Staff Solutions to Problem Set 4

STAFF NOTE: Topics: Partial Order Scheduling, Digraphs, State machine invariants (not derived vars)

Problem 1.
Let $\prec$ be a partial order on a set, $A$, and let

$$A_k := \{a \mid \text{depth}(a) = k\}$$

where $k \in \mathbb{N}$.

(a) Prove that $A_0, A_1, \ldots$ is a parallel schedule for $\prec$ according to Definition 7.5.5.

Solution. Proof. We have to show that if $a \in A_k, b \in A_j$ and $b \prec a$, then $j < k$.

Now since depth $(b) = j$ by definition of $A_j$, there is a length $j$ chain, $C$, that ends at $b$. Since $b$ is the maximum element of $C$ and $b \prec a$, the set $C \cup \{a\}$ is a chain of length $j + 1$ that ends at $a$. So depth $(a) \geq j + 1$ by definition of depth, and

$$k = \text{depth}(a) \geq j + 1 > j.$$ 

(b) Prove that $A_k$ is an antichain.

Solution. Proof. Suppose $a \neq b$ are elements of $A$ and depth $(a) = \text{depth}(b)$. We want to show that $a$ and $b$ are incomparable.

Suppose to the contrary that $a$ and $b$ are comparable, say, $b \prec a$. By definition of depth, there is a chain, $C$, of length depth $(b)$ ending at $b$. Since $b$ is the maximum element of $C$ and $b \prec a$, the set $C \cup \{a\}$ is a chain of length depth $(b) + 1$ that ends at $a$. It follows that depth $(a) > \text{depth}(b)$, contradicting the assumption that depth $(a) = \text{depth}(b)$.

Problem 2.
In a round-robin tournament, every pair of distinct players play against each other just once. For a round-robin tournament with no tied games, a record of who beat whom can be described with a tournament digraph, where the vertices correspond to players and there is an edge $x \rightarrow y$ if $x$ beat $y$ in their game.

A ranking is a directed simple path that includes all the players.
(a) Give an example of a tournament digraph with more than one ranking.

**Solution.** Let \( n = 3 \) with edges \( u \rightarrow v \rightarrow w \rightarrow u \). There are three different rankings starting with \( u \), \( v \), and \( w \), respectively.

(b) If a tournament digraph is a DAG, then it has a unique ranking. Explain.

**Solution.** If it’s a DAG, then its path relation defines a partial order, and since every pair of elements are comparable, this is in fact a total order. So there’s no alternative but to rank the elements from smallest to largest.

(c) Prove that every tournament digraph has a ranking. *Hint:* Induction on the size of the tournament.

**Solution.** By induction on \( n \) with induction hypothesis

\[
P(n) := \text{every tournament digraph with } n \text{ vertices has a ranking.}
\]

**Base case** \( n = 1 \): Trivial.

**Inductive step:** Let \( G \) be a tournament digraph with \( n + 1 \) vertices. Remove one vertex, \( v \), to obtain the subgraph, \( H \), with the \( n \) remaining vertices. Clearly, \( H \) is also a tournament digraph, so by induction hypothesis it has a ranking. Now if the last player in this \( H \)-ranking beat player \( v \), then \( v \) can be added at the end to form a ranking in \( G \). On the other hand, if \( v \) beat the last player in the \( H \)-ranking, then there will (by WOP) be a first player in the \( H \)-ranking that \( v \) beats. Inserting \( v \) just before that first player gives a ranking for \( G \). Since \( G \) was an arbitrary \( n + 1 \) vertex tournament graph, we conclude that \( P(n + 1) \) holds, which completes the proof.

Problem 3.

A robot named Wall-E wanders around a two-dimensional grid. He starts out at \((0, 0)\) and is allowed to take four different types of step:

1. \((+2, -1)\)
2. \((+1, -2)\)
3. \((+1, +1)\)
4. \((-3, 0)\)

Thus, for example, Wall-E might walk as follows. The types of his steps are listed above the arrows.

\[
(0, 0) \rightarrow (2, -1) \rightarrow (3, 0) \rightarrow (4, -2) \rightarrow (1, -2) \rightarrow \ldots
\]

Wall-E’s true love, the fashionable and high-powered robot, Eve, awaits at \((0, 2)\).

(a) Describe a state machine model of this problem.
Solution. Let the set of states be $\mathbb{Z} \times \mathbb{Z}$. The start state is $(0, 0)$. The possible transitions are

$$(x, y) \rightarrow (x, y) + (u, v) \tag{1}$$

for $(u, v) \in \{(+2, -1), (+1, -2), (+1, +1), (-3, 0)\}$.

(b) Will Wall-E ever find his true love? Either find a path from Wall-E to Eve or use the Invariant Principle to prove that no such path exists.

