(1) 22.10.09

אימות פורמלי

אימות מערכות תוכנה וחומרה

המרצה: אורנה קופרמן   רוס 219

שעת קבלה - בשיחת מייל

יש לינק לאתר מהאתר של אורנה. cs.huji.ac.il/~ornak

יהיו 4 תרגילים, כל אחד מהם 10 נקודות (חצי מגן - אם ציון נמוך יהיו רק 5 נק')

מבחן - 60 נק'.

נניח שיש בניין רב קומות עם 3 מעליות. זו מערכת שמגיבה לסביבה שלה. למשל יש קונפיגורציה מאוד ברורה. לכל מעלית יש את הקומה שבה היא נמצאת וכנראה גם לכל אחת יש רשימה של הכפתורים שנלחצו.

עכשיו רוצים לתכנן את האלגוריתם של המעליות. יש דרישות הגיוניות כמו למשל שאף אחד לא יחכה לנצח לקומה שלו. אבל אולי גם רוצים למזער את זמן ההמתנה. יש המון שיקולים שיכולים להניע אותנו. ואת המערכת הזאת כמובן נרצה לאמת. יש פה מספר מצבים סופי (מאוד גדול) אז אם לא אכפת לנו מזיכרון אפשר לעשות גרף מאוד גדול שמתאר את הפרוטוקול שלנו ולבדוק אותו.

ב- intro כשרצינו לבדוק שתוכנית נכונה, הרצנו קלטים לדוגמא. בדיקות זה אכן חשוב וגם שם יש הרבה מחקר (איך לייצר קלטים טובים לבדיקה וכו') אבל זה לא מה שנתמקד בו. אנחנו נדבר על אימות פורמלי - formal verification. כאן המטרה היא להבטיח איכות/נכונות ב-100%. נתונים מפרט ψ ומערכת S ומטרתנו לוודא ש-S מקיימת את ψ.

עד שנות ה-80 עבדו בשיטת proof-based.

דוגמה: נתבונן בתכנית הבאה:

```
P:  read(x,y)
    ---------
    z := 0
    while y>0 do
        z := z+x
        y := y-1
    ---------
    return(z)
```

התכנית הנ"ל אם $y \in N$, $y>0$ בהכרח אז מובטח שבסוף $z=xy$.

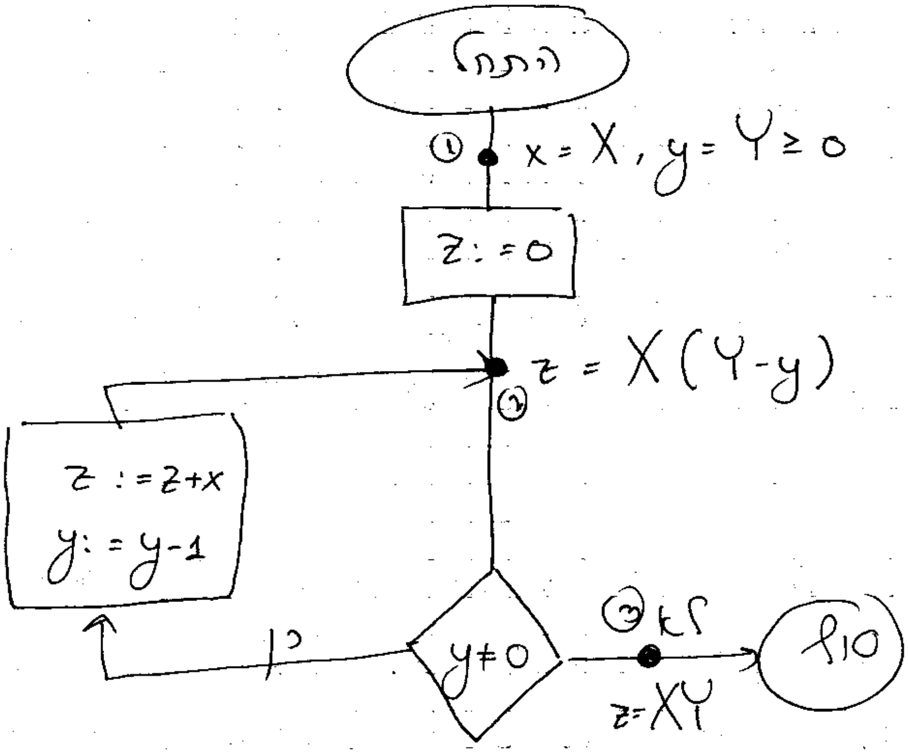
פורמלית כותבים את זה ככה:

$$< x=X, y=Y \ge 0 > P < z=XY >$$

בהוכחת נכונות חלקית מוכיחים שאם התכנית תעצור אז זה מתקיים.

בהוכחה מלאה צריך גם להוכיח שהתכנית תמיד עוצרת.

לתכנית הנ"ל אפשר לרשום תרשים זרימה:



השיטה של פלויד (1960) להוכחת תכניות:

- נבחר קבוצה C של נקודות חתך (של היציאה מ"התחל", של הכניסה ל"סוף" ולפחות נקודת חתך אחת בכל מעגל פשוט)
- נקשר את נקודות החתך בשמורות - טענות שתמיד נכונות בנקודות החתך האלה.
- נבדוק גרירה בין נקודות סמוכות - כלומר אם עוברים מ-a ל-b אז יש גרירה של התנאים בין נקודות עוקבות.

למשל, אם אנחנו בנקודה (2) ו- $y \neq 0$ אנחנו שוב מגיעים ל-(2) וצריך להראות שהשמורה עדיין נכונה:

$$z = X(Y-y) \wedge y \neq 0 \wedge z' = z+x \wedge y' = y-1$$
$$\rightarrow z' = X(Y-y')$$

את זה אפשר לבדוק בצורה אוטומטית באמצעות therorem prover

פלויד הוכיח שאם עושים את התהליך הזה והוא הצליח סימן שהתכנית נכונה. זה נחמד אבל יש פה כמה חסרונות:

- מציאת השמורות יקרות (הן לא נמצאות באופן אוטומטי)
- יחסי קלט-פלט של תכניות מורכבות

בשנות ה-80 התחילו לפתח שיטות שאין להן החסרונות האלה ושלהן אנחנו נלמד. - State Exploration Methods

הרעיון הוא מאוד פשוט.

1. מערכת ← "מודל מתמטי" (אנחנו בד"כ נדבר על גרפים)
2. תכונה שרוצים לוודא ← מפרט פורמלי
3. אלגוריתם שבודק האם המודל המתמטי מקיים את המפרט הפורמלי.

(זה נקרא model checking)

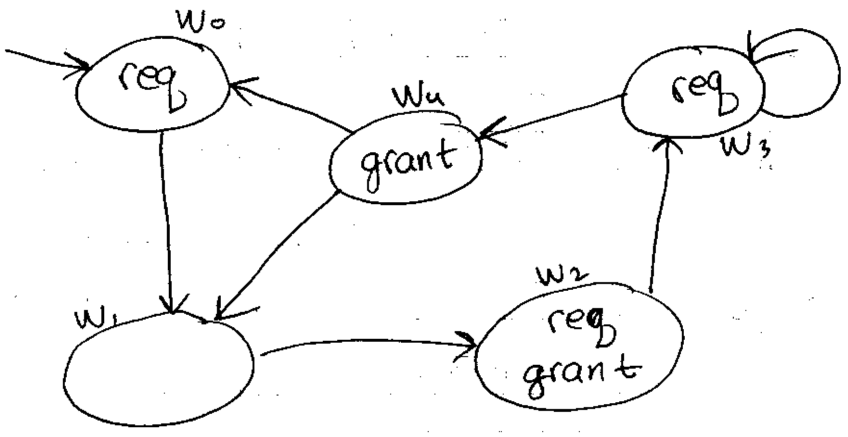
הגדרה: מבנה קריפקה (Kripke) הוא חמישייה

$$K = \langle AP, W, W_0, R, L \rangle$$

כאשר

- AP : אטומים (הא"ב)
- W : מצבים
- $W_0 \subseteq W$: מצבים התחלתיים
- $R \subseteq W \times W$: יחס מעברים טוטאלי (לכל w קיים לפחות w' אחד כך ש- $R(w,w')$)
- $L : W \rightarrow 2^{AP}$ - פונקציית סימון

דוגמה: יש מערכת שאפשר להגיש לה בקשה ולקבל אישור.



$W = \{w_0, \dots, w_4\}$

$W_0 = \{w_0\}$

$R(w_4, w_1)$ ...

$L(w_4) = \{grant\}$, $L(w_1) = \emptyset$, ...

עם המערכת הזאת אפשר לשאול שאלות. למשל, האם תמיד (req $\rightarrow$ בסופו של דבר יש grant)?

לא! כי ב- w_3 יש לולאה שנשארת רק ב- req.

תשובה: שים לב שמרחב המצבים יכול להיות מעריכי בגודל הייצוג (אם יש n משתנים בוליאניים יש 2^n מצבים אפשריים). לכן נרצה למצוא אלגוריתמים שיצליחו לעבוד על ייצוגים קומפקטיים יותר.

לוגיקה עתית (1977) Temporal Logic

- בלוגיקה יש אטומים $p, q \in AP$ שיכולים להיות נכונים או לא.
- פעולות בוליאניות: $\neg, \vee, \wedge, \ldots$

בינתיים זה נותן לנו propositional logic, אבל אנחנו רוצים יכולת לעבוד עם גרפים ולכן יש גם

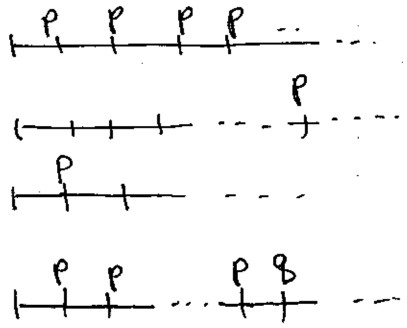
- אופרטורים עתיים:

p נכון בכל המצבים — always (Globally) : Gp

p נכון בסופו של דבר — eventually (future) : Fp

p נכון במצב הבא — next : Xp

p נכון עד ש-q נכון — until : pUq



את האופרטורים האלה אפשר לשלב. למשל GFp (always eventually p) זה אומר שכל פעם שמגיעים ל-p וממשיכים הלאה שוב נגיע מתישהו ל-p. כלומר p מופיע אינסוף פעמים (לא בהכרח ברצף). ברור שאם Gp אז GFp.
אם רוצים רק מספר סופי של p אפשר כמובן $\neg GFp$ אבל זה שקול ל-$FG\neg p$ - בסופו של דבר יש רק $\neg p$.

LTL: linear temporal logic (למצב יש רק עוקב יחיד)

- סינטקס: ביחס ל-AP
  - $p \in AP$ היא נוסחה ב-LTL
  - אם φ_1 ו-φ_2 הן נוסחאות ב-LTL אז גם $\neg\varphi_1$, $\varphi_1 \vee \varphi_2$, $X\varphi_1$, $\varphi_1 U \varphi_2$ נוסחאות ב-LTL

(מאלה אפשר לקבל גם את $F\varphi_1$, $G\varphi_1$, $\varphi_1 \to \varphi_2$, כלומר $\{\neg, \vee, X, U\}$ מע' קשרים שלמה).

• סמנטיקה - ביחס למילים ב- $(2^{AP})^\omega$ (ω זה כמו * רק אינסופי)

כלומר ביחס למילים $\sigma_1 \to \sigma_2 \to \sigma_3 \to \dots$ $\quad \sigma_i \subseteq AP$

סימונים: $\pi = \sigma_1\sigma_2\sigma_3\dots$

$\pi_i = \sigma_i$ האות ה-i של π

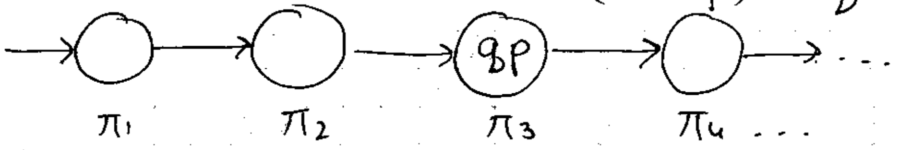
$\pi^i = \sigma_i\sigma_{i+1}\sigma_{i+2}\dots$ הסיפא של π שמתחילה במקום ה-i

עכשיו אפשר להגדיר את הסמנטיקה באינדוקציה על מבנה הנוסחה:

- עבור אטומים: $\pi \models p \leftrightarrow p \in \pi_1$
- $\pi \models \neg\varphi_1 \leftrightarrow \pi \not\models \varphi_1$ (φ_1 ← נוסחה יותר קטנה מ- $\neg\varphi_1$)
- $\pi \models \varphi_1 \vee \varphi_2 \leftrightarrow \pi \models \varphi_1 \vee \pi \models \varphi_2$
- $\pi \models X\varphi_1 \leftrightarrow \pi^2 \models \varphi_1$
- $\pi \models \varphi_1 U \varphi_2 \leftrightarrow \exists k \geq 1 (\pi^k \models \varphi_2 \wedge \forall 1 \leq j < k \ \pi^j \models \varphi_1)$

(באתר יש דף שמסכם את הסינטקס והסמנטיקה)

דוגמה: $(XXp)Uq$



זה לא מספק את $(XXp)Uq$ כי π^2 לא מספק את XXp.

אם נוסיף p ל- π_4 אז זה כן יספק.

דוגמה: נגדיר את הקשרים שנותרו באמצעות הקשרים הקיימים

$Fp \equiv true \ U \ p$ (true מספק בכל דבר)

$Gp \equiv \neg F \neg p$

(weak until) $pWq \equiv pUq \vee Gp$ או שתמיד p או שנפסק ויש q אחרי ה-p.

הלוגיקה הזמנית מאפשרת לנו להביע הרבה תכונות של מע'
שעניינו חיים. למשל, חוסר הרעבה במע' הפעלה:

$$G(T_1 \to F\, CS_1)$$

תמיד - אם מישהו מבקש, אז בסופו של דבר הוא "ייכנס
לקטע הקוד הקריטי".

הלוגיקה CTL^* Computation Tree Logic

כאן הסמנטיקה תהיה על גרפים ולא רק על מסלולים שלהם.

- סינטקס: ביחס ל- AP

נוסחאות מצב:

- כל $p \in AP$ הוא נוסחת מצב (אטום אכן מתאר מצב ולא מסלול)
- אם φ_1, φ_2 נוסחאות מצב אז גם $\neg\varphi_1$, $\varphi_1 \vee \varphi_2$ נוסחאות מצב
- אם ψ נוסחת מסלול (מיד נגדיר) אז $A\psi$ (זה כמו $\forall\psi$ רק של מסלולים) היא נוסחת מצב.

נוסחאות מסלול:

- כל נוסחת מצב היא גם נוסחת מסלול
- אם ψ_1, ψ_2 נוסחאות מסלול אז גם $\neg\psi_1$, $\psi_1 \vee \psi_2$, $X\psi_1$, $\psi_1 U \psi_2$ נוסחאות מסלול.

דוגמה:

$$AG(p \to EFq)$$

E – כמו $\exists$ של מסלולים

גרף מקיים את הנוסחה אם לכל המצבים אם מצב
מקיים p אז יש אפשרות להתפתח למסלול שיש בו בסוף q.

(5) 29.10.09

אימות פורמלי

בשיעור הקודם ראינו LTL והגדרנו את הסינטקס של CTL*.

תזכורת:

נכתוב את הסינטקס של LTL בדקדוק חסר הקשר:

$AP \rightarrow p \mid q \mid r \mid \dots$ // אטומים

$\psi \rightarrow AP \mid \neg\psi \mid \psi \vee \psi \mid X\psi \mid \psi U \psi$

__דוגמא:__ האם $\psi_1 = pU(Xq) \equiv X(pUq) = \psi_2$? לא!

נסתכל במסלול (p q) אז זה $\models \psi_2$ אבל $\not\models \psi_1$.

ואם נסתכל על (p p q) אז זה $\not\models \psi_2$ אבל $\models \psi_1$.

אז אין פה גרירה בשום כיוון.

לפעמים רוצים לשאול אם נוסחת LTL מתקיימת בגרף.
אז הכוונה היא אם כל המסלולים בגרף מקיימים את הנוסחה.

נכתוב את הסינטקס של CTL*:

$\varphi \rightarrow AP \mid \neg\varphi \mid \varphi \vee \varphi \mid A\psi$ // נוסחת מצב

$\psi \rightarrow \varphi \mid \neg\psi \mid \psi \vee \psi \mid X\psi \mid \psi U \psi$

הסמנטיקה היא ביחס למבנה קריפקה - $K = \langle AP, W, W_0, R, L \rangle$

מסלול $\pi = w_1 w_2 w_3 \dots$ ב-K הוא סדרה אינסופית של
מצבים מ-W כך ש- $R(w_i, w_{i+1})$ לכל $1 \le i$
(לא בהכרח $w_1 \in W_0$) ($w_i \in 2^{AP}$).

סמנטיקה של נוסחאות מצב:

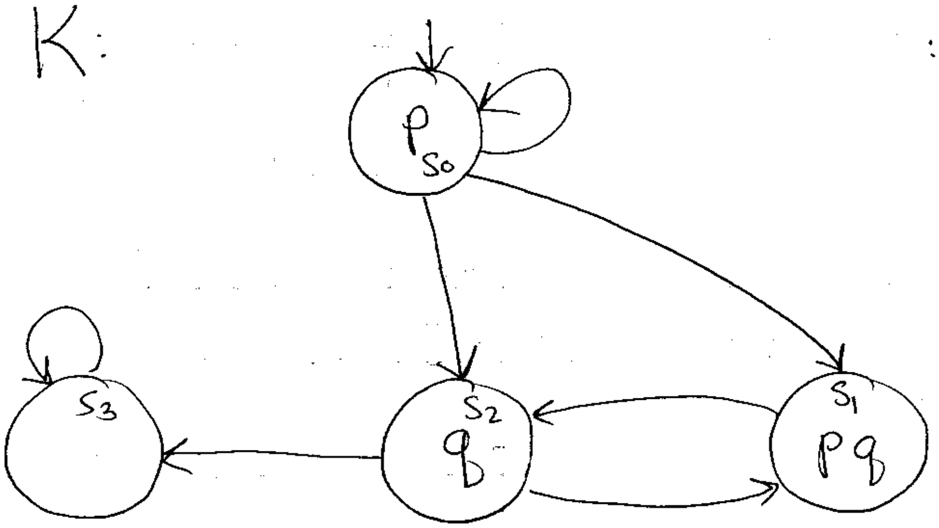
- עבור $p \in AP$, $w \models p \iff p \in L(w)$
- $w \models \neg\varphi \iff w \not\models \varphi$
- $w \models \varphi_1 \vee \varphi_2 \iff w \models \varphi_1 \vee w \models \varphi_2$
- $w \models A\psi \iff$ לכל מסלול $\pi = w_1 w_2 \dots$ כך ש- $w_1 = w$ מתקיים $\pi \models \psi$

סמנטיקה של נוסחאות מסלול

- עבור נוסחת מצב φ: $\pi \models \varphi \iff w_1 \models \varphi$ ($\pi = w_1 w_2 \ldots$)
  כלומר אם φ מסתפקת במצב הראשון של π.
- $\pi \models \neg\psi \iff \pi \not\models \psi$
- $\pi \models \psi_1 \vee \psi_2 \iff \pi \models \psi_1 \vee \pi \models \psi_2$
- $\pi \models X\psi \iff \pi^2 \models \psi$
- $\pi \models \psi_1 U \psi_2 \iff$ קיים $1 \le k$ כך ש- $\pi^k \models \psi_2$ ולכל $1 \le i < k$ $\pi^i \models \psi_1$

דוגמה:

K:



- $ApUq$ - האם הנוסחה מתקיימת? אנחנו תמיד שואלים שאלות על נוסחאות מצב. ובמבנה קריפקה הכוונה היא ש- $K \models \varphi$ אמ"מ לכל $w \in W_0$, $w \models \varphi$.
  אז האם $K \models ApUq$? לא! נוסחת המצב אומרת שבכל מסלול יש p עד q. וזה לא מתקיים כאן רק במסלול $s_0 s_0 s_0 \ldots$ זה לא מתקיים.
- ומה לגבי $ApWq$? נזכור ש- $\psi_1 W \psi_2 = \psi_1 U \psi_2 \vee G\psi_1$
  אז כאן דווקא $K \models ApWq$, אפשר לעבור על כל האפשרויות ולוודא שזה נכון.
- האם $K \models A(GFp \vee FG\neg p)$? זאת טאוטולוגיה!
  או שבכל המסלולים יש ∞ p או שיש מספר סופי של p.
  זה כמובן תמיד נכון בכל מודל.

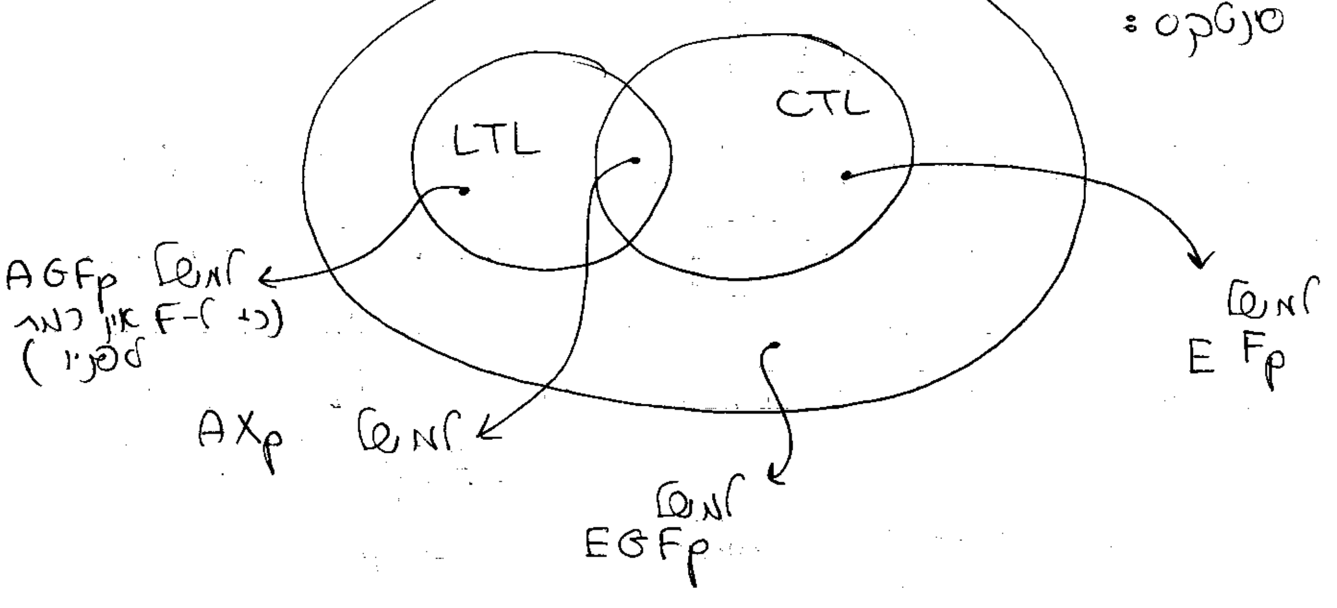
- האם $K \models AFGq$? שומר האם בסופו של דבר תמיד q ? לא! רואים ב- ...$s_0 s_0 s_0$ אין q.
- האם $K \models EFGq$? כאן אנחנו שואלים אם יש לפחות מסלול אחד שבו תמיד יש q בסופו של דבר. אך יש מסלול כזה – $s_0 s_1 s_2 s_1 s_2 \ldots$.
- האם $K \models AFEXEGq$? נבדוק מי מספק את- EGq ? את המצבים s_1, s_2. אז אפשר לסמן את- EGq כאטום חדש r ולהוסיף אותו ל- $L(s_1)$ ול- $L(s_2)$ ואז להמשיך לבדוק את $AFEXr$. רק אפשר להמשיך באופן דומה וכאן ודא שזה אכן מסתפק.

CTL: כל אופרטור זמני יש כמת מסלול לפניו. זה כמו בדוגמה האחרונה שלנו.

$$\varphi \to AP \mid \neg\varphi \mid \varphi \vee \varphi \mid AX\varphi \mid A\varphi U \varphi \mid EX\varphi \mid E\varphi U \varphi$$

זה תת מחלקה של CTL* אך הרי למעשה הגדרנו את הסמנטיקה של CTL.
היתרון של CTL הוא שבדיקה אם נוסחה מתקיימת במבנה היא לינארית באורך הנוסחה. וזה נחמד.

סיכום ביניים:
סינטקס:



CTL*:

$\varphi \rightarrow AP \mid \neg\varphi \mid \varphi \vee \varphi \mid A\psi$

$\psi \rightarrow \varphi \mid \neg\psi \mid \psi \vee \psi \mid X\psi \mid \psi U \psi$

LTL:

$\varphi \rightarrow A\psi$

$\psi \rightarrow AP \mid \neg\psi \mid \psi \vee \psi \mid X\psi \mid \psi U \psi$

CTL:

$\varphi \rightarrow AP \mid \neg\varphi \mid \varphi \vee \varphi \mid A\psi$

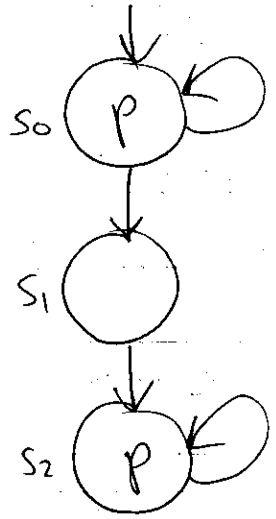
$\psi \rightarrow X\varphi \mid \neg X\varphi \mid \varphi U \varphi \mid \neg \varphi U \varphi$

דוגמה: $\varphi_1 = AFGp \in CTL^*/CTL$

$\varphi_2 = AFAGp \in CTL$

הנוסחאות דומות מאוד אבל הן לא שקולות!

אנחנו נראה מבנה קריפקה K כך ש- $K \models \varphi_1$ אבל $K \not\models \varphi_2$:



די להראות ש- $K \models \varphi_1$ צריך להראות שכל מסלול שמתחיל מההתחלה π מתקיים $\pi \models FGp$ אבל זה ברור כי או שהמסלול תקוע ב- s_0 או שהוא נתקע ב- s_2

מאידך $K \not\models \varphi_2$ כי המסלול $s_0 s_0 s_0 \ldots$ לא מגיע בסופו של דבר ל- AGp.

הצעה: לבחון בבית את השקילות $EGEF\neg p$ ו- $EGF\neg p$.

כח הבעה של לוגיקה עתית

הגדרות:

- עבור שתי נוסחאות φ_1, φ_2 נאמר ש- φ_1 ו- φ_2 שקולות ונסמן $\varphi_1 \equiv \varphi_2$ אם לכל מבנה קריפקה K, $K \models \varphi_1 \Leftrightarrow K \models \varphi_2$.
- עבור שתי שפות ספציפיקציה L_1, L_2 נאמר ש- L_1 חזקה לפחות כמו L_2 ונסמן $L_2 \leq L_1$ אם לכל נוסחה φ_2 ב- L_2 קיימת נוסחה שקולה ב- L_1.

דוגמה: ברור ש- $CTL \leq CTL^*$ ו- $LTL \leq CTL^*$.
חשוב להבין שזה לא קשור לסינטקס: למשל $AXXp$ (כל הנחדים מקיימים את p) היא לא בסינטקס של CTL אבל יש לה נוסחה שקולה שם - $AXAXp$.

I נראה ש- $CTL \nleq LTL$:
עבור מבנה קריפקה (להלן מ"ק) K נגדיר את השפה של K $L(K) \subseteq (2^{AP})^\omega$ באופן הבא:

$L(K) = \{ L(\pi) :$ π מסלול התחלתי של K $\}$

$L(w_1 w_2 \dots) = L(w_1) \cdot L(w_2) \cdot \dots$

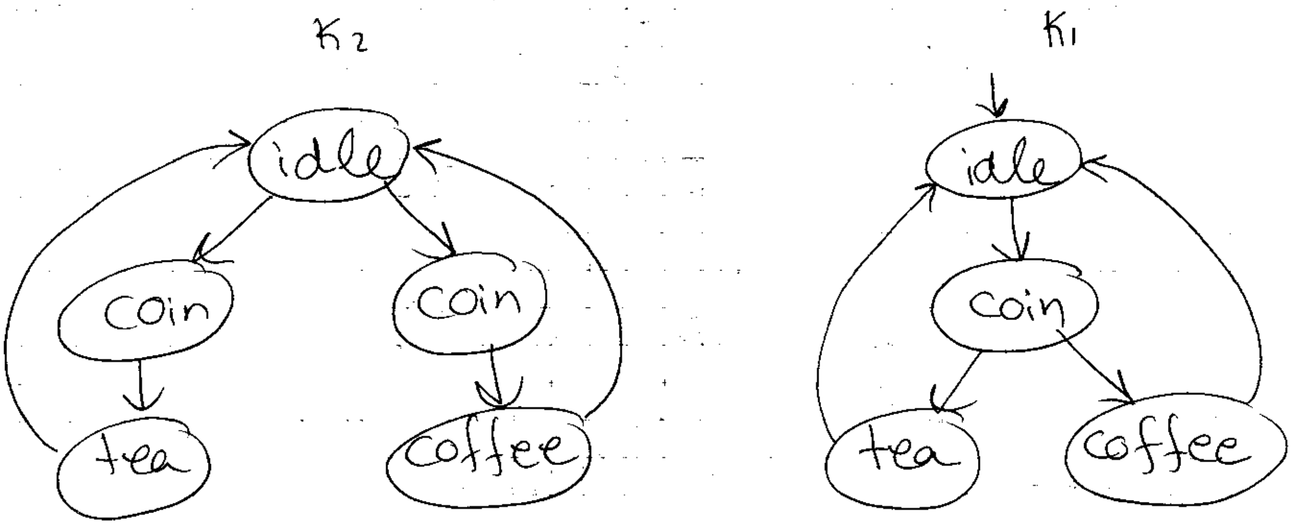
משפט1(קל): לכל שני מ"ק K_1 ו- K_2 אם $L(K_1) \subseteq L(K_2)$ אז לכל נוסחה φ ב- LTL אם $K_2 \models \varphi$ אז $K_1 \models \varphi$
(ל- K_2 יש יותר מסלולים - יותר קשה לו לספק נוסחה).

