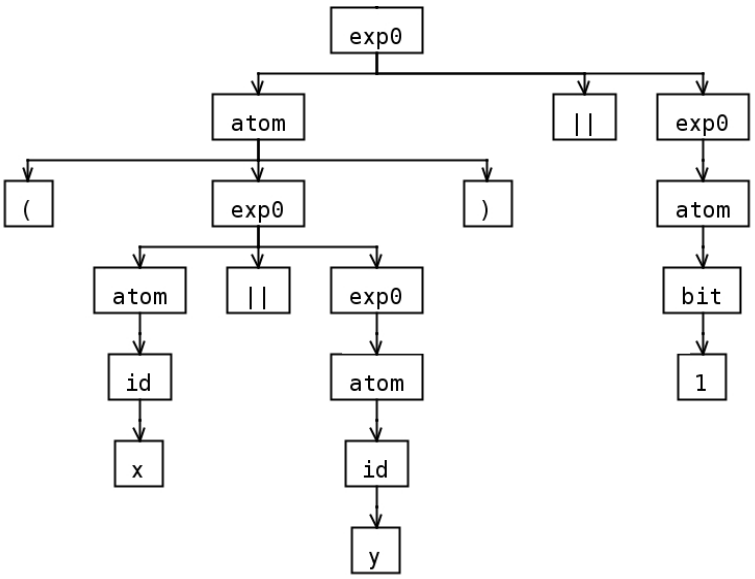
Linguaggi e Programmazione Orientata agli Oggetti

Soluzioni della prova scritta

a.a. 2011/2012

31 agosto 2012

1. (a) Albero di derivazione per la stringa `(x||y)||1`.



   (b) Grammatica estesa.

```
Exp0  ::= Exp1 || Exp0 | Exp1
Exp1  ::= Atom && Exp1 | Atom
Atom ::= Id | Bit | ( Exp0 )
Id ::= x | y | z
Bit ::= 0 | 1
```

2. (a)
```
let rec select1 = function
    (a,_)::l -> a::select1 l
  | [] -> [];;
```

   (b)
```
let rec filter p = function
    e::l -> let l2 = filter p l in if p e then e::l2 else l2
  | [] -> [];;
```

   (c)
```
let select1where2eq e l = select1 (filter (fun (_,b) -> b=e) l);;
```

3. (a)
```
public final class Pair<T1, T2> implements IPair<T1, T2> {
    private final T1 fst;
    private final T2 snd;

    public Pair(T1 fst, T2 snd) {
        this.fst = fst;
        this.snd = snd;
    }

    @Override
    public T1 getFst() {
        return fst;
    }

    @Override
    public T2 getSnd() {
        return snd;
    }

}
```

(b)
```
IPredicate<String, Integer> pred1 = new IPredicate<String, Integer>() {
    @Override
    public boolean isTrueOn(String el1, Integer el2) {
        return el2 >= 1968;
    }
};
```

(c)
```
@Override
public boolean hasNext() {
    return it.hasNext();
}
@Override
public T1 next() {
    return it.next().getFst();
}
```

(d)
```
public boolean hasNext() {
    return it.hasNext();
}
@Override
public Pair<T1, T2> next() {
    Pair<T1, T2> res = it.next();
    goToNext();
    return res;
}
```

4. (a) `(0.5, 0.5)`

(b) `(1.5, 2.5)`

(c) `(1.5, 2.5)`

(d) Errore di compilazione: il metodo `move(int,int)` non è applicabile ad argomenti di tipo **`float`**.

(e) `(1, 2)`

(f) `(-0.5, -1.5)`