אוניברסיטת חיפה
החוג למדעי המחשב

אימות פורמלי

סיכומי ההרצאות של ד"ר גיא אבני

נכתב על ידי בר וייסמן

סמסטר חורף תשפ"ד

הקדמה

- בעיה: במערכות/קוד יש באגים.
- Testing: מנסים עד שנמאס.
- מטרת התחום: להוכיח נכונות של מערכות.
- פרה היסטוריה: Hoare, Dijkstra: הוכחות ידניות.
- היסטוריה:

- LTL: Pnueli ('77)
  - לוגיקה טמפורלית: הגדרת מפרטים (התנהגויות חוקיות) באופן פורמלי, למשל:
    1. $\neg(\text{ eventually bug})$.
    2. $\text{always}\,(\text{req} \rightarrow \text{ eventually grant})$.
    3. $\text{always}\,(\neg\,(\text{proc}_1 \text{ in CS} \wedge \text{proc}_2 \text{ in CS}))$.

- Model Checking: Emerson& Clarke ('81), Sifikis & Quielle

$$System \rightarrow M \;\; (\text{model})$$

$$Spec. \rightarrow \varphi \;\; (\text{LTL})$$

  - מהמערכת גוזרים את המודל: ההתנהגויות האפשריות של המערכת.
  - מהמפרט - ההתנהגויות החוקיות.
  - האם המודל עומד במפרט? $\iff$ $M \overset{?}{\neq} \varphi$.

- Vardi & Wolper ('83)
  - תרגום נוסחת LTL φ לאוטומט ששפתו כל המסלולים שמקבלים את φ.
  - כעת, ניתן לבדוק האם $L(M) \overset{?}{\subseteq} L(A_\varphi) \iff L(M) \cap L\left(\overline{A_\varphi}\right) = \emptyset$.

- Synthesis: Pnueli & Rosner ('89)
  - סינתזה של המפרט ע"י רדוקציה למשחק, ופתרון של המשחק.
  - תמריץ טוב לחקור משחקים.

- BDD: Clarke, McMillan et al. ('92)
  - מבנה נתונים לבדיקת מערכות בזמן לוגריתמי במספר המצבים.

- Bounded MC: Clarke ('99)
  - רדוקציה ל-SAT, ופתרון נוסחת ה-SAT ע"י solvers.

מבנה הקורס - GandALF.

- A: Automata.
- L: Logic.
- F: Formal Verification.
- G: Games.

תוכן העניינים

חלק I
אוטומטים מעל מילים אינסופיות

המערכות שנחקור במסגרת הקורס מגיבות לסביבה, ולעולם לא עוצרות. על כן, האוטומטים שבהם נעסוק הם מעל מילים אינסופיות.

<u>תזכורת.</u>

1. אוטומט סופי לא-דטרמיניסטי (NFA) מוגדר ע"י החמישייה

$$A=\left(\underbrace{\Sigma}_{\text{א"ב}},\underbrace{Q}_{\text{מצבים}},\underbrace{\delta}_{\text{פונק' מעברים}},\underbrace{Q_0}_{\text{מצבים התחלתיים}},\underbrace{F}_{\text{מצבים מקבלים}}\right)$$

, כאשר $\delta:Q\times\Sigma\to2^Q$.

2. ריצה של NFA A על מילה $w=\sigma_1\cdots\sigma_n\in\Sigma^*$ מוגדרת ע"י סדרת מצבים

$$r=r_0,r_1,\ldots,r_n\in Q^*$$

כך ש-$r_0\in Q_0$ ולכל i מתקיים $r_{i+1}\in\delta\left(r_i,\sigma_i\right)$. מתקבלת $\iff$ $r_n\in F$.

3. השפה של A היא $L\left(A\right)=\{w\mid$ קיימת ריצה של A על w שמתקבלת$\}$.

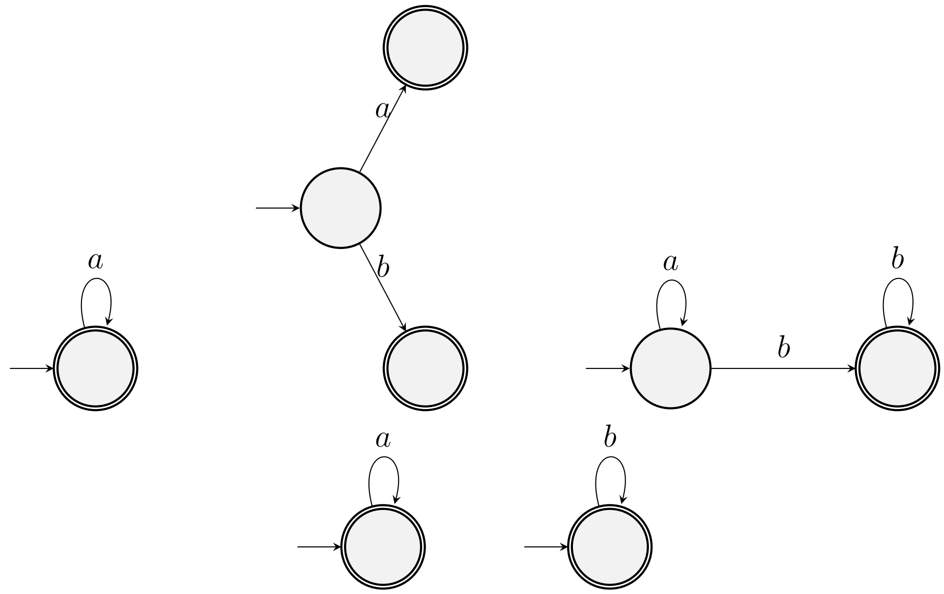
4. ביטויים רגולריים:

   (א) $\varepsilon,a\in\Sigma,\emptyset$.

   (ב) $r_1\circ r_2,r_1+r_2,r_1^*$ עבור ביטויים רגולריים r_1,r_2.

משפט. ביטויים רגולריים = NFA.

דוגמה. מספר ביטויים רגולריים וה-NFA-ים המתאימים להם.



איור 1: משמאל לימין, מלמעלה למטה: האוטומטים עבור a^*, $a+b$, a^*b^*, a^*+b^*.

5. באוטומט סופי דטרמיניסטי (DFA) מתקיים $|Q_0| = 1$ וגם $\delta : Q \times \Sigma \to Q$:
$D = (\Sigma, Q, \delta, q_0, F)$.

משפט. ל-NFA ול-DFA אותו כוח הבעה.

- לכל NFA A קיים DFA D כך ש-$L(A) = L(D)$, בנייה ע"י Subset Construction (פיצוץ מצבים: מ-n ל-2^n מצבים).

6.

משפט. (Myhile-Nerode) לכל שפה רגולרית $L \subseteq \Sigma^*$ קיים DFA מינימלי <u>יחיד</u> שמזהה אותה.

1 אוטומטי Buchi

אוטומט בוקי (NBW) $A = \left(\Sigma, Q, \delta, Q_0, \underbrace{\alpha}_{\text{מצבים מקבלים}}\right)$.

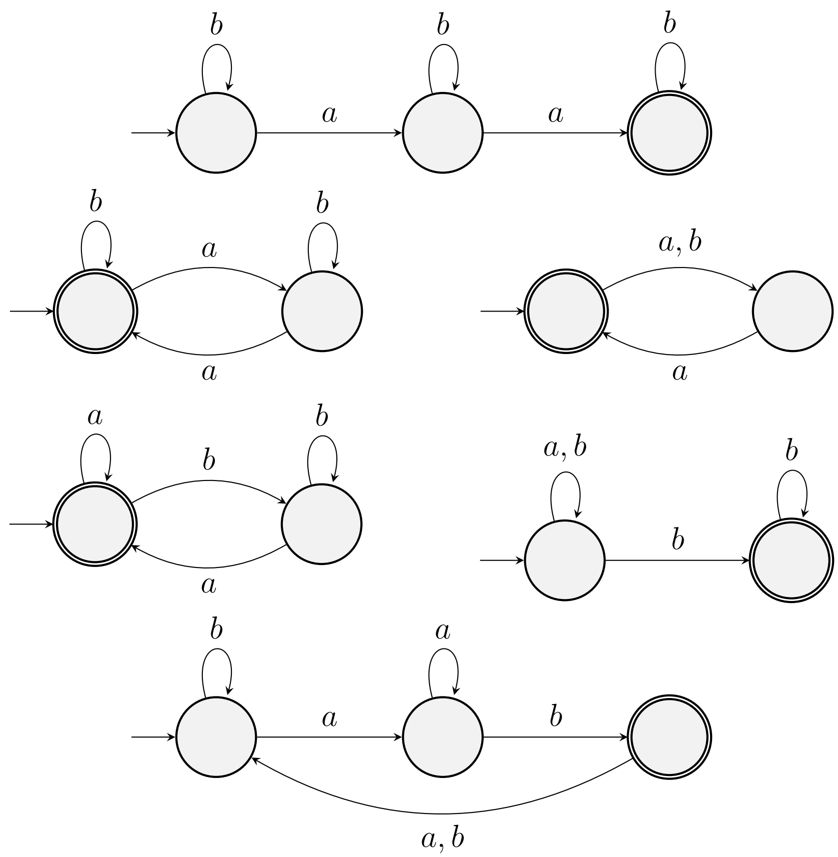
סמנטיקה מעל מילים אינסופיות:

- ריצה של A על מילה $w=\sigma_1\sigma_2\ldots$ היא $r=r_0r_1r_2\ldots$, **כך ש-** $r_0\in Q_0$ וגם לכל $i\geq 1$ מתקיים $r_i\in\delta\left(r_{i-1},\sigma_i\right)$.
- ריצה <u>מקבלת</u> $\iff$ מבקרת מצבים מקבלים אינסוף פעמים. באופן פורמלי, נגדיר

$$\inf\left(r\right)=\left\{q\in Q\mid \text{קיימים } \infty\ i \text{ כך ש-} r_i=q\right\}$$

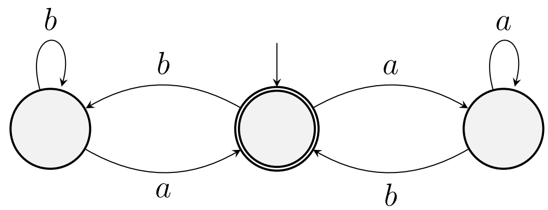
וכך r מקבלת $\iff \inf\left(r\right)\cap\alpha\neq\emptyset$.

דוגמה. מספר שפות ואוטומטי בוקי המתאימים להם.



איור 2: משמאל לימין, מלמעלה למטה: L_1 היא שפת כל המילים בהן $2\geq {}_a\#$. L_2 היא (מס' זוגי של $a\wedge a$ $\neg\infty a$) $\vee\infty a$. L_3 היא שפת כל המילים שבהן a מופיע במקומות הזוגיים. $L_4=\infty a$. $L_5=\neg\infty a=(a+b)^*\circ b^\omega$. $L_6=\infty a\wedge\infty b$.

הערה. דרך נוספת לבניית אוטומט לשפה L_6.



איור 3: L_1 היא שפת כל המילים בהן $\#_a \geq 2$. L_2 היא (מס' זוגי של a $\wedge \neg\infty a) \vee \infty a$. L_3 היא שפת כל המילים שבהן a מופיע במקומות הזוגיים. $L_4 = \infty a$. $L_5 = \neg\infty a = (a+b)^* \circ b^\omega$. $L_6 = \infty a \wedge \infty b$.

הערה. לעומת אוטומטים מעל מילים סופיות, דואליזציה של DFA ‏($F \to Q \setminus F$) לא עובדת.

הגדרה. ביטוי ω-רגולרי הוא:

- $\emptyset$.
- $b_1 + b_2, r \circ b_1, r^\omega$, עבור ביטוי רגולרי r, וביטויים ω-רגולריים b_1, b_2.

משפט. ביטויים ω-רגולריים $=$ NBW.

דוגמה. דוגמאות לביטויים ω-רגולריים ושפות מתאימות.

$$\infty a \iff (b^* \circ a)^\omega$$

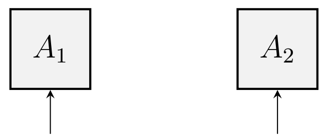
$$\neg\infty a \iff (a+b)^* \circ b^\omega$$

1.1 תכונות סגור

אוטומטי Buchi סגורים תחת איחוד, חיתוך והשלמה.

1.1.1 איחוד

בהינתן A_1, A_2 NBW, רוצים A NBW כך ש-$L(A) = L(A_1) \cup L(A_2)$.



איור 4: אוטומט האיחוד A, שמקיים $L(A) = L(A_1) \cup L(A_2)$.

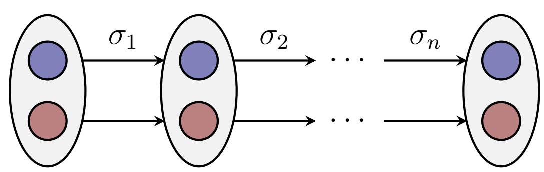
- בעיה: גם אם A_1, A_2 דטרמיניסטיים, A בהכרח אינו דטרמיניסטי.
- בהמשך נראה בנייה בה אוטומט האיחוד של שני DBW יהיה דטרמיניסטי.

1.1.2 חיתוך

בהינתן DBW A_1, A_2, רוצים DBW A כך ש-$L(A) = L(A_1) \cap L(A_2)$ (תקף גם ל-NBW).

<u>תזכורת</u>: עבור DFA-ים $A_i = (\Sigma, Q_i, \delta_i, q_0^i, F_i)$, $i = 1, 2$, נבנה את אוטומט המכפלה $A_1 \times A_2 = (\Sigma, Q_1 \times Q_2, \delta, (q_0^1, q_0^2), F_1 \times F_2)$, כאשר

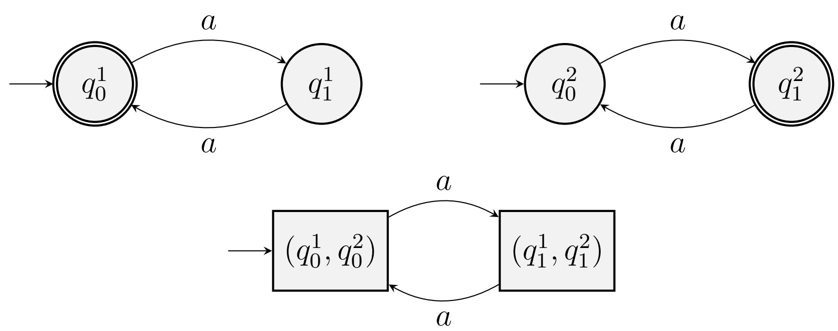
$$\delta((q,p), \sigma) = (\delta_1(q, \sigma), \delta_2(p, \sigma))$$



איור 5: ריצה על אוטומט המכפלה $A_1 \times A_2$ (מצבי A_1 בכחול, ו-A_2 באדום).

טענה. הבנייה הנ"ל לא תעבוד עבור DBW.

הוכחה. נסתכל על האוטומטים הבאים.



איור 6: משמאל לימין: A_1, A_2, $A_1 \times A_2$. ב-$A_1 \times A_2$ אין מצבים מקבלים! (למרות ש-$a^\omega \in L(A_1) \cap L(A_2)$).

$\square$

<u>אינטואיציה (בנייה שעובדת):</u>

- נתחזק שני עותקים של אוטומט המכפלה.
- נרצה להכריח את הריצה לבקר אינסוף פעמים גם במצבים מקבלים של A_1 וגם של A_2.
- נייצג זאת באמצעות מעבר מעותק אחד לאחר רק דרך סוג אחד של מצבים מקבלים.
- אם במהלך הריצה נתקענו באחד מהעותקים, משמע שלא מבקרים במצבים מקבלים של אחד מהאוטומטים.

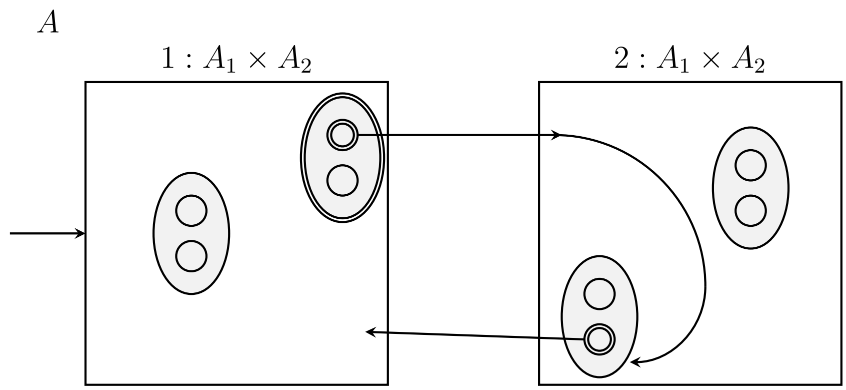
<u>סימונים:</u> למען הנוחות, נוכיח את נכונות הבנייה עבור DBW.

$$A_i = \left(\Sigma, Q_i, \delta_i, q_i^0, \alpha_i\right)$$

$$A = \left(\Sigma, Q_1 \times Q_2 \times \{1,2\}, \delta, \left(q_1^0, q_2^0, 1\right), \alpha_1 \times Q_2 \times \{1\}\right)$$

$$\delta\left(\left(q^1, q^2, i\right), \sigma\right) = \left(\delta_1\left(q^1, \sigma\right), \delta_2\left(q^2, \sigma\right), i'\right)$$

$$i' = \begin{cases} 1 & i = 2 \wedge q^2 \in \alpha_2 \\ 2 & i = 1 \wedge q^1 \in \alpha_1 \\ i & \text{אחרת} \end{cases}$$



איור 7: המחשה של בניית A. צד שמאל מכיל את המצבים $Q_1 \times Q_2 \times \{1\}$, וימין את $Q_1 \times Q_2 \times \{2\}$.

תרגיל. תהי מילה $w \in \Sigma^\omega$. נסמן את הריצה של A_1 על w ב-$r^1 = r_1^1 r_2^1 \ldots$, ושל A_2 על w ב-$r^2 = r_1^2 r_2^2 \ldots$. הוכיחו (באינדוקציה) כי הריצה של A על w היא

$$r = \begin{pmatrix} r_1^1 \\ r_1^2 \\ i_1 \end{pmatrix} \begin{pmatrix} r_2^1 \\ r_2^2 \\ i_2 \end{pmatrix} \ldots$$

טענה. $L(A) = L(A_1) \cap L(A_2)$.

הוכחה. נוכיח באמצעות הכלה דו-כיוונית.

- ($\supseteq$) אם $w \in L(A_1)$ וגם $w \in L(A_2)$ אז $w \in L(A)$. כלומר, r^1 וגם r^2 מקבלות $\Longleftarrow$ r מקבלת. סימונים:

$$I_j = \left\{ \alpha_j\text{-האינדקסים בהם } r^j \text{ מבקרת ב} \right\}$$

מאחר ו-r^j מקבלת, I_j אינסופית. בנוסף,

$$I_1 \underset{\text{אבחנה}}{\supseteq} I = \{\text{האינדקסים בהם } r \text{ מבקרת במצב מקבל}\}$$

נראה כי I אינסופית:

- <u>I לא ריקה:</u> נסמן ב-$l_1 = \min I_1$ את הביקור הראשון של r^1 ב-α_1.

$$r = \underbrace{\begin{pmatrix} r_0^1 \\ r_0^2 \\ 1 \end{pmatrix}}_{1} \cdots \underbrace{\begin{pmatrix} r_{l_1}^1 \\ r_{l_1}^2 \\ 1 \end{pmatrix}}_{l_1} \begin{pmatrix} r_{l_1+1}^1 \\ r_{l_1+1}^2 \\ \mathbf{2} \end{pmatrix}$$

ולכן $l_1 \in I$.

- לכל $l \in I$ קיים $l' > l$ כך ש-$l' \in I$: יהי $l \in I$. נגדיר $l'' = \min\{i \in I_2 \mid i > l\}$, ואת $l' = \min\{i \in I_1 \mid i > l''\}$, ונקבל:

$$r = \cdots \underbrace{\begin{pmatrix} r_l^1 \\ r_l^2 \\ 1 \end{pmatrix}}_{\text{מצב מקבל}} \begin{pmatrix} r_{l+1}^1 \\ r_{l+1}^2 \\ 2 \end{pmatrix} \cdots \underbrace{\begin{pmatrix} r_{l''}^1 \\ r_{l''}^2 \\ 2 \end{pmatrix}}_{\text{מצב מעבר (לא מקבל)}} \begin{pmatrix} r_{l''+1}^1 \\ r_{l''+1}^2 \\ 1 \end{pmatrix} \cdots \underbrace{\begin{pmatrix} r_{l'}^1 \\ r_{l'}^2 \\ 1 \end{pmatrix}}_{\text{מצב מקבל}}$$

ולכן r מקבלת.