משפט2(קל): לכל שני מ"ק K_1 ו- K_2 אם $L(K_1) = L(K_2)$ אז לכל נוסחה φ ב- LTL - $K_1 \models \varphi$ אמ"מ $K_2 \models \varphi$.

כדי להראות ש- $CTL \nleq LTL$ נראה שלנוסחה EFp אין נוסחה שקולה ב- LTL.
נתבונן בשני המבנים:

נשים לב ש- $L(K_1)\subseteq L(K_2)$ וכן $K_2\models EFp$, $K_1\not\models EFp$.
אילו הייתה φ ב- LTL שקולה ל- EFp היינו מקבלים
סתירה למשפט 1.

<u>דוגמה</u>: יש שתי מכונות שתייה:



השפה של המכונות זהה:

$$L(K_1)=L(K_2)=(idle, coin, (tea+coffee))^{\omega}$$

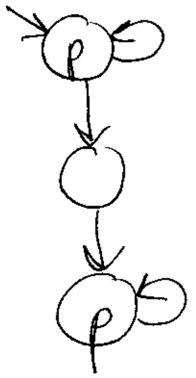
אז לפי משפט 2 אין נוסחה שמבדילה ביניהן ב- LTL.
אבל נסתכל בנוסחה ב- CTL

$$AX(AX\,tea \vee AX\,coffee)$$

לכל הבנים - או שהבנים שלהם הם תה או שהבנים שלהם הם קפה. זו נוסחה שמבדילה ביניהן! או הנוסחה הדואלית - $EX(EX\,tea \wedge EX\,coffee)$
זו מתקיימת רק ב- K_1. קיים בן שיש לו גם בן תה וגם בן קפה.

בלוגיקה מדברים גם על כח הפרדה (distinguishing power).
אז ראינו שכח ההפרדה של CTL גדול מזה של LTL.
אם כח ההפרדה גדול יותר אז כמובן גם כח ההבעה (expressive
power) גדול יותר. אבל הכיוון ההפוך לא נכון.

II נראה ש- $LTL \not\subseteq CTL$. (נראה שלנוסחה $AF(p \wedge Xp)$
אין נוסחה שקולה ב- CTL. הנוסחה אומרת שעל מסלול בסופו
של דבר מגיעים לשני p-ים רצופים.
מאוד מפתה להסתכל על $AF(p \wedge AXp)$ אבל זו לא נוסחה
שקולה. הדוגמה היא כמו קודם:



ההוכחה היא לא פשוטה. היא פורסמה ב- 83 ע"י
Emerson ; Halpern (שעושה כאן שבתון
השנה). נעשה אותה בפעם הבאה.

CTL^* + עבר - יש עוד שני אופרטורים עתיים:
Y - yesterday
S - since
הלוגיקה הזו לא מוסיפה כח הבעה.

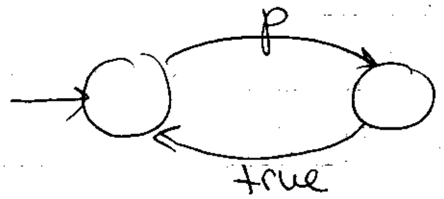
(9) 5/11/09
אימות פורמלי

עדכנו אחרי שראינו ש- LTL ≨ CTL.

ננסה להביע ב- LTL את התכונה $G_2 p$ = p נכון בכל המקומות הזוגיים, כלומר $p \wedge XXp \wedge X^4 p \wedge \ldots$
אבל זו נוסחה באורך אינסופי. אז זה לא טוב לנו.
היו מיני הצעות אבל אף אחת לא הייתה טובה.

<u>משפט</u> (Wolper 1979): אין נוסחת LTL ששקולה ל- $G_2 p$.

<u>הערה</u>: זה משפט ממש מרגיז כי הרי יש אוטומט לשפה הנ"ל:



<u>למה</u>: לכל נוסחת LTL φ, אם יש ב-φ לכל היותר n כאתי X, אז לכל $i > n$, φ לא מבדילה בין $p^i \neg p p^\omega$.

<u>הוכחה</u>: נסמן ב- $\|\varphi\|_j$ את ערך האמת של φ ב- $p^j \neg p p^\omega$. הטענה שאנחנו צריכים להראות הוא שאם $n \geq \#_X \varphi$ אז $\|\varphi\|_{n+1} = \|\varphi\|_{n+2} = \|\varphi\|_{n+3} = \ldots$ כלומר $p^i \neg p p^\omega \models \varphi$ אמ"מ $p^k \neg p p^\omega \models \varphi$ לכל $i,k > n$.

המשפט אכן נובע מהלמה. נניח בשלילה שיש נוסחה ψ ששקולה ל- $G_2 p$. זו נוסחה סופית ולכן יש לה מספר סופי של X-ים. נסמנו ב-n. אז $\|\psi\|_{n+1} = \|\psi\|_{n+2}$ (מהלמה). אבל הרי $\|G_2 p\|_{n+1} \neq \|G_2 p\|_{n+2}$ כי המיקום של ה-$\neg p$ היחיד במסלול זוגי רק באחד מהם.

דוגמה: נניח שיש לנו נוסחה φ שבה $\#_X(\varphi) \leq 1$, כלומר יש לה לכל היותר X אחד. נראה שאי אפשר להפריד את

$pp\neg pp^\omega$, $p^3\neg pp^\omega$, $p^4\neg pp^5 \ldots$

נניח באינדוקציה ~~φ~~ מבנה הנוסחה.

- בסיס: אם $\varphi = p$ (אטום) אז $n=0$!

  $\|\varphi\|_1 = \|\varphi_2\| = \ldots = true$

  כי מעניינת אותנו רק האות הראשונה.

- אם $\varphi = \neg\psi$, מהנחת האינדוקציה אם $n \geq \#_X(\psi)$ אז ψ לא מפרידה בין $p^i\neg pp^\omega$ עבור $n<i$. אבל לה אין המצב כי נתון $n \geq \#_X(\varphi)$ והרי $\varphi = \neg\psi$. אז

  $\|\psi\|_{n+1} = \|\psi\|_{n+2} = \ldots$

  ולכן מהסמנטיקה של $\varphi = \neg\psi$ נובע גם ש-

  $\|\varphi\|_{n+1} = \|\varphi\|_{n+2} = \ldots$

- אם $\varphi = \psi_1 \vee \psi_2$ זה בדיוק כמו קודם. מאחר ש- $n \geq \#_X(\varphi)$ נובע ש- $n \geq \#_X(\psi_1)$; $n \geq \#_X(\psi_2)$ ולכן ψ_1 לא מפרידה וגם ψ_2 לא מפרידה. ולכן ברור שגם $\psi_1 \vee \psi_2$ לא מפרידה.

- אם $\varphi = X\psi$; $n \geq \#_X(\varphi)$ אז $n-1 \geq \#_X(\psi)$ ולכן מהנחת האינדוקציה ψ לא מפרידה בין $p^k\neg pp^\omega$ לכל $k \geq n-1$. כלומר $\|\psi\|_n = \|\psi\|_{n+1} = \ldots$ אבל $\|X\psi\|_{n+1} = \|\psi\|_k$ לכל k. ולכן

  $\|X\psi\|_{n+1} = \|\psi\|_n = \|\psi\|_{n+1} = \|X\psi\|_{n+2} = \ldots$

- אם $\varphi = \psi_1 U \psi_2$ אז כמוקודם ברור ש- $n \geq \#_X(\psi_1)$ וגם $n \geq \#_X(\psi_2)$ ולכן ניתן להשתמש עליהן בהנחת האינדוקציה. נשים ♡ ש-

  $\|\psi_1 U \psi_2\|_j = \|\psi_2\|_j \vee (\|\psi_1\|_j \wedge \|\psi_1 U \psi_2\|_{j-1})$

  (כי $pUq \equiv q \vee (p \wedge X(pUq))$).

  נראה ש- $\|\psi_1 U \psi_2\|_{n+i}$ לא תלוי ב-i. אכן

  $\|\psi_1 U \psi_2\|_{n+i} = \|\psi_2\|_{n+i} \vee (\|\psi_1\|_{n+i} \wedge$
  $\|\psi_2\|_{n+i-1} \vee (\|\psi_1\|_{n+i-1} \wedge$
  $\vdots$
  $\|\psi_2\|_{n+1} \vee (\|\psi_1\|_{n+1} \wedge \|\psi_1 U \psi_2\|_n) \ldots))$

ו[illegible] הדברים האלה לא תלויים ב-i (מהנחת האינדוקציה).

אז $\|\psi_2\|_{n+i} = \|\psi_2\|_{n+i-1} = \dots = \|\psi_2\|_{n+1}$;

$\|\psi_1\|_{n+i} = \|\psi_1\|_{n+i-1} = \dots = \|\psi_1\|_{n+1}$

עכשיו מהסמנטיקה של $\wedge$, $\vee$ אפשר לראות שלכל ערך אחד של $\|\psi_1\|_{n+1}, \|\psi_2\|_{n+1}$; $\|\psi_1 U \psi_2\|_n$ לכל i באמת יוצא אותו ערך אחד.

(11)

<u>טענה</u>: אין נוסחת CTL שקולה ל- $\psi = AF(p \wedge Xp)$

<u>הוכחה</u>: נראה שתי סדרות של מבנים $N_1, N_2, N_3, \dots$ $M_1, M_2, \dots$ כך ש-

① לכל $0 \le i$, $N_i \not\models AF(p \wedge Xp)$, $M_i \models AF(p \wedge Xp)$

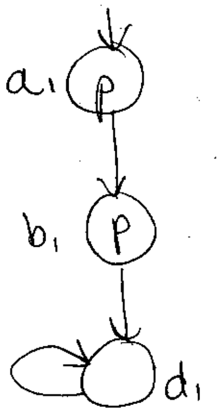
② לכל נוסחת CTL φ, אם $i \ge |\varphi|$ אז

$$N_i \models \varphi \iff M_i \models \varphi$$

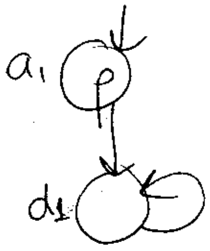
זה יגרור את הטענה כי אם בשלילה יש נוסחת CTL φ שקולה ל-ψ אז $N_{|\varphi|} \models \varphi \iff M_{|\varphi|} \models \varphi$ משלילה לכך ש-ψ מפרידה בין $M_{|\varphi|}$; $N_{|\varphi|}$.

אז נגדיר את המבנים:

M_1

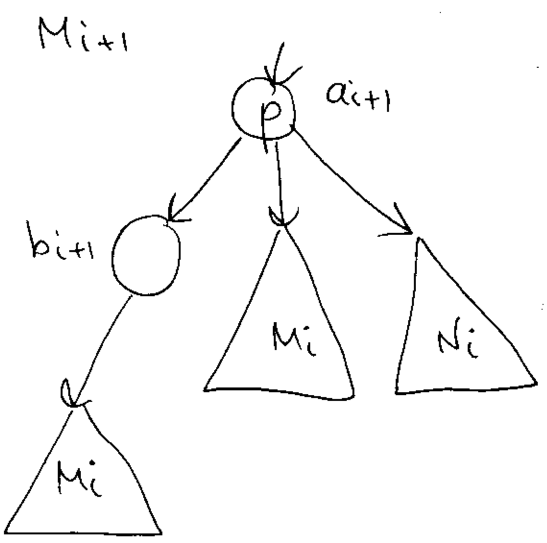


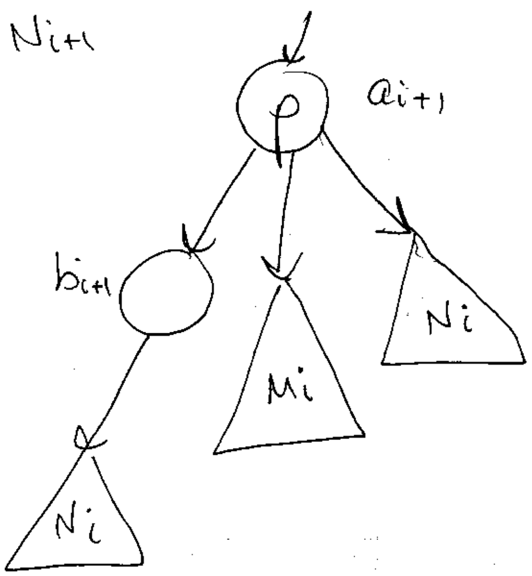
N_1



קל לראות ש-① מתקיימת עבור מבנים אלה.

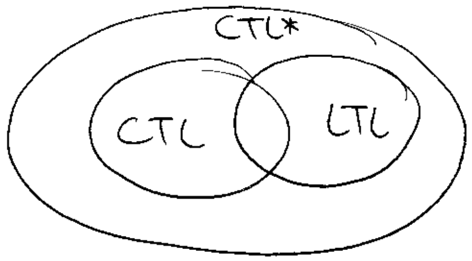
נגדיר באינדוקציה את M_{i+1} ואת N_{i+1}.





נראה ש- $M_i \models \psi$ לכל i
באינדוקציה על i. עבור $i=1$ ראינו. כעת נתבונן
ב- M_{i+1}. בשני המסלולים הימניים ברור שיש שני p-ים
כי הקודקוד הראשון הוא p והקודקוד הראשון בשאר
העצים הוא גם p. ואילו במסלול השמאלי זה ברור
כי המסלולים בתת עץ שלו מספקים $F(p \wedge Xp)$.
מאידך לכל i $N_i \not\models \psi$. צריך להראות שתמיד קיים מסלול
שלא מספק את ψ. וזה ברור כי המסלול השמאלי ביותר
הוא מהצורה $(p\neg p)^*(p\neg)^\omega$.
נותר להוכיח שלכל i אם $|\varphi| \le i$ אז
(*) $N_i, a_i \models \varphi \iff M_i, a_i \models \varphi$ (לכל φ ב-CTL).
נשים לב ש-(*) גורר שלכל i אם $|\varphi| \le i$ אז
$N_{i+1}, b_{i+1} \models \varphi \iff M_{i+1}, b_{i+1} \models \varphi$
מפה אפשר להמשיך באינדוקציה על מבנה הנוסחה.
פירוט באתר הקורס. ☺

אז הגענו למצב



בעיה: בהינתן נוסחה ב- LTL, הכרע האם יש לה נוסחה שקולה ב- CTL ?

($A\Psi$)

ידוע: בהינתן נוסחה ב- CTL, האם יש לה נוסחה שקולה ב- LTL ?
יש רק מועמד אחד שצריך לבדוק. עבור נוסחה Ψ ב- CTL נסמן ב- Ψ^d את נוסחת ה LTL שמתקבלת מ- Ψ ע"י השמטת כל כמתי המסלול. לדוגמא אם $\Psi = AXp \vee AXq$ אז $\Psi^d = Xp \vee Xq$.

משפט: יש ל- Ψ נוסחה שקולה ב- LTL אמ"מ $\Psi \equiv A\Psi^d$.

לבדוק שקילות בין נוסחאות זה משהו שאנחנו יודעים ולכן אפשר לענות על השאלה.

CTL Model Checking
(1981 Emerson, Clarke)

נתאר את הבעיה:
קלט: מבנה קריפקה K, נוסחת CTL Ψ
פלט: האם $K \models \Psi$?

תזכורת: CTL זו השפה עם הזוגות
CTL: p, $\neg$, $\vee$, AX, EX, AU, EU

אנחנו נניח שנוסחאות CTL הן מהצורה
$$\varphi ::= AP \mid \neg\varphi \mid \varphi \vee \varphi \mid EX\varphi \mid EG\varphi \mid E\varphi U \varphi$$
(לדוגמא $AXp \rightsquigarrow \neg EX \neg p$. בתרגול נראה שזו אכן הנחה לגיטימית). זה גם לא מנפח את הנוסחה יותר מדי.

נגדיר: **הסגור** (closure) של נוסחה הוא אוסף תת הנוסחאות של הנוסחה - זו הקבוצה הקטנה ביותר של

נוסחאות המקיימות:

① $\varphi \in cl(\varphi)$

② אם $\neg\psi$, $EX\psi$ או $EG\psi$ שייכים ל- $cl(\varphi)$

אז גם $\psi \in cl(\varphi)$.

③ אם $\psi_1 U \psi_2$ או $\psi_1 \vee \psi_2$ שייכים ל $cl(\varphi)$

אז גם $\psi_1, \psi_2 \in cl(\varphi)$.

דוגמה:

$$cl(E(AXp)Uq) = \{E(AXp)Uq,\ AXp,\ q,\ p\}$$

נשים לב ש- $|cl(\varphi)| \le |\varphi|$

<u>ModelCheck(K, φ)</u>

let $\varphi_1,\dots,\varphi_n$ be $cl(\varphi)$ partially ordered:

$\varphi_1 \le \varphi_2 \iff \varphi_1 \in cl(\varphi_2)$

(כלומר נוסחאות פנימיות מופיעות ראשונות)

for $i = 1 .. n$

label(K, φ_i)

(12) 11/11/09
אימות פורמלי

CTL Model Checking

הגדרנו את הסגור של נוסחה φ והגדרנו יחס סדר חלקי
על נוסחאות - $\varphi_1 \le \varphi_2$ אמ"מ $\varphi_1 \in Cl(\varphi_2)$.
הרעיון של האלגוריתם הוא להתחיל מהנוסחה הכי פנימית ובלולאה
לבדוק איזה מהמצבים מקיים את φ_i.

MC(K, φ)

1) let $\varphi_1 \le \varphi_2 \le \dots \le \varphi_n$ be the formulas in $cl(\varphi)$
2) for $i = 1, \dots, n$
   label(K, φ_i)

מה זה label? יש לנו $L: W \to 2^{AP}$ ואותה אפשר
להרחיב ל- $L': W \to 2^{cl(\varphi)}$. אז label למעשה
מחשבת את L'.

label(K, φ)

1) case φ is
   - $\varphi = p \in AP$: $\forall w$, if $p \in L(w)$ then $L'(w) = L'(w) \cup \{p\}$
   - $\varphi = \varphi_1 \vee \varphi_2$: $\forall w$, if $\varphi_1 \in L'(w)$ or $\varphi_2 \in L'(w)$
     then $L'(w) = L'(w) \cup \{\varphi_1 \vee \varphi_2\}$
   - $\varphi = \neg \varphi_1$: $\forall w$, if $\varphi_1 \notin L'(w)$ then $L'(w) = L'(w) \cup \{\neg \varphi_1\}$
   - $\varphi = EX\varphi_1$: $\forall w$, if $\exists w'$ s.t. $R(w, w')$ and $\varphi_1 \in L'(w')$
     then $L'(w) = L'(w) \cup \{EX\varphi_1\}$
   - $\varphi = E\varphi_1 U \varphi_2$: $T = \{w : \varphi_2 \in L'(w)\}$
     $\forall w \in T$ do $L'(w) = L'(w) \cup \{E\varphi_1 U \varphi_2\}$
     while $T \neq \emptyset$
       let $t \in T$
       $T = T \setminus \{t\}$

$\forall w'$ s.t. $R(w',t)$

[אם עוד לא הוכנס] if $E\varphi_1 U\varphi_2 \notin L(w')$ and $\varphi_1 \in L'(w')$

then $T = T \cup \{w'\}$

$L'(w') = L'(w) \cup \{E\varphi_1 U\varphi_2\}$

נשים ♡ שכל מצב נכנס ל-T רק פעם אחת, כי אחרי הכניסה מסמנים אותו. וכאשר מצב נכנס ל-T הוא דורש טיפול שהוא O(מציאת הכניסה שלו). לכן זה לינארי במספר הקשתות בגרף.

- $\varphi = EG\varphi_1$:

Tarjan: מציאת רכיבים קשירים היטב מקסימליים ב- O(|E|)

נתון גרף מכוון $G = \langle V, E \rangle$

קבוצת מצבים $S \subseteq V$ היא רכיב קשיר היטב אם לכל $u, v \in S$ יש מסלול (מכוון) מ-u ל-v, העובר דרך קודקודים ב-S בלבד.

S הוא רכיב קשיר היטב מקסימלי אם S רק"ם וגם $S \cup S'$ אינו קשיר היטב לכל $S' \neq \emptyset$.

יש חלוקה יחידה ל- MSCC ואפשר למצוא אותה ב- O(|E|).

נתון $K = \langle AP, W, W_0, R, L \rangle$. עבור $S \subseteq W$ נגדיר את הצמצום של K ל-S:

$$K_S = \langle AP, S, W_0 \cap S, R \cap (S \times S), L \rangle$$

מצב w ב-K מקיים EG_S (קיים מסלול שכל מצביו ב-S) אמ"מ קיים ב-K_S מסלול מ-w ל- MSCC לא טריוויאלי (אורך לפחות קשת אחת)

$S = \{w : \varphi_1 \in L'(w)\}$

$R_S = R \cap S \times S$

$\mathcal{S} = \{c : c \text{ is a non trivial MSCC in } \langle S, R_S \rangle\}$

$T = \{w : \exists c \in \mathcal{S} \text{ s.t } w \in c\}$

$\forall w \in T$ do

$\quad L'(w) = L'(w) \cup \{EG\,\varphi_1\}$

while $T \neq \emptyset$

$\quad$ let $t \in T$

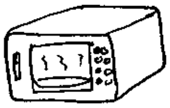
$\quad T = T \setminus \{t\}$

$\quad$ for all $s \in S$

$\quad\quad$ if $EG\,\varphi_1 \notin L'(s)$ and $R(s,t)$

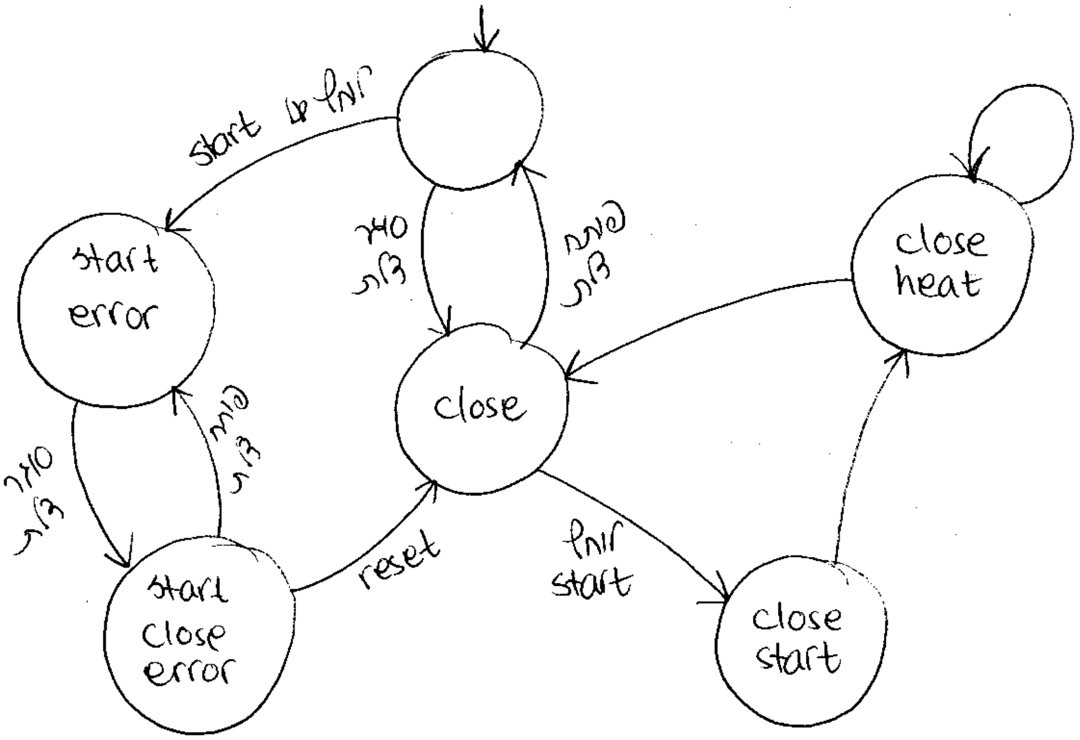
$\quad\quad$ then $T = T \cup \{s\}$

$\quad\quad\quad L'(s) = L'(s) \cup \{EG\,\varphi_1\}$



<u>דוגמה:</u> (ציור מצב של מיקרוגל)

$AP = \{start, close, heat, error\}$



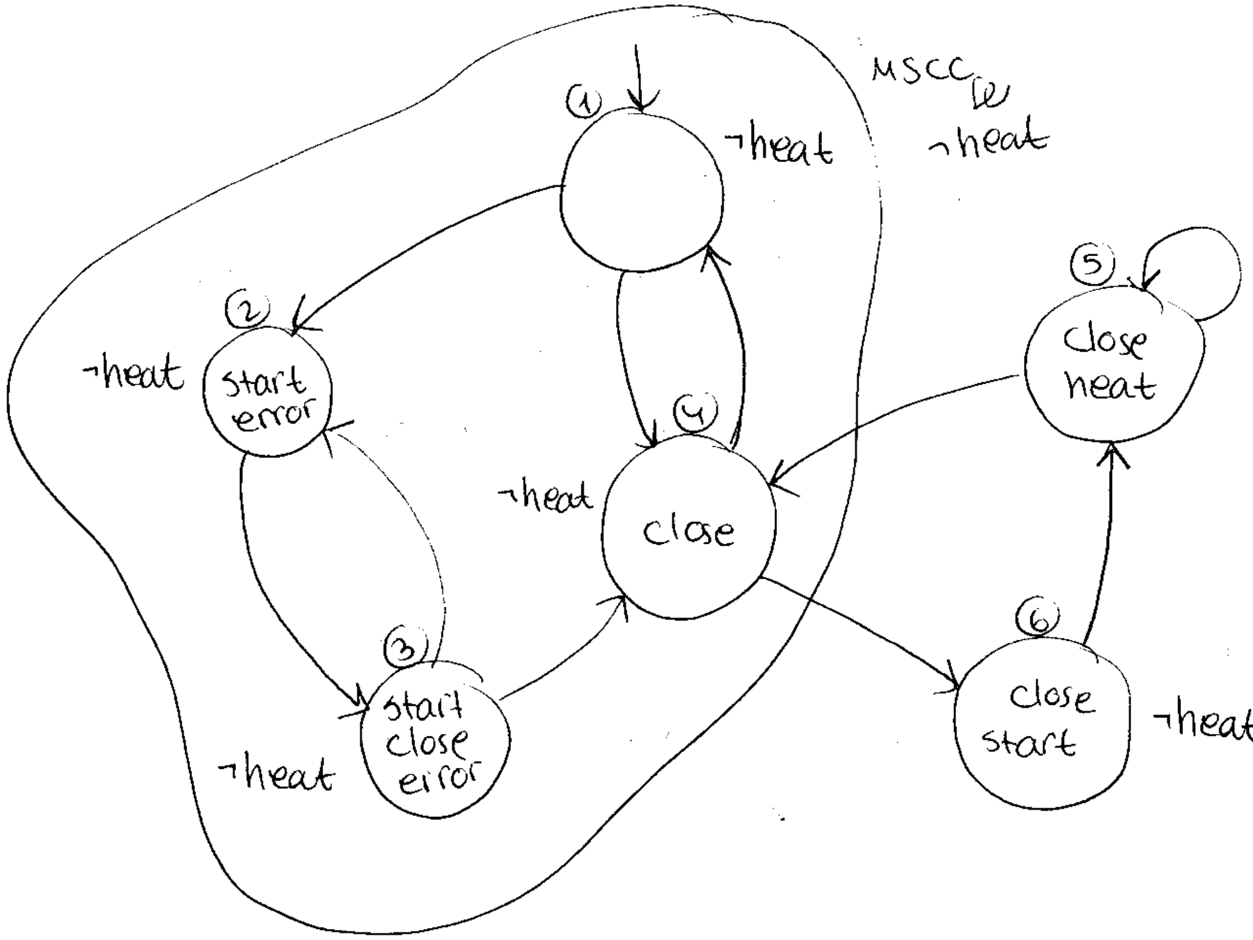
נושא model checking ונוסחה

$AG(Start \longrightarrow AF\ heat)$

נעבור לבדוק באלגוריתם אנית:

$\Psi = \neg EF(start \wedge EG \neg heat)$

$cl(\Psi) = \{ start, heat, \neg heat, EG\neg heat, start \wedge EG\neg heat, EF(start \wedge EG\neg heat), \Psi \}$



נסמן $\|\Psi\| = \{w : w \models \Psi\}$ אז

$\|EG\neg heat\| = \{w_1, w_2, w_3, w_4\}$

$\|start \wedge EG\neg heat\| = \{w_1, w_2\}$

$\|EF(start \wedge EG\neg heat)\| = W$

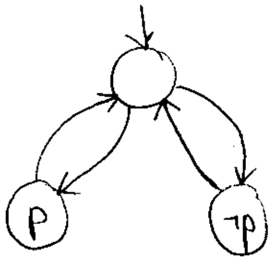
$\|\Psi\| = \emptyset$

כי מהמשוואות של הקודם.

