

Tempo a disposizione: ore 2.

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

Scrivere nome, cognome e matricola su ogni foglio consegnato.

FOGLIO **1** ▷ 1. Fornire una definizione regolare per la categoria sintattica *identificatore*, che deve soddisfare i seguenti requisiti: (i) iniziare con una lettera maiuscola, (ii) contenere almeno una lettera minuscola, (iii) contenere almeno una cifra, (iv) contenere solo lettere o cifre.

FOGLIO **1** ▷ 2. Si definisca un semplice automa che generi il linguaggio $L = \{a^n b^m \mid n+m=2 \times k, k \geq 0\}$. Tale linguaggio è regolare? Motivare la risposta.

FOGLIO **1** ▷ 3. Si consideri la grammatica G con simbolo iniziale S :

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow aA \mid \epsilon \\ B &\rightarrow aBb \mid \epsilon \end{aligned}$$

(i) Quale linguaggio genera la grammatica G ? (ii) Calcolare i first e i follow per tutti i nonterminali di G . (iii) Verificare se G sia di classe LL(1). (iv) In caso affermativo, costruire la tabella di parsing LL(1); altrimenti, argomentare se G possa essere di classe LL(k) per qualche $k \geq 2$.

FOGLIO **1** ▷ 4. Si consideri la grammatica G con simbolo iniziale S del punto precedente. (i) Si calcolino tutti gli item LR(0). (ii) Si calcoli solo lo stato iniziale dell'automata canonico LR(0). (iii) In base a quanto calcolato, si può concludere che G non sia di classe LR(0)?

FOGLIO **2** ▷ 5. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che usa scoping dinamico e shallow binding:

```
x: integer;
procedure ass_x(n: integer)
  {x = n;
  }
procedure stampa_x
  {write_integer(x);
  }
procedure pippo(S, P: function; n: integer)
  {x: integer;
  if n=1 then ass_x(n)
  else S(n);

  P;
  }
ass_x(0);
pippo(ass_x, stampa_x, 1);
stampa_x;
ass_x(0);
pippo(ass_x, stampa_x, 2);
stampa_x;
```

- FOGLIO 2 ▷ 6. Si dica cosa viene stampato dal seguente frammento di codice scritto in uno pseudo-linguaggio che ammette parametri per *riferimento*.

```
int X[10];
int i = 1;
X[0] = 0;
X[1] = 0;
X[2] = 0;
void foo (reference int Y,J){
    X[J] = J+1;
    write(Y);
    J++;
    X[J]=J;
    write(Y);
}
foo(X[i],i);
write(X[i]);
```

- FOGLIO 3 ▷ 7. Si indichi quali dei sistemi di equazioni di tipo elencati sotto sono o no risolvibili, riportando brevemente il ragionamento seguito; nel caso positivo, indicare uno dei possibili assegnamenti leciti delle variabili di tipo. Le equazioni usano la notazione Int , Str , per i tipi base, X , Y per le variabili di tipo e $X \rightarrow \text{Int}$ e $X \rightarrow \text{Int} \rightarrow Y$ per indicare il tipo funzione. Per la risoluzione delle equazioni, può essere utile ricordare la logica dell'algoritmo di unificazione. Ad esempio, il sistema di equazioni $X \rightarrow X \rightarrow X = \text{Int} \rightarrow Y$, $X \rightarrow X = Y$ ha soluzione $X = \text{Int}$, $Y = \text{Int} \rightarrow \text{Int}$.

S1. $Y = X \rightarrow \text{Int}$, $X \rightarrow Y = \text{Int} \rightarrow X$

S3. $Y = \text{Int} \rightarrow \text{Str}$, $X = \text{Int}$, $Z \rightarrow Y = \text{Int} \rightarrow X \rightarrow \text{Str}$, $X \rightarrow Y = \text{Int} \rightarrow Z \rightarrow \text{Str}$

S2. $Z = Y \rightarrow X$, $X \rightarrow Y = Z \rightarrow \text{Int}$

S4. $X \rightarrow Y \rightarrow Z = \text{Int} \rightarrow X$, $Y = Z \rightarrow \text{Int}$

- FOGLIO 3 ▷ 8. Viene dato uno pseudolinguaggio dove **new** crea un nuovo oggetto nello heap e dove le strutture A, B e C hanno il campo next che può puntare ad un oggetto di tipo A, B o C. Un oggetto di tipo A occupa 2 byte e sia oggetti di tipo B che C occupano 4 byte. Il linguaggio usa stop-and-copy garbage collection, attivando la collection superata la soglia del 80% di occupazione della memoria disponibile al collector. Lo heap ha 30 byte di memoria complessiva (inizialmente vuota). Indicare, spiegando brevemente il ragionamento seguito, quante volte viene chiamato il garbage collector nell'esecuzione del seguente frammento di codice.

```
A a = new A();
for ( int j = 0; j < 3; j++ ) {
    a.next = new A();
    b = new B();
    b.next = new C();
    a.next.next = b;
    a.next = a;
}
```