**PHƯƠNG PHÁP GIẢI QUYẾT XUNG ĐỘT GIỮA CÁC BIẾN ĐỌC VÀ GHI TRONG CHƯƠNG TRÌNH ĐA LUỒNG BẰNG THỰC THI TƯỢNG TRƯNG**

## Ví dụ minh hoạ

|  |  |
| --- | --- |
| int x = 1, y = 1, m = 0, n = 0; | |
| void\* thr1(void \* arg) {  x = y + 1;  m = y;  x = 0;  } | void\* thr2(void \* arg) {  y = x + 1;  n = x;  y = 0;  } |
| assert (!(m == 1 && n == 1)); | |

**ABSTRACTION METHOD**

**φ0 := ϕinit ∧ ρ ∧ ζ**

**Trong đó:**

**ϕinit**: trạng thái khởi tạo

**ρ** : mã hóa mỗi luồng riêng biệt

**ζ** : phát biểu rằng “mỗi hành động đọc của một biến ***v*** bất kì có thể đọc được kết quả của bất kì hành động ghi nào của ***v***

|  |  |
| --- | --- |
| int x­­­­0 = 1, y0 = 1, m0 = 0, n0 = 0; | |
| void\* thr1(void \* arg) {  x1 = y1 + 1;  m1 = y2;  x2 = 0;  } | void\* thr2(void \* arg) {  y3 = x3 + 1;  n1 = x4;  y4 = 0;  } |
| assert (!(m2 == 1 && n2 == 1)); | |

1. **Định nghĩa**

* ev,i là sự kiện tương ứng với vi.
* sv,i,j biểu thị cho liên kết đọc – ghi giữa vi và vj : *ev,j sẽ đọc giá trị được ghi bởi ev,i*.
* Ψv,i biểu thị các giá trị có thể đọc được của vi

*Xét ví dụ ψx,3 ở dưới:*

* + *Khi sx,i,j thì ex,j đọc được giá trị ghi vào bởi ex,i , hay xj = xi*

*(sx,0,3 ⇒ (x3 = x0)) ∧ (sx,1,3 ⇒ (x3 = x1)) ∧ (sx,2,3 ⇒ (x3 = x2))*

* + *ex,3 có thể đọc được giá trị ghi vào từ {ex,0 ; ex,1 ; ex,2}:*

*(sx,0,3 ∨ sx,1,3 ∨ sx,2,3)*

* + *ex,3 chỉcó thể đọc được 1 trong 3 giá trị ghi vào từ {ex,0 ; ex,1 ; ex,2}:*

*(sx,0,3 ⇒ ¬ (sx,1,3 ∨ sx,2,3))*

*∧ (sx,1,3 ⇒ ¬ (sx,0,3 ∨ sx,2,3))*

*∧ (sx,2,3 ⇒ ¬ (sx,0,3 ∨ sx,1,3))*

* ζv là tất cả các hành vi đọc khả thi của v

1. **Mã hóa ϕ**

ϕinit := (x0 = 1) ∧ (y0 = 1) ∧ (m0 = 0) ∧ (n0 = 0)

∧ (sm,1,2) *(do em,2 thực hiện sau khi toàn bộ thread hoàn thành)*

∧ (sn,1,2) *(do en,2 thực hiện sau khi toàn bộ thread hoàn thành)*

ϕerr := (m2 = 1) ∧ (n2 = 1)

1. **Mã hóa ρ**

ρthr1 := (x1 = y1 + 1) ∧ (m1 = y2) ∧ (x2 = 0)

ρthr2 := (y3 = x3 + 1) ∧ (n1 = x4) ∧ (y4 = 0)

ρmain := true

ρ := ρthr1 ∧ ρthr2 ∧ ρmain

1. **Mã hóa ζ**

* **x bao gồm tập đọc {x3; x4} và tập ghi {x0; x1; x2}**

ψx,3 := (sx,0,3 ⇒ (x3 = x0)) ∧ (sx,1,3 ⇒ (x3 = x1)) ∧ (sx,2,3 ⇒ (x3 = x2))

∧ (sx,0,3 ∨ sx,1,3 ∨ sx,2,3)

∧ (sx,0,3 ⇒ ¬ (sx,1,3 ∨ sx,2,3))

∧ (sx,1,3 ⇒ ¬ (sx,0,3 ∨ sx,2,3))

∧ (sx,2,3 ⇒ ¬ (sx,0,3 ∨ sx,1,3))

ψx,4 := (sx,0,4 ⇒ (x4 = x0)) ∧ (sx,1,4 ⇒ (x4 = x1)) ∧ (sx,2,4 ⇒ (x4 = x2))

∧ (sx,0,4 ∨ sx,1,4 ∨ sx,2,4)

∧ (sx,0,4 ⇒ ¬ (sx,1,4 ∨ sx,2,4))

∧ (sx,1,4 ⇒ ¬ (sx,0,4 ∨ sx,2,4))

∧ (sx,2,4 ⇒ ¬ (sx,0,4 ∨ sx,1,4))

**ζx := ψx,3 ∧ ψx,4**

* **y bao gồm tập đọc {y1; y2} và tập ghi {y0; y3; y4}**

ψy,1 := (sy,0,1 ⇒ (y1 = y0)) ∧ (sy,3,1 ⇒ (y1 = y3)) ∧ (sy,4,1 ⇒ (y1 = y4))

