אוטומטים מעל עצמים אינסופיים ־ 67663

חיים שחור ־ סיכומי הרצאות של אורנה קופרמן

י"ח אדר תשע"ג (שעור 1)

הערה 0.1 מי שמעוניין לסייע בשרטוט האוטומטים מתבקש לפנות אלי.

בחישוביות דיברנו על אוטומטים ושפות רגולריות. שפות חסרות הקשר ־ לא מוסיף כוח. אח"כ במכונות טיורינג ושפות כריעות יש מגמה להרחיב את אוסף השפות המזוהות. בכל השלב הזה דברנו על שפות $L \subseteq \Sigma^*$. ההנחה היא שהמכונה מקבלת קלט סופי, ומוציאה פלט סופי. בקורס הנוכחי נדבר לא על שפות ב־Σ^* אלא שפות שמייצגות אינטראקציה מתמשכת עם הסביבה.

למשל, שתי מעליות בבנין עם 20 קומות. בכל קומה יש שני כפתורים לעלות ולרדת, ואנחנו מחפשים אלגוריתם שיפעל כל הזמן, והמטרה היא למזער את זמן ההמתנה למעלית. הפעילות היא מסוג on-going. האינטראקציה היא בכל קומה האם לחוץ למעלה או לא, ואם לחוץ למטה או לא. במקרה הזה $\Sigma = 2^{\{\uparrow,\downarrow\}\times[20]}$. תכנון האלגוריתם יכול להתחשב בתפוסת המעלית, האם מותר להעביר אנשים ממעלית למעלית. בהתחלה מגיע $x_0 \in \Sigma$, ונפלט y_0. אח"כ מגיע x_1 ונפלט y_1, וכך הלאה. לכן אנחנו לא מדברים על שפות סופיות, אלא שפות של מילים אינסופיות.

גם מערכות הפעלה מתעסקות בהתנהגות on-going. אפשר לדבר על מילים אינסופיות, או עצים אינסופיים.

הקורס הוא בעיקר תיאורטי. נזכיר את המוטיבציה.

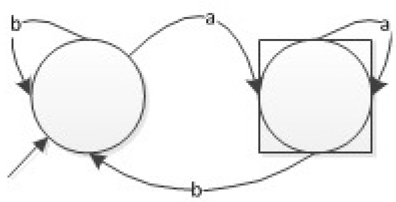
אדמיניסטרציה ־ יש אתר מודל. יהיו בקורס 3־4 תרגילי בית, עם 25% מהציון (חצי מגן). אפשר להגיש בזוגות. במודל פרק בספר שמדבר על אוטומטים.

1 אוטומט מעל מילה אינסופית

סוג השפות: אם הא"ב הוא $\Sigma = \{a, b\}$, אזי $L_1 = \{w \mid w\, contains\, \infty\, "a"\}$.
לדוגמא, $ababab \ldots \in L_1$ אבל $aabbbbbb \ldots \notin L_1$.

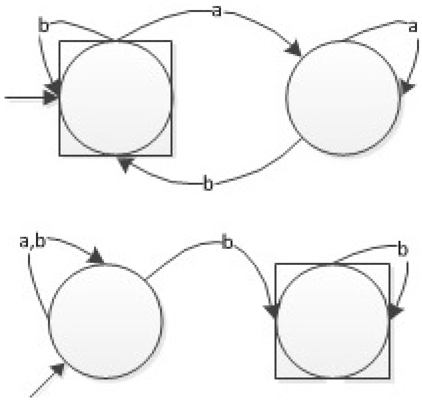
1.1 תנאי קבלה בוקי Buchi

אוטומט ייראה $\mathcal{A} = \langle \Sigma, Q, Q_0, \delta, \alpha \rangle$. עם $\delta : Q \times \Sigma \to 2^Q$ (בלי מעברי ε) ו־$\alpha \subseteq Q$.
למשל, האוטומט הסופי שמקבל את המילים המסתיימות ב־a (שני מצבים, a הולך למצב המקבל, b למצב ההתחלה). אם האוטומט מקבל כקלט מילה אינסופית (בוקי), ריצה תוגדר באופן דומה. נגדיר ריצה $r : \mathbb{N} \to Q$ מעל המילה $w = \sigma_1 \sigma_2 \ldots$ כך ש־$r(0) \in Q_0$, ולכל $i \geq 0$, $r(i+1) \in \delta(r(i), \sigma_{i+1})$. באוטומטים סופיים היינו בודקים את המצב בסוף המילה. במקרה של עצמים אינסופיים, נדבר על $inf(r) = \{q \in Q \mid r(i) = q\, for\, \infty\, i\}$. תנאי הקבלה של האוטומט: r מקבלת אס"ם $inf(r) \cap \alpha \neq \emptyset$. A מקבל את w אם קיימת ריצה r מקבלת של $\mathcal{A}_1$ על w. לכן במקרה של האוטומט שתיארנו, עוברים למצב מקבל כל פעם כשרואים a. לכן $L(\mathcal{A}_1) = L_1$.



