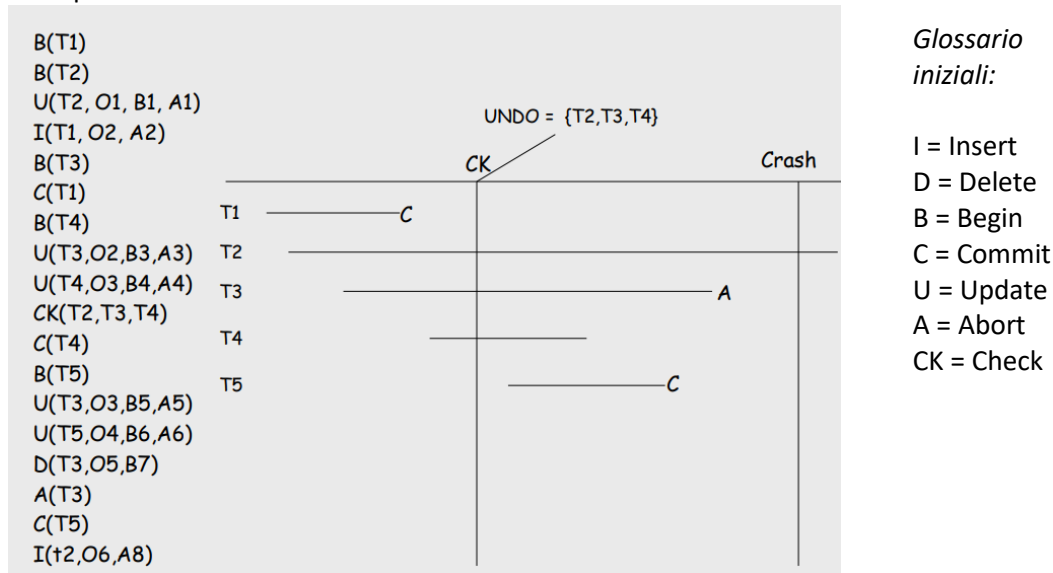


Nel caso delle transazioni, siamo interessati alla ripresa a caldo, composta da 4 fasi:

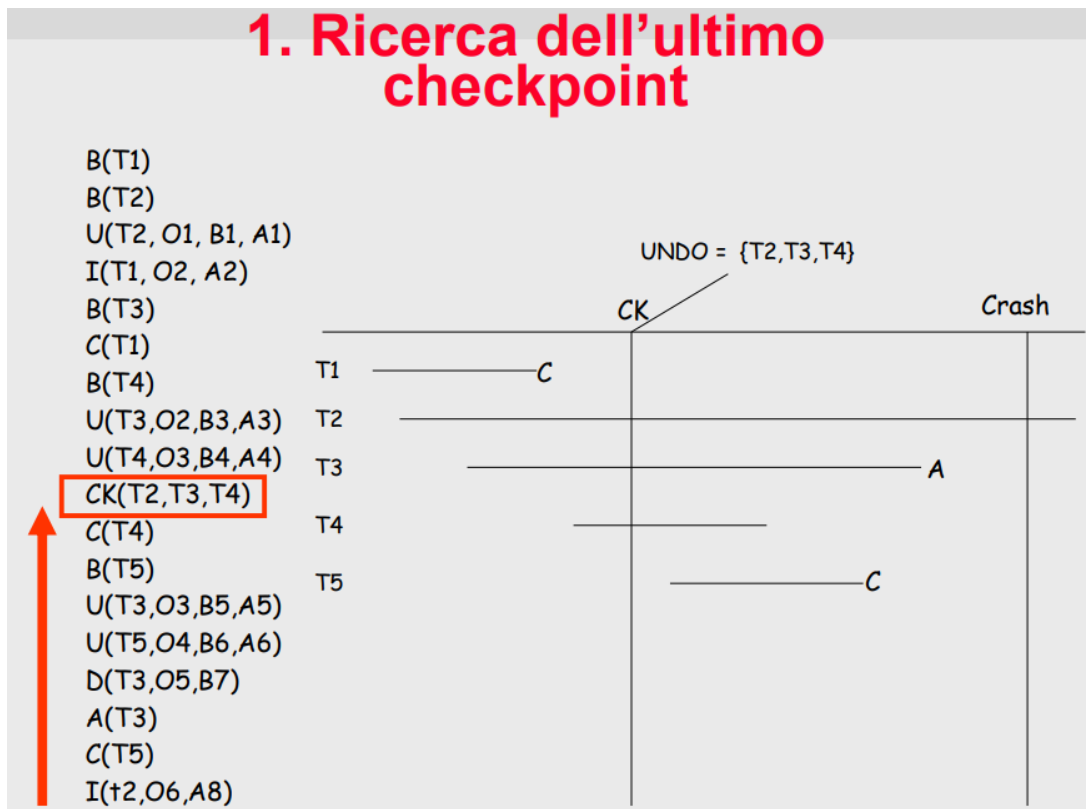
1. trovare l'ultimo checkpoint (ripercorrendo il log a ritroso)
2. costruire gli insiemi UNDO (transazioni da disfare) e REDO (transazioni da rifare)
3. ripercorrere il log all'indietro, fino alla più vecchia azione delle transazioni in UNDO e REDO, disfacendo tutte le azioni delle transazioni in UNDO
4. ripercorrere il log in avanti, rifacendo tutte le azioni delle transazioni in REDO