$\Rightarrow \quad K \not\models \Psi$

Fairness - הוגנות

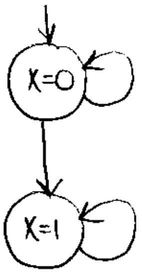
נניח שיש לנו גרף כזה



האם AFp ? לא! כי יש לנו מסלול שיש בו רק $\neg p$. אבל אנחנו רוצים להגיד ל- scheduler שמשאיר אותנו רק במעגל הימני הוא לא הוגן ואנחנו לא רוצים להסתכל על מסלולים לא הוגנים.

נסתכל בתכנית הבאה:

```
x = 0
while x = 0
    skip or x=1      // בחירה לא דטרמיניסטית
skip
```



מבנה קריפקה שמתאר את התכנית:

האם $\models AF\, x=1$?

לא! כי יש מסלול שנשאר רק במצב הראשון, אבל זה לא מסלול הוגן.

אך באופן הוגן $\models_F AF\, x=1$.

איך נגדיר זה פורמלית.

מבנה קריפקה עם הוגנות (Fair KS) הוא שישיה $K=\langle AP, W, W_0, R, L, F\rangle$ כאשר $\langle AP, W, W_0, R, L\rangle$ מבנה קריפקה ו- F תנאי הוגנות (unconditional) $F=\{F_1,\dots,F_k\} \subseteq 2^W$

עבור מסלול π ב- K

$$inf(\pi) = \{w : \pi \text{ מבקר ב-} w \text{ אינסוף פעמים}\} =$$

$$= \{w : w=w_i \text{ עבור } \infty \text{ i-ים}\}$$

$\pi = w_1 w_2 \dots$

π הוגן ביחס ל- F אם לכל $1 \le i \le k$ $\quad inf(\pi) \cap F_i \neq \emptyset$

Fair CTL* - הסינטקס כמו CTL* רגיל.

סמנטיקה- $w \models_F E\psi$ אמ"מ קיים מסלול הוגן π רק ש- $\pi \models \psi$.

יהי $F = \{\theta_1, \dots, \theta_k\}$ כאשר θ_i פרדיקט מעל AP, אז $F_i = \|\theta_i\|$.

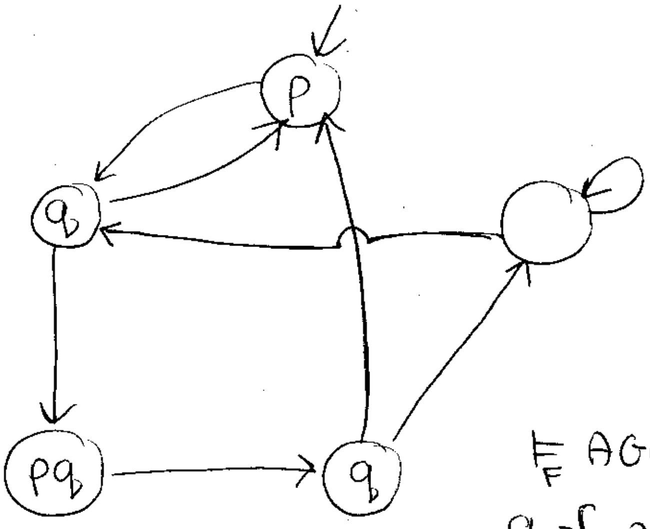
נתונה ψ ב- CTL* ורוצים ψ' רק שלכל K
$K \models \psi' \Leftrightarrow K \models_F \psi$ (כלומר הוגנות לא מוסיפה כח הבעה).

ψ' תתקבל רק: נתחיל אינדוקציות פנימיות. ואז

- נחליף את $E\psi$ ב- $E(\psi \wedge GF\theta_1 \wedge \dots \wedge GF\theta_k)$
- נחליף את $A\psi$ ב- $A((GF\theta_1 \wedge \dots \wedge GF\theta_k) \rightarrow \psi)$

ב- CTL ההוגנות דווקא כן מוסיפה כח הבעה.

נחזור למיקרוגל שלנו. נניח ש- $F = \{\|\neg error\|\}$. האם עכשיו מתקיימת $AG(start \rightarrow AF\, heat)$? עדיין לא, כי המסלול $w_1 w_2 w_3 w_4 w_1 \dots$ הוא הוגן - יש לו ∞ מיקרים ב- w_4 וב- w_1 שאין בהם error וגם ישנו start אבל אין heat...



<u>דוגמה:</u>

$\not\models AGAFq$
כי יש את המצב שלו מתקיים בו כלום.

אבל אם נוסיף תנאי הוגנות $F = \{\|p\|\}$ אז $\models_F AGAFq$
כי אחרי כל מצב q יש מעבר ל-q
וכן אם נבקר ב-p אינסוף פעמים מכאן גם נבקר ב-q אינסוף פעמים.

נראה שצריך להרחיב את אלגוריתם הבדיקה שלנו.
ברור שכל הטיפול באופרטורים לוגיים נשאר זהה. צריך רק לטפל באופרטורים הזמניים.

Fair-CTL MC

נגדיר אטום חדש fair ונאמר ש- $w \models fair$ אמ"מ קיים מ-w מסלול הוגן, כלומר אמ"מ $w \models_F EG\, true$.
נסמן את כל המצבים ה- fair בעיבוד מקדים.

case φ is

$\varphi = EX\varphi_1$:

$$w \models_F EX\varphi_1 \Longleftrightarrow w \models EX(\varphi_1 \wedge fair)$$

$\varphi = E\varphi_1 U \varphi_2$:

$$w \models_F E\varphi_1 U \varphi_2 \Longleftrightarrow w \models E\varphi_1 U(\varphi_2 \wedge fair)$$

$\varphi = EG\varphi_1$:

$$w \models_F EG\varphi_1 \Longleftrightarrow ?$$

<u>למה</u>: $w \models_F EG\varphi_1$ אם יש ב- $K_{\|\varphi_1\|}$ MSCC לא טריוויאלי <u>הוגן</u> שאפשר להגיע אליו מ- w. [MSCC S הוגן אם $S \cap F_i \neq \emptyset$ לכל $F_i \in F$]

<u>הוכחה</u>: כי אפשר לעשות מסלול שיבקר בכל ה-F_i-ים שהרי כל אחד מהם מופיע ב- MSCC לפחות פעם אחת.

נוכיח תחילה ש- $w \models_F E\varphi_1 U \varphi_2$ אמ"מ $w \models E\varphi_1 U(\varphi_2 \wedge fair)$.

$w \models_F E\varphi_1 U \varphi_2$ אמ"מ קיים מסלול π כך ש- $\pi \models \varphi_1 U \varphi_2$ כך ש- $inf(\pi) \cap F_i \neq \emptyset$ לכל $1 \le i \le k$

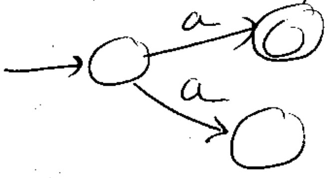
הכיוון $\Leftarrow$ הוא ברור. אבל גם $\Rightarrow$ לא קשה. יהי $\pi' = w_1 w_2 \ldots$ מסלול הוגן שמתחיל במצב שבו φ_2 מתקיים.
אז $\pi = t_1 \ldots t_j w_2 w_3 \ldots$ כאשר $t_1, \ldots, t_j, t_{j+1} \ldots$ הוא המסלול שמתחיל ב- w ושבו $t_j \models \varphi_2 \wedge fair$ ו- $t_l \models \varphi_1$ לכל $1 \le l < j$

(16) 19/11/09
אימות פורמלי

אוטומטים מעל מילים אינסופיות (Büchi, 62)

<u>תזכורת</u>: אוטומט הוא חמישיה $A=\langle\Sigma,Q,Q_0,\delta,\alpha\rangle$
כאשר - Σ קבוצה סופית של א"ב
- Q קבוצת מצבים
- $Q_0 \subseteq Q$ מצבים התחלתיים
- $\delta: Q\times\Sigma\to 2^Q$ פונקצית מעברים (לא דטרמיניסטית)
- $\alpha\subseteq Q$ מצבים מקבלים

שפה של אוטומט היא שפת כל המילים שהוא מקבל. כאשר מדובר במילים סופיות שפות רגולריות סגורות לחיתוך, למשלים, לאיחוד, ולשרשור.
את הסגירות לחיתוך ואיחוד אפשר לראות ע"י הסתכלות באוטומט המכפלה (שגודלו מכפלת הגדלים).
לאוטומט לא דטרמיניסטי אפשר לעשות חירצון - תהליך שבו האוטומט הופך לדטרמיניסטי ויש לו אותה שפה. בתהליך הזה האוטומט יכול לגדול אקספוננציאלית כי $Q'=2^Q$ ואת δ' מגדירים ע"י $\delta'(S,\sigma)=\bigcup_{s\in S}\delta(s,\sigma)$. המצבים המקבלים הם $\alpha'=\{S: S\cap\alpha\neq\emptyset\}$.
מתהליך החירצון רואים שלא סגור להשלמה - אחרצנים אוטומט ואז עושים דואליזציה - כל מצב מקבל הופך ללא מקבל ולהפך. <u>חשוב</u> לעשות קודם חירצון. למשל



$L(A)=\{a\}$

אם לא נעשה דואליזציה נקבל $L(\tilde{A})=\{\varepsilon, a\}$
וזה לא המשלים...
אפשר להראות שלתהליך חירצון יש חסם $\Omega(2^n)$ - חסם תחתון.
וההשלמה עושים בעזרת חירצון, אבל זה לא אומר שגם השלמה

בהכרח כרוכה בגידול מעריכי. למשל השפה הסופית

$L_n = \{ww : w \in \{a,b\}^n\}$

היא רגולרית (כי היא סופית). אבל דווקא למשלים

שלה יש אוטומט פשוט כי רק צריך דוגמה, אבל

אוטומט לא דטרמיניסטי יכול לנחש איפה אמצע המילה.

נעבור לאוטומטים מעל מילים אינסופיות.

במקרה זה הקלט לאוטומט הוא מילה אינסופית $w=w_1w_2\ldots \in \Sigma^\omega$

ריצה היא כמו ריצה במקרה הסופי, רק אינסופית -

$q_0q_1q_2\ldots$ כאשר לכל $i \geq 0$ $q_{i+1} \in \delta(q_i, w_{i+1})$

אפשר לומר שריצה r היא פונקציה $r : \mathbb{N} \to Q$.

איך מקבלים מילה? קבוצת המצבים היא סופית, אבל

ריצה היא אינסופית. לכן יש מצב שחוזר אינסוף פעמים.

נגדיר

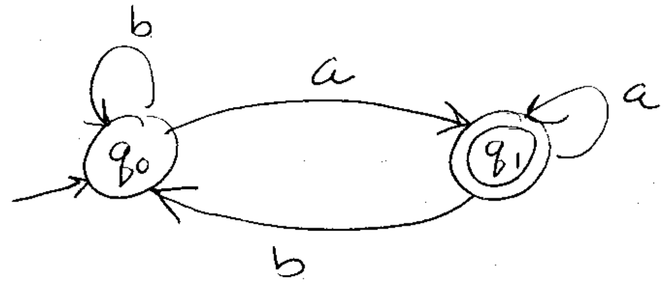
$inf(r) = \{q : r(i) = q$ עבור אינסוף i-ים$\}$

- קבוצת המצבים ש-r מבקרת בהם אינסוף פעמים

אז נאמר שריצה מקבלת אם $inf(r) \cap \alpha \neq \emptyset$, כלומר יש

מצב מקבל ש-r מבקרת בו אינסוף פעמים.

דוגמה:

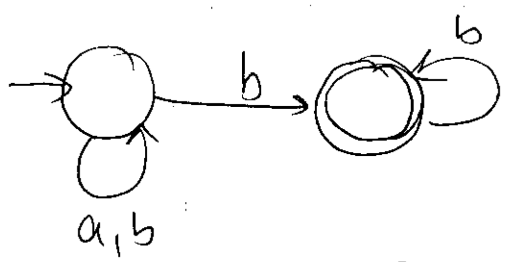


$L(A) =$ כל המילים עם אינסוף a-ים

$= (b^*a)^\omega$ (= אינסוף מקומות שיש בהם a באמצע ו-b-ים)

דוגמה: נצייר אוטומט לשפה שיש בה רק מספר סופי של a-ים.

יתכן שיהיה לא דטרמיניסטי :

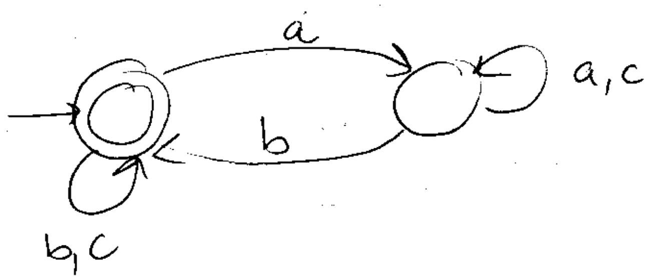


אם יש רק מספר סופי של a-ים אז יש זנב שיש בו רק b-ים.
אז בהתחלה האוטומט יקרא את החלק שיש בו את כל ה-a-ים
ואז יעבור לנצח ה b-ים. גם רואים את זה בביטוי
הרגולרי $(a+b)^* b^\omega$.

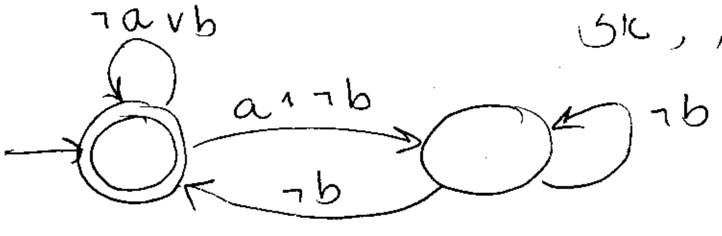
מעניין לשים ♡ : כאן דואליזציה לא היתה עובדת!
אם היינו עושים דואליזציה לאוטומט הראשון היינו מקבלים
את שפת המילים שיש בהן ∞ b-ים וזה לא המשלים!

<u>דוגמה:</u> $G(a \to Fb)$ $\Sigma=\{a,b,c\}$

כאן בכל רגע יכול להיות רק אחד מהם.



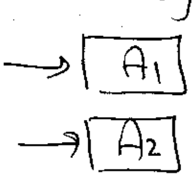
ואם ניקח $\Sigma = 2^{\{a,b,c\}}$, כלומר בכל רגע יכול להיות כל צירוף שלהם, אז:



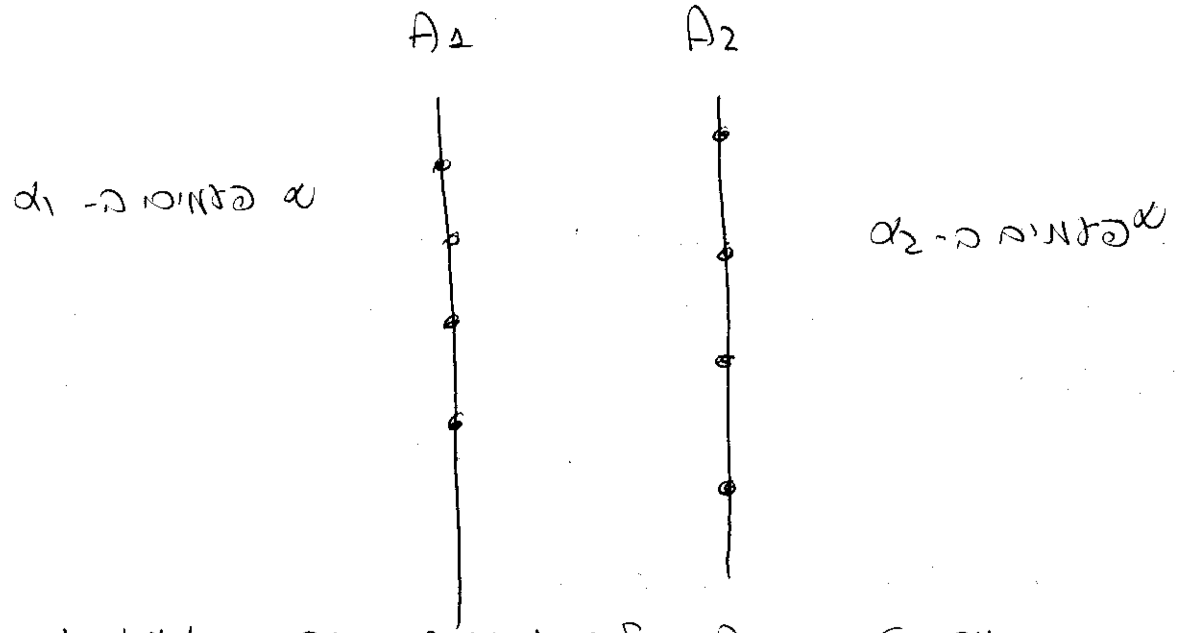
כאשר x = כל הקבוצות ש-x מופיע בהן
¬x = כל הקבוצות ש-x לא מופיע בהן.

תכונות סגור

- <u>איחוד</u> - זה ברור! פשוט נשים את שני האוטומטים אחד ליד השני.



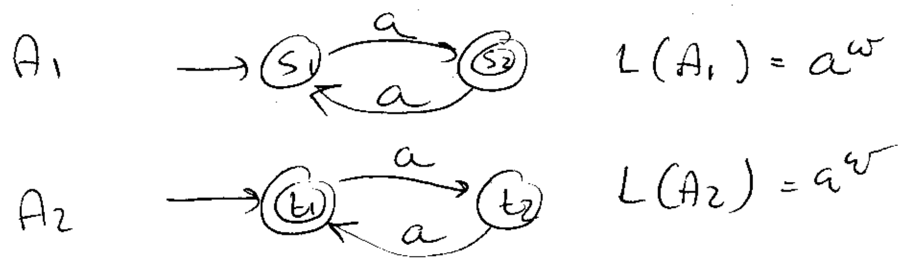
- <u>חיתוך</u> - האם אוטומט המכפלה מגדיר את החיתוך?



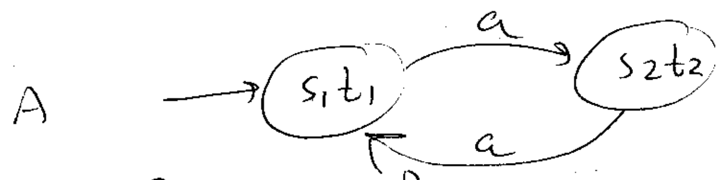
אוטומט המכפלה לא יעבוד כי הגדרנו $\alpha=\alpha_1\times\alpha_2$

וזה יודרש סינכרוניזציה - אנחנו צריכים שהמעבר ב- α-ות יקרה באותו זמן, דוגמה:

(אם $\Sigma=\{x\}$ אז לכל אוטומט A או ש- $L(A)=\{x^\omega\}$ או ש- $L(A)=\emptyset$)



אם נעשה אוטומט מכפלה נקבל



אבל $L(A)=\emptyset$ כי אף אחד מהמצבים לא מקבל.

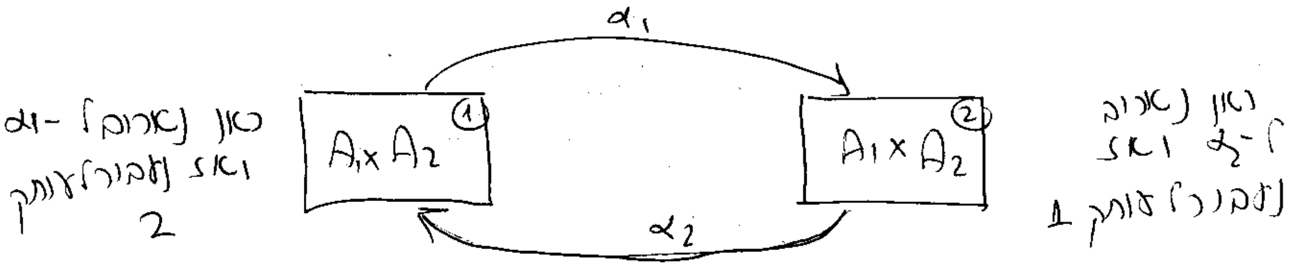
<u>מושגים:</u> NBW סגורים לחיתוך.

NBW: non deterministic, Büchi (תנאי הקבלה), Word (שאומר האוטומט רץ על מילים אינסופיות).

<u>הוכחה:</u> בהינתן A_1 ו- A_2 נבנה A כך ש- $L(A)=L(A_1)\cap L(A_2)$

כבר ראינו שסתם אוטומט מכפלה לא ילך.

אז ניקח שני אוטומטי מכפלה



פורמלית:

$$Q = Q_1 \times Q_2 \times \{1,2\}$$
$$Q^0 = Q_1^0 \times Q_2^0 \times \{1\}$$
$$\delta(\langle q_1, q_2, i\rangle, \sigma) = \delta_1(q_1,\sigma) \times \delta_2(q_2,\sigma) \times \{j\}$$

כאשר:

- $j=2$ אז ($i=1$ ו- $q_1 \in \alpha_1$ או $i=2$ ו- $q_2 \notin \alpha_2$)
- $j=1$ אז ($i=2$ ו- $q_2 \in \alpha_2$ או $i=1$ ו- $q_1 \notin \alpha_1$)

ונגדיר $\alpha = \alpha_1 \times Q_2 \times \{1\}$

אם היינו ב- α אינסוף פעמים, פירוש הדבר היינו ב- $\alpha_1 \times Q_2$ בעותק הראשון אינסוף פעמים. אבל לפי ההגדרה של δ זה אומר שבהכרח עברנו לעותק השני. אבל גם חזרנו ממנו לעותק הראשון. לכן בין שני ביקורים ב- α_1 בעותק הראשון בהכרח היינו ב- α_2 בעותק השני. לכן בהכרח מבקרים בשניהם אינסוף פעמים.

באופן דומה ניתן היה להגדיר $\alpha = \alpha_2 \times Q_1 \times \{2\}$.

(:

משפט: אין אוטומט דטרמיניסטי עבור $(a+b)^* b^\omega$

(Landweber 69 שישלו אותו והודיה בבית שמש)

הוכחה: נניח בשלילה שיש אוטומט A עם n מצבים.

נתבונן ב- $w_1 = b^\omega$. כמובן $w_1 \in L(A)$ ⇐ הריצה של A על w_1 מקבלת והיא עוברת אינסוף פעמים ב- α. לכן יש

i_1 כך ש- $\delta(b^{i_1}) \in \alpha$ (-המצב שאליו A מגיע אחרי קריאת b^{i_1} ! -A דטרמיניסטי ולכן הוא מוגדר באופן יחיד).

נתבונן ב- $w_2 = b^{i_1}ab^{\omega}$. אם w_2 בשפה אז יש i_2 כך ש- $\delta(b^{i_1}ab^{i_2}) \in \alpha$.

נתבונן ב- $w_3 = b^{i_1}ab^{i_2}ab^{\omega}$ אם היא בשפה אזי יש i_3 כך ש- $\delta(b^{i_1}ab^{i_2}b^{i_3}) \in \alpha$.

נמשיך הלאה. יש $i_{|\alpha|+1}, \ldots, i_1$ כך ש-

$\delta(b^{i_1}ab^{i_2}a\ldots ab^{i_k}) \in \alpha$ לכל $1 \le k \le |\alpha|+1$

אבל אז יש $j<l$ כך ש- $\delta(b^{i_1}a\ldots ab^{i_j}) = \delta(b^{i_1}a\ldots ab^{i_l})$

נתבונן ב- $w = b^{i_1}a\ldots ab^{i_j}(ab^{i_{j+1}}a\ldots ab^{i_l})^{\omega}$

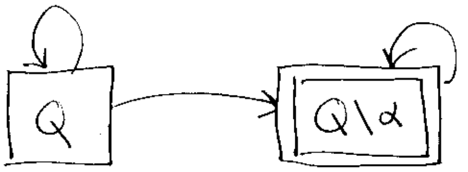
מצד אחד $w \notin L(A)$ כי יש בה ∞ a-ים, אבל מצד שני האוטומט יקבל אותה כי בכל פעם שאנחנו מסיימים לקרוא את המעגל שחוזר על עצמו אנחנו מגיעים למצב ב- α והרי האוטומט דטרמיניסטי ולכן המעגל תוך חוזר על עצמו. ☺

<u>מסקנה</u>: NBW <u>לא</u> סגורים לחירגון! אז

DBW < NBW

- <u>השלמה</u> - האם בהינתן NBW אפשר לקבל $\overline{NBW}$? כלומר בהינתן $A \in NBW$ רוצים $\bar{A} \in NBW$ כך ש- $L(\bar{A}) = \Sigma^{\omega} \setminus L(A)$

איך עושים את זה לאוטומט דטרמיניסטי? בדוגמה לדוגמה שעשינו עם $(a+b)^* b^{\omega}$:



פשוט מנחשים מתי סיימנו לקרוא את הרישא הסופית שיש בה את כל ההתחלה.

(19)

נגדיר:

$Q' = Q \times \{1,2\}$

$\delta'(\langle q, i \rangle, \sigma) = ?$

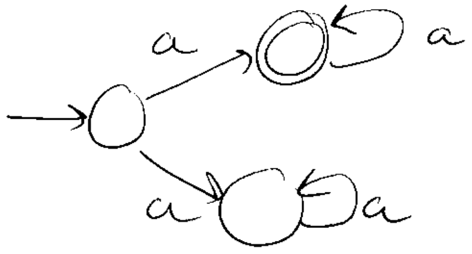
אם $i=1$; $\delta(q,\sigma)\in\alpha$ אז נלך ל- $\{\langle\delta(q,\sigma),1\rangle\}$

אם $i=1$; $\delta(q,\sigma)\notin\alpha$ אז נלך ל- $\{\langle\delta(q,\sigma),1\rangle, \langle\delta(q,\sigma),2\rangle\}$

אם $i=2$; $\delta(q,\sigma)\notin\alpha$ אז נלך ל- $\{\langle\delta(q,\sigma),2\rangle\}$

אם $i=2$; $\delta(q,\sigma)\in\alpha$ אז נלך ל- $\emptyset$.

הדרישה ש- A דטרמיניסטי היא חיונית:



$L(A) = \{a^\omega\}$

אבל אם נעשה לזה השלמה לפי השיטה שראינו נקבל $L(\tilde{A})=\{a^\omega\}$!

<u>השלמה ל- NBW</u> - ב-62 הראו בניה ב- $2^{2^{O(n)}}$

ב-86 הראו בניה משופרת - $2^{O(n^2)}$

ב-88 שיפרו ל- $2^{O(n\log n)}$

ויש גם חסם תחתון! אבל לא הדוק...

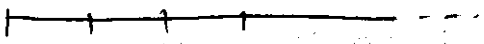
למה זה חשוב לנו? כי בדיקה אם $L(A_1) \subseteq L(A_2)$ מספיק לבדוק ש- $L(A_1) \cap \overline{L(A_2)} = \emptyset$.

איך בודקים אם $L(A) = \emptyset$? $L(A)\neq\emptyset$ אמ"מ יש מצב מקבל ישיג מעצמו.

(20) 26/11/09
אימות פורמלי

היום נראה איך מתרגמים LTL ל- NBW. (Vardi, Wolper 86)

הרעיון: נניח שיש לנו חישוב אינסופי ונוסחה



$XpUq$

נסתכל במקום הראשון בחישוב. אם q מתקיים שם אז אנחנו בסדר. אחרת אנחנו צריכים לבדוק ש- Xp ולהתקדם הלאה. רק מורישים תכונות שהאוטומט צריך לבדוק.

בהתחלה נראה תרגום יותר פשוט ל- NGBW
(G B = generalized Buchi)

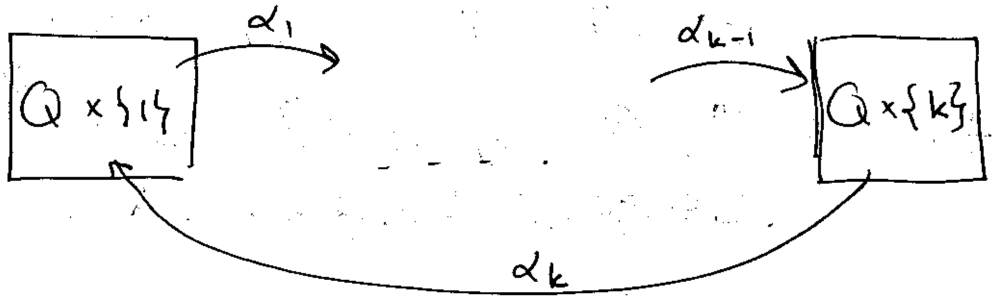
ב- Buchi רגיל היה לנו $\alpha\subseteq Q$ ותנאי הקבלה היה $inf(r)\cap\alpha\neq\emptyset$.

