Cognome e Nome:_____ Matr.:_____

Linguaggi di programmazione – A
11 giugno 2018

Esercizio 1.A – Rispondere, in maniera concisa, alle seguenti domande (12 punti)

1. Quali sono in difetti del comando GOTO?

2. Descrive la struttura di una definizione di type class in Haskell.

3. Cosa distingue una comunicazione sincrona da una asincrona?

4. Cosa si intende per pragmatica di un linguaggio di programmazione?

5. Quali controlli sul codice vengono svolti dall'analisi semantica?

6. Cosa sono le funzioni di ordine superiore?

7. Cosa si intende per valutatine corto-circuito un'operazione booleana?

8. Cosa prevede il principio dell'incapsulamento?

9. Nella gestione dello scoping statico, cosa si intende per catena statica?

10. Cos'è il dope vector?

11. Cosa si intende per equivalenza strutturale tra i tipi

12. In un linguaggi ad oggetti cosa distingue i metodi pubblici, da quelli protetti?

13. Che tipo di linguaggi vengono riconosciuti dagli analizzatori sintattici?

14. Cos'è il Frame Pointer?

15. Cos'è una dangling reference?

1. Cognome e Nome:_____ Matr.:_____

– 11 giugno 2018– A

Esercizio 2.A – Grammatiche (4 punti)

1. Data la grammatica:

$$L \rightarrow \epsilon \mid aLcLc \mid LL$$

   - si scrivano tutte le parole di lunghezza minore di 11 generati dalla grammatica,
   - si stabilisca se la grammatica è ambigua,
   - nel caso lo sia, si definisca una grammatica non ambigua equivalente,
   - si descriva il linguaggio generato,
   - si stabilisca se è un linguaggio regolare,
   - nel caso lo si descriva come espressione regolare.
   - abbozzare le regole YACC-Happy, per la creazione di un programma che dalla stringa costruisce l'albero sintattico.

Esercizio 3.A – Scoping (6 punti)

1. Si mostri l'evoluzione dello stack di attivazione e dell'output del seguente frammento di programma espresso in un linguaggio C-like con scoping statico, deep binding, assegnamento che calcola l-value dopo r-value, valutazione delle espressioni da sinistra a destra e indici vettori inizianti da 0:

```
int y = 1 , v [] = { 2, 3, 4 } ;
int H(name int x ) {
  v[ x mod 3] = ++y + x ;
  write ( v[0] ,v[1] , v[2] ) ;
  return x ;
}
int F( int G( name int ), ref int x  ) {
  int y = x++ ;
  write G( x + y-- ) ;
  return y ;
}
write ( H(v[y]) ) ;
write ( F( H, v[1] ) ) ;
```

1. Cognome e Nome:_____ Matr.:_____

– 11 giugno 2018– A

Esercizio 4.A – Haskell (7 punti)

1. Scrivere una funzione Haskell che controlli se due liste sono l'una una permutazione dell'altra. L'unica operazione permessa sugli elementi della lista è il test di uguaglianza. Si definisca inoltre il tipo Haskell della funzione definita e delle funzioni ausiliarie.

   In alternativa, se l'esercizio precedente risultasse troppo difficile, scrivere le opportune definizioni di tipo e una funzione che conti il numero di occorrenze di un elemento in un albero binario.

Esercizio 5.A – Sistema di assegnazione di tipo (4 punti)

1. Nel sistema di tipi per il linguaggio F1, costruire la derivazione di tipo per la seguente espressione:

   ```
   (\x :(Ref Nat) -> (\y :  Unit -> (deref x) )(x := ((deref x) + 1))) (ref 3)
   ```

   Nota: i primi passi di derivazione, a partire dagli assiomi, se elementari o ripetizioni di derivazioni già fatte, possono essere omessi.

   Determinare inoltre il valore restituito dall'espressione nel caso di valutazione eager, e nel caso di valutazione lazy.