Linguaggi per il Web – appello del 24/9/2014

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:

$$S \to dAS \mid c$$
$$A \to BC$$
$$B \to bS \mid cc \mid aA$$
$$C \to SA$$

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:

$$S \to SSa \mid SSb \mid ab \mid SSz \mid abc$$
$$A \to AAbB \mid AAb \mid bbBC \mid bbBCA$$
$$B \to BBaC \mid BBS \mid cccC \mid ccabC$$
$$C \to cc \mid cca \mid SC \mid SCA$$

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 uno o più metodi 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 `double`. 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 `double`; (ii) una assegnazione, il cui lato destro può essere una costante intera, una costante di tipo `double`, 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 `double`, o un identificatore di variabile, o a sua volta una invocazione di metodo; (iv) una istruzione di tipo `if`, con ramo `else` obbligatorio, 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 `double`, 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 `double`, 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, -0.23E-12); }
  while (a==-3.42) x23=-0.2;
}
```

```
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=2; else a=1;}
}
```

```
void metodo3(double x, int z) {
  double a12;
  { paperino=-0.32e-3;
    { x=met1(x,abc12,0,a,y); y=0; } }
  if (z==3.14) y=x;
  else { pluto(25,x); 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)
- regione di residenza (da scegliere da un menu che riporta le regioni italiane)
- indirizzo (casella di testo editabile lunga 42 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 il codice fiscale contenga esattamente 16 caratteri; (iv) verifica che, se la regione selezionata è il Lazio, il campo indirizzo non è vuoto. 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 (s|t)+>
  <!ELEMENT s (((a,b,r*|(c, d)),e,(b|(c,c)|a)*,(b,c,t))>
  <!ELEMENT t (((a,b,r*|(c, d)),e,(b|(c,c)|a)*,(b,c))>
  <!ELEMENT a (e,a*,(c,b)+)>
  <!ELEMENT b (#PCDATA|a)*>
  <!ELEMENT c (b?,((r,a)|c|s|t)*,d)>
  <!ELEMENT d EMPTY>
  <!ELEMENT e ANY>
  <!ATTLIST t attra CDATA #IMPLIED>
  <!ATTLIST d attrf CDATA #REQUIRED>
]>
```

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 (t+,(w,z)*)>
  <!ELEMENT y EMPTY>
  <!ELEMENT t EMPTY>
  <!ELEMENT w (#PCDATA)>
  <!ELEMENT z (#PCDATA)>
  <!ATTLIST y attrc CDATA #IMPLIED>
  <!ATTLIST z attra CDATA #REQUIRED>
]>
```

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 `e`, e viene creato, come sottoelemento di tale elemento `e`, un elemento che ha per nome `oldName` e che contiene come testo il nome dell'elemento di input corrente. Inoltre, il contenuto dell'elemento di input viene ricorsivamente trasformato; 2) ogni elemento che è figlio dell'elemento radice viene copiato in output, e il suo contenuto viene ricorsivamente trasformato; 3) ogni elemento che è figlio di un figlio dell'elemento radice viene copiato in output come figlio dell'elemento radice, e il suo contenuto viene ricorsivamente trasformato; 4) ogni elemento che è figlio di un figlio di un figlio dell'elemento radice viene copiato in output come figlio dell'elemento radice, e viene creato per tale elemento un attributo di nome `nomeElemento` e valore uguale al nome dell'elemento di input corrente. Inoltre il contenuto di tale elemento viene ricorsivamente trasformato; 6) tutti gli altri elementi non vengono copiati e il loro contenuto viene ricorsivamente trasformato: 7) tutte le parti testuali del documento di input non vengono copiate in output.

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

```
<a>
  <x>testo 1</x>
  <b>
    <p>
      <w>testo 2</w>
    </p>
  </b>
  <y>
    <p/>
    <y>testo 3
      <d>
        <z>testo 4</z>
      </d>
    </y>
  </y>
</a>
```

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

```
<e>
  <oldName>a</oldName>
  <x></x>
  <b></b>
  <p></p>
  <w>
    <nomeElemento>w</nomeElemento>
  </w>
  <y></y>
  <p/>
  <y></y>
  <d>
    <nomeElemento>d</nomeElemento>
  </d>
</e>
```