

① L Regolare
R libero det.

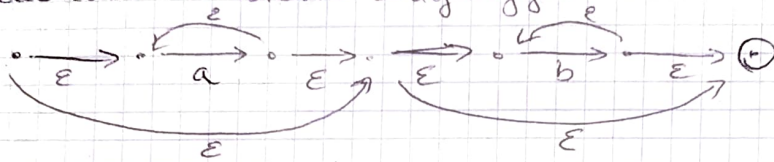
$L \cap \bar{R} = \text{Libero (det)}$ \rightarrow facoltativo

② $e = a^*(a/\epsilon)(b|b^*)$
 $d = a^*b^*$

Sono equivalenti?

$$\begin{aligned} \mathcal{L}[e] &= \mathcal{L}[a^*] \cdot \mathcal{L}[a/\epsilon] \cdot \mathcal{L}[b|b^*] = \\ &= \{a^n \mid n \geq 0\} \cdot \{a, \epsilon\} \cdot (\{b\} \cup \mathcal{L}[b^*]) = \\ &= \{a^n \mid n \geq 0\} \cdot \{b^m \mid m \geq 0\} = \mathcal{L}[a^*] \cdot \mathcal{L}[b^*] = \mathcal{L}[d] \end{aligned}$$

\Rightarrow denotano lo stesso linguaggio



③ $\left. \begin{aligned} S &\rightarrow aSB \mid bSA \\ A &\rightarrow \epsilon \mid a \mid B \\ B &\rightarrow \epsilon \mid b \mid A \end{aligned} \right\} G$

	first	follow
S	a, b	\$, a, b
A	ϵ, a, b	\$, a, b
B	ϵ, a, b	\$, a, b

④ verifico che G è di classe LL(1)

$\text{follow}(A) \cap \text{first}(a) \stackrel{?}{=} \emptyset$
 $\{\$, a, b\} \cap \{a\} = \emptyset \Rightarrow G \text{ non è LL(1)}$

② Eliminare le produzioni ϵ ③ Eliminare le produzioni unitarie

$N(G) = \{A, B\}$

$S \rightarrow aSB \mid aS \mid bSA \mid bS$

$A \rightarrow a \mid b$

$B \rightarrow b \mid a$

$S \rightarrow aSB \mid aS \mid bSA \mid bS$

$A \rightarrow a \mid b$

$B \rightarrow b \mid a$

④ Quale linguaggio genera G?

non genera mai una stringa di terminali, $\mathcal{L}(G) = \emptyset$

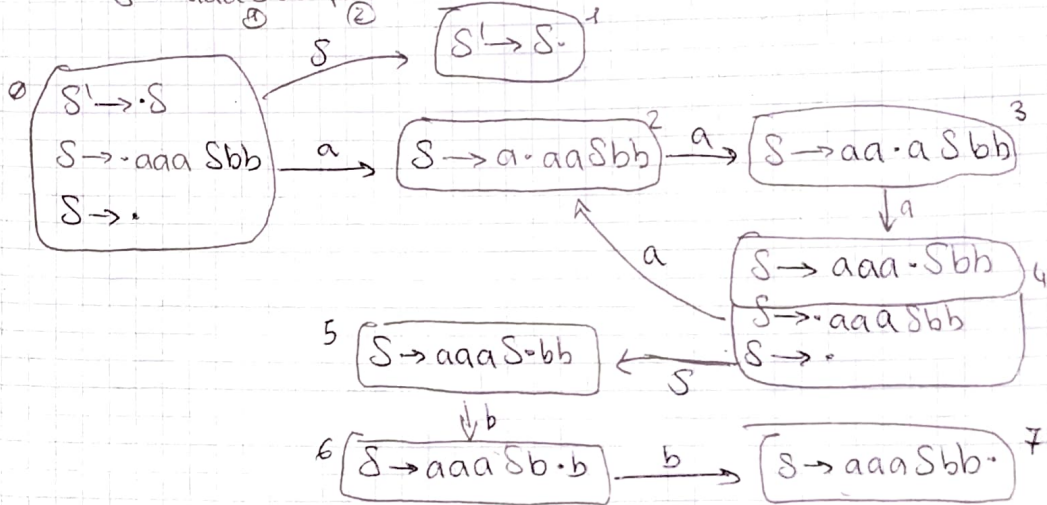
⑤ È un linguaggio LL(1)?

Sì, perché tutti i lang. regolari sono di classe LL(1)

④ $L = \{a^{3n} b^n \mid n \geq 0\}$ è SLR(1)?

• Trovo ^{una} ~~le~~ grammaticale che lo genera

$$S \rightarrow aaaSbb \mid \epsilon$$



S	First	Follow
S	a, ϵ	$\$, b$

• tabelle di parsing

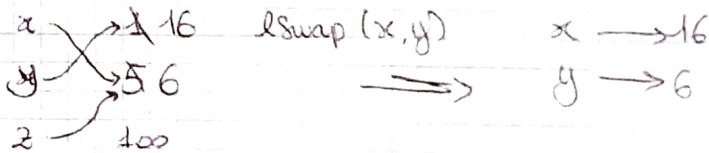
	a	b	$\$$	S
0	S2	R2	R2	G1
1			ACC	
2	S3			
3	S4			
4	S2	R2	R2	G5
5		S6		
6		S7		
7		R1	R1	

non ci sono conflitti, dunque L è un linguaggio SLR(1)

⑤ lswap (N1, N2)



assign (N1, N2)



⑥ efo, non influenza perché con i display, se accedo ad una var locale, prendo il puntatore che mi punta alle posizioni delle dichiarazioni. efo in tutti i casi tranne nel 6 in cui il tempo è costante

7

```
int * p(= NULL);
```

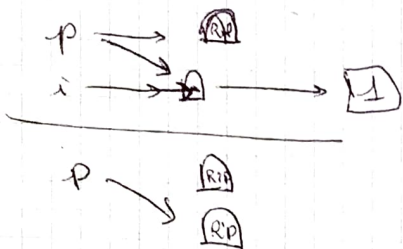
→ facultativo

```
{ int i = 1;
```

```
  p = &i;
```

→ dangling reference

Con le tombeone:



8

```
class A { double x = 3.0;
```

```
  double f() { return x; }
```

```
  double g() { return f(); }
```

```
}
```

```
class B extends A {
```

```
  double x = 2.0;
```

```
  → double f(double x) { return x/10; }
```

```
  → double g(double x) { return f(F(g()) + F()) + f() + f(g()) }
```

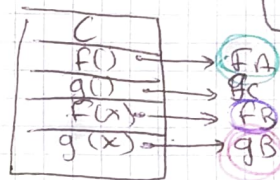
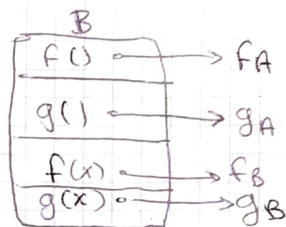
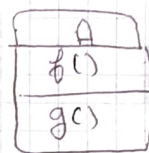
```
}
```

```
class C extends B { double g() { return f(x) - x; }
```

```
}
```

```
B c = new C;
```

```
write (c.g(c.x));
```



⇒ output: 3.14