Prova P2 | **Disciplina: Teoria da Computação**

Professor: Ney Laert Vilar Calazans

Aluno: 04/dezembro/2019

1. (4 pontos) Projete, elabore e desenhe uma máquina de Turing (TM) que aceita a linguagem definida sobre o alfabeto $\mathbf{\Sigma=\{a,b\}}$ formada tão somente por palavras com a forma $\mathbf{a^n(aba)b^n}$, com n≥0, ou seja, a linguagem contendo todas e somente as palavras que têm ao centro a sub-cadeia **aba**, sub-cadeia esta precedida por n **a**s e sucedida por n **b**s. Após elaborar sua TM, responda às seguintes questões sobre ela:

   a) A linguagem aceita por esta TM é regular ou não? Comente a resposta, fornecendo evidências que ela faz sentido no contexto de definição de linguagens formais.
   b) O requisito de alterar a fita é necessário para realizar a TM que aceita esta linguagem? Em outras palavras, seria possível gerar uma TM que aceitasse a mesma linguagem e que não alterasse a fita de entrada?

2. (3,0 pontos). Verdadeiro ou Falso. Abaixo aparecem 10 afirmativas. Marque com V as afirmativas verdadeiras e com F as falsas. Se não souber a resposta correta, melhor deixar em branco, pois cada resposta correta vale 0,3 pontos, mas cada resposta incorreta desconta 0,2 pontos do total positivo de pontos. Não é possível que a questão produza uma nota menor do que 0 pontos.

   a) ( ) Se um dia for provado que **P=NP**, ainda restará resolver o problema de achar soluções eficientes para a classe de problemas NP-Completos, que são os problemas decidíveis mais complexos.
   b) ( ) A linguagem $\mathbf{a^nb^na^n}$ não pode ser reconhecida por um autômato de pilha (1PDA).
   c) ( ) Se um problema de decisão **X** da classe **NP** pode ser transformado em uma instância de SAT, o primeiro problema provado ser NP-completo, **X** é NP-Completo.
   d) ( ) Uma máquina de Turing com fita infinita para ambos os lados não quebra se o cursor avançar à esquerda da posição inicial deste, o que faz com que ela seja capaz de resolver problemas mais complexos.
   e) ( ) Dado um problema **X**, assuma que $\mathbf{X=O(n^2)}$ e $\mathbf{X=\omega(n)}$. Então, pode-se afirmar que $X=\Theta(\mathbf{n^2})$.
   f) ( ) Um enunciado simples do problema do Caixeiro Viajante (*Traveling Salesman*) é: "Dado um conjunto de **n** cidades $\mathbf{C=\{c_0, c_1, ..., c_n\}}$, e o conjunto de todas as distâncias entre cidades $\mathbf{D=\{(c_i,c_j) \mid c_i, c_j \in C\}}$, achar o caminho mais curto para partir de uma cidade qualquer $\mathbf{c_i}$, visitar todas as demais cidades e voltar ao ponto de partida". Assim enunciado, este problema caracteriza-se como um problema de decisão.
   g) ( ) A função $\mathbf{f=325n^2+300.456n+1.000.000}$ atende à sentença matemática $\mathbf{f=O(n^2)}$.
   h) ( ) A função $\mathbf{f=325n^2+300.456n+1.000.000}$ atende à sentença matemática $\mathbf{f=o(n^3)}$.
   i) ( ) A comparação das classes de problemas P e NP não é simples porque cada uma destas classes é definida em termos de duas maneiras distintas de resolver problemas.
   j) ( ) Não existem problemas de decisão decidíveis classificáveis fora das classes P ou NP. Em outras palavras, todo problema de decisão decidível pode ser provado como pertencendo a P, a NP ou a ambas.

3. (1 ponto) Observe a sentença abaixo:
   ALAN é uma linguagem recursivamente enumerável,
   portanto
   ALAN não pode ser reconhecida por nenhuma máquina de Turing.