- ($\subseteq$) אם $w \in L(A)$ אז $w \in L(A_1)$ וגם $w \in L(A_2)$. כלומר, אם r מקבלת אז r^1 וגם r^2 מקבלות.

  - r^1 מקבלת: קל. בכל פעם ש-r מבקרת במצב מקבל, גם r_1, ולכן r_1 מבקרת אינסוף פעמים ב-α_1 והיא מקבלת.
  - r^2 מקבלת: בין כל שני ביקורים ב-α חייב להיות ביקור של r^2 ב-α_2. יש אינסוף ביקורים ב-α ולכן אינסוף ביקורים ב-α_2.

□

1.1.3 השלמה

- ב-NFA: $\overline{\text{DFA}} \overset{\text{דואליזציה}}{\rightarrow} \text{DFA} \overset{\text{S.C.}}{\rightarrow} \text{NFA}$.

משפט. $\text{DBW} < \text{NBW}$.

הוכחה. נראה כי אין DBW עבור השפה $\neg\infty a$. נניח בשלילה שיש DBW D עם n מצבים כך ש-$L(D) = \neg\infty a$.

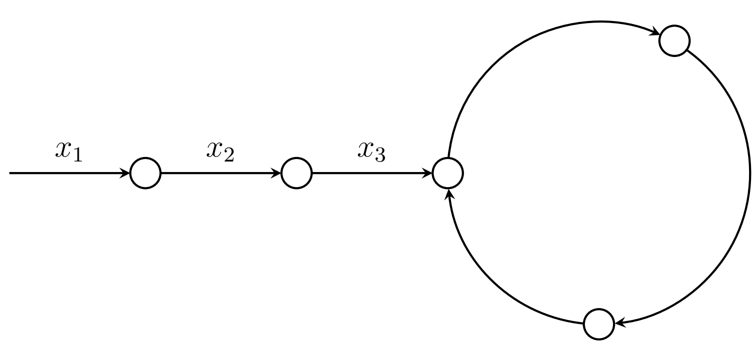
- מכאן, $b^\omega \in L(D)$. נסתכל על הריצה המקבלת של D על b^ω, $r_0, r_1, \dots$.
  - נסמן ב-n_1 את הביקור הראשון במצב מקבל

$$\underbrace{r_0, r_1, \dots, r_{n_1}}_{x_1}, \dots$$

- נסתכל על המילה $b^{n_1} \circ a \circ b^\omega \in L(D)$. מאחר והאוטומט דטרמיניסטי, הריצה תתחיל באופן זהה.

$$\underbrace{b, \dots, b}_{x_1}, a, \underbrace{b, \dots, b}_{x_2}, \dots$$

  - נסמן את אורך המסלול של x_2 ב-n_2.
- באופן דומה, נסתכל על המילה $b^{n_1}ab^{n_2}ab^\omega \in L(D)$. נחזור על התהליך $n+1$ פעמים.
  - לפי עיקרון שובך היונים, ביקרנו באותו מצב מקבל פעמיים.

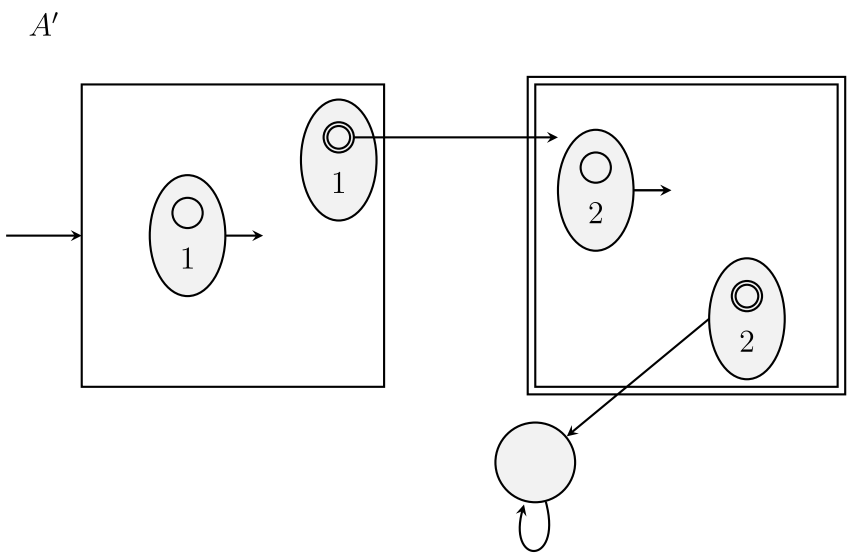


איור 8: דפוס הלאסו - עיגולים חלולים מסמנים מצבים מקבלים.

- נסמן ב-u את הרישא עד הלאסו, וב-v את האותיות שנקראות על הלאסו. אזי, ב-v יש לפחות a אחד.
- הלולאה של הלאסו מבקרת במצב מקבל, ולכן $uv^{\omega} \in L(D)$.
- בסך הכל, קיבלנו כי $uv^{\omega} \in L(D) \setminus (\neg\infty a)$, סתירה.

$\square$

- השלמת DBW: לא תמיד אפשר.
  - למשל, $\infty a \in$ DBW אבל $\neg\infty a \notin$ DBW.
  - דואליזציה לא תעזור - נקבל את ∞b.
- בהינתן DBW A, נבנה NBW A' כך ש-$L(A') = \overline{L(A)}$.
  - A' יכיל שני עותקים של A: מקבל ולא מקבל.



איור 9: האוטומט A', שמורכב מעותק מקבל ועותק לא מקבל של A.

- הניחוש מייצג: הריצה לא תראה יותר מצבים מקבלים.
- באופן פורמלי,

$$\text{DBW } A = (\Sigma, Q, \delta, q_0, \alpha) \longrightarrow \text{NBW } A' = (\Sigma, Q \times \{1,2\}, \delta', (q_0, 1), Q \times \{2\})$$

$$\delta'\left(\left(q,1\right),\sigma\right)=\left\{\left(\delta\left(q,\sigma\right),1\right),\left(\delta\left(q,\sigma\right),2\right)\right\}$$

$$\delta'\left(\left(q,2\right),\sigma\right)=\begin{cases}\left(\delta\left(q,\sigma\right),2\right) & q\notin\alpha\\ \underbrace{\emptyset}_{\text{הריצה נתקעת}} & q\in\alpha\end{cases}$$

טענה. $L\left(A'\right)=\overline{L\left(A\right)}$.

הוכחה. נוכיח את שני כיווני הטענה.

- $w\in L\left(A'\right)\Longleftarrow w\notin L\left(A\right)$.
  - אם $w\in L\left(A'\right)$, קיימת ריצה $r=\begin{pmatrix}\cdot\\1\end{pmatrix}\cdots\underbrace{\begin{pmatrix}\cdot\\1\end{pmatrix}}_{l}\underbrace{\begin{pmatrix}\cdot\\2\end{pmatrix}\begin{pmatrix}\cdot\\2\end{pmatrix}\cdots}_{r\text{ לא מבקרת ב-}\alpha}$ של A' על w.
  - החלק העליון של r הוא הריצה <u>היחידה</u> (A דטרמיניסטי) של A על w.
  - החל מ-l, r לא מבקרת יותר ב-α, ולכן $w\notin L\left(A\right)$.
- $w\in L\left(A\right)\Longleftarrow w\notin L\left(A'\right)$.
  - נניח בשלילה שיש ריצה מקבלת r של A' על w.
  - מכאן, r חייבת לנחש מעבר לעותק המקבל - נסמן את נקודת המעבר ב-l.
  - מאחר והריצה מקבלת ב-A, קיים $l'>l$ שבו הריצה ב-A מבקרת במצב מקבל, מה שיגרום ל-r להיתקע.

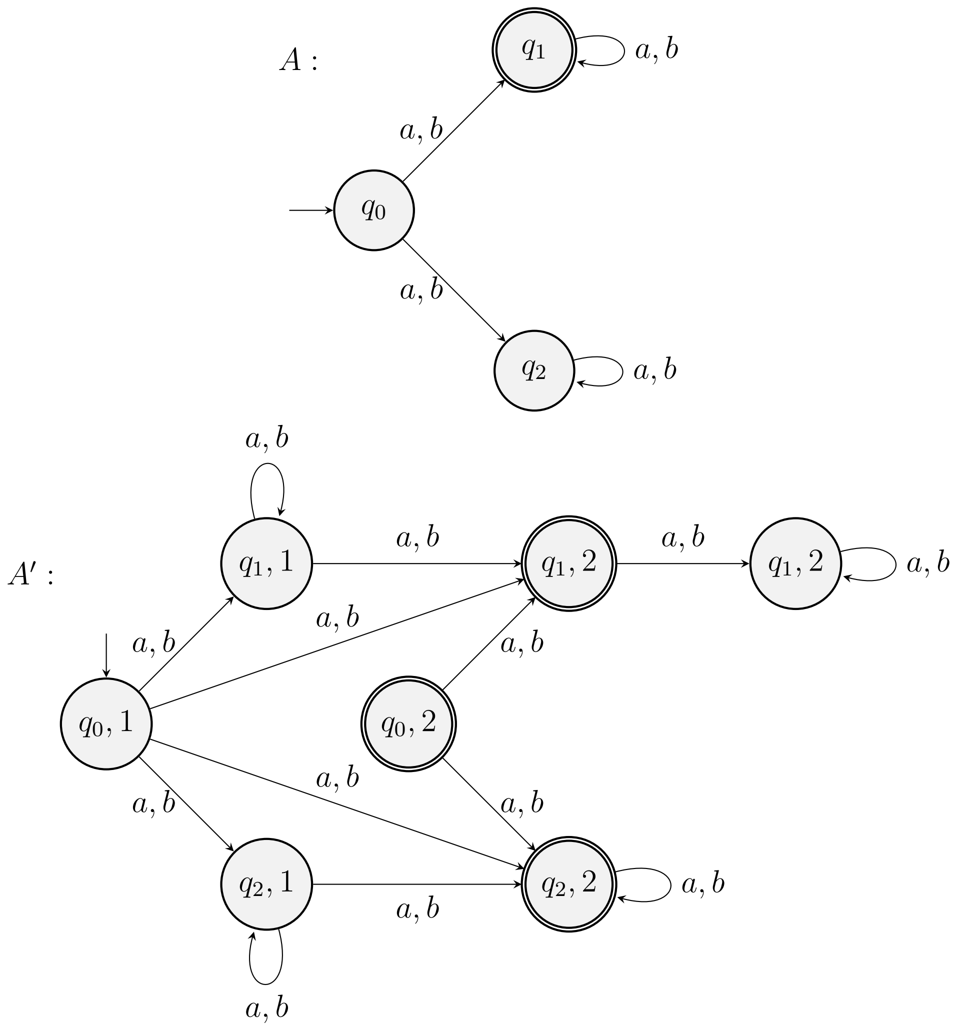
□

- האם הבנייה תעבוד על NBW?
  - אינטואיציה: אם היינו מגדירים באותה הצורה של A', השפה של A' הייתה

$$L\left(A'\right)=\left\{w\in\Sigma^{\omega}\mid \text{קיימת ריצה לא מקבלת של } A \text{ על } w\right\}$$

עם זאת, מאחר ו-A לא-דטרמיניסטי, ייתכן ש-$w\in L\left(A\right)$ וגם קיימת ריצה לא מקבלת של A על w.

- נפריך באמצעות דוגמא נגדית:



איור 10: ה-NFA A ו-A' המתאים לו. **לא מתקיים ש-**$L(A') = \overline{L(A)}$.

<u>תזכורת:</u> השלמת NFA.

משפט. השלמת NFA עם n מצבים אפשרית ב-$2^{\Theta(n)}$.

הוכחה. נוכיח חסם עליון וחסם תחתון.

- <u>חסם עליון:</u> האלגוריתם הבא מבצע השלמת NFA ב-$2^{\Theta(n)}$.

$$\underbrace{\text{NFA A}}_{n \text{ מצבים}} \xrightarrow{\text{S.C.}} \underbrace{\text{DFA } A'}_{2^n \text{ מצבים}} \xrightarrow{\text{דואליזציה}} \underbrace{\text{DFA } A'}_{2^n \text{ מצבים}}$$

$$L\left(A'\right) = \overline{L\left(A\right)}$$

- <u>חסם תחתון:</u> אין אלגוריתם "יותר טוב": שבונה NFA משלים עם $2^n >$ מצבים.
<u>סכמה כללית להוכחות שכאלה:</u>

  1. להגדיר משפחת שפות $\{L_n \mid n \in \mathbb{N}\}$.
  2. להראות כי קיים NFA "קטן" שמזהה את L_n.
  3. להוכיח כי כל NFA שמזהה את $\overline{L_n}$ הוא "גדול".

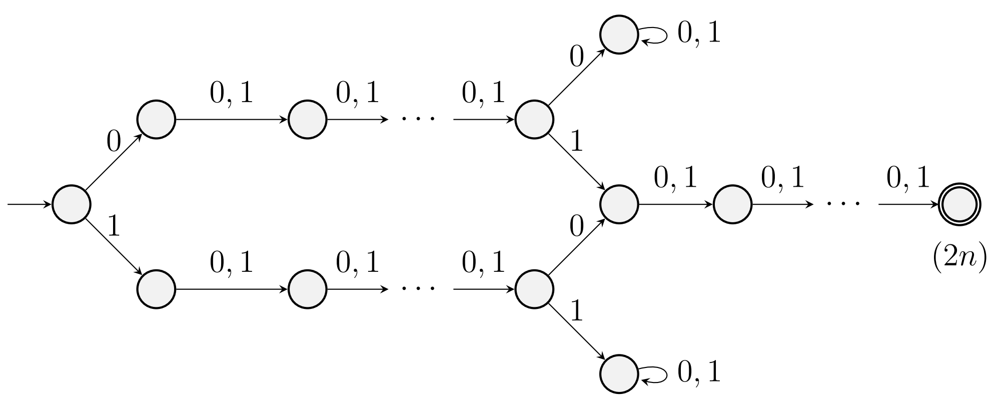
- נעבוד ע"פ הסכמה.

  1. נגדיר את L_n באופן הבא:

$$\Sigma = \{0, 1\}$$

$$L_n = \{u \circ v \mid u, v \in \Sigma^n \wedge u \neq v\}$$

  2. לכל n, קיים NFA קטן שמזהה את L_n. אם היינו צריכים לבדוק שוני באינדקס הראשון, האוטומט היה נראה כמתואר באיור 11.



איור 11: אוטומט שמזהה את השפה: מילים באורך $2n$ בהן האות הראשונה שונה מהאות ה-$n+1$.

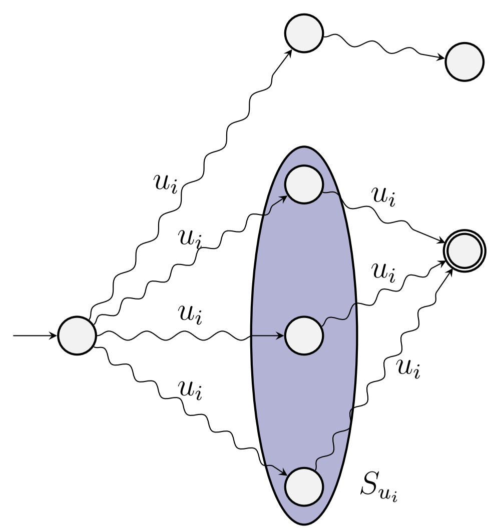
באופן דומה, נבנה n אוטומטים $\{A_i\}_{i=1}^{n}$, כאשר A_i מזהה שוני באינדקס i, ונגדיר את $A^n = \bigcup_{i=1}^{n} A_i$. A^n קטן: $|A^n| = \mathcal{O}\left(n^2\right)$.

3. נמצא את $\overline{L_n}$:

$$\overline{L_n} = \{u \cdot u \mid u \in \Sigma^n\} \cup \bigcup_{m \neq 2n} \Sigma^m$$

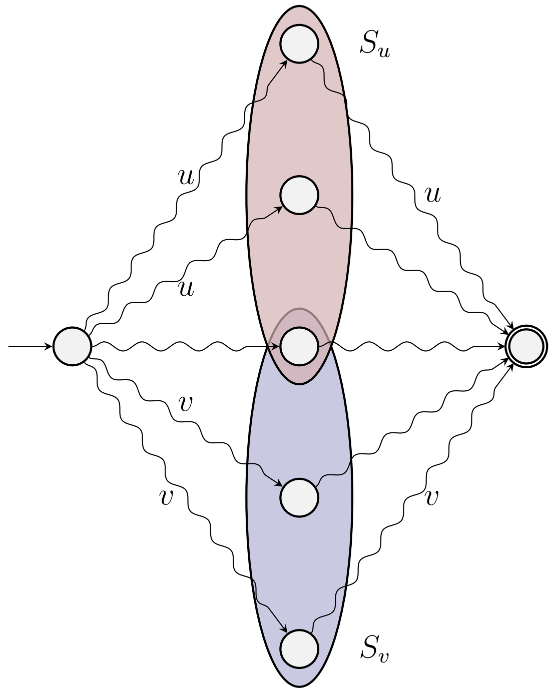
נראה כי כל NFA שמזהה את $\overline{L_n}$, גודלו $\leq 2^n$ (טענה יותר חלשה, אך נסתפק בה):

- נניח בשלילה שקיים NFA A, כך ש-$|A| < 2^n$ ומזהה את $\overline{L_n}$.
- קיימים 2^n וקטורים בינאריים באורך n, $u_0, \ldots, u_{2^n-1}$. נסתכל על מילה $u_i u_i \in \overline{L_n}$.



איור 12: הקבוצה $S_{u_i} = \{q \in \delta^*(u_i) \wedge \delta^*(q, u_i) \cap F \neq \emptyset\}$ בכחול, כל המצבים שאפשר להגיע אליהם ע"י קריאת u_i, ואפשר להגיע מהם למצב מקבל בקריאת u_i נוסף.

- אבחנה: לכל u_i, $S_{u_i} \neq \emptyset$. לכן, לפי עיקרון שובך היונים, קיימים $u \neq v$ כך ש-$S_u \cap S_v \neq \emptyset$.



איור 13: u ו-v כך ש-$S_u \cap S_v \neq \emptyset$.

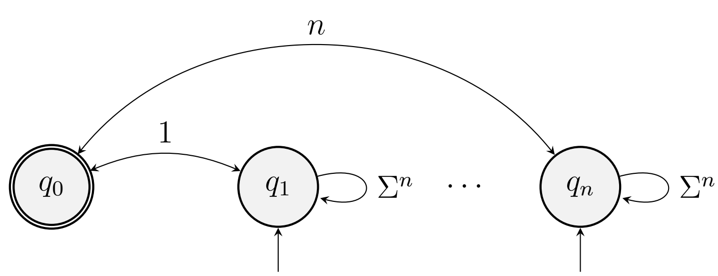
- מכאן, נקבל ש-$L(A) \ni u \circ v$: ניתן להגיע למצב כלשהו ב-$S_u \cap S_v$ ע"י קריאת u, ולהמשיך ממנו למצב מקבל ע"י קריאת v, והגענו לסתירה.

$\square$

1.1.4 השלמת NBW

משפט. השלמת NBW בן n מצבים אפשרית ב-$2^{\Theta(n\log n)}$.

- חסם עליון: לא נראה, הבנייה של ספרא/קופרמן-ורדי.
- חסם תחתון: נוכיח לפי השלבים:
  1. הגדרת משפחת שפות $\{L_n \mid n\in\mathbb{N}\}$.
  2. בניית NBW "קטן" שמזהה את L_n.
  3. הוכחה כי כי NBW שמזהה את $\overline{L_n}$ הוא "גדול".
- נתחיל מ-2: נסתכל על האוטומט A_n, שמתואר באיור 14, תחת $\Sigma=\{1,\ldots,n,\#\}$.



איור 14: האוטומט A_n.

נגדיר $L_n=L\left(A_n\right)$, ונשים לב ש-$|A_n|=n+1$. מהי L_n? אילו מילים יש בה?

