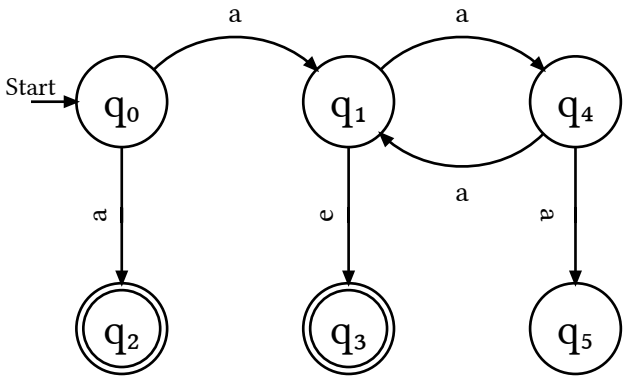
Soluzione Linguaggi Totale

2024-07-02

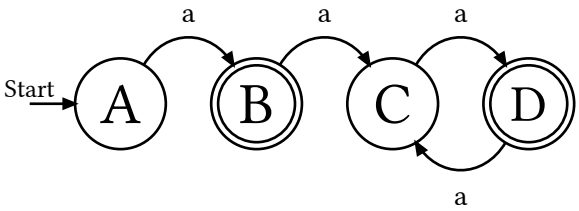
NOTA: Questa è una soluzione proposta da me, non è detto che sia giusta. Se trovate errori segnalateli o meglio correggeteli direttamente :)

1. Le due affermazioni sono entrambe false:
   a) Supponiamo che A se sia la classe dei linguaggi regolari e che B sia la classe dei linguaggi liberi deterministici. Ovviamente $A \subseteq B$. Tuttavia A è chiusa rispetto all'unione mentre B non lo è.
   b) Supponiamo che B sia la classe dei linguaggi liberi e che A sia la classe dei linguaggi liberi deterministici. Ovviamente $A \subseteq B$. Tuttavia B è chiusa rispetto all'unione mentre A non lo è.

2. Rappresentiamo l'NFA:



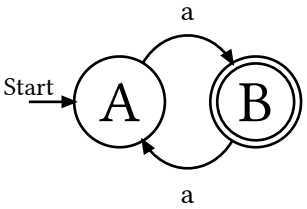
   Il DFA ottenuto per costruzione di sottoinsiemi:



   Controlliamo che sia minimo costruendo la tabella a scala:

B	x_0		
C		x_0	
D	x_0		x_0
	A	**B**	**C**

   Costruiamo il DFA minimo M':



   Il linguaggio riconosciuto è $L[M'] = \{a^{2n+1} \mid n \geq 0\}$. L'espressione regolare associata: $a(aa)^*$

3. Costruiamo la tabella dei first e dei follow:

	First	Follow
S	a, b, ε	$
A	a, ε	b, $
B	b, ε	$

i) Verifichiamo che la grammatica è di classe $LL(1)$:

$$- \ \text{First}(aAb) \cap \text{First}(\varepsilon) = \emptyset$$
$$\{a\} \cap \{\varepsilon\} = \emptyset$$

$$- \ \text{First}(aAb) \cap \text{Follow}(A) = \emptyset$$
$$\{a\} \cap \{b, \$\} = \emptyset$$

$$- \ \text{First}(bB) \cap \text{First}(\varepsilon) = \emptyset$$
$$\{b\} \cap \{\varepsilon\} = \emptyset$$

$$- \ \text{First}(bB) \cap \text{Follow}(B) = \emptyset$$
$$\{b\} \cap \{\$\} = \emptyset$$

G è di classe $LL(1)$ per un noto teorema visto a lezione.

ii) Costruiamo la tabella di parsing $LL(1)$:

	a	**b**	**$**
S	$S \to AB$	$S \to AB$	$S \to AB$
A	$A \to aAb$	$A \to \varepsilon$	$A \to \varepsilon$
B		$b \to bB$	$B \to \varepsilon$

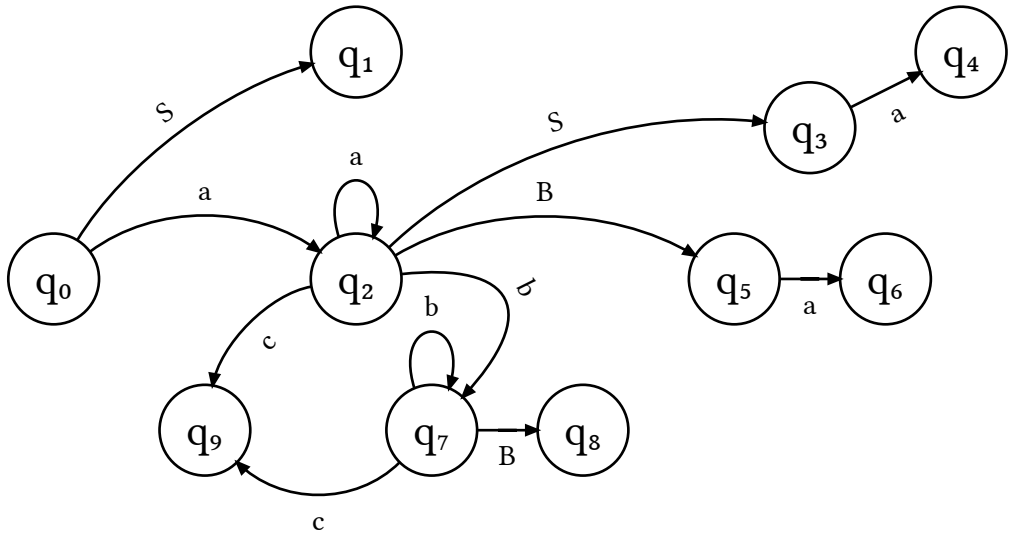
iii) Facciamo vedere il parsing sull'input: abb

