Basics of Parallel Programs
CS 536: Science of Programming, Fall 2021

A. Why
• Parallel programs are more flexible than sequential programs but their execution is more complicated.
• Parallel programs are harder to reason about because parts of a parallel program can interfere with other parts.

B. Objectives
At the end of this work you should be able to
• Draw evaluation graphs for parallel programs.

C. Problems
In general, for the problems below, if it helps you with the writing, feel free to define other symbols. ("Let \( S \equiv \text{some program}, \" for example.)

1. What is the sequential nondeterministic program that corresponds to the program from Example 4, \( [x := v || y := v+2 || z := v*2] \).

2. Let configuration \( C_2 \equiv (S_2, \sigma) \) where \( S_2 \equiv [x := 1 || x := -1] \).
   a. What is the sequential nondeterministic program that corresponds to \( S_1 \) ?
   b. Draw an evaluation graph for \( C_2 \).

3. Repeat Problem 2 on \( C_2 \equiv (S_3, \sigma[v \mapsto 0]) \) where \( S_3 \equiv [x := v+3; v := v*4 || v := v+2] \).
   Note that in the first thread, the two assignments must be done with \( x \) first, then \( v \). Because adding 3 and adding 2 are commutative, two of the (normally-different) nodes will merge.

4. Repeat Problem 2 on \( C_2 \equiv (S_4, \sigma[v \mapsto \delta]) \) where \( S_4 \equiv [v := v*y; v := v+\beta || v := v+\alpha] \).
   This problem is similar to Problem 3 but is symbolic, and the commutative plus operator has been moved, so the shape of the graph will be different from Problem 3.
5. Let $C_5 \equiv (W, \sigma)$ where $W \equiv \text{while } x \leq n \text{ do } [x := x+1 \mid \mid y := y*2] \text{ od}$ and let $\sigma$ of $x$, $y$, and $z$ be 0, 1, and 2 respectively. Note the parallel construct is in the body of the loop.
   a. Draw an evaluation graph for $C_5$. (Feel free to to say something like “Let $T \equiv \ldots$” for the loop body, to cut down on the writing.
   b. Draw another evaluation graph for $C_5$, but this time, use the $\rightarrow^3$ notation to get a straight line graph. Concentrate on the configurations of the form $(W, \ldots)$.

6. In $[S_1 \mid \mid S_2 \mid \mid \ldots \mid \mid S_n]$ can any of the threads $S_1$, $S_2$, ..., $S_n$ contain parallel statements? Can parallel statements be embedded within loops or conditionals?

7. Say we know $\{p_1\} S_1 \{q_1\}$ and $\{p_2\} S_2 \{q_2\}$ under partial or total correctness.
   a. In general, do we know how $\{p_1 \land p_2\} \ [S_1 \mid \mid S_2] \ {q_1 \land q_2}$ will execute? Explain briefly.
   b. What if $p_1 \equiv p_2$? i.e., if we know $\{p\} S_1 \ {q_1}$ and $\{p\} S_2 \ {q_2}$, then do we know how $\{p\} \ [S_1 \mid \mid S_2] \ {q_1 \land q_2}$ will work?
   c. What if in addition, $q_1 \equiv q_2$? i.e., If we know $\{p\} S_1 \ {q}$ and $\{p\} S_2 \ {q}$, do we know how $\{p\} \ [S_1 \mid \mid S_2] \ {q}$ will work? (This problem is harder)
   d. For parts (a) – (c), does it make a difference if we use $\lor$ instead of $\land$?
Solution to Practice 22

Class 22: Basics of Parallel Programs

1. Sequential nondeterministic equivalent of \([x := v \parallel y := v+2 \parallel z := v*2]\):
   
   \[
   \begin{align*}
   & if \ T \rightarrow x := v; y := v+2; z := v*2 \\
   & \quad □ T \rightarrow x := v; z := v*2; y := v+2 \\
   & \quad □ T \rightarrow y := v+2; x := v; z := v*2 \\
   & \quad □ T \rightarrow y := v+2; z := v*2; x := v \\
   & \quad □ T \rightarrow z := v*2; y := v+2; x := v \\
   & fi
   \end{align*}
   \]

2. (Program \([x := 1 \parallel x := -1] ; y := y+x]\)
   
   a. Equivalent sequential nondeterministic program
   
   \[
   \begin{align*}
   & if \ T \rightarrow x := 1; x := -1 \quad □ T \rightarrow x := -1; x := 1 fi
   \end{align*}
   \]

   b. Evaluation graph for \(\langle [x := 1 \parallel x := -1]; y := y+x, σ \rangle\)
   
   \[
   \begin{align*}
   & \langle [x := 1 \parallel x := -1]; y := y+x, σ \rangle \\
   & \quad \downarrow \\
   & \quad \langle [E \parallel x := -1]; y := y+x, σ[x \mapsto 1] \rangle \\
   & \quad \quad \downarrow \\
   & \quad \quad \langle [E \parallel x := -1]; y := y+x, σ[x \mapsto -1] \rangle
   \end{align*}
   \]

3. (Program \([v := v+3; \ v := v*4 \parallel v := v+2]\)
   
   a. Equivalent sequential nondeterministic program
   
   \[
   \begin{align*}
   & if \ T \rightarrow v := v+3; \ if \ T \rightarrow v := v*4; \ v := v+2 \quad □ T \rightarrow v := v+2; \ v := v*4 fi \\\n   & □ T \rightarrow v := v+2; \ v := v+3; \ v := v*4 fi
   \end{align*}
   \]
b. Evaluation graph for \((v := v+3; \ v := v^*4|| v := v+2), \sigma[{v \mapsto 0}]\). Note that two of the execution paths happen to merge, so there are only two final states instead of three.

