Teste Intermédio da UC de Teoria da Computação (LEI, FCT UNL)

7 Novembro 2012

Nome:	**Número:**

Grupo I

Nas perguntas 1 a 3, deverá escolher a opção correcta e justificar brevemente.

1. Defina o conjunto de todos os números racionais que multiplicados por 2 ou por 3 dão um número inteiro.

   (a) $\{r \in RAT \mid \exists n, p.\ n, p \in NAT \wedge\ r = n/p \wedge \exists k.\ k \in INT \wedge 2 * n = k * p\}$
   (b) $\{r \in RAT \mid \exists n, p.\ n, p \in NAT \wedge\ r = n/p \wedge \exists k.\ k \in INT \wedge (2 * n = k * p \wedge 3 * n = k * p)\}$
   (c) $\{r \in RAT \mid \exists n, p.\ n, p \in NAT \wedge\ r = n/p \wedge \exists k.\ k \in NAT \wedge 2 * n = k * 3\}$
   (d) $\{r \in RAT \mid \exists n, p.\ n, p \in NAT \wedge\ r = n/p \wedge \exists k.\ k \in INT \wedge (2 * n = k * p \vee 3 * n = k * p)\}$
   (e) Nenhuma das anteriores.

2. Defina a função g que mapeia cada conjunto de números naturais $X \subset \wp(NAT)$ no seu subconjunto de números que são *emparelhados* com outro número no conjunto X. Dois números n e m são emparelhados quando consecutivos. Por exemplo 3 é emparelhado com 4, e 7 é emparelhado com 6. Por exemplo $g(\{7, 3, 4, 1\}) = \{3, 4\}$, $g(\{3, 1, 10, 8, 2, 11\}) = \{1, 2, 3, 10, 11\}$.

   (a) $g =_{def} \{s \mapsto s' \in \wp(NAT) \times \wp(NAT) \mid s' = \{n \in s \mid \exists k \in s.k = n + 1\}\}$
   (b) $g =_{def} \{s \mapsto s' \in \wp(NAT) \times \wp(NAT) \mid s' = \{n \in s \mid \exists k \in s.k = n + 1 \wedge n = k + 1\}\}$
   (c) $g =_{def} \{s \mapsto s' \in \wp(NAT) \times \wp(NAT) \mid s' = \{n \in s \mid \exists k \in s.k = n + 1 \vee n = k + 1\}\}$
   (d) $g =_{def} \{s \mapsto s' \in \wp(NAT) \times \wp(NAT) \mid \exists k \in s'.\exists n \in s.k = n + 1 \vee n = k + 1\}$
   (e) Nenhuma das anteriores.

3. Defina a função parcial M que recebendo um conjunto não vazio X de números NAT devolve o seu elemento mínimo. Por exemplo $M(\{9, 2, 4, 10, 1\}) = 1$.

   (a) $M =_{def} \{s \mapsto k \in \wp(NAT) \times NAT \mid \forall x \in s.k > x\}$
   (b) $M =_{def} \{s \mapsto k \in \wp(NAT) \times NAT \mid k \in s \wedge \forall x \in s.k \leq x\}$
   (c) $M =_{def} \{s \mapsto k \in \wp(NAT) \times NAT \mid \forall x \in s.k \leq x\}$
   (d) $M =_{def} \{s \mapsto k \in \wp(NAT) \times NAT \mid k \in s \Rightarrow \forall x \in s.k \leq x\}$
   (e) Nenhuma das anteriores.

4. Considere o conjunto $SUITS \triangleq \{\clubsuit, \heartsuit, \spadesuit, \diamondsuit\}$.

   (a) Defina a função $LEWIS$ que dado um par $(S, n) \in \wp(SUITS) \times SUITS$ devolve o conjunto P de todos os subconjuntos de S que não contêm o naipe n. Por exemplo, se $S = \{\clubsuit, \heartsuit, \spadesuit\}$ e $n = \heartsuit$ então $LEWIS(S, n) = \{\{\clubsuit, \spadesuit\}, \{\clubsuit\}, \{\spadesuit\}, \{\}\}$.
   (b) Indique, justificando se os seguintes conjuntos são finitos, infinitos contáveis, ou infinitos não contáveis, indicando o seu cardinal ($k \in NAT$, ω - contável ou $\mathfrak{c}$ - não contável).

i. $NAT \times SUITS$.
ii. $\wp(SUITS)$.
iii. $NAT \times \wp(SUITS)$.
iv. $\wp(NAT) \times SUITS$.

Grupo II

Um sistema de ficheiros mantém informação sobre ficheiros, directorias, e os ficheiros que estas contêm. Assuma que cada directoria consiste num nome $n \in STRING$ e num conjunto finito que representa os items que contém. Assuma também que cada ficheiro consiste apenas num nome (vamos ignorar o conteúdo dos ficheiros).

Para simplificar, vamos também admitir que o sistema de ficheiros aceita apenas um nível de directorias, ou seja apenas poderão existir directorias na raiz do sistema de ficheiros.

O conjunto de estados do sistema de ficheiros pode ser modelado com o seguinte conjunto FS.

$$FS = \wp(STRING \cup (STRING \times \wp(STRING)))$$

Exemplo de um estado $f \in FS$: $f = \{``boot", ``config", (``users", \{``joe", ``mary"\}), (``etc", \{``passwd"\})\}$ O estado f tem quatro elementos, duas directorias (“*users*” e “*pwd*”) e dois ficheiros na raiz (“*boot*” e “*config*”). A directoria “*users*” contém os ficheiros “*joe*” e “*mary*”.

