מידול מקביליות באמצעות שזירה

גרא וייס
המחלקה למדעי המחשב
אוניברסיטת בן-גוריון

הרכבה מקבילית כפעולה מתמטית על מערכות מעברים

בהנתן מערכות מעברים $TS_1, TS_2, \dots, TS_n$

המטרה היא להגדיר אופרטור $|||$ כך ש

$$TS = TS_1 ||| TS_2 ||| \dots ||| TS_n$$

היא מערכת מעברים
המתארת את ההפעלה במקביל של המערכות

הגדרה: שזירת מערכות מעברים

עבור שתי מערכות מעברים (שני רכיבים הרצים במקביל):

$$TS_i = \langle S_i, Act_i, \rightarrow_i, I_i, AP_i, L_i \rangle, \qquad i = 1,2$$

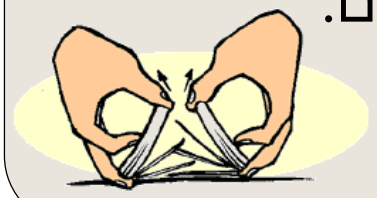
נגדיר מערכת מעברים המתארת הרצה ע"י שזירה:

$$TS_1 \,|||\, TS_2 = \langle S_1 \times S_2, Act_1 \cup Act_2, \rightarrow, I_1 \times I_2, AP_1 \cup AP_2, L \rangle$$

$$\frac{s_1 \xrightarrow{\alpha}_1 s_1'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1', s_2 \rangle} \qquad \frac{s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s'_2 \rangle}$$

מתייגים כל מצב בכל התיוגים שקיבל משני הרכיבים:

$$L(\langle s_1, s_2 \rangle) = L_1(s_1) \cup L_2(s_2)$$



דוגמה

עבור שתי מערכות מעברים TS_1 ו-TS_2 המוגדרות על ידי:

$$TS_i = \boxed{\underbrace{\{s_1, s_2\}}_{S_i}, \underbrace{\{\alpha\}}_{Act_i}, \underbrace{\left\{s_1 \xrightarrow{\alpha} s_2\right\}}_{\rightarrow_i}, \underbrace{\{s_1\}}_{I_i}, \underbrace{\{\langle i,1\rangle, \langle i,2\rangle\}}_{AP_i}, \underbrace{\{s_1 \mapsto \{s_1\}, s_2 \mapsto \{s_2\}\}}_{L_i}}$$

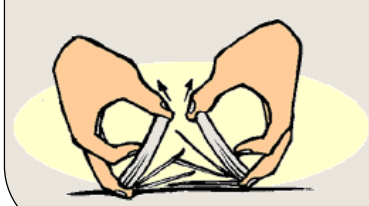
מערכת מעברים המתארת הרצה ע"י שזירה $TS_1 \,|||\, TS_2$ היא:

$$\boxed{\langle \underbrace{\{\langle s_1,s_1\rangle, \langle s_1,s_2\rangle, \langle s_2,s_1\rangle, \langle s_2,s_2\rangle\}}_{S_1 \times S_2}, \underbrace{\{\alpha\}}_{Act_1 \cup Act_2}, \rightarrow, \underbrace{\{\langle s_1,s_1\rangle\}}_{I_1 \times I_2}, \underbrace{\{\langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 2,2\rangle\}}_{AP_1 \cup AP_2},}$$

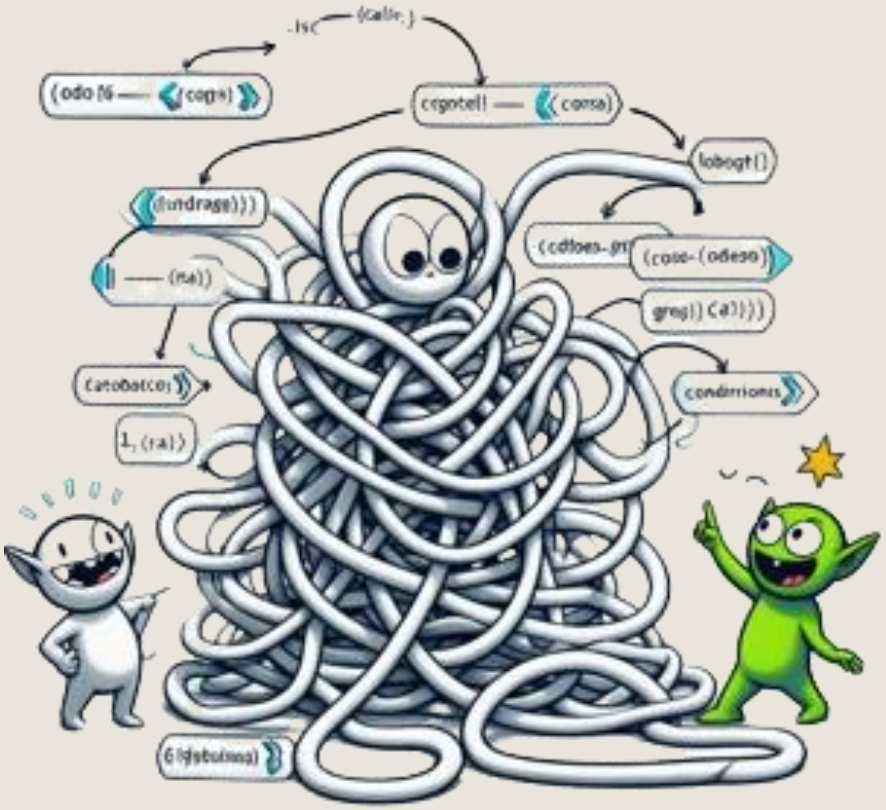
$$\rightarrow = \{\, \langle s_1,s_1\rangle \xrightarrow{\alpha} \langle s_1,s_2\rangle, \langle s_1,s_2\rangle \xrightarrow{\alpha} \langle s_2,s_2\rangle,$$

$$\frac{s_2 \xrightarrow{\alpha}_2 s'_2}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1, s'_2\rangle}$$