איור 1: $\mathcal{A}_1$

איך מייצרים אוטומטים משלימים? באוטומטים סופיים דטרמיניסטיים הופכים את המצבים המקבלים. אם נהפוך כאן, נקבל את $L(\mathcal{A}_2) = \infty b$. השפה $L_3 := \{w \mid \#_a(w) \in \mathbb{N}\}$ נצטרך לבנות עבורו אוטומט. בשפה זו, מתישהו מתחיל זנב של b. לכן ניצור אוטומט $\mathcal{A}_3$ א"ד שיקבל זנב כזה, נוכיח שאין אוטומט דטרמיניסטי לשפה הזו. אוטומט חורצני (דטרמיניסטי). לחרצן - לעשות דטרמיניזציה. אפשר לחרצן אוטומטים מעל מילים סופיות כי הוא מכיל מספר סופי של אפשרויות. כאן יש לנו מצב אינסופי של אפשרויות.

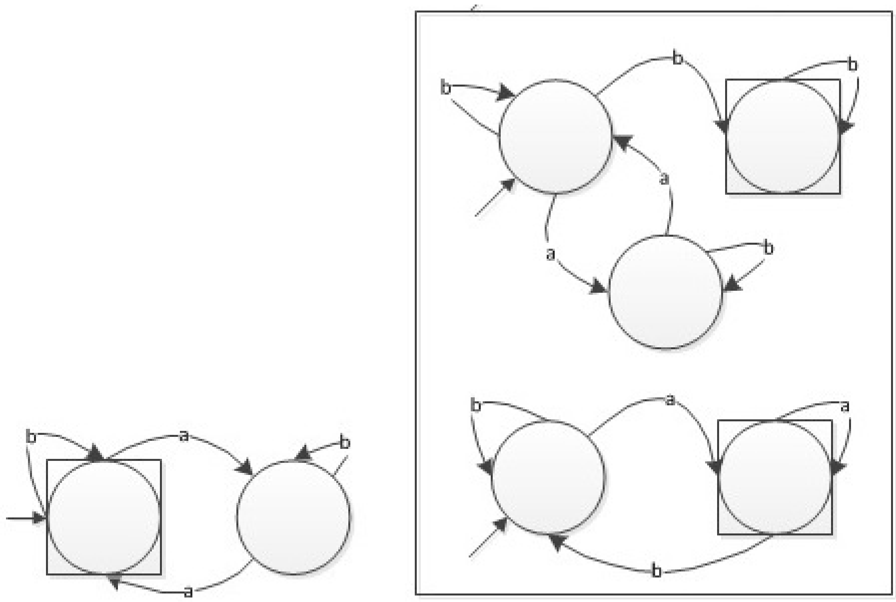


איור 2: $\mathcal{A}_2$ ו־$\mathcal{A}_3$

למעלה נמצא האוטומט הדטרמיניסטי $\mathcal{A}_2$, ולמטה האוטומט הא"ד $\mathcal{A}_3$.

ננסה לצייר אוטומט A_4 עבור השפה בה $L_4 := \{w \mid \#_a(w) \in 2\mathbb{N} \vee \#_a(w) = \infty\}$. קל להכליל את A_3 שיכלול מספר סופי זוגי, ואפשר לעשות אוטומט איחוד ע"י הנחת שני האוטומטים ביחד, ואיחוד המצבים המתחילים בהם.

האם קיים A_4 עם שני מצבים? יש לנו אוטומט שמחליף מצבים ב־a. צריך להוכיח: אם המילה מכילה מספר סופי של a, היא תעשה את הסיבוב מס' סופי, ותעצור במצב ההתחלה, ותשאר שם בצורה סופית. אם יש מס' אינסופי, היא תעשה אינסוף סיבובים, ותבקר במצב ההתחלה אינסוף פעמים. אם המספר אי־זוגי, ניתקע במצב הלא מקבל.



איור 3: $\mathcal{A}_4$

מימין אוטומט האיחוד הא"ד לשפה L_4, משמאל אוטומט דטרמיניסטי לאותה שפה עם שני מצבים.

סימון: NBW, DBW הוא סימון לאוטומטים מעל מילים אינסופיות.