ב- GBuchi יהיו לנו כמה קבוצות $\alpha_1,\alpha_2,\ldots,\alpha_k\subseteq Q$
וריצה r מקבלת אמ"מ $inf(r)\cap\alpha_i\neq\emptyset$ לכל $1\leq i\leq k$.

טענה: לכל NGBW עם n מצבים ואינדקס k יש NBW שקול עם $O(nk)$ מצבים.

מסקנה: יש תרגום מ- LTL ל- NBW כי יש תרגום $LTL\rightarrow NBW$ ומשם אפשר לעבור ל- NGBW

הוכחת הטענה: נעשה משהו דומה למכפלה משבוע שעבר.
מרחב המצבים יהיה: $Q\times\{1,\ldots,k\}$



ותנאי הקבלה הוא שמבקרים אינסוף פעמים ב- $\alpha_k\times\{k\}$
כי אם היינו שם אינסוף פעמים זה אומר שעברנו ל- $Q\times\{1\}$
אינסוף פעמים ועברנו מכולם בדרך אינסוף פעמים. :)

הגדרה: עבור נוסחה ψ ב-LTL, נגדיר $cl(\psi)$ כאוסף את הנוסחאות של ψ ושלילותיהן. כלומר, הסגור הוא הקבוצה הקטנה ביותר המקיימת

1) $\psi \in cl(\psi)$
2) if $\psi_1 \vee \psi_2$ or $\psi_1 \, U \, \psi_2$ in $cl(\psi)$ then $\psi_1, \psi_2 \in cl(\psi)$
3) if $\neg\psi_1, X\psi_1 \in cl(\psi)$ then $\psi_1 \in cl(\psi)$
4) if $\psi_1 \in cl(\psi)$ then $\neg\psi_1 \in cl(\psi)$

הערה: אנחנו מזהים את ψ עם $\neg\neg\psi$ אחרת הדרישה לסגירות לשלילה **לא** יוצרת לנו קבוצה אינסופית.

דוגמה:

$$cl(p \wedge (Xp \, U \, q)) =$$
$$= \{p \wedge (Xp \, U \, q), \neg(p \wedge (Xp \, U \, q)), p, \neg p, Xp \, U \, q, \neg(Xp \, U \, q), Xp, \neg Xp, q, \neg q\}$$

נבצע את התרגום: $LTL \to NGBW$.
בהינתן נוסחת LTL ψ. נגדיר

$$A_\psi = \langle 2^{AP}, Q, \delta, Q_0, \alpha \rangle \quad Q \subseteq 2^{cl(\psi)}$$

כל מצב הוא קבוצה $S \subseteq cl(\psi)$.

הרעיון: ש-A_ψ ממצב S הוא יקבל בדיוק את כל המילים שמקיימות את כל הנוסחאות S.

(21)

קבוצה S היא <u>טובה</u> אם,

① אם $\varphi \in cl(\psi)$ אז $\varphi \in S$ אמ"מ $\neg\varphi \notin S$

② אם $\varphi_1 \wedge \varphi_2 \in cl(\psi)$ אז
$\varphi_1 \wedge \varphi_2 \in S$ אמ"מ $\varphi_1 \in S$ וגם $\varphi_2 \in S$

נגדיר את המצבים $Q = \{S \subseteq cl(\psi) : S$ טובה$\}$

עכשיו צריך להגדיר את פונקצית המעברים δ.

$S' \in \delta(S,\sigma)$ אמ"מ

① $\sigma = S \cap AP$

② לכל $X\varphi \in cl(\psi)$, $X\varphi \in S$ אמ"מ $\varphi \in S'$

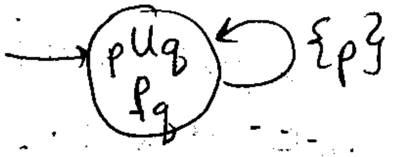
③ לכל $\varphi_1 U \varphi_2 \in cl(\psi)$,
$\varphi_1 U \varphi_2 \in S$ אמ"מ $\varphi_2 \in S$ או $\left(\begin{matrix}\varphi_1 \in S \text{ וגם} \\ \varphi_1 U \varphi_2 \in S'\end{matrix}\right)$

הורשת ה-U למצב הבא

נשים לב שמ-(3) נובע ש-
$\neg\varphi_1 U \varphi_2 \in S$ אמ"מ
$\neg\varphi_2 \in S$ וגם $\left(\begin{matrix}\neg\varphi_1 \in S \text{ או} \\ \neg\varphi_1 U \varphi_2 \in S'\end{matrix}\right)$

את המצבים ההתחלתיים נגדיר $Q_0 = \{S : \psi \in S\}$

החלק שנשאר הוא להגדיר את תנאי הקבלה. הבעיה היא שאנחנו לא רוצים שהורשה של U תדחה את ההחלטה לנצח. למשל אם $\psi = A\,pUq$ אז יש לנו



אבל אנחנו לא רוצים שהמילה $\{p\}^\omega$ תתקבל כי לא מופיע בה q בכלל...

לכל $\varphi_1 U \varphi_2 \in cl(\psi)$ יש קבוצה בתנאי α:

$F_{\varphi_1 U \varphi_2} = \{S : \varphi_2 \in S$ או $\neg(\varphi_1 U \varphi_2) \in S\}$

דוגמה:

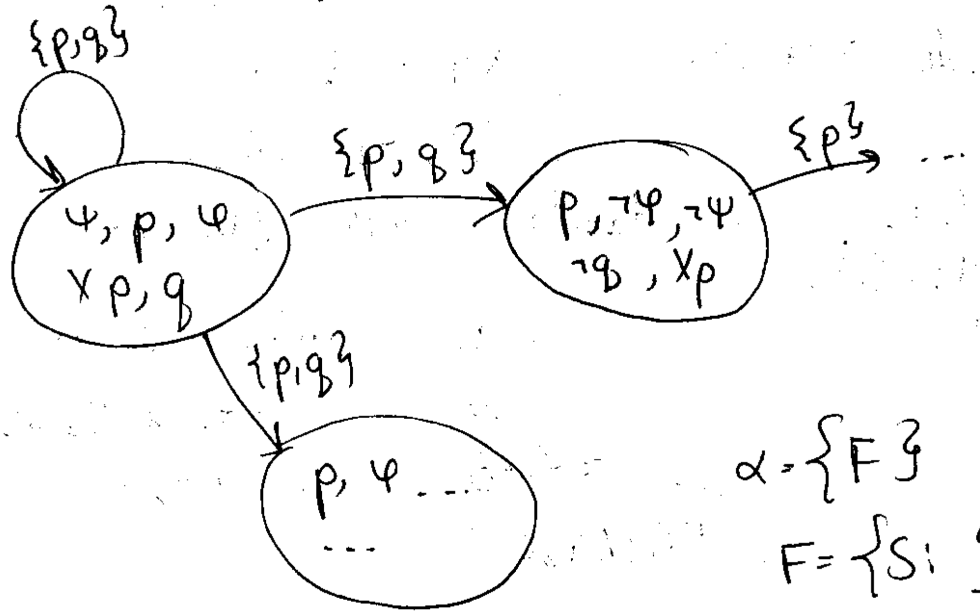
$\psi = p \wedge (Xp \, U \, q)$ $\qquad \neg(p \wedge (Xp \, U \, q))$

p $\qquad \neg p$

$\varphi = Xp \, U \, q$ $\qquad \neg(Xp \, U \, q)$

Xp $\qquad \neg Xp$

q $\qquad \neg q$



$\alpha = \{F\}$

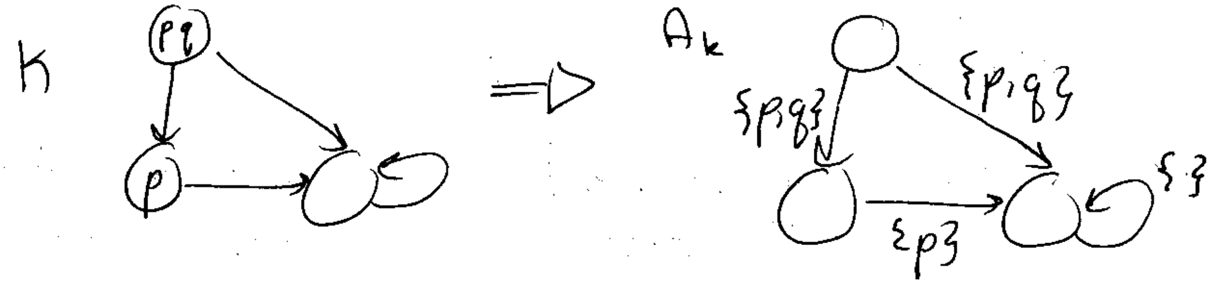
$F = \{S : q \in S \vee \neg\varphi \in S\}$

מה שנחמד פה הוא שזה אוטומטי לגמרי! אפשר לחפש בגרף המצבים כל מיני דברים שעושים את זה. וברגע שיש לנו את התרגום הזה אז גם פתרנו את model checking.

עכשיו אנחנו יודעים להפיק ספיקות ל- LTL. ψ ספיקה אמ"מ $L(A_\psi) \neq \emptyset$ ואת זה אנחנו יודעים לבדוק - אם קיים מצב מקבל שישיג ממצב התחלתי ואם ישיג מעצמו. ואת זה אנחנו פותרים ב- PSPACE (בעיית הריקנות ל- NBW היא NLOGSPACE אבל הגודל שלו מעריכי ב- $|\psi|$ אז סה"כ יוצא PSPACE).

בדיקת מודל LTL -

$K \models \psi$ אמ"מ $L(K) \subseteq L(A_\psi)$ מסתכלים על K
כאוטומט ואז יש לו שפה



אז עכשיו נותר לנו לבדוק הכלה. ידוע ש-

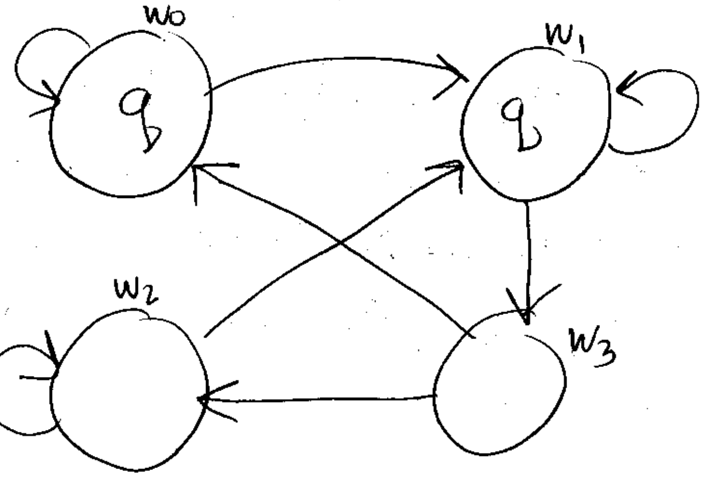
$$L(A) \subseteq L(B) \iff L(A) \cap \overline{L(B)} = \emptyset$$

וזה עדיף כי יש פה השלמה. אז שילמנו מחיר מעריכי
כדי לעבור ל-NBW ואז משלם לעבור שמשלים זה
עוד פעם מעריכי. אז במקום לבדוק $L(A) \cap \overline{L(B)} = \emptyset$
נבדוק אם $L(K) \cap L(A_{\neg\psi}) = \emptyset$. כלומר. אם
$L(K \times A_{\neg\psi}) = \emptyset$ וזה לוקח זמן $O(|K| \cdot 2^{O(|\psi|)})$
והמקום הוא PSPACE, אם בודקים מה שמטריד
אותנו הוא $|K|$ ולא האקספוננט $|\psi|$, כי הנוסחה
בד"כ קטנה אבל $|K|$ יכול להיות גדול מאוד.

אלגוריתמים סימבוליים

נסתכל על מבנה קריפקה

זה ייצוג מפורש. יש לנו רשימה של כל הקודקודים וכל הקשתות ואפשר לעבוד איתם.



אנחנו רוצים לעבור לייצוג סימבולי

למשל, אפשר לקחת שני משתנים x_1, x_2 -בוליאניים
והם יכולים לייצג את המצבים -
$w_0 = 00$
$w_1 = 01$
$w_2 = 10$
$w_3 = 11$

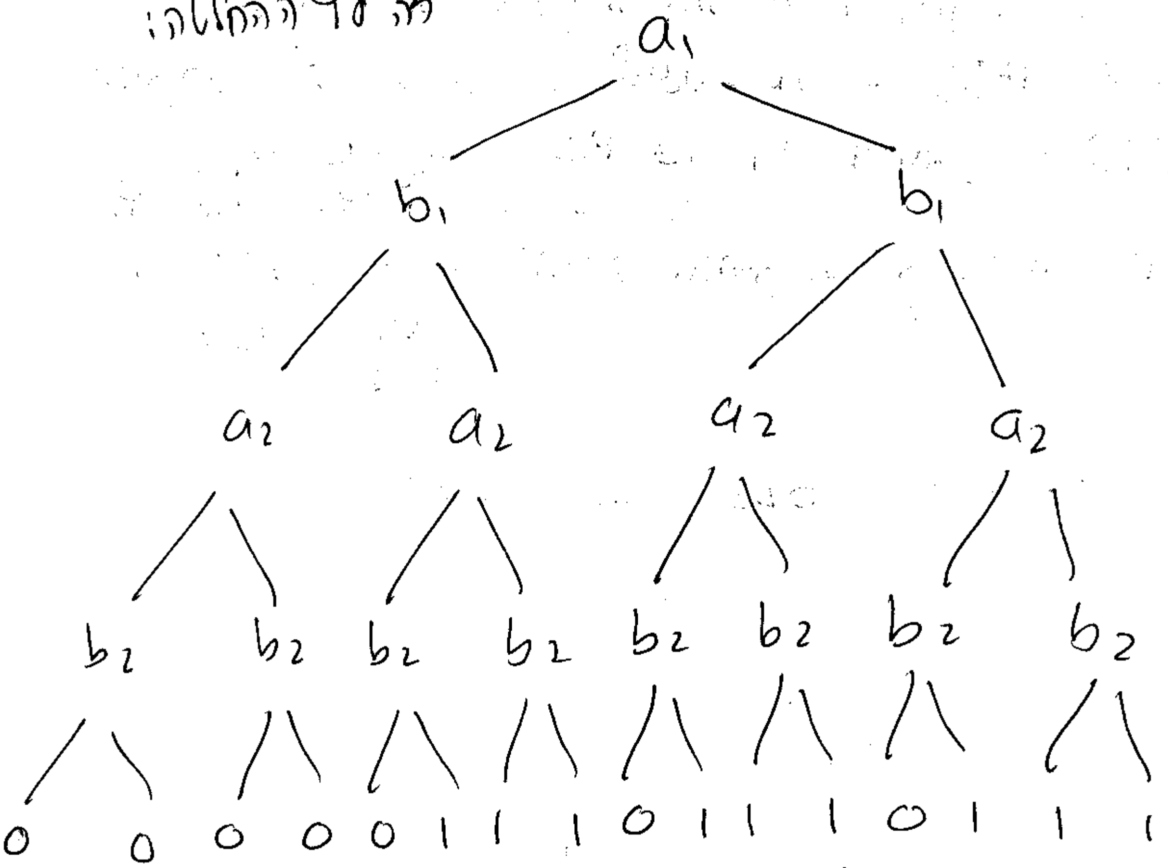
ואז את העובדה ש- $\|q\| = \{w_0, w_1\}$ אפשר להציג בנוסחה $\|q\| = \neg x_1$. קבוצת הצלעות תהיה
$R = \{ \langle w_0, w_0 \rangle, \langle w_0, w_1 \rangle, \dots \}$
גם את זה אפשר לקודד באמצעות משתנים x_1, x_2, x_1', x_2'.
למשל המעבר $w_0 w_0$ מיוצג ע"י $\bar{x}_1 \bar{x}_2 \bar{x}_1' \bar{x}_2'$
המוטיבציה לדבר הזה היא שאולי באופן סימבולי הייצוג שלנו יכול להיות הרבה יותר קטן מהייצוג המפורש.
אז ישלנו מבנה קריפקה K ואותו אפשר להעביר לפונקציה f_R מעל $\vec{x}\vec{x}'$ ולכל $p \in AP$ יש פונקציה f_p מעל $\vec{x}$ שאומרת את המצבים שבהם p נכון. לרוב, הייצוג הזה מצריך יותר קטן מהמבנה עצמו.
נרצה להפעיל את האלגוריתמים שלנו על קלט שנתון בצורה כזו (ללא הסתכלות בייצוג המפורש) ואז אולי זה יהיה מהיר יותר וייקח פחות מקום. מאז שהפכו את האלגוריתמים לסימבוליים (92) זה גם ממש שימושי בתעשייה.

BDD- Binary Decision Diagram

מבנה נתונים לתאור פונקציות בוליאניות.
אם יש n משתנים אפשר לתאר כל פונקציה בטבלת אמת בגודל $O(2^n)$. גם עץ החלטה הוא בגודל $O(2^n)$.
BDD מאפשר גם לתאר פונק' בוליאניות וכמעט תמיד הוא מאוד קטן. צריך לעבוד קשה כדי לתת דוגמה שבה הוא לא קטן יותר.

דוגמה: $(a_1 \vee b_1) \wedge (a_2 \vee b_2)$

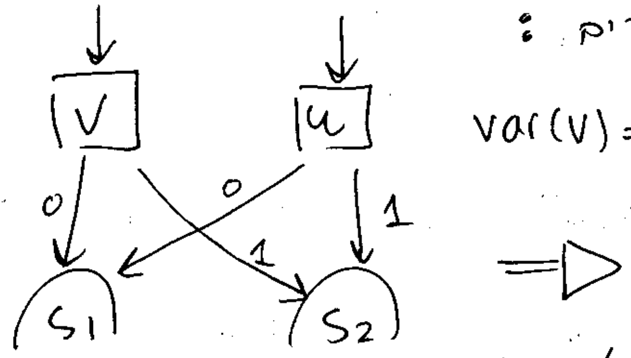
זה עץ ההחלטה:



קבלת BDD מעץ החלטה:

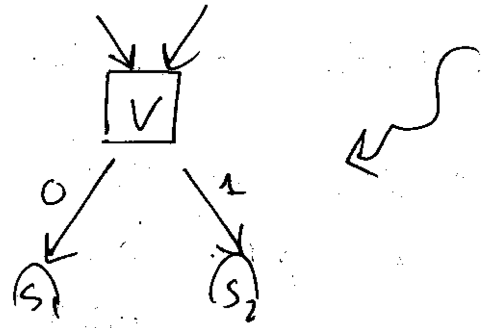
① אחד עלים זהים.

② סלק צמתים פנימיים חוזרים:

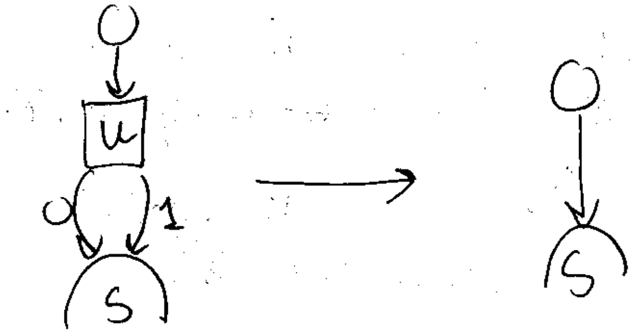


var(v)=var(u)

מוחקים את u ומכוונים קשתות שנכנסו אליו לתוך v.



③ סלק צמתים מיותרים

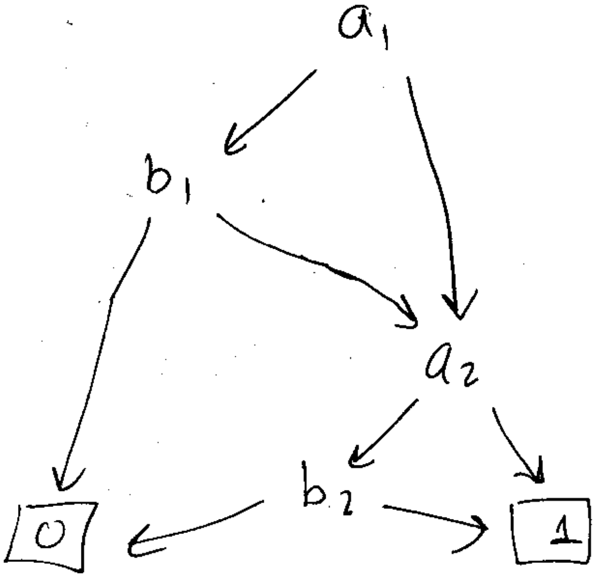


הפעלת התיקונים האלה בסדר כלשהו תביא אותנו ל- BDD של הפונקציה וזה "די" קנוני. אם שתי פונקציות הן שקולות לוגית אז ה- BDD שלהן זהה.

$f_1 \equiv f_2$ אז ה- BDD של f_1 איזומורפי ל- BDD של f_2.
(באותו סדר של המשתנים)

כדי לבדוק אם f סתירה מספיק לבדוק אם ה- BDD של f הוא $\boxed{0}$

למשל, ה- BDD שראינו קודם הופך ל-



זה הרבה יותר קטן!! לינארי!

אם הנוסחה היא $(a_1 \vee b_1) \wedge \dots \wedge (a_n \vee b_n)$
ה- BDD שלה היא בגודל $2n+2$!!

סידור לא טוב של משתנים עבור פונקציה זו הוא קודם לבדוק את a_1, a_2 ואח"כ לבדוק b_1, b_2. אז מרוויחים פחות ב- BDD.

הבעיה למצוא את סדר המשתנים האופטימלי היא PSPACE-COMPLETE. אבל יש כלי אוטומטי שעוזרים.

<u>עובדה</u>: כמעט לכל נוסחה יש סידור משתנים שאיתו יש BDD פולינומיאלי.

(24) 3/12/09

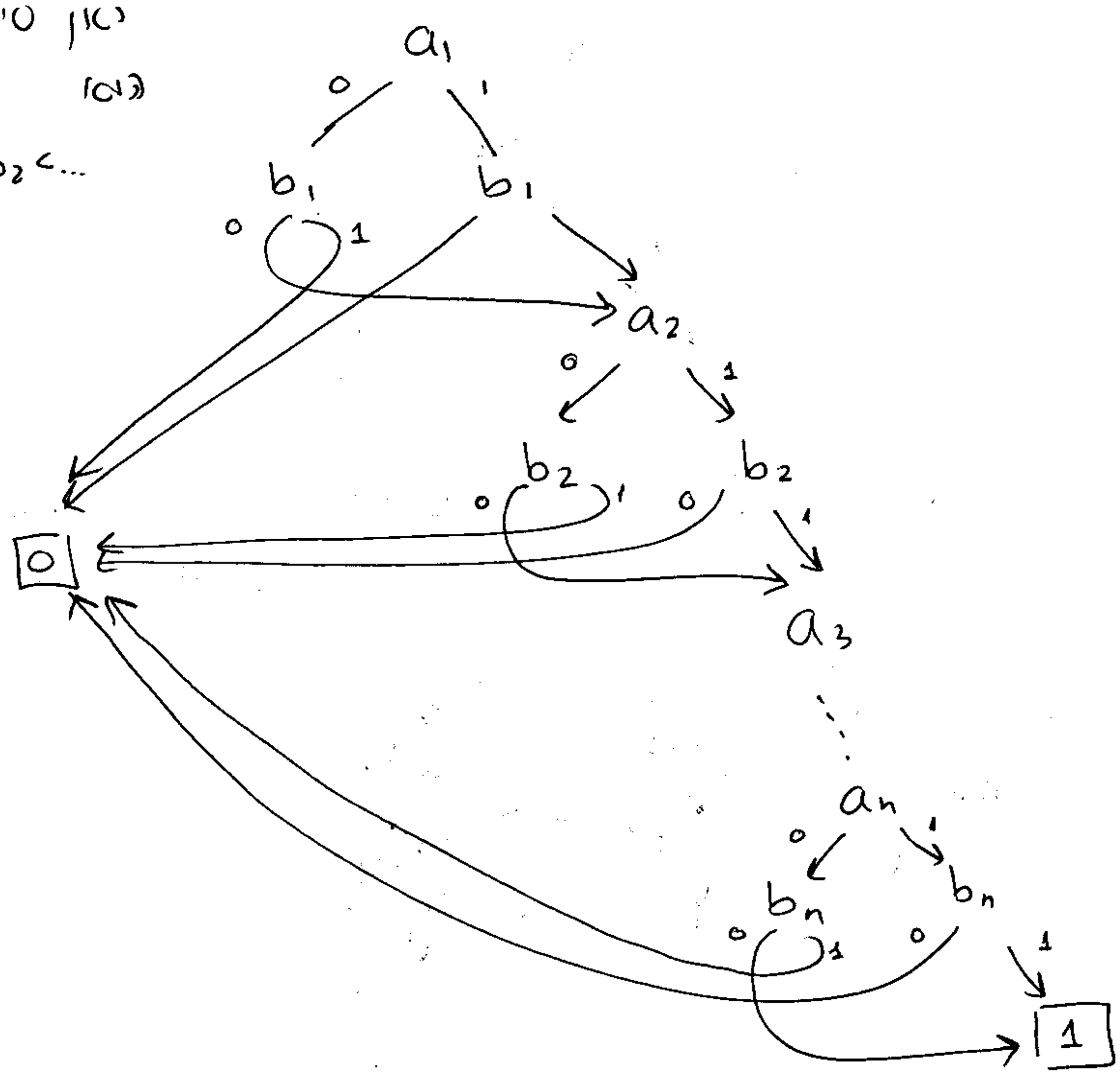
אימות פורמלי

דיברנו על BDD.

דוגמה: $a_1 \leftrightarrow b_1 \wedge a_2 \leftrightarrow b_2 \wedge \dots \wedge a_n \leftrightarrow b_n$

כאן סידור המשתנים הוא

$a_1<b_1<a_2<b_2<\dots$



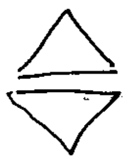
אז נמנענו שיהיה לנו מספר אקספוננציאלי של קודקודים ומצאנו יצוג עם $3n+2$ קודקודים – יש שלושה לכל (a_n,b_n) ועוד שני קודקודים - [0] ו-[1].

סידור המשתנים חשוב כאן! למשל נסתכל מסידור

$a_1<\dots<a_n<b_1<\dots<b_n$

כאן אי אפשר לעשות קיצורים כי חייבים לזכור את הערכים של כל ה-a_i-ים לפני שקוראים את ה-b-ים.

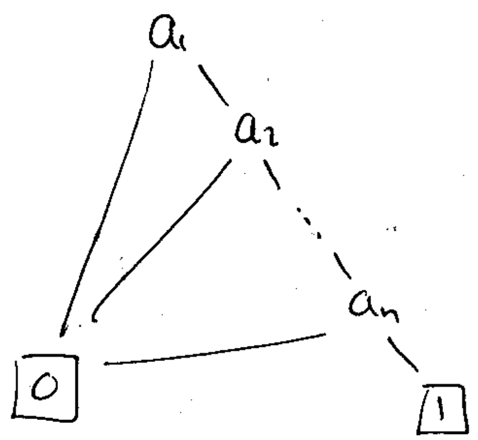
יוצא כאן מבנה כזה:



עם $3\cdot 2^n-1$ קודקודים.

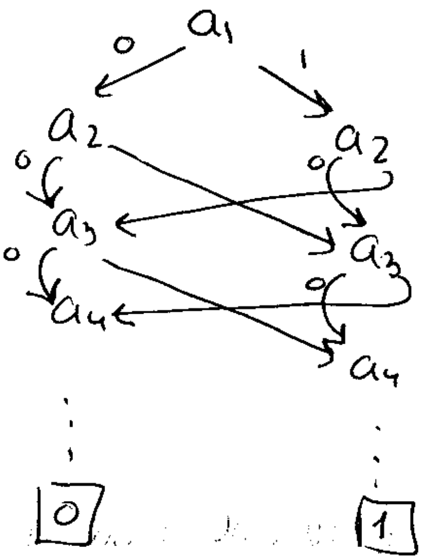
אנחנו לא נראה זאת אבל ידוע שהבעיה של מציאת סדר אופטימלי למשתנים היא PSPACE-שלמה.

דוגמה לפונקציה עם BDD ליניארי לכל סידור של המשתנים:
דוגמה ממש טריוויאלית: $a_1 \wedge a_2 \wedge \dots \wedge a_n$. ראינו יוצא

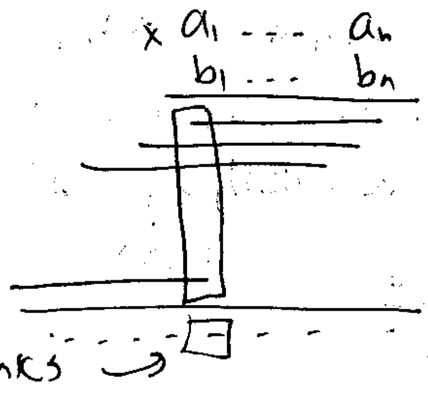


דוגמה קצת פחות טריוויאלית: $a_1 \oplus \dots \oplus a_n$ XOR
שבודקת זוגיות:

פה תמיד יש
$2n+1$
מצבים.



