Exercícios de Revisão

Questões de Formalização:

Estruturas de Dados - Propriedades de Max-Heap

- $A[i]$ representa o valor armazenado no nó de índice i do array que implementa a Max-Heap;
- $Parent(i) = \lfloor i/2 \rfloor$ é a função que retorna o índice do pai do nó de índice i;
- HeapSize é o tamanho total da heap.

$$\boxed{\forall i \in \{i > 1 \land i \leq HeapSize\},\ A[Parent(i)] \geq A[i]}$$

Banco de Dados Relacional - Dependência Funcional

- Seja R um esquema de relação com atributos $A_1, A_2, \dots, A_n$
- Seja $X \subseteq R$ e $Y \subseteq R$ subconjuntos dos atributos de R
- Seja t_1, t_2 duas tuplas de R, e $t_1[A]$ denotando o valor da tupla t_1 para o atributo A

$$\boxed{\forall t_1, t_2 \in R,\ (t_1[X] = t_2[X]) \rightarrow (t_1[Y] = t_2[Y])}$$

Matemática Discreta / Teoria dos grafos - Fechamento Transitivo

- Seja $R \subseteq A \times A$ uma relação binária sobre o conjunto A;
- O fechamento transitivo de R, denotado por R^+, é a relação que inclui todos os pares (a, b) tais que existe uma sequência de elementos $x_0, x_1, \ldots, x_k \in A$ com $k > 1$ e $(x_i, x_{i+1}) \in R$ para $i = 0, 1, \ldots, k-1$, com $x_0 = a$ e $x_k = b$.

$$\boxed{(a,b) \in R^+ \longleftrightarrow \exists x_0, x_1, \ldots, x_k \in A \mid x_0 = a,\ x_k = b \land (x_i, x_{i+1}) \in R\ \forall i = 0, 1, \ldots, k-1}$$

Teoria da Complexidade - NP-Completude

- Pertencer à classe NP: O problema L deve pertencer a NP, ou seja, a solução proposta deve ser verificada em tempo polinomial;
- Ser difícil (NP-hard): Todo outro problema $L' \in NP$ pode ser reduzido a L em tempo polinomial.

$$\boxed{L \in NP \land \forall L' \in NP,\ L' \leq_p L}$$

Redes de Computadores - Longest Prefix Match (LPM)

- Corresponde(D, p): Função que verifica se o endereço IP D corresponde ao prefixo p;
- Comprimento(p): Função que retorna o número de bits do prefixo p;
- S_{out} é a saída associada ao prefixo p_{out}.

$$\boxed{\begin{gathered} S_{out} = s, \text{ onde } (p_{out}, s) \in T \land \text{Corresponde}(D, p_{out}) \land \forall (p, s) \in T \mid \text{Corresponde}(D, p), \\ \text{Comprimento}(p_{out}) \geq \text{Comprimento}(p) \end{gathered}}$$

Questões de Verificação

Knowledge Vault - Fato Altamente Credível

- $\exists s_1, s_2 \in S$: existem pelo menos duas fontes s_1 e s_2 no conjunto S;
- $s_1 \neq s_2$: as fontes são diferentes entre si;
- $Realible(s_1) \wedge Realible(s_2)$: as fontes s_1 e s_2 são ~~independentes~~ confiáveis;
- $Independent(s_1, s_2)$: as fontes s_1 e s_2 são independentes entre si;
- $Asserts(s_1, f) \wedge Asserts(s_2, f)$: as fontes s_1 e s_2 afirmam o fato f.

$$\boxed{HighlyCredible(f) \leftrightarrow \exists s_1, s_2 \in S \mid s_1 \neq s_2 \wedge Realible(s_1) \wedge Realible(s_2) \wedge Independent(s_1, s_2) \wedge Asserts(s_1, f) \wedge Asserts(s_2, f)}$$

- Portanto, a formalização está <u>correta</u>.

Álgebra Relacional - Junção Natural

- $r \in R$ e $s \in S$ são tuplas de R e S respectivamente;
- $\forall a \in CommonAttr, r[a] = s[a]$ significa que para todos os atributos comuns a, os valores de r e s para a devem ser iguais;
- $t = r \cup s$ significa que a tupla resultante t é formada pela união dos atributos de r e s.

$$\boxed{R \bowtie S = \{t \mid \exists_r \in R, \exists_s \in S \mid (\forall a \in CommonAttr, r[a] = s[a]) \wedge (t = r \cup s)\}}$$

- Portanto, a formalização está <u>incorreta</u>.

Formal Methods / Alloy - Invariante de Sistemas de Arquivos

- Dir: é o conjunto de todos os diretórios;
- RootDir: é o diretório raiz;
- parent(d): é o diretório pai do diretório d.

$$\boxed{\forall d \in Dir \setminus RootDir \mid (\exists p \in Dir \mid parent(d) = p)}$$

- Portanto, a formalização está correta.

Agentes Inteligentes BDI - Consistência Crença - Intenção

- A formula diz que não pode acontecer que um agente tenha a intenção de alcançar φ e ao mesmo tempo acreditar que φ é impossível;
- Ou seja, se o agente a tem a intenção de alcançar φ e acredita que φ é impossível, isso é uma contradição.

$$\boxed{\forall a, \forall \varphi : (INT(a, \varphi) \land BEL(a, \varphi)) \rightarrow \bot}$$

- Portanto a formalização está incorreta.

Redes de Computadores - Longest Prefix Match (LPM)

- A função matches seleciona os prefixos que correspondem ao Destino D
- A formalização busca o prefixo mais longo entre os correspondentes
- A saída s_{out} é corretamente associada ao prefixo mais longo encontrado

$$\boxed{(p_{out}, s_{out}) \in Matches(D, T) \land (\forall (p', s') \in Matches(D, T) \mid Comprimento(p_{out}) \geq Comprimento(p'))}$$

- Portanto, a formalização está correta.