Esempio concreto di istruzioni:



1) Si cerca l'ultimo checkpoint, partendo da sotto e arrivando a:

1. Ricerca dell'ultimo checkpoint



2) Si costruiscono gli insiemi UNDO e REDO.

All'interno di UNDO andremo ad inserire:

- inizialmente T2 e T3, dato che su di esse pone controllo il check
- si ha il commit di T4 in C(T4) e la transazione viene posta in REDO
- qualsiasi altra operazione (D/A/I/U) prevede che la transazione rimanga in UNDO; è il caso di T5 che inizia (B/BEGIN), di T3 che aggiorna l'oggetto O3 in B5, di T5 che aggiorna l'oggetto O4 in B6, la cancellazione dell'oggetto O5 in B7 e l'abort di A3 (che fanno in modo T2,T3,T5 rimangano in UNDO)
- si ha il commit di T5 e alla fine si rimane con:
 $C(T5) \rightarrow UNDO = \{T2, T3\}. REDO = \{T4, T5\}$

2. Costruzione degli insiemi UNDO e REDO

B(T1)	0. UNDO = {T2,T3,T4}. REDO = {}	
B(T2)		
8. U(T2, O1, B1, A1)	1. C(T4) → UNDO = {T2, T3}. REDO = {T4}	
I(T1, O2, A2)	2. B(T5) → UNDO = {T2,T3,T5}. REDO = {T4}	Setup
B(T3)	3. C(T5) → UNDO = {T2,T3}. REDO = {T4, T5}	
C(T1)		
B(T4)		
7. U(T3,O2,B3,A3)		
9. U(T4,O3,B4,A4)		
CK(T2,T3,T4)		
1. C(T4)		
2. B(T5)		
6. U(T3,O3,B5,A5)		
10. U(T5,O4,B6,A6)		
5. D(T3,O5,B7)		
A(T3)		
3. C(T5)		
4. I(T2,O6,A8)		

3) Dato l'insieme degli UNDO, si ripercorre dalla fine all'inizio tutto il log, rifacendo le operazioni che riguardano T2 e T3. Quindi:

- si ha un INSERT di T2 su O6; questo dovrà essere cancellato (quindi quando si ha un INSERT si ha la cancellazione, per permettere di rifarla → D(O6)
- si ha il DELETE di O5 in B7 e dunque → O5 = B7 (che era lo stato precedente, ma è la convenzione che si adotta)
- si ha un UPDATE DI T3 su O3 e → O3=B5
- prima del CHECK si ha un UPDATE di T3 su O2 e → O2=B3
- finalmente, si ha un UPDATE di T2 su O1 in B1 → O1=B1

3. Fase UNDO

B(T1)	0. UNDO = {T2,T3,T4}. REDO = {}	
B(T2)		
8. U(T2, O1, B1, A1)	1. C(T4) → UNDO = {T2, T3}. REDO = {T4}	
I(T1, O2, A2)		
B(T3)	2. B(T5) → UNDO = {T2,T3,T5}. REDO = {T4}	Setup
C(T1)	3. C(T5) → UNDO = {T2,T3}. REDO = {T4, T5}	
B(T4)		
7. U(T3,O2,B3,A3)	4. D(O6)	
9. U(T4,O3,B4,A4)	5. O5 = B7	
CK(T2,T3,T4)		
1. C(T4)	6. O3 = B5	Undo
2. B(T5)		
6. U(T3,O3,B5,A5)	7. O2 = B3	
10. U(T5,O4,B6,A6)	8. O1=B1	
5. D(T3,O5,B7)		
A(T3)		
3. C(T5)		
4. I(T2,O6,A8)		

4) Per la fase di REDO, si fa la stessa cosa al contrario, dunque partendo dall'inizio e arrivando alla fine. Sull'insieme si ha da considerare T4 e T5

Si nota che le operazioni che li riguardano sono:

- U(T4,O3,B4,A4) e quindi porta a dire O3 = B4
- U(T5,O4,B6,A6) e quindi porta a dire O4 = A6