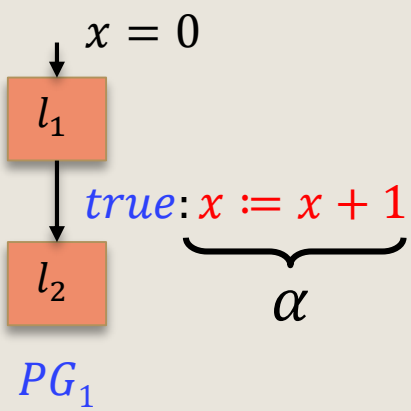
$$\frac{s_1 \xrightarrow{\alpha}_1 s}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s'_1, s_2\rangle}$$

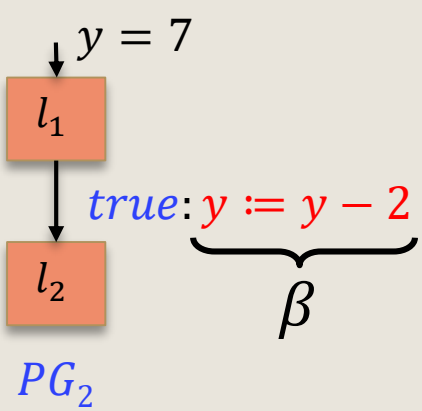


שזירת גרפי תוכנית?

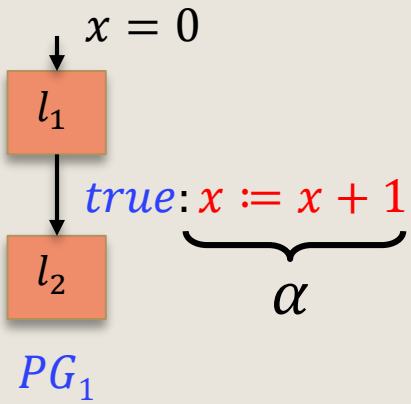


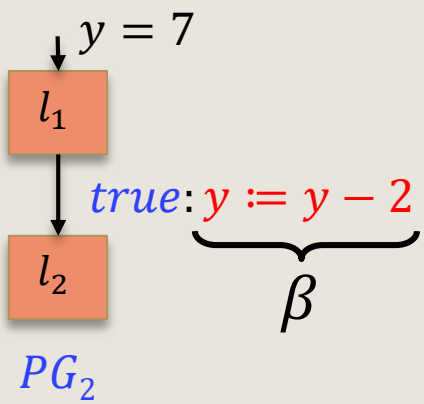
דוגמה: איך נרכיב תוכניות בלי משתנים משותפים?