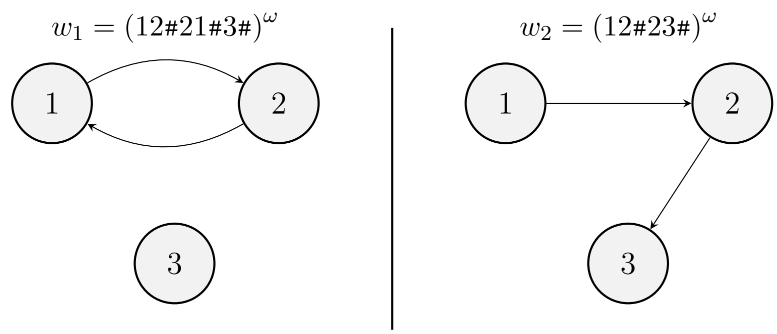
$$(11\#)^\omega\in L_n$$
$$(12\#21\#)^\omega\in L_n$$
$$(12\#23\#31\#)^\omega\in L_n$$
$$(12\#23\#)^\omega\notin L_n$$

<u>אפיון L_n.</u>

- נרצה "אלגוריתם", שבהינתן מילה $w\in\Sigma^\omega$ מכריע האם $w\in L_n$.
- נבנה גרף (מכוון) $G_w=(V,E)$, כך ש-$V=\{1,\ldots,n\}$ ו-

$$E=\{(i,j)\mid \text{הרצף } ij \text{ מופיע } \infty \text{ פעמים ב-} w\}$$

דוגמה. מילים w והגרף G_w המושרה מהן, באיור 15.



איור 15: דוגמאות למילים w והגרף G_w המושרה מהן.

טענה. $w \in L_n \iff$ ב-G_w יש מעגל.

הוכחה. נוכיח את שני כיווני הטענה.

- ($\Rightarrow$) נניח כי המעגל הוא $i_0 \to i_1 \to \cdots \to i_k \to i_0$. נתאר ריצה מקבלת של A על w: כאשר המיקום בגרף יהיה i_j, נרצה שהריצה תהיה במצב q_{i_j}.
  - <u>בסיס:</u> המיקום ההתחלתי הוא i_0, וכך הריצה מתחילה ב-q_{i_0}.
  - <u>צעד:</u> נניח שהמיקום הוא i_j והריצה ב-q_{i_j}. הקשת $i_j \to i_{j+1}$ נמצאת ב-$E\left(G_w\right)$, ולכן הרצף $i_j i_{j+1}$ מופיע ∞ פעמים ב-w. הריצה "תמתין" בלולאה העצמאית של q_{i_j} עד למופע הבא של $i_j i_{j+1}$, ואז תעבור ל-$q_{i_{j+1}}$ והמיקום יעודכן ל-i_{j+1}, והריצה תיראה כך:

$$q_{i_0} \underbrace{\xrightarrow{\Sigma^*}}_{i_0, i_1 \text{ נדלג עד}} q_{i_0} \xrightarrow{i_0} q_0 \xrightarrow{i_1} q_{i_1} \underbrace{\xrightarrow{\Sigma^*}}_{i_1, i_2 \text{ נדלג עד}} q_{i_1} \xrightarrow{i_1} q_0 \xrightarrow{i_2} q_{i_2} \to \cdots$$

  אבחנות:

  1. הריצה לא נתקעת.
  2. המיקומים מטיילים על מעגל ב-G_w, ולכן יכולים לטייל עליו ∞ פעמים.
  3. בכל פעם שמיקום מתעדכן, הריצה מבקרת במצב מקבל.

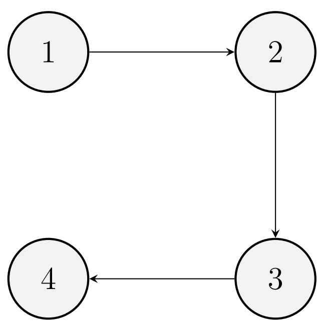
  - מכאן, הריצה מקבלת.

- ($\Leftarrow$) תהי r ריצה מקבלת של A_n על w, נמצא מעגל ב-G_w.

- r מקבלת, ולכן עוברת ב-q_0 ∞ פעמים. נסמן ביקור ב-q_0, מהצורה $q_i \to q_0 \to q_{i'}$, בתור $x_j = (i, i')$, ונסמן $I = \{x_i\}_{i=1}^{\infty}$.
- יש n^2 רצפים מהצורה $q_i \to q_0 \to q_{i+1}$, ולכן חייב להיות רצף $q_{i_0} \xrightarrow{i_0} q_0 \xrightarrow{i_1} q_{i_1}$ שמופיע ∞ פעמים ב-r.
- אם $i_0 = i_1$, סיימנו - מצאנו מעגל.
- אחרת, נסתכל על $I \supseteq I' = \{x_j \mid x_j = (i_0, i_1)\}$, שהיא אינסופית. לכל רצף $i_0 i_1$, חייב להיות רצף $j i_0$ שדרכו הריצה נכנסה ל-i_0.
- מאחר ויש מספר סופי של j-ים, קיים j עבורו יש אינסוף רצפים מהצורה $j i_0 i_1$. אם $j \in \{i_0, i_1\}$ מצאנו מעגל, ואחרת נמשיך.
- מאחר ויש מספר סופי של מצבים, בשלב מסוים נחזור בשנית לאותו הקודקוד, ונמצא מעגל.

$\square$

מסקנה. עבור פרמוטציה $\pi \in S_n$, מתקיים $(\pi\#)^{\omega} \notin L_n$. לדוגמא, נסתכל על המילה $w \in (1234\#)^{\omega}$, ועל הגרף G_w המתאים לה באיור 16, שהוא חסר מעגלים.



איור 16: הגרף G_w עבור המילה $(1234\#)^{\omega}$.

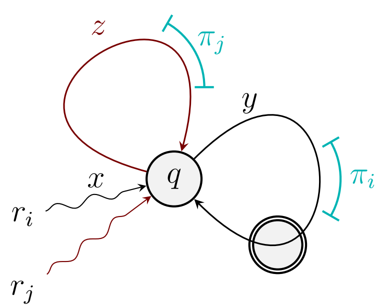
טענה. כל אוטומט שמזהה את $\overline{L_n}$, גודלו $2^{\Omega(n \log n)}$.

הוכחה. נניח בשלילה שקיים $\overline{A_n}$ בגודל $< n!$ שמזהה את $\overline{L_n}$, ונסמן $S_n = \{\pi_1, \ldots, \pi_{n!}\}$. לכל פרמוטציה $\pi_i \in S_n$, נסתכל על הריצה המקבלת r_i של $\overline{A_n}$ על $(\pi_i\#)^{\omega}$, ועל $\inf(r_i)$:

$$\begin{array}{lll}
\pi_1 = \begin{pmatrix} 1 & 2 & \cdots & n \end{pmatrix} & r_1 & \inf(r_1) \\
\pi_2 = \begin{pmatrix} 2 & 1 & \cdots & n \end{pmatrix} & r_2 & \inf(r_2) \\
\vdots & \vdots & \vdots \\
\pi_{n!} = \begin{pmatrix} n & n-1 & \cdots & 1 \end{pmatrix} & r_{n!} & \inf(r_{n!})
\end{array}$$

מהמסקנה הקודמת, $\{(\pi_i\#)^\omega\}_{i=1}^{n!} \subseteq L\left(\overline{A_n}\right)$, ולכן לכל $1 \le i \le n!$: $|\text{inf}\,(r_i)| \ge 1$.

- מחד גיסא, $\left|\overline{A_n}\right| < n!$, ומאידך יש $n!$ פרמוטציות, ולכל אחת $\text{inf}\,(r_i) \ge 1$.
- מכאן, לפי עיקרון שובך היונים, קיימים π_i, π_j כך ש-$\text{inf}\,(r_i) \cap \text{inf}\,(r_j) \ne \emptyset$.
- נרצה למצוא מילה ב-L_n ש-$\overline{A_n}$ מקבל. יהי $q \in \text{inf}\,(r_i) \cap \text{inf}\,(r_j)$.
- הריצות r_i ו-r_j עוברות מעגל גדול כרצוננו שחוזר ל-q.
- נבחר מעגלים y ו-z, כך שבמעבר על y קוראים π_i, במעבר על z קוראים π_j, ו-y עובר במצב מקבל (מאחר ומבקרים ∞ פעמים במצב מקבל, קיים מעגל כזה), כמתואר באיור 17.

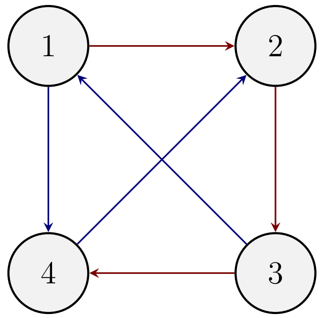


איור 17: $i \ne j$ כך ש-$q \in \text{inf}\,(r_i) \cap \text{inf}\,(r_j)$.

- נתבונן במילה $x\,(yz)^\omega \in L\left(\overline{A_n}\right)$.
  - מאיור 17 ניתן לראות מסלול מקבל.
  - מצד שני, ב-$G_{x(yz)^\omega}$ יש מעגל (תרגיל) $\Leftarrow$ $x\,(yz)^\omega \in L_n$. סתירה!

$\square$

דוגמה. אינטואיציה להוכחה, נסתכל על $\pi_j = (3142\#)^\omega\,, \pi_i = (1234\#)^\omega$, ועל הגרפים המתאימים באיור 18.



איור 18: G_w עבור π_i (אדום) ו-π_j (כחול). בין פרמוטציות שונות יש זוג אחד לפחות שמתחלף, מה שגורם למעגל.

מסקנה. כל $\overline{A_n}$ שמזהה את $\overline{L_n}$ גודלו $\leq n!$:

$$\left|\overline{A_n}\right| \geq 2^{\log n!} = 2^{\Omega(n \log n)}$$

2 תנאי קבלה נוספים

סינטקס: $A = (\Sigma, Q, \delta, Q_0, \alpha)$. ריצה r היא מקבלת אם:

- סמנטיקת Buchi: r מבקרת ∞ פעמים ב-α.
- סמנטיקת co-Buchi: r מבקרת $\neg\infty$ פעמים ב-α.
- בהמשך, נראה את Parity, Generalized Buchi, Streett, Rabin.

נשתמש בקיצורים למחלקות:

$$\underbrace{\left\{\underbrace{N}_{\text{Nondeterm.}}, \underbrace{D}_{\text{Determ.}}, \underbrace{A}_{\text{Alternating}}, \underbrace{U}_{\text{Universal}}\right\}}_{\text{פיצול באוטומט}} \times \underbrace{\{B, C, GB, P, R, S\}}_{\text{תנאי קבלה}} \times \underbrace{\left\{\underbrace{W}_{\text{Word}}, \underbrace{T}_{\text{Tree}}\right\}}_{\text{מה קוראים}}$$

למשל, NPW עבור Nondeterministic Parity Word Automaton.

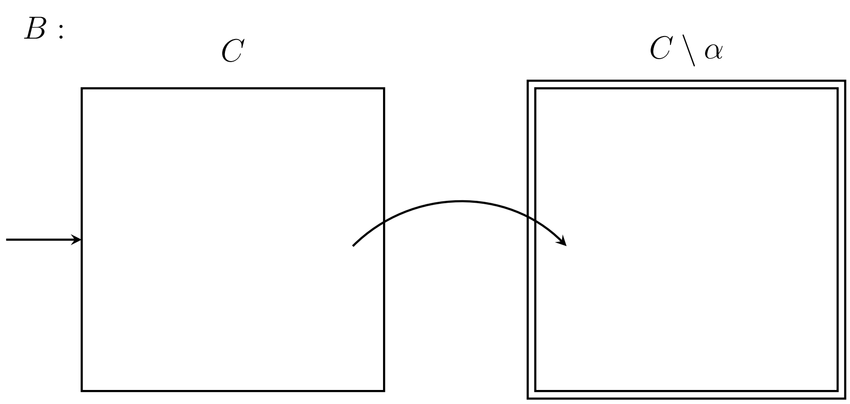
2.1 co-Buchi

- Buchi.
  - $\alpha\subseteq Q$.
  - ריצה מקבלת $\iff$ מבקרת ב-α ∞ פעמים.
- co-Buchi.
  - $\alpha\subseteq Q$.
  - ריצה מקבלת $\iff$ מבקרת ב-α מספר סופי של פעמים (באופן שקול, היא בסופו של דבר נתקעת ב-$Q\setminus\alpha$).

משפט. לכל $L\subseteq\Sigma^{\omega}$, $\overline{L}\in\mathrm{DCW}\iff L\in\mathrm{DBW}$.

טענה. יהי C NCW. אזי קיים B NBW כך ש-$L\left(B\right)=L\left(C\right)$.

הוכחה. בהינתן C, נבנה את B באופן הבא



איור 19: ה-NBW B כך ש-$L\left(B\right)=L\left(C\right)$.

המשך ההוכחה זהה להוכחה של בניית המשלים של NBW. $\square$

מסקנה. $\infty a\notin\mathrm{DCW}$. אחרת, קיים DCW עבור ∞a, וכך קיים DBW עבור $\neg\infty a$, סתירה!

2.2 Generalized Buchi

- $\alpha = \{\alpha_1, \ldots, \alpha_k\}$, $\alpha_i \subseteq Q$.
- ריצה מקבלת $\Longleftrightarrow$ מבקרת ∞ פעמים בכל α_i.

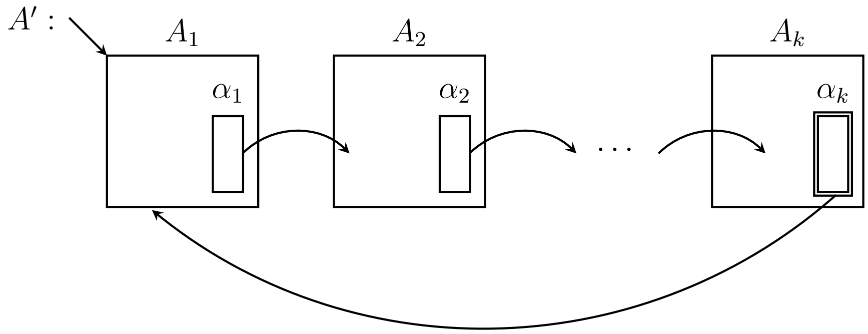
משפט. $\text{NBW} = \text{NGBW}$.

הוכחה. נראה הכלה דו-כיוונית.

- $(\subseteq)$ בהינתן NBW $A = (\Sigma, Q, \delta, Q_0, \alpha)$, נתרגם סינטקטית ל-NGBW A':

$$A' = (\Sigma, Q, \delta, Q_0, \{\alpha\})$$

- $(\supseteq)$ בהינתן NGBW $A = (\Sigma, Q, \delta, Q_0, \alpha)$, נבנה NBW A'. לכל $1 \leq i \leq k$ נגדיר $A_i = (\Sigma, Q, \delta, Q_0, \alpha_i)$, וכך A' הוא $\bigcap_{i=1}^{k} A_i$, כמתואר באיור 20.



איור 20: ה-NBW A' עבור ה-NGBW A.

נשים לב שאם $|A| = n$, אז $|A'| = nk$.

$\square$

2.3 Rabin

- $\alpha = \{(B_1, G_1), \ldots, (B_k, G_k)\}$, כך ש-$B_i, G_i \subseteq Q$.
- ריצה מקבלת $\Longleftrightarrow$ קיים i כך ש-r מבקרת ∞ פעמים ב-G_i, וגם $\neg\infty$ פעמים ב-B_i.

2.4 Streett

הדואלי של Rabin. נרצה להגדיר אותו באופן דומה לדואליות של DBW ו-DCW: אוטומט DRW עם תנאי קבלה Streett יקבל את השפה המשלימה.

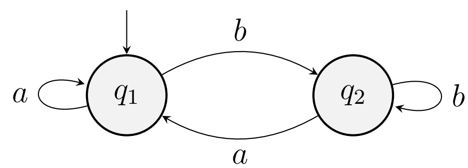
- $\alpha = \{(B_1, G_1), \ldots, (B_k, G_k)\}$, כך ש-$B_i, G_i \subseteq Q$.
- ריצה מתקבלת $\iff$ לכל i, r מבקרת $\neg\infty$ פעמים ב-G_i, או ∞ פעמים ב-B_i.

משפט. NBW $=$ DRW.

הערה. לא נראה את ההוכחה - ספרא, 8x':

$$\underset{n}{\text{NBW}} \to \underset{2^{\mathcal{O}(n \log n)}}{\text{DRW}}$$

דוגמה. השפה $\neg\infty a \in$ NBW $\setminus$ DBW. נבנה DRW שמזהה את השפה.



איור 21: אוטומט DRW עבור השפה $\neg\infty a$, עם תנאי הקבלה $B_1 = \{q_1\}, G_1 = Q$. כדי לקבל את השפה ∞a, נגדיר $B_1 = \emptyset, G_1 = \{q_1\}$.

הערה. כאשר $B_i = \emptyset$, ההגבלה שקולה ל-Buchi.

משפט. NBW $=$ NRW.

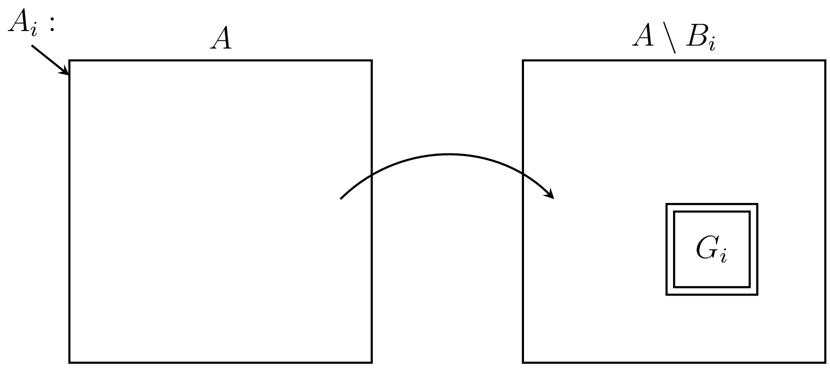
הוכחה. נראה הכלה דו-כיוונית.

- $(\subseteq)$ בהינתן NBW $A = (\Sigma, Q, \delta, Q_0, \alpha)$, נתרגם סינטקטית ל-NRW A':

$$A' = (\Sigma, Q, \delta, Q_0, \{(\emptyset, \alpha)\})$$

- $(\supseteq)$ בהינתן NRW A עם תנאי קבלה $\{(B_1, G_1), \ldots, (B_k, G_k)\}$,

- נבנה k אוטומטים $\{A_i\}_{i=1}^{k}$, כך שבאופן אינטואטיבי, "r מתקבלת ב-A_i $\iff$ r ב-A מספקת את (B_i, G_i)", כמו באיור 22.



איור 22: ה-NBW A' עבור ה-NRW A.

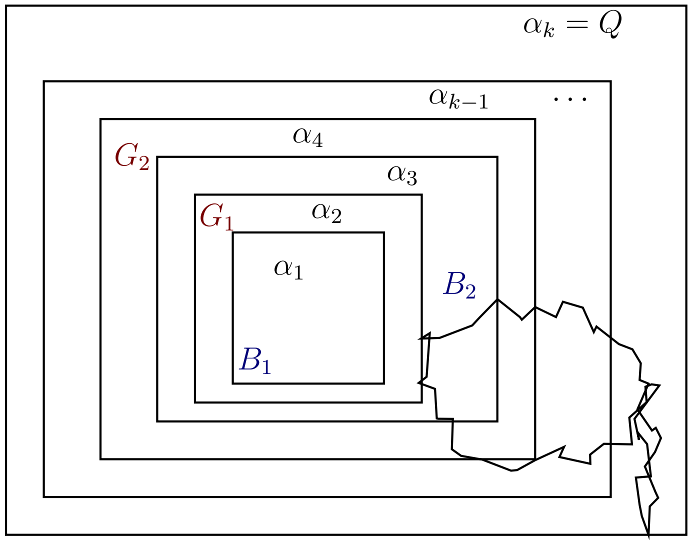
הניחוש האי-דטרמיניסטי מסמל את הנקודה שהחל ממנה לא נגיע יותר למצבים ב-B_i. המצבים המקבלים בעותק השני יהיו G_i. A' הוא אוטומט האיחוד של A_i.