דוגמה לפונקציה (רעה) עם BDD מעריכי לכל סידור משתנים:
יש $a_1 \dots a_n$, $b_1 \dots b_n$ מספרים בינאריים
והפונקציה מחזירה את הביט האמצעי של המכפלה.



כאן התשובה ממש תלויה בסדר המשתנים.

פעולות על BDD

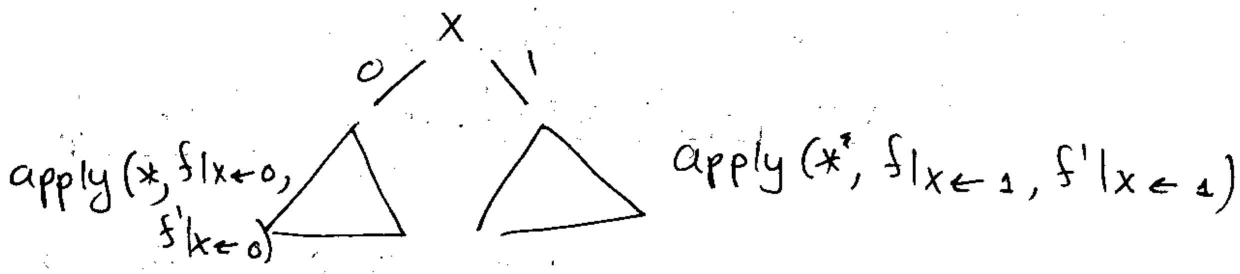
* $apply(*, f)$ - מחזירה BDD עבור $*f$
(ממש שלנו BDD ל-f ורוצים BDD ל-$\neg f$).
פשוט מפעילים את $*$ על העלים $\boxed{0}$, $\boxed{1}$ של ה-BDD של f.

* $apply(*, f, f')$ - מחזירה BDD עבור $f * f'$
כאשר f, f' מעל $x_1, \dots, x_n$ עם אותו סידור
$x_1 < \dots < x_n$.

הרעיון: Shanon expansion:

$$f = (\neg x \wedge f|_{x \leftarrow 0}) \vee (x \wedge f|_{x \leftarrow 1})$$

זה יוצר הגדרה רקורסיבית שכזו:



סימונים: צומת v ב-BDD יכולה להיות עלה ואז יש לה $val(v) \in \{0,1\}$ או שהיא צומת פנימית ואז יש לה
- $var(v)$ (המשתנה של v)
- $low(v) \in V$ הבן ה-0 של v
- $high(v) \in V$ הבן ה-1 של v.

$$f|_{x_i \leftarrow b} = f(x_1, \dots, x_{i-1}, b, x_{i+1}, \dots, x_n)$$

מימוש: הקשתות שנכנסות ל-v רק ש-$var(v) = x_i$, מכוונות במקום הצמתים הקודם ל-$low(v)$ אם $b=0$ וה-$high(v)$ אם $b=1$. אחרי זה צריך לצמצם את ה-BDD.

עכשיו יש לנו את הכלים לממש את $apply(*, f, f')$.

נניח ש- v,v' השורשים של f, f' בהתאמה ונסמן

$var(v) = x$

$var(v') = x'$

(לא בהכרח $x = x'$ כי ייתכן להיות ש-x לא משתתף בהגדרה של f'.)

① אם v ו-v' עלים אז נחזיר $val(v) * val(v')$

② אם v עלה ו-v' אינו עלה, החלף כל עלה של f' ב- ערך $val(v) *$ ערך העלה, כלומר

$\boxed{0} \longrightarrow val(v) * 0$

$\boxed{1} \longrightarrow val(v) * 1$

③ אם v לא עלה ו-v' עלה זה סימטרי ל-②

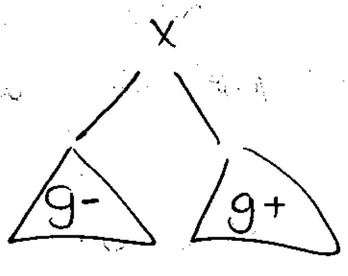
④ אחרת.

• אם $x = x'$,

$$(\neg x \wedge \underbrace{apply(*, f|_{x\leftarrow 0}, f'|_{x\leftarrow 0})}_{g^-}) \vee (x \wedge \underbrace{apply(*, f|_{x\leftarrow 1}, f'|_{x\leftarrow 1})}_{g^+})$$

כלומר ה- BDD הינו

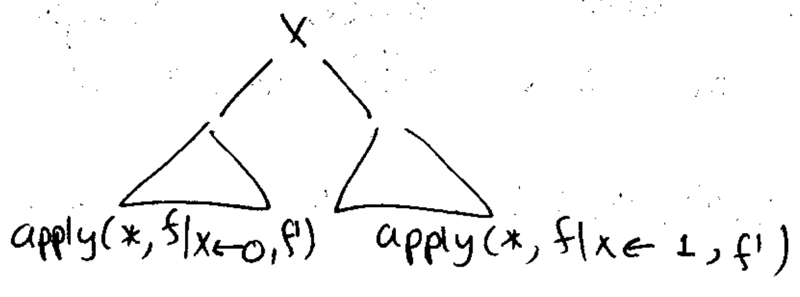
וזה לגמרי.



והרקורסיה הזאת מתכנסת כי מספר המשתנים ב-g-ים קטן יותר.

• אם $x \neq x'$, אם $x < x'$ אז f' לא תלוי ב-x ולכן נחזיר

וזה לגמרי



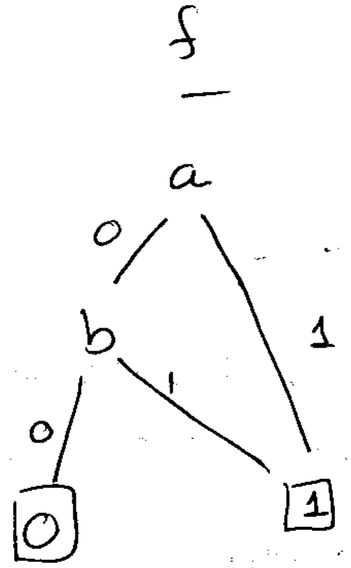
(26)

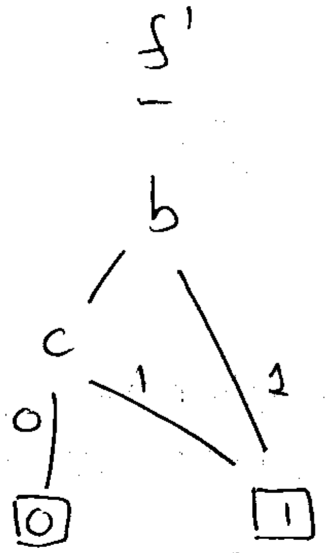
דוגמה:

$f = a \vee b$

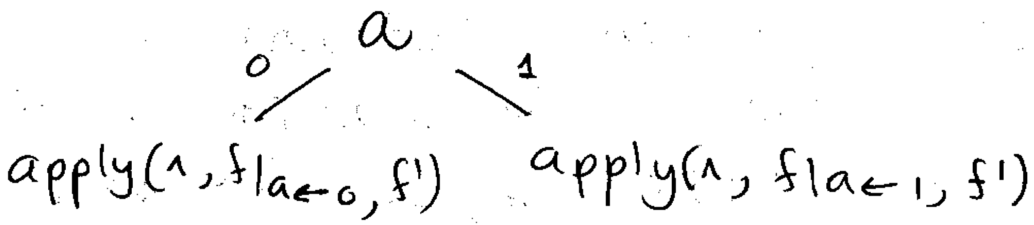
$f' = c \vee b$ $\qquad a < b < c$

$* = \wedge$

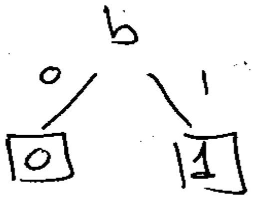




נפצל לפי השלבים שלנו

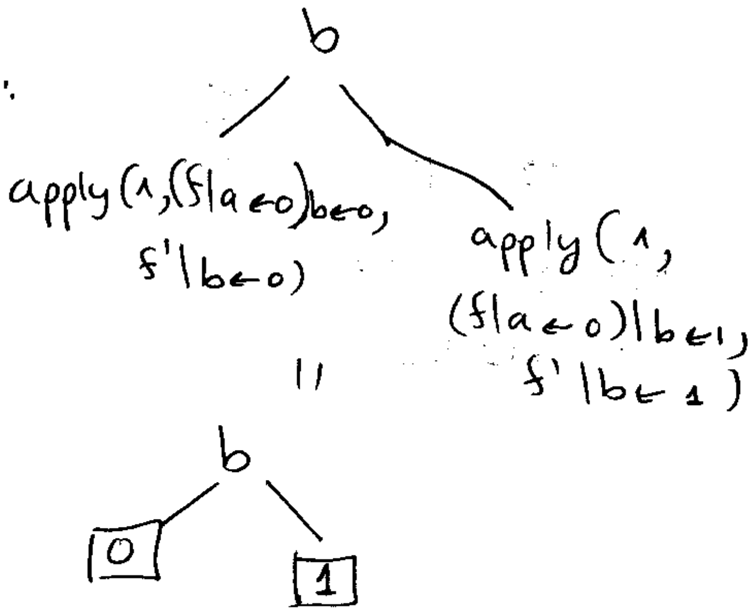


$f|a\leftarrow 0$:



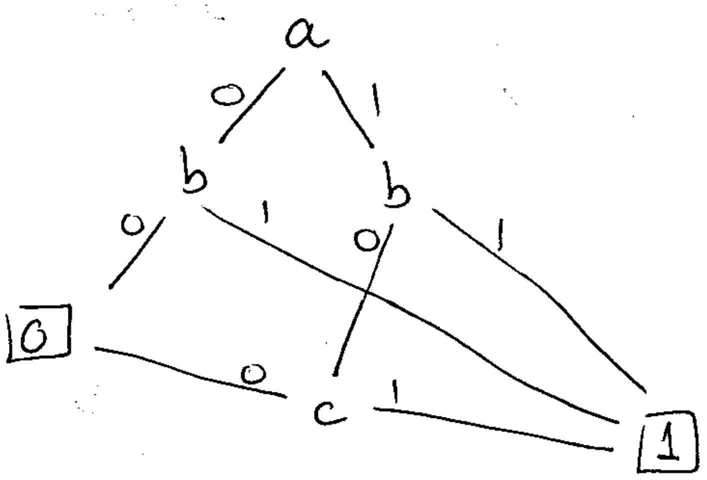
$\Rightarrow$

$\Rightarrow$ apply$(\wedge, f|a\leftarrow 0, f')$:



וכן הלאה...

בסוף מקבלים

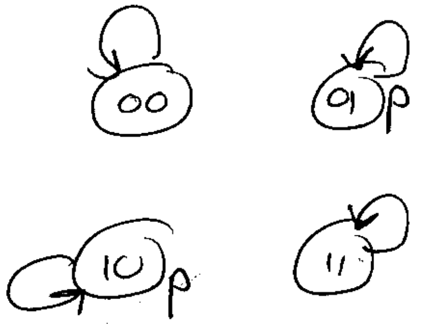


האלגוריתם הזה נרוויח כי לא צריך בהתחלה לבנות עץ גדול ואז לצמצם. אנחנו כל הזמן זורקים עצים קטנים יותר.

בדיקת מודל סימבולית - CTL

יש גרף $G=\langle W,R\rangle$. ויש משתנים שמקודדים את מרחב המצבים. אז תת קבוצה של המצבים אפשר לייצג ע"י פונקציה בוליאנית אבל יש להן ייצוג ב- BDD.

הצלעות $R \subseteq W \times W$. אז נשתמש בפונקציה מעל $x_1,\dots,x_n\ x'_1,\dots,x'_n$ כאשר $x_1,\dots,x_n$ המשתנים של מרחב המצבים. באמצעות פונקציה כזו אפשר לייצג צלעות. למשל



$f: x_1 \leftrightarrow x'_1 \wedge x_2 \leftrightarrow x'_2$

ההשמות היחידות שמספקות את f הן

x_1	x_2	x'_1	x'_2
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

$\|p\| = \{01, 10\} =$

$= x_1 \oplus x_2$

אם יש צלע מ- (x_1,x_2) ל- (x'_1,x'_2) אמ"מ $x_1=x'_1$ וגם $x_2=x'_2$

פעולה בסיסית

- נתונה קבוצה $S \subseteq W$ של מצבים (BDD f_S מעל $x_1 \dots x_n$)
- נתונים מעברים ($R \subseteq W \times W$) ע"י BDD f_R מעל ($x_1 \dots x_n, x'_1 \dots x'_n$)

רוצים BDD עבור $EX\,S$ (מצבים שמהם יש מעבר למצב ב-S). (כמובן רוצים שהסיבוכיות תהיה פולינומיאלית בייצוג הסמבולי של f ו-R)

סימון: בהינתן BDD f (מעל $x_1 \dots x_n$) יש BDD עבור $\exists x_i f$ (שהיא פונקציה מעל $x_1,\dots,x_{i-1},x_{i+1},\dots,x_n$)

$$\exists x_i f(x_1 \dots x_n) = f(x_1,\dots,x_{i-1},0,x_{i+1},\dots,x_n) \vee f(x_1,\dots,x_{i-1},1,x_{i+1},\dots,x_n)$$

כלומר ה-BDD ל-$\exists x_i f$ מתקבל ע"י

$$apply(\vee, f|x_i \leftarrow 0, f|x_i \leftarrow 1)$$

באופן דומה, ה-BDD ל-$\forall x_i f$ מתקבל ע"י

$$apply(\wedge, f|x_i \leftarrow 0, f|x_i \leftarrow 1)$$

ומה הקשר של זה לגרפים?

$w \in \|EX\,S\|$ אמ"מ $\exists w' : (R(w,w') \wedge S(w'))$

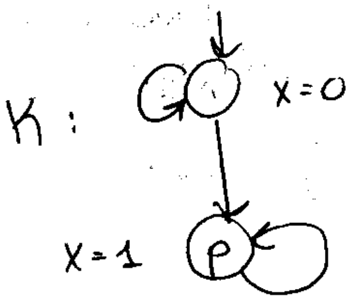
אז ברור אם יש לנו BDD ל-S ו-BDD ל-R אז אפשר לחשב את זה:

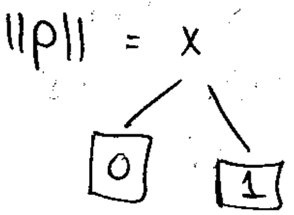
① יהי f'_S עותק של f_S שמוגדר על $x'_1,\dots,x'_n$. נשים לב שניתן להתייחס ל-f'_S כאל BDD מעל $x_1 \dots x_n x'_1 \dots x'_n$ שאינו תלוי ב-$x_1 \dots x_n$. אז נקבל למעשה פרדיקט מעל $x_1 \dots x_n x'_1 \dots x'_n$ שמתקיים בזוגות $\langle w,w' \rangle$ רק ש-$w' \in S$.

② נעשה $g = apply(\wedge, f'_S, f_R)$ ונקבל פרדיקט שמתקיים ב-$\langle w,w' \rangle$ אם $R(w,w')$ וגם $w' \in S$.

③ אותנו מעניין $\exists w' g$ כלומר $\exists x'_1 \dots \exists x'_n g$ ואת זה אנחנו יודעים לחשב.

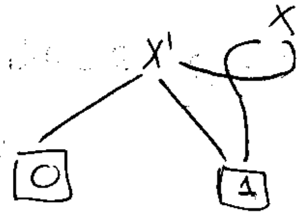
דוגמה:





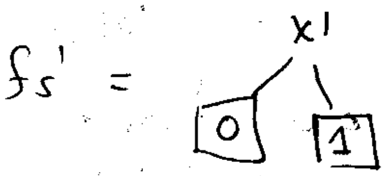
$f_R =$

x	x'	f_R
0	0	1
0	1	1
1	0	0
1	1	1

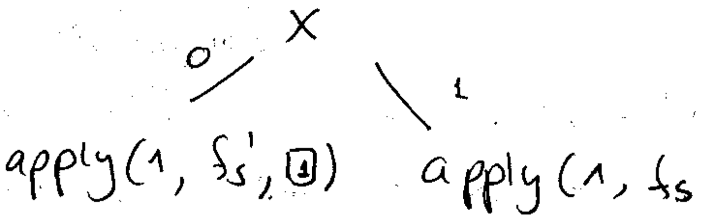


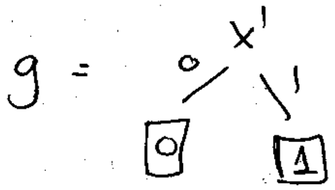
נעשה את EX_p

① נהפוך את ה- BDD של f_s להיות מעל x':



② נחשב $apply(\wedge, f_s', f_R)$



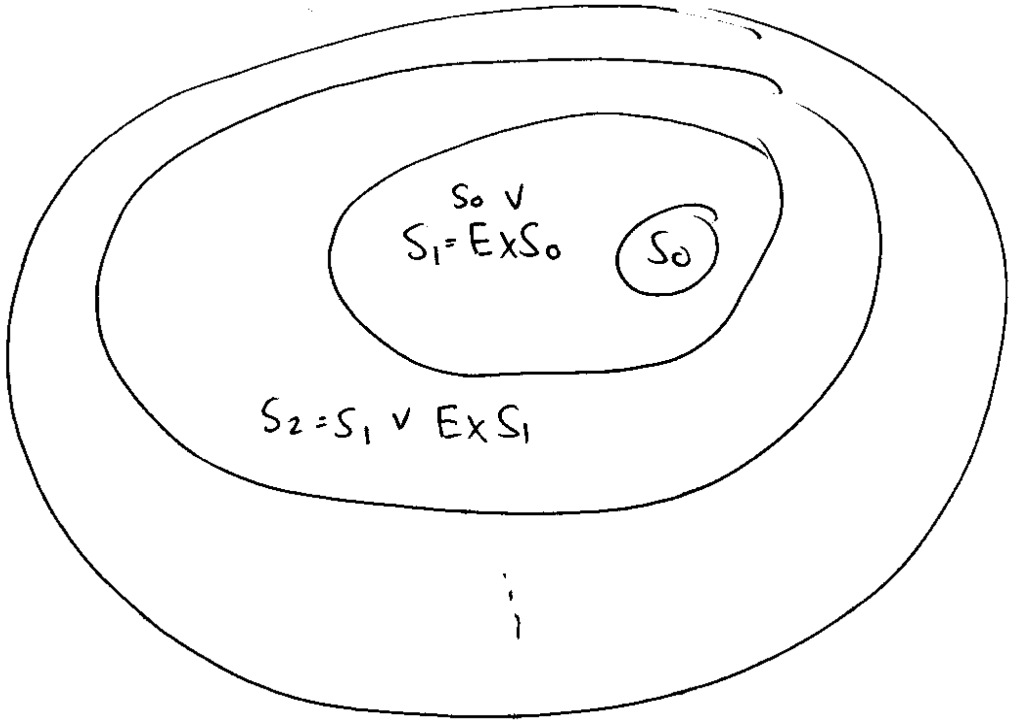


$\exists x' g = \boxed{1}$

$\Rightarrow ||EX_p|| = \boxed{1}$

<u>ישימויות</u>: יש BDD f_s עבור $S \subseteq W$
ונתון BDD f_R עבור R.

רוצים BDD שמורכב הצמתים ש-S ישיג מהם.



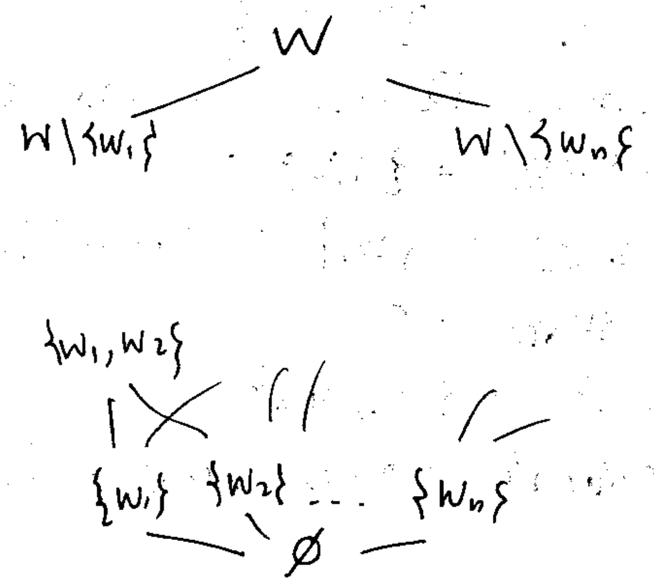
ממשיכים עד ה- fix-point - עד שהתהליך מפסיק לגדול...

(29) 10/12/09
אימות פורמלי

ראינו BDD-ים. כדי לייצג פונק' בוליאניות אבל אפשר
גם לייצג איתם תת קבוצות של מצבים.
ראינו שאם יש מבנה קריפקה אפשר לייצג אותו ע"י משתנים.
בהינתן BDD f_S של $S\subseteq W$, BDD f_R
של $R\subseteq W\times W$ אפשר לייצר BDD של G המצבים
שמקיימים $EX\,S$ (בפעולות פרופוזיציוניות לגבי) ה-BDD
וזה טוב כי גודל ה-BDD בד"כ אקספוננציאלית קטן יותר מהייצוג
הטריוויאלי).

לוגיקה μ-calculus

נניח שיש לנו מבנה קריפקה $K=\langle AP, W, R, W_0, L\rangle$
אז היחס $\subseteq$ על 2^W הוא יחס סדר חלקי:



($\subseteq$ הוא יחס סדר אבל לא כל שני איברים ניתנים להשוואה.)
כל $S\subseteq W$ ניתן לחשוב כעל פרדיקט מעל W.
predicate transformer הוא פונקציה $\tau: 2^W \to 2^W$ -
הוא לוקחת פרדיקט אחד ומחזירה פרדיקט אחר.
למשל $\tau(S)=\{w: \exists w'\ R(w,w'),\ w'\in S\}$
זה מחזיר את המצבים שיש מהם מעבר ל-S.

עוד פונקציה מיוחדת שקוראים לה : $\exists pre$

$$\exists pre(S) = \{w : \exists w' \ R(w,w'),\ w' \in S\}$$

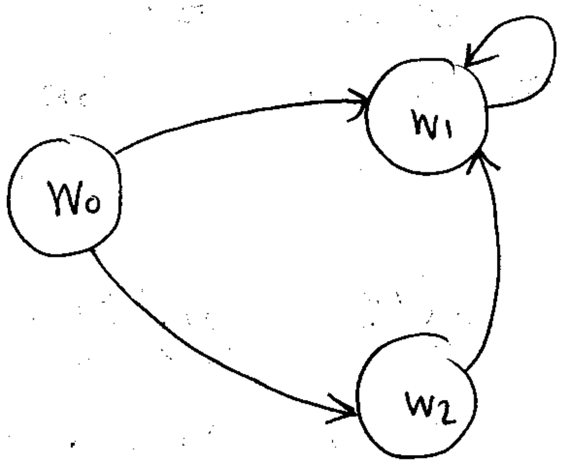
ובאופן דומה מגדירים

$$\forall pre(S) = \{w : \forall w' \ R(w,w') \rightarrow w' \in S\}$$

$$\exists post(S) = \{w : \exists w' \ R(w',w),\ w' \in S\}$$

$$\forall post(S) = \{w : \forall w' \ R(w',w) \longrightarrow w' \in S\}$$

דוגמה:



- $\forall pre(\{w_2\}) = \{\} = \emptyset$

  כי אין אף אחד שכל הבנים שלו הולכים ל- w_2.
- $\exists pre(\{w_2\}) = \{w_0\}$
- $\exists post(\{w_2\}) = \{w_1\} =$
- $\forall post(\{w_2\}) = \emptyset$
- $\forall pre(\{w_1\}) = \{w_1, w_2\}$
- $\forall pre(\{w_1, w_2\}) = \{w_0, w_1, w_2\} = W$

תכונות של Predicate Transformers

<u>הגדרה:</u> נאמר ש- $\tau : 2^W \to 2^W$ <u>מונוטונית</u> אם לכל S_1, S_2 אם $S_1 \subseteq S_2$ אז $\tau(S_1) \subseteq \tau(S_2)$.

קל לראות שכל הפונקציות שהגדרנו למעלה הן מונוטוניות כי ככל ש- S גדולה יותר, קל יותר לקיים את התנאי.

דוגמה לפונקציה לא מונוטונית: $\tau(S) = W \setminus S$
זאת מונוטונית יורדת אבל אפשר גם לתת דוגמאות לפונקציות שהן לא עולות ולא יורדות.

<u>הגדרה</u>: τ היא <u>$\bigcup$-רציפה</u> אם לכל סדרה ב-W, $P_1 \subseteq P_2 \subseteq \dots$
מתקיים $\tau(\bigcup_i P_i) = \bigcup_i \tau(P_i)$
τ היא <u>$\bigcap$-רציפה</u> אם לכל סדרה ב-W, $P_1 \supseteq P_2 \supseteq \dots$
מתקיים $\tau(\bigcap_i P_i) = \bigcap_i \tau(P_i)$

<u>דוגמאות</u>:

- ה-6 pre/post $\forall|\exists$ שהגדרנו הן גם $\bigcap$ וגם $\bigcup$-רציפות. (אפילו עבור W אינסופית).
- דוגמה לפונק' לא $\bigcup$-רציפה. נסמן $W = \{w_0, w_1, \dots\}$ אינסופית ונגדיר

$$\tau(S) = \begin{cases} S & \text{אינסופית } S \\ \emptyset & \text{סופית } S \end{cases}$$

נתבונן בסדרה $\{w_0\}, \{w_0, w_1\}, \{w_0, w_1, w_2\}, \dots$
כל איבר בסדרה הוא סופי ולכן $\bigcup_i \tau(P_i) = \bigcup \emptyset = \emptyset$
אבל $\tau(\bigcup_i P_i) = \tau(W) = W$ כי W אינסופית.

- $\tau(S) = W \setminus S$ אינה $\bigcup$-רציפה. נתבונן בסדרה $\emptyset \subseteq W \dots W$:
אז $\tau(\bigcup_i P_i) = \tau(W) = \emptyset$ אבל
$\bigcup_i \tau(P_i) = W \cup \emptyset = W$
פונקציית ההשמה היא גם לא חייבת $\bigcup$-רציפה.
- דוגמה לפונק' לא $\bigcap$-רציפה: $W = [0,1]$ (קטע סגור)
ונתבונן ב- $[0,1] \supseteq [0,\frac{1}{2}] \supseteq [0,\frac{1}{4}] \supseteq \dots$

$$\tau([0,i]) = \begin{cases} [0,1] & i \neq 0 \\ [0,0] & i = 0 \end{cases}$$

$\bigcap P_i = \{0\}$
$\Rightarrow \tau(\bigcap P_i) = \{0\}$
$\bigcap \tau(P_i) = [0,1]$ #

משפט: אם W סופית ו-τ מונוטונית, אז τ היא $\cup$-רציפה ו-$\cap$-רציפה.

הוכחה:

נתבונן בסדרה ב-W: $p_1 \subseteq p_2 \subseteq \dots$

W סופית ולכן בהכרח קיים אינדקס שהחל ממנו

$$p_n = p_{n+1} = p_{n+2} = \dots$$

$$\Leftarrow \quad \cup p_i = p_n$$

$$\tau(\cup p_i) = \tau(p_n)$$

מהמונוטוניות $\tau(p_1) \subseteq \tau(p_2) \subseteq \dots$

$$\tau(p_n) = \tau(p_{n+1}) = \tau(p_{n+2}) = \dots$$

$$\Leftarrow \quad \cup \tau(p_i) = \tau(p_n)$$

$$\Leftarrow \quad \tau(\cup p_i) = \cup \tau(p_i)$$

ו-τ $\cup$-רציפה.

ההוכחה של $\cap$-רציפות דומה. ☺

סימון: עבור $0 \le i$ מגדירים $\tau^i(S)$ באינדוקציה:

$$\tau^0(S) = S$$

$$\tau^{i+1}(S) = \tau(\tau^i(S))$$

הגדרה: נקודת שבת (fixed point) של τ היא $S \subseteq W$ כך ש- $\tau(S) = S$.

דוגמה: נניח ש-$T \subseteq W$.

$$\tau(S) = T \cup \exists pre(S)$$

כלומר את שהאיברים ב-T או שקיים להם מן ב-S.

$$\tau(W) = W$$

כי R טוטלית (לכל מצב יש לפחות מן אחד) ולכן $\exists pre(W) = W$.

נקודת שבת יותר מעניינת היא קבוצת המצבים ש-T ושיש מהם (כלומר קיים מסלול מהם ל-T). נסמנה ב-P:

רוצים להראות ש- $\tau(P) = P$. אכן, המצבים ב-P הם או ב-T או שיש להם בן שאפשר להגיע ממנו ל-T.