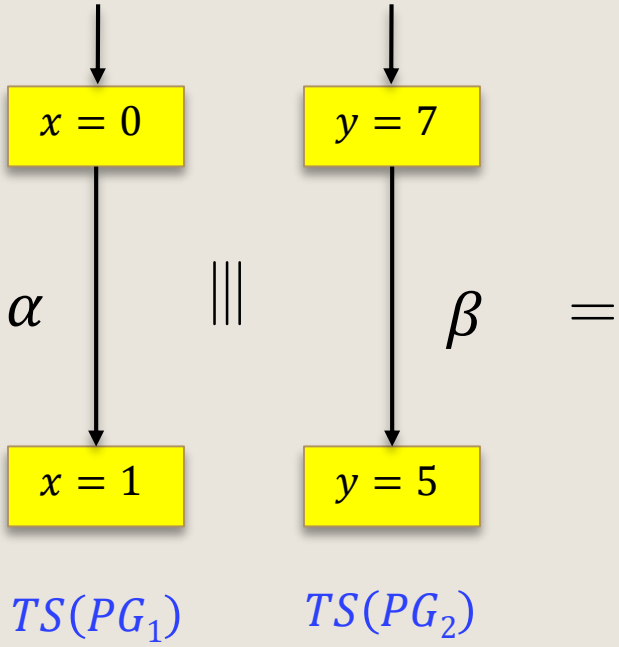


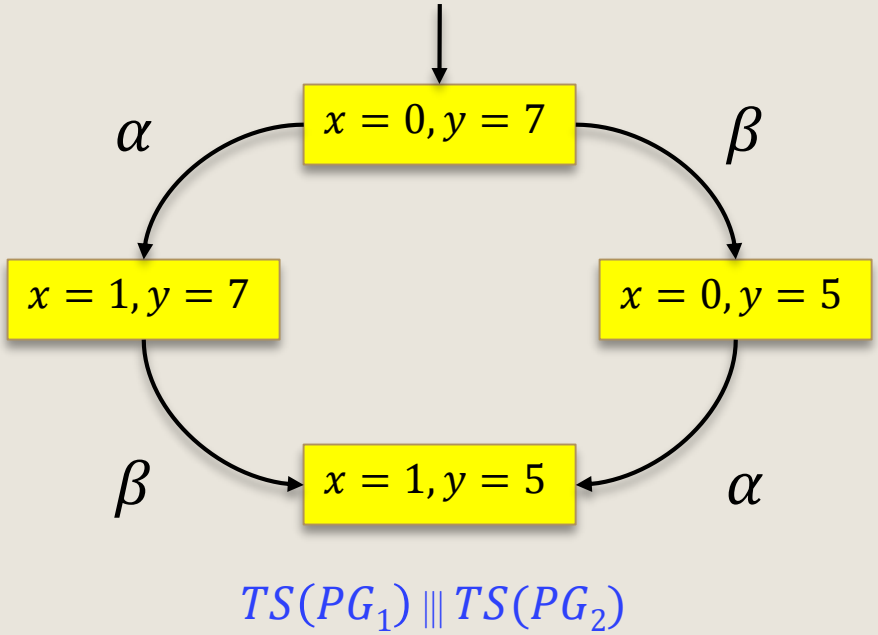


תשובה: שזירת הפריסות









כשאין משתנים משותפים:
שזירת הפריסות = פריסת השזירה

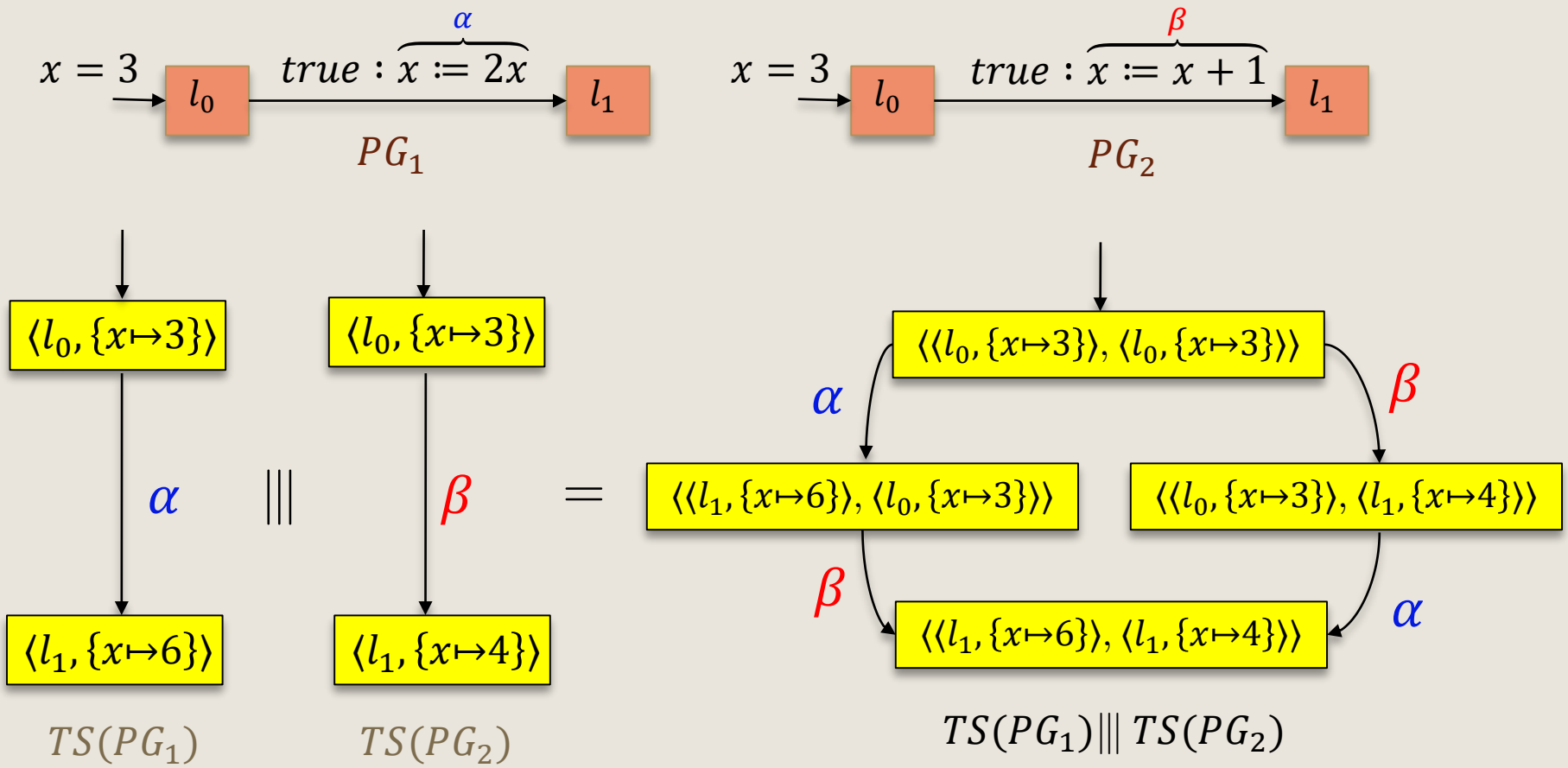
עבור גרפי התוכנית PG_1 (עם משתנים Var_1) ו PG_2 (עם משתנים Var_2)

כשאין משתנים משותפים: $Var_1 \cap Var_2 = \emptyset$

$TS(PG_1) \parallel TS(PG_2)$ מתאר נכון את ההתנהגות המקבילית

מה אם יש משתנים משותפים?