- נשפר את מספר המצבים ע"י עותק אחד של A, וניחוש אי-דטרמיניסטי לאחד מהעותקים השניים. מ-$2nk$ ל-$(k+1)n$.

$\square$

2.5 Parity

- $\alpha = \{\alpha_1, \dots, \alpha_k\}$, $\alpha_1 \subseteq \alpha_2 \subseteq \cdots \subseteq \alpha_k = Q$.
- ריצה r מתקבלת $\iff$ $\min_i \{\inf(r) \cap \alpha_i \neq \emptyset\}$ הוא זוגי, כמתואר באיור 23.



איור 23: הקבוצות α_i בתנאי הקבלה Parity.

- מקרה פרטי של Rabin: ה-α_i הזוגיים יהיו ה-G-ים, והאי-זוגיים יהיו ה-B-ים.
- נשים לב שאם $\inf(r) \cap \alpha_i \neq \emptyset$, אז גם $\inf(r) \cap \alpha_j \neq \emptyset$ לכל $j > i$. אם ה-i המינימלי שמקיים זאת הוא זוגי, הזוג (B, G) המתאים לו ול-α_{i-1} יקבל ב-Rabin.

טבלה 1 מסכמת את תנאי הקבלה לעיל.

תנאי קבלה	מצבים מקבלים	ריצה r מקבלת
Buchi	$\alpha \subseteq Q$	$\inf(r) \cap \alpha \neq \emptyset$
co-Buchi	$\alpha \subseteq Q$	$\inf(r) \cap \alpha = \emptyset$
Generalized Buchi	$\alpha = \{\alpha_i\}_{i=1}^{k}, \alpha_i \subseteq Q$	$\forall i : \inf(r) \cap \alpha_i \neq \emptyset$
Rabin	$\alpha = \{(B_i, G_i)\}_{i=1}^{k}, B_i, G_i \subseteq Q$	$\exists i : G_i \cap \inf(r) \neq \emptyset \wedge B_i \cap \inf(r) = \emptyset$
Streett	$\alpha = \{(B_i, G_i)\}_{i=1}^{k}, B_i, G_i \subseteq Q$	$\forall i : G_i \cap \inf(r) = \emptyset \vee B_i \cap \inf(r) \neq \emptyset$
Parity	$\alpha = \{\alpha_i\}_{i=1}^{k}, \alpha_i \subseteq \alpha_{i+1}$	$\min_i \{\inf(r) \cap \alpha_i\} \neq \emptyset$ הוא זוגי.

טבלה 1: תנאי קבלה לאוטומטים מעל מילים אינסופיות.

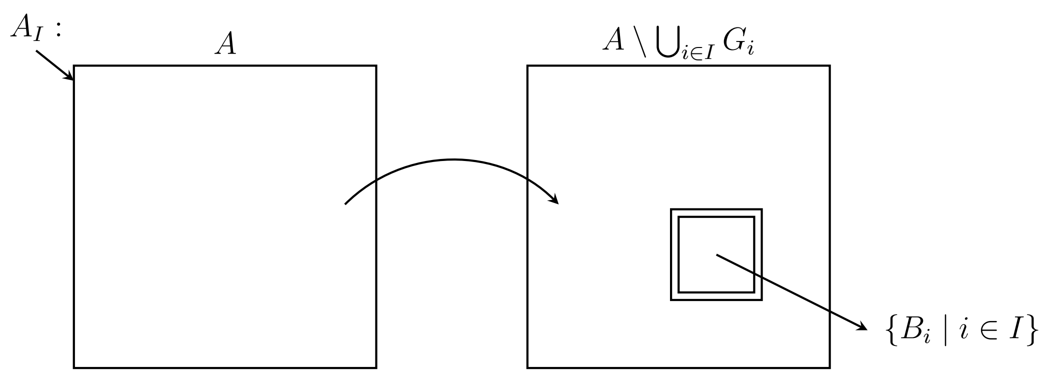
משפט. $\mathrm{DPW} = \mathrm{NBW}$.

הוכחה. דומה להוכחת $\mathrm{DRW} = \mathrm{NBW}$. $\square$

משפט. $\mathrm{NBW} = \mathrm{NSW}$.

הוכחה. כיוון $\subseteq$ קל - שינוי סינטקטי. הכיוון $\supseteq$ דומה ל-$\mathrm{NRW} = \mathrm{NBW}$, אך עם שלילה.

- ננחש $I \subseteq \{1, \dots, k\}$. נגדיר A_I, כך שריצה r מקבלת ב-A_I אם $i \in I \Rightarrow$ r מבקרת ∞ פעמים ב-B_i, r מבקרת $\neg\infty$ פעמים ב-G_i $i \notin I \Rightarrow$".
- נגדיר $A' = \bigcup_I A_I$, A_I בנוי כ-NGBW באיור 24, ומכאן קל להמיר ל-NBW.



איור 24: ה-NBW A_I.

- כ-NGBW, האוטומט יהיה בגודל $|A'| = 2n \cdot 2^k$, וכ-NBW ב-$2n \cdot k \cdot 2^k$.

$\square$

הערה. ספרא: זהו חסם הדוק לגודל A'.

- בסך הכל, השלמת ספרא:

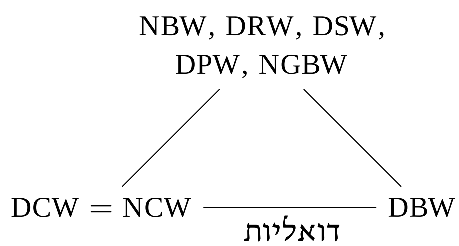
$$\underset{n}{\mathrm{NBW}} \to \underset{\underbrace{2^{\mathcal{O}(n\log n)}}_{\text{מצבים}},\ \underbrace{n}_{k}}{\mathrm{DRW}} \to \underset{2^{\mathcal{O}(n\log n)},n}{\mathrm{DSW}} \to \underset{2^{n\log n}\cdot 2^n = 2^{\mathcal{O}(n\log n)}}{\overline{\mathrm{NBW}}}$$

פיצוץ מצבים!

3 פיצוץ מצבים

- עד כה, דנו ב-

1. Expresivity: מה כל מחלקה יכולה להביע, כמסוכם באיור 25.



איור 25: יחסים בין המחלקות השונות.

2. Succinctness: מה המחיר במעבר בין מחלקות - פיצוץ מצבים.

3.1 Succinctness

הגדרה. נאמר שמחלקה (class) של אוטומטים $\mathcal{C}$ היא $\mathcal{C}'$-type אם לכל שפה L כך ש-$L \in \mathcal{C} \cap \mathcal{C}'$, לכל אוטומט $A = (\Sigma, Q, \delta, Q_0, \alpha)$ עבור L במחלקה $\mathcal{C}$ יש אוטומט $A' = (\Sigma, Q, \delta, Q_0, \alpha')$ עבור L במחלקה $\mathcal{C}'$.

משפט. DBW הם DCW-type.

הוכחה. שאלה 4 בתרגיל. ניסוח מקורי: עבור שפה $L \in$ DBW, כך ש-$\underbrace{\overline{L} \in \text{DBW}}_{\iff L \in \text{DCW}}$, בהינתן DBW ל-$L$ ניתן לשנות את α (מתייחסים לתנאי קבלה co-Buchi) כך שיתקבל DBW עבור המשלים. אפשר להראות גם ש-DRW הם DBW-type. $\square$

טענה. DSW הם <u>לא</u> DBW-type.

3.1.1 תרגום NBW → NCW

הערה. לא תמיד אפשר לתרגם NBW ל-NCW.

<u>היסטוריה:</u>

- חסם עליון, ע"י ספרא, $\underset{n}{\text{NBW}} \to \underset{2^{O(n \log n)}}{\text{NCW}}$.

- חסם תחתון, לא ידעו האם NBW הם NCW-type.
- [Aminof, Kuperman, Lev]: קיימת משפחה L_n שאפשר לזהות ע"י NBW עם $2n$ מצבים, וכל NCW שמזהה את L_n, $3n$ מצבים: $\Omega(1.5n)$.
- [Boker, Kuperman] $\underset{n}{\text{NBW}} \to \underset{O(n\cdot 2^n)}{\text{NCW}}$ + חסם תחתון הדוק.

טענה. (חסם תחתון לא הדוק) תרגום מ-NBW ל-NCW (כשזה אפשרי) מצריך פיצוץ של לפחות $\Omega(n^2)$.

<u>שלבי ההוכחה:</u>

1. הגדרת L_n לכל n.
2. שני חלקים:
   (א) יש NBW "קטן" שמזהה את L_n.
   (ב) אפשר לזהות את L_n ע"י NCW.
3. כל NCW שמזהה את L_n הוא גדול.

הוכחה. נוכיח לפי השלבים.

1. לכל $k \in \mathbb{N}$, נגדיר

$$S_k := \{i \cdot k + j \cdot (k+1) \mid i, j \in \mathbb{N}, i + j > 0\}$$

למשל, עבור $k = 4$:

$$S_4 = \{4 = 4 \cdot 1, 5 = 5 \cdot 1, 8 = 4 \cdot 2, 9 = 4 \cdot 1 + 5 \cdot 1, 10 = 5 \cdot 2, 12 = 4 \cdot 3, \dots\}$$

הגדרה. נגדיר $\text{th}(k) := k^2 - k - 1$. למשל $\text{th}(4) = 4^4 - 4 - 1 = 11$.

טענה. $\text{th}(k) \notin S_k$, וגם לכל $l > \text{th}(k)$ מתקיים $l \in S_k$ (קל להוכיח באינדוקציה).

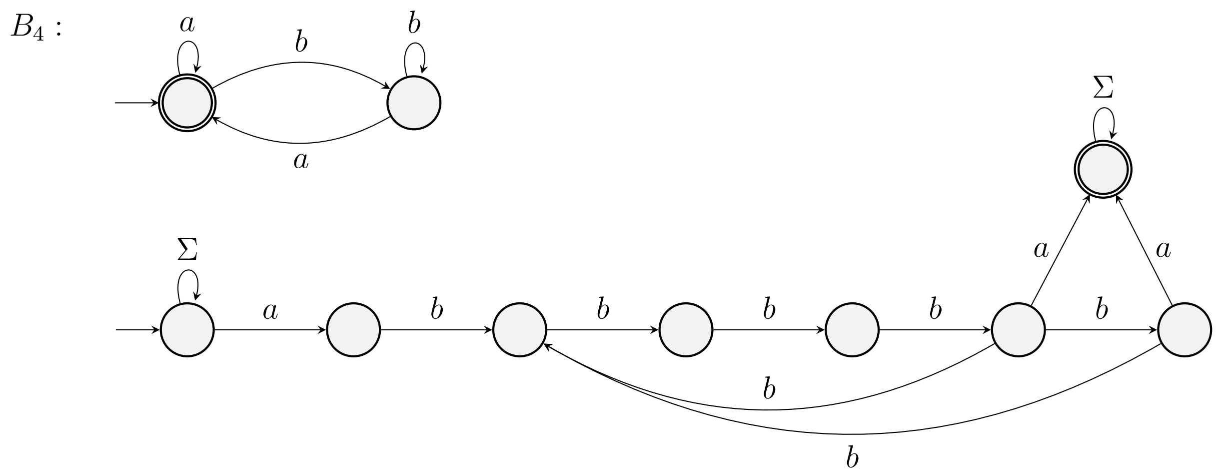
כעת, עבור $\Sigma = \{a, b\}$ נגדיר לכל n: $L_n := \infty a \cup \left\{\Sigma^* a b^l a \Sigma^\omega \mid l \in S_n\right\}$.
נסתכל על L_4:

$$b^\omega \notin L_4$$
$$aaaabbbbbab\cdots \in L_4$$
$$(ab^6a)^\omega \in L_4$$
$$ab^{11}ab^\omega \notin L_4$$

עבור מילים מהצורה $ab^{11}b\cdots b\cdots$: אם בהמשך יש b^ω, המילה לא בשפה. אחרת, יש a, ומהטענה הקודמת המילה בשפה.

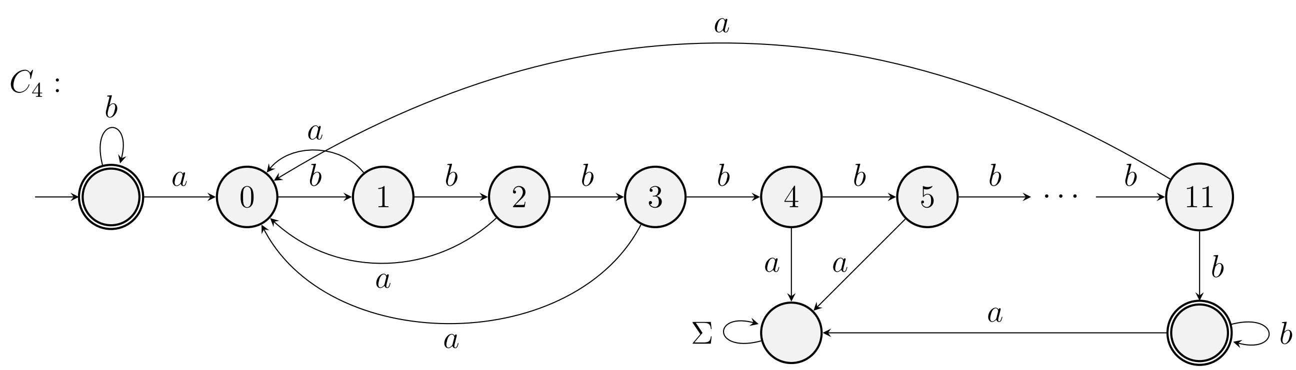
2.

(א) נזהה את L_k עם NBW ״קטן״. נבנה אוטומט עבור ∞a, ועבור החלק השני ננחש את תחילת הרצף המעניין ואת המקדמים i, j, כמתואר באיור 26. בנוסף, $|B_k| = \mathcal{O}(k)$.



איור 26: האוטומט B_4 עבור השפה L_4.

(ב) נבנה NCW עבור L_4, C_4, כמתואר באיור 27, בגודל $\text{th}(k) = \mathcal{O}(k^2)$.



איור 27: האוטומט B_4 עבור השפה L_4.

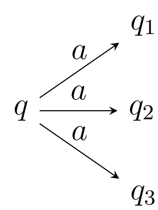
3. בבית.

□

4 אוטומטים מתחלפים

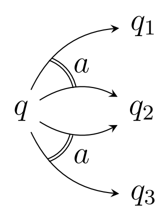
4.1 סינטקס

- אוטומט NFA, כמתואר באיור 28, מקבל את המילה $a \circ x$ אם קיים שכן q_i כך שהמילה x מתקבלת מ-q_i.



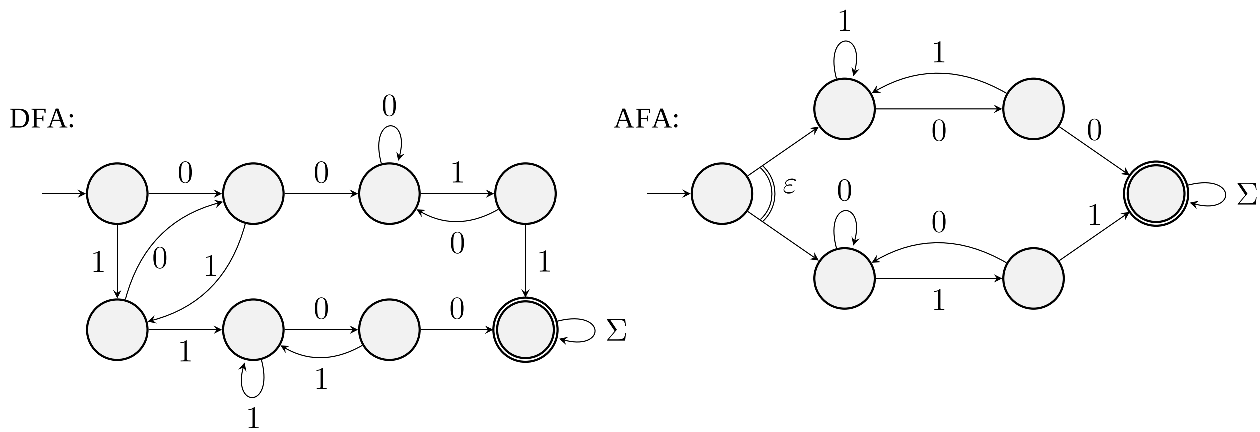
איור 28: קבלה ב-NFA.

- אוטומט מתחלף, AFA, כמתואר באיור 29, מקבל את המילה $a \circ x$ אם x מתקבל מ-q_1 וגם q_2, או מ-q_2 וגם q_3.



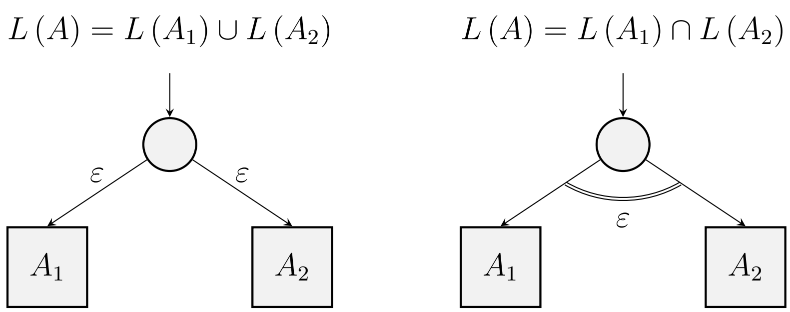
איור 29: קבלה ב-AFA.

דוגמה. נתבונן בשפה $L = \{w \in \{0,1\}^* \mid 11$ וגם רצף 00 יש רצף w-ב$\}$, וב-DFA ו-AFA שמזהים אותה, באיור 30.



איור 30: אוטומטי DFA ו-AFA עבור השפה L.

דוגמה. סגירות של AFA תחת איחוד וחיתוך.
באיור 31 מתוארות הבניות לאיחוד וחיתוך AFA-ים.



איור 31: איחוד וחיתוך AFA-ים.

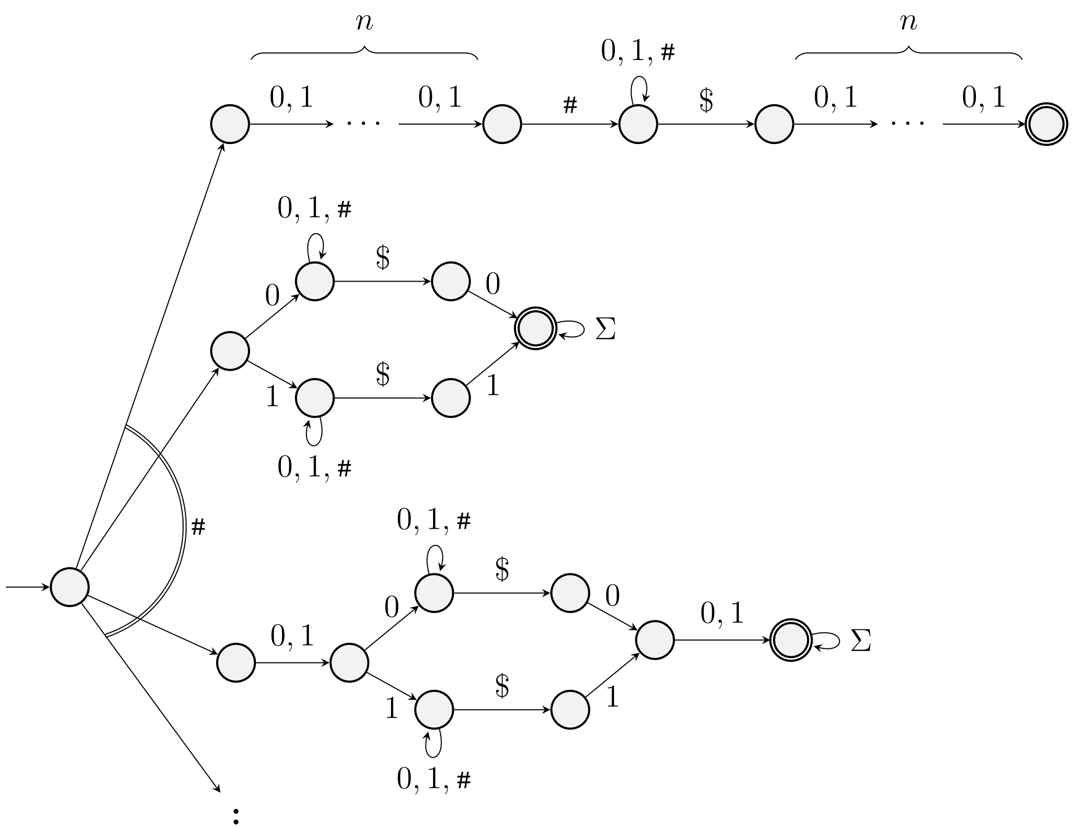
משפט. תרגום מ-AFA ל-NFA אפשרי ב-$2^{\Theta(n)}$.

