

Linguaggi per il Web – appello del 13/1/2014

COGNOME:
NOME:
MATRICOLA:

Autorizzo la pubblicazione del mio voto di questo esame sul sito web http://www.dis.uniroma1.it/~rosati/lw , secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede,

Esercizio 1 (a) Data la seguente grammatica G :

$$\begin{aligned} S &\rightarrow aA \mid bB \mid cCAB \\ A &\rightarrow aA \mid b \\ B &\rightarrow bB \mid c \\ C &\rightarrow S \end{aligned}$$

è possibile stabilire se G è una grammatica LL(1) senza costruire esplicitamente gli insiemi FIRST e FOLLOW e la tabella di parsing di G ? motivare la risposta;

(b) Data la seguente grammatica G :

$$\begin{aligned} S &\rightarrow XYS \mid XSY \mid YXS \mid YSX \\ X &\rightarrow XXx \mid xyx \mid Xy \mid xxy \\ Y &\rightarrow YY \mid YYZw \mid y \mid yYS \\ Z &\rightarrow zz \mid Sz \mid zS \end{aligned}$$

scrivere una grammatica G' tale che G' non presenti né ricorsione sinistra diretta né prefissi comuni e tale che $\mathcal{L}(G') = \mathcal{L}(G)$.

Esercizio 2 Si consideri il frammento del linguaggio Java costituito dalle stringhe che corrispondono alla definizione di un metodo Java. Un metodo ha una intestazione e un corpo. L'intestazione è costituita da un tipo che deve essere `void`, dal nome del metodo e da una sequenza di zero o più argomenti tra parentesi tonde e separati da virgole. Ogni argomento è di tipo `int` o `float`. Il corpo è racchiuso tra parentesi graffe ed è costituito da una sequenza di zero o più istruzioni. Ogni istruzione può essere: (i) una dichiarazione di una o più variabili (separate da virgole) di tipo `int` o `float`; (ii) una assegnazione, il cui lato destro può essere una costante intera, una costante di tipo `float`, un identificatore di variabile, o una invocazione di metodo; (iii) una invocazione di metodo, in cui ogni argomento (parametro attuale) può essere una costante intera, una costante di tipo `float`, o un identificatore di variabile, o a sua volta una invocazione di metodo; (iv) una istruzione di tipo `if`, con ramo `else` opzionale, il cui ramo `then` e l'eventuale ramo `else` contengono una istruzione, e la cui condizione è un confronto di uguaglianza o disuguaglianza tra due espressioni, ognuna delle quali può essere una costante intera, una costante di tipo `float`, un identificatore di variabile, o una invocazione di metodo; (v) una istruzione `while` il cui corpo contiene una istruzione e la cui condizione è un confronto di uguaglianza o disuguaglianza tra due espressioni, ognuna delle quali può essere una costante intera, una costante di tipo `float`, o un identificatore di variabile, o una invocazione di metodo; (vi) una sequenza (anche vuota) di istruzioni delimitata da parentesi graffe.

Esempi di stringhe appartenenti a questo linguaggio sono i seguenti:

```
void metodo1() {
    int z,w;
    x=metodo3(123);
    metodo4();
    { y = p(x, 3.21); }
    while (a== -3.42)
        x23=-0.23E-12;
}

void metodo2(int z) {
    { z=metodo5();
      { y=met3(x,-1,3.0);
        z=metodo6(z,1); }
      if (p!=pluto(y)) a=z;
      else if (x==0) { a=.25; } }
}

void metodo3(float x, int z) {
    float a12;
    { paperino=-0.32e-3;
      { x=met1(x,abc12,0,a,y); y=0; } }
    if (z==3.14) y=x;
    else {
        pluto(25,x); if (p4!=0) x=1.15; }
}
```

Scrivere una grammatica non contestuale per tale linguaggio, dividendo la specifica del lessico del linguaggio (che va definita mediante espressioni regolari) dalla specifica della sintassi vera e propria. Scrivere preferibilmente tale specifica come specifica JavaCC.