4. Fase REDO

B(T1)	0. UNDO = {T2,T3,T4}. REDO = {}	
B(T2)		
8. U(T2, O1, B1, A1)	1. C(T4) → UNDO = {T2, T3}. REDO = {T4}	
I(T1, O2, A2)	2. B(T5) → UNDO = {T2,T3,T5}. REDO = {T4}	Setup
B(T3)	3. C(T5) → UNDO = {T2,T3}. REDO = {T4, T5}	
C(T1)		
B(T4)		
7. U(T3,O2,B3,A3)	4. D(O6)	
9. U(T4,O3,B4,A4)	5. O5 = B7	
CK(T2,T3,T4)		
1. C(T4)	6. O3 = B5	Undo
2. B(T5)		
6. U(T3,O3,B5,A5)	7. O2 = B3	
10. U(T5,O4,B6,A6)	8. O1=B1	
5. D(T3,O5,B7)		
A(T3)	9. O3 = A4	
3. C(T5)		Redo
4. I(T2,O6,A8)	10. O4 = A6	

UNDO(disfare): in caso di fallimento della transazione deve essere possibile "disfare" l'azione svolta sui dati

In merito alle transazioni che richiedono UNDO:

- Nel seguente esempio, UNDO dovrà essere eseguito su T5, T6, T8 perché T7 è stata committata, mentre T8 esegue un INSERT senza commit e T6 esegue un UPDATE senza commit (anche T5 se eseguisse cose sarebbe persa e richiede UNDO)

Sia data la seguente porzione di log fino al guasto: CK(T5,T6), B(T7), U(T7,O6,B6,A6), U(T6, O3, B7, A7), B(T8), C(T7), I(T8,O5,A5). Quali

1. T5,T6,T8
2. T5,T6,T7,T8
3. T7
4. T7,T8

REDO(rifare): se la transazione ha avuto successo ma le modifiche al DB non sono state rese permanenti, le modifiche vanno ripetute.

In merito alle transazioni che richiedono REDO:

Sia data la seguente porzione di log fino al guasto: CK(T5,T6), B(T7), U(T7,O6,B6,A6), U(T6, O3, B7, A7), B(T8), C(T7), I(T8,O5,A5). Quali

Il commit è stato fatto su T7 che però non ha eseguito operazioni particolari di scrittura, è stata solo committata. Richiede quindi REDO per fare la sua modifica.

Quindi REDO → T7

Esercizio 1

Descrivere la ripresa a caldo, indicando la costituzione progressiva degli insiemi di UNDO e REDO e le azioni di recovery, a fronte del seguente log:

```
DUMP, B(T1), B(T2), B(T3), I(T1, O1, A1), D(T2, O2, B2), B(T4),  
U(T4, O3, B3, A3), U(T1, O4, B4, A4), C(T2), CK(T1, T3, T4), B(T5), B(T6),  
U(T5, O5, B5, A5), A(T3), CK(T1, T4, T5, T6), B(T7), A(T4),  
U(T7, O6, B6, A6), U(T6, O3, B7, A7), B(T8), A(T7), guasto
```

Si percorre il log a ritroso fino al più recente checkpoint, cioè CK(T1, T4, T5, T6).

Si mette tutto in UNDO → UNDO = {T1, T4, T5, T6}

Si considera dentro l'UNDO anche T7 che fa l'ABORT e T8 che era già nell'UNDO.

Nessuna transazione ha fatto il commit e si ha REDO{}

UNDO = {T1, T4, T5, T6, T7, T8}

A questo punto si parte al contrario e si ripetono le seguenti operazioni: (mettendo precedente e successivo all'operazione):

- 1) U (O3, B7)
- 2) U (O6, B6)
- 3) U (O5, B5)
- 4) U (O4, B4)
- 5) U (O3, B3)
- 6) Delete di O1 (essendo l'oggetto perso O1, vado a cancellarlo così lo reinserirò ancora)

Viene poi ripercorso in avanti tutto il log per rieseguire le operazioni di REDO.

Prima quindi faccio la ripresa a freddo, prendendo lo stato del disco al momento del dump e ripeto tutte le operazioni fino a prima del guasto.

Con la ripresa a caldo, si crea un hardware utile che esegue correttamente le operazioni.

Dato il seguente schedule nel log fino ad un guasto

```
..., CHECKPOINT(T1), BEGIN_TR(T2), ..., BEGIN_TR(T3), ...,  
COMMIT(T1), ..., COMMIT (T3), Guasto
```

Vengono omessi update, insert and delete per leggibilità e perché non rilevanti alla domanda. Di quale/i transazione/i occorre fare il REDO?

Essendo che guardiamo solo le transazioni che fanno un commit, allora andremo a fare il REDO di T1 e T3.