

## Enunciado do Terceiro Trabalho Prático

**Entrega e Apresentação:** 16 de Maio de 2012, 14h

1. Resolve **um** dos problemas seguintes relacionados com “model checking” para LTL ou CTL. Quando tiveres decidido qual o problema que vais resolver envia um email para `nam@dcc.fc.up.pt` indicando o tema. As linguagens de programação preferenciais são Haskell e Python, mas podes usar também Java, C ou C++. Cada programa deverá ter um pequeno relatório explicativo (ou comentários claros no código) e ser acompanhado de alguns exemplos que demonstrem o seu funcionamento. Excepto, possivelmente, o problema de todos os problemas são para ser resolvidos individualmente.

**Nota importante:** Para nenhum dos problemas é necessário considerar análise sintáctica dos dados, apenas ter estruturas internas que representem os diferentes dados. Mais uma vez se lembra que **cópias** levarão à exclusão da disciplina e a um possível processo disciplinar.

(a) Implementa os seguintes algoritmos:

- i) Construção do autómato alternado de Büchi  $\mathcal{A}_{\neg\phi}$  a partir de uma fórmula em LTL,  $\phi$  (usando as conectivas temporais X, U, F e G).
- ii) Algoritmo para determinar se um autómato alternado de Büchi aceita a linguagem vazia.
- iii) Método de conversão entre autómatos alternados de Büchi e autómatos não determinísticos de Büchi (devem ser implementadas as duas conversões).

**Referência:** [Var94]

(b) Implementa os seguintes algoritmos:

- i) Construção do autómato não determinístico de Büchi  $\mathcal{A}_{(\mathcal{M},s)}$  a partir de um modelo  $\mathcal{M}$  e um estado  $s$ .
- ii) Autómato não determinístico de Büchi equivalente à intersecção de dois autómatos não determinísticos de Büchi.
- iii) Algoritmo para determinar se um autómato não determinístico de Büchi aceita a linguagem vazia.

**Referência:** [Var94]

**Nota** Caso haja dois alunos que implementem a e b, depois poderão juntar os trabalhos de modo a implementarem o algoritmo completo de model checking para LTL.

- (c) Método alternativo de implementar o model checking de LTL usando autómatos de Büchi generalizados. Caso pretenda este tema será melhor contactar a docente da disciplina.

**Referência:** [DG11]

- (d)
  - i) Algoritmo para obter um OBDD reduzido (a partir de um não reduzido ou duma fórmula proposicional), com indicação da ordenação de variáveis.
  - ii) Implementar as funções `apply` e `restrict`, com aplicação para as operações booleanas  $+$ ,  $\neg$  e  $..$
  - iii) (\*) Construção de um OBDD a partir de um modelo  $M$  e implementação dos conjuntos  $pre_{\forall}$  e  $pre_{\exists}$ .
  - iv) (\*) Algoritmo de etiquetagem para formulas CTL apenas com as conectivas temporais  $EX$ ,  $EU$  e  $EG$  (e um subconjunto completo de operadores proposicionais).

**Referência:** [HR04] e [And99] (esta última tem a descrição mais pormenorizada dos algoritmos para as duas primeiras alíneas).

(\*) Caso o trabalho seja feito em grupo de dois alunos.

- (e) Implementação do algoritmo de etiquetagem para CTL para fórmulas gerais (todas as conectivas consideradas nas aulas) com fairness (sem uso de BDDs).

**Referência:** [HR04]

2. Considera o modelo  $\mathcal{M} = (S = \{s_0, s_1, s_2, s_3\}, \{s_0 \rightarrow s_2, s_0 \rightarrow s_1, s_1 \rightarrow s_1, s_1 \rightarrow s_2, s_1 \rightarrow s_3, s_2 \rightarrow s_0, s_2 \rightarrow s_1, s_2 \rightarrow s_2, s_3 \rightarrow s_0, s_3 \rightarrow s_3\}, L(s_0) = \{x_1, x_2\}, L(s_1) = \{x_1\}, L(s_2) = \{\}, L(s_3) = \{x_2\})$ .
  - (a) Utilizando a ordem  $[x_1, x_2]$ , determina OBDD's para representar os conjuntos de estados  $\{s_0, s_1\}$  e  $\{s_0, s_2\}$ .
  - (b) Determina a tabela de verdade para a relação de transição utilizando a ordem  $[x_1, x'_1, x_2, x'_2]$ .
  - (c) Desenha o OBDD para a relação de transição (utilizando a ordem da alínea anterior).
  - (d) Aplica o algoritmo de etiquetagem (adaptado à representação por OBDD's e utilizando a ordem  $[x_1, x_2]$ ) ao modelo  $\mathcal{M}$ , para determinar os conjuntos de estados onde se verificam respectivamente as fórmulas seguintes.
    - $AG(x_1 \vee \neg x_2)$ ;
    - $E[x_2 U x_1]$
3. Considerando o mesmo modelo da pergunta anterior  $\mathcal{M}$ ,
  - (a) Representa  $(\mathcal{M}, s_0)$  por um autómato não determinístico de Büchi  $\mathcal{A}_{(\mathcal{M}, s_0)}$ .
  - (b) Determina um autómato alternado de Büchi  $\mathcal{A}_{\neg(x_1 U x_2)}$  para  $\neg(x_1 U x_2)$ .
  - (c) Usando os autómatos anteriores determina justificando se  $\mathcal{M}, s_0 \models x_1 U x_2$ ?

## Referências

- [And99] H. R. Andersen. An introduction to binary decision diagrams. *Lecture notes*, 1999.
- [DG11] Stéphane Demri and Paul Gastin. Specification and verification using temporal logics. In Deepak D'Souza and Priti Shankar, editors, *Modern applications of automata theory*, volume 2 of *IISc Research Monographs*. World Scientific, December 2011.
- [HR04] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and reasoning about systems*. CUP, 2004.
- [Var94] M. Vardi. An automata-theoretic approach to linear temporal logic. In *Banff'94*, 1994.