Nota: se lhe for conveniente definir funções ou relações auxiliares, poderá fazê-lo.

1. Defina (com uma função ou relação) a operação *mkdir* que cria uma nova directoria vazia (na raiz) com um nome indicado $n \in STRING$. N.B.: Se já existir um ficheiro ou directoria com o mesmo nome na raiz, a operação não deve fazer nada.

2. Defina (com uma função ou relação) a operação *mkfs* que apaga todo o conteúdo do sistema de ficheiros.

3. Defina (com uma função ou relação) a operação *mkfiled* que cria um ficheiro com o caminho (path) M. O caminho M é dado por um par $(d, f) \in (STRING \times STRING)$ onde d indica o nome da directoria e f o nome do ficheiro. Se a directoria d não existir, ou se já existir um ficheiro com o nome f na directoria d, a operação não deve fazer nada.

4. Defina (com uma função ou relação) a operação *rm* que apaga o ficheiro com o caminho indicado M. Neste caso, o caminho M pode ser dado como um $f \in STRING$ ou como um par $(d, f) \in (STRING \times STRING)$, consoante o ficheiro f estiver na raiz ou na directoria de nome d.

5. Neste exercício, para simplificar, assumimos que o sistema de ficheiros tinha apenas dois níveis (o da raiz, e o das directorias debaixo da raiz). Se quisessemos modelar um sistema de ficheiros sem esta restrição, para suportar qualquer número de subdirectorias embutidas, como poderiamos definir o seu conjunto de estados GFS? (Sugestão: use uma definição indutiva do conjunto GFS).

Grupo III

1. Considere o seguinte alfabeto (com 4 símbolos) $DICE = \{\texttt{face1}, \texttt{face2}, \texttt{face3}, \texttt{face4}\}$ que representa um dado de 4 lados. Neste exercício considere que o dado é lançado várias vezes em sequência.

   (a) Especifique um autómato finito determinista (DFA) que só aceite sequências de lançamentos que terminem com uma ocorrência de `face4`.

   (b) Especifique um DFA que só aceite sequências de lançamentos nos quais exista pelo menos uma ocorrência de `face2 face3`.

   (c) Especifique um DFA que só aceite sequências de lançamentos em que cada ocorrência de `f3` é seguida de imediato por um número impar de ocorrências de `face1`.

   Ou seja, a palavra `face2 face3 face1 face1 face1 face4` deve ser aceite, enquanto a palavra `face1 face3 face1 face1 face4 face1` deve ser rejeitada.

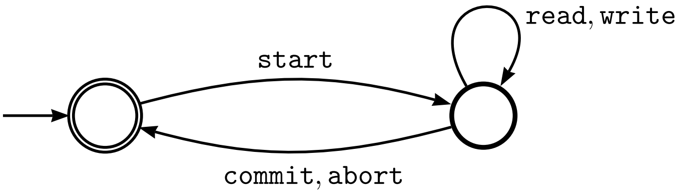
(d) Especifique um DFA que só aceite sequências de lançamentos com valores crescentes. Por exemplo, o lançamento `face2 face2 face3` tem valores crescentes.

2. Considere o alfabeto $\mathit{TRANS} = \{\texttt{start}, \texttt{read}, \texttt{write}, \texttt{commit}, \texttt{rollback}, \texttt{abort}\}$ usado para definir transacções sobre uma base de dados.

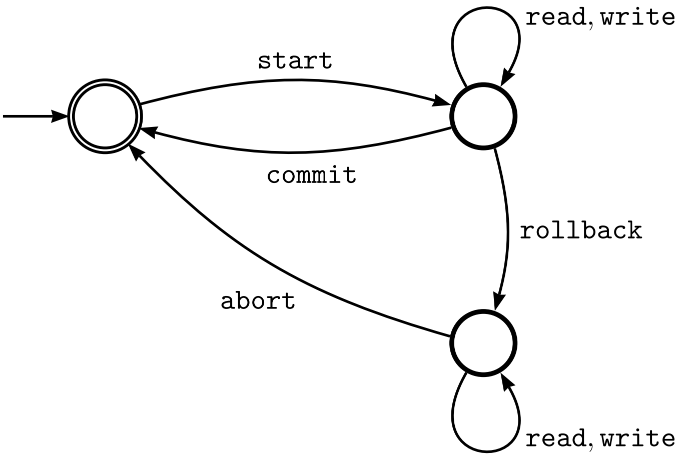
   Para cada um dos seguintes dois DFAs, proponha uma expressão regular sobre TRANS que denote a linguagem por si reconhecida.

   (Não precisa de usar nenhuma técnica sofisticada para resolver este exercício: queremos basicamente avaliar se consegue perceber a linguagem reconhecida pelo autómato, e depois exprimi-la através de uma expressão regular).

(a)



(b)



3. Indique quais das seguintes palavras são aceites pelo autómato finito determinista da alínea 2b:

   (a) ϵ
   (b) `start commit`
   (c) `start read read rollback commit`
   (d) `start read read rollback write abort`
   (e) `start read commit start rollback abort`
   (f) `start commit read write commit`