

Fondamenti di Programmazione – Appello 21/01/2022
Anno Accademico 2021/22

Traccia T1

Cognome:	Nome:	Matricola:	Anno Immatricolazione:
I parziale	Svolgere Esercizio 1, 2 e 3		Tempo: 2 ore
II Parziale	Svolgere Esercizio 4, 5		Tempo: 2 ore
Totale	Svolgere tutti gli esercizi		Tempo: 3 ore

Esercizio1

a) Dare una grammatica libera dal contesto per il linguaggio L di tutte le stringhe sull'alfabeto {**viaggiare, si potrà, potremo, presto, resistiamo, siamo ottimisti, divertirci**} che:

- cominciano con almeno un “siamo ottimisti”,
- finiscono con almeno un “resistiamo”,
- “si potrà” è sempre immediatamente seguito da almeno un’occorrenza di “viaggiare”
- “potremo” è sempre immediatamente seguito da almeno un’occorrenza di “divertirci”
- “viaggiare” si alterna con “divertirci”.

Nelle stringhe del linguaggio, quindi, possiamo trovare il pattern (**...si potrà viaggiare*... potremo divertirci* ...**)*

[Nota bene: * sta a indicare che il pattern può esserci o non esserci, e può ripetersi tante volte. Se aiuta a essere veloci, usare la seguente codifica per gli elementi dell'alfabeto V=viaggiare, S=si potrà, P=potremo, T=presto, R=resistiamo, O=siamo ottimisti, D=divertirci]

Al posto dei puntini si possono avere stringhe possibilmente vuote sull'alfabeto {presto}.

Ad esempio le seguenti stringhe appartengono al linguaggio L:

1. “siamo ottimisti resistiamo”
2. “siamo ottimisti si potrà viaggiare viaggiare potremo divertirci resistiamo resistiamo resistiamo”
3. “siamo ottimisti siamo ottimisti presto si potrà viaggiare presto potremo divertirci divertirci divertirci presto presto resistiamo”.
4. “siamo ottimisti si potrà viaggiare potremo divertirci presto presto si potrà viaggiare potremo divertirci resistiamo”.

- b) Mostrare che la stringa “***siamo ottimisti si potrà viaggiare presto potremo divertirci divertirci presto resistiamo resistiamo resistiamo***” può essere generata dalla grammatica da voi definita, scrivendo la catena o l'albero di derivazioni di derivazioni.
- c) Dire se la grammatica definita è ambigua. Se non lo fosse, argomentare la risposta. Se lo fosse, dimostrarlo.

Esercizio2

Dato il linguaggio L definito nell'esercizio precedente, specificare un sistema di transizione per L $S_L = (\Gamma_L, T_L, \rightarrow_L)$ in modo che la semantica di una stringa $s \in L$ sia:

- <turista> se il numero di “viaggiare” è superiore a “divertirsi”
- <goliardico> se il numero di “divertirsi” è superiore a “viaggiare”
- <non chiaro> altrimenti

Per esempio, la semantica di:

- “siamo ottimisti resistiamo” è <non chiaro>
- “siamo ottimisti si potrà **viaggiare viaggiare** potremo divertirci resistiamo resistiamo resistiamo” è <turista>
- “siamo ottimisti siamo ottimisti presto si potrà **viaggiare** presto potremo **divertirci divertirci divertirci** presto presto resistiamo” è <goliardico>

Esercizio 3 (solo per il primo parziale)

Si supponga di estendere la sintassi dei comandi d con del core di Java (dispensa 2.ElementiSemanticaOperazionale) con la nuova produzione:

Com ::= repeat Exp time Com

Il significato intuitivo di repeat E time C è il seguente:

il comando C viene ripetuto tante volte quanto indicato da E (espressione nei numeri naturali).

Si diano le regole di semantica del nuovo comando.

Esercizio 4

Dati i due programmi +/- Java, dire se i due programmi sono equivalenti. Se non lo sono dire in quali stati i due programmi sono debolmente equivalenti. Dimostrare le affermazioni.