∧ (sy,0,1 ∨ sy,3,1 ∨ sy,4,1)

∧ (sy,0,1 ⇒ ¬ (sy,3,1 ∨ sy,4,1))

∧ (sy,3,1 ⇒ ¬ (sy,0,1 ∨ sy,4,1))

∧ (sy,4,1 ⇒ ¬ (sy,0,1 ∨ sy,3,1))

ψy,2 := (sy,0,2 ⇒ (y2 = y0)) ∧ (sy,3,2 ⇒ (y2 = y3)) ∧ (sy,4,2 ⇒ (y2 = y4))

∧ (sy,0,2 ∨ sy,3,2 ∨ sy,4,2)

∧ (sy,0,2 ⇒ ¬ (sy,3,2 ∨ sy,4,2))

∧ (sy,3,2 ⇒ ¬ (sy,0,2 ∨ sy,4,2))

∧ (sy,4,2 ⇒ ¬ (sy,0,2 ∨ sy,3,2))

**ζy := ψy,1 ∧ ψy,4**

* **m bao gồm tập đọc {m2} và tập ghi {m0; m1}**

ψm,2 := (sm,0,2 ⇒ (m2 = m0)) ∧ (sm,1,2 ⇒ (m2 = m1))

∧ (sm,0,2 ∨ sm,1,2)

∧ (sm,0,2 ⇒ ¬ sm,1,2) ∧ (sm,1,2 ⇒ ¬ sm,0,2)

**ζm := ψm,2**

* **n bao gồm tập đọc {n2} và tập ghi {n0; n1}**

ψn,2 := (sn,0,2 ⇒ (n2 = n0)) ∧ (sn,1,2 ⇒ (n2 = n1))

∧ (sn,0,2 ∨ sn,1,2)

∧ (sn,0,2 ⇒ ¬ sn,1,2) ∧ (sn,1,2 ⇒ ¬ sn,0,2)

**ζn := ψn,2**

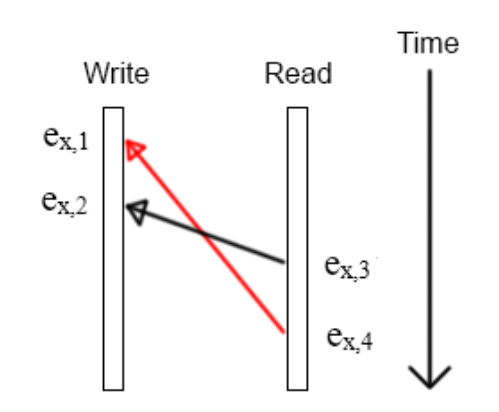
**ζ = ζx** ∧ **ζy** ∧ **ζm** ∧ **ζn**

**KERNEL REASON**

* Khác với các câu lệnh trong các luồng khác nhau có thể thực thi với thứ tự khác nhau, **các câu lệnh trong cùng một luồng bắt buộc phải có thứ tự thực thi là tuyến tính**
* Do đó, nếu sx,2,3 (hay ex,3 đọc được giá trị ghi bởi ex,2) thì:
  + Với thao tác ghi ex,1 xảy ra trước thao tác ghi ex,2 trong thr1
  + Với thao tác đọc ex,4 xảy ra sau thao tác đọc ex,3 trong thr2

⇒ ex,4 không thể đọc được giá trị ghi bởi ex,1 , hay ¬ sx,1,4

Tuy nhiên, ex,4 vẫn có thể đọc được giá trị ghi bởi ex,2



Mối quan hệ tuyến tính trong cùng một luồng

1. **Định nghĩa**

* Kernel reason κ là ràng buộc đọc ghi các biến dùng chung dựa trên mối quan hệ tuyến tính trong cùng một luồng của chương trình. Nó biểu thị rằng khi một hành động đọc *r* đọc được giá trị của hành động ghi *w* thì:
  + Các hành động đọc xảy ra sau *r* không thể đọc được giá trị của hành động ghi xảy ra trước hành động ghi *w*
  + Các hành động đọc xảy ra trước hành động đọc *r* không thể đọc được giá trị của các hành động ghi xảy ra sau hành động ghi *w*

1. **Mã hóa κ**

* **Kernel reason của x:**

κx,3 := (sx,1,3 ⇒ ¬ (sx,0,4)) ∧ (sx,2,3 ⇒ ¬ (sx,0,4 ∨ sx,1,4))

κx,4 := (sx,0,4 ⇒ ¬ (sx,1,3 ∨ sx,2,3)) ∧ (sx,1,4 ⇒ ¬ (sx,2,3))

κx := κx,3 ∧ κx,4

* **Kernel reason của y:**

κy,1 := (sy,3,1 ⇒ ¬ (sy,0,2)) ∧ (sy,4,1 ⇒ ¬ (sy,0,2 ∨ sy,3,2))

κy,2 := (sy,0,2 ⇒ ¬ (sy,3,1 ∨ sy,4,1)) ∧ (sy,3,2 ⇒ ¬ (sy,4,1))

κy := κy,1 ∧ κy,2

* **Kernel reason của m:**

κm  := true

* **Kernel reason của n:**

κn  := true

**κ := κx ∧ κy ∧ κm ∧ κn**