אפשר להוכיח שאין זפוק׳ הנצאת עוד נקודות שבת.

least/greatest fixed point

<u>הגדרה</u>: Z היא נקודת שבת מינימלית של τ אם Z היא נקודת שבת של τ ולכל S' נק' ש- S' נקודת שבת מתקיים ש- $Z \subseteq S'$ (ולמעשה יותר נכון להגיד ש- $S' \not\subset Z$ כי אולי הן לא ניתנות להשוואה אבל אח"כ אנחנו נראה ש- קיימת מינימלית והיא החיתוך של כולן, לכן למעשה יוצא שהיא באמת מוכלת בכל נק' השבת האחרות).

את נק' השבת המינימלית מסמנים $\mu y\, \tau(y)$.

Z היא נקודת שבת מקסימלית של τ אם היא נקודת שבת ולכל נק' שבת S' מתקיים $S' \subseteq Z$. מסמנים $\nu y\, \tau(y)$

<u>משפט</u> (אפיון נקודת שבת) Tarski-Knaster

אם τ מונוטונית

(1) $\mu y\, \tau(y) = \bigcap_{\tau(S)=S} S$

(2) $\nu y\, \tau(y) = \bigcup_{\tau(S)=S} S$

זה משפט מאוד כבד, אנחנו לא הולכים להוכיח אותו.

<u>למה 1</u>: אם τ מונוטונית אז לכל i $\tau^i(\emptyset) \subseteq \tau^{i+1}(\emptyset)$
$\tau^i(W) \supseteq \tau^{i+1}(W)$

<u>הוכחה</u>: באינדוקציה על i:

עבור $i=0$, $\tau^0(\emptyset) = \emptyset \subseteq \tau^1(\emptyset)$

כי $\emptyset$ מוכל בכל קבוצה.

$\tau^0(W) = W \supseteq \tau^1(W)$

כי הכל ב-W.

נניח עבור i ונוכיח ל- $i+1$:

$$\tau^{i+1}(\emptyset) = \tau(\tau^i(\emptyset)) \subseteq \tau(\tau^{i+1}(\emptyset)) = \tau^{i+2}(\emptyset)$$

מונוטוניות וה"א

$$\tau^{i+1}(W) = \tau(\tau^i(W)) \supseteq \tau(\tau^{i+1}(W)) = \tau^{i+2}(W)$$

הנ׳ ומונוטוניות

☺

דוגמה: $\tau(S) = T \vee \exists pre(S)$

$\tau(\emptyset) = T$

$\tau(T) = T \vee EXT$ ורק הלאה

למה 2: אם τ מונוטונית ו-W סופית אז יש $j \geq 0$ כך שלכל $j \leq k$ $\tau^k(\emptyset) = \tau^j(\emptyset)$!

$$\mu y \tau(y) = \tau^j(\emptyset)$$

כאמור מהתנאים הנ"ל כדי למצוא נק' שבת מינימלית אפשר להפעיל את τ על $\emptyset$ שוב ושוב עד שנתקעים במקום.

למה 3: אם τ מונוטונית ו-W סופית אז יש $j \geq 0$ כך שלכל $j \leq k$ $\tau^k(W) = \tau^j(W)$!

$$\nu y \tau(y) = \tau^j(W)$$

הערה: ה-j משתי הלמות הוא ממש לא מהכרח אותו j....

⇐ אלגוריתם לחישוב נקודות שבת:

```
proc lfp (τ : predicate transformer)
    Q = ∅
    Q' = τ(Q)
    while Q' ≠ Q
        Q = Q'
        Q' = τ(Q')
    return Q
```

דוגמה:

$$\tau(S) = p \cup \exists pre(S)$$
$$\Rightarrow \mu y\, \tau(y) = EFp$$

$$\tau(S) = p \cap \exists pre(S)$$
$$\Rightarrow \nu y\, \tau(y) = EGp$$

$$\tau(S) = p \cup \forall pre(S)$$
$$\Rightarrow \mu y\, \tau(y) = AFp$$

$$\tau(S) = p \cap \forall pre(S)$$
$$\Rightarrow \nu y\, \tau(y) = AGp$$

$$\tau(S) = q \cup (p \cap \exists pre(S))$$
$$\Rightarrow \mu y\, \tau(y) = EpUq$$

איך מגיעים לזה? "(פרום)"

$$EpUq \rightarrow q \vee (p \vee EX(q \vee (p \wedge EX(\ldots$$

$$EpUq = q \vee (p \wedge EX(EpUq))$$

זה האפיון fixed point

(33) 24|12|09

אימות פורמלי

הלוגיקה μ-calculus

מה יש בה?

- AP - atomic propositions
- משתנים V
- נוסחאות:
  - true/false הן נוסחאות
  - $p \in AP$ נוסחה
  - $v \in V$ נוסחה
  - אם φ_1, φ_2 נוסחאות אז גם $EX\varphi_1$, $\varphi_1 \vee \varphi_2$, $\neg\varphi_1$
  - אם $\varphi(y)$ נוסחה עם משתנה חופשי חיובי y אז $\mu y.\varphi(y)$ נוסחה.

y משתנה חופשי ב- $\varphi(y)$ אם הוא לא בטווח של כמת μ או ν. הוא גם חיובי אם הוא תחת מספר זוגי של שלילות ("הנוסחה מונוטונית ב-y").

קיצורים:

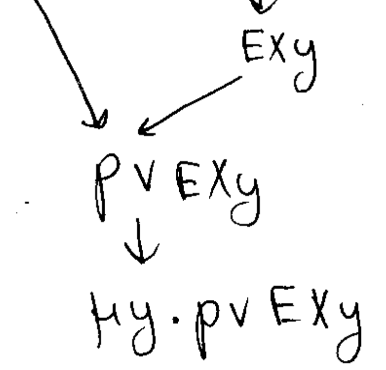
$$AX\varphi \equiv \neg EX \neg \varphi$$

$$\nu y.\varphi(y) \equiv \neg(\mu y.\neg\varphi(\neg y))$$

דוגמאות:

(1) $\mu y.p \vee EXy$

הוכחה:



סמנטיקה ביחס למבנה קריפקה K והשמה למשתנים החופשיים. אי אפשר לדעת אם נוסחה נכונה או לא בלי שיודעים מה לשים במשתנים. כמו שאי אפשר לדעת מה ערך האמת של "$x>5$" בלי לדעת את הערך של x.

השמה למשתנים מסומנים כך:

$$\mathcal{V} = \{\langle y_1, W_1 \rangle, \dots, \langle y_k, W_k \rangle\} \quad . \quad W_i \in W$$

עבור השמה $\mathcal{V}$ נסמן

$$\varphi^K(\mathcal{V}) = \{ w : w \text{ מספק את } \varphi \text{ תחת ההשמה } \mathcal{V} \}$$

נגדיר רקורסיבית:

$$true(\mathcal{V}) = W$$ – כל המצבים מספקים את true.

$$false(\mathcal{V}) = \emptyset$$

$$p(\mathcal{V}) = \|p\| = \{w : p \in L(w)\}$$

$$y(\mathcal{V}) = W_i : \langle y, W_i \rangle \in \mathcal{V}$$

$$\neg\varphi_1(\mathcal{V}) = W \setminus \varphi_1(\mathcal{V})$$

$$(\varphi_1 \vee \varphi_2)(\mathcal{V}) = \varphi_1(\mathcal{V}) \cup \varphi_2(\mathcal{V})$$

$$EX\varphi_1(\mathcal{V}) = \exists pre(\varphi_1(\mathcal{V}))$$

$$\mu y.\varphi_1(y)(\mathcal{V}) = \bigcap\{S \subseteq W : \varphi_1(\mathcal{V}[y \leftarrow S]) = S\}$$

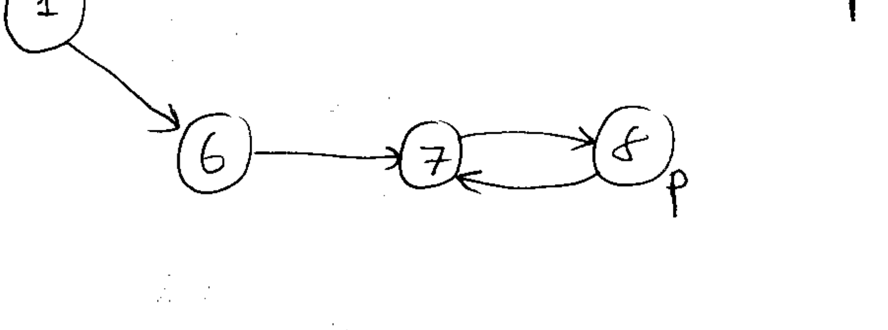
$$\left(= \bigcup_i \varphi^i(false)\right)$$ סמנטיקה שקולה

$\{\langle y, \emptyset \rangle\}$ "

ההשמה המתקבלת מ- $\mathcal{V}$ ע"י הוספת ההשמה $\langle y, S \rangle$ והשמטת השמות קודמות ל- y (אם היו כאלה).

דוגמה:

K:



נמצא את $\mu y.\ p \vee EXEX\,y$ (נסמן $\varphi(y) = p \vee EXEX\,y$ ונפעל ע"י האלגוריתם $(\varphi^i(\emptyset)$ כדי למצוא נקודת שבת.

$$\varphi(false) = \| p \vee EXEX\, false \| = \|p\| = \{5,8\}$$
$$\varphi(\{5,8\}) = \| p \vee EXEX\{5,8\} \| = \{3,6,5,8\}$$
$$\varphi(\{3,6,5,8\}) = \{1,3,6,5,8\}$$
$$\varphi(\{1,3,6,5,8\}) = \{1,3,6,5,8\}$$

$\Leftarrow$ $\{1,3,6,5,8\}$ היא נקודת השבת. אפשר לראות שזה מסתדר עם הסמנטיקה

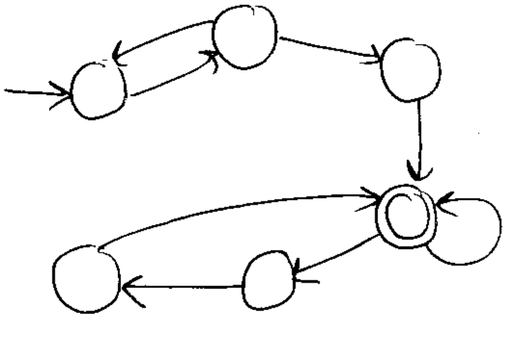
$$\mu y.\ \varphi(y)(\vartheta) = \cap\{S \subseteq W : \varphi(\vartheta[y \leftarrow S]) = S\}$$

אנחנו יודעים שאין נוסחה ב- CTL^* עבור "p נכון בכל המקומות הזוגיים".
נמצא את זה ב- μ-calculus. אם אפשר היה לעשות נוסחה אינסופית היינו כותבים

$$p \wedge AXAX(p \wedge AXAX(p \wedge \dots))$$

שזה כמו $\nu y\ p \wedge AXAX\,y$.
אין פה טריק אינטואיטיבי. צריך פשוט לראות הרבה דוגמאות.

הנה משהו שימושי: נוסחה לבדיקת ריקנות של אוטומט בוקי: אנחנו צריכים מצב מקבל ישיג שהוא גם ישיג מעצמו.



טומר צריך נוסחת μ-calc עבור $EGF\alpha$.
נזכור שבשביל לבדוק אם $K \models \varphi$ עבור נוסחת LTL φ השתמשנו ב- $L(A_{\neg\varphi}) = \{w : w \not\models \varphi\}$
וזה טוב לנו כי $L(K) \cap L(A_{\neg\varphi}) = \emptyset \Leftrightarrow K \models \varphi$
במקום של חיתוך עושים "בניית" אוטומט המכפלה. אז אנחנו שואלים האם $L(K \times A_{\neg\varphi})$ היא ריקה?

ומה האוטומט הזה צריך לשאול? אם יש במבנה האוטומט מסלול שמבקר ∞ פעמים ב- α וזה מתאים לנוסחה $EGF\alpha$.

נסתכל קודם על מקרה פשוט יותר. נניח שאנחנו רוצים $EGEFp$ (שהיא לא שקולה ל- $EGFp$).
כי נוסחת LTL אנחנו יודעים להתמודד איתה.

$$EG\underbrace{EFp}_{q} \longrightarrow \nu y. q \wedge EXy$$

$$EFq \longrightarrow \mu z. p \vee EXz$$

$$\Rightarrow EGEFp \longrightarrow \nu y.(\mu z. p \vee EXz) \wedge EXy$$

אלגוריתם סימבולי ל- $EGFp$
ראשית, ישנו אלגוריתם סימבולי לישיגות רגילה:

$$EFp \equiv \mu y. p \vee EXy$$

$$\exists pre^*(\|p\|) \begin{cases} do: & S_0 = \|p\| \\ & S_{i+1} = S_i \cup \exists pre(S_i) \\ until: & S_{i+1} = S_i \end{cases}$$

עכשיו נראה אלגוריתם סימבולי לישיגות לא ריקה, כלומר רוצים מסלולים שהם לא ריקים $EFEXp$
אז מי שמקיים את p הוא לא אוטומטית מפנים.

$$EFEXp = \mu y (EXp) \vee (EXy) =$$
$$= \mu y\, EX(p \vee y)$$

אז האלג' הוא:

$$\exists pre^+(\|p\|) \begin{cases} do: & S_0 = \exists pre(\|p\|) \\ & S_{i+1} = S_i \cup \exists pre(S_i) \\ until: & S_{i+1} = S_i \end{cases}$$

בעזרת שני אלה נפתור את הבעיה שלנו.

$S_0 \rightsquigarrow p$ נזכיר $S_0 = \exists pre^+(\|p\|)$

$S_1 \rightsquigarrow p \rightsquigarrow p$ $S_1 = \exists pre^+(\|p\| \cap S_0)$

כי הישיגות לא ריקה

אלה המסלולים שרואים את p פעמיים (אולי זה אותו ה-p אבל מגיעים אליו פעמיים שונות).

$S_2 = \exists pre^+(\|p\| \cap S_1)$ כאן רואים את p 3 פעמים.

⋮

$S_{i+1} = \exists pre^+(\|p\| \cap S_i)$. כאן רואים את p $i+2$ פעמים

ואת זה עושים עד ש- $S := S_i = S_{i+1}$ וזו נק' השבת שלנו.

$$S = \exists pre^+(\|p\| \cap S) = \|EGFp\|$$

מה הסיבוכיות של האלג'? כל איטרציה מורידה לפחות מצב אחד ולכן יש לכל היותר n איטרציות. אבל בכל איטרציה מחשבים את $\exists pre^+$ שהוא עצמו לכל היותר n איטרציות. סה"כ $O(n^2)$. ($n =$ מספר המצבים).

[אם עובדים ישירות על הגרף אז בעזרת MSCC אפשר לעשות את זה בזמן $O(|E|)$. בעיה פתוחה - האם אפשר למצוא אלג' סימבולי לפתרון הבעיה בזמן יותר טוב? יש ספרות ענפה בנושא - detection bad cycle]

ב- μ-calc הנוסחה שמתקבלת היא

$$EGFp \equiv \nu y.\mu z.(EX(p \wedge y) \vee z)$$

ביסימולציה, סימולציה, אבסטרקציה

עבור LTL ראינו שאם $L(K_1) = L(K_2)$ אז לכל נוסחת LTL φ $K_1 \models \varphi \Leftrightarrow K_2 \models \varphi$

אז אם יש לנו מבנה קריפקי נשאף לעבור למבנה יותר קטן שיש לו אותה שפה.

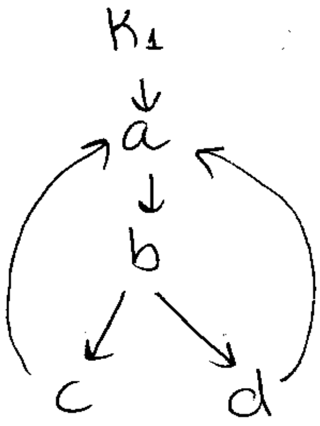
עוד משהו שניתן לעבור הוא שאם $L(K_1) \subseteq L(K_2)$

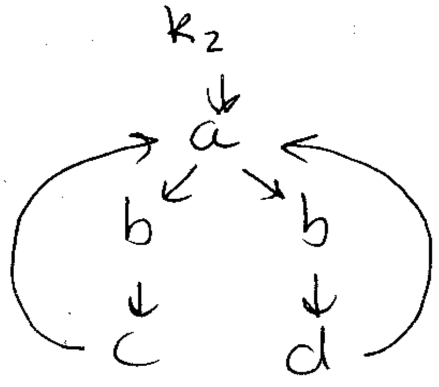
אז לכל נוסחת LTL ‏φ, ‏$K_2 \models \varphi \Rightarrow K_1 \models \varphi$

אז מספיק יהיה לנו לעבור למע' שיש לה יותר התנהגויות

(ונזכיר שיתכן עם זה היא תהיה קטנה יותר שזה טוב).

דוגמה:





ראינו $L(K_1) = L(K_2) = (ab(c+d))^\omega$

אבל יש נוסחת CTL שמפרידה ביניהם:

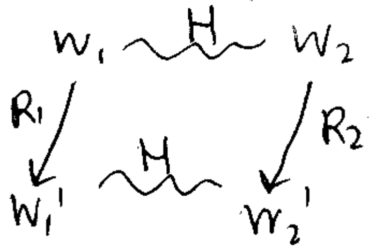
$\varphi = EX(EXc \wedge EXd)$. ‏$K_1 \models \varphi$ אבל $K_2 \not\models \varphi$.

נתונים $K_1 = \langle AP, W_1, R_1, W_1^0, L_1 \rangle$

$K_2 = \langle AP, W_2, R_2, W_2^0, L_2 \rangle$

נגדיר יחס ביסימולציה בין K_1 ו- K_2: זהו יחס $H \subseteq W_1 \times W_2$ המקיים לכל w_1, w_2 כך ש- $H(w_1,w_2)$:

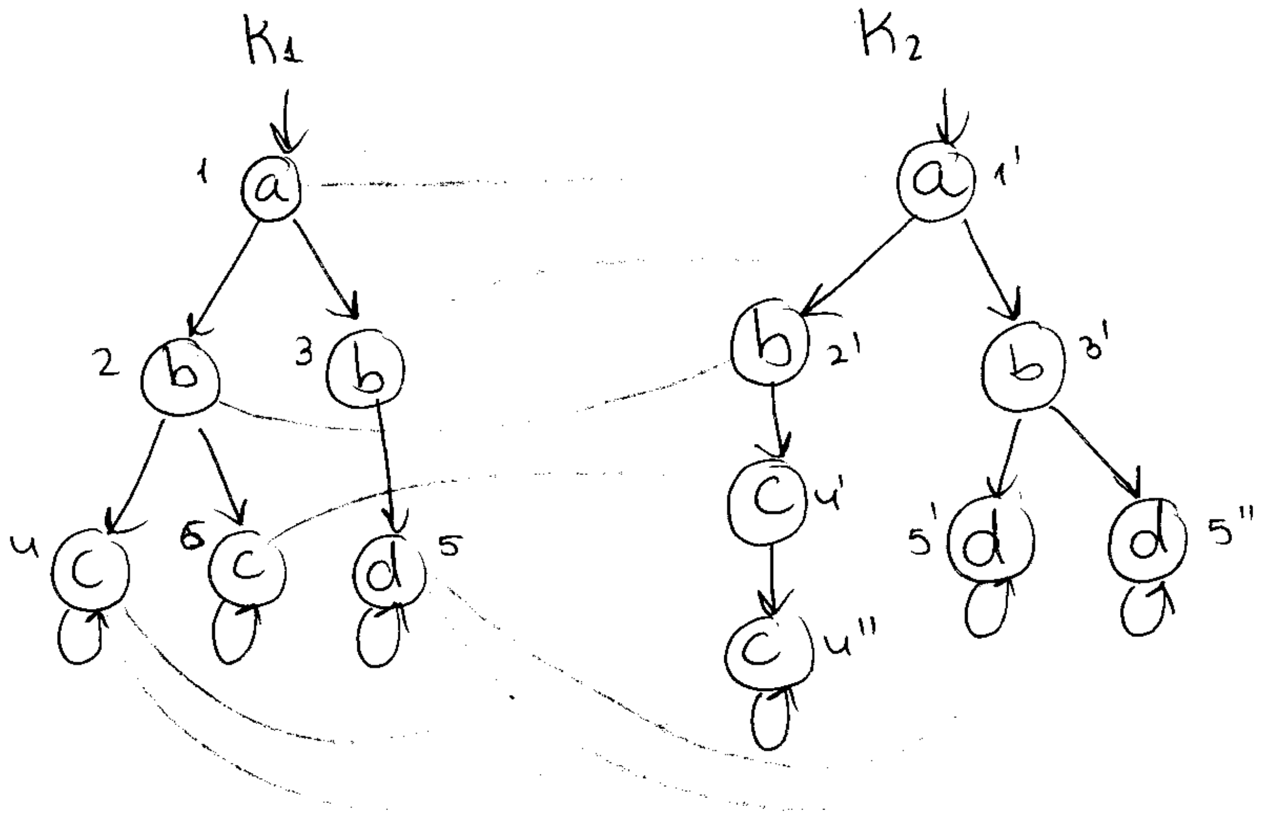
- $L_1(w_1) = L_2(w_2)$ (מסומנים באותו דבר)
- לכל $w_1' \in W_1$ כך ש $R_1(w_1,w_1')$ קיים $w_2' \in W_2$ כך ש- $R_2(w_2,w_2')$ ו- $H(w_1',w_2')$.
- לכל $w_2' \in W_2$ כך ש- $R_2(w_2,w_2')$ קיים $w_1' \in W_1$ כך ש- $R_1(w_1,w_1')$ ו- $H(w_1',w_2')$



נאמר ש- K_1 ו- K_2 הם ביסימולרים זה לזה אם קיים יחס ביסימולציה H בין K_1 ו- K_2 כך ש-

- לכל $w_1 \in W_1^0$ קיים $w_2 \in W_2^0$ כך ש- $H(w_1,w_2)$
- לכל $w_2 \in W_2^0$ קיים $w_1 \in W_1^0$ כך ש- $H(w_1,w_2)$

דוגמה:



זה אכן יחס ביסימולציה. למשל,

$H(2,2')$: $L_1(2) = L_2(2')$

לכל בן של 2 $(4,6)$ קיים בן של $2'$ $(4')$

רק ש- $H(4,4')$ ו- $H(6,4')$

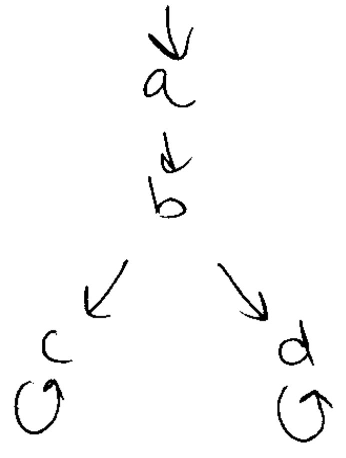
ולכל בן של $2'$ $(4')$ קיים בן של 2 $(4,6)$

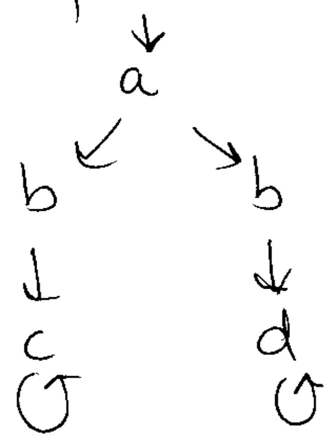
רק שמתקיים מה שדרוש.

מה נשמר תחת ביסימולציה?

- פרימה של אטגלים

דוגמה למצב שבו אין ביסימולציה

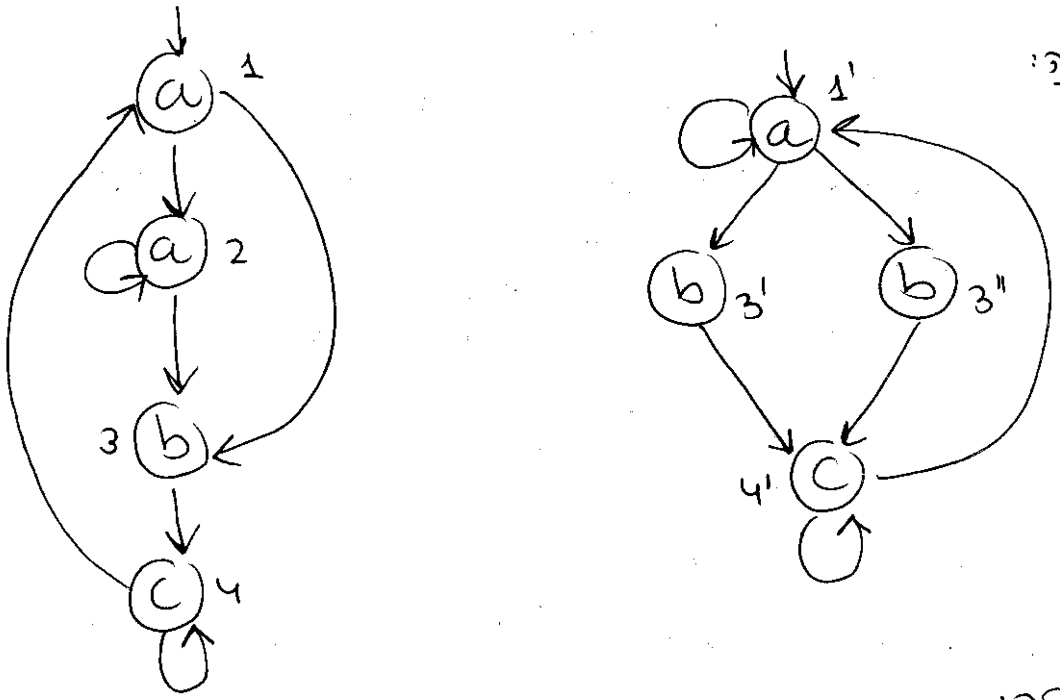




37 31/12/09
אימות פורמלי

התחלנו ללמוד ביסימולציה.

דוגמה:



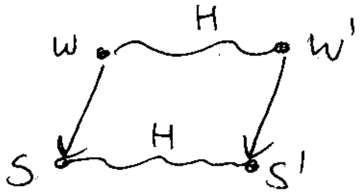
השפה:

$$(a^+bc^+)^\omega +$$
$$+ (a^+bc^+)^*(a^\omega + a^+bc^\omega)$$

השפה של שניהם זהה אבל יש סיכוי שהם ביסימולריים.
אם השפות לא זהות הם בטוח לא ביסימולריים.

האם היחס $H = \{(1,1'), (2,1'), (3,3'), (3,3''), (4,4')\}$ הוא יחס ביסימולציה?

הם מסמנים אותו דבר אז זה בסדר. הוא מקשר כמו שצריך בין המצבים ההתחלתיים ואפשר גם לוודא ששאר התנאים מתקיימים:



איך בודקים אם מבנים הם ביסימולריים?
קודם כל, ברור שזה ב- NP כי מספיק לנו עד H שהוא יחס ביסימולציה. בהינתן יחס H הבדיקה אם זו ביסימולציה זה חישוב קל וכל הבדיקות הן מקומיות.

אם האמת היא שלה גם ב- P.

ביסימולציה H היא יחס $H \subseteq W \times W'$. נניח שיש לנו שתי ביסימולציות H_1, H_2 אז גם $H_1 \cup H_2$ היא ביסימולציה (ק) לוודא את התנאים).

ביסימולציה מקסימלית היא ביסימולציה מקסימלית ביחס להכלה.

אז למעשה $H = \bigcup_{\substack{h \subseteq W \times W' \\ h \text{ bisimulation}}} h$ היא הביסימולציה המקסימלית היחידה.

איך למצוא ביסימולציה מקסימלית בין M ל- M':

$$H_0 = \{ \langle w, w' \rangle : L(w) = L(w') \}$$

(ברור שביסימולציה מקסימלית מוכלת ב- H.)

$$H_{i+1} = H_i \cap \{ \langle w, w' \rangle : \langle w, w' \rangle \text{ is good with respect to } H_i \}$$

נאמר ש- $\langle w, w' \rangle$ טוב ביחס ל- H_i אם

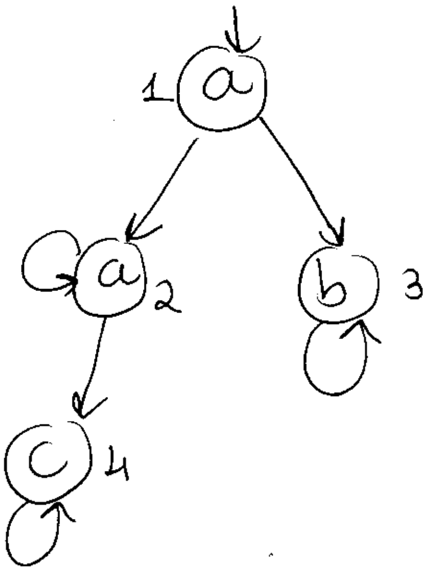
- לכל s כך ש- $R(w,s)$ קיים s' כך ש- $R'(w',s')$ ו- $H_i(s,s')$
- לכל s' כך ש- $R'(w',s')$ קיים s כך ש- $R(w,s)$ ו- $H_i(s,s')$.