Solution. Let $P(x, y)$ be the property that $x + 2y$ is a multiple of 3. we claim $P$ is a preserved invariant. To show this, we must show that if $3 \mid x + 2y$, and Wall-E moves to $(x, y) + (u, v)$, then $3$ divides

$$(x + u) + 2(y + v). \tag{2}$$

But this value equals

$$(x + 2y) + (u + 2v), \tag{3}$$

and since

$$3 \mid u + 2v$$

for each of the four possible moves $(u, v)$ listed above (as is easily checked), we conclude that $3$ divides both terms in the sum (3) and therefore divides the whole sum. This proves implies that $3$ divides (2), completing the proof that $P$ is preserved by transitions.

Now $P$ holds in the start state, since $3 \mid (0 + 2 \cdot 0)$. However, $P$ does not hold for Eve’s position, $(0, 2)$, since $0 + 2 \cdot 2 = 4$ is not a multiple of 3. Therefore, by the Invariant Principle, Eve’s position is not a reachable state.

Problem 4.
In the late 1960s, the military junta that ousted the government of the small republic of Nerdia completely outlawed built-in multiplication operations, and also forbade division by any number other than 3. Fortunately, a young dissident found a way to help the population multiply any two nonnegative integers without risking persecution by the junta. The procedure he taught people is:

procedure multiply$(x, y$: nonnegative integers$)$

$r := x; \\
 s := y; \\
a := 0; \\
$while $s \neq 0$ do \\
    if $3 \mid s$ then \\
        $r := r + r + r; \\
        s := s/3; \\
   $else if $3 \mid (s - 1)$ then \\
        $a := a + r; \\
        r := r + r + r; \\
        s := (s - 1)/3; \tag{4}$


else
    \[ a := a + r + r; \]
    \[ r := r + r + r; \]
    \[ s := (s - 2)/3; \]
return \( a \);

We can model the algorithm as a state machine whose states are triples of nonnegative integers \((r, s, a)\). The initial state is \((x, y, 0)\). The transitions are given by the rule that for \(s > 0\):

\[
(r, s, a) \rightarrow \begin{cases} 
(3r, s/3, a) & \text{if } 3 \mid s \\
(3r, (s - 1)/3, a + r) & \text{if } 3 \mid (s - 1) \\
(3r, (s - 2)/3, a + 2r) & \text{otherwise}.
\end{cases}
\]

(a) List the sequence of steps that appears in the execution of the algorithm for inputs \(x = 5\) and \(y = 10\).

Solution. \((5, 10, 0) \longrightarrow (15, 3, 5) \longrightarrow (45, 1, 5) \longrightarrow (135, 0, 50)\)

(b) Use the Invariant Method to prove that the algorithm is partially correct — that is, if \(s = 0\), then \(a = xy\).

Solution. Let

\[
P((r, s, a)) ::= [rs + a = xy].
\]

Clearly, \(P\) holds for the start state because

\[
P((x, y, 0)) \iff [xy + 0 = xy].
\]

Now, we show that \(P\) is indeed a preserved invariant, namely, assuming \(P((r, s, a))\),

\[
rs + a = xy, \tag{4}
\]

holds and \((r, s, a) \rightarrow (r', s', a')\) is a transition, then \(P((a', b', p'))\),

\[
r's' + a' = xy, \tag{5}
\]

holds.

We consider three cases:

If \(3 \mid s\), then we have that \(r' = 3r, s' = s/3, a' = a\). Therefore,

\[
\begin{align*}
r's' + a' &= 3r \cdot \frac{s}{3} + a \\
&= rs + a \\
&= xy \quad \text{(by (4)).}
\end{align*}
\]

else
If $3 \mid s - 1$, then $r' = 3r, s' = (s - 1)/3, a = a + r$. So:

$$r's' + a' = 3r \cdot \frac{s - 1}{3} + a + r$$

$$= r \cdot (s - 1) + a + r$$

$$= rs + a$$

$$= xy$$

(by (4)).

Otherwise, we have $r' = 3r, s' = (s - 2)/3, a = a + 2r$. So:

$$r's' + a' = 3r \cdot \frac{s - 2}{3} + a + 2r$$

$$= r \cdot (s - 2) + a + 2r$$

$$= rs + a$$

$$= xy$$

(by (4)).

So in all three cases, (5) holds, proving that $P$ is indeed a preserved invariant.

Since the procedure’s only termination condition is that $s = 0$, partial correctness will follow if we can show that if $s = 0$, then $a = xy$. But this follows immediately from (4).

(c) Prove that the algorithm terminates after at most $1 + \log_3 y$ executions of the body of the do statement.

Solution. We first notice that $s \in \mathbb{N}$ is a preserved invariant. Also, each transition corresponds to an execution of the do statement body, and each transition reduces $s$ to at most $s/3$. Hence, after at most $1 + \log_3 y$ executions of the body, the value of $s$ is at most its initial value, $y$, times $(1/3)^{1+\log_3 y} = 1/3y$. That is, the value of $s$ is at most $1/3$. Since $s \in \mathbb{N}$, it follows that $s$ will be 0 after this many executions of the body, if it wasn’t 0 earlier. But with $s = 0$, the procedure terminates.