HƯỚNG DẪN BÀI TẬP CHƯƠNG 2

GIAO TÁC & ĐIỀU KHIỂN ĐỒNG THỜI CÁC GIAO TÁC

GV: Phạm Thị Bạch Huệ

Các dạng bài tập

1. Input: Lịch biểu S. (TH1: S gồm Read và Write TH2: S gồm cả Rlock và Wlock)
   Output: S khả tuần tự?
2. Input: S và các loại khóa khác nhau.
   Output: Quá trình thực hiện S.
3. Input: Các giao tác Ti.
   Output: Lịch biểu (gồm các Ti) khả tuần tự dùng các kỹ thuật khác nhau.
4. Kiểm tra khả năng xử lý đồng thời của các mức cô lập.
5. Input: Ứng dụng thực tế, yêu cầu xử lý đồng thời.
   Output: Dùng các phương thức khóa để đáp ứng yêu cầu ứng dụng.
6. Input: Lịch biểu S.
   Output: S có bị deadlock? Giải quyết hoặc ngăn ngừa?
7. Input: Lịch biểu S.
   Output: Điều khiển S dùng kỹ thuật nhãn thời gian.
   (TH1: NTG toàn phần, TH2: NTG bán phần, TH3: NTG đa phiên bản).
8. Input: Lịch biểu S.
   Output: Quá trình thực hiện S khi dùng kỹ thuật xác nhận hợp lệ.

Dạng bài tập 1: Input: Lịch biểu S

Output: S có khả tuần tự? Dùng đồ thị ưu tiên.

- **Dạng bài tập 1a**
- **Input: S: r1(A); w1(A); r2(A); w2(A); r1(B); w1(B); r2(B); w2(B)**
- **Output: S khả tuần tự?**

S

T1	T2
r(A)	
w(A)	
	r(A)
	w(A)
r(B)	
w(B)	
	r(B)
	w(B)

Bằng 4 phép hoán vị các thao tác khả hoán vị

S’

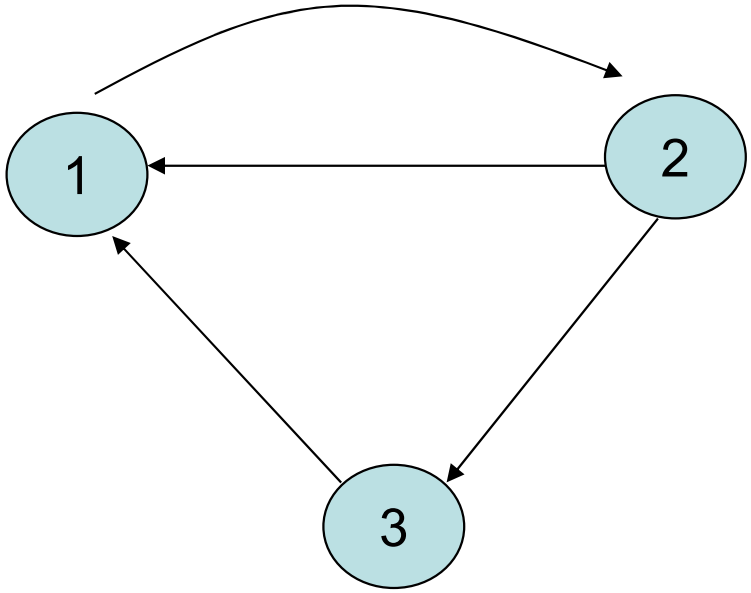
T1	T2
r(A)	
w(A)	
r(B)	
w(B)	
	r(A)
	w(A)
	r(B)
	w(B)

S khả tuần tự vì S khả tuần tự xung đột, T1 < T2

r2(Z), r2(Y), w2(Y), r3(Y), r3(Z), r1(X), w1(X), w3(Y), w3(Z), r1(X), r1(Y), w1(Y), w2(X)

T1	T2	T3
	r2(Z)	
	r2(Y)	
	w2(Y)	
		r3(Y)
		r3(Z)
r1(X)		
w1(X)		
		w3(Y)
		w3(Z)
r1(X)		
r1(Y)		
w1(Y)		
	w2(X)	

