

~~A~~
~~Finali $\{A, B, D\}$~~

N_{11} : Sia N_{11} la NFA con stato iniziale A, insieme di stati finali $\{A, B, D\}$ e con la seguente funzione di transizione

| | ϵ | a | b |
|---|-------------|-------------|-------------|
| A | \emptyset | $\{A, D\}$ | $\{B\}$ |
| B | $\{A, C\}$ | $\{B, D\}$ | $\{C\}$ |
| C | $\{A, B\}$ | \emptyset | \emptyset |
| D | \emptyset | $\{A\}$ | $\{A\}$ |

D_{11} : Sia D_{11} il DFA con stato iniziale A, insieme di stati finali $\{A, B, C\}$ e con la seguente funzione di transizione

| | a | b |
|---|---|---|
| A | D | B |
| B | E | |
| C | | F |
| D | F | E |
| E | B | |
| F | F | C |

G_{11} :
 $S \rightarrow aSbS \mid bSaS \mid CA \mid \epsilon$
 $A \rightarrow CA \mid \epsilon$

$f_{1.1}$: Sia $f_{1.1}$ il file per analizzatore lexico di cui sono riportate nel seguito le sezioni delle dichiarazioni delle regole:

```
digit [0-9]
letters [a-z, A-Z]
other [\\ * \? " <"]
id ({other?} | {digit?} | {letters?}) +
%%
({other?} ? ({other?} ? {id} ({digit?}) + ({id?} ? {printf("SI ")});
{id?} {printf("FORSE ")};
{printf("NO ")};
%%
```

$f_{2.1}$: Sia $f_{2.1}$ il file per analizzatore lexico di cui è riportata nel seguito la sezione delle regole:

```
%%
"a" {return a;}
"b" {return b;}
[0-9] + {yyval = atoi(yytext); return unum;}
[-+\\u] return *yytext;
[\\t] ;
{yyerror("invalid character");}
%%
```


f2.y Sia f2.y il file per analizzatore sintattico di cui sono riportate nel seguito le sezioni delle dichiarazioni e delle regole:

```
% Token a b wwe
%%
S:
    S E '\u' { printf( "%d \u", $2); }
    ;
E:
    wwe { $$ = $1; }
    | E a E { $$ = $1 * $3; }
    | E b E { $$ = $1 + $3; }
    ;
%< PUNTO E VIRGOLA NON J! %>
```

f3.y: Sia f3.y il file per analizzatore sintattico di cui sono riportate nel seguito dichiarazioni e regole:

```
% Token a b c d e
%%
S: a A d
    | a B c
    | b a A c
    | b a B d
    | c A d
    | c B c
    ;
A: c e
    ;
B: c C
    ;
C: e D
    ;
D:
    ;
%%
```

- 1) Chiamiamo A l'automato LR(0) per G_M , H lo stato iniziale di A, H_{aa} lo stato di A che si raggiunge da H tramite il cammino aa, T la tabella di parsing LALR(1) per G_M . Se non ci sono violazioni nello stato H_{aa} di T, rispondere "No". Altrimenti dire quali entry dello stato H_{aa} di T contengano violazioni e specificare di che violazioni si tratta.
- 2) Si consideri la seguente affermazione: "la grammatica G con produzioni $S \rightarrow aS | bS | a | b | \epsilon$ è ambigua perché il non terminale S ha due produzioni che iniziano con il simbolo a ." Se l'affermazione è vera come dimostrazione dell'ambiguità di G rispondere "Si", altrimenti rispondere "No".
- 3) Chiamo D il DFA ottenuto da N_M per subset construction. Dire quanti stati finali ha D.

4) Chiamo A l'automa LR(1) per G_{11} , J lo stato iniziale di A, J_{bb} lo stato di A che si raggiunge da J tramite il cammino bb , T la tabella di parsing LR(1) per G_{11} . Se T non contiene alcun conflitto nello stato J_{bb} rispondere "No". Altrimenti: dire quali entry dello stato J_{bb} di T contengono conflitti e specificare di che conflitti si tratta.

5)

Se D_{11} è minimo rispondere "Si" altrimenti dire quanti stati ha il DFA ottenuto dalle minimizzazioni di D_{11} .

6) Scrivere l'intera riga delle Tabelle di parsing $U(1)$ per G_{11} relativa al non-terminale S .

7) Chiamo A l'automa LR(0) per G_{11} , I lo stato iniziale di A, I_{asb} lo stato di A che si raggiunge da I tramite il cammino asb , T la tabella di parsing SLR(1) per G_{11} . Se T non contiene alcun conflitto nello stato I_{asb} rispondere "No". Altrimenti: dire quali entry dello stato I_{asb} di T contengono conflitti e specificare di che conflitti si tratta.

8) Sia $exec$ l'eseguibile ottenuto utilizzando l'analizzatore ritattico su $f2.y$ e l'analizzatore linciale su $f2.l$. Dire quale output viene prodotto applicando $exec$ nell'input: $2a3b4a2$

9)

Se $\{a^u b^m \mid u, m \geq 0\}$ è un linguaggio regolare rispondere "Si", altrimenti "No".

10) Sia $exec$ l'eseguibile ottenuto utilizzando l'analizzatore linciale su $f1.l$. Dire quale output viene prodotto applicando $exec$ all'input: $CC * < amb ? = 27 >$

11) Sia $r = \epsilon | b | (\epsilon | b) (a | \epsilon | b)^* (a | \epsilon | b)$ e sia D il DFA minimo per il riconoscimento di $L(r)$. Dire quanti stati ha D e quanti di questi stati sono finali.

12) Si consideri il parser ottenuto dall'analizzatore ritattico per $f3.y$. Tra le informazioni di dettaglio fornite dal parser troviamo, tra l'altro, le seguenti:

State 12 conflicts: 1 reduce / reduce

State 12

7 A: ϵ .

9 C: ϵ, D

Formare una stringa appartenente al linguaggio generato dalla grammatica il cui parsing potrebbe presentare ad un dato conflitto menzionati

13) Nel seguito è riportata la tabella LALR(1) per la grammatica G .
 Elencare le produzioni di G

| | a | b | \$ | S | A |
|---|-------|-------|-----|---|---|
| 0 | S4 | S3 | | 1 | 2 |
| 1 | S4 | S3 | acc | 6 | 5 |
| 2 | S4 | S3 | | 7 | 2 |
| 3 | r2 | r2 | r2 | | |
| 4 | r4 | r4 | | | |
| 5 | S4/r3 | S3/r3 | | 7 | 2 |
| 6 | S4 | S3 | | 6 | 5 |
| 7 | S4/r1 | S3/r1 | r1 | 6 | 5 |