הוכחה. נוכיח חסם תחתון.

1. משפחת שפות: נגדיר L_n תחת $\Sigma = \{0, 1, \#, \$\}$, להיות

$$L_n = \{x \circ \#v\#y \circ \$v \mid x, y \in \{0, 1, \#\}^*, v \in \{0, 1\}^n\}$$

2. נראה AFA A_n "קטן" שמזהה את L_n (בדומה להוכחת חסם תחתון של השלמה).



איור 32: אוטומט A_n שמזהה את L_n.

נשים לב ש-$|A_n| = \mathcal{O}\left(n^2\right)$.

3. כל NFA D שמזהה את L_n, יש לו $\leq 2^{2^n}$ מצבים (וכך נוכיח NFA $\overset{2^{\Omega(\sqrt{n})}}{\rightarrow}$ AFA).
נניח בשלילה שקיים NFA D שמזהה את L_n בגודל $< 2^{2^n}$.

- נשים לב שיש 2^n מילים באורך n מעל $\{0, 1\}$, וכך יש 2^{2^n} תתי-קבוצות של מילים באורך n מעל $\{0, 1\}$.
- תהי $S \subseteq \{0, 1\}^n = \{w_1, \ldots, w_k\}$. נבנה מילה w_S בתור

$$w_S = \#w_1\#w_2 \cdots \#w_k$$

נשים לב שלכל i מתקיים $w_S\$w_i \in L_n$, ולכל j כך ש-$w_j \notin S$ מתקיים $w_S\$w_j \notin L_n$.

- $|D| < 2^{2^n}$, ולכן קיימות $S \neq S'$ כך ש-$q = \delta_D^*\left(w_S\$\right) = \delta_D^*\left(w_{S'}\$\right)$.
  - מאחר ו-$S \neq S'$, בה"כ קיים $w_i \in S \setminus S'$.

$$\Rightarrow w_S\$w_i \in L_n \wedge w_{S'}\$w_i \notin L_n$$

מכאן, אם $\delta_D^*\left(q, w_i\right)$ מקבל, נקבל $w_{S'}\$w_i \in L_n$, ואחרת נקבל $w_S\$w_i \notin L_n$ - סתירה!

□

הגדרה. עבור קבוצה X, נסמן ב-$B^+\left(X\right)$ את הנוסחאות החיוביות מעל x.

$$\varphi \coloneqq x \in X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$

נגדיר AFA $A = \left(\Sigma, Q, \delta, q_0, F\right)$, כך ש-$\delta : Q \times \Sigma \to B^+\left(Q\right)$.

דוגמה. עבור הדוגמא מאיור 29, נקבל $\delta\left(q, \sigma\right) = \left(q_1 \wedge q_2\right) \vee \left(q_2 \wedge q_3\right)$.

הערה. אוטומט AFA הוא אי-דטרמיניסטי. בפרט, NFA הוא AFA עם δ עם כמתי $\vee$ בלבד.

<u>שלילה:</u> בהינתן $A = \left(\Sigma, Q, \delta, q_0, F\right)$, נשלים אותו ע"י $A' = \left(\Sigma, Q, \overline{\delta}, q_0, \overline{F}\right)$, כך ש-

$$\begin{cases} \overline{\delta}\left(q, \sigma\right) = \left(q_1 \vee q_2\right) \wedge \left(q_2 \vee q_3\right) \\ \overline{F} = Q \setminus F \end{cases}$$

<u>אינטואיציה:</u> כעת, הדרך לקרוא את פונקציית המעברים היא: מילה מתקבלת ע"י האוטומט המשלים אם היא **לא** מתקבלת מ-q_1 או q_2 , וגם **לא** מתקבלת מ-q_2 או q_3.

הגדרה. $S \subseteq Q$ מספקת את $\delta\left(q, \sigma\right)$ באופן מינימלי אם

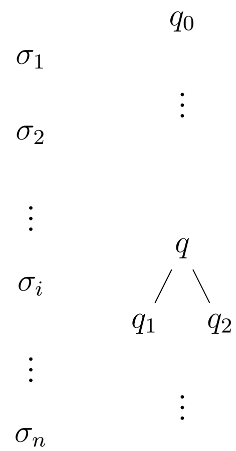
1. מספקת: נגדיר $f_S\left(q'\right) = \begin{cases} T & q' \in S \\ F & q' \notin S \end{cases}$ ומתקיים $f_S \models \delta\left(q, \sigma\right)$.
2. מינימלית: לכל $S' \subset S$ מתקיים $f_{S'} \not\models \delta\left(q, \sigma\right)$.

דוגמה. עבור $\delta\left(q, \sigma\right) = \left(q_1 \wedge q_2\right) \vee \left(q_3 \wedge q_4\right)$.

- $\{q_1, q_2\}$ ו-$\{q_2, q_3\}$ מספקות את $\delta\left(q, \sigma\right)$ באופן מינימלי.
- $\{q_1, q_3\}$ לא מספקת את $\delta\left(q, \sigma\right)$.
- $\{q_1, q_2, q_3\}$ מספקת את $\delta\left(q, \sigma\right)$ באופן לא מינימלי.

4.2 סמנטיקה

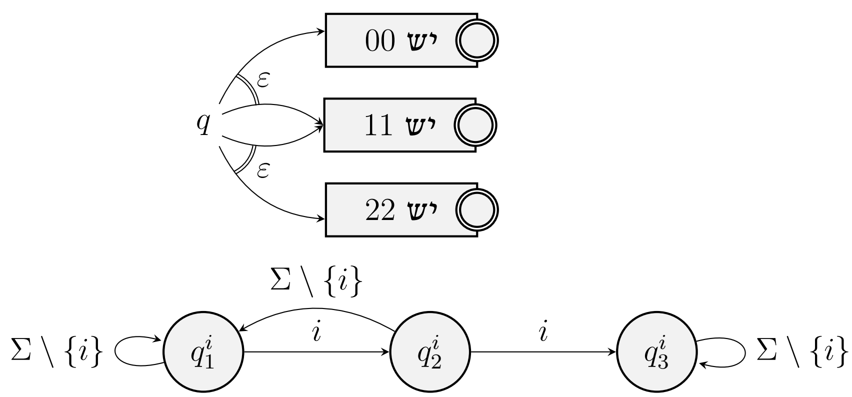
<u>סמנטיקה:</u> ריצה של AFA A על מילה $w = \sigma_1 \cdots \sigma_n$ היא עץ, כמתואר באיור 33.



איור 33: העץ המייצג של ריצת AFA A על מילה $w = \sigma_1 \cdots \sigma_n$.

בכל רמה i, קבוצת הילדים של q צריכה לספק את $\delta(q, \sigma_i)$ באופן מינימלי.

דוגמה. תהי $L = \{w \mid$ ב-w יש 00 וגם 11 או ב-w יש 11 וגם 22$\}$. נסתכל על AFA שמזהה אותה באיור 34.



איור 34: אוטומט AFA עבור השפה L.

נסתכל על הריצות האפשריות למילה 0011, באיור 35.

$$\begin{array}{ccccccc}
 & \multicolumn{2}{c}{q_0} & \quad & \multicolumn{2}{c}{q_0} \\
\varepsilon & / & \backslash & & / & \backslash \\
 & q_1^0 & q_1^1 & & q_1^1 & q_1^2 \\
0 & | & | & & | & | \\
 & q_2^0 & q_1^1 & & q_1^1 & q_1^2 \\
0 & | & | & & | & | \\
 & q_{acc} & q_1^1 & & q_1^1 & q_1^2 \\
1 & | & | & & | & | \\
 & q_{acc} & q_2^1 & & q_2^1 & q_1^2 \\
1 & | & | & & | & | \\
 & q_{acc} & q_{acc} & & q_{acc} & q_1^2
\end{array}$$

איור 35: הריצות האפשריות של האוטומט על המילה 0011.

$$\Rightarrow \delta\left(q_0, \varepsilon\right) = \left(q_1^1 \wedge q_1^0\right) \vee \left(q_1^1 \wedge q_1^2\right)$$

מסקנה. ריצה מקבלת $\Longleftrightarrow$ כל העלים מקבלים.

הערה. ייתכן שיהיו מספר מסלולים בעץ שמובילים לאותו המצב באותה הרמה - נאחד אותם. כעת, לא מדובר בעץ, אלא ב-runDAG. בכל רמה, כל מצב מופיע ≤ 1.

הוכחה. חסם עליון: בהינתן AFA $A = \left(\Sigma, Q, \delta, q_0, F\right)$, נבנה NFA A' כך ש-$L\left(A\right) = L\left(A'\right)$, ע"י ניחוש של הרמה ב-runDAG (מסובבים את ה-runDAG).

$$A' = \left(\Sigma, 2^Q, \delta', \left\{q_0\right\}, 2^F\right)$$

$$\delta'\left(S, \sigma\right) = \left\{S' \mid \text{מספקת באופן מינימלי את } S' \ \bigwedge_{q \in S} \delta\left(q, \sigma\right)\right\}$$

$\square$

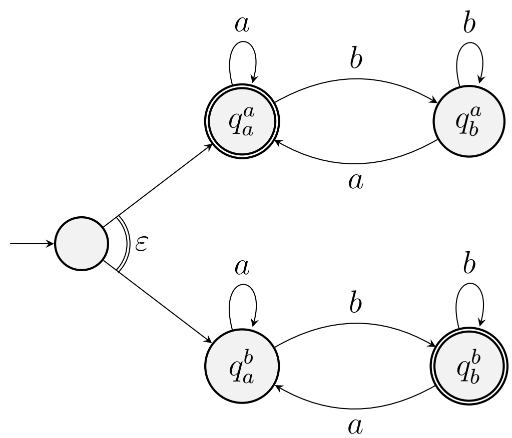
כעת, נכליל עבור מילים אינסופיות: Alternating Buchi Autmaton.

- סינטקס זהה ל-AFA: $\left(\Sigma, Q, \delta, q_0, \alpha\right)$.
- ריצה של A על $w \in \Sigma^\omega$ זו runDAG אינסופי, שמקבל $\Longleftrightarrow$ כל מסלול מבקר ב-α ∞ פעמים.

משפט. תרגום מ-ABW ל-NBW אפשרי ב-$3^{\Theta(n)}$.

הוכחה. (Miyano-Hayashi) חסם עליון.

- נסיון 1: כמו במילים סופיות - ננחש את הרמה ב-runDAG.
  - לא עובד - נסתכל על ABW עבור $\infty a \cap \infty b$: איור 36.



איור 36: אוטומט ABA עבור השפה $\infty a \cap \infty b$.

נסתכל על ה-runDAG באיור 37.

$$
\begin{array}{ccc}
 & q_a^a & q_a^b \\
a & | & | \\
 & q_a^a & q_a^b \\
b & | & | \\
 & q_b^a & q_b^b \\
a & | & | \\
 & q_a^a & q_a^b \\
b & | & | \\
 & q_b^a & q_b^b
\end{array}
$$

איור 37: הריצות האפשריות של האוטומט מאיור 36 על המילה $abab$.

האוטומט נכשל באופן דומה לכך שאוטומט מכפלה נכשל בחיתוך NBW.

- נסיון עובד: כל מצב הוא מהצורה $\left(\underbrace{O}_{\text{כל הענפים שעדיין "חייבים ביקור" ב-}\alpha}, \underbrace{S}_{\text{ניחוש רמה ב-runDAG}}\right)$, $O \subseteq S$, המצבים המקבלים הם $(S, \emptyset)$. מאתחלים את O להיות $S \setminus \alpha$.

$\square$

<u>השלמת NBW.</u>

- קופרמן-ורדי.

$$\underset{n}{\text{NBW}} \overset{\text{דואליזציה}}{\to} \underset{n}{\overline{\text{UCW}}} \overset{\text{KV '97}}{\to} \underset{n^2}{\overline{\text{ABW}}} \overset{\text{MH}}{\to} \underset{3^{n^2}}{\overline{\text{NBW}}}$$

- אוטומט N_W הוא אוטומט AFA כך ש-δ מכיל כמתי $\vee$ בלבד. לכן, כאשר נשלים את δ יהיו רק כמתי $\wedge$ - זה אוטומט אוניברסלי.
- ניתוח יותר זהיר של המעבר מניב $2^{n \log n}$.

הערה. $\text{NCW} = \text{DCW}$.
רמז: הראו כי $UFA \to DFA$.

חלק II
מידול מערכות

קונטקסט

Model Checking: בהינתן מודל (התנהגויות אפשריות) + מפרט (התנהגויות חוקיות) - האם יש באג? אופן הפעולה:

- תיאור מערכות ע"י מודל פורמלי
- תיאור המפרט ע"י לוגיקה.

1 מבנה קריפקה

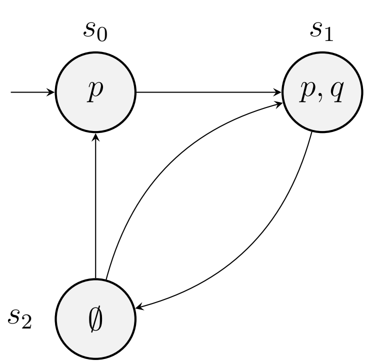
סמנטיקה: [Kripke '63] מבנה קריפקה הוא חמישייה

$$k = (AP, S, I, R, C),$$

כאשר

- AP: atomic propositions.
- S: מצבים.
- I: מצבים התחלתיים, $I \subseteq S$.
- R: יחס מעברים, $R \subseteq S \times S$.
- $L : S \to 2^{AP}$ labeling.

דוגמה. מבנה קריפקה, כמתואר באיור 38.



איור 38: דוגמא למבנה קריפקה.

$$\begin{gathered}AP=\{p,q\}\\ S=\{s_0,s_1,s_2\}\\ I=\{s_0\}\\ R=\{(s_2,s_0),(s_2,s_1),\ldots\}\\ L(s_0)=\{p\},L(s_2)=\emptyset,L(s_1)=\{p,q\}\end{gathered}$$

סמנטיקה: חישוב של k הוא מסלול ש-

1. מתחיל ב-I.
2. מכבד את R.

הערה. (overloading ל-L) עבור חישוב $\tau=\tau_1\tau_2\cdots$,

$$L(\tau)=L(\tau_1),L(\tau_2),\ldots$$

דוגמה. דוגמא לחישובים של k מאיור 38.

$$\tau_1=s_0(s_1s_2)^\omega$$

$$L(\tau_1)=\{p\},\{p,q\},\emptyset,\{p,q\},\emptyset,\ldots$$

$$\tau_2=(s_0s_1s_2)^\omega$$

$$L(\tau_2)=(\{p\},\{p,q\},\emptyset)^\omega$$

הערה. (overloading ל-L) עבור מבנה קריפקה k,

$$L(k)=\left\{\pi\in\left(2^{AD}\right)^\omega\mid\exists\tau:L(\tau)=\pi\right\}$$

1.1 תרגום $\text{kripke} \to \text{NBW}$

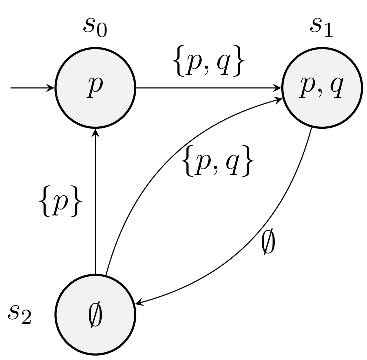
בהינתן $k = (AP, S, I, R, L)$, נרצה $A = (\Sigma, Q, \delta, Q_0, \alpha)$ כך ש-

$$L(A) = L(k)$$

<u>בנייה:</u> אינטואיציה - "מושכים את האותיות אחורה".

- $\Sigma = 2^{AP}$.
- $Q = S \cup \{q_0\}$.
- $\delta(q, \sigma) = \{q' \in Q \mid R(q, q') \wedge L(q') = \sigma\}$.
- $Q_0 = \{q_0\}$.
- $\alpha = Q$.

דוגמה. אוטומט NBW עבור k מאיור 38.



איור 39: ה-NBW המתאים לקריפקה מ-38.

1.2 Model Checking

איך ממדלים מערכת ע"י מבנה קריפקה? כללי אצבע:

1. מהו מצב המערכת?
2. מהם המעברים? (שאלה קשורה - מה הקלט למערכת? מה הפעולות האפשריות?)

דוגמה. (לא רצינית) משחק Pacman בלוח $n \times n$.

1. מצב המערכת הוא מיקום ה-Pacman:

$$S = [n] \times [n]$$

$$AP = [n] \times [n]$$

2. המעברים האפשריים:

$$\begin{array}{ccc} & (x, y+1) & \\ & U \uparrow & \\ (x-1, y) & \overset{L}{\leftarrow} (x, y) \overset{R}{\rightarrow} & (x+1, y) \\ & D \downarrow & \\ & (x, y-1) & \end{array}$$

הערה. מה אם במשחק היה בור או נקודת סיום?
זו האחריות של המפרט.

דוגמה. (רצינית) נתונה מערכת עם שני תהליכים, Proc. 1 ו-Proc. 2, שמריצים את הקיוד באיור 40.

Process 1	**Process** 2
0. while **True** :	0. while **True** :
1. wait $(turn = 1)$	1. wait $(turn = 2)$
2. C.S.	2. C.S.
3. $turn \leftarrow 2$	3. $turn \leftarrow 1$

איור 40: פסודו-קוד של שני התהליכים.

היינו רוצים: (אחריות המפרט)

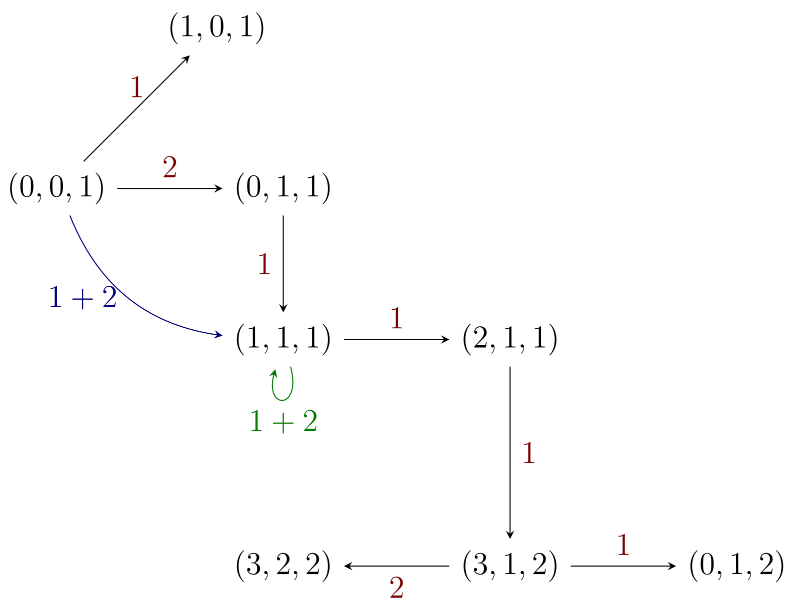
1. Safety: אסור ששני התהליכים יהיו ביחד בקטע הקריטי.
2. Liveness: כל אחד מהתהליכים נכנס ל-C.S. ∞ פעמים (אין starvation).

נמדל את המערכת:

1. מצב המערכת - הערך של $turn$, באיזו שורה כל אחד נמצא ($PC_1 \times PC_2 \times t$):

$$AP = \{0,\ldots,3\} \times \{0,\ldots,3\} \times \{1,2\}$$

2. המעברים - קריאת השורה הבאה, או context switch. כמודגם באיור 41.



איור 41: מעברים במודל התהליכים. התיוגים באדום לצורך המחשה בלבד - איזה תהליך מריץ את הפקודה הבאה. בכחול - תלוי ב-scheduler המערכת. בירוק - תלוי האם ה-wait הוא busy wait או blocking.

2 Linear Temporal Logic (LTL)

הלוגיקה שבאמצעותה נתאר מפרטים.

2.1 הגדרות

סינטקס: נוסחת LTL מוגדרת מעל AP.