É correto afirmar:
   a) **Que as duas afirmativas estão corretas e que a segunda é uma decorrência da primeira.**
   b) **Que as duas afirmativas estão corretas, mas a segunda não está relacionada com a primeira.**
   c) **Que a primeira afirmativa é correta e a segunda é falsa.**
   d) **Que a segunda afirmativa é correta e a primeira é falsa.**
   e) **Que nenhum dos quatro itens acima está correto.**

4. (1 ponto) Observe a sentença abaixo:
   P é o conjunto de problemas de decisão que possuem uma solução computável em tempo polinomial,
   portanto
   NP é a classe de problemas de decisão que podem ser verificados em tempo polinomial.

É correto afirmar:
   a) **Que as duas afirmativas estão corretas e que a segunda é uma decorrência da primeira.**
   b) **Que as duas afirmativas estão corretas, mas a segunda não está relacionada com a primeira.**
   c) **Que a primeira afirmativa é correta e a segunda é falsa.**
   d) **Que a segunda afirmativa é correta e a primeira é falsa.**
   e) **Que nenhum dos quatro itens acima está correto.**

5. (1 ponto) Observe a sentença matemática abaixo (uma implicação):
   Se um algoritmo A é tal que $A=O(n^{100})$,
   então
   $A \in NP$.

É correto afirmar:
   a) **Que o antecedente justifica o consequente.**
   b) **Que o consequente está incorreto.**
   c) **Que se o antecedente está incorreto, ainda assim o consequente está correto.**
   d) **Que a segunda afirmativa significa que A é NP-completo.**
   e) **Que nenhum dos quatro itens acima está correto.**

Prova P2 **Disciplina: Teoria da Computação**

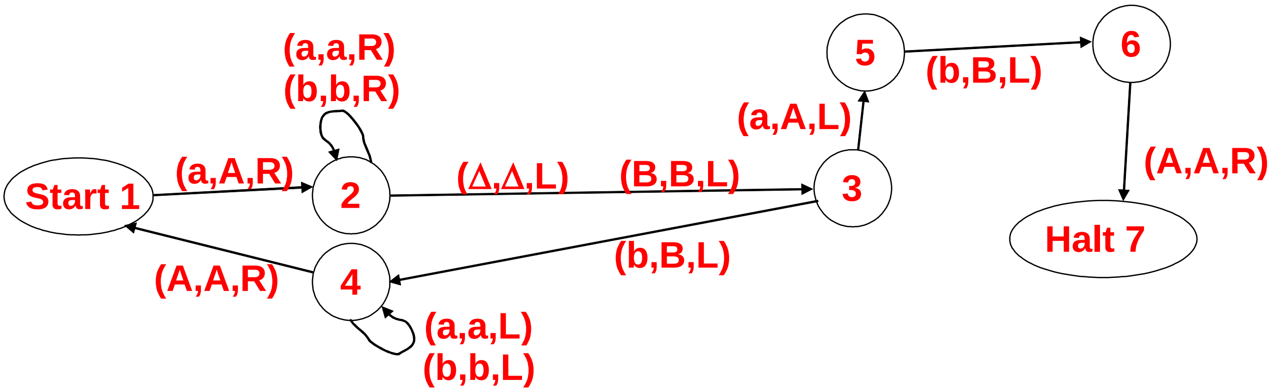
Professor: Ney Laert Vilar Calazans

Aluno: **Gabarito** 04/dezembro/2019

1. (4 pontos) Elabore e desenhe uma máquina de Turing (TM) que aceita a linguagem definida sobre o alfabeto **Σ={a,b}** formada tão somente por palavras com a forma $\mathbf{a^n(aba)b^n}$, com n≥0, ou seja, a linguagem contendo todas e somente as palavras que tem ao seu centro a sub-cadeia **aba**, sub-cadeia esta precedida por n **a**s e sucedida por n **b**s. Após elaborar sua TM, responda às seguintes questões sobre ela:

   a) A linguagem aceita por esta TM é regular ou não? Comente a resposta, fornecendo evidências que ela faz sentido no contexto de definição de linguagens formais.
   b) O requisito de alterar a fita é necessário para realizar a TM que aceita esta linguagem? Em outras palavras, seria possível gerar uma TM que aceitasse a mesma linguagem e que não alterasse a fita de entrada?