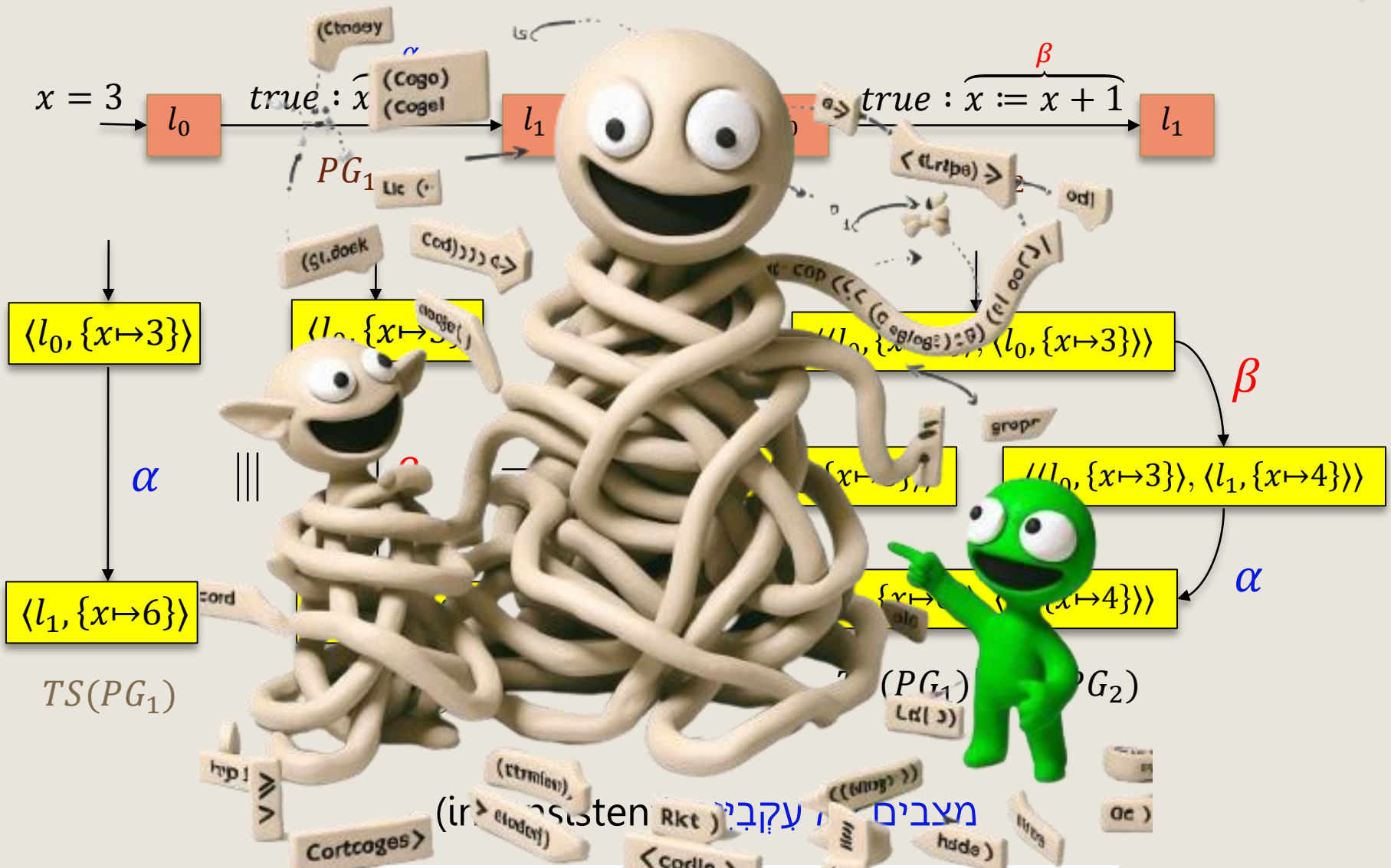
דוגמה בעייתית: פעולות עם משתנים משותפים



מצבים לא עִקְבִיִים (inconsistent)

המודל לא מתאר נכון ביצוע במקביל של α ושל β

דוגמה בעייתית: פעולות עם משתנים משותפים



מצבים עקביים

המודל לא מתאר נכון ביצוע במקביל של α ושל β

שזירת הפריסות או פריסת השזירה ?

אם PG_1 ו PG_2 לא משתפות משתנים: $TS(PG_1) \,|||\, TS(PG_2)$

שזירת מערכות מעברים המתקבלות מפריסת גרפי התוכנית

אם PG_1 ו PG_2 משתפות חלק מהמשתנים: $TS(PG_1 \parallel PG_2)$

שזירת גרפי התוכנית (כפי שנראה בשקפים הבאים) ופריסת התוצאה

- באופן כללי:

$$TS(PG_1) \,|||\, TS(PG_2) \neq TS(PG_1 \parallel PG_2)$$

הגדרה: שזירת גרפי תוכנית

לשני גרפי תוכנית, עבור $i = 1,2$,

$PG_i = \langle Loc_i, Act_i, \mathrm{Effect}_i, \rightarrow_i, Loc_{0,i}, g_{0,i} \rangle$ מעל המשתנים Var_i

נגדיר את גרף התוכנית $PG_1 \parallel PG_2$ מעל $Var_1 \cup Var_2$:

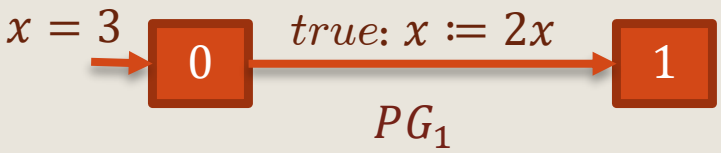
$$\langle Loc_1 \times Loc_2, Act_1 \cup Act_2, \mathrm{Effect}, \rightarrow, Loc_{0,1} \times Loc_{0,2}, g_{0,1} \wedge g_{0,2} \rangle$$

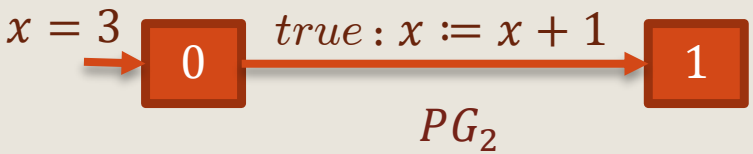
באשר $\rightarrow$ מוגדר באמצעות כללי ההסק:

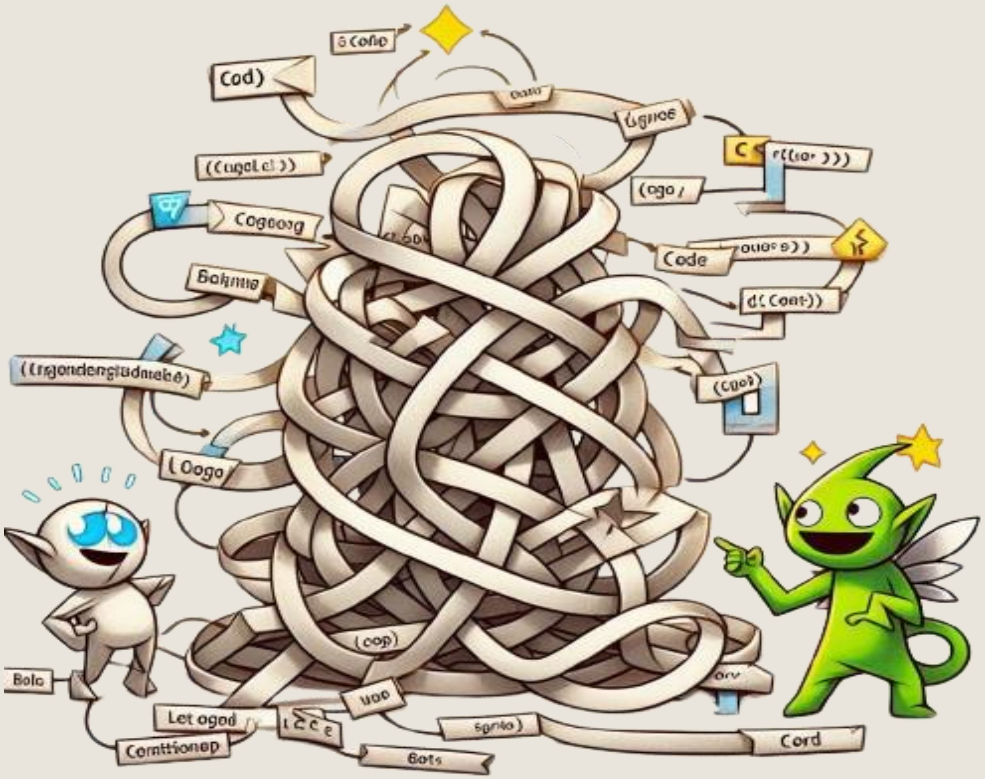
$$\frac{l_1 \xrightarrow{g:\alpha}_1 l_1'}{\langle l_1, l_2 \rangle \xrightarrow{g:\alpha} \langle l_1', l_2 \rangle} \qquad \frac{l_2 \xrightarrow{g:\alpha}_2 l_2'}{\langle l_1, l_2 \rangle \xrightarrow{g:\alpha} \langle l_1, l'_2 \rangle}$$

