CS 525, Formal Methods for System Design
Mid-semester Exam, Winter 2018-2019
Department of Computer Science and Engineering
IIT Guwahati
Time: Two hours

Important

1. No questions about the paper will be entertained during the exam.
2. You must answer each question in the space provided for that question in the **answer sheet**. Answers appearing outside the space provided will not be considered.
3. Keep your rough work separate from your answers. A supplementary sheet is being provided for rough work. **Do not attach your rough work to the answer sheet.**
4. This exam has 4 questions over 4 pages, with a total of 100 marks.
5. **Write your roll number at the top of every page in the answer sheet.**

1. Is the following formula satisfiable in the theory of integer linear arithmetic? If it is satisfiable, then give a model for the formula (*i.e.*, a variable assignment that makes the formula true). If not, then justify your answer.

$$(x - y \leq 2) \ \wedge \ (y - z \leq -1) \ \wedge \ (z - x \leq -1).$$

(10)

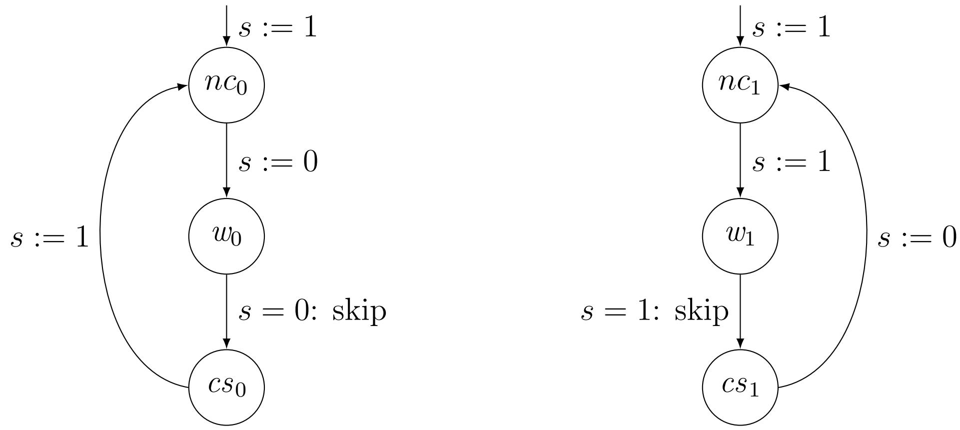
Solution: Satisfiable. Take the variable assignment $\{x \mapsto 2, y \mapsto 0, z \mapsto 1\}$.

2. The following is a proposed algorithm for mutual exclusion for two processes P_0 and P_1. The pseudo-code for P_i for $i = 0, 1$ is given below. Here the single shared variable s is either 0 or 1, and is initially set to 1.

```
ℓ0: loop forever do
 begin
   ℓ1: Noncritical Section
   ℓ2: s := i;
   ℓ3: wait until s = i;
   ℓ4: Critical Section
   ℓ5: s := 1 − i
 end
```

(a) Give the program graph representations for processes P_0 and P_1. (10)

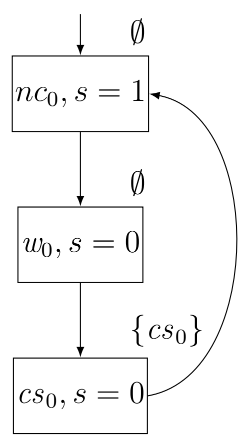
Solution:



(b) Give the reachable part of the transition system for P_0. Do not forget to include the set AP of atomic propositions and the labelling function relevant for answering part (c) below. (10)

Solution:

TS for P_0, $AP = \{cs_0\}$



(c) Formally state the properties of mutual exclusion and starvation freedom for this algorithm as languages of infinite words over $\Sigma = 2^{AP}$. Are these two properties satisfied by the algorithm? (10)

Solution:

$$\text{ME} = \{A_0 A_1 A_2 \ldots \mid \forall i.\ cs_0 \notin A_i \vee cs_1 \notin A_i\}$$

$$\text{SF} = \{A_0A_1A_2\ldots \mid \overset{\infty}{\exists}\, i.\ cs_0 \in A_i \wedge \overset{\infty}{\exists}\, i.\ cs_1 \in A_i\}$$

or alternatively, according to the interpretation in the book,

$$\begin{aligned}\text{SF} = \{A_0A_1A_2\ldots \mid & (\overset{\infty}{\exists}\, i.w_0 \in A_i) \Rightarrow (\overset{\infty}{\exists}\, j.cs_0 \in A_j)] \\ & \wedge \\ & (\overset{\infty}{\exists}\, i.w_1 \in A_i) \Rightarrow (\overset{\infty}{\exists}\, j.cs_1 \in A_j)]\}\end{aligned}$$

Neither of these properties, ME or SF, are satisfied by the algorithm.

3. Consider the set of atomic propositions $AP = \{A, B\}$. Using mathematical notation formally describe the following properties as linear-time properties (*i.e.*, as languages of infinite words over the alphabet $\Sigma = 2^{AP}$). Also, for each property state whether it is an invariant property, or a safety property (if it is not an invariant property), or a liveness property or none of these with a brief justification of your answer. Do not use any atomic proposition other than A and B in your answer.

   (a) A should never occur. (10)

   Solution:

   $$P = \{A_0A_1A_2\ldots \mid \forall i.\ A \notin A_i\}$$

   This is an invariant with the invariant condition $\neg A$.

   (b) A should occur exactly once. (10)

   Solution:

   $$P = \{A_0A_1A_2\ldots \mid \exists i.\ [A \in A_i \wedge \forall j(A \in A_j \Rightarrow j = i)]\}$$

   This is neither a safety property nor a liveness property since the word B^ω is not in P but has no bad prefix and any finite word over Σ where A occurs more than once cannot be extended to a word in P.

   (c) A and B alternate infinitely often starting with A. This means only A is true in the first step, then only B is true in the next step, and this alternation between A and B repeats infinitely often. (10)

Solution:

$$P = \{A_0A_1A_2\ldots \mid \forall i.\ [A_{2i} = \{A\} \wedge A_{2i+1} = \{B\}]\}$$

This is a safety property since any finite word over Σ where A and B do not alternate starting with $\{A\}$ is a bad prefix.

(d) Every B is strictly preceded by an A, *i.e.*, for every B there is an earlier occurrence of A. (10)

Solution:

$$P = \{A_0A_1A_2\ldots \mid \forall i.\ [B \in A_i \Rightarrow \exists j.\ (j < i \wedge A \in A_j)]\}$$

This is a safety property since any finite word over Σ where an occurrence of a B is not preceded by an occurrence of an A is a bad prefix.

4. Let P be a liveness property and P' a safety property over some set of atomic propositions AP. Answer the following questions with proper justification.

(a) Is $P \cup P'$ always a liveness property? (10)

Solution: Yes, $P \cup P'$ is always a liveness property since any nonempty finite word w can be extended to an infinite word $\sigma \in P$, as P is a liveness property and hence also to a word in $P \cup P'$ as $P \subseteq P \cup P'$.

(b) Is $P \cap P'$ always a liveness property? (10)

Solution: No. Take $P' = \emptyset$ which is a safety property with all nonempty finite words over $\Sigma = 2^{AP}$ as the set of bad prefixes. Then $P \cap P' = \emptyset$ which is not a liveness property since no nonempty word can be extended to a word in this set.