\[
( v := v+3; \ v := v^*4|| v := v+2), \sigma[{v \mapsto 0}] \\
\]

- Evaluation graph for \((v := v^*4|| v := v+2), \sigma[{v \mapsto 3}]\)
- Evaluation graph for \((v := v^*4|| E), \sigma[{v \mapsto 2}]\)

- Evaluation graph for \((E|| v := v+2), \sigma[{v \mapsto 12}]\)
- Evaluation graph for \((v := v^*4|| E), \sigma[{v \mapsto 5}]\)

- Evaluation graph for \((E || E), \sigma[{v \mapsto 14}]\)
- Evaluation graph for \((E || E), \sigma[{v \mapsto 20}]\)

4. (Program \([v := v^*γ; \ v := v+β|| v := v+α]\).

a. Equivalent sequential nondeterministic program
   \[
   if \ T \rightarrow v := v^*γ; \ if \ T \rightarrow v := v+β; \ v := v+α \ □ \ T \rightarrow v := v+α; \ v := v+β \ fi
   \]
   \[
   □ \ T \rightarrow v := v+α; \ v := v^*γ; \ v := v+β
   \]
   \[
   fi
   \]

b. Evaluation graph for \((v := v^*γ; \ v := v+β|| v := v+2), \sigma[{v \mapsto δ}]\)

- Evaluation graph for \((v := v^*γ; \ v := v+β|| v := v+α), \sigma[{v \mapsto δ}]\)
- Evaluation graph for \((v := v^*γ; \ v := v+β|| E), \sigma[{v \mapsto δ+α}]\)
- Evaluation graph for \((v := v+β|| E), \sigma[{v \mapsto δγ+β + α}]\)
- Evaluation graph for \((v := v+β|| E), \sigma[{v \mapsto (δ+α)(γ+β)}]\)
- Evaluation graph for \((v := v+β|| E), \sigma[{v \mapsto (δ+α)+γ+β}]\)
- Evaluation graph for \((v := v+β|| E), \sigma[{v \mapsto δγ+β + α}]\)

- Evaluation graph for \((E|| v := v+α), \sigma[{v \mapsto δγ+β}]\)
- Evaluation graph for \((E|| v := v+α), \sigma[{v \mapsto δγ+β + α}]\)
- Evaluation graph for \((E|| E), \sigma[{v \mapsto δγ+β + α}]\)
5. \((\text{while } x \leq n \text{ do } [x := x+1 \ || \ y := y*2] \text{ od, if } \sigma(x) = 0, \ \sigma(y) = 1, \text{ and } \sigma(n) = 2.\) Below, let \(T \equiv [x := x+1 \ || \ y := y*2]\) (just to cut down on the writing).

a. A full evaluation graph. Just to be explicit, I wrote \(\sigma[x \mapsto 0][y \mapsto 1]\) below but just \(\sigma\) is fine.

\[
\begin{align*}
&\langle W, \sigma[x \mapsto 0][y \mapsto 1] \rangle \\
\rightarrow &\langle T; W, \sigma[x \mapsto 0][y \mapsto 1] \rangle \\
&\langle [E || y := y*2]; W, \sigma[x \mapsto 1][y \mapsto 1] \rangle \\
&\langle [x := x+1 || E]; W, \sigma[x \mapsto 0][y \mapsto 2] \rangle \\
\rightarrow &\langle W, \sigma[x \mapsto 1][y \mapsto 2] \rangle \\
&\langle T; W, \sigma[x \mapsto 1][y \mapsto 2] \rangle \\
&\langle [E || y := y*2]; W, \sigma[x \mapsto 2][y \mapsto 2] \rangle \\
&\langle [x := x+1 || E]; W, \sigma[x \mapsto 1][y \mapsto 4] \rangle \\
\rightarrow &\langle W, \sigma[x \mapsto 2][y \mapsto 4] \rangle \\
&\langle T; W, \sigma[x \mapsto 2][y \mapsto 4] \rangle \\
&\langle [E || y := y*2]; W, \sigma[x \mapsto 3][y \mapsto 4] \rangle \\
&\langle [x := x+1 || E]; W, \sigma[x \mapsto 2][y \mapsto 8] \rangle \\
\rightarrow &\langle W, \sigma[x \mapsto 3][y \mapsto 8] \rangle \\
&\langle E, \sigma[x \mapsto 3][y \mapsto 8] \rangle
\end{align*}
\]

b. Evaluation graph abbreviated using \(\rightarrow^3\) notation:

\[
\begin{align*}
&\langle W, \sigma[x \mapsto 0][y \mapsto 1] \rangle \rightarrow^3 \langle W, \sigma[x \mapsto 1][y \mapsto 2] \rangle \rightarrow^3 \langle W, \sigma[x \mapsto 2][y \mapsto 4] \rangle \\
&\rightarrow^3 \langle W, \sigma[x \mapsto 3][y \mapsto 8] \rangle \rightarrow \langle E, \sigma[x \mapsto 3][y \mapsto 8] \rangle
\end{align*}
\]
6. No, in \([S_1 \parallel S_2 \parallel \ldots \parallel S_n]\) the threads cannot contain parallel statements, but yes, parallel statements can be embedded within loops and conditionals.

7. In general, even if \(\{p_1\} S_1 \{q_1\}\) and \(\{p_2\} S_2 \{q_2\}\) are both valid sequentially, we can't compose them in parallel, even if \(p_1 \equiv p_2\) and \(q_1 \equiv q_2\). An example is how \(\{x > 0\} x := x-1 \{x \geq 0\}\) is valid but \(\{x > 0\} [x := x-1]|x := x–1\} \{x \geq 0\}\) is not. The first \(x := x-1\) to execute ends with \(x \geq 0\), which is too weak for the second \(x := x-1\) to work correctly.