PROGRAMMA 1	PROGRAMMA 2
<pre>01. class BottigliaVino { 02. private int costo; 03. private int annata; 04. 05. public BottigliaVino (int x) { 06. this.costo = x; 07. } 08. public int getCosto (int limit) { 09. if (this.annata<limit) this.costo=this.costo*2; 10. return this.costo; 11. } 12. public void setAnnata(int a){ 13. this.annata=a; 14. } 15. } 16. 17.class CassaVino { 18. private int grandezza ; 19. private BottigliaVino bv; 20. 21. public CassaVino (BottigliaVino x) { 22. this.bv = x; 23. this.grandezza = 6 ; 24. } 25. public int getCosto (int l) { 26. BottigliaVino b=this.bv; 27. int costo= this.grandezza*b.getCosto(l); 28. return costo; 29. } 30.} 31. 32.public class Program { 33. public static void main(String[] foo) { 34. int budget = 360; 35. int costoC1; 36. int costoC2; 37. int n=0; 38. int invecchiamento = ?; 39. BottigliaVino b1= new BottigliaVino(10); 40. CassaVino c1= new CassaVino(b1); 41. BottigliaVino b2= new BottigliaVino(20); 42. CassaVino c2= new CassaVino(b2); 43. 44. b1.setAnnata(2015); 45. costoC1=c1.getCosto(invecchiamento); 46. b2.setAnnata(2021); 47. costoC2=c2.getCosto(invecchiamento); 48. 49. while (budget>=costoC1) {n=n+1; 50. budget = budget - costoC1;} 51. } 52. if (budget >= costoC2) n=n+1; 53.}</pre>	<pre>01. class BottigliaVino { 02. private int costo; 03. private int annata; 04. 05. public BottigliaVino (int x) { 06. this.costo = x; 07. } 08. public int getCosto (int limit) { 09. if (this.annata<limit) this.costo=this.costo*2; 10. return this.costo; 11. } 12. public void setAnnata(int a){ 13. this.annata=a; 14. } 15. } 16. 17.class CassaVino { 18. private int grandezza ; 19. private BottigliaVino bv; 20. 21. public CassaVino (BottigliaVino x) { 22. this.bv = x; 23. this.grandezza = 6 ; 24. } 25. public int getCosto (int l) { 26. BottigliaVino b=this.bv; 27. int costo= this.grandezza*b.getCosto(l); 28. return costo; 29. } 30.} 31. 32.public class Program { 33. public static void main(String[] foo) { 34. int budget = 360; 35. int costoC1; 36. int costoC2; 37. int n=0; 38. int invecchiamento = ?; 39. BottigliaVino b1= new BottigliaVino(10); 40. CassaVino c1= new CassaVino(b1); 41. BottigliaVino b2= new BottigliaVino(20); 42. CassaVino c2= new CassaVino(b2); 43. 44. b1.setAnnata(2015); 45. b2.setAnnata(2021); 46. costoC1=c1.getCosto(invecchiamento); 47. costoC2=c2.getCosto(invecchiamento); 48. 49. while (budget>=costoC2) {n=n+1; 50. budget = budget - costoC2;} 51. } 52. if (budget >= costoC1) n=n+1; 53.}</pre>

Esercizio 5

Considerato il PROGRAMMA 1 di +-Java dell'esercizio 4 e assumendo che il “?” di riga 38. venga sostituito con “**2014**”, si mostri:

- a) lo stato risultante dopo l'esecuzione delle righe 1-31 (senza specificare le regole applicate);
- b) lo stato risultante dopo l'esecuzione delle righe 1-44 (senza specificare le regole applicate);
- c) la derivazione di valutazione del comando **costoC1=c1.getCosto(invecchiamento);** di riga 45 a partire dallo stato calcolato al punto b)
- d) lo stato risultante dopo l'esecuzione di tutto il programma e le regole applicate da riga 44 a riga 53.