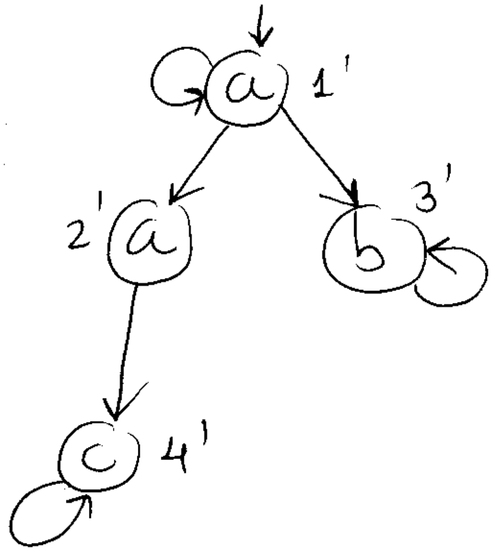
מחשבים את החיתוכים עד שמגיעים לנקודת שבת H^*.

לפני שנראה נכונות, למה זה פולינומיאלי?

יש לכל היותר ח·ח' איטרציות כי בכל שלב נוריד לפחות איבר אחד בחיתוך. כל איטרציה גם היא פולינומיאלית (למעשה זה ריבועי בקשתות אבל זה פחות קריטי לנו).

דוגמה:





קל לראות שיש שוויון שפות.
נתחיל להפעיל את האלגוריתם:

$H_0 = \{\langle 1,1' \rangle, \langle 2,1' \rangle, \langle 1,2' \rangle, \langle 2,2' \rangle, \langle 3,3' \rangle, \langle 4,4' \rangle\}$

$H_1 = \{\langle 3,3' \rangle, \langle 4,4' \rangle, \langle 1,1' \rangle\}$

$H_2 = \{\langle 3,3' \rangle, \langle 4,4' \rangle\} = H_3$

אכן המבנים לא ביסימולריים כי אין לנו התאמה למצבים ההתחלתיים.

הוכחת נכונות: צריך להראות ש- H^* ביסימולציה ושהיא מקסימלית.

① H^* ביסימולציה:

$H^*(w,w') \Rightarrow H_0(w,w') \Rightarrow L(w)=L(w')$.

$H^*(w,w')$ - לכל s, $R(w,s)$ יש s' כך ש- $R'(w',s')$ ו- $H^*(s,s')$ כי כל צעד הוא שוב בתוך ה- H^*. ובאופן דומה עבור בנים של w'.

② H^* מקסימלית. יהי H יחס ביסימולציה ונראה ש- $H \subseteq H^*$.
נראה שלכל i, $H \subseteq H_i$ ולכן מהגדרת H^*, $H \subseteq H^*$.
את $H \subseteq H_i$ נראה באינדוקציה על i (תרגיל).

סימון: אם M ו-M' ביסימולים נסמן $M' \approx M$.

משפט: $M' \approx M$ אמ"מ M ו-M' מסכימים על כל נוסחאות CTL^*.

אם $M' \approx M$ ברור שהם מסכימים. לצורך הוכחת הכיוון השני נציג גישה מבוססת משחקים לביסימולציה.

בהינתן M ו-M' נתבונן במשחק הבא:
- שחקן I: רוצה לטעון שאין ביסימולציה.
- שחקן II: רוצה לטעון שיש ביסימולציה.
- מהלך המשחק:

(א) שחקן I בוחר מצב $w_0 \in W_0$ או $w_0' \in W_0'$.

(ב) שחקן II בוחר מצב $w_0' \in W_0'$ או $w_0 \in W_0$ (בהתאמה לבחירה של שחקן I ב-(א)) רק ש-$L(w_0)=L(w_0')$.

קיבלנו $\langle w_i, w_i' \rangle \in W \times W'$ (עמדה במשחק)

(ג) שחקן I בוחר $w_{i+1} \in W$ רק ש-$R(w_i, w_{i+1})$
או $w_{i+1}' \in W'$ רק ש-$R'(w_i', w_{i+1}')$

(ד) שחקן II בוחר $w_{i+1}' \in W'$ רק ש-$R'(w_i', w_{i+1}')$
או $w_{i+1} \in W$ רק ש-$R(w_i, w_{i+1})$ (בהתאמה לבחירה של שחקן II ב-(ג)) ו-$L(w_{i+1})=L(w_{i+1}')$;

(ה) חוזרים ל-(ג)

אסטרטגיה היא פונקציה שאומרת מה לעשות לכל שחקן בכל מצב של המשחק.

אסטרטגיה לשחקן I:

$$f_I : (W \times W')^* \longrightarrow W \cup W'$$

רק ש- $f_I(\varepsilon) \in W_0 \cup W_0'$ ולכל i אם

$$f_I(\langle w_0, w_0' \rangle, \dots, \langle w_i, w_i' \rangle) = s$$

אם $s \in W$ — $R(w_i, s)$ או
$s \in W'$ — $R'(w_i', s)$

אסט' לשחקן II: $f_{II} : (W \times W')^* \cdot (W \cup W') \longrightarrow W \cup W' \cup \{\bot\}$

כך ש-
$f_{II}(w_0) = w_0' \in W_0'$
$f_{II}(w_0') = w_0 \in W_0$
$L(w_0) = L(w_0')$

($\bot$ אם אין w_0 או w_0' כאלה).

$f_{II}(\langle w_0, w_0' \rangle, \dots, \langle w_i, w_{i+1} \rangle, w_{i+1}) = w'_{i+1}$

כך שאם $w_{i+1} \in W$ אז $w'_{i+1} \in W'$
ו- $R(w_i', w'_{i+1})$ ו- $L(w_{i+1}) = L'(w'_{i+1})$
ואם $w_{i+1} \in W'$ אז $w'_{i+1} \in W$
ו- $R'(w_i', w'_{i+1})$ ו- $L(w_{i+1}) = L(w'_{i+1})$
אם אין כזה אז $w'_{i+1} = \bot$.

אם המשחק ממשיך לאינסוף איטרציות אז שחקן II מנצח ואם המשחק נתקע אז שחקן II משחק $\bot$ ומפסיד.

$outcome(f_I, f_{II}) = \langle w_0, w_0' \rangle, \dots, \langle w_i, w_i' \rangle, \dots$

וזה אינסופי או מסתיים בהצבה עם $\bot$.
אינסופי ← II מנצח; מסתיים בהצבה עם $\bot$ ← I מנצח

f_{II} היא אסט' ניצחון עבור שחקן II אם לכל אסט' f_I עבור שחקן I, $outcome(f_I, f_{II})$ מנצח עבור II.
באופן דומה, אסט' f_I של שחקן I היא אסט' ניצחון אם היא מנצחת כל אסט' f_{II} של שחקן II.

משפט: ① במשחק שתיארנו, לשחקן II יש אסט' ניצחון אמ"מ יש יחס ביסימולציה בין M ל- M'.
② ל- I יש אסט' ניצחון אמ"מ אין יחס ביסימולציה.

<u>הוכחה:</u>

(1) ($\Leftarrow$) אם יש ביסימולציה אז יש אסט' ניצחון. היא פשוט עוקבת אחרי יחס הביסימולציה. תמיד השחקן האחרון מקבל ל-II את w נבחר w' רק ש- $H(w,w')$, ואם w הוא הקלט היחיד אז w' הוא מצב התחלתי.

צ"ל נבחר w' רק ש- $H(w,w')$;

$$f_{II}(w) = w' : H(w,w'),\ w' \in W_0'$$
$$f_{II}(w') = w : H(w,w'),\ w \in W_0$$
$$f_{II}(\pi \cdot w) = w' : H(w,w')$$

($\not\Leftarrow$) נניח שאין ביסימולציה ונראה שיש אסט' ניצחון לשחקן I (אז לא יכולה להיות אסט' נצחון לשחקן II)

נראה שאם $\langle w,w' \rangle \notin H_i$ אז יש לשחקן I אסט' לנצח את שחקן II תוך i צעדים במשחק שבו המצב ההתחלתי של M הוא w והמצב ההתחלתי של M' הוא w'. זה יוכיח את הטענה שלנו כי אם אין ביסימולציה אז יש $w \in W_0$ רק שלכל $w' \in W_0'$ $\neg H^*(w,w')$ (או הדואלי לזה). אז אם שחקן I יתחיל ב- w לא יהיה לשחקן II מה לענות לו כי כל מענה ידפוק אותו.

נוכיח את טענת העזר באינדוקציה על i.

עבור $i=0$, $\langle w,w' \rangle \notin H_0$ האסט' של I תבחר למשל את w ושחקן II חייב לענות w' ולכן $\perp$

נניח נכונות עבור i ; $\langle w,w' \rangle \notin H_{i+1}$ אז $\langle w,w' \rangle$ לא טוב ביחס ל- H_i, כלומר קיים בן s של w רק שלכל בן s' של w' $\langle s,s' \rangle \notin H_i$ (או הדואלי)

מהנחת האינדוקציה יש ל-I אסט' לנצח ב- $i+1$ צעדים מ- $\langle s,s' \rangle$, אז אסט' ב- $i+2$ צעדים מ- $\langle w,w' \rangle$: הולך ל- s ולא חשוב איך השני יענה, I ינצח!

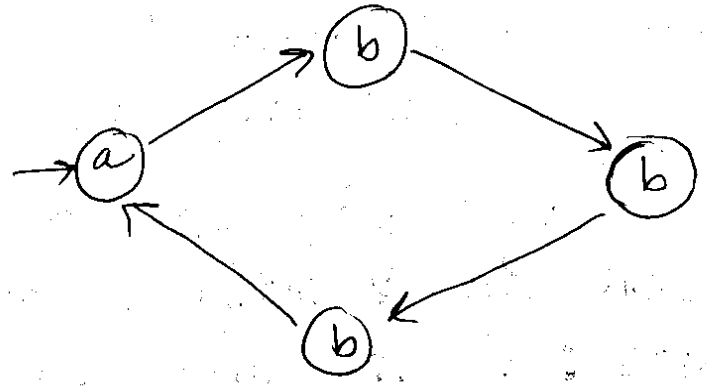
4/11/2010
אימות פורמלי —

ביסימולציה

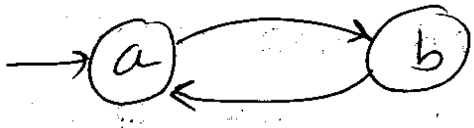
ראינו ש- $M \approx M'$ אומר M ו- M' מסכימים על
כל נוסחאות CTL^*
וראינו גם אלגוריתם לחישוב ביסימולציה מקסימלי (אלג' שמחפש נקודת שבת) וראינו גישה מבוססת משחקים לביסימולציה.

דוגמה:

מבנה I



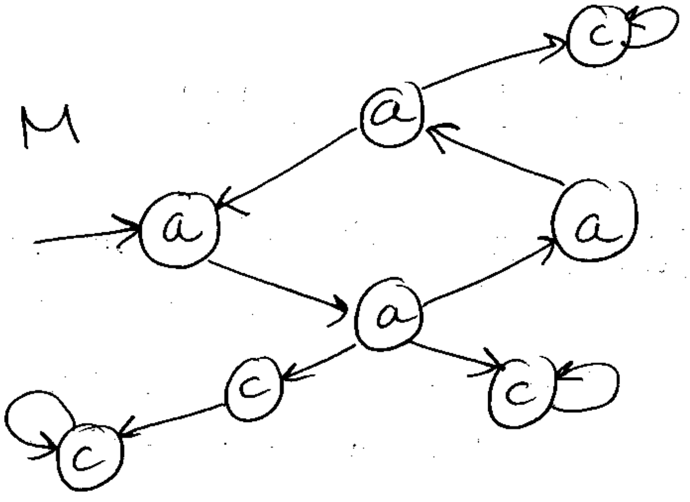
מבנה II



ברור שהם ביסימולריים אבל II הרבה יותר קטן ופשוט!

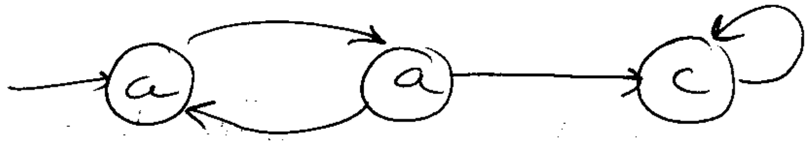
אז באופן כללי המטרה שלנו היא <u>אבסטרקציה</u>:
בהינתן מבנה קריפקה M, אנחנו מחפשים מבנה M_{abs} מינימלי כך ש- $M \approx M_{abs}$.

דוגמה:



מה שהיינו רוצים הוא:

Mabs



זה כמובן הרבה יותר פשוט. למה שמישהו בכלל ירצה את התכנית המסובכת M? יכול להיות שבמקור היו שם הרבה יותר אטומים, אבל אם אנחנו רוצים לבחון רק תכונות שקשורות ל-a ול-c אז מספיק לזרוק את כל שאר הקומפוננטות ואימות של תכונות כאלה מספיק לעשות על תכנית קטנה יותר.

אז איך עושים אבסטרקציה כזאת?
ניתן להגדיר ביסימולציה של מצבי מבנה יחיד $H\subseteq W\times W$ וכך מתקבל יחס שקילות על W: ברור ש-H רפלקסיבי (ברור שמצב ביסימולרי לעצמו), H כמובן סימטרי (זה גם ברור מההגדרה) וגם לא קשה לראות ש-H גם טרנזיטיבי: אם $H(w_1,w_2)$, $H(w_2,w_3)$ אז ברור שהם מסומנים באותו תג ושיש להם בנים מתאימים.
כעת, המצבים של Mabs יהיו מחלקות השקילות W/H. כלומר

$$M_{abs} = \langle AP, W/H, W^0_{abs}, R_{abs}, L_{abs} \rangle$$

כאשר

$$W^0_{abs} = \{ [w_0] : w_0 \in W^0 \}$$

המחלקה של w_0

$$R_{abs} = \{ ([w],[w']) : \text{קיים } s\in[w] ;\ s'\in[w'] \text{ כך ש- } (s,s')\in R \}$$

$$L_{abs}([w]) = L(w)$$

וזה מוגדר היטב כי כל הנציגים של מחלקת שקילות מסומנים באותו אופן.

(91)

משפט: $M_{abs} \approx M$ ויהי M', אם $M' \approx M$ אז $|M'| \geq |M_{abs}|$.

הוכחה: כדי להראות ש- $M_{abs} \approx M$ צריך להראות שיש יחס ביסימולציה $G \subseteq W \times W_{abs}$. נגדיר $G(s,[w])$ אמ"מ $s \in [w]$

נראה שזו אכן ביסימולציה. נראה שהאלגוריתם לחישוב ביסימולציה מגיע ל-fixed point.

כ הצגנו ב- H_0 תנאים ב- G: לכל בן s' של s קיים בן $[w']$ של $[w]$ כך ש- $G(s',[w'])$ (וגם להפך)

כעת נניח בשלילה שיש M' כך ש- $M' \approx M$ אבל $|M'| < |M_{abs}|$. אז יש שני מצבים w_1, w_2 של M ומצב s של M' כך ש- $\neg H(w_1,w_2)$; $A(s,w_1), A(s,w_2)$.

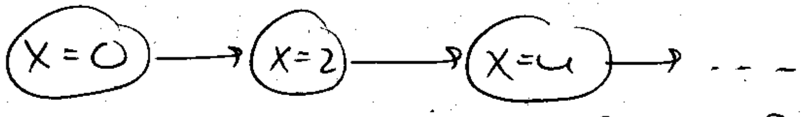
$\neg H(w_1,w_2)$ אז קיים נוסחה מפרידה ψ, כלומר $w_1 \models \psi$; $w_2 \not\models \psi$. האם $s \models \psi$? אם כן אז מצב אחד כן ומצב שני לא. זו סתירה. לכן M_{abs} מינימלי! ☺

משתמשים בביסימולציה כדי לעשות אימות של תוכנה. ביסימולציה מוגדרת גם על מבני קריפקה אינסופיים.

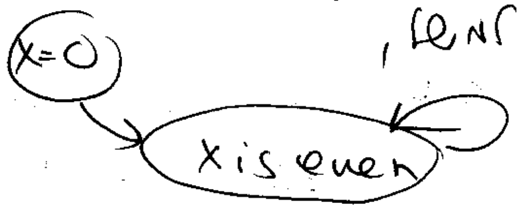
דוגמה:

```
x := 0
while x<1000   (or while true)
   do x = x+2
   if x is odd then ERROR
```

predicate abstraction: במקום להסתכל במבנה האינסופי



נתבונן בקבוצות, למשל,



* באחד עשו את זה ל-MSWindows

דוגמה: נתון x,:

אם x זוגי $\leftarrow \frac{x}{2}$

אם x אי-זוגי $\leftarrow 3x+1$

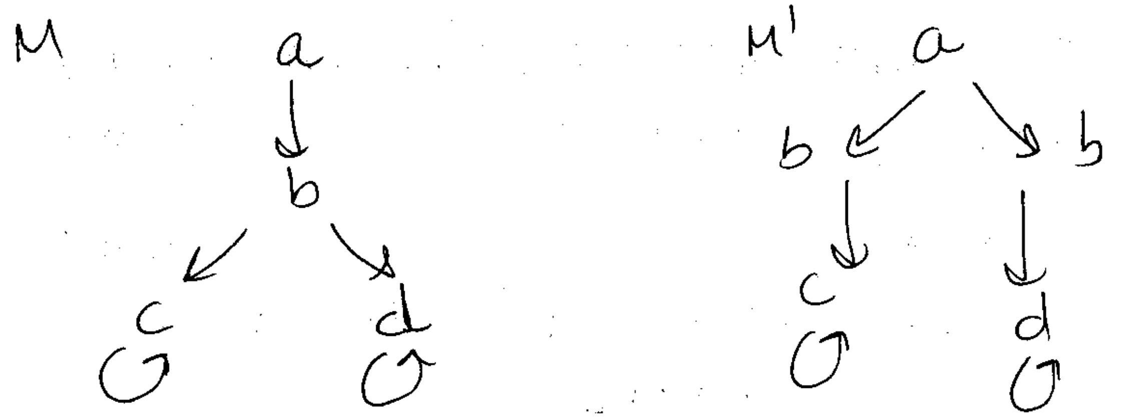
שאלה פתוחה: האם זה תמיד מגיע ל-1?

ביסימולציה הייתה יחס סימטרי - לשני המבנים הייתה בדיוק אותה התנהגות.
אפשר לדבר על משהו חלש יותר.
סימולציה $H \subseteq W \times W'$. היא לא יחס סימטרי והיא אומרת ש-M' יש יותר התנהגויות מ-M. נסמן $M \leq M'$.
H יחס סימולציה אמ"מ לכל $H(w,w')$

① $L(w) = L'(w')$

② לכל s כך ש- $R(w,s)$ קיים s' כך ש- $R'(w',s')$ ו-$H(s,s')$
(אבל לא להפך בהכרח).

דוגמה: ראינו שאלה לא ביסימולריים:



כאן $M' \leq M$ אבל לא $M \leq M'$. ל-M יש התנהגות שאחרי b יכול להיות גם c וגם d

נאמר ש-M' מסמלץ את M ($M \leq M'$) אם לכל $w_0 \in W_0$ יש $w_0' \in W_0'$ כך ש-$H(w_0,w_0')$.

הערה: ייתכן להיות ש- $M \leq M'$; $M' \leq M$ אבל $M \not\equiv M'$!!

לא נראה: אלגוריתם לחישוב סימולציה מקסימלי.

- אפשר להגדיר משחק כמו משחק הביסימולציה אבל שחקן I מתקדם רק ב-M. תוצאת המשחק מתאימה לקיום ואי-קיום סימולציה.
- יש סימולציה אמ"מ כל התנהגות של M קיימת גם ב-M'.

אפיון לוגי

ראינו ש- $M \approx M'$ אמ"מ לכל נוסחה ψ ב- CTL^*

$$M \models \psi \Leftrightarrow M' \models \psi$$

עבור סימולציה ההקשר יותר עדין. $M \leq M'$ אמ"מ לכל נוסחה ψ ב- $\forall CTL^*$ $\quad M \models \psi \Leftarrow M' \models \psi$

$\forall CTL^*$ היא כמו CTL^* רק שאין בה כמת קיום: נוסחאות המצב הן:

- p ו- $\neg p$ עבור $p \in AP$
- $\varphi_1 \vee \varphi_2$, $\varphi_1 \wedge \varphi_2$ עבור $\varphi_1, \varphi_2 \in \forall CTL^*$ נוסחאות מצב
- $A\psi$ עבור נוסחת מסלול ב- $\forall CTL^*$

לוגיקה כזו מדברת נוסחאות שככל שיש יותר התנהגויות, יותר קשה לספק אותן.

נראה שאם $M \leq M'$ אז לכל נוסחה $\psi \in \forall CTL^*$ מתקיים $M \models \psi \Leftarrow M' \models \psi$.

באינדוקציה על מבנה הנוסחה ψ:

יהי $H \subseteq W \times W'$ יחס הסימולציה. נראה ש-אם $H(w,w')$ אז $w \models \psi \Leftarrow w' \models \psi$ ואז תנבע הטענה הכללית (בגלל שהמצבים ההתחלתיים מתאימים)

- אם $\psi = p$ או $\psi = \neg p$ זה ברור כי $L(w') = L(w)$
- אם $\psi = \varphi_1 \vee \varphi_2$ מהנחת האינדוקציה $w \models \varphi_1 \leftarrow w' \models \varphi_1$, $w \models \varphi_2 \leftarrow w' \models \varphi_2$

ולכן $w \models \varphi_1 \vee \varphi_2 \Leftarrow w' \models \varphi_1 \vee \varphi_2$

• אם $\psi = \varphi_1 \wedge \varphi_2$ כנ"ל אותו דבר.

• אם $\psi = A\theta$ עבור נוסחת מסלול θ.

אם $w' \models \psi$ אז לכל המסלולים שיוצאים מ-w' מספקים את θ. יהי $\pi = w_1 w_2 \ldots$ מסלול שיוצא מ-w (כלומר $w_1 = w$). אז יש $\pi' = w_1' w_2' \ldots$ מסלול מתאים ($H(w_i, w_i')$ לכל $i \geq 1$) שיוצא מ-w'. לכן π' מספק את θ ולכן גם $\pi \models \theta$.

אימות מודולרי

הרעיון: לאמת מפרטים של מרכיבי המערכת (שהם הרבה יותר קטנים) ולהסיק על המערכת כולה.

דוגמה:

```
M1:   read x
      if x is even then  y=6
      else               y=7

M2:   read y
      if y>3 then  z=5
      else         z=8
```

נניח שהמבנה סדרתי $x \to [M_1] \xrightarrow{y} [M_2] \to z$

מפרט עבור M: אומרים מה ההנחות ומה מובטח אם הן מתקיימות assume-guarantee

$\langle\varphi\rangle M \langle\psi\rangle$: לכל סביבה M', אם $M \| M' \models \varphi$ אז $M \| M' \models \psi$

(הרכבה רצים במקביל)

למשל, בדוגמה שלנו:

$$\langle true \rangle\ M_1\ \langle y>3 \rangle$$
$$\langle y>3 \rangle\ M_2\ \langle z=5 \rangle$$
$$\Rightarrow\ \langle true \rangle\ M_1 \| M_2\ \langle z=5 \rangle$$

אז זה נהדר. בדקנו משהו על המודולים הקטנים והסקנו משהו לגבי המערכת הגדולה.

<u>המערכת $M \| M'$</u> : נניח ש-

$$M = \langle AP, W, W_0, R, L, F \rangle$$
$$M' = \langle AP', W', \dots \rangle$$

כאן חשוב להסתכל על תנאי הוגנות כי מעניין אותנו לדון במקרים שבהם ה- scheduler לא יזנח אף מודול..

כאן הוגנות חזקה (Streett):

$$F = \{\langle L_1, U_1 \rangle, \dots, \langle L_k, U_k \rangle\} \quad L_i, U_i \subseteq W$$

מסלול π הוא הוגן אם לכל זוג $\langle L_i, U_i \rangle$ אם $inf(\pi) \cap L_i \neq \emptyset$ אז $inf(\pi) \cap U_i \neq \emptyset$

(במה שאנחנו למדנו קודם למעשה היה לנו $L_i = W$ ואז רק רצינו לוודא שמבקרים ב- U_i אינסוף פעמים).

אז נגדיר

$$M \| M' = \langle AP'', W'', W_0'', R'', L'', F'' \rangle$$

כאשר

$$AP'' = AP \cup AP'$$
$$W'' = \{\langle w, w' \rangle : L(w) \cap AP' = L'(w') \cap AP\}$$

זוגות w, w' מסכימים על האטומים ב- $AP \cap AP'$.

$$W_0'' = (W_0 \times W_0') \cap W''$$

$$R''(\langle w, w' \rangle, \langle s, s' \rangle) \Leftrightarrow R(w,s) \wedge R'(w',s')$$

$$L''(\langle w, w' \rangle) = L(w) \cup L'(w')$$

מסלול "זר" הוא ב- $M||M'$ אמ"מ - F''

ההטלה של W הוגנת ב- M וגם

ההטלה של W' הוגנת ב- M'. כלומר

$$F'' = \{ \langle (L_1 \times W') \cap W'', (U_1 \times W') \cap W'' \rangle, \ldots, \langle (L_k \times W') \cap W'', (U_k \times W') \cap W'' \rangle,$$

$$\langle (W \times L'_1) \cap W'', (W \times U'_1) \times W'' \rangle, \ldots, \langle (W \times L'_{k'}) \cap W'', (W \times U'_{k'}) \times W'' \rangle \}$$

אימות מודולרי

Assume-Guarantee

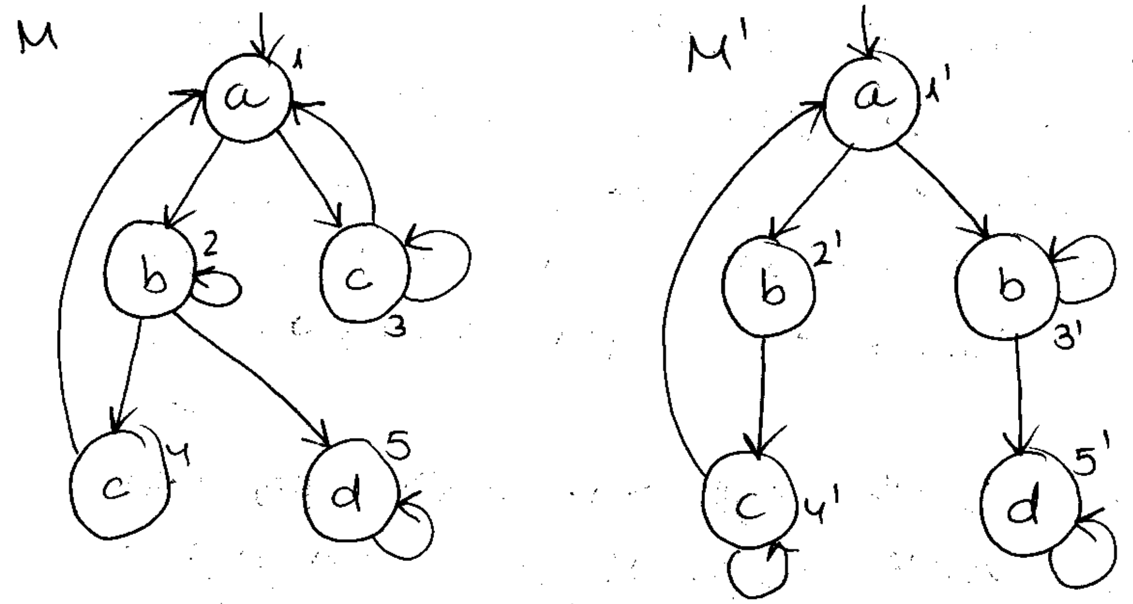
$\langle\varphi\rangle M \langle\psi\rangle$ - לכל M' אם $M||M' \models \varphi$ אז $M||M' \models \psi$

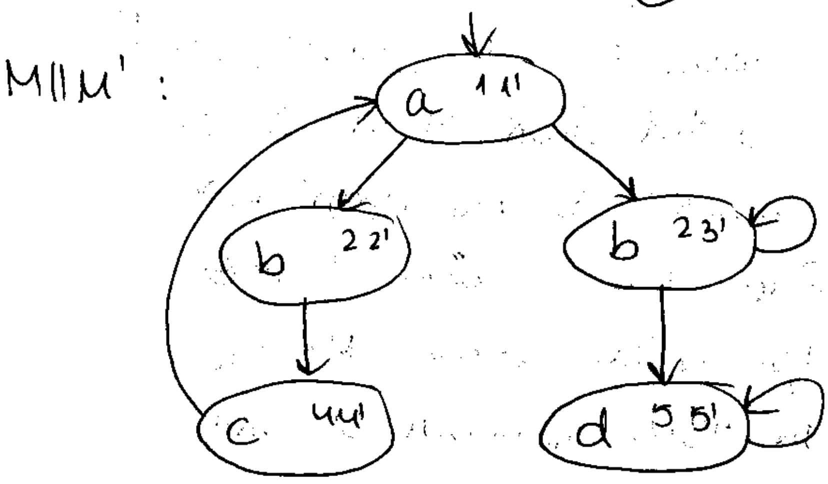
הרעיון הוא להוכיח על חלקים של המערכת ואז לשלב כדי לקבל טענה על המערכת הגדולה. למשל:

$$\left.\begin{array}{l} \langle true\rangle M_1 \langle\varphi_1\rangle \\ \langle\varphi_1\rangle M_2 \langle\psi_1\rangle \\ \langle true\rangle M_2 \langle\varphi_2\rangle \\ \langle\varphi_2\rangle M_2 \langle\psi_2\rangle \end{array}\right\} \Rightarrow \langle true\rangle M_1||M_2 \langle\psi_1 \wedge \psi_2\rangle$$

למה זה טוב? סביר שיותר קל להראות דברים על המודולים הקטנים יותר, שהם (עשויים להיות) מעריכית קטנים יותר מהמערכת השלמה.

