Basics of Parallel Programs

CS 536: Science of Programming, Spring 2023

Solved

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 $S \equiv$ *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 $C_2 \equiv \langle S_2, \sigma\rangle$ 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 $C_3 \equiv \langle S_3, \sigma[v \mapsto 0] \rangle$ 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 $C_4 \equiv \langle S_5, \sigma[v \mapsto \delta]\rangle$ where $S_4 \equiv [v := v*\gamma;\ v := v+\beta \,||\, v := v+\alpha]$. 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 $C_5 \equiv \langle W, \sigma\rangle$ where $W \equiv$ ***while*** $x \le n$ ***do*** $[x := x+1 \,||\, y := y*2]$ ***od*** and let σ 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 $\langle W, \ldots \rangle$.

6. In $[S_1 \,||\, S_2 \,||\, \ldots \,||\, S_n]$ can any of the threads $S_1, S_2, \ldots, S_n$ 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 $\{p_1 \wedge p_2\}\ [S_1 \,||\, S_2]\ \{q_1 \wedge q_2\}$ 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 $\{p\}\ [S_1 \,||\, S_2]\ \{q_1 \wedge q_2\}$ 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 $\{p\}\ [S_1 \,||\, S_2]\ \{q\}$ will work? (This problem is harder)
   d. For parts (a) – (c), does it make a difference if we use $\vee$ instead of $\wedge$?

8. What is a race condition? If a parallel program can produce different possible results, is this necessarily a race condition?

Solution to Practice 22

Class 22: Basics of Parallel Programs

1. Sequential nondeterministic equivalent of [x := v || y := v+2 || z := v*2]:

    if T ➞ x := v; y := v+2; z := v*2
    ◻ T ➞ x := v; z := v*2; y := v+2
    ◻ T ➞ y := v+2; x := v; z := v*2
    ◻ T ➞ y := v+2; z := v*2; x := v
    ◻ T ➞ z := v*2; x := v; y := v+2
    ◻ T ➞ z := v*2; y := v+2; x := v
    fi

This also works:

if T ➞ x := v; ***if*** T ➞ y := v+2; z := v*2 ◻ T ➞ z := v*2; y := v+2 ***fi fi***
◻ T ➞ y := v+2; ***if*** T ➞ x := v ; z := v*2 ◻ T ➞ z := v*2 ; x := v ***fi fi***
◻ T ➞ z := v*2; ***if*** T ➞ x := v ; y := v+2 ◻ T ➞ y := v+2 ; x := v ***fi fi***
fi

2. (Program [x := 1 || x := –1] ; y := y+x])
    a. Equivalent sequential nondeterministic program

        if T ➞ x := 1; x := –1 ◻ T ➞ x := –1; x := 1 ***fi***

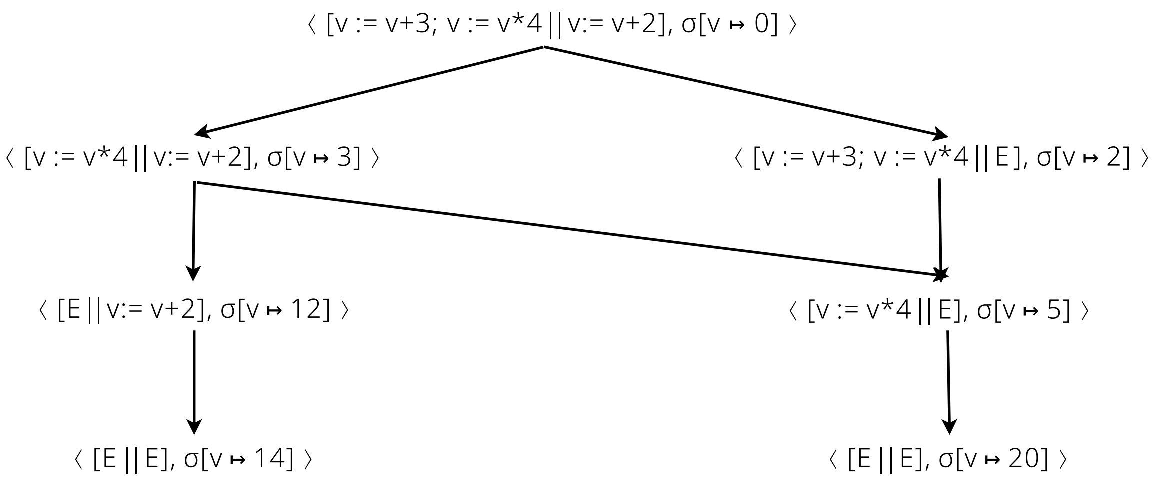
    b. Evaluation graph for ⟨ [x := 1 || x := –1]; y := y+x, σ ⟩