Esercizio 3 (a) Scrivere un documento HTML contenente una form che presenta i seguenti campi:

- cognome e nome (casella di testo editabile lunga 50 caratteri)
- codice fiscale (casella di testo editabile lunga 16 caratteri)
- sesso (selezionabile tramite due bottoni radio)
- provincia di residenza (da scegliere da un menu che riporta le 5 provincie del Lazio)
- telefono fisso (casella di testo editabile lunga 12 caratteri)
- telefono cellulare (casella di testo editabile lunga 12 caratteri)
- bottone di invio
- bottone di reset

(b) Aggiungere al documento HTML una funzione JavaScript che esegue i seguenti controlli: (i) verifica che il campo cognome e nome non sia vuoto; (ii) verifica che il campo cognome e nome non sia un numero; (iii) verifica che sia stata selezionata una provincia; (iv) verifica che o il telefono fisso o il telefono cellulare siano non vuoti. Inoltre, fare in modo che, nel documento HTML, tale funzione JavaScript venga eseguita quando l'utente invia la form.

Esercizio 4 Data la seguente DTD:

```

<!DOCTYPE r [
  <!ELEMENT r ((x|y|z),y,r*,((s,y)|(z,s)))>
  <!ELEMENT s (#PCDATA|r)*>
  <!ELEMENT x (x?,w,(y|r|s)*)>
  <!ELEMENT y (z,(u|s)*,(z,x)+)>
  <!ELEMENT z (#PCDATA|z)*>
  <!ELEMENT w EMPTY>
  <!ATTLIST s attrs CDATA #REQUIRED>
  <!ATTLIST w attrw CDATA #IMPLIED>
]>

```

1) dire se la DTD è corretta ed in caso negativo evidenziare gli errori presenti e correggerli; 2) scrivere un documento XML che sia valido rispetto alla DTD (eventualmente corretta) e che contenga un numero minimo di elementi.

Esercizio 5 Data la seguente DTD:

```

<!DOCTYPE r [
  <!ELEMENT r (x|y)*>
  <!ELEMENT x EMPTY>
  <!ELEMENT y ((z,w)*,t+)>
  <!ELEMENT z (#PCDATA)>
  <!ELEMENT w (#PCDATA)>
  <!ELEMENT t EMPTY>
  <!ATTLIST z attra CDATA #REQUIRED>
  <!ATTLIST t attrc CDATA #IMPLIED>
]>

```

scrivere un XML Schema corrispondente a tale DTD.

Esercizio 6 Scrivere un foglio di stile XSL che, dato un documento XML, restituisce il documento tale che: 1) l'elemento radice di input viene trasformato in un elemento `r`, e il suo contenuto viene ricorsivamente trasformato; 2) per ogni elemento che è figlio dell'elemento radice viene creato in output un elemento `figlio`, e viene creato, come sottoelemento di tale elemento `figlio`, un elemento vuoto che ha per nome il nome dell'elemento di input corrente. Inoltre, il contenuto dell'elemento di input viene ricorsivamente trasformato; 3) per ogni elemento che è figlio di un figlio dell'elemento radice viene creato in output un elemento `nipote`, e viene creato per tale elemento un attributo di nome `nomeElemento` e valore uguale al nome dell'elemento di input corrente. Inoltre, il contenuto dell'elemento di input viene ricorsivamente trasformato; 6) tutti gli altri elementi non vengono copiati e il loro contenuto viene ricorsivamente trasformato; 7) nessuna parte testuale del documento di input viene copiata in output.

Ad esempio, se il documento XML di input è il seguente:

```

<a>
  <x>testo 0
</x>
  <b>
    <p>
      <w>testo 2</w>
    </p>
  </b>
  <y>
    <p/>
    <y>testo 3
      <d>testo 4</d>
    </y>
  </y>
</f/>
</a>

```

il foglio di stile applicato al documento deve restituire il documento seguente:

```

<a>
  <figlio>
    <x/>
  </figlio>
  <figlio>
    <b/>
    <nipote nomeElemento="p"/>
  </figlio>
  <figlio>
    <y/>
    <nipote nomeElemento="p"/>
    <nipote nomeElemento="y">
  </figlio>
  <figlio>
    <f/>
  </figlio>
</a>

```