

PRIMA ESERCITAZIONE

Laboratorio di
Linguaggi e computabilità

PRIMA ESERCITAZIONE

- Scaricare la cartella «Materiale primo laboratorio»
- Creare 3 cartelle e inserirvi i file scaricati
 1. subst.l e inputFile.txt
 2. group.y
 3. group2.l e group2.y
- Guardare con attenzione i file e osservare quanto riportato di seguito
- N.B. Sono file di testo, qualsiasi editor di testo va bene per editarli (es. notepad++)

SUBST

[non serve yacc, ma solo jflex]

- L'estensione **.l** è equivalente all'estensione **.flex**
- Il file **subst.l** contiene i pattern/regole con cui cerca match nel testo che si analizzerà, e il metodo di **yytext()**, usato nella parte "azione", restituisce la stringa su cui una certa regola trova match
- La dichiarazione **%standalone** genera un **main()** che rende la classe generata compilabile ed eseguibile
- Il file specifica il nome della classe Java generata ("Subst")
- Il file dichiara e poi usa una **variabile in Java**: "name"
- Dopo aver eseguito **jflex subst.l** guardare con attenzione il file generato (**Subst.java**)
- "java Subst" prende (per default) in input un file di testo
- Il lavoro di Subst è
 - estrarre e memorizzare il valore per la variabile "name" quando trova nel proprio input del testo come "name xyz". In questo caso "xyz" diventa il valore memorizzato
 - se nel testo di input trova "hello" o "Hello", stampa il valore di "name" (e un messaggio)
- Creare un file di testo (es. **inputFile.txt**) e inserirvi un insieme di righe che producano (o meno) match
- Compilare con **"javac Subst.java"**
- Eseguire con **"java Subst inputFile.txt"** e verificare il comportamento

GROUP

[non serve jflex, ma solo yacc]

- Il file elabora come **input** il contenuto della variabile "String in" presente nel file (è accettato in input qualunque sequenza di + e –)
- Osservare con attenzione i due metodi (obbligatori per yacc): **yyerror()** e **yylex()**
- Stampa l'input mano a mano che lo analizza, riconoscendo i – in gruppi e i + uno a uno
- Eseguire byacc/j con il comando **yacc -J group.y**
- Compilare ed eseguire l'output generato (provando a modificare il contenuto della variabile di input)

Esercizio: modificare il file group.y per rendere la gestione dei + uguale a quella dei –

GROUP2

[usare sia jflex che yacc]

- prende input da tastiera
- il lexer accetta anche caratteri diversi da + o – e li “butta” (si veda la riga "[^] {}" nel file .l)
- N.B.: gli 'a capo' sono caratteri che il sistema analizza come tutti gli altri e che quindi vengono “buttati”

Esercizio: aggiungere il trattamento di un terzo carattere oltre + e –

INFDYCK

[usare sia jflex che yacc]

- Useremo questi file come punto di partenza per gli esercizi della prossima esercitazione, guardare nel dettaglio il loro contenuto e provarne il comportamento
- **Comportamento:** sono accettate righe con parentesi bilanciate, alternate a lettere minuscole
- Osservare in particolare
 - lexer che ritorna in "yylval" la stringa con match, e il token corrispondente con return
 - la corrispondenza tra i token: definiti in .y e usati in .l