Solução: Uma TM que aceita a linguagem em questão é dada abaixo.



Respostas aos itens e observações sobre esta TM:

a) **A linguagem não é regular, pois com um número finito de estados em um FA não seria possível reconhecer qualquer palavra com qualquer valor de n as e bs;**

b) **Sim, para memorizar a contagem dos n as e bs, seguindo as premissas do lema do bombeamento; Logo, não seria possível realizar uma TM que não alterasse a fita e ainda assim fosse capaz de reconhecer esta linguagem.**

2. (3,0 pontos). Verdadeiro ou Falso. Abaixo aparecem 10 afirmativas. Marque com V as afirmativas verdadeiras e com F as falsas. Se não souber a resposta correta, melhor deixar em branco, pois cada resposta correta vale 0,3 pontos, mas cada resposta incorreta desconta 0,2 pontos do total positivo de pontos. Não é possível que a questão produza uma nota menor do que 0 pontos.

   a) (**F**) Se um dia for provado que **P=NP**, ainda restará resolver o problema de achar soluções eficientes para a classe de problemas NP-Completos, que são os problemas decidíveis mais complexos.
   Explanação: Ora, por definição NP-Completos ⊆ NP. Logo, se P=NP, todos os problemas NP-Completos possuem solução em tempo polinomial, ou NP-Completos ⊆ P.
   b) (**V**) A linguagem $\mathbf{a^n b^n a^n}$ não pode ser reconhecida por um autômato de pilha (1PDA).

Explanação: Esta linguagem é sensível ao contexto, e não livre do contexto. Ela pode ser reconhecida por uma máquina de Post ou por uma TM ou por um 2PDA, mas não por um 1PDA.

c) (**F**) Se um problema de decisão **X** da classe **NP** pode ser transformado em uma instância de SAT, o primeiro problema provado ser NP-completo, **X** é NP-Completo.
Explanação: São duas condições para demonstrar que um problema X é NP-Completo (NPC): (i) um problema NPC (como SAT) precisa ser mapeável para X; e (ii) X precisa ser mapeável para um problema sabido ser NPC (como SAT). Se apenas uma das condições é demonstrada, não se pode afirmar com isto que X seja NPC.

d) (**F**) Uma máquina de Turing com fita infinita para ambos os lados não quebra se o cursor avançar à esquerda da posição inicial deste, o que faz com que ela seja capaz de resolver problemas mais complexos.
Explanação: Como visto ao estudar a tese de Church qualquer problema solúvel é solúvel por uma TM com fita infinita em uma direção. Assim, tornar a fita infinita em duas direções não aumenta o poder computacional (mas pode aumentar a eficiência das soluções de problemas, mas isto é outro assunto...).

e) (**F**) Dado um problema **X**, assuma que $\mathbf{X=O(n^2)}$ e $\mathbf{X=\omega(n)}$. Então, pode-se afirmar que $X=\Theta(\mathbf{n^2})$.
Explanação: Não, não se pode afirmar isto, pois a definição de $\Theta(n^2)$ implica limites estritos acima e abaixo de X. Além da questão usar limite inferior não estrito, ela usa uma função com crescimento assintoticamente mais lento.

f) (**F**) Um enunciado simples do problema do Caixeiro Viajante (*Traveling Salesman*) é: "Dado um conjunto de **n** cidades $\mathbf{C=\{c_0, c_1, \ldots, c_n\}}$, e o conjunto de todas as distâncias entre cidades $\mathbf{D=\{(c_i,c_j) \mid c_i, c_j \in C\}}$, achar o caminho mais curto para partir de uma cidade qualquer $\mathbf{c_i}$, visitar todas as demais cidades e voltar ao ponto de partida". Assim enunciado, este problema caracteriza-se como um problema de decisão.
Explanação: Um problema de decisão tem que aceitar como solução uma resposta Sim ou Não, o que não é o caso deste enunciado.