דוגמה:





תכונות הרכבה מקביליות:

- לכל M, M' מתקיים $M \| M' \leq M$ (וגם $M \| M' \leq M'$)
יחס הסימולציה הוא $H(\langle w, w' \rangle, w)$.
- לא בהכרח מתקיים $M \leq M \| M'$ (ראינו בדוגמה קודם).
- אם $M \leq M'$ אז לכל M'' $M \| M'' \leq M' \| M''$

בדיקת Assume-guarantee

בהינתן M, φ, ψ, האם $\langle\varphi\rangle M \langle\psi\rangle$?
כלומר האם לכל M' $M \| M' \models \varphi \Rightarrow M \| M' \models \psi$?
זה נשמע קשה כי לכאורה צריך לבדוק עבור אינסוף מבנים M'.

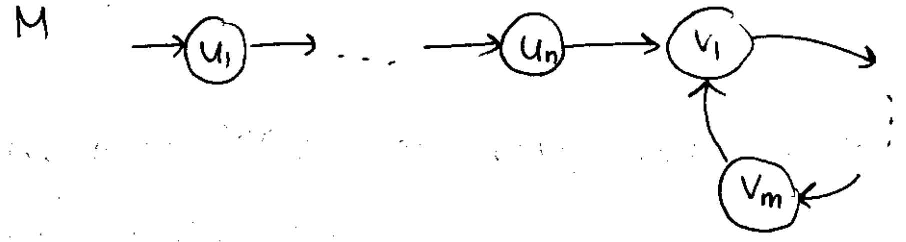
<u>למה</u>: עבור φ, ψ ב- LTL,
$\langle\varphi\rangle M \langle\psi\rangle$ אמ"מ $M \models A(\varphi \rightarrow \psi)$

<u>הוכחה</u>:
($\Rightarrow$) תהי M המקיימת $M \models A(\varphi \rightarrow \psi)$. ונתבונן ב- M' רק ש- $M \| M' \models A\varphi$ (ב- LTL זה כמו לכתוב $M \| M' \models \varphi$). אנחנו יודעים (מתכונות ההרכבה) ש- $M \| M' \leq M$. $A(\varphi \rightarrow \psi)$ היא נוסחה אוניברסלית לכן $M \| M' \models A(\varphi \rightarrow \psi)$. אבל $M \| M' \models A\varphi$ לכן $M \| M' \models A\psi$ וזה מה שרצינו.

($\Leftarrow$) נתון $\langle\varphi\rangle M \langle\psi\rangle$ ונניח בשלילה ש- $M \not\models A(\varphi \rightarrow \psi)$.
$\Leftarrow$ קיים מסלול π ב- M כך ש- $\pi \models \varphi$ אבל $\pi \not\models \psi$

למה/תזכורת: אם θ נוסחת LTL ; $M \not\models \theta$
אז יש ב- M מסלול מהצורה $uv^\omega \not\models \theta$
אפשר לתרגם את θ לאוטומט בוקי $A_{\neg\theta}$.
$M \not\models \theta$ אמ"מ $L(M \times A_{\neg\theta}) \neq \emptyset$. המכפלה
$M \times A_{\neg\theta}$ היא גם אוטומט בוקי ואם הוא לא ריק זה
אומר שיש מילה מקבלת, שיש בה מעגל. זה משרה את uv^ω

רק יש מסלול π כזה מהצורה $\pi=uv^{\omega}$. אז יש מבנה סופי M_π שמתאר את המסלול:



נתבונן ב- $M \| M_\pi$: מתקיים

* - $M_\pi \leq M\|M_\pi$ (כי M לא מגביל את M_π)

** - $M\|M_\pi \leq M_\pi$ (תכונות ההרכבה)

נתון $\langle\varphi\rangle M \langle\psi\rangle$ $\Leftarrow$ אם $M\|M_\pi \models A\varphi$ אז $M\|M_\pi \models A\psi$. אכן $M\|M_\pi \models A\varphi$ (מגן) ** כי $M_\pi \models A\varphi$)

אבל $M\|M_\pi \not\models A\psi$ כי $M_\pi \not\models A\psi$ ותכונות הסימולציה! - * אומר אם $M\|M_\pi \models A\psi$ אז $M_\pi \models A\psi$.

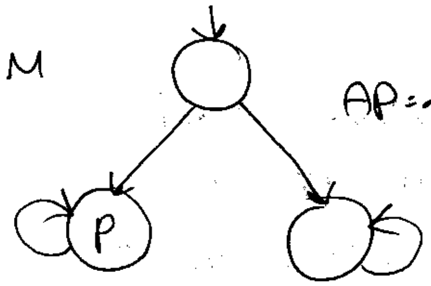
קיבלנו סתירה. ובזאת סיימנו. ☺

<u>טענה</u>: המשפט הקודם לא עובד ל- ACTL: יש נוסחאות φ ו- ψ ב- $\forall CTL$ ומבנה M רק ש-

$$\langle\varphi\rangle M \langle\psi\rangle \not\Leftrightarrow M \models \varphi \rightarrow \psi$$

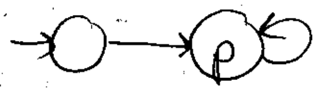
<u>הוכחה</u>: צריך לתת דוגמה.

צריך למצוא שתי נוסחאות φ,ψ רק ש- $M \models \varphi \rightarrow \psi$



וגם בזאת יש M' רק ש- $M\|M' \models \varphi$ אבל $M\|M' \not\models \psi$.

מתקיים $M \models \varphi \rightarrow \psi$ (באופן ריק כי לא תמיד יש בסוף p)

ניקח M':  . אז $M\|M' = M'$

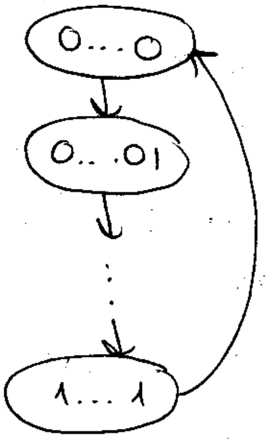
ו- $M\|M' \models \varphi$ אבל $M\|M' \not\models \psi$. ☺

$\Leftarrow$ צריך אלגוריתם חדש שונה מ- MC לבדיקת $\langle\varphi\rangle M \langle\psi\rangle$ אם φ,ψ לא ב- LTL. מה אפשר אם לא נמצא את זה.

סינתזה

יש נוסחה ורוצים להפוך אותה באופן אוטומטי למערכת (חלומו הרטוב של כל מתכנת)

התהליך הזה כמובן קשור באופן הדוק לספיקות.



__דוגמה__: סינתזה למונה n ביטים

נרצה לכתוב נוסחה שמתארת מונה כזה.

נשתמש באטומים $p_1,\dots,p_n$

שמייצגים את הביטים וב- $c_1,\dots,c_n$

שמייצגים את ה- carry.

אתחול למצב הראשון:

$\bigwedge_{i=1}^{n} \neg p_i$, $\bigwedge_{i=2}^{n} \neg c_i$, $G c_1$

והחיבור מוגדר ע"י

$$G(\neg p_i \wedge \neg c_i \rightarrow X(\neg p_i \wedge \neg c_{i+1}))$$
$$G(((\neg p_i \wedge c_i) \vee (p_i \wedge \neg c_i)) \rightarrow X(p_i \wedge \neg c_{i+1}))$$
$$G(p_i \wedge c_i \rightarrow X(\neg p_i \wedge c_{i+1}))$$

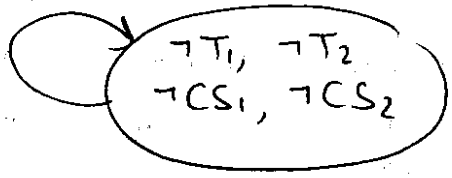
איך עושים סינתזה ל- LTL? נהפוך את ψ לאוטומט. אם הוא ריק אז אי אפשר. אחרת, קיבלנו את מה שרצינו וכל זה ב- PSPACE.

__דוגמה__: critical sections

$$G(\neg CS_1 \wedge \neg CS_2)$$
$$G(T_1 \rightarrow F CS_1)$$
$$G(T_2 \rightarrow F CS_2)$$

נעשה סינתזה ונקבל מע' אופטימלית לניהול קטעים קריטיים.

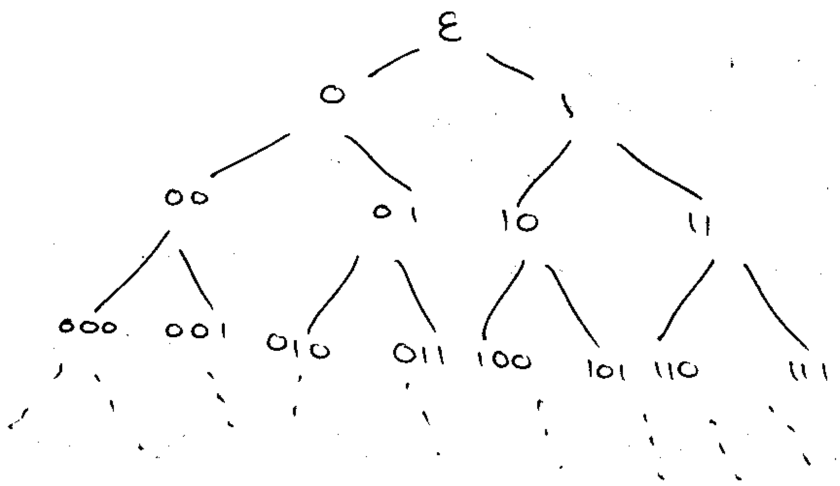
אבל זה לא טוב כי למשל רצינו לקבל את המערכת



וזה בכלל לא מה שרצינו... זה מגדל שהמערכת שאנחנו מעוניינים בה היא פתוחה - מקבלת גם קלטים מהסביבה.
במערכת סגורה שפועלת ללא תלות בסביבה סינתזה אכן שקולה לספיקות.

<u>הגדרה:</u> עבור קבוצה D של כיוונים, D-tree (עץ-D) הוא הקבוצה D^*

<u>דוגמה:</u> $D=\{0,1\}$ אז את D^* אפשר לתאר בעץ:



<u>הגדרה:</u> עבור עץ D^* ועבור קבוצה Σ, Σ-labeled D-tree הוא העץ D^* עם פונקציית תיוג
$\tau : D^* \to \Sigma$

אם I קבוצת קלטים, O- קבוצת פלטים אז העץ עם $D = 2^I$ הוא למעשה כל סדרות הקלט האפשריות -;
2^O-labeled 2^I-tree הוא תגובה אפשרית של המערכת -
$\tau : (2^I)^* \to 2^O$. τ גם נקראת <u>אסטרטגיה</u>.

<u>הגדרה:</u> בהינתן אסטרטגיה $\tau : (2^I)^* \to 2^O$ ומערכת,
חישוב של τ הוא סדרה

$\tau(\varepsilon),\ i_1 \cup \tau(i_1),\ i_2 \cup \tau(i_1 \cdot i_2) \ldots$

זכור סדרת קלטים $(2^I)^\omega \in i_1, i_2, \ldots, i_n, \ldots$

נוסחה ψ ב- LTL מעל $O \cup I$ היא בת-סינתזה

(realizable) אם קיימת אסטרטגיה $\tau: (2^I)^* \rightarrow 2^O$

של תשובות מספקים את ψ

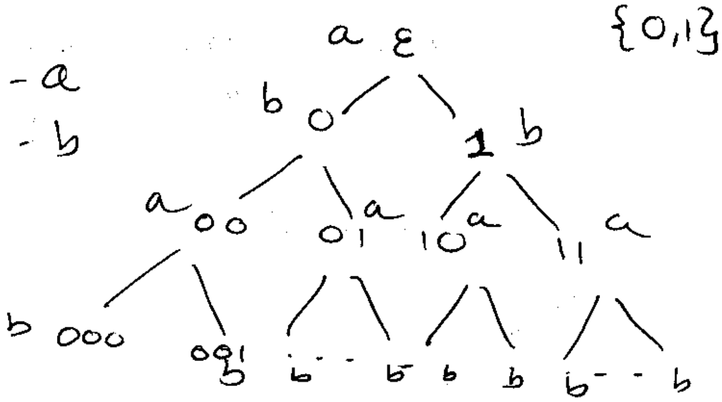
<u>הגדרה:</u> נאמר ש- τ Σ-labeled D-tree הינו רגולרי

אם לכל אות $\sigma \in \Sigma$ הקבוצה $\tau^{-1}(\sigma)$ רגולרית

<u>דוגמה:</u> $\{0,1\}^*$

a - בקומות זוגיות

b - בקומות אי-זוגיות



$\Leftarrow$ $\tau^{-1}(a) = (\Sigma\Sigma)^*$ שפה רגולרית.

אם ה-a-ים היו בקומות הראשוניות זה <u>לא</u> היה רגולרי.

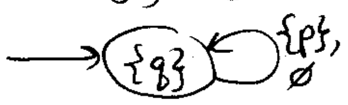
אסטרטגיה רגולרית ניתנת לתיאור ע"י משרן (transducer)

סופי $A = \langle D, Q, q_0, M, \tau: Q \rightarrow \Sigma \rangle$, $D = 2^I$, $\Sigma = 2^O$

זה אוטומט שבכל מצב שלו יש פלט

<u>דוגמה:</u> $I = \{p\}$, $O = \{q\}$. האם realizable?

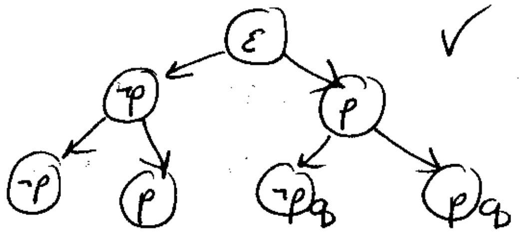
נוסחה שאינה ספיקה בוודאי לא בת-סינתזה.

① Gq . ✓ זה משפן 

② Gp . ✗ זה ספיק אך לא בר-סינתזה. וסדרת

הקלט ש-$\neg p$ לא ניתן להבטיח סדרת פלט

③ $G(p \rightarrow Xq)$ ✓



(4) $G(\neg p \rightarrow X\neg q) \wedge Fq \wedge \neg q$. ✗

כי נסתר עם $(\neg p)^\omega$

(5) $G(q \leftrightarrow Xp)$ ✗

כאן לכל סדרת קלט קיימת סדרת פלט שהחשוב המשותף מספק את ψ אבל אי אפשר לייצר מזה עץ כי צריך איזה lookahead שאין לנו.
לכל נקודת יש מן q ומן $\neg q$ - העתיד מאוזן ולכן לא ניתן להסתדר.

(48) 21/1/2010
אימות
פורמלי

סינתזה:

קלט: נוסחת LTL ψ מעל $I \cup O$

פלט: משהו שממש אסטרטגיה $f:(2^I)^* \longrightarrow 2^O$
שכל תשובותיו מספקים את ψ.

נפתור את הבעיה בעזרת אוטומטים מעל עצים. אמרנו ש-f כמו היא בעצם 2^O-labeled 2^I-tree. הכוונה הוא שהקלט של האוטומט הוא עץ והאוטומט יודע להתפצל לכל מצב. אז אוטומט כזה הוא $A=\langle \Sigma, D, Q, \delta, q_0, \alpha \rangle$
ומה שמעניין כאן הוא להגיד מהי פונקצית המעברים δ.
למשל, נניח ש- $D=\{0,1\}$, כלומר העצים הם בינאריים.
אז כשהאוטומט נמצא במצב q_0 וקורא את האות a
אז ל-a יש שני בנים וצריך להגיד מה המצב שמתאים לכל אחד מהבנים האלה. אז

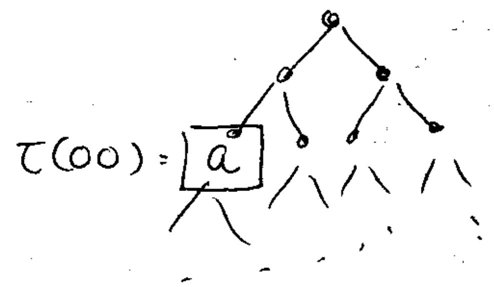
$$\delta : Q \times \Sigma \longrightarrow Q \times Q$$

ריצה r של A על $\langle D^*, \tau \rangle$ הינה Q-labeled D-tree
$\langle D^*, r \rangle$ המקיים:

- $r(\varepsilon)=q_0$ (הריצה מתחילה במצב התחלתי)
- לכל $x \in D^*$ $\langle r(x\cdot 0), r(x\cdot 1) \rangle = \delta(r(x), \tau(x))$

פשוט עץ שמסמן מצבי הריצה של האוטומט

דוגמה: אוטומט A עבור שפת העצים $\langle \{0,1\}^*, \tau \rangle$ רק ש- $\tau(00)=a$. כלומר מעניינים אותנו העצים מהצורה



$\delta(q_0, \frac{a}{b}) = \langle q_1, q_{acc} \rangle$

$\delta(q_1, \frac{a}{b}) = \langle q_2, q_{acc} \rangle$

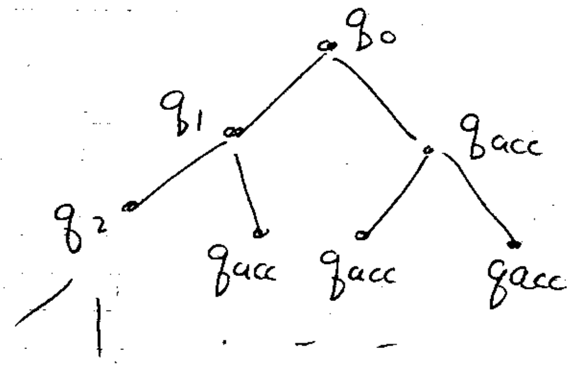
$\delta(q_2, a) = \langle q_{acc}, q_{acc} \rangle$

$\delta(q_2, b) = \langle q_{rej}, q_{rej} \rangle$

$\delta(q_{acc}, \frac{a}{b}) = \langle q_{acc}, q_{acc} \rangle$

$\alpha = \{q_{acc}\}$

כל ריצה תגיע בסוף ל-q_2 ושם או שיהיה a או שלא ואז האוטומט יחליט אם לקבל או לדחות



מתי ריצה מקבלת?

עבור מסלול π בעץ וריצה r נסמן

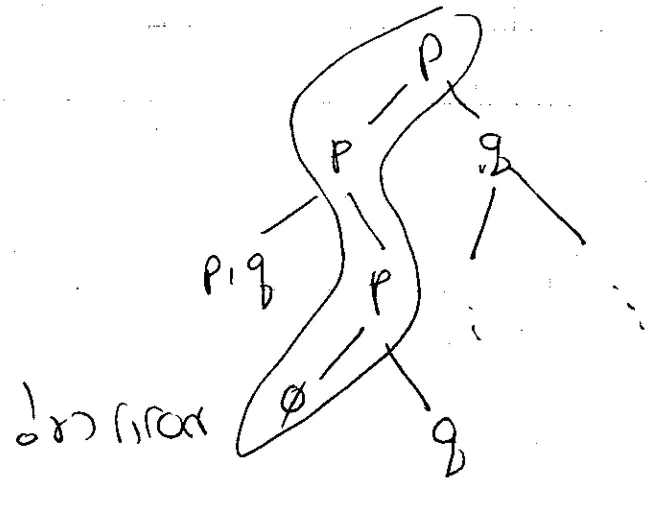
$inf(r|\pi) = \{q :$ עבור אינסוף x-ים ב-π $r(x)=q \}$

אלה המצבים שמבוקרים ב-π אינסוף פעמים.

ריצה $\langle D^*, r \rangle$ מקבלת אם $inf(r(\pi) \cap \alpha) \neq \emptyset$ לכל מסלול π בעץ.

דוגמה: אוטומט עבור $A\, p\, U\, q$. כאן $\Sigma = 2^{\{p,q\}}$

דוגמה לעץ שאמור להידחות:



האוטומט יעבוד כך:

$$\delta(q_0, \emptyset) = \langle q_{rej}, q_{rej} \rangle$$
$$\delta(q_0, \overset{\{q\}}{\{p,q\}}) = \langle q_{acc}, q_{acc} \rangle$$
$$\delta(q_0, \{p\}) = \langle q_0, q_0 \rangle$$
$$\alpha = \{q_{acc}\}$$

אם היינו מגדירים $\alpha = \{q_{acc}, q_0\}$ היינו מקבלים Weak until - $ApWq$ כי היינו שמחים אם היינו מקבלים p אבל אם היינו נתקעים לנצח ב- q_0, כלומר אם היינו נתקעים לנצח עם p זה גם היה בסדר.

נעבור להגדיר ל- D כללי (לא מוכרח בינארי). אז ההכללה הטבעית היא $\delta : Q \times \Sigma \to Q^D$

(Q^D – קבוצת הפונקציות מ- D ל-Q.)

אז $\delta(q,\sigma)$ היא בעצם פונקציה שלכל כיוון $d \in D$ אומרת מה המצב שמתאים לה.

במקרה שהאוטומט דטרמיניסטי אפשר להגדיר באותה מידה וגם $\delta : Q \times \Sigma \times D \to Q$

אבל כך נוח לי להרחיב את ההגדרה הכללית למקרה הלא דטרמיניסטי.

ועכשיו סוף סוף נפתור את בעיית הסינתזה. בהינתן נוסחה ψ ב- LTL נגדיר בשבילה אוטומט A_ψ דטרמיניסטי מעל מילים $A_\psi = \langle 2^{I \cup O}, Q, q_0, \delta, \alpha \rangle$ (ראינו איך אפשר לעשות אוטומט לא דטרמיניסטי ואותו אפשר לתרגם, אם כי אולי יהיה צורך בתנאי קבלה חזק יותר).

כעת נגדיר אוטומט מעל 2^O-labeled 2^I-trees:

$$A'_\psi = \langle 2^O, 2^I, Q, q_0, \delta', \alpha \rangle$$

$$\delta : Q \times 2^O \to Q^{2^I}$$ ראש

$$\delta'(q, o)(i) = \delta(q, i \cup o)$$
($o \in 2^O$, $i \in 2^I$)

<u>משפט</u>: A_ψ' מקבל בדיוק את כל האסטרטגיות של
חישוביהן מספקים את ψ

<u>"הוכחה"</u>: קודם כל, ברור שכל החישובים מספקים את ψ
כי הם למעשה מגיעים מהחישוב הדטרמיניסטי של A_ψ.
חוץ מזה בגלל שזה עץ הוא מייצג את כל הקלטים
האפשריים ולכן אם השפה של האוטומט אינה
ריקה אז בדיוק מצאנו משהו ספציפי שמייצר עץ
שמתקבל. ☺

סקירה של נושאים שלא למדנו

- שפות אחרות

μ-calculus, CTL, CTL*, LTL

לפני כמה שנים IEEE החליטו שצריך לעשות
סטנדרטיזציה ושכולם ישתמשו בתעשייה באותה
שפה. אז המציאו את PSL שזה כמו LTL
אבל הוסיפו לזה שכבה רגולרית וכמו TRIGGERS
אם r, e ביטויים רגולריים אז

$$w \models r \text{ TRIGGERS } e$$

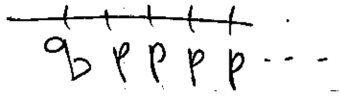
אם לכל $i \geq 0$ אם $w[0,\dots,i] \in L(r)$ אז
קיים $j \geq i$ רק ש- $w[i,\dots,j] \in L(e)$ (כלומר אם עד
עכשיו היה r אז מעכשיו צריך להיות e)
אז למשל $G(req \to F\ grant)$ זה כמו

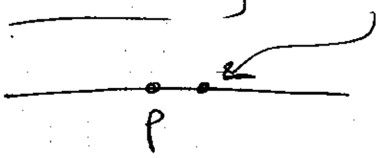
(true* req) TRIGGERS (true* grant)

- לוגיקות עם עבר

אנחנו למדנו את האופרטורים הטמפורליים X, U
אבל יש להם גם חברים Yesterday ; Since

Since: pSq Yesterday: Yp





ול-F יש חבר Past. למשל אולי היינו רוצים לכתוב משהו כמו $G(grant \to P\ req)$

- לוגיקות עם כמתים

אולי נרצה לכמת אטומים. למשל רוצים לדעת אם $\exists p\ G(q \to X p)$. זה הרבה יותר עשיר מלוגיקה רגילה.
ראינו שאין ב-LTL נוסחה שאומרת שבכל המקומות הזוגיים יש p ובאחרים לא אכפת לנו. אבל עם כמתים אפשר לכתוב

$$\exists q\ q \wedge G X \neg q \wedge G(q \leftrightarrow XXq) \wedge G(q \to p)$$

אז פה בעצם נוכל לקחת אטום חדש שלא היה קודם ולשים אותו $q\ \neg q\ q\ \neg q$ וכו', ואז מדויק יוצא לנו.

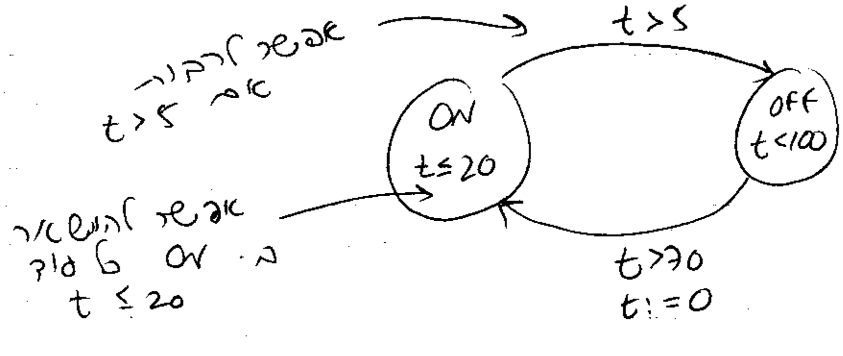
- memoryful CTL* - לא שוכחים את העבר. בכל פעם שבנוסחה מדברים על קיום מסלול הכוונה היא לא למסלול שמתחיל בקודקוד של נקודת הפיצול אלא למסלול שמתחיל בשורש וזוכר דרך הקודקוד שלנו.

- בעיית התפוצצות המצבים

ראינו אלג' סימבוליים, אבסטרקציה ואימות מודולרי.
בתעשייה משתמשים גם ב-SAT-based methods ; SAT-solvers. זה אמנם NP-קשה אבל יש הרבה

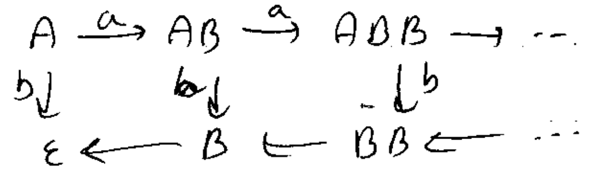
כלו׳, אבל וכיום בד״כ אלגוריתם לפתור SAT.
אז עושים רדוקציה מבדיקת מודל ל- SAT.
דוגמה נגדית ל- Gp זה למשל $p \to p \to \neg p$.
קיום של מסלול כזה אפשר להביע באמצעות נוסחת SAT
אז אם יש דוגמה נגדית סופית אפשר לתת את זה
ל- SAT-solver ולקבל דוגמה נגדית.
עוד שיטות שפחות משתמשים בהן בפועל מתבססות
על סימטריה, partial order methods (אם יש חישובים
שדומים עד כדי פרמוטציה של הפעולות אז אולי אפשר
לבדוק רק נציג אחד מכל מחלקה), אימות
פרמטרי (יש מע׳ עם הרבה מודולים זהים אבל
לא יודעים כמה בדיוק אז צריך איכשהו להוכיח את זה
ל- n כלשהו. אי אפשר עם מבנה קריפקה כי צריך
אינסוף מבנים - לכל n)

- מערכות זמן אמת - כל מה שעשינו היה דיסקרטי.
אבל החיים הם לא דיסקרטיים, ולכן מעניין אותנו
להכניס למע׳ אלמנט של מדידת זמן, למשל, מע׳ תיאום:



מה שלא דיסקרטי כאן זה השעונים.

- גם infinite-state - יש דקדוק $S=\langle\{A,B\},\{a,b\},R,A\rangle$
וגזירות $A \xrightarrow{a} AB$, $A \xrightarrow{b} \varepsilon$, $B \xrightarrow{b} \varepsilon$. זה
משרה מבנה קריפקה אינסופי.



אבל התיאור שלו סופי!

(51)

אז היינו רוצים לעשות model checking ומה'
כבאת שהוא אינסופית אבל נתונה ע"י דקדוק סופי.
עושים את זה בעזרת אוטומט מעל עצים.

- בעיות נוספות
- sanity checks (למשל לבדוק שהנוסחה לא עברה באופן ריק. vacuity checking
- coverage - למצוא שהבדיקות בודקות הכל
- certificate - להבטיח שזה באמת דבר נכון

ועוד ועוד....

אורנה תפרסם מבחנים לדוגמה.
אמור להיות יותר קל מהש"ב.