ו- $\mathrm{Effect}(\alpha,\eta) = \mathrm{Effect}_i(\alpha,\eta)$ אם $\alpha \in Act_i$

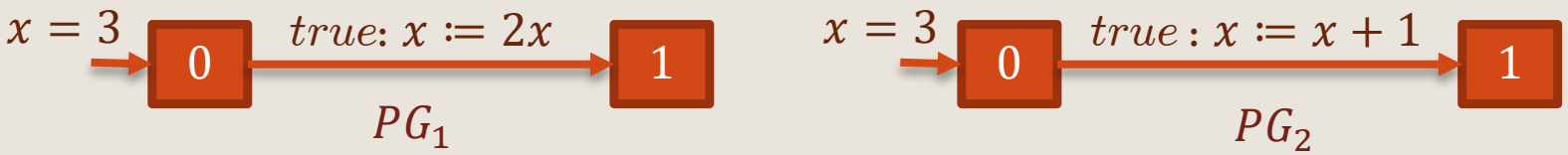
דוגמה





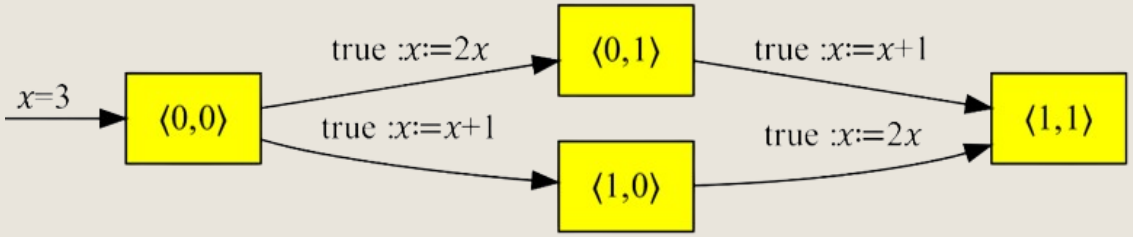


דוגמה



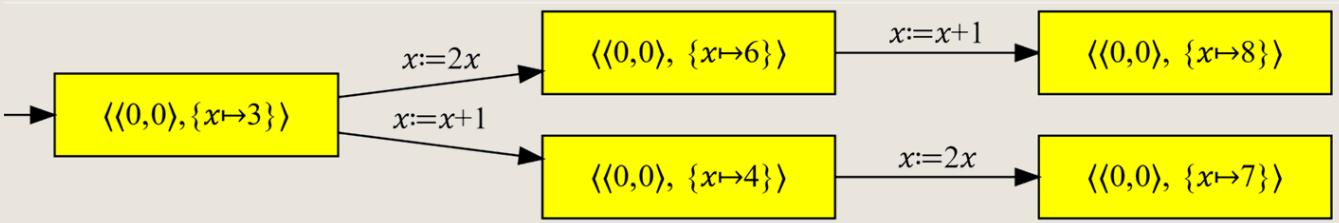
$PG_1 \,||\, PG_2 \;=$

שזירת גרפי תכנית



$TS(PG_1 \,||\, PG_2) \;=$

פריסת השזירה



$$TS(PG_1) \,|||\, TS(PG_2) \neq TS(PG_1 \,||\, PG_2)$$

על אטומיות

לא אטומי

```
x:=x+1;  y:=2x+1  ;  z  :=  y/x  ||  x:=0
```

מקטע ריצה אפשרי:

$$\langle x\mapsto 1, y\mapsto 2, z\mapsto 3\rangle \xrightarrow{x;=x+1} \langle x\mapsto 2, y\mapsto 2, z\mapsto 3\rangle \xrightarrow{y;=2x+1} \langle x\mapsto 2, y\mapsto 5, z\mapsto 3\rangle \xrightarrow{x:=0} \langle x\mapsto 0, y\mapsto 5, z\mapsto 3\rangle \xrightarrow{z:=y/x} \text{Error!!!}$$

אטומי

```
⟨x:=x+1;  y:=2x+1;  z  :=  y/x⟩  ||  x:=0
```

אחד התהליכים (הימני או השמאלי) מסתיים קודם:

$$\langle x\mapsto 1, y\mapsto 2, z\mapsto 3\rangle \xrightarrow{x;=x+1;y:=2x+1;z=y/x} \langle x\mapsto 2, y\mapsto 5, z\mapsto 2.5\rangle \xrightarrow{x:=0} \langle x\mapsto 0, y\mapsto 5, z\mapsto 2.5\rangle$$

אלגוריתם של פטרסון (Peterson) למניעה הדדית (mutual exclusion)

Gary L. Peterson

P_i: loop forever {

(* non-critical section *)

$\langle b_i := \ true;\ x := 1 - i \rangle;$

wait until $(x = i \vee \neg b_{1-i})$

(* critical section *)

$b_i := false;$

}

המשתנה b_i מקבל ערך אמת אם ורק אם התהליך P_i ממתין על הקטע הקריטי. אם שני התהליכים ממתינים, המשתנה x מכריע מי ייכנס.

דוגמה: הפרוטוקול של פטרסון

תהליך ימני

```
        while (true) {
nc:         …
            ⟨b1, x⟩ := ⟨true, 0⟩
wt:         wait until (x = 1 ∨ ¬b0);
cs:         …
            b1 := false;
        }
```