- נוסחאות הבסיס הן $p \in AP, true, false$.

- עבור שתי נוסחאות LTL φ_1,φ_2, גם אלה נוסחאות LTL.

$$\begin{array}{c} \neg\varphi_1 \\ \varphi_1 \wedge \varphi_2 \\ X\varphi_1 \\ \varphi_1 U \varphi_2 \end{array}$$

באופן שקול:

$$\varphi \to \begin{array}{c} p \in AP \\ true \\ false \end{array} \mid \neg\varphi \mid \varphi \wedge \varphi \mid X\varphi \mid \varphi U \varphi$$

- קיצורים:

$$\varphi_1 \vee \varphi_2 \equiv \neg\left(\neg\varphi_1 \wedge \neg\varphi_2\right)$$

$$\varphi_1 \to \varphi_2 \equiv \left(\neg\varphi_1 \vee \varphi_2\right)$$

- טמפורליים: F - eventually, G - always.

$$F\varphi, G\varphi$$

<u>סמנטיקה:</u> עבור חישוב $\pi = \pi_1, \pi_2, \dots \in \left(2^{AP}\right)^\omega$, מתי $\pi \vDash \varphi$?

<u>סימון:</u> לכל i, $\pi^i = \pi_i \pi_{i+1} \dots$.

הגדרה. סיפוק של נוסחא φ ע"י חישוב π.

- בסיס:
  - $\pi \vDash p \iff p \in \pi_1$.
  - $\pi \vDash true \iff T$.
  - $\pi \vDash false \iff F$.
- צעד:

- $\pi \models \neg\varphi_1 \iff \pi \not\models \varphi_1$.
- $\pi \models \varphi_1 \wedge \varphi_2 \iff \pi \models \varphi_1 \wedge \pi \models \varphi_2$.
- $\pi \models X\varphi_1 \iff \pi^1 \models \varphi_1$.
- $\pi \models \varphi_1 U \varphi_2 \iff \exists k \forall 1 \leq i < k : \pi^i \models \varphi_1 \wedge \pi^k \models \varphi_2$.
- $\pi \models F\varphi_1 \iff \exists k : \pi^k \models \varphi_1$.
- $\pi \models G\varphi_1 \iff \forall k : \pi^k \models \varphi_1$.

דוגמה. פרמול של התנהגות חוקית.

1. Safety:

$$G\left(\neg\left(PC1 = C.S. \wedge PC2 = C.S.\right)\right)$$

2. Liveness:

$$G\left(F\left(PC1 = C.S.\right) \wedge F\left(PC2 = C.S.\right)\right)$$

3. Server:

$$G\left(\text{reg} \rightarrow F\text{grant}\right)$$

4. Reach $(10, 10)$ (Pacman):

$$F\left(x = 10 \wedge y = 10\right)$$

5. Avoid $(5, 5)$ (Pacman):

$$G\left(\neg\left(x = 5 \wedge y = 5\right)\right)$$

הערה. הבעה של $F\varphi_1, G\varphi_2$ באמצעות X, U:

$$F\varphi_1 = true\, U\varphi_1$$

$$G\varphi_2 = \neg F \neg \varphi_2$$

תרגיל. הבע את "p מתקיים באינדקסים האי-זוגיים ורק בהם" באמצעות נוסחת LTL, כאשר $AP = \{p, q\}$.

פתרון. נתאר באמצעות האינווריאנטה:

$$\underbrace{p}_{\text{אתחול}} \wedge \underbrace{G\left(\underbrace{p \to \neg Xp}_{\psi_1}, \underbrace{\neg p, Xp}_{\psi_2}\right)}_{\text{תחזוקה}}$$

2.2 סיפוק נוסחת LTL

דוגמה. איך בודקים אם מילה π מספקת נוסחא φ? נסתכל על מסלול ספציפי.

- לכל אינדקס i, נרשום אילו נוסחאות מסתפקות ע"י π^i.
  - נעבוד מנוסחאות קטנות לגדולות.

$$\begin{array}{cccccccccc}
\{p\} & \emptyset & \{p,q\} & \{q\} & \{p\} & \emptyset & \{p,q\} & \emptyset & \{p,q\} & \ldots \\
p, \neg q & \neg p, \neg q & p, q & \neg p, q & p, \neg q & \neg p, \neg q & p, q & \neg p, \neg q & p, q & \\
\neg Xp & Xp & \neg Xp & Xp & \neg Xp & \ldots & & & & \\
\psi_1 & \psi_1 & \psi_1 & \psi_1 & & & & & & \\
\psi_2 & \psi_2 & \psi_2 & \psi_2 & \ldots & & & & & \\
\psi_1 \wedge \psi_2 & \psi_1 \wedge \psi_2 & \ldots & & & & & & & \\
G\left(\psi_1 \wedge \psi_2\right) & G\left(\psi_1 \wedge \psi_2\right) & \ldots & & & & & & &
\end{array}$$

דוגמה. (אנטי-דוגמא)

$$\begin{array}{cccccccc}
\{p\} & \emptyset & \{p\} & \{p\} & \emptyset & \{p\} & \emptyset & \ldots \\
p & \neg p & p & p & \neg p & p & \neg p & \\
\neg Xp & Xp & Xp & \neg Xp & Xp & \neg Xp & Xp & \\
\psi_1 & \psi_1 & \neg\psi_1 & \psi_1 & \psi_1 & \psi_1 & & \\
\psi_2 & \psi_2 & \psi_2 & \psi_2 & \psi_2 & \psi_2 & \ldots & \\
\psi_1\wedge\psi_2 & \psi_1\wedge\psi_2 & \neg\left(\psi_1\wedge\psi_2\right) & \psi_1\wedge\psi_2 & \psi_1\wedge\psi_2 & \psi_1\wedge\psi_2 & \ldots & \\
\neg G\left(\psi_1\wedge\psi_2\right) & \neg G\left(\psi_1\wedge\psi_2\right) & \neg G\left(\psi_1\wedge\psi_2\right) & G\left(\psi_1\wedge\psi_2\right) & G\left(\psi_1\wedge\psi_2\right) & \ldots & &
\end{array}$$

תרגיל. הבע את $\infty p, \neg\infty p$ באמצעות נוסחת LTL.

$$\infty p \iff GFp$$

אחרת, החל ממקום מסוים אין יותר p-ים, ויהיה מקום בו יסופק $\neg Fp$.

$$\neg\infty p \iff FG\neg p \equiv \neg GFp$$

בהמשך: המרה מ-LTL ל-NBW.

$$\begin{array}{ccc}
\text{LTL} & \overset{\text{v.w.}}{\rightarrow} & \text{NBW} \\
\varphi & \rightarrow & A_\varphi
\end{array}$$

$$L\left(A_\varphi\right) = \{\pi \mid \pi \vDash \varphi\}$$

3. Model Checking

<u>קלט:</u> מערכת kripke - k (כל החישובים האפשריים) ו-LTL φ (כל החישובים הנכונים).

<u>מטרה:</u> להכריע האם כל חישוב $\pi \in L(k)$ מקיים $\varphi \vDash \pi$ (כלומר, כל חישוב אפשרי הוא נכון).

3.1 בניית Vardi-Wolper

<u>מטרה:</u> בהינתן LTL φ, נבנה NBW A_{φ} כך ש-$L(A_{\varphi}) = \{\pi \mid \pi \vDash \varphi\}$.

כך, בהינתן kripke k נבנה NBW A_k, ואז נותר לבדוק האם $L(A_k) \overset{?}{\subseteq} L(A_{\varphi})$ (לא בהכרח שוויון - ייתכן ש-k מגביל התנהגויות חוקיות, אך זה לא משנה).

3.1.1 סימונים

- עבור נוסחת LTL φ, $\mathrm{Cl}(\varphi)$ היא קבוצת כל תת-הנוסחאות של φ ושלילותיהן (הגדרה פורמלית בהמשך).
- הסימון S בזמן i הוא תת-קבוצה של $\mathrm{Cl}(\varphi)$ כך שהסיפא π^i

1. מספקת את כל הנוסחאות ב-S.
2. לכל $\psi \in \mathrm{Cl}(\varphi)$, או ש-$\psi \in S$ או ש-$\neg\psi \in S$ (XOR, כלומר $\psi \in S \oplus \neg\psi \in S \equiv T$).

דוגמה. מספר נוסחאות LTL וה-Cl שלהן.

1. $AP = \{p\}, \varphi_1 = Xp$.

$$\mathrm{Cl}(Xp) = \{p, \neg p, Xp, \neg Xp\}$$

	$\{p\}$	$\{p\}$	$\emptyset$	$\{p\}$	$\emptyset$	$\{p\}$
$\rightarrow$	p	p	$\neg p$	p	$\neg p$	p
	Xp	$\neg Xp$	Xp	$\neg Xp$	Xp	$\neg Xp$

2. $AP = \{p, q\}, \varphi_2 = pUq$.

$$\mathrm{Cl}\,(pUq) = \{p, \neg p, q, \neg q, pUq, \neg pUq\}$$

	$\{p\}$	$\{p\}$	$\{p\}$	$\{q\}$	$\emptyset$	$\{p\}$	$\{p\}$	$\emptyset$	$\{p\}$	$\{p\}$	$\{q\}$	
$\rightarrow$	$p, \neg q$	$p, \neg q$	$p, \neg q$	$\neg p, q$	$\neg p, \neg q$	$\cdots$						
	pUq	pUq	pUq	pUq	$\neg(pUq)$	$\neg(pUq)$	$\neg(pUq)$	$\neg(pUq)$	pUq	pUq	pUq	$\cdots$

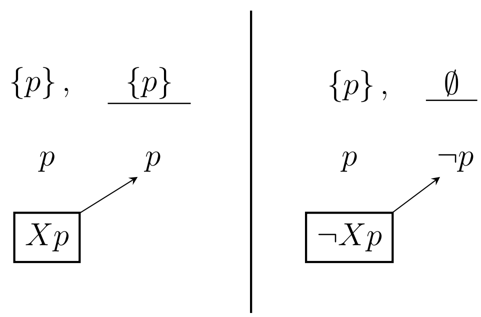
בכל צעד, הסימון "מביע דעה" על כל תת-נוסחא של φ.

סימון on the fly

- מקבלים את החישוב אות-אות. ננחש!

דוגמה. סימון on the fly.

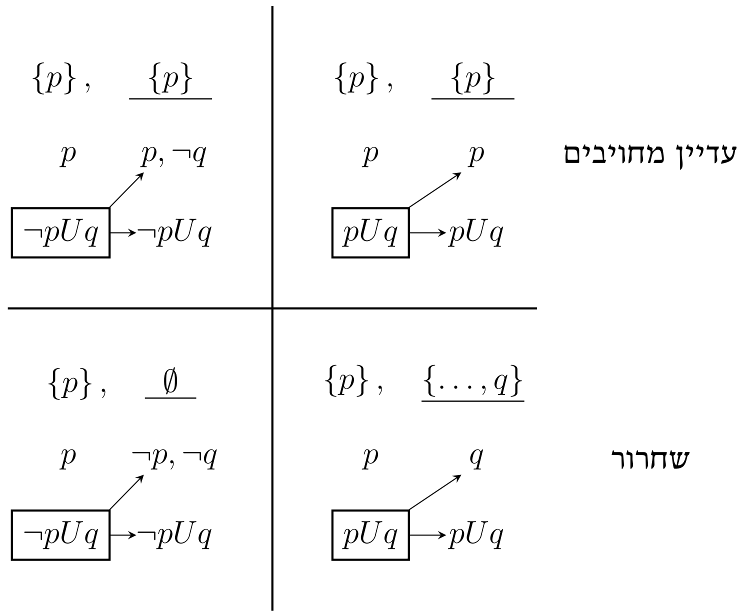
1. $AP = \{p\}, \varphi_1 = Xp$, המצבים השונים כמתואר באיור 42.



איור 42: סימון on-the-fly עבור $AP = \{p\}, \varphi_1 = Xp$.

- ניחוש Xp בזמן i מחייב ש-p יהי בסימון של זמן $i+1$, וכנ"ל עבור $\neg Xp$.
- נשים לב: אפשר לוודא את נכונות הניחוש בצעד הבא - אם ניחשנו Xp וקראנו $\emptyset$ בצעד הבא, הניחוש אינו נכון.

2. $AP = \{p, q\}, \varphi_2 = pUq$, המצבים השונים כמתואר באיור 43.



איור 43: סימון on-the-fly עבור $AP = \{p,q\}\,, \varphi_1 = pUq$.

- ניחוש pUq בזמן i מחייב אחד משניים מצבים:
  (א) q נכון בצעד הבא, מה שמשחרר את ההתחייבות של pUq.
  (ב) אם q אינו נכון, ההתחייבות עוד לא השתחררה. לכן, אנחנו עדיין מחויבים ל-pUq ונדרשים ש-p יהיה דלוק בצעד הבא.
- נשים לב: כאן לא תמיד אפשר לוודא את הניחוש באופן לוקאלי. יש צורך לוודא שלא נשאר מחוייבים לעד, מאחר ו-$\{p\}^{\omega} \nvDash pUq$ (נעשה בהמשך).
- המקרה של $\neg(pUq)$ דואלי.

הערה. בפועל, ננחש את כל הסימון לפי שנראה את האות הבאה.

3.1.2 בנייה

- רפרנס: Principles of Model Checking; Baier, Katoen, פרק 5.2.

הגדרה. עבור נוסחת LTL φ, $\mathrm{Cl}(\varphi)$ מקיים

1. $\varphi \in \mathrm{Cl}(\varphi)$.
2. $\psi \in \mathrm{Cl}(\varphi) \iff \neg\psi \in \mathrm{Cl}(\varphi)$.

3. $\psi_1 \wedge \psi_2 \in \mathrm{Cl}(\varphi) \Rightarrow \psi_1 \in \mathrm{Cl}(\varphi) \wedge \psi_2 \in \mathrm{Cl}(\varphi)$.
4. $X\psi \in \mathrm{Cl}(\varphi) \Rightarrow \psi \in \mathrm{Cl}(\varphi)$.
5. $\psi_1 U \psi_2 \in \mathrm{Cl}(\varphi) \Rightarrow \psi_1 \in \mathrm{Cl}(\varphi) \wedge \psi_2 \in \mathrm{Cl}(\varphi)$.

תרגיל. נסתכל על $\varphi = p \wedge (XpUq)$.

$$\mathrm{Cl}(\varphi) = \{p, \neg p, q, \neg q, Xp, \neg Xp, XpUq, \neg(XpUq), \varphi, \neg\varphi\}$$

הגדרה. תת-קבוצה $S \subseteq \mathrm{Cl}(\varphi)$ היא "טובה" (מתאימה לסימון) $\iff$

1. S "מביעה דעה" על כל תת-נוסחא של φ:

$$\forall \psi \in \mathrm{Cl}(\varphi) : \psi \in S \iff \neg\psi \notin S$$

2. S קונסיסטנטית:

$$\psi_1 \wedge \psi_2 \in S \iff \psi_1 \in S \wedge \psi_2 \in S$$

בהינתן נוסחא φ, נגדיר אוטומט A_φ ע"י:

$$A_\varphi = \left(\underbrace{\Sigma}_{2^{AP}}, \underbrace{Q}_{\{S \subseteq \mathrm{Cl}(\varphi) \mid S \text{ טובה}\}}, \delta, \underbrace{Q_0,}_{\{S \mid \varphi \in S\}} \alpha \right)$$

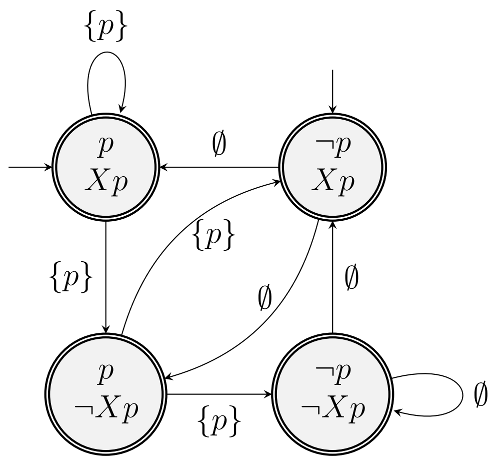
$S' \in \delta\left(S, \underbrace{\sigma}_{\subseteq AP}\right)$ אם:

1. $\sigma = AP \cap S$ (למשל $p, \neg q \Rightarrow \{p\}, \neg p, \neg q \Rightarrow \emptyset$).
2. לכל $X\psi \in \mathrm{Cl}(\varphi)$, $\psi \in S' \iff X\psi \in S$.
3. לכל $\psi_1 U \psi_2 \in \mathrm{Cl}(\varphi)$, $\psi_1 U \psi_2 \in S \iff$
   (א) $\psi_2 \in S$.

(ב) $\psi_1 \in S \land \psi_1 U \psi_2 \in S'$.

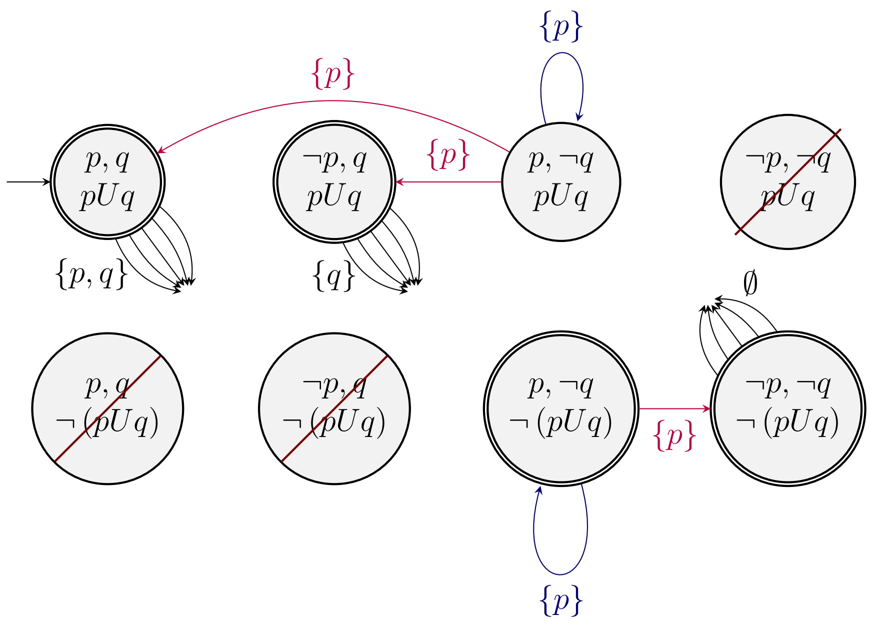
דוגמה. מספר אוטומטי A_φ עבור LTL φ.

1. $AP = \{p\}$, $\varphi_1 = Xp$, באיור 44.



איור 44: אוטומט NBW עבור $\varphi_1 = Xp$, $AP = \{p\}$.

2. $AP = \{p, q\}$, $\varphi_2 = pUq$, באיור 45.



איור 45: אוטומט NBW עבור $AP = \{p, q\}, \varphi_2 = pUq$. הקשתות האדומות מסמלות "שחרור", והכחולות מסמלות "עדיין מחויבים".

נשים לב שלא כל המצבים מקבלים - אחרת קיימת ריצה מקבלת על $\{p\}^\omega$.

הערה. באופן כללי, נגדיר לכל $\psi_1 U \psi_2 \in \mathrm{Cl}(\varphi)$

$$\alpha_{\psi_1 U \psi_2} = \{S \mid \psi_2 \in S \vee \neg(\psi_1 U \psi_2) \in S\}$$

וכך קיבלנו NGBW, ניתן לתרגם ל-NBW.

<u>אבחנה:</u> $|A_\varphi| = 2^{|\varphi|}$.

3.2 M.C. באמצעות Vardi-Wolper

בהינתן kripke k ו-LTL φ בגודל n, נרצה $L(k) \subseteq \{\pi \mid \pi \vDash \varphi\}$.

1. אלגוריתם

$$k \overset{\text{מושכים אותיות אחורה}}{\rightarrow} A_k$$

$$\varphi \overset{\text{VW}}{\rightarrow} A_\varphi$$

- כעת נבדוק האם $L(A_k) \overset{?}{\subseteq} L(A_\varphi) \iff L(A_k) \cap \overline{L(A_\varphi)} = \emptyset$.

$$A_\varphi \overset{\text{safra/KV}}{\rightarrow} \overline{A_\varphi}$$

כעת נבדוק האם $L\left(\overline{A_\varphi} \times A_k\right) \overset{?}{=} \emptyset$.