T2 < T3



Đồ thị có chu trình. Lịch biểu đã cho là không khả tuần tự xung đột

Dạng bài tập 2:

Input: Lịch biểu S, các phương thức khóa HQT CSDL dùng để thực hiện S, gồm:

a. Khóa đơn giản, dùng {L, UN}

b. Khóa đọc - ghi, dùng {SL, XL, UN}

c. Khóa đọc - ghi, cho phép tăng cấp, dùng {SL, XL, UN, upgrading)

d. Khóa đọc - ghi, update lock, dùng {SL, XL, UL, UN}

e. Khóa đọc - ghi và khóa tăng/giảm, dùng {SL, XL, IL, UN}

Output: Quá trình thực hiện S

Cho biết quá trình thực hiện của S

S: r1(A), r2(B), r3(C), r1(B), r2(C), r3(D), w1(A), w2(B), w3(C)

a. Dùng {lock, unlock}

T1	T2	T3
L(A)		
R(A)		
	L(B)	
	R(B)	
		L(C)
		R(C)
L(B) Chờ		
	L(C) Chờ	
		L(D)
		R(D)
		W(C)
		UN(C)

T1	T2	T3
	L(C)	
	R(C)	
		UN(D)
	W(B)	
	UN(B)	
L(B)		
R(B)		
	UN(C)	
W(A)		
UN(A)		
UN(B)		

S: r1(A), r2(B), r3(C), r1(B), r2(C), r3(D), w1(A), w2(B), w3(C)

a. Dùng {SL, XL}

T1	T2	T3
XL(A)		
R1(A)		
	XL(B)	
	R2(B)	
		XL(C)
		R3(C)
SL(B) Chờ		
	SL(C) Chờ	
		SL(D)
		R3(D)
		W3(C)
		Un(C)
	SL(C)	
	R2(C)	
		Un(D)
	W2(B)	
	Un(B)	
SL(B)		
R1(B)		
	Un(C)	
W1(A)		
Un(A)		
Un(B)		

Lịch tuần tự tương đương: T3 < T2 < T1

S: r1(A), r2(B), r3(C), r1(B), r2(C), r3(D), w1(A), w2(B), w3(C)

b. Dùng {SL, XL và khóa tăng cấp} upgrading

T1	T2	T3
SL(A)		
R1(A)		
	SL(B)	
	R2(B)	
		SL(C)
		R3(C)
SL(B)		
R1(B)		
	SL(C)	
	R2(C)	
		SL(D)
		R3(D)
XL(A)		
W1(A)		
	XL(B) Chờ	
		XL(C) Chờ
Un(A)		
Un(B)		
	XL(B)	
	W2(B)	
	Un(B)	
	Un(C)	
		XL(C)
		W3(C)
		Un(C)
		Un(D)

Lịch tuần tự tương đương: T1 < T2 < T3

S: r1(A), r2(B), r3(C), r1(B), r2(C), r3(D), w1(A), w2(B), w3(C)

c. Dùng {SL, XL và UL}

T1	T2	T3
UL(A)		
R1(A)		
	UL(B)	
	R2(B)	
		UL(C)
		R3(C)
SL(B) Chờ		
	SL(C) Chờ	
		SL(D)
		R3(D)
		XL(C)
		W3(C)
		Un(C)

	SL(C)	
	R2(C)	
		Un(D)
	XL(B)	
	W2(B)	
	Un(B)	
SL(B)		
R1(B)		
		Un(C)
XL(A)		
W1(A)		
Un(A)		
Un(B)		

Lịch tuần tự tương đương: T3 < T2 < T1

S: r1(A), r2(B), inc1(B), inc2(C), w2(D)

d. Dùng {SL, XL và IL}

T1	T2
SL(A)	
R1(A)	
	SL(B)
	R2(B)
IL(B) Chờ	
	IL(C)
	Inc(C)
	XL(D)
	W2(D)
	Un(B)
	Un(C)
	Un(D)
IL(B)	
Inc(B)	
Un(A)	
Un(B)	

