

Linguaggi Formali e Compilatori

Prof. L. Breveglieri e S. Crespi Reghizzi

Prova scritta 15/07/2005 - Parte I: Teoria

COGNOME E NOME:.....

MATRICOLA:.....FIRMA:.....

ISTRUZIONI:

- L'esame si compone di due parti:
 - I (80%) Teoria:
 1. espressioni reg. e automi finiti
 2. grammatiche
 3. analisi sintattica
 4. traduzione e semantica
 - II (20%) Esercitazioni Flex e Bison
- Per superare l'esame l'allievo deve superare entrambe le parti (I e II) nello stesso appello oppure in appelli diversi della stessa sessione d'esame.
- Per superare la parte I (teoria) occorre dimostrare sufficiente conoscenza di tutte le quattro sottoparti (1-4).
- Tempo: Parte I (esercitazioni): 30 min - Parte II (teoria): 2h:30min
- È permesso consultare libri e appunti personali.

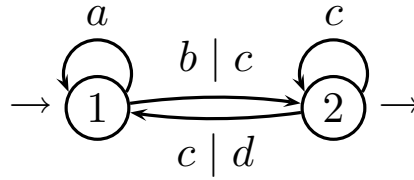
1 Espressioni regolari e automi finiti 20%

1. Dato il linguaggio di alfabeto $\Sigma = \{a, b, c\}$:

$$L = a\Sigma^*c - \Sigma^* \{ab, ac\} \Sigma^*$$

- (a) Si elenchino la (o le) frasi di lunghezza ≤ 4 .
- (b) Si costruisca l'automa deterministico che riconosce L . Si commenti il procedimento seguito.
- (c) (facoltativo) Si minimizzi l'automa così calcolato.

2. È dato l'automa M seguente:



- (a) Si ricavi l'espressione regolare non ambigua che genera $L(M)$.
- (b) Si costruisca l'automa (nondeterministico o deterministico, a scelta) che riconosce il sottolinguaggio

$$L_1 = (L(M) \cap \Sigma^* c \Sigma^*)^R$$

Si noti che L_1 è la riflessione di tutte le stringhe contenenti almeno un carattere c . Si commenti il procedimento seguito.

2 Grammatiche 20%

1. Dato l'alfabeto $\Sigma = \{a, b, c, d\}$, trovare la grammatica che genera il linguaggio di palindromi $ucv dv^R cu^R$, dove $u, v \in \{a, b\}^*$, e i due vincoli seguenti sono entrambi soddisfatti: u contiene un numero pari di a e v un numero dispari di b . Tracciare anche l'albero sintattico della seconda stringa tra gli esempi dati sotto.

Esempi: *cdbdc, abacdbdcaba, aabbcbbabdbabbcaabb*

Contresempi: *acbbdbbbca, aacbdbbbca*

2. Costruire la grammatica EBNF non ambigua che genera programmi scritti nel linguaggio di programmazione semplificato seguente:

- (a) Le due costanti booleane 0 e 1 (che rappresentano falso e vero).
- (b) Identificatori di variabile ed etichette, sia gli uni sia le altre formati da una sequenza di lettere minuscole $[a-z]$, di cifre $[0-9]$ e dal carattere “underscore” $_$, e iniziati con una lettera minuscola ma non terminanti con “underscore”.
- (c) Espressioni booleane, anche con parentesi tonde, formate da identificatori di variabile e costanti, e dagli operatori logici $|$ (or), $\&$ (and) e $!$ (not); le precedenze sono: $!$ precede $\&$ precede $|$.
- (d) Assegnamenti, del tipo:

`a0 := 1 & (b_12 | ! c);`

- (e) Cicli del tipo mostrato nell’esempio conclusivo dato sotto, dove il corpo del ciclo può contenere, in un ordine qualsiasi:
 - nessuno, uno o più assegnamenti;
 - nessuna, una o più clausole del tipo seguente:

`break ((a0 | ! c) & (! d_17a | 1));`
 - nessuno, uno o più altri cicli annidati
- (f) Un’istruzione è un assegnamento oppure un ciclo.
- (g) Infine, un programma è una sequenza qualunque di istruzioni, non vuota.

Esempio complessivo:

```
a0 := 1 & (b_12 | ! c);
perenne loop
  a0 := b_12;
  break (a0 | ! c);
  c := b_12 & a0;
end perenne;
...
```

Quali aspetti semantici del programma non sono modellabili mediante la grammatica proposta?

