

קורס 1-02-326: הנדסת מערכות בסיסי נתונים

פרויקט 1: גילוי קונפליקטים בטרנזקציות

מרצה: מייקל מאי
תאריך הגשה: 11 לאפריל 2010

1 הקדמה

דיברנו בכיתה על מספר סוגי קונפליקטים (התנגשויות) שונים שיכולים להתרחש בין טרנזקציות. ההתנגשויות הכי שכיחות הן Read-Write (RW), Write-Read (WR) ו Write-Write (WW). כמו-כן דיברנו על המושגים View Serializability (VS) ו Conflict Serializability (CS), שהן תכונות של תזמונים האומרות האם ניתן לסדר את התזמון באופן סדרתי (טרנזקציות עוקבות) ועדיין לשמור על התוצאה הסופית. Strict Two Phase Locking (Strict 2PL), אלגוריתם שגם עליו דיברנו בכיתה, מבטיח שכל התזמונים השזורים ניתנים לסידור. Strict 2PL מאוד שמרני וחוסם הרבה תזמונים שלמעשה ניתנים לסידור. בתרגיל זה אתם תפתחו תוכנה שתנתח תזמונים שונים ותקבע באופן אוטומטי האם הם ניתנים לסידור והאם הם מותרים ע"י Strict 2PL.

2 תיאור הקלט

הקלט לתכנה יהיה קובץ יחיד שיכיל תזמון שזור עבור טרנזקציה אחת או יותר. כדי להקל על ניתוח הקובץ, על הקלט להיות מסודר בצורה הבאה:
להלן תזמון עבור T_1 ; T_2 ; T_3 :

T_1	T_2	T_3
R(A)		
W(C)	R(B)	
	R(C)	R(C)
R(A)	W(A)	
Commit		R(A)
	Commit	Commit

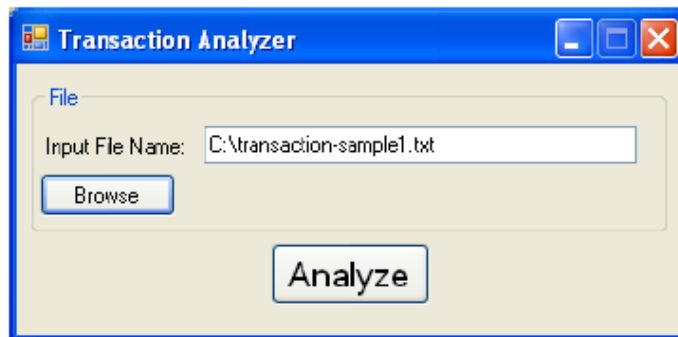
הקובץ עבור התזמון הנ"ל יכיל:

3
T1;T2;T3
3
A;B;C

11
T1:R(A)
T2:R(B)
T1:W(C)
T2:R(C)
T3:R(C)
T1:R(A)
T2:W(A)
T1:Commit
T3:R(A)
T2:Commit
T3:Commit

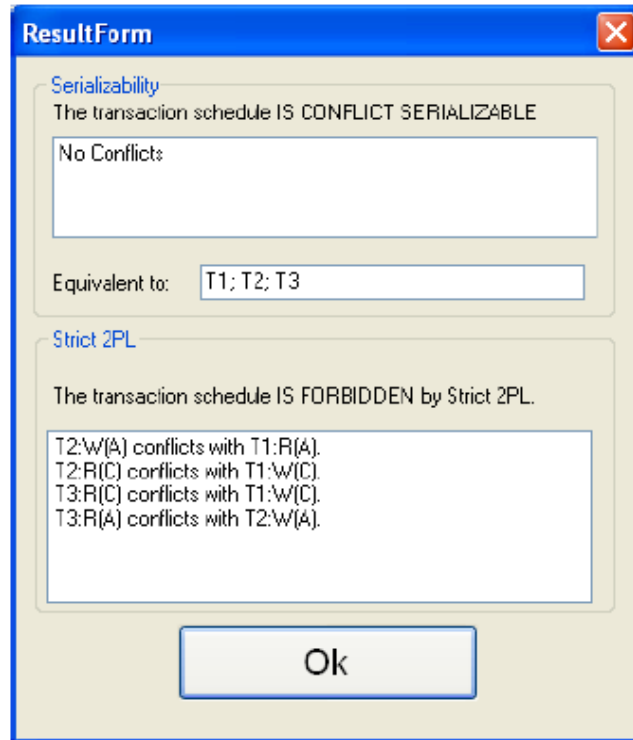
שורה ראשונה: מספר הטרנזקציות (3). שורה שנייה: רשימה של שמות הטרנזקציות מופרדות ע"י נקודה-פסיק (T1;T2;T3). שורה שלישית: מספר האובייקטים (3). שורה רביעית: שמות האובייקטים, מופרדים ע"י נקודה-פסיק. שורה חמישית ריקה. שורה שישית: מספר האירועים. אחד-עשר השורות הבאות הן האירועים לפי סדר התרחשותם; שם הטרנזקציה, נקודה-פסיק, הפעולה (R, W, Commit, Abort) והאובייקט עליו היא נעשית (רק עבור קריאה או כתיבה).