Lịch tuần tự tương đương: T2 < T1

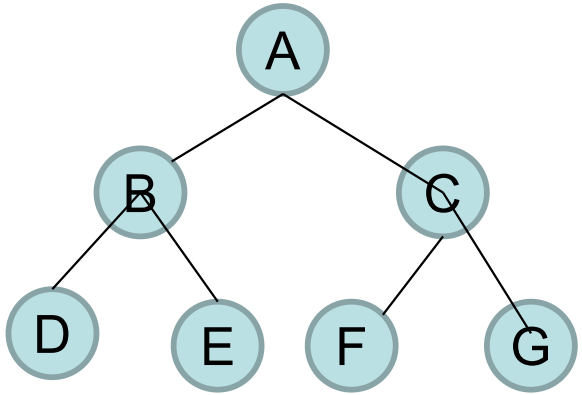
Dạng bài tập 3:
Input: Các GT Ti
Output: Lịch biểu (gồm các Ti) khả tuần tự dùng một KT ĐKĐT nào đó.
a.Lịch biểu hợp lệ gồm các GT nhất quán và thỏa 2PL.
b.Ti thao tác trên dữ liệu có cấu trúc cây

Cho T1: r(A), w(B), r(C), w(E)
T2: r(C), w(F)
T3: r(B), r(E)
a. Đề nghị một S khả tuần tự gồm T1, T2, T3 dựa trên nguyên tắc: “Lịch biểu hợp lệ gồm các GT nhất quán và thỏa 2PL thì khả tuần tự”. T2<T3<T1

T1	T2	T3
	SL(C)	
	R(C)	
SL(A)		
R(A)		
		SL(B)
		R(B)
	XL(F)	
	W(F)	
XL(B) chờ		
		SL(E)
		R(E)
	Un(C)	

T1	T2	T3
		Un(B)
		Un(E)
XL(B)		
W(B)		
SL(C)		
R(C)		
XL(E)		
W(E)		
Un(A)		
Un(B)		
Un(C)		
Un(E)		

Cho T1: r(A), w(B), r(C), w(E)
T2: r(C), w(F)
T3: r(B), r(E)
b. Đề nghị một S khả tuần tự gồm T1, T2, T3 dựa trên nguyên tắc khóa trên ĐVDL phân cấp.



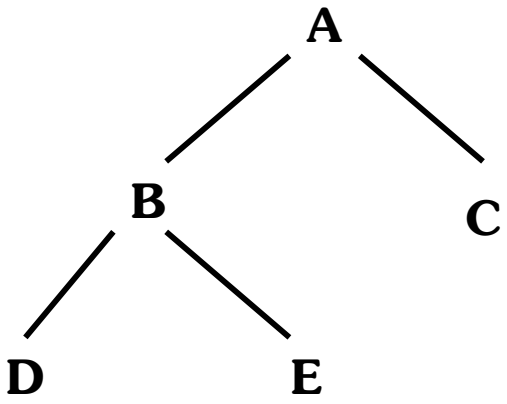
T1	T2	T3
	L(C)	
	R(C)	
		L(B)
		R(B)
L(A)		
R(A)		
	L(F)	
	W(F)	
		L(E)
		R(E)
	Un(C)	
	Un(F)	

		Un(B)
		Un(E)
L(B)		
W(B)		
L(C)		
R(C)		
L(E)		
W(E)		
Un(A)		
Un(B)		
Un(C)		
Un(E)		

Dạng bài tập 3a

Cho T1:r(A);r(B); w(A); r(C); w(E) T2: r(A); w(C) T3: r(E); r(D)

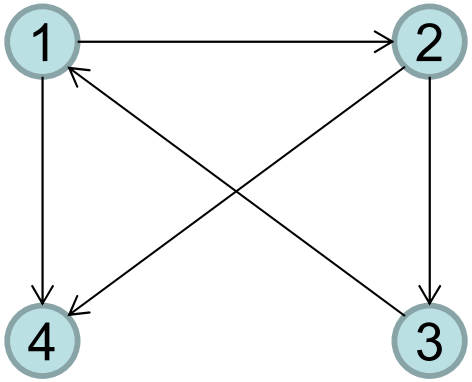
Cho một lịch biểu (gồm các Ti) khả tuần tự dùng lock, unlock và áp dụng nghi thức khóa phân cấp.