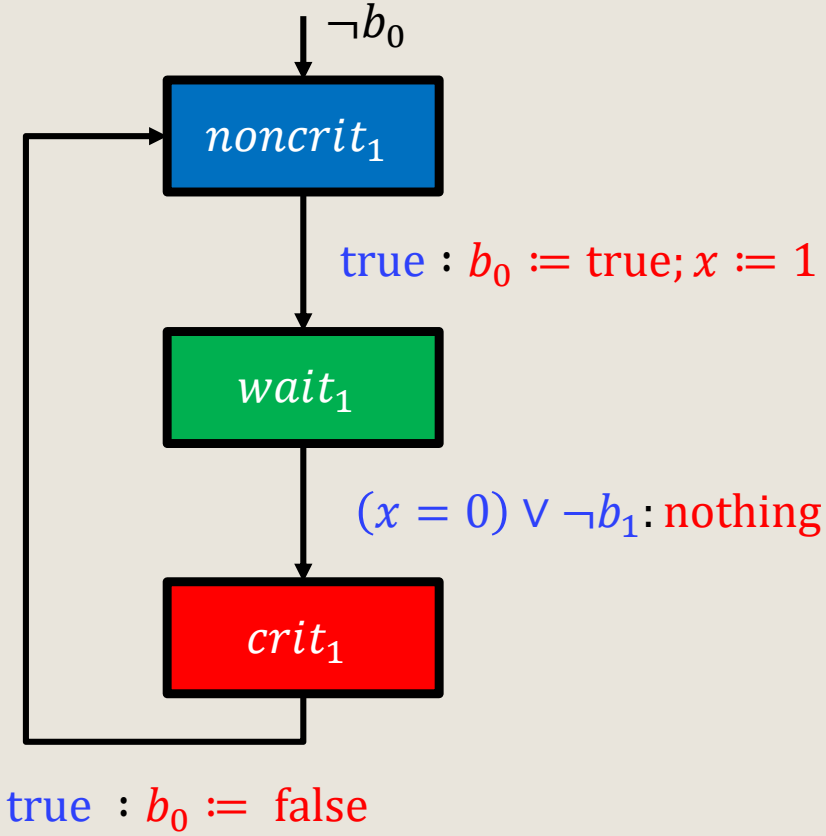
תהליך שמאלי

```
        while (true) {
nc:         …
            ⟨b0, x⟩ := ⟨true, 1⟩
wt:         wait until (x = 0 ∨ ¬b1);
cs:         …
            b0 := false;
        }
```

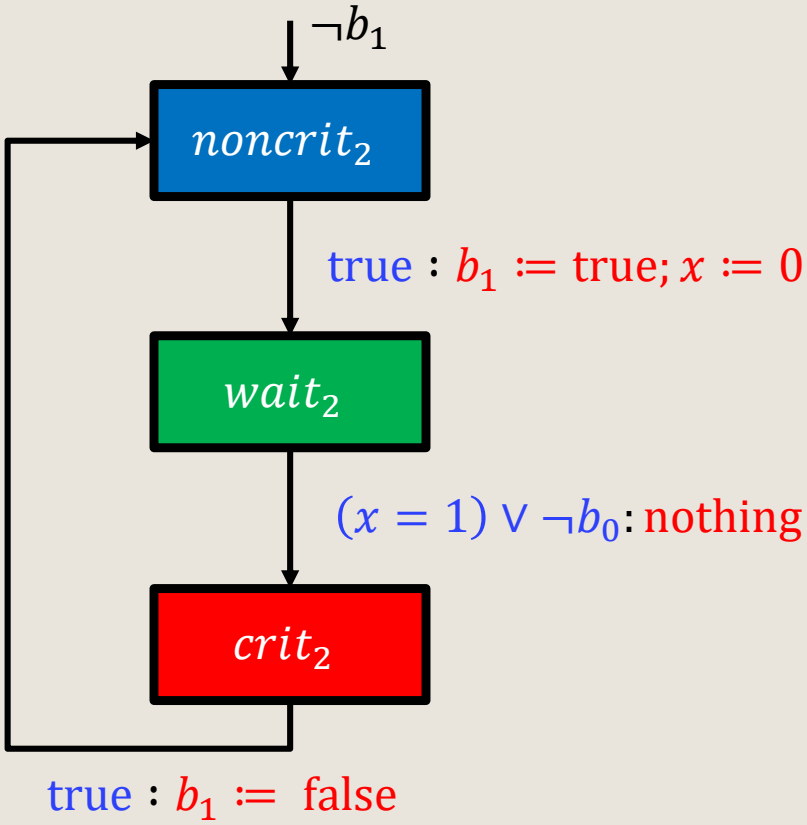
האם אנחנו יכולים להבטיח ששני התהליכים לא ייכנסו יחד לקטע הקריטי?

גרפי תוכנית עבור הפרוטוקול של פטרסון

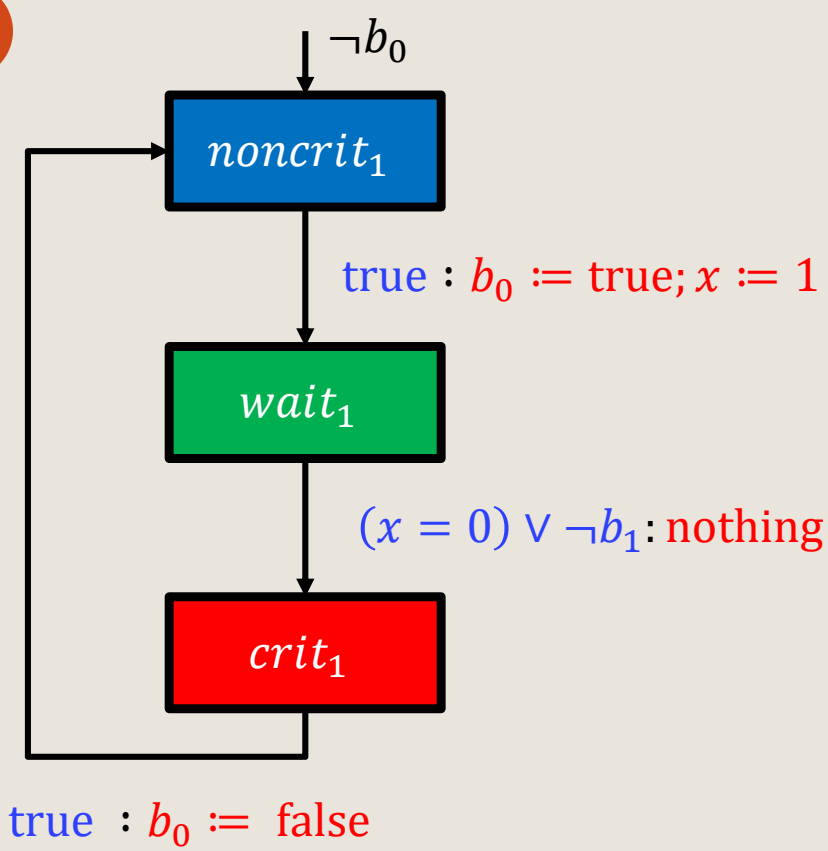
דוגמה לתוכניות שהתקשורת ביניהן היא באמצעות משתנים משותפים