כל קובץ יכיל תזמון יחיד. התכנה שלכם אמורה לאפשר טיפול במספר קבצים, אחד אחרי השני. דוגמה למסך הקלט ניתנת להלן:



3 פלט

הפלט של התכנה יכול להיות חלון הודעה המציג האם התזמון CS והאם הוא מורשה ע"י Strict 2PL. אם התזמון אינו CS, יש להציג את רשימת ההתנגשויות. אם התזמון כן CS, יש להציג סידור מתאים. אם הסידור לא מתאים ל-Strict 2PL, יש להציג את רשימת ההתנגשויות. לדוגמה, עבור התזמון הנ"ל החלון הבא יוצג:



רמז: דרך אחת לקבוע האם תזמון הוא CS היא לבנות גרף קדימות (Precedence Graph) ולבדוק האם הוא מכיל מעגלים. אם לא (סימן שהתזמון CS), לבצע מיון טופולוגי כדי להראות סידור מתאים.

4 עמידות

התכנה שלכם אמורה לדעת לטפל במקרים הבאים מבלי להתרסק:

- קובץ קלט בפורמט לא תקין
- קובץ קלט עם נתונים או מידע לא נכונים
- קובץ קלט עם סדר לא הגיוני של פקודות

בכל המקרים, התכנה אמורה להראות הודעת תקלה ולבקש מהמשתמש לנסות שנית. **אין לנסות לנחש את כוונת כותב הקובץ, או לנסות לתקנו.**

4.1 אופציית בונוס

בצעו זאת רק לאחר שסיימתם עם החלקים הקודמים! לכל תזמון, קבעו האם הוא VS ומהו סידור מתאים. הוסיפו שורה מתאימה לחלון הפלט והציגו את הסידור המתאים (אם קיים).

הערה: בעיית קביעת VS הינה *NP Complete*. אין זאת אומרת שפתרון הינו בלתי-אפשרי, רק שאין אלגוריתם יעיל המסוגל לבצע זאת בזמן פולינומיאלי. לא קשה לכתוב אלגוריתם לזמן אקספוננציאלי.

5 ציון והגשה

המטלה העיקרית שווה 100 נקודות. התוספת שווה 25 נקודות נוספות. הציון יחולק כך:

1. זיהוי נכון של Strict 2PL וההתנגשויות שלו: 30 נקודות
2. זיהוי נכון של CS וההתנגשויות שלו: 60 נקודות
3. עמידות בפני תקלות: 10 נקודות
4. זיהוי נכון של VS: 25 נקודות תוספת (במידה וכל השאר עובד)

ניתן לעבוד בקבוצות של עד 3 תלמידים.
יש להגיש את העבודה בדוא"ל או בתל"מ עד 11 באפריל 2010, שעה 23:50. שימו לב לחוקי וענשי האיחור הרשומים בסילבוס
יש להגיש את העבודה בקובץ ZIP (לא RAR) יחיד המכיל את:

- קבצי הפרויקט (C# או VB)
 - כל הקבצים הנדרשים להידור הפרויקט
 - קובץ הרצה מהודר
 - קובץ README המכיל את שמות הסטודנטים, מספר השעות שנדרשו לביצוע המטלה וכל ההערות או הדוקומנטציה הנדרשות להבנת ולהפעלת התכנה.
- אני מציע להשתמש בדוא"ל, מאחר ותל"מ עושה בעיות עם קבצי ZIP.