T1	T2	T3
l1(A); r1(A)		
l1(B); r1(B)		
w1(A);		
l1(C); r1(C)		
l1(E);w1(E)		
U1(A),U1(B);u1(C; u1(E)		
	l2(A);r2(A)	
		l3(E); r3(E)
		l3(D); r3(D)
	l2(C);w(C)	
	u2(A), u2(C)	
		u3(E); u3(D)

Dạng bài tập 4: Input: Lịch biểu S gồm các Ti cho trước
Output: Có deadlock không? Giải quyết?

T1: l1(A); r1(A); l1(B); w1(B); u1(A); u1(B)
T2: l2(C); r2(C); l2(A); w2(A); u2(C); u2(A)
T3: l3(B); r3(B); l3(C); w3(C); u3(B); u3(C)
T4: l4(D); r4(D); l4(A); w4(A); u4(D); u4(A)



T1	T2	T3	T4
l1(A); r1(A);			
	l2(C); r2(C);		
		l3(B); r3(B);	
			l4(D); r4(D);
	l2(A);		
		l3(C);	
			l4(A);
l1(B);			

Đồ thị có chu trình → có deadlock. Giải quyết deadlock bằng cách hủy giao tác ứng với node có nhiều cung vào ra nhất.

Dạng bài tập 5: Input: Lịch biểu S gồm các GT Ti (có xảy ra deadlock)
Output: Hãy ngăn ngừa deadlock dùng pp sắp xếp ĐVDL theo thứ tự alphabet.

T1: l1(A); r1(A); l1(B); w1(B); u1(A); u1(B) → lock trên đvdl theo thứ tự alphabet
T2: l2(C); r2(C); l2(A); w2(A); u2(C); u2(A) → T2: l2(A); l2(C); r2(C); w2(A);u2(C); u2(A)
T3: l3(B); r3(B); l3(C); w3(C); u3(B); u3(C) → lock trên đvdl theo thứ tự alphabet
T4: l4(D); r4(D); l4(A); w4(A); u4(D); u4(A) → T4: l4(A); l4(D); r4(D); w4(A); u4(D); u4(A)

T1	T2	T3	T4
L1(A)			
R1(A)			
	L2(A) Chờ		
		L3(B)	
		R3(B)	
			L4(A) Chờ
		L3(C)	
		W3(C)	
L1(B) Chờ			
		U3(B)	
		U3(C)	

L1(B)			
W1(B)			
U1(A)			
	L2(A)		
	L2(C)		
U1(B)			
	R2(C)		
	W2(A)		
	U2(C)		
	U2(A)		
			L4(A)
			L4(D)
			R4(D)
			W4(A)
			U4(D)
			U4(A)

Dạng bài tập 6a: Input: Lịch biểu S gồm các Ti cho trước
Output: Hãy ngăn ngừa deadlock dùng thuật toán Wait-Die

st1< st2<st3<st4

T1: l1(A); r1(A); l1(B); w1(B); u1(A); u1(B)
T2: l2(C); r2(C); l2(A); w2(A); u2(C); u2(A)
T3: l3(B); r3(B); l3(C); w3(C); u3(B); u3(C)
T4: l4(D); r4(D); l4(A); w4(A); u4(D); u4(A)

T1	T2	T3	T4
l1(A), r1(A)			
	l2(C), r2(C)		
		l3(B), r3(B)	
			l4(D) r4(D)
	l2(A) Rollback		
		l3(C); W(C)	
			l4(A) Rollback
L1(B) Chờ			

		W3(C)	
		U3(B)	
		U3(C)	
L1(B)			
W1(B)			
U1(A)			
U1(B)			
	L2(C)		
	R2(C)		
			L4(D)
			R4(D)
	L2(A)		
	w2(A)		
	U2(A)		
	U2(C)		
			L4(A), w4(A)
			U4(A), u4(D)

