Linguaggi per il Web – appello del 23/6/2014

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: $S \to aX \mid bY \mid cXY$
$X \to bX \mid c$
$Y \to Sc$
$Z \to aXY \mid b$

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:

$S \to SAB \mid SSC \mid BS \mid BC$
$A \to BBx \mid Ayx \mid yw \mid ABxy$
$B \to zz \mid Sz \mid zS$
$C \to CCa \mid CCBAw \mid y \mid yS$

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 `int` o `double`, 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 o di tipo `double`, un identificatore di variabile, o una invocazione di metodo, in cui ogni argomento (parametro attuale) può essere una costante intera o di tipo `double`, un identificatore di variabile, o a sua volta una invocazione di metodo; (iii) 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 o di tipo `double`, un identificatore di variabile, o una invocazione di metodo; (iv) 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 o di tipo `double`, un identificatore di variabile, o una invocazione di metodo; (v) una istruzione `return` che deve obbligatoriamente comparire in ogni metodo come ultima istruzione del corpo del metodo, e il cui argomento può essere una costante intera, una costante di tipo `double`, un identificatore di variabile, o una invocazione di metodo.

Esempi di stringhe appartenenti a questo linguaggio sono i seguenti:

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

```
double 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;
  return pluto(a);
}
```

```
int metodo3(double x, int z) {
  double a12;
  paperino=-0.32e-3;
  x=met1(x,abc12,0,a,y);
  if (z==3.14) y=x;
  else z=pluto(25,x);
  return -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)
- matricola (casella di testo editabile lunga 12 caratteri)
- sesso (selezionabile tramite due bottoni radio)
- città di residenza (da scegliere da un menu che riporta alcune città italiane)
- 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 e contenga almeno 5 caratteri; (ii) verifica che il campo matricola sia non vuoto e sia un numero; (iii) verifica che sia stata selezionata una città di residenza; (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 ((r*|c),(b|(c,c)|a),((f,c)|(a,f)),e)>
  <!ELEMENT a (#PCDATA|a)*>
  <!ELEMENT b (b?,((r,a)|c|f)*,d)>
  <!ELEMENT c ((u|f)*,a,(a,b)+)>
  <!ELEMENT d EMPTY>
  <!ELEMENT e ANY>
  <!ELEMENT f (#PCDATA|r)*>
  <!ATTLIST a attra CDATA #IMPLIED>
  <!ATTLIST f 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 (z+,(t|w)?)>
  <!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 il suo contenuto viene ricorsivamente trasformato; 2) per ogni elemento che è figlio dell'elemento radice viene creato in output un elemento `child`, e viene creato, come sottoelemento di tale elemento `child`, un elemento vuoto che ha per nome `old` e che contiene un attributo il cui nome è il nome dell'elemento di input corrente e il cui valore è `"0"`. 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 `new`, 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) tutte le parti testuali del documento di input vengono copiate 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:

```
<e>
  <child>
    <old x="0"/>
    testo 0
  </child>
  <child>
    <old b="0"/>
    <new nomeElemento="p">testo 2</new>
  </child>
  <child>
    <old y="0"/>
    <new nomeElemento ="p"></new>
    <new nomeElemento ="y">testo 3 testo 4</new>
  </child>
  <child>
    <old f="0"/>
  </child>
</e>
```