1. Write-read conflict

|  |  |
| --- | --- |
| T1 | T2 |
| R(x) |  |
| R(y) |  |
| W(x) |  |
|  | R(x) |
|  | R(y) |
|  | W(y) |
|  | R(x) |
|  | R(y) |
|  | W(x) |
|  | R(z) |
|  | W(z) |

T1 writing x in before T2 writing x may cause a dirty read. Since T1 never commit its updates, rollback T1 would cause both T2 reading x give us a false result that does not match with

The actual value written by T2

1. Read-write conflict

|  |  |
| --- | --- |
| T1 | T2 |
|  | R(x) |
|  | R(y) |
|  | W(y) |
| R(x) |  |
| R(y) |  |
| W(x) |  |
|  | R(x) |
|  | R(y) |
|  | W(x) |
|  | R(z) |
|  | W(z) |

This might also cause a read-write conflict since T2 read x twice while T1 updated x in between.

The same read query might result in different value, causing an unrepeatable read

1. Write-write conflict

|  |  |
| --- | --- |
| T1 | T2 |
|  | R(x) |
|  | R(y) |
|  | W(y) |
|  | R(x) |
| R(x) |  |
| R(y) |  |
| W(x) |  |
|  | R(y) |
|  | W(x) |
|  | R(z) |
|  | W(z) |

The T1 write x is overwritten by T2 write x happens after T1 actions, causing a lose update.