Dạng bài tập 6b: Input: Lịch biểu S gồm các GT Ti (có xảy ra deadlock)
Output: Hãy ngăn ngừa deadlock dùng thuật toán Wound-Wait

st1< st2<st3<st4

T1: l1(A); r1(A); l1(B); w1(B); u1(A); u1(B)
T2: l2(C); r2(C); l2(A); w2(A); u2(C); u2(A)
T3: l3(B); r3(B); l3(C); w3(C); u3(B); u3(C)
T4: l4(D); r4(D); l4(A); w4(A); u4(D); u4(A)

T1	T2	T3	T4
l1(A), r1(A)			
	l2(C), r2(C)		
		l3(B) r3(B)	
			l4(D), r4(D)
	l2(A) Chờ		
		l3(C) Chờ	
			l4(A) Chờ
L1(B) Cho T3 rollback			

W1(B)			
U1(A)			
U1(B)			
	L2(A)		
	w2(A)		
	U2(A)		
	U2(C)		
			L4(A)
			w4(A)
			U4(A), u4(D)
		L3(B), r3(B)	
		L3(C), w3(C)	
		U3(B)	
		U3(C)	

Dạng bài tập 7: Input: Lịch biểu S
Output: Quá trình thực hiện của S dùng KT ĐKĐT dựa trên NTG

T1	T2	T3	T4	A
150	200	175	225	RT = WT = 0
R1(A)				RT = 150
W1(A)				WT = 150
	R2(A)			RT = 200
	W2(A)			WT = 200
		R3(A)		T3 rollback
			R4(A)	RT = 225