<u>קלט:</u> NBW A.

<u>פלט:</u> $L(A) \overset{?}{=} \emptyset$.

- נבנה גרף ע"י מחיקת האותיות - G_A.
- נחפש לאסו מקבל - מסלול ממצב התחלתי, שמגיע לקודקוד שנמצא במעגל שעובר במצב מקבל. פתרונות:

(א) נריץ BFS מכל קודקוד, ונבדוק האם קיים $q \in R(q_0)$ כך שקיים $q' \in \alpha$ כך ש-$q' \in R(q) \wedge q \in R(q')$ (זמן ריצה $\mathcal{O}(n^2)$).

(ב) נמצא SCC של G_A. נבדוק האם קיים רכיב קשירות חזקה לא טריוויאלי שמכיל מצב מקבל וישיג ממצב התחלתי ($\mathcal{O}(n)$).

- בעיה: $|A_\varphi| = 2^n$, ולכן $\left|\overline{A_\varphi}\right| \sim 2^{2^n}$! לא טוב.

2. אלגוריתם 2.

$$\underbrace{\varphi}_{n} \rightarrow \neg\varphi \overset{\text{VW}}{\rightarrow} \underbrace{A_{\neg\varphi}}_{2^n}$$

כך, זמן הריצה הוא $\mathcal{O}(n \cdot 2^n)$.

<u>חסם תחתון:</u> הבנייה של VW הדוקה.

- ניתן להראות שהבעיה היא NP-קשה (ואז כל עוד $P \neq NP$, החסם הדוק במידה מסוימת).
- נוכיח בשיטה הסטנדרטית.

1. משפחת שפות L_n, נבחר את המשפחה

$$L_n = \left\{x\#v\#y\$v\#^\omega \mid v \in \{0,1\}^n, x, y \in \{0,1,\#\}^*\right\}$$

2. יש LTL קטנה שמזהה את L_n - φ בגודל $\mathcal{O}(n^2)$.
3. כל NBW שמזהה את L_n הוא גדול - הראנו זאת כבר.

3.3 הדיקות בניית VW

טענה. בניית VW הדוקה.

הוכחה. נוכיח בשיטה הסטנדרטית.

1. נבחר את משפחת השפות

$$L_n = \left\{ x\#v\#y\$v\#^{\omega} \mid v \in \{0,1\}^n, x,y \in \{0,1,\#\}^* \right\}$$

2. נראה LTL בגודל $\mathcal{O}\left(n^2\right)$:

(א) יש במילה $ אחד - באמצעות $\neg\$U\left(\$ \wedge XG\neg\$\right)$.

(ב) שוויון בין n התווים אחרי ה-# הראשון ובין n התווים אחרי ה-$.

- הביטוי $X^i0 \wedge G\left(\$ \rightarrow X^i0\right)$ בודק האם האות ה-i שווה ל-0 וגם האות ה-i אחרי ה-$ שווה ל-0 ($\$ \rightarrow X^i0$ מתקיים באופן ריק בכל מקום פרט למקומות הדרושים לנו).
- באופן דומה עבור 1. בסך הכל:

$$F\left(\# \wedge \left(\bigwedge_{i=1}^{n} X^i0 \wedge G\left(\$ \rightarrow X^i0\right)\right) \vee \left(\bigwedge_{i=1}^{n} X^i1 \wedge G\left(\$ \rightarrow X^i1\right)\right)\right)$$

(ג) מיקומי ה-#-ים, ובפרט שיש $\#^{\omega}$ בסוף המילה - $FG\#$.
בסך הכל, גודל φ היא בערך n+$\sum X^i$, כלומר

$$n + 2\sum_{i=1}^{n} i = \mathcal{O}\left(n^2\right)$$

3. הראנו בעבר שכל DFA צריך פיצוץ של 2^{2^n}, עם אותה משפחת שפות ללא ה-$\#^{\omega}$.
לכן, שלב זה נשאר כתרגיל.

□

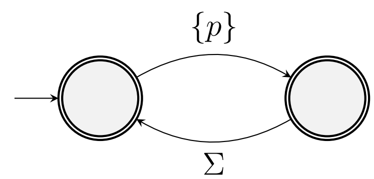
3.4 LTL < NBW

משפט. LTL < NBW.

הוכחה. עבור $AP = \{p\}$, נגדיר את השפה

$$L = \left\{\pi \in \left(2^{AP}\right)^{\omega} \mid p \text{ במקומות הזוגיים} \right\}$$

- אוטומט NBW ל-L באיור 46.



איור 46: אוטומט NBW עבור L.

- נוכיח שלא קיימת נוסחת LTL שמזהה את L.
- לכל $n \in \mathbb{N}$ נגדיר מסלול

$$\pi^n := (\{p\})^n \, \emptyset \, \{p\}^{\omega}$$

- עבור מסלול π ונוסחא φ, נסמן

$$[\![\pi, \varphi]\!] := \begin{cases} 1 & \pi \models \varphi \\ 0 & \pi \not\models \varphi \end{cases}$$

למה. עבור נוסחת LTL φ עם n X-ים,

$$\forall k \geq 1 : [\![\pi^{n+1}, \varphi]\!] = [\![\pi^{n+k}, \varphi]\!]$$

נשתכנע באינודקציה על $|\varphi|$:

- בסיס: $\varphi = p$ ואז ברור ש-$\cdots = [\![\pi^2, \varphi]\!] = [\![\pi^1, \varphi]\!]$.

- צעד: נפריד למקרים.

1. $\varphi = \neg\psi$, ואז

$$\llbracket \pi^n, \neg\psi \rrbracket = 1 - \llbracket \pi^n, \psi \rrbracket \underbrace{=}_{\text{צעד}} 1 - \llbracket \pi^{n+1}, \psi \rrbracket = 1 - \llbracket \pi^{n+1}, \varphi \rrbracket$$

2. $\varphi = X\psi$, ואז

$$\llbracket \pi^{n+1}, X\psi \rrbracket = \llbracket \pi^n, \psi \rrbracket \underbrace{=}_{\text{צעד}} \llbracket \pi^{n+1}, \psi \rrbracket = \llbracket \pi^{n+2}, \varphi \rrbracket$$

3. נותרו $\wedge, U$, שלא נעשה פה.

- הלמה גוררת את נכונות המשפט: נניח בשלילה שקיימת φ כך ש-$n = |\varphi|$ סופי, ואז

$$\llbracket \pi^{n+1}, \varphi \rrbracket = \llbracket \pi^{n+2}, \varphi \rrbracket$$

סתירה!

$\square$

3.5 סיבוכיות של Model Checking

תיאור הבעיה:

- קלט: (k, φ).
- פלט: האם $L(k) \subseteq L(\varphi)$?

משפט. MC היא co-NP-hard.

הוכחה. $G \in \text{HAMPATH} \iff$ ב-G יש מסלול המילטוני (כלומר, עובר בכל צומת בדיוק פעם אחת). נעשה רדוקציה - בהינתן גרף $G = (V, E)$, נגדיר

$$k = (AP, S, R, I, L)$$

כך ש-

$$I = V \qquad S = V \cup \{s\} \qquad AP = V$$
$$L(v) = v, L(s) = \emptyset \quad (u,v) \in E, uRs, sRs \iff (u,v) \in E : uRv$$

נשים לב ש-

$$L(k) = \{\text{כל המסלולים ב-}G\text{ + מותר ליפול ל-}s\text{ ולהישאר שם}\}$$

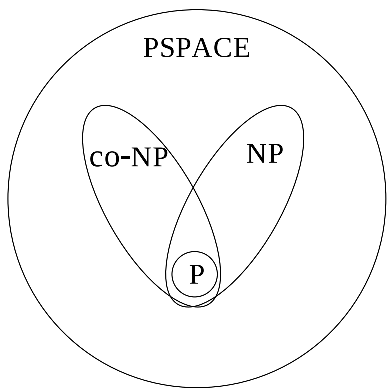
$$\varphi = \bigwedge_{v \in V} \underbrace{Fv}_{v\text{ מופיע}} \wedge \underbrace{G(v \to XG\neg v)}_{\leq 1\text{ מופעים}}$$

מכאן, ניתן לראות שכל המסלולים המילטוניים $\iff k \vDash \varphi$. מכאן,

$$k \vDash \neg\varphi \iff G \notin \text{HAMPATH}$$

וסיימנו. □

הערה. $\text{co-NP} = \left\{\overline{L} \mid L \in NP\right\}$, המחשה של היחסים השונים באיור 47.



איור 47: מחלקות סיבוכיות.

.

משפט. (העשרה) MC היא PSPACE-complete.

<u>שאלה:</u> מה הסיבוכיות כאשר φ "קטנה"?
נתמקד בנוסחאות מהצורה $\varphi = G\neg\underbrace{T}_{\text{המצבים הרעים}}$. כלומר, safety language.

- $A_{\neg\varphi} \times A_k$ לינארי בגודל k, מאחר ו-$A_{\neg\varphi}$ קבוע.
- בדיקת ריקנות בזמן לינארי - SCC.
- בסך הכל, לינארי בגודל של k.
- מחלקת סיבוכיות נוספת, NL - זיכרון לוגריתמי בגודל הקלט.
- M.C כאשר φ מהצורה הזאת הוא ב-NL.

3.6 בדיקת מודל סימבולית

<u>תזכורת:</u> $k = (AP, S, R, I, L)$.

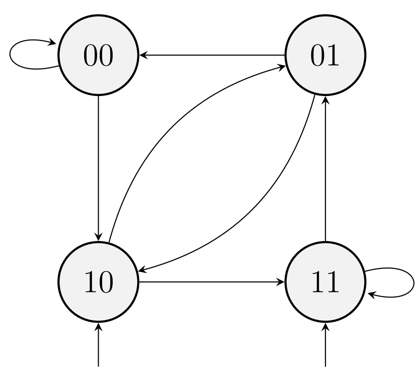
היום נסתכל על kripke סימבולי:

$$k = \left(AP = X = \{x_1, \ldots, x_n\}, S = 2^n, \varphi_R, \varphi_I, L : L(s) = s,\right)$$

כאשר φ_I היא נוסחא מעל X, ו-φ_R נוסחא מעל $X \cup X'$ ($X' = \{x' \mid x \in X\}$).

- נייצג קבוצה S באמצעות נוסחא φ_S בתור $S = \{s : \varphi_S(s) = \text{true}\}$.
- נייצג מעברים R באמצעות נוסחא φ_R בתור $R = \{(s, s') : \varphi_R(s, s') = \text{true}\}$.

דוגמה. נסתכל על $\varphi_I = x, \varphi_R = (x \leftrightarrow y'), X = \{x, y\}$. בנייה מפורשת באיור 48.



איור 48: אוטומט kripke בייצוג סימבולי.

המצבים ההתחלתיים הם אלו שמספקים את φ_I:

$$\begin{aligned}\varphi_I(00) &= \text{false}\\ \varphi_I(01) &= \text{false}\\ \varphi_I(10) &= \text{true}\\ \varphi_I(11) &= \text{true}\end{aligned}$$

כלומר, 10 ו-11. קשת $x \to y$ תופיע $\iff \varphi_R(x,y) = \text{true}$. למשל:

$$\begin{aligned}\varphi_R(00,01) &= \text{false}\\ \varphi_R(11,01) &= 1 \leftrightarrow 1 = \text{true}\end{aligned}$$

במקרה זה, משמעות φ_R: יש קשת xy ל-$x'y'$ $\iff x = y'$.

דוגמה. (Shift Register) לפי הקוד באיור 49.

Shift-Register (x,y,z) :
assert $(x=0 \vee y=0 \vee z=0)$
while **True** :
$x \leftarrow y$
$y \leftarrow z$
$z \leftarrow 1$

איור 49: פסודו-קוד של Shift-Register.

נסתכל על מבנה הקריפקה שמתאר את המודל:

$$X = \{x,y,z\}, \varphi_I = \neg x \vee \neg y \vee \neg z$$

$$\varphi_R = x' \leftrightarrow y \wedge y' \leftrightarrow z \wedge z'$$

דוגמה. (Critical Section) לפי הקוד באיור 50.

Process 1	**Process** 2
while **True** :	while **True** :
1. wait $(turn = 1)$	1. wait $(turn = 2)$
2. C.S.	2. C.S.
3. $turn \leftarrow 2$	3. $turn \leftarrow 1$

איור 50: פסודו-קוד של שני התהליכים.

$$X = (\text{PC1}, \text{PC2}, \text{turn})$$

$$\varphi_I = (\text{PC1} = 1) \wedge (\text{PC2} = 1) \wedge \underbrace{(\text{turn} = 1)}_{\text{שאלת מידול}}$$

$$\begin{aligned}
\varphi_R = & \neg \overbrace{(\text{PC1} \leftrightarrow \text{PC1}' \wedge \text{PC2} \leftrightarrow \text{PC2}')}^{\text{לפחות תהליך אחד זז}} \wedge \overbrace{(\ldots)}^{\text{רק תהליך אחד זז}} \wedge \\
& \wedge (\text{PC1} = 1 \wedge \text{turn} = 1) \leftrightarrow (\text{PC1}' = 2 \wedge \text{turn}' = 1) \\
& \wedge (\text{PC1} = 1 \wedge \text{turn} = 1) \leftrightarrow (\text{PC1}' = 1 \wedge \text{turn}' = 2) \\
& \wedge (\ldots) \wedge \\
& \wedge (\text{PC1} = 1 \wedge \text{PC1} \neq \text{PC1}') \leftrightarrow (\text{PC1}' = 1 \wedge \text{turn}' = 2) \\
& \vdots
\end{aligned}$$

הערה. מעתה נייצג kripke באמצעות $k = (X, \varphi_R, \varphi_I)$.

כעת, הקלט הוא מהצורה $k = (X, \varphi_R, \varphi_I)$ ו-$\varphi_T = G\neg T$ (T מייצג את קבוצת המצבים הרעים).

3.6.1 BDD-based M.C

[Clarke, McMillan, ...; 92′]

- בהמשך ההיסטוריה: [Biere, ..., Clarke, ...; '99] BMC.
- BDD הוא מבנה נתונים שמתחזק פונקציה - קבוצה $S = \{s \mid f_S(s) = \text{true}\}$.
- אפשר לעשות פעולות על $1+$ BDDs: למשל, $f_{S_1} \wedge f_{S_2} \longleftarrow f_{S_1 \wedge S_2}$:

$$f_{S_1 \wedge S_2}(s) = \text{true} \iff f_{S_1}(s) = \text{true} \wedge f_{S_2}(s) = \text{true}$$

כעת, הקלט הוא f_T, $k = \left(X, \underbrace{f_R}_{X \cup X' \text{ מעל BDD}}, \underbrace{f_I}_{X \text{ מעל BDD}}\right)$.

<u>מטרה:</u> לייצר BDD שמייצג את כל המצבים שיציגים מ-I. כך, נוכל לבדוק את ריקנות BDD החיתוך. באינדוקציה:

$$\begin{array}{l} f_0 = f_I \\ while \qquad\qquad f_{i+1} \neq f_i \\ \qquad\qquad f_{i+1} = f_i \vee \exists x \in X.\ f_i \wedge f_R \end{array}$$

$\exists x f_i \wedge f_R$ זו פונקציה:

$$\left(\exists x f_i \wedge f_R\right)\left(\underbrace{y}_{X'\text{-השמה ל}}\right) = \text{true} \iff \exists x : f_i\left(x\right) = \text{true} \wedge f_R\left(x, y\right) = \text{true}$$

פלט ה-M.C:

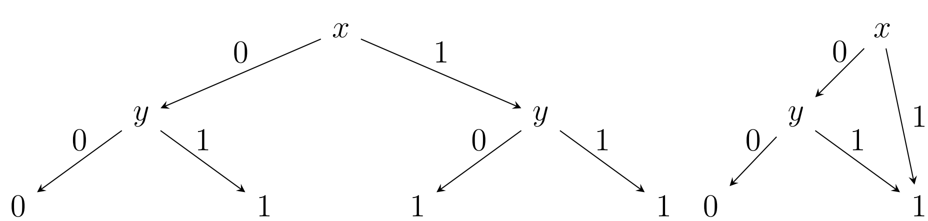
$$\boxed{\text{ret}\ \left(f_n \wedge f_T \equiv 0\right)}$$

דוגמה. איך נראה BDD? למשל, עבור

$$X = \{x, y\}$$

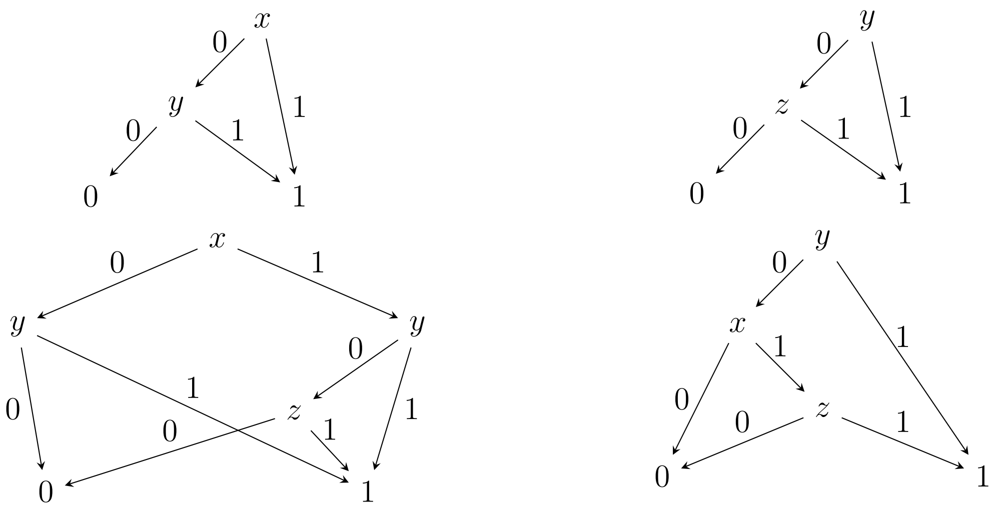
$$\varphi_S = x \vee y = \{10, 01, 11\}$$

מייצגים את הפונקציה הבוליאניות באמצעות עץ, ואז מצמצמים, למשל באיור 51.



איור 51: משמאל לימין: עץ החלטה עבור φ_S, BDD עבור φ_S.

דוגמאות נוספות ל-BDDs, עבור $f_{S_1} = x \vee y$, $f_{S_2} = y \vee z$ ו-$f_{S_1 \wedge S_2}$, באיור 52.



איור 52: למעלה: BDDs עבור f_{S_1}, f_{S_2}. למטה: שני מבני BDD עבור $f_{S_1 \wedge S_2}$.

- גודל ה-BDD בפרקטיקה: לינארי ב-n.
- סידור המשתנים מאוד משנה. למשל:

$$(x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2) \wedge \cdots \wedge (x_n \leftrightarrow y_n)$$

  - אם נבחר את הסדר להיות $x_1 y_1, x_2 y_2, \ldots$ מעולה .?. $2n \leq$ קודקודים ב-BDD.
  - אם נבחר את הסדר להיות $x_1 \ldots x_n y_1 \ldots x_n$. כאן $\mathcal{O}(2^n)$ קודקודים.

3.6.2 Bounded M.C

<u>בעיה:</u> בהינתן $K = (X, \varphi_R, \varphi_I), \varphi_T$, האם יש מסלול באורך k שמסתיים ב-T?

נסתכל על $\varphi_R, \varphi_I, \varphi_T$ כנוסחאות SAT.

<u>פתרון:</u> ננחש k השמות ל-X ונוודא שהן מסכימות על φ_R ו-φ_I.

- סימון: עבור φ מעל X, $\varphi[Y]$ היא החלפת המופעים של X ב-Y.
  - למשל, $Y = \{y_1, y_2\}, \varphi = (x_1 \vee x_2)$ ו-$\varphi[Y] = (y_1 \vee y_2)$.

- נסמן את k העותקים ב-$0 \leq i \leq k$, $X^i = \{x_1^i, \dots, x_n^i\}$. רדוקציה ל-SAT:

$$\varphi^k = \varphi_I \left[X^0\right] \wedge \bigwedge_{i=1}^{k} \varphi_R \left[X^{i-1}, X^i\right]$$

טענה. כל השמה מספקת ל-φ מתאימה למסלול באורך k ב-K.

