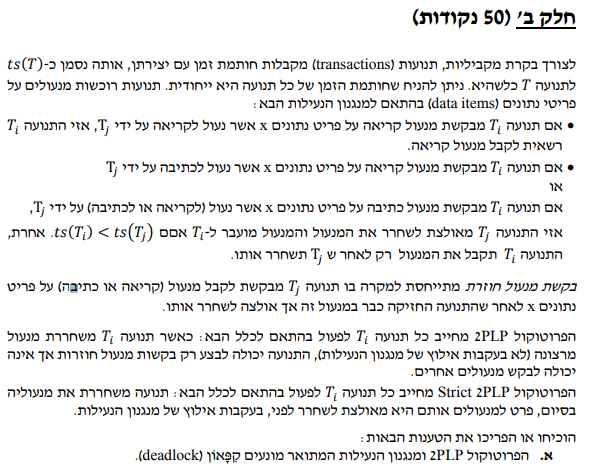
בס"ד



***פתרון:*** נוכיח, שלא מתקיים קיפאון, בהינתן כללי הנעילה, והפרוטוקול 2PLP.

יהיו , ונניח בה"כ, ש- .

נשים לב, כי אם שחרר בעבר מנעול מרצונו, אין הוא יכול לבקש מנעולים, שלא היו ברשותו.

נניח בהוכחה, שאם אכן שחרר מנעולים מרצונו החופשי, הוא יבקש רק מנעולים חוזרים, שהיו אצלו בעבר והיה מאולץ לשחרר אותם (זה נכון לכל המשך ההוכחה בסעיף שמקיימים את הדרישה של מנעול חוזר).

אחרת, הוא חופשי לבקש מנעולים ואם לא, אין לו הגבלות על בקשת המפתחות.

נחלק למקרים.

1. נניח, ש- מחזיק מנעול קריאה על x, ו- מבקש מנעול קריאה על x.

במקרה הזה מקבל את מנעול הקריאה- ללא קשר לחותמת הזמן, ולא מתבצע קיפאון (מהגדרת כללי הנעילות).

1. נניח, ש- מחזיק מנעול כתיבה על x, ו- מבקש מנעול קריאה\כתיבה על x או מבקש מנעול קריאה כאשר מחזיק מנעול כתיבה.

מהגדרת כללי הנעילה, בכלל קיום אי- השיוויון (\*), מאולצת לשחרר את המנעול, והוא מועבר ל-.

במקרה הזה לא יקרה קיפאון.

1. נניח, ש- מחזיק מנעול כתיבה על x, ו- מבקש מנעול קריאה\כתיבה על x.

בגלל (\*), מחכה, ש-- ישחרר את המנעול.

* אם משחרר את המנעול בסוף הפעולה שלו (כולל המקרה, שהוא מבצע פעולות נוספות על משתנים, שלא נעולים ע"י ), סיימנו. במקרה הזה, לא יכול לבקש יותר מנעולים, מלבד המנעולים, שמקיימים את התנאי של בקשת מנעול חוזרת.
* אחרת, נניח, שהוא לא יכול לשחרר את המנעול, כי הוא מחכה לשחרור של מנעול, שמחזיק .

ומכיוון ש-(\*) מתקיים, מאולץ לשחרר לו את המנעול, ו- תופס את המנעול, ממשיך ומסיים את הפעולה, ולשחרר את המנעול, ש- מחכה לו.

לכן במקרה הזה לא יקרה קיפאון.

לסיכום, עברנו על כל המקרים האפשריים, והראנו, שבכל אחד מהם לא יהיה קיפאון.



***פתרון:*** נראה דוגמה נגדית, שבה טרנזקציה אחת מבצעת שינוי אסור לנתונים, ונגיע למצב של *Read Dirty, שיוביל למצב של cascading abort.*

**דוגמה:**

יהיו טרנזקציות, כלהלן:

*ונתבונן במקרה, שמתקיים: .*

*נעיין בהיסטוריה הבאה:*

*בהיסטוריה הנתונה, מחזיק מנעול כתיבה על x, משחרר אותו באופן מאולץ לבקשת (לפי הגדרת כללי הנעילה), קורא את המשתנה x, ש- שינה, בטרם שחרור המנעול. נניח, שברגע הזה יעשה abort (למשל, הוא הכניס ל-x ערך, שאסור על פי אילוץ מהגדרת המסד), הערך של x יחזור להיות כמו שהיה לפני תחילת הטרנזקציות. אבל יחזיק את הערך של x, ששונה ע"י . לכן המידע, שברשותו, לא יהיה אמין (במקרה הזה מתבצע Read Dirty –). עקב כך, גם ייאלץ לעשות abort, מה שיגרור cascading abort.*