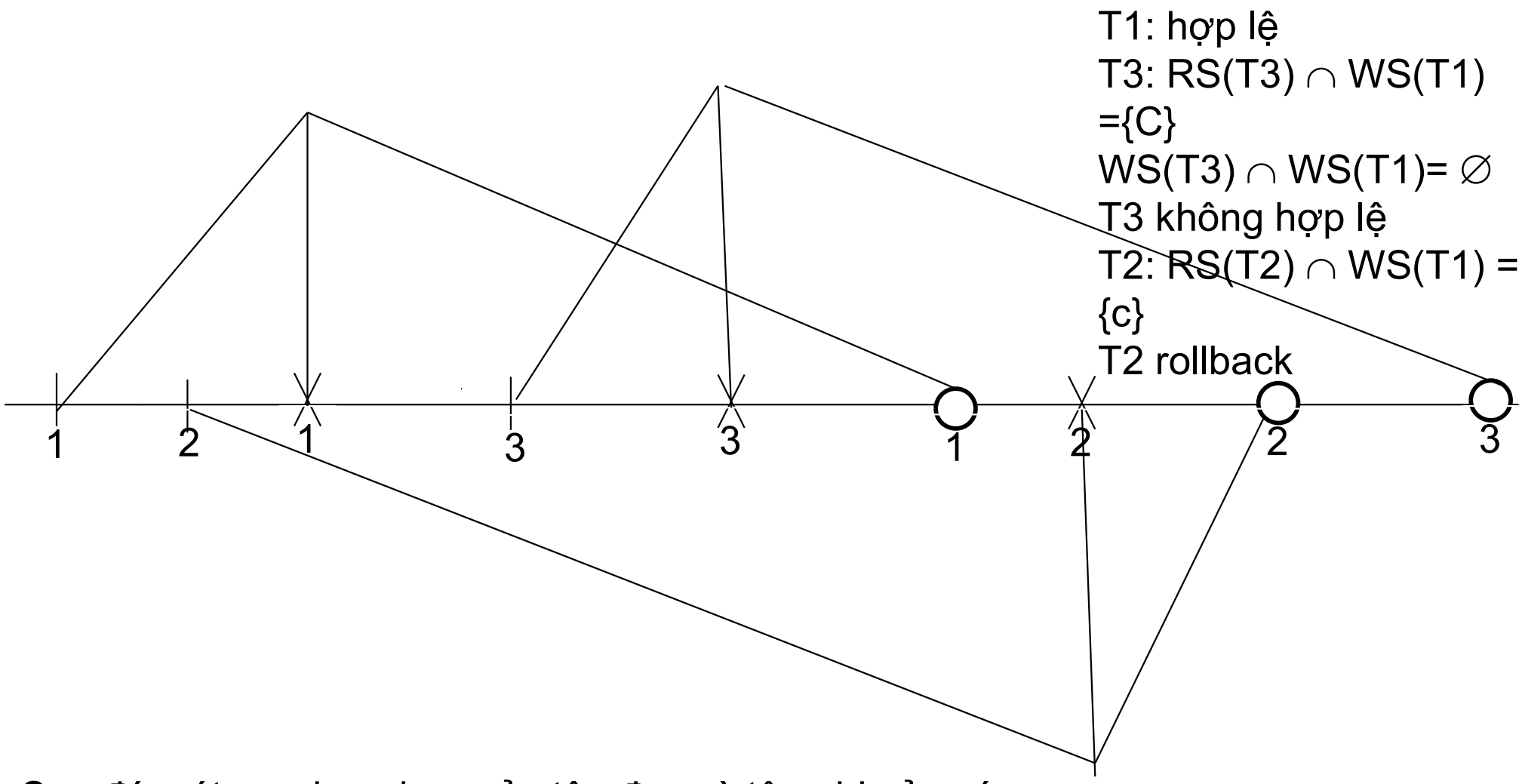
Dạng bài tập 8: Input: Lịch biểu S

Output: Quá trình thực hiện của S dùng KT ĐKĐT dựa trên NTG đa phiên bản

T1	T2	T3	T4	A_0	A_{150}	A_{200}
150	200	175	225	RT = WT = 0		
R1(A)				RT = 150		
W1(A)					RT=WT = 150	
	R2(A)				RT = 200	
	W2(A)					RT= WT = 200
		R3(A)			RT = 200	
			R4(A)			RT = 225

Dạng bài tập 9: Input: Lịch biểu S
Output: Quá trình thực hiện của S dùng KT xác nhận hợp lệ

$R_1(A, B), R_2(B, C), V_1, R_3(C, D), V_3, W_1(C), V_2, W_2(A), W_3(D)$

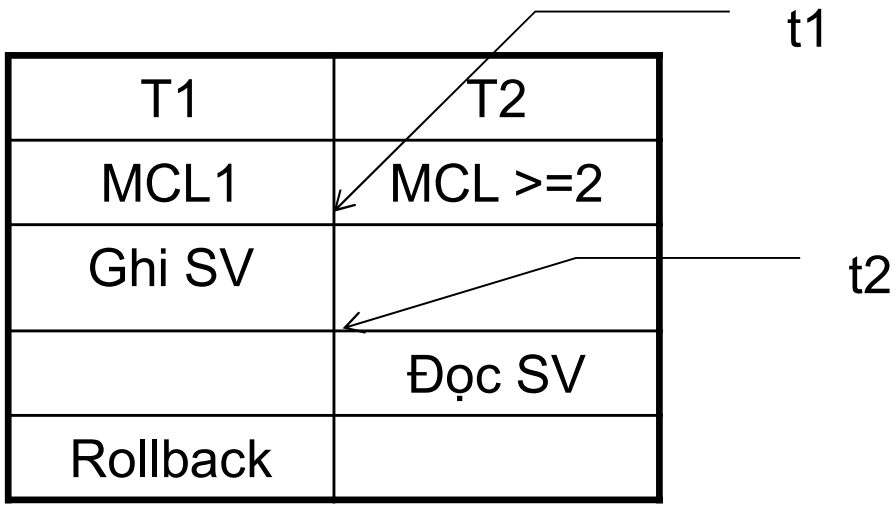


Sau đó, xét sự giao nhau của tập đọc và tập ghi của các giao tác như trong ví dụ của bài giảng.