$$\begin{tikzcd}
& \langle\, [x := 1 \,\|\, x := -1]\,;\, y := y+x,\ \sigma \,\rangle \arrow[dl] \arrow[dr] & \\
\langle\, [E \,\|\, x := -1];\ y := y+x,\ \sigma[x \mapsto 1\} \,\rangle \arrow[d] & & \langle\, [x := 1 \,\|\, E];\ y := y+x,\ \sigma[x \mapsto -1] \,\rangle \arrow[d] \\
\langle\, [E \,\|\, E];\ y := y+x,\ \sigma[x \mapsto -1\} \,\rangle \arrow[d] & & \langle\, [E \,\|\, E];\ y := y+x,\ \sigma[x \mapsto 1\} \,\rangle \arrow[d] \\
\langle\, E,\ \sigma[x \mapsto -1\}[y \mapsto \sigma(y)-1\} \,\rangle & & \langle\, E,\ \sigma[x \mapsto 1\}[y \mapsto \sigma(y)+1\} \,\rangle
\end{tikzcd}$$

3. (Program [v := v+3; v := v*4 || v := v+2])
   a. Equivalent sequential nondeterministic program

      if T → v := v+3; ***if*** T → v := v*4; v := v+2 □ T → v := v+2; v := v*4 ***fi***
      □ T → v := v+2; v := v+3; v := v*4
      fi

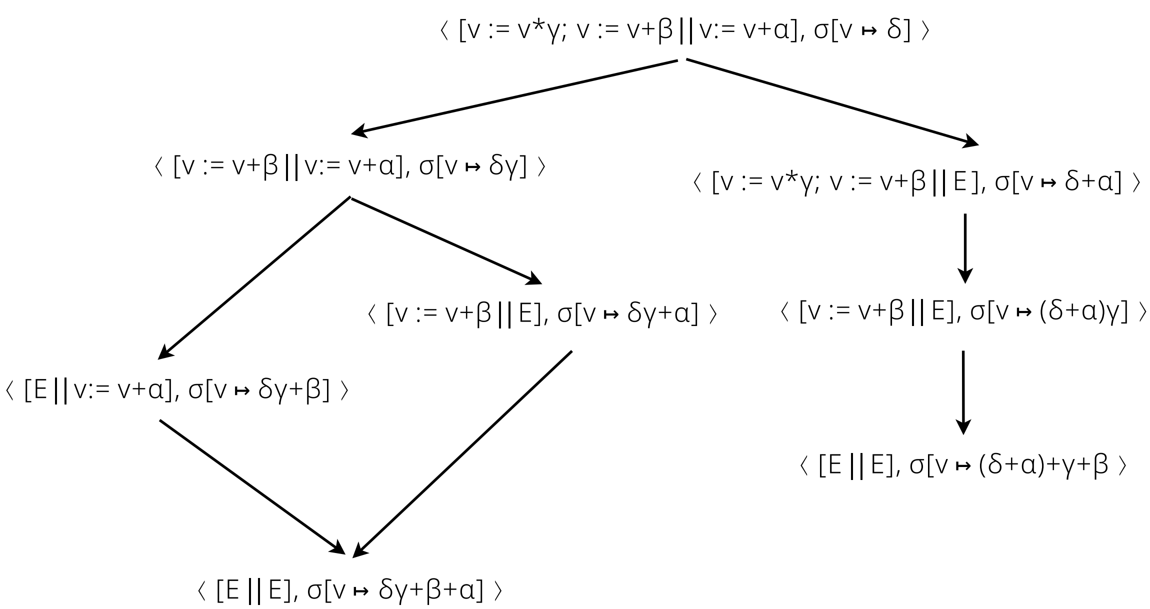
   b. Evaluation graph for 〈 [v := v+3; v := v*4 || v := v+2], σ[v ↦ 0] 〉. Note that two of the execution paths happen to merge, so there are only two final states instead of three.