$abb\$$	$S\$$
	$AB\$$
	$aAbB\$$
bb	$AbB\$$
	$bB\$$
b	$B\$$
b	$bB\$$
	$B\$$
	$\$$

Vediamo con la stringa ε:

$	$S\$$
$	$AB\$$
$	$B\$$
$	$\$$

4. Costruiamo il parser $LR(0)$. Non so come fare stati grandi con tutti gli items in typst quindi scrivo l'automa e poi definisco gli stati a parte:



q0	$S' \to .S$ $S \to .aSa$ $S \to .aBa$
q1	$S' \to S.$
q2	$S \to a.Sa$ $S \to a.Ba$ $S \to .aSa$ $S \to .aBa$ $B \to .bB$ $B \to .c$
q3	$S \to aS.a$
q4	$S \to aSa.$
q5	$S \to aB.a$
q6	$S \to aBa.$
q7	$B \to b.B$ $B \to .bB$ $B \to .c$
q8	$B \to bB.$
q9	$B \to c.$

Costruiamo la tabella $LR(0)$ sapendo che $\text{Follow}(S) = \{\$, a\}$ e $\text{Follow}(B) = \{a\}$:

	a	**b**	**c**	**$**	**S**	**B**
0	S2				G1	
1				Acc		
2	S2	S7	S9		G3	G5
3	S4					
4	R1			R1		

	a	b	c	$	S	B
5	S6					
6	R2			R2		
7		S7	S9			G8
8	R3					
9	R4					

Non essendoci conflitti G è di classe SLR(1). Il Linguaggio generato da G è:

$$L(G) = \{a^n b^m c a^n \mid n \geq 1, m \geq 0\}$$

Questo perché $L(B) = b^*c = \{b^m c \mid m \geq 0\}$ e con la produzione $S \to aBa$ genero stringhe di tipo $ab^m ca$ e con la produzione $S \to aSa$ posso aggiungere a^k all'inizio e in coda con $k \geq 0$.

5.

6.

7. Risposta:

```
ListPersona { p: Persona, next: ListPersona } + None
ListOggetto { o: Oggetto, next: ListOggetto } + None
Persona { name: string, figli: ListPersona, averi: ListOggetto }
Oggetto { name: string, prop: ListPersona }

Oggetto scudo = { name: "scudo", prop: None }
Oggetto spada = { name: "spada", prop: None }
Oggetto lancia = { name: "lancia", prop: None }

Persona ares = { name: "Ares", figli: None, averi: {
  o: scudo, next: {o: spada, next: None }}}

Persona atena = { name: "Atena", figli: None, averi: {
  o: scudo, next: {o: lancia, next: None }}}

scudo.prop = { p: ares, next: { p: atena, next: None }}
spada.prop = { p: ares, next: None }
lancia.prop = { p: atena, next: None }

Persona Zeus = { name: "Zeus", figli: { p: ares, next: { p: atena, next: None }}}
```

8. Come possiamo vedere eseguendo il codice sottostante il risultato è **2** e **42**.

```
class A {
  int i = 0;

  public A c(A a, int n) {
    if (n <= 1) { i = 1; }
    else {
      int b = new B().c(new A(), a.i()).i;
      int c = new B().c(new A(), n - 1 - a.i()).i;
      i = i + b * c;
    }
    return this;
  }

  public int i() {
```

```
    return i;
  }
}

class B extends A {
  int i = 0;

  public A c(A a, int n) {
    a.c(this, n);
    if (++i < n) { c(a, n); }
    return a;
  }

  public int i() {
    return i;
  }
}

class esercizio {
  public static void main(String[] argv) {
    A b = new B();
    int x = 2;
    System.out.println(b.c(new A(), x).i);

    b = new B();
    x = 5;
    System.out.println(b.c(new A(), x).i);
  }
}
```

Sinceramente è un esercizio complesso e molto lungo. Giallorenzo ha dato il massimo dei punti anche se si rispondeva solo il risultato di $x = 2$ e non si faceva $x = 5$. La soluzione che ha scritto alla lavagna è la seguente, però non so quanto possa essere utile:

```
c(..., 0) = 1
c(..., 1) = 1
c(..., 2) = 2
i0 = 0 + 1 * 1 = 1
i1 = 1 + 1 * 1 = 2
c(..., 3) =
i0 = 0 + 1 * 2 = 2
i1 = 2 + 1 * 1 = 3
i2 = 3 + 2 * 1 = 5
c(..., 4) =
i0 = 0 + 1 * 5 = 5
i1 = 5 + 1 * 2 = 7
i2 = 7 + 2 * 1 = 9
i3 = 9 + 5 * 1 = 14
c(..., 5) = 42
i0 = 0 + 1 * 14 = 14
i1 = 14 + 1 * 5 = 19
i2 = 19 + 2 * 2 = 23
i3 = 23 + 5 * 1 = 28
i4 = 28 + 14 * 1 = 42
```

Per i più curiosi è la seguenza di Catalan