Dạng bài tập 4 Kiểm tra khả năng xử lý đồng thời của các mức cô lập

Read Uncommitted (MCL1)	Read committed (MCL2)	Repeatable read (MCL3)	Serializable (MCL4)
1. Khóa ghi? 2. Khóa đọc? 3. Còn xảy ra tình trạng gì của ĐKĐT?	1. Khóa ghi? 2. Khóa đọc? Thời gian khóa đọc? 3. Còn xảy ra tình trạng gì của XLĐT?	1. Khóa ghi? 2. Khóa đọc? Thời gian khóa đọc? 3. Còn xảy ra tình trạng gì của XLĐT?	1. Khóa ghi? 2. Khóa đọc? Thời gian khóa đọc? 3. Còn xảy ra tình trạng gì của XLĐT?
3.1 Lost Update (TH1)	6.1 Lost Update (TH1)	9.1 Lost Update (TH1)	12.1 Lost Update (TH1)
3.2 Dirty read	6.2 Dirty read	9.2 Dirty read	12.2 Dirty read
3.3 Repeatable read	6.3 Repeatable read	9.3 Repeatable read	12.3 Repeatable read
3.4 Phantom	6.4 Phantom	9.4 Phantom	12.4 Phantom

TH1: MCL1 có khóa ghi khi ghi?



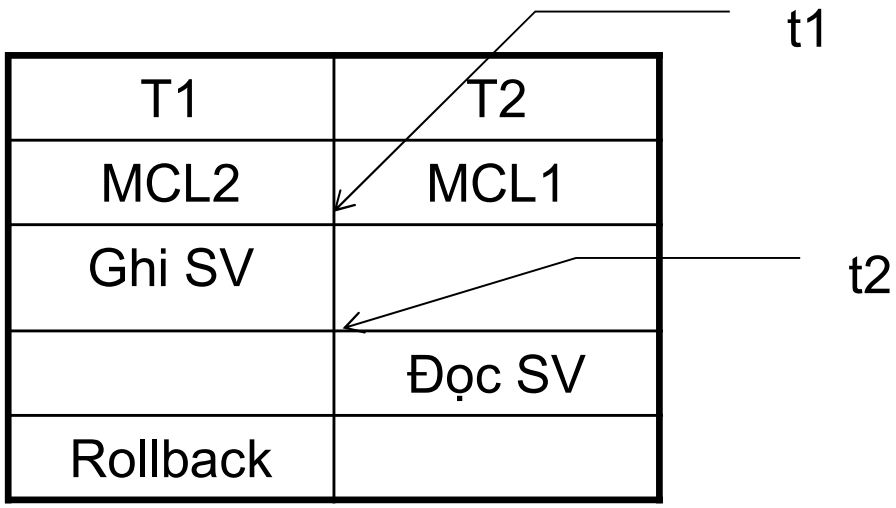
Giả sử ta biết rằng T2 có mức cô lập >=2 thì sẽ khóa đọc khi đọc.

Nếu T2 đọc ra dữ liệu tại thời điểm t2 có nghĩa là T1 không khóa ghi trên SV nên T2 khóa đọc và đọc dữ liệu được ngay.

Nếu T2 đọc ra dữ liệu tại thời điểm t1 có nghĩa là T1 có khóa ghi trên SV và T2 chờ cho T1 hoàn tất, nhả khóa ghi xong rồi mới đọc, nghĩa là T1 có khóa trên bảng SV khi ghi.

→ MCL1 khắc phục được Lost Update TH1.

TH2: MCL1 có khóa đọc khi đọc?



Giả sử ta biết rằng T2 có mức cô lập =2 thì sẽ khóa ghi khi ghi.

Nếu T2 đọc ra dữ liệu tại thời điểm t2 có nghĩa là T2 không khóa đọc trên SV nên đọc dữ liệu được ngay và đọc ra dữ liệu sai.

Nếu T2 đọc ra dữ liệu tại thời điểm t1 có nghĩa là T2 có khóa đọc trên SV và T2 chờ cho T1 hoàn tất, nhả khóa ghi xong rồi mới đọc.

→ MCL1 vẫn còn gặp phải tình trạng Dirty read.