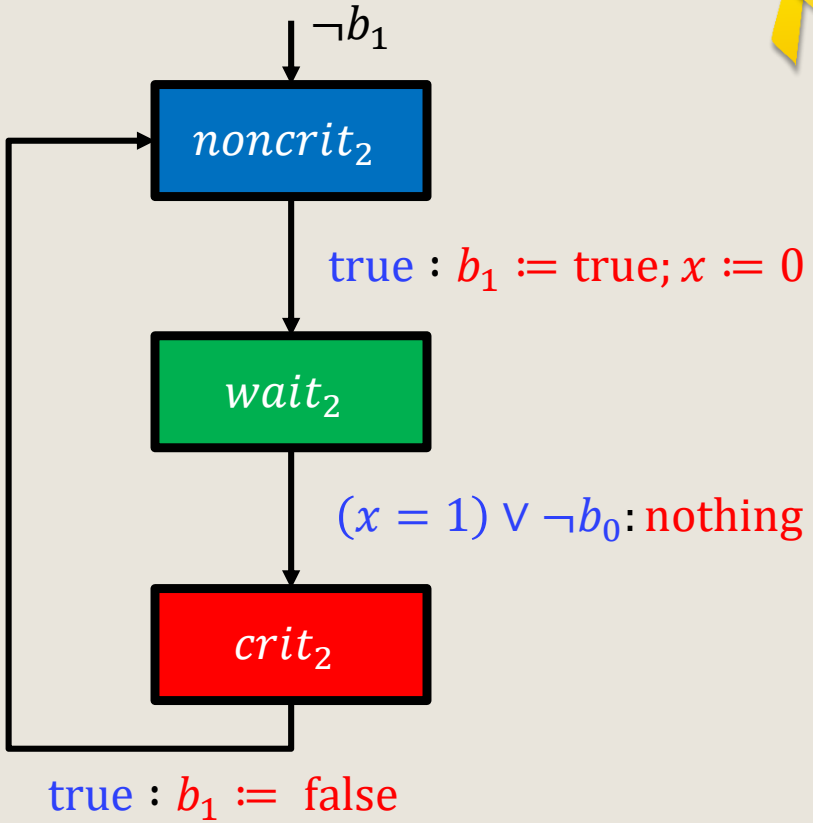
PG_0



PG_1



PG_0

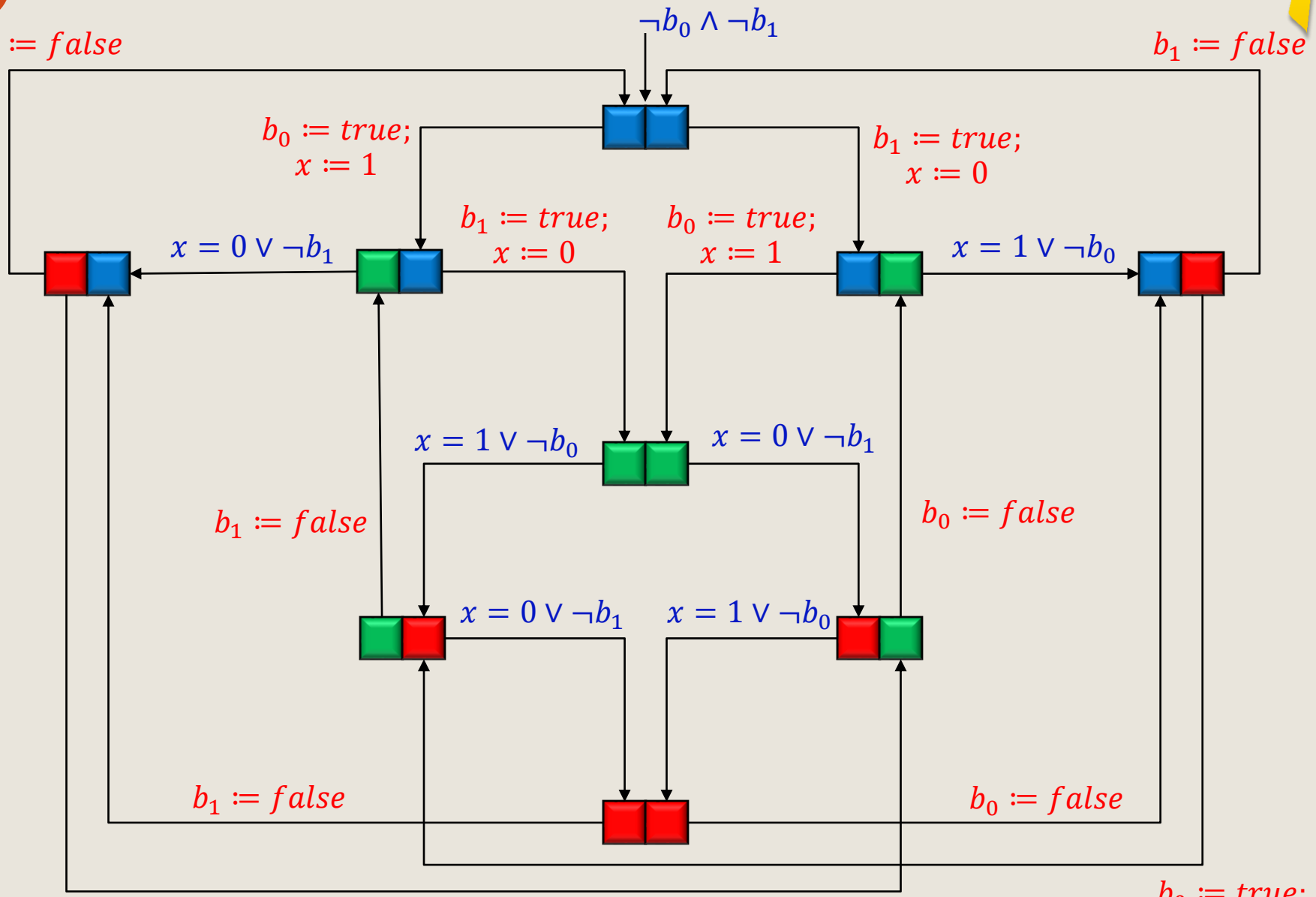


PG_1

$$\frac{l_1 \xrightarrow{g:\alpha}_1 l_1'}{\langle l_1, l_2\rangle \xrightarrow{g:\alpha} \langle l_1', l_2\rangle}$$

$$\frac{l_2 \xrightarrow{g:\alpha}_2 l_2'}{\langle l_1, l_2\rangle \xrightarrow{g:\alpha} \langle l_1, l'_2\rangle}$$

