

Tempo a disposizione: ore 2.

Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.

1. Si consideri l'espressione regolare $e = a^*(\epsilon|b)a^*$. Si determini il linguaggio $\mathcal{L}[e]$. Si costruisca l'associato NFA, secondo la costruzione vista a lezione. Infine, si costruisca l'associato DFA, utilizzando la costruzione per sottoinsiemi.
2. Si costruisca un semplice automa che riconosca il linguaggio $L = \{a^n b^m \mid n \geq 1, m \geq n\}$.
3. Si consideri la grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow aSB \mid bSA \\ A &\rightarrow \epsilon \mid a \mid B \\ B &\rightarrow \epsilon \mid b \mid A \end{aligned}$$

- (i) Verificare che G non è di classe LL(1). (ii) Manipolare la grammatica G , rimuovendo prima le produzioni epsilon e poi le produzioni unitarie. (iii) Quale linguaggio genera G ? (iv) Il linguaggio $L(G)$ è di classe LL(1)? Giustificare la risposta.
4. Si costruisca un parser LR(0) per il linguaggio $L = \{a^n b^{n+1} \mid n \geq 0\}$ e si mostri il suo funzionamento per gli input abb e aba .

5. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping dinamico, deep binding e passaggio per nome (dove specificato):

```
int x = 100;
procedure ass_x(n:int)
  {x = x+n;
  }
procedure stampa_x
  {write_integer(x);
  }
procedure pippo(function S, P ; int name n )
  { int x= 10;
    if n=1 then {
      ass_x(n);
      stampa_x;
    }

    if n=3 then {
      S(n);
      P;
    }
  }
}
{
int x = 30;
int n = 1;
pippo(ass_x, stampa_x, n++);
}
```

6. In un linguaggio che permette overloading si consideri il seguente frammento di codice

```
int x,y;
float z,w;
x=2;
y=3;
z=3.5;
w= 3.0;
```

Dire, motivando la risposta, quali fra le seguenti quattro espressioni sono corrette nel contesto del precedente frammento, assumendo che + abbia al massimo due significati sovraccaricati:

```
x+y; z+w; x+z; x+w;
```

7. Si dica, motivando la risposta, cosa viene stampato dal seguente frammento di programma Java, supponendo che Y sia dichiarata come sottoclasse di X.

```
void f() throws Y {
  throw new Y();
}

void g (int sw) throws X , Y {
  if (sw == 0) {f();}
  try {f();} catch (X e) {write("in_g");}
}

...
try {g(1);}
  catch (Y e) {write("in_main");}
```

8. Si discutano brevemente le tre diverse nozioni di variabile esistenti nei paradigmi imperativo, funzionale e logico.