4. (Program [v := v*γ; v := v+β || v:= v+α]).
   a. Equivalent sequential nondeterministic program

      if T → v := v*γ; ***if*** T → v := v+β; v := v+α □ T → v := v+α; v := v+β ***fi***
      □ T → v := v+α; v := v*γ; v := v+β
      fi

   b. Evaluation graph for 〈 [v := v*γ; v := v+β || v := v+2], σ[v ↦ δ] 〉



5. (***while*** x ≤ n ***do*** [x := x+1 || y := y*2] ***od***, if σ(x) = 0, σ(y) = 1, and σ(n) = 2.) Below, let T ≡ [x := x+1 || y := y*2] (just to cut down on the writing).

   a. A full evaluation graph. Just to be explicit, I wrote σ[x ↦ 0][y ↦ 1] below but just σ is fine.

$$\begin{tikzcd}
& \langle W, \sigma[x \mapsto 0][y \mapsto 1] \rangle \arrow[d] & \\
& \langle T; W, \sigma[x \mapsto 0][y \mapsto 1] \rangle \arrow[dl] \arrow[dr] & \\
\langle [E \,||\, y{:=}y*2]; W, \sigma[x \mapsto 1][y \mapsto 1] \rangle \arrow[dr] & & \langle [x{:=}x+1 \,||\, E]; W, \sigma[x \mapsto 0][y \mapsto 2] \rangle \arrow[dl] \\
& \langle W, \sigma[x \mapsto 1][y \mapsto 2] \rangle \arrow[d] & \\
& \langle T; W, \sigma[x \mapsto 1][y \mapsto 2] \rangle \arrow[dl] \arrow[dr] & \\
\langle [E \,||\, y{:=}y*2]; W, \sigma[x \mapsto 2][y \mapsto 2] \rangle \arrow[dr] & & \langle [x{:=}x+1 \,||\, E]; W, \sigma[x \mapsto 1][y \mapsto 4] \rangle \arrow[dl] \\
& \langle W, \sigma[x \mapsto 2][y \mapsto 4] \rangle \arrow[d] & \\
& \langle T; W, \sigma[x \mapsto 2][y \mapsto 4] \rangle \arrow[dl] \arrow[dr] & \\
\langle [E \,||\, y{:=}y*2]; W, \sigma[x \mapsto 3][y \mapsto 4] \rangle \arrow[dr] & & \langle [x{:=}x+1 \,||\, E]; W, \sigma[x \mapsto 2][y \mapsto 8] \rangle \arrow[dl] \\
& \langle W, \sigma[x \mapsto 3][y \mapsto 8] \rangle \arrow[d] & \\
& \langle E, \sigma[x \mapsto 3][y \mapsto 8] \rangle &
\end{tikzcd}$$

b. Evaluation graph abbreviated using $\rightarrow^3$ notation. This one is nice and linear:
$\langle$ 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, $\sigma[x \mapsto 3][y \mapsto 8]$ $\rangle$

6. No, in $[S_1 \| S_2 \| \ldots \| S_n]$ 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.

8. In a race condition, the correctness of a parallel program depends on the relative speeds of the processors involved (i.e., their interleaving at execution time). Simply producing different results doesn't necessarily indicate a race condition: If all results meet the specification, then no race condition has occurred.