g) (**V**) A função $\mathbf{f=325n^2+300.456n+1.000.000}$ atende à sentença matemática $\mathbf{f=O(n^2)}$.
Explanação: Sim, imagine uma constante c=1.000.000. A função $1.000.000n^2$ vai claramente assumir valores maiores que o polinômio da f a partir de um dado valor n_0 que se pode determinar.

h) (**V**) A função $\mathbf{f=325n^2+300.456n+1.000.000}$ atende à sentença matemática $\mathbf{f=o(n^3)}$.
Explanação: Sim, mesmo uma constante pequena, digamos c=0.0001 multiplicada por n^3 um dia vai ultrapassar o valor do polinômio f a partir de um dado valor n_0 que se pode determinar. Além disto, o crescimento assintótico de n^3 é muito mais pronunciado que o de n^2.

i) (**V**) A comparação das classes de problemas P e NP não é simples porque cada uma destas classes é definida em termos de duas maneiras distintas de resolver problemas.
Explanação: De fato, P é definido em termos de solubilidade em tempo polinomial e NP em termos da verificabilidade em tempo polinomial de uma solução, formas bem distintas de classificar problemas.

j) (**F**) Não existem problemas de decisão decidíveis classificáveis fora das classes P ou NP. Em outras palavras, todo problema de decisão decidível pode ser provado como pertencendo a P, a NP ou a ambas.
Explanação: Existem sim, há múltiplos exemplos de problemas que não podem ser verificáveis em tempo polinomial, mas podem ser resolvidos e/ou verificados em tempo não-polinomial. Sim, existe vida fora de NP...

3. (1 ponto) Observe a sentença abaixo:
   ALAN é uma linguagem recursivamente enumerável,
   portanto
   ALAN não pode ser reconhecida por nenhuma máquina de Turing.

É correto afirmar:

a) **Que as duas afirmativas estão corretas e que a segunda é uma decorrência da primeira.**

b) **Que as duas afirmativas estão corretas, mas a segunda não está relacionada com a primeira.**
c) **Que a primeira afirmativa é correta e a segunda é falsa.**
d) **Que a segunda afirmativa é correta e a primeira é falsa.**
e) **Que nenhum dos quatro itens acima está correto.**
Explanação: ALAN foi o exemplo desenvolvido em aula de uma linguagem que não pode ser reconhecida por nenhuma máquina de Turing, logo ela não é uma linguagem recursivamente enumerável. A segunda afirmação é então correta e a primeira é falsa.

4. (1 ponto) Observe a sentença abaixo:
P é o conjunto de problemas de decisão que possuem uma solução computável em tempo polinomial,
portanto
NP é a classe de problemas de decisão que podem ser verificados em tempo polinomial.

É correto afirmar:

a) **Que as duas afirmativas estão corretas e que a segunda é uma decorrência da primeira.**
b) **Que as duas afirmativas estão corretas, mas a segunda não está relacionada com a primeira.**
c) **Que a primeira afirmativa é correta e a segunda é falsa.**
d) **Que a segunda afirmativa é correta e a primeira é falsa.**
e) **Que nenhum dos quatro itens acima está correto.**
Explanação: As afirmativas são corretas, mas as classes de problemas não estão relacionadas como um par causa-consequência. De fato, um problema de P pode sempre ser verificado em tempo polinomial.

5. (1 ponto) Observe a sentença matemática abaixo (uma implicação):
Se um algoritmo A é tal que A=O(n^{100}),
então
A $\in$ NP.

É correto afirmar:

a) **Que o antecedente justifica o consequente.**
b) **Que o consequente está incorreto.**
c) **Que se o antecedente está incorreto, ainda assim o consequente está correto.**
d) **Que a segunda afirmativa significa que A é NP-completo.**
e) **Que nenhum dos quatro itens acima está correto.**
Explanação: Um algoritmo resolve um problema. Se ele é O(n^{100}) ele pode ser resolvido em tempo polinomial (pois n^{100} é um polinômio), logo o mesmo pode ser verificado em tempo polinomial, logo A $\in$ NP.