

Verificação de Software e Sistemas
Bounded Model Checking de Programas Multitarefa
Quinta Lista de Exercícios

Responda as seguintes perguntas com base no texto do artigo: “Cordeiro, L. and Fischer, B. *Verifying Multi-threaded Software using SMT-based Context-Bounded Model Checking*. In Intl. Conf. on Software Engineering (ICSE), pp. 331-340, IEEE/ACM, 2011.”

Um dado sistema de automação industrial consiste em monitorar e controlar um ambiente externo usando sensores, atuadores e outras interfaces de entrada/saída. Usualmente, tal sistema pode ser implementado através de um programa concorrente que consiste basicamente de uma coleção de processos computacionais que executam em paralelo e que podem interagir entre si. Para este caso em particular, considere que o sistema de automação foi implementado usando dois processos (i.e., P_1 e P_2) e cada processo consiste de dois comandos (e.g., a_1 e b_1 para o processo P_1):

P_1 :	P_2 :
a_1	a_2
b_1	b_2

Note que um programa concorrente pode produzir diferentes linhas de execução (i.e., intercalações) dependendo do algoritmo de escalonamento utilizado. O número de intercalações é exponencial no número de processos e comandos. Exemplos de possíveis intercalações são:

a_1, b_1, a_2, b_2 ;
 a_2, b_2, a_1, b_1 ;
 a_1, a_2, b_1, b_2 .

Devido ao critério de consistência sequencial, você não encontrará uma intercalação onde, por exemplo, b_1 executa antes de a_1 e que a escrita ocorra antes da leitura. De posse destas informações, você deve responder as seguintes questões:

- 1) Quais são todas as possíveis intercalações deste sistema?
- 2) Implementa um programa em C/C++ baseado no algoritmo da *lazy exploration* para imprimir todas as possíveis intercalações. Qual é a complexidade desta sua solução?
- 3) Considere que as ações de todos os processos consistem basicamente em incrementar uma variável global "x", que é inicializada para zero antes de executar todos os processos. Neste caso, você geraria intercalações redundantes (i.e., intercalações que sempre produzem o mesmo resultado). Implemente um programa em C/C++ para eliminar as intercalações redundantes. Qual seria a complexidade desta sua solução?

Data da entrega: 02 de junho de 2016 (quinta-feira).

Após esta data será descontado 2 pontos por dia de atraso.

A lista de exercícios deve ser resolvida e entregue individualmente.

24/05/2016