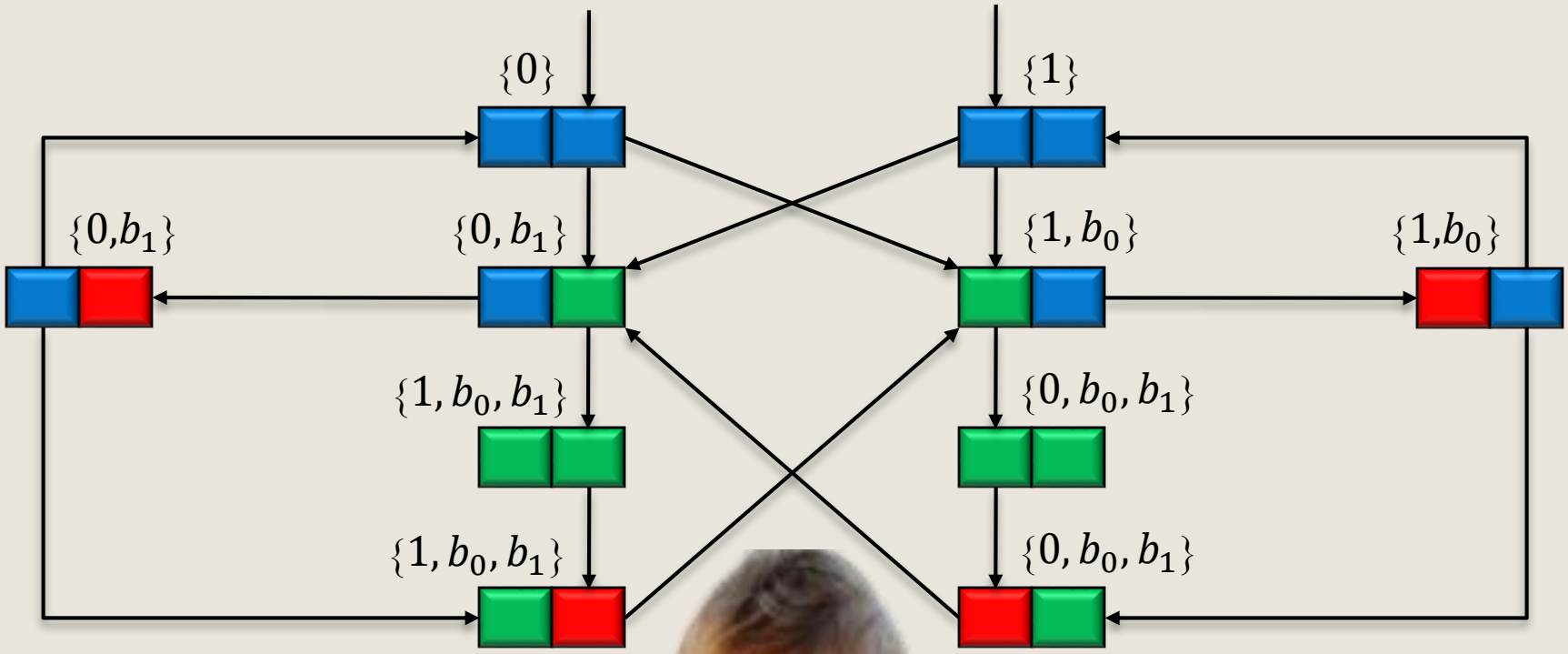
שזירת גרפי התוכנית



$PG_0 \parallel PG_1$

פריסת השזירה של גרפי התוכנית

$$TS_{Pet} = TS(PG_0 \parallel PG_1)$$



בדיקה ידנית אם שניים יכול ביחד לקטע הקריטי : לא

דוגמה: הפרוטוקול של פטרסון
בלי השמות אטומיות

התהליך הימני:

```
        while (true) {
nc:         …
            x := 0
rq:         b1 := true
wt:         wait until (x = 1 ∨ ¬b0);
cs:         …
            b1 := false;
        }
```

התהליך השמאלי:

```
        while (true) {
nc:         …
            x := 1
rq:         b0 := true
wt:         wait until (x = 0 ∨ ¬b1);
cs:         …
            b0 := false;
        }
```

האם אנחנו יכולים להבטיח ששני התהליכים לא ייכנסו יחד לקטע הקריטי?

הגרסה בלי האטומיות לא עובדת
(הייתה עובדת אם היינו כותבים בסדר הפוך)

$$\langle \mathrm{nc}_0,\ \mathrm{nc}_1,\ x{=}0,\ b_0{=}\mathrm{false},\ b_1{=}\mathrm{false}\rangle$$
$$\langle \mathrm{nc}_0,\ \mathrm{rq}_1,\ \boldsymbol{x{=}0},\ b_0{=}\mathrm{false},\ b_1{=}\mathrm{false}\rangle$$
$$\langle \mathrm{rq}_0,\ \mathrm{rq}_1,\ \boldsymbol{x{=}1},\ b_0{=}\mathrm{false},\ b_1{=}\mathrm{false}\rangle$$
$$\langle \mathrm{wt}_0,\ \mathrm{rq}_1,\ x{=}1,\ \boldsymbol{b_0{=}\mathrm{true}},\ b_1{=}\mathrm{false}\rangle$$
$$\langle \mathrm{cs}_0,\ \mathrm{rq}_1,\ x{=}1,\ b_0{=}\mathrm{true},\ b_1{=}\mathrm{false}\rangle$$
$$\langle \mathrm{cs}_0,\ \mathrm{wt}_1,\ x{=}1,\ b_0{=}\mathrm{true},\ \boldsymbol{b_1{=}\mathrm{true}}\rangle$$
$$\langle \mathrm{cs}_0,\ \mathrm{cs}_1,\ x{=}1,\ b_0{=}\mathrm{true},\ b_1{=}\mathrm{true}\rangle$$

הפרת תכונת המניעה ההדדית!