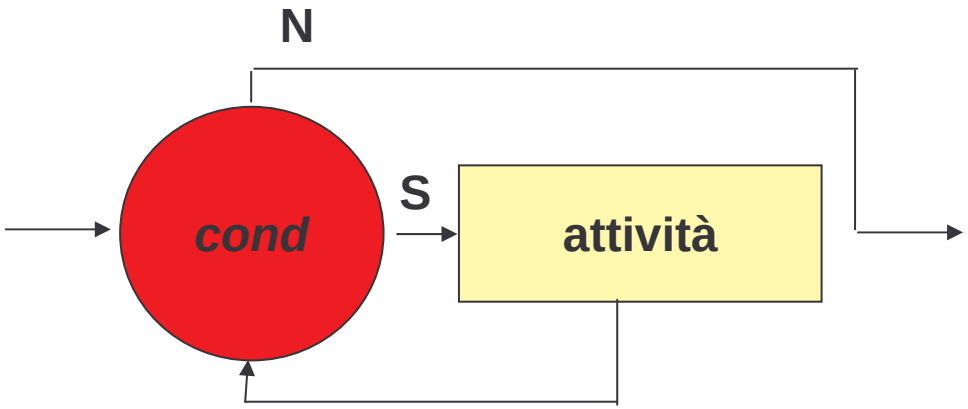


Sequenza

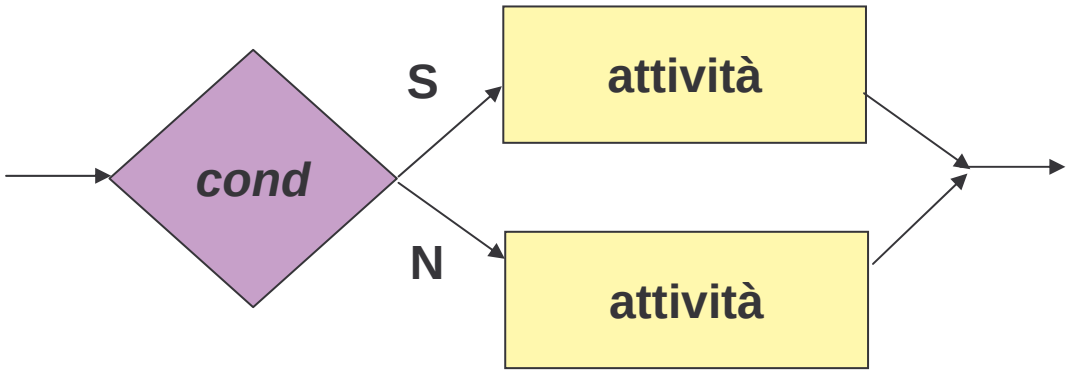


Parallelo

Blocchi di composizione (2)

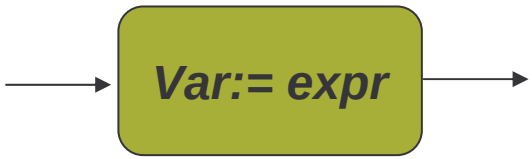


Ciclo: ripete l'attività posta sul ramo **S** se la condizione ***cond*** è vera



Scelta: dove ***cond*** è una condizione sulle variabili
S condizione verificata,
N condizione non verificata.

Blocco base



Assegnamento: ***Var*** è una variabile

expr è un'espressione aritmetica (+,-,*,/) costituita da numeri interi e variabili

Nota

- Il blocco che descrive il ciclo è potenzialmente in grado di esprimere cicli infiniti dipendentemente dalla formulazione della condizione logica e dallo stato delle variabili
- La trattazione di transition system con cicli infiniti va al di fuori del presente elaborato
- Per ovviare a tale problema si inserisca un parametro di progetto dell'interprete (Nmax) che rappresenta il numero massimo di possibili cicli eseguibili oltre il quale il ciclo termina comunque.