Disjoint Conditions

CS 536: Science of Programming, Spring 2023

Solved

A. Why

- Disjoint parallel programs ensure that no thread can interfere with the execution of another thread.
- Disjoint conditions ensure that no thread can interfere with the conditions of a triple.
- Disjoint parallel programs with disjoint conditions can be proved correct by combining the proofs of their individual threads.

B. Objectives

At the end of this work you should be able to

- Recognize disjoint parallel programs and correctness triples with disjoint conditions
- Use the rules for sequentialization and disjoint parallelism

C. Questions

1. Let $\{p_1 \wedge p_2\}\ [\{p_1\}\ S_1\ \{q_1\} \parallel \{p_2\}\ S_2\ \{q_2\}]\ \{q_1 \wedge q_2\}$ be a DPP with DC. Changes to the p's or q's obviously can change whether the result still has DC. Describe the kinds of changes to the p's or q's that causes the result to not be a DPP. Or, argue briefly that no such changes exist.

2. Are the following programs parallel disjoint with disjoint conditions?
   - $\{T\}\ x := 1\ ;\ y := 1\ \{x = 1\}$
   - $\{x = 0\}\ z := 0\ \{x = z\}$

i	*j*	*Change i*	*Vars j*	*Free j*	*Disjoint Program?*	*Disjoint Conditions?*
1	2					
2	1					

3. Are the following programs parallel disjoint with disjoint conditions?
   - {T} x := 1 ; y := 0 {x = 1}
   - {z = 0} z := z*x {z = 0}

i	*j*	*Change i*	*Vars j*	*Free j*	*Disjoint Program?*	*Disjoint Conditions?*
1	2					
2	1					

4. Are the following programs parallel disjoint with disjoint conditions?
   - {T} ***if*** x > 0 ***then*** y := 1; z := 2 ***fi*** {x ≤ 0 → z = 2}
   - {T} ***if*** x ≤ 0 ***then*** z := 2; y := 3 ***fi*** {x ≤ 0 → y = 3}

i	*j*	*Change i*	*Vars j*	*Free j*	*Disjoint Program?*	*Disjoint Conditions?*

5. Are the following programs parallel disjoint with disjoint conditions?
   - {T} x := u ; y := u {x = y}
   - {z > 0} z := z–1 ; v := z {v = z}
   - {w ≥ u} w := w+1 {w > u}

i	*j*	*Change i*	*Vars j*	*Free j*	*Disjoint Program?*	*Disjoint Conditions?*
1	2					
1	3					
2	1					
2	3					
3	1					
3	2					

6. Design a parallel program $\{p_1 \wedge p_2\}\ [\{p_1\}\ S_1\ \{q_1\} \parallel \{p_2\}\ S_2\ \{q_2\}]\ \{q_1 \wedge q_2\}$ where (1) the threads are not disjoint programs, (2) the threads don't have disjoint conditions, but (3) in fact, running the program in any state works correctly. There are any number of possible answers — it's easiest if you write the threads so that each one produces the same results whether you run it sequentially or in parallel with the other thread.

7. What do you get if you expand outlines (a), (b), and (d) from ***Example 6*** to get full outlines?

Solution to Practice 24

1. Changes to the conditions of $\{p_1 \wedge p_2\}\ [\{p_1\}\ S_1\ \{q_1\} \parallel \{p_2\}\ S_2\ \{q_2\}]\ \{q_1 \wedge q_2\}$ don't change whether the result is a DPP. The definition of DPP only depends on the programs, not the conditions.

Feel free to write {x, y} as x, y or just x y (which is what I used below.)