דוגמה. (Shift Register)

$$\varphi_I = (\neg x \vee \neg y \vee \neg z)$$

$$\varphi_R = y \leftrightarrow x' \wedge z \leftrightarrow y' \wedge z$$

$$X^i = \left\{x^i, y^i, z^i\right\} \forall 0 \leq i \leq k$$

$$\begin{aligned}\varphi^k = &\left(\neg x^0 \vee \neg y^0 \vee \neg z^0\right)\\ &\wedge \left(\left(y^0 \leftrightarrow x^1\right) \wedge \left(z^0 \leftrightarrow y^1\right) \wedge z^1\right)\\ &\wedge \left(\left(y^1 \leftrightarrow x^2\right) \wedge \left(z^1 \leftrightarrow y^2\right) \wedge z^2\right)\\ &\vdots\end{aligned}$$

טענה. יש מסלול מ-I ל-T באורך $k \iff \varphi^k \wedge \varphi_T$ ספיקה.

פסודו-קוד של BMC באיור 53.

```
BMC (K, φ_T) :
    for k = 1, 2, ··· :
        construct φ^k
        if φ^k ∧ φ_T is SAT :
            ret k ⊭ G¬T
```

איור 53: פסודו-קוד של BMC.

לא עוצרים עד שמוצאים באג.

חלק III
סינתזה ומשחקים

1 Reactive Synthesis

1.1 הקדמה

1. (שאלה 1 מתרגיל הבית) בהינתן קוד שמורץ ע"י שני תהליכים, האם תיתכן גישה משותפת לקטע הקריטי?

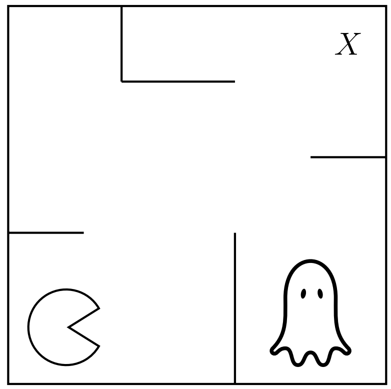
- אי-דטרמיניזם של הסביבה: ה-scheduler שבוחר איזה משני התהליכים להריץ בכל נקודת זמן.
- מטרה: האם יש בחירות של הסביבה שמובילות למצב רע (רוצים להראות שאין כאלה ⇐ המערכת חוקית)?

2. (שאלה 2 מתרגיל הבית) בהינתן מבוך, האם ניתן להגיע מנקודת ההתחלה לנקודת הסיום?

- אי-דטרמיניזם של המערכת: בחירה של צעד בכל נקודה.
- מטרה: האם יש בחירות של המערכת שמובילות למצב טוב?

3. משחק pacman, כמתואר באיור 54 - האם ה-pacman יכול להגיע ל-X בלי ש-ghost יגיע אליו?

- אי-דטרמיניזם של הסביבה (תזוזה אדברסריאלית של ghost) ושל המערכת (הצעדים של pacman).
- מטרה: לתכנן controller עבור pacman כך שללא תלות בהתנהגות של ghost מתנהג, pacman מנצח.



איור 54: משחק pacman.

1.2 מידול

נחלק את ה-AP לשתי קבוצות זרות $AP = I \sqcup O$, input ו-output. I מייצגות את ה-actions של ה-ghost, ו-O את ה-actions של pacman.

הגדרה.

1. $f_O : \left(2^I\right)^+ \to 2^O$ (ה-controller של pacman).
2. $f_I : \left(2^O\right)^+ \to 2^I$ (ה-controller של ghost).
3. $\left(2^{AP}\right)^\omega \ni out\left(f_I, f_O\right) = \underbrace{i_1 o_1}_{\pi_1 \in 2^{AP}} \underbrace{i_2 o_2}_{\pi_2} \cdots$, מוגדר אינדוקטיבית:
   - $i_1 = f_I\left(\varepsilon\right)$.
   - לכל $j \geq 1$, נגדיר

$$i_j = f_I\left(o_1, \ldots, o_{j-1}\right)$$

$$o_j = f_O\left(i_1, \ldots, i_j\right)$$

סינתזה:

- קלט: LTL φ מעל $AP = I \sqcup O$.

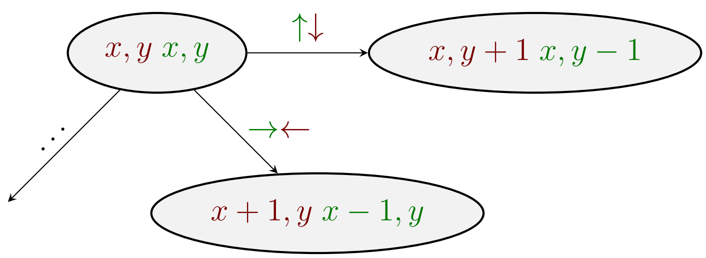
- פלט: f_O (אם קיים) כך שלכל f_I מתקיים $out\left(f_O, f_I\right) \vDash \varphi$.

הגדרה. φ היא realizable אם קיים f_O שטוב נגד כל f_I.

דוגמה. (פתרון pacman באמצעות סינתזה).
הפעולות של pacman הן $O = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$, ושל ghost $I = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$.

$$AP = \left\{ \underbrace{1 \leq x, y \leq n}_{\text{מיקום ה-pacman}}, \underbrace{1 \leq x, y \leq n}_{\text{מיקום ה-ghost}} \right\} \cup O \cup I$$

המחשה של מבנה הקריפקה באיור 55.



איור 55: מבנה הקריפקה עבור pacman (16 יציאות מכל מצב).

כעת, נגדיר את נוסחת המעברים, φ_R:

- בתרגיל, ייצגנו את המעברים בצורה סימבולית בתור $(x = x' \wedge y = y' + 1) \Rightarrow\uparrow$.
- כאן, נייצג באמצעות הדינמיקה של המשחק.

נסתכל על חישוב כלשהו $out\left(f_I, f_O\right)$:

$$\begin{array}{c|c} x\ y\ x\ y\ \text{Act} & \text{Act} \\ \hline \underbrace{0\ 0\ 10\ 10\ \uparrow}_{o_i} & \underbrace{\leftarrow}_{i_1} \\ \underbrace{0\ 1\ 9\ 10\ \rightarrow}_{o_2} & \underbrace{\downarrow}_{i_2} \\ \underbrace{1\ 1\ 9\ 9\ \cdots}_{o_3} & \vdots \end{array}$$

מכאן, נפרמל לפי הדינמיקה של pacman:

$$\begin{aligned}\varphi = & \left(x_p = 0 \wedge y_p = 0 \wedge x_g = 10 \wedge x_g = 10\right) \\ & \wedge G\left(\uparrow \rightarrow \left(x_p = Xx_p \wedge (y_p + 1) = Xy_p\right)\right) \\ & \wedge G\left(\leftarrow \rightarrow \left(x_g = X\left(x_g - 1\right) \wedge y_g = Xy_g\right)\right) \\ & \vdots\end{aligned}$$

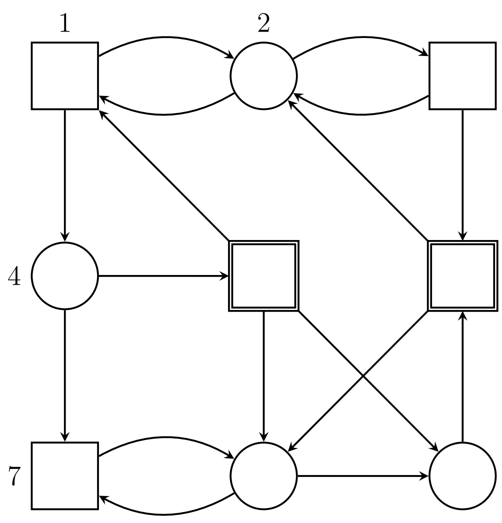
בנוסף, המפרט:

$$\underbrace{F\left(x_p = 10 \wedge y_p = 10\right)}_{\text{pacman מגיע למטרה}} \wedge \underbrace{G\left(\neg\left(x_p = x_g \wedge y_p = y_g\right)\right)}_{\text{ghost לא אוכל את pacman}} \wedge G\left(\neg\left(\underbrace{\cdots}_{\text{pacman לא מתנגש בקירות}}\right)\right)$$

נרצה למצוא f_O ש-realizes את φ.

2 משחקים על גרפים

משחק על גרף הוא מבנה מהצורה $\left(\underbrace{V, E}_{\text{גרף}}, (v_0), \underbrace{\alpha}_{\text{objective}}\right)$, כך ש-$V = V_1 \sqcup V_2$. נסמן בעיגולים שייכים לשחקן 1 (V_1) וריבועים ל-2 (V_2), כמו בדוגמא באיור 56.



איור 56: משחק על גרף.

סמנטיקה: למשל, משחקי ישיגות: בהינתן קבוצת מצבים מקבלים, שחקן 1 מנצח אם הוא יכול להכריח את המטבע להגיע לקודקוד מטרה. למשל, בדוגמא מאיור 56, כאשר קודקודי המטרה מסומנים בקו כפול.

- קל לראות שניתן לנצח מקודקוד 4 וקודקוד 7.
- עם זאת, הדבר לא אפשרי מקודקוד 1: שחקן 2 תמיד יוכל להכריח חזרה לקודקוד 2.

הגדרה. אסטרטגיה היא $f_1 : V^* \cdot V_1 \to V, f_2 : V^* \cdot V_2 \to V$. $V^* \cdot V_1$ זו היסטוריה שמסתיימת בקודקוד של שחקן 1.

הערה. אסטרטגיה חסרת זיכרון היא מהצורה $f_1 : V_1 \to V, f_2 : V_2 \to V$. האם באופן כללי כשיש אסטרטגיה מנצחת כלשהי לשחקן 1, בהכרח קיימת אסטרטגיה חסרת זיכרון?

הגדרה. עבור f_1, f_2, $out(v_0, f_1, f_2) = \pi_1\pi_2 \cdots \in V^\omega$, כך ש-$\pi_1 = v_0$ ולכל $i \geq 2$:

$$\pi_i = \begin{cases} f_1(\pi_i, \pi_{i-1}) & \pi_{i-1} \in V_1 \\ f_2(\pi_i, \pi_{i-1}) & \pi_{i-1} \in V_2 \end{cases}$$

הגדרה. אסטרטגיה f_1 מנצחת מ-v_0 אם לכל f_2 מתקיים $out(v_0, f_1, f_2) \vDash \alpha$.

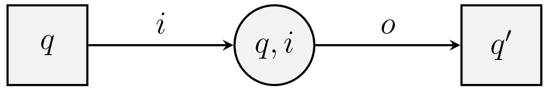
2.1 סינתזה ← פתירת משחק על גרף

בהינתן φ מעל $I \sqcup O$, נבנה A_φ דטרמיניסטי: V.W. לקבלת A_φ NBW, ואז עם Safra :DRW

$$A_\varphi = \left(\underbrace{\Sigma}_{2^{I \sqcup O}}, Q, \delta, q_0, \alpha\right)$$

$$\varphi \xrightarrow[\text{Safra}]{\text{V.W.}} A_\varphi$$

משחק על A_φ: נגדיר $V_1 = Q \times 2^I$, $V_2 = Q$, $E = \left\{(q,(q,i)) \mid i \in 2^I\right\} \cup \left\{((q,i), q') \mid \exists o \in 2^O : \delta(q, i \cup o) = q'\right\}$, באופן דומה לאיור 57.



איור 57: משחק על A_φ, $q' = \delta\left(q, i \cup o\right)$.

כעת, φ היא realizable $\iff$ שחקן 1 מנצח מ-q_0 במשחק על A_φ.

סיבוכיות: נניח שיש אלגוריתם פולינומיאלי שפותר משחקים על גרפים, ואז

$$\underset{n}{\varphi} \overset{\text{VW}}{\to} \underset{2^n}{\text{NBW } A_\varphi} \overset{\text{Safre}}{\to} \underset{2^{2^n n}}{\text{DRW } D_\varphi}$$

סינתזה היא 2-EXPTIME-hard (!).

<u>בעיה 1:</u> בהינתן G, v_0, הכרע האם שחקן 1 מנצח מ-v_0.

<u>בעיה 2:</u> בהינתן G, החזר חלוקה $V = W^1 \sqcup W^2$ כך ש-W^i מכיל את הקודקוד מהם שחקן i מנצח.

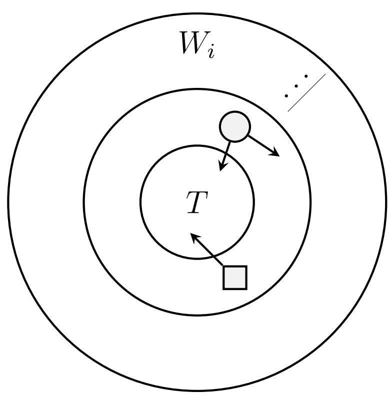
2.2 משחקי ישיגות (Reachability)

מוגדר ע"י $G=(V,E,T)$, $T\subseteq V$ מהווה את קודקודי המטרה. שחקן 1 מנצח עם π $\iff$ $\pi\vDash FT$.

- נבנה סדרה של קבוצות שמוכלות אחת בשנייה: כמתואר באיור 58. נסמן את הקבוצות ב-$W_0,W_1,\ldots$, כאשר W_i זו קבוצת הקודקודים מהם שחקן 1 יכול לנצח בתוך $i\geq$ צעדים.
  - אינדוקטיבית, $W_0=T$ ולכל $i\geq 0$:

$$\begin{aligned}W_{i+1}=W_i&\cup\{v\in V_1\mid\exists u\in W_i:(v,u)\in E\}\\&\cup\{v\in V_2\mid\forall v\in V:(v,u)\in E\Rightarrow u\in W_i\}\end{aligned}$$

  - עוצרים כאשר מגיעים לfixed point: $W_n=W_{n+1}$.
- נקרא לקודקודים של W_n לאחר הגעה ל-fixed point $\text{Reach}_1(T)$, ו-$\text{Safe}_2(T)=V\setminus\text{Reach}_1(T)$.
- אם $v_0\in\text{Reach}_1(T)$, שחקן 1 יכול לנצח.
- אחרת, שחקן 2 יכול לנצח: להכריח שהמשחק לא יגיע לעיגול הירוק.



איור 58: הקבוצות W_i בגרף.

2.3 משחקי Buchi

מוגדר עי $G=(V,E,T)$, $T\subseteq V$. שחקן 1 מנצח עם π $\iff$ $\pi\vDash GFT$.
נרצה לפתוח אלגוריתם:

- נתחיל מ-reachability על G עם שחקן 1 (הגעה ל-T), ונגדיר $W_0=\text{Safe}_2^G\left(T\right)$.

טענה. $W_0\subseteq W^2$: שחקן 2 יכול למנוע אפילו ביקור אחד ב-T, קל וחומר מספר סופי של ביקורים.

- נמשיך עם reachability על G עם שחקן 1 (הגעה ל-W_0), ונגדיר $W_1=\text{Reach}_2^G\left(W_0\right)$.

טענה. $W_1\subseteq W^2$: שחקן 2 יכול לגרור את המשחק ל-W_0 ומשם מנצח.

- כעת, נגדיר את הגרף $G'=\left(\underbrace{V\setminus\left(W_0\cup W_1\right)}_{V'},\underbrace{E|_{V'}}_{E'},\underbrace{T|_{V'}}_{T'}\right)$. נפעיל reachability על G' עם שחקן 1 (הגעה ל-T'), ונגדיר $W_0'=\text{Safe}_2^{G'}\left(T'\right)$.

טענה. $W_0'\subseteq W^2$: שחקן 2 מכריח את G' להישאר ב-W_0'. אם שחקן 1 בורח מ-W_0', הוא יגיע ל-$W_1\cup W_0$.

- נמשיך באופן הזה, עד להגעה ל-fixed point ב-G^n, בו $\text{Reach}_1^{G^n}\left(T\right)=V^n$.

טענה. $\text{Reach}_1^{G^n}\left(T\right)=W^1$.

הוכחה. תחילה, $\text{Reach}_1^{G^n}\left(T\right)\supseteq W^1$ מאחר ומשאר הקודקודים שחקן 2 מנצח. עבור $\text{Reach}_1^{G^n}\left(T\right)\subseteq W^1$, יהי $v\in\text{Reach}_1^{G^n}\left(T\right)$:

1. אם $v\notin T$, שחקן 1 גורר את המשחק ל-T.
2. אחרת, יש ל-v שכן ב-V^n: בחר שכן שרירותית וחזור ל-1.

כך בכל מקרה נגיע לגרירה לעבר T, ולכן $v\in W^1$. □

תיאור האלגוריתם הפורמלי באיור 59. סיבוכיות הזמן היא $\mathcal{O}\left(|V|\,|E|\right)=\mathcal{O}\left(|V|^3\right)$. Chatterjee, Henzinger: $\mathcal{O}\left(|V|^2\right)$.

$$
\begin{array}{|l|}
\hline
\textbf{Buchi}\,(V,E,T):\\
\quad \textbf{if}\ \text{Reach}_1(T)=V:\\
\quad\quad \textbf{ret}\ W^1=V, W^2=\emptyset\\
\quad W_0=\text{Safe}_2(T), W_1=\text{Reach}_2(W_0)\\
\quad W^1,W^2=\\
\textbf{Buchi}\,(V\setminus(W_0\cup W_1), E|_{V'}, T|_{V'})\\
\quad \textbf{ret}\ W^1, W^2\cup W_1\cup W_0\\
\hline
\end{array}
$$

איור 59: פסודו-קוד של האלגוריתם למשחקי Buchi.

2.4 משחקי Rabin ו-Parity

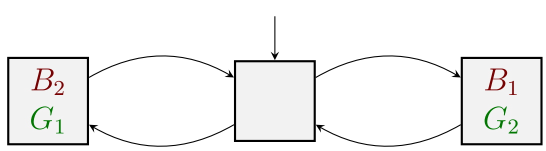
מוגדר ע"י $G=\left(V,E,\{(B_i,G_i)\}_{i=1}^{k}\right)$. שחקן 1 מנצח מסלול π $\Longleftrightarrow$

$$\pi\vDash\bigvee_{i=1}^{k}\left(FG\neg B_i\wedge GFG_i\right)$$

משפט. (אסטרטגיה חסרת זיכרון)

1. אם שחקן 1 מנצח, יש לו אסטרטגיה מנצחת חסרת זיכרון: $f_1:V_1\to V$.
2. אם שחקן 2 (שחקן Streett) מנצח, לא בהכרח יש לו אסטרטגיה מנצחת חסרת זיכרון: $f_1:V_1\to V$.

דוגמה. הוכחת הסעיף השני: במשחק המתואר באיור 60 קיימת אסטרטגיה עם זיכרון, ולא קיימת אחת חסרת זיכרון ($\infty G_1\to\infty B_1\wedge\infty G_2\to\infty B_2$).



איור 60: משחק Rabin ללא אסטרטגיה מנצחת לשחקן 2.

יש אסטרטגיה עם שימוש בזיכרון, ואין אחת חסרת זיכרון.

משפט. פתירת משחקי Rabin היא NP-שלמה.

<u>חסם עליון:</u> נראה ש-Rabin ב-NP.

1. ננחש אסטרטגיה חסרת זיכרון של שחקן 1, $f_1 : V_1 \to V$.
2. נוודא ש-f_1 מנצחת מ-v_0: נחפש תגובה f_2 של שחקן 2 שמנצחת את f_1:
   - נבנה גרף ע"י מחיקת כל הקשתות שלא מסכימות עם f_1:

$$E' = \{(u,v) \in E \mid u \in V_2\} \cup \{(u, f_1(u)) \in E \mid u \in V_1\}$$

   - נחשוב על G^1 כאוטומט מעל א"ב Σ, $|\Sigma| = 1$ עם תנאי קבלה $\overline{\alpha}$, ונבדוק ריקנות.
     - G^1 לא ריק $\iff$ יש תגובה של שחקן 2 שמנצחת את f_1 $\iff$ f_1 לא מנצחת.

משפט. בדיקת ריקנות של אוטומטי Streett היא ב-P.

משחקי Parity

- אם שחקן 1 מנצח, יש לו אסטרטגיה מנצחת חסרת זיכרון.
- הדואלי של Parity, הוא Parity.
- בדיקת ריקנות של אוטומט Parity היא ב-P.

$\Rightarrow$ Parity $\in$ NP $\cap$ co-NP