***Parallel Programs***

*CS 536: Science of Programming*

*Due Mon Nov 28 (no late assignments)*

Problems [50 points total]

1. [6 points] Draw an evaluation graph for the following program/state configuration and give *Mp* of the program in the state: 〈*S*, σ〉 where *S* ≡ [x := 18; y := x/2 || z := x/3] and σ(x) = 12. (Note that in the first thread the assignments are done sequentially.)

Ans: Mp(S, σ) = {σ[x↦18][y↦9][z↦6], σ[z↦4][x↦18][y↦9]}

〈 [x :=18；y :=x/2 || z := x/3] , σ〉

<[y:=x/2||z:=x/3],σ[x↦18]> <[x:=18;y:=x/2||E],σ[z↦4]>

<[E||z:=x/3],σ[x↦18][y↦9]> <[y:=x/2||E],σ[x↦18][z↦6]> <[y:=x/2||E],σ[z↦4][x↦18]>

<[E||E],σ[x↦18][y↦9][z↦6]> <[E||E],σ[z↦4][x↦18][y↦9]>

1. [6 = 2\*3 points] For each of (a) – (d) below, say whether it’s a legal program; if it is, say whether or not it has disjoint parallel programs as threads. If it’s illegal or doesn’t have parallel disjoint threads, briefly explain why.
   1. **[** x:= y \* y; y:= y+1 || while z > w do z:= z/2 od **]**
   2. **[** x:= v || y:= x + z || z:= x \* x **]**

**Ans:** a.

1. Yes. It has disjoint parallel program. Because x:= y \* y; y:= y+1 and while z > w do z:= z/2 od are disjoint, so [ x:= y \* y; y:= y+1 || while z > w do z:= z/2 od ] is a DPP.
2. Yes. It doesn’t have disjoint parallel program. Because x:= v and y:= x + z are not disjoint, change( x:=v ) ⋂ vars( y:=x+z ) = x, so [ x:= v || y:= x + z || z:= x \* x ] isn’t a DPP.
3. [9 = 3\*3 points] What are the *Change* and *Free* sets for the triples below? Are they parallel disjoint with disjoint conditions?
   * {x ≠ y} x := u ; y := u {x = y}
   * {v = z} z := z+1 ; v := v+1 {v = z}, and
   * {w ≥ y} w := w+1 {w > y}

Ans:

No. Thread 1 interferes with the program of thread 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | j | Change i | Vars j | Free j | Disjoint Pgm? | Disjoint Cond? |
| 1 | 2 | x, y | z, v | v, z | Y | Y |
| 1 | 3 | x, y | w | w, y | Y | No |
| 2 | 1 | v, v | x, y, u | x, y | Y | Y |
| 2 | 3 | v, v | w | w, y | Y | Y |
| 3 | 1 | w | x, y, u | x, y | Y | Y |
| 3 | 2 | w | z, v | v, z | Y | Y |

1. [4 = 2\*2 points] Consider the following program
   * 1. := 0; y := 2; k := 0;

**while** y < n **do**

u := n - y - x; x := x + f(y);

* + 1. := y\*y;

v := n - y - x; k := k+1

**od**

* 1. Is {x} a set of auxiliary variables for this program? If so, what program results after removing them?
  2. Is {x, u, v} a set of auxiliary variables for this program? If so, what program results after removing them?

Ans:

1. {x} is not a set of auxiliary variables: If x is auxiliary then u and v must be  
   auxiliary too, since x is involved in assignments to them.
2. {x, u, v} is a set of auxiliary variables for this program. Removing them yields  
   y := 2; k := 0;  
   **while** y < n do  
    y := y\*y; k := k+1  
   **od**
3. [9 = 3\*3 points] List the interference freedom checks for the following two standard proof outlines. Below, assume S₁, S₂, and S₄ are atomic.
   * {*q*₁} *S*₁ {*q*₂}
   * {*p*₁} *S*₂; {**inv** *p*₂} **while** *B* **do** {*p*₃} *S*₄ **od** {*p*₄}

Ans: The first thread has one atomic region and two conditions; the second thread has two  
atomic regions and four conditions, so altogether have (1 × 4) + (2 × 2) = 8 tests  
(unless some are duplicates).  
{q₁} S₁ {…} vs p₁, p₂, p₃, and p₄:  
{q₁ ∧ p₁} S₁ {p₁}  
{q₁ ∧ p₂} S₁ {p₂}  
{q₁ ∧ p₃} S₁ {p₃}  
{q₁ ∧ p₄} S₁ {p₄}  
{p₁} S₂ {…} vs q₁ and q₂:  
{p₁ ∧ q₁} S₂ {q₁}  
{p₁ ∧ q₂} S₂ {q₂}  
{p₃} S₄ {…} vs q₁ and q₂:  
{p₃ ∧ q₁} S₄ {q₁}  
{p₃ ∧ q₂} S₄ {q₂}

1. [6 points] Draw a full evaluation graph for the following program, starting in an arbitrary state σ. Indicate deadlocked configurations. Remember, evaluation of an await is atomic (is shown with exactly one arrow). y := 2; [y := y∗3; y := y+10 || *A*] where *A* ≡ **await** y < 9 **then** **if** y < 5 **then** y := y+3 **fi** **end**

Ans:   
Here’s an execution graph for this program

〈y := 2; [y := y∗3;y := y+10 || A], ∅〉

〈[y := y∗3;y := y+10 || A], {y = 2}〉

〈[y := y+10 || E], {y = 6}〉 〈[y := y∗3;y := y+10 || E], {y = 5}〉

〈[E || A], {y = 16}〉 〈[y := y+10 || A], {y = 6}〉 〈[y := y+10 || E], {y = 15}〉

(BLOCK)

〈[E || E], {y = 16}〉 〈[E || E], {y = 25}〉

1. [10 = 5\*2 points] Give the set of deadlock conditions for:

[ {*p*₁} *S*₁; {*q*₁} **await** *B*₁ **then** *T*₁ **end** {*r*₁}

|| {*p*₂} **await** *B*₂ **then** *S*₂ **end;** {*p*₃} **await** *B*₃ **then** *T*₃ **end** {*r*₂} || {*p*₃} *S*₃ {*r*₃} ]

Ans: The three threads have 1, 2, and 0 await statements respectively, which gives us  
(1+1) × (2+1) × (0+1) - 1 = 5 potential deadlock conditions (unless some are  
duplicates):  
(q₁ ∧ B̅₁) ∧ (p₂ ∧ B̅₂) ∧ r₃  
(q₁ ∧ B̅₁) ∧ (p₃ ∧ B̅₃) ∧ r₃  
(q₁ ∧ B̅₁) ∧ r₂ ∧ r₃  
r ₁ ∧ (p₂ ∧ B̅₂) ∧ r₃  
r ₁ ∧ (p₃ ∧ B̅₃) ∧ r₃