2. No: Thread 1 interferes with the conditions of thread 2

i	*j*	*Change i*	*Vars j*	*Free j*	*Disjoint Program?*	*Disjoint Conditions?*
1	2	x y	z	x z	Yes	No (because of) x
2	1	z	x y	x	Yes	Yes

3. No: Thread 1 interferes with the program of thread 2.

i	*j*	*Change i*	*Vars j*	*Free j*	*Disjoint Program?*	*Disjoint Conditions?*
1	*2*	x y	x z	z	No (because of) x	Yes
2	*1*	z	x y	x	Yes	Yes

4. No: Each interferes with the other's programs and conditions.

i	*j*	*Change i*	*Vars j*	*Free j*	*Disjoint Program?*	*Disjoint Conditions?*
1	2	y z	x y z	x y	No: y, z	No: y
2	1	y z	x y z	x z	No: y, z	No: z

5. Yes, these are parallel disjoint with disjoint conditions

i	*j*	*Change i*	*Vars j*	*Free j*	*Disjoint Program?*	*Disjoint Conditions?*
1	2	x y	v z	v z	Yes	Yes
1	3	x y	w	u w	Yes	Yes
2	1	v z	u x y	x y	Yes	Yes
2	3	v z	w	u w	Yes	Yes
3	1	w	u x y	x y	Yes	Yes
3	2	w	v z	v z	Yes	Yes

6. It helps if the threads change the state at opposite times. Here's one solution:

   $\{T\}$

   $[\ \{T\}$ ***if*** $x > 0$ ***then*** $y := 0$ ***fi*** $\{x > 0 \to y = 0\}$

   $\|\ \{T\}$ ***if*** $x \le 0$ ***then*** $y := 0$ ***fi*** $\{x \le 0 \to y = 0\}]$

   $\{(x > 0 \to y = 0) \wedge (x \le 0 \to y = 0)\}$

   $\{y = 0\}$

More generally, if neither S_1 nor S_2 modify x, then the outline below is correct.

   $\{T\}$

   $[\ \{T\}$ ***if*** $x > 0$ ***then*** S_1 ***fi*** $\{x > 0 \to sp(x > 0, S_1)\}$

   $\|\ \{T\}$ ***if*** $x \le 0$ ***then*** S_2 ***fi*** $\{x \le 0 \to sp(x \le 0, S_2)\}\ ]$

   $\{(x > 0 \to sp(x > 0, S_1)) \wedge (x \le 0 \to sp(x \le 0, S_2))\}$

7. Below are one result of expanding outlines (a), (b), and (d) from Example 6 to get full outlines. (There can be more than one right answer.)

   a. $\{x \ge 0 \wedge y \le 0\}$

      $[\ \{x \ge 0\}\ z := x\ \{z \ge 0\}$

      $\|\ \{y \le 0\}\ w := -y\ \{y \le 0 \wedge w = -y\}\ \{w \ge 0\}]$

      $\{z \ge 0 \wedge w \ge 0\}$

   b. $\{z = 0\}$

      $[\ \{z = 0\}\ x := z+1\ \{z = 0 \wedge x = z+1\}\ \{x \le z = 0\}$

      $\|\ \{z = 0\}\ y := z\ \{z = 0 \wedge y = 0\}]$

      $\{x \le z = 0 \wedge y = z = 0\}$

      $\{x \le y = z = 0\}$

   d. $\{x = y = z = c\}$

      $\{x = c \wedge y = c \wedge z = c\}$

      $[\{x = c\}\ \{x^2 = c^2\}\ x := x^2\ \{x = c^2\}$

      $\|\ \{y = c\}\ \{y^2 = c^2\}\ y := y^2\ \{y = c^2\}$

      $\|\ \{z = c\}\ z := (z-d)*(z+d)\ \{z_0 = c \wedge z = (z_0-d)*(z_0+d)\}\ \{z = c^2-d^2\}$

      $]\ \{x = c^2 \wedge y = c^2 \wedge z = c^2-d^2\}$

      $\{x = y = z+d^2\}$

We can also use forward assignment on threads 1 and 2:

$\{x = y = z = c\}$

$\{x = c \wedge y = c \wedge z = c\}$

$[\ \{x = c\}\ x := x^2\ \{x_0 = c \wedge x = x_0^2\}\ \{x = c^2\}$ (*)

$\|\ \{y = c\}\ y := y^2\ \{y_0 = c \wedge y = y_0^2\}\ \{y = c^2\}$ (*)

$\|\ \{z = c\}\ \{(z-d)*(z+d) = c^2 - d^2\}\ z := (z-d)*(z+d)\ \{z = c^2 - d^2\}$ (*)

$]$

$\{x = c^2 \wedge y = c^2 \wedge z = c^2 - d^2\}$

$\{x = y = z + d^2\}$

