

Tempo a disposizione: ore 2.

Svolgere gli esercizi 1–4, 5–6 e 7–8 su tre fogli separati.

1. Siano date due macchine astratte M_0 e M_1 , con relativo linguaggio L_0 e L_1 , rispettivamente. Inoltre, siano dati un interprete $\mathcal{I}_{L_2}^{L_0}$, che realizza la macchina astratta M_2 sopra la macchina ospite M_0 , un compilatore $\mathcal{C}_{L_2, L_1}^{L_2}$ ed un interprete banale $\mathcal{I}_{L_2}^{L_2}$. Utilizzando il software disponibile, è possibile realizzare un interprete per L_2 scritto in L_1 ?
2. Determinare tutti gli item LR(0) che si possono ottenere dalla grammatica $S \rightarrow \epsilon | c | aSb$. Quale linguaggio genera questa grammatica?
3. Sia data la grammatica G con simbolo iniziale S

$$S \rightarrow aS|aa$$

(i) Quale linguaggio genera G ? (ii) È la grammatica G regolare? (iii) Determinare una grammatica G' , equivalente a G , in forma normale di Chomsky.

4. Si consideri la grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow aBc \\ B &\rightarrow bCb|b \\ C &\rightarrow \epsilon \end{aligned}$$

(i) Verificare se G sia di classe SLR(1). (ii) Verificare che G non è di classe LL(1). (iii) Determinare il linguaggio generato $L(G)$. (iv) Verificare che $L(G)$ è un linguaggio di classe LL(1).

5. Si dica, motivando la risposta, se un programma scritto in un qualsiasi linguaggio che prevede l'allocazione dinamica di memoria con stack e heap può essere tradotto in un programma equivalente scritto in un linguaggio che permette solo l'allocazione statica della memoria
6. L'esecuzione del seguente frammento di codice su una certa implementazione risulta nella stampa dei valori 4 e 13.

```
int W[10];
int x = 4;
for (int i=0, i<10, i++) W[i]=i;
void foo(int x; int y){
x = x+2;
y= y + (x++)
}
foo (x, W[x])
write (W[4])
write (W[5])
write (W[6])
```

Si fornisca una possibile spiegazione.

7. Dato il frammento di codice sotto, indicare, spiegando brevemente il ragionamento seguito, se esistono errori rilevati da un controllore dei tipi che supporta equivalenza (con nozione di compatibilità) di tipo strutturale. Che errori, se esistono, vengono rilevati da un controllore alternativo che supporta duck typing?

```
type A = record { int a, int b }
type B = record { int a }
type C = record { B x, A y }
type D = record { A x, B y }
f( C a, D b ){
a = b;
return a.x.b;
}
```

8. Cosa stampa il frammento di codice sotto, considerando di avere un linguaggio ad oggetti che supporta ereditarietà (B **extends** A indica che B eredita da A e **super** riferisce l'oggetto come super-classe), indirizzamento dinamico e accesso statico dei campi? Spiegare brevemente il ragionamento seguito.

```
class A { int x = 3; int f( int y ){ return y + x; } }
class B extends A { int x = 1; int f( int y ){ return super.f( x ) + y; } }
class C extends B { int x = 4; int f( int y ){ return super.f( y ) + x; } }
B x = new C();
print( x.f( x.x ) );
```