3. (facoltativo) Dato l'alfabeto $\Sigma = \{a, b\}$, costruire l'automa a pila deterministico che riconosce il linguaggio seguente:

$$L = \{w \mid 2|w|_a = 3|w|_b\}$$

Nota bene: l'ordine di a e b nella stringa w non ha importanza, contano i numeri di a e b .

Esempi: ε , $aaabb$, $ababa$, $abaabbabaa$

Contresempi: aab , $abbb$

3 Grammatiche e analisi sintattica 20%

1. La grammatica EBNF seguente definisce una lista di assegnamenti:

$$S \rightarrow A^+$$

$$A \rightarrow T = E$$

$$T \rightarrow v \mid v \text{ ' (' } E \text{ (, } E \text{) }^* \text{ ') '}$$

$$E \rightarrow T \text{ (+ } T \text{) }^*$$

Si noti che non vi è il punto e virgola tra gli assegnamenti.

Si progetti una grammatica BNF equivalente alla precedente in modo che essa risulti $LL(k)$, per $k \geq 1$. Si calcolino gli insiemi guida delle regole.

2. Per la grammatica:

$$S \rightarrow aXb$$

$$S \rightarrow Ab$$

$$X \rightarrow aXb$$

$$X \rightarrow \varepsilon$$

$$A \rightarrow aA$$

$$A \rightarrow a$$

- (a) Costruire il riconoscitore deterministico dei prefissi ascendenti e verificare se la grammatica sia LR(1).
- (b) Se la grammatica non fosse LR(1), si individui la causa del conflitto e si scriva una grammatica equivalente LR(1).

4 Traduzione e semantica 20%

1. Il linguaggio sorgente L_s contiene certe espressioni aritmetiche in scrittura postfissa, con operatori associativi che ammettono $n \geq 2$ argomenti. Esso è definito dalle regole seguenti:

$$S \rightarrow ' (' S, S L ') ' \text{ add}$$

$$S \rightarrow [1 - 9]$$

$$L \rightarrow , S L$$

$$L \rightarrow \varepsilon$$

- (a) Si progetti, modificando se necessario la grammatica sorgente data sopra, uno schema di traduzione (semplice, senza attributi semantici) per convertire una frase di L_s nella corrispondente stringa in scrittura prefissa, con operatori binari associativi a destra.

Esempi:

frase sorgente	frase pozzo
$(3, 6) \text{ add}$	$+3\ 6$
$(3, 5, 4) \text{ add}$	$+3\ +\ 5\ 4$
$((3, 5, 4) \text{ add}, 7\ 9\ 2) \text{ add}$	$+ +\ 3\ +\ 5\ 4\ +\ 7\ +\ 9\ 2$

- (b) Si disegnino gli alberi sintattici sorgente e pozzo dell'ultimo esempio dato sopra.

2. Sono definite delle espressioni aritmetiche con gli operatori $+$, \times , la variabile v e le parentesi, per mezzo della sintassi seguente:

$$S \rightarrow S + S$$

$$S \rightarrow S \times S$$

$$S \rightarrow '(S)'$$

$$S \rightarrow v$$

- (a) Progettare le funzioni semantiche di una grammatica ad attributi per eliminare le parentesi superflue dall'espressione.

Esempi:

sorgente		pozzo
$v + (v + v)$	\Rightarrow	$v + v + v$
$v \times (v)$	\Rightarrow	$v \times v$
$(v + v) \times v$	\Rightarrow	$(v + v) \times v$
$(v \times v) + v$	\Rightarrow	$v \times v + v$
$((v + v)) \times v$	\Rightarrow	$(v + v) \times v$

Data una forma di frase del tipo (S) , le parentesi tonde risultano superflue se vale una delle due condizioni seguenti:

- S è una variabile, un prodotto o un'espressione tra parentesi
- (S) non è né preceduta né seguita da \times

Si usi l'attributo seguente, aggiungendone altri se necessario: " τ of S ", di tipo stringa e sintetizzato; è la traduzione da produrre.

- (b) Disegnare l'albero dell'ultimo esempio decorato con i valori degli attributi e le frecce delle dipendenze funzionali.
- (c) Verificare se la valutazione degli attributi possa essere svolta con il metodo L e scrivere, almeno in parte, lo pseudocodice del valutatore semantico, supponendo che l'albero sintattico sia stato già costruito dal parsificatore.

Sintassi	Funzioni semantiche
$S \rightarrow S + S$	
$S \rightarrow S \times S$	
$S \rightarrow '(S)'$	
$S \rightarrow v$	