באופן כללי, המודלים מכילים $\{N, D, U, A\}$ - א"ד, דט', אוניברסלי (כל הריצות מקבלות). סוג המודל $\{F, B, C, P, R, S\}$, ורץ על $\{W, T\}$ (מילים או עצמים).

סימון כמו $NFW = DFW$ אומר שניתן לחרצן אוטומטים מעל מילים סופיות, אבל $NBW > DBW$ אומר שלא ניתן לחרצן כל אוטומט מעל מילים אינסופיות.

ביטויים ω־רגולריים הם $L \subseteq \Sigma^\omega$. נזכיר כי ביטוי רגולרי כולל: $\emptyset, \varepsilon, \sigma \in \Sigma, r_1 + r_2, r_1 r_2, r_1^*$. זה מגדיר שפה שהיא תת־קבוצה של Σ^*.

ביטויים ω־רגולריים:

- $\emptyset$.
- r^ω, rs_1, $s_1 + s_2$ עבור ביטוי רגולרי r וביטויים ω־רגולריים s_1, s_2.

הערה: $\omega = |\mathbb{N}|$ ($(ab)^\omega = abababab\ldots$). שרשור ניתן לביצוע רק לאיבר באורך סופי.

השפה $L = \{w \mid \#_a(w) \geq 2\}$ מבוטאת ע"י הביטוי הבא: $b^*ab^*a\,(a+b)^\omega$. כל המילים עם אינסוף a היא $(b^*a)^\omega$. ביטוי עבור מספר סופי של a הוא $(a+b)^*\,b^\omega$.

עבור ביטוי ω־רגולרי s, ניתן להגדיר $L(s)$.

נדבר על תכונות סגור של אוטומטים. חיתוך וכד'.

1.1.1 תכונות סגור של NBW

1. איחוד ־ עובד באותו אופן של איחוד NFW.
2. חיתוך ־ עבור NFW השתמשנו באוטומט המכפלה.

$$A_1 \times A_2 = \left\langle \Sigma, Q_1 \times Q_2, Q_1^0 \times Q_2^0, \delta, F_1 \times F_2 \right\rangle$$

כאשר $\delta(\langle q_1, q_2 \rangle, \sigma) = \delta_1(q_1, \sigma) \times \delta_2(q_2, \sigma)$. מתי זה לא יעבוד לאוטומטי בוקי? המשמעות בNFW היא שאנחנו מדמים ריצה במקביל על שני האוטומטים, ובודקים האם הסוף הוא מצב מקבל בשניהם. ב־NBW, אם נגדיר $\alpha = \alpha_1 \times \alpha_2$, זה ידרוש שהביקורים האינסופיים במצב המקבל יהיו אינסופיים. למשל, שני אוטומטים לשפה $L = \{a^\omega\}$ עם שני מצבים מזגזגים, בכל אחד מצב מקבל אחר. אם נגדיר אוטומט מכפלה בצורה הזו, השפה תהיה $\emptyset$.

נרצה לשרטט אוטומט של שפה שמכיל אינסוף a ואינסוף b. ניצור אוטומט עם מצב התחלתי מקבל, למצב a נלך ונחזור עם b, ולמצב b נלך ונחזור עם a. זהו אוטומט דטרמיניסטי עם 3 חלקים. אפשר גם אוטומט אחר ־ במצב ההתחלה מחכים ל־a, במצב הזה מחכים ל־b ועוברים למצב מקבל, ומשם חוזרים לנקודת ההתחלה. אנחנו לפעמים מוותרים על כמות מסויימת של b בלי לספור אותם, כי יש לנו ∞ bים. זה גם מה שנעשה בהגדרה של החיתוך.

יש לנו סימולציה במקביל של A_1, A_2. נרצה לוודא שיש אינסוף ביקורים בα_1, α_2. נתחיל מ־A_1, כשנגיע לביקור ב־α_1, נחפש ביקור ב־α_2 בסימולציה השניה, וחוזר חלילה. אם יש אינסוף, נמצא אותם מתישהו. לכן אוטומט החיתוך בצורה פורמלית יהיה $A_1 = \langle \Sigma, Q_1, Q_0^1, \delta_1, \alpha_1 \rangle$ ו־$A_2 = \langle \Sigma, Q_2, Q_0^2, \delta_2, \alpha_2 \rangle$, נגדיר את אוטומט המכפלה

$$A = A_1 \times A_2 = \langle \Sigma, Q, Q_0, \delta, \alpha \rangle$$

כאשר:

- $Q = Q_1 \times Q_2 \times \{1, 2\}$.
- $Q_0 = Q_1^0 \times Q_2^0 \times \{1\}$.

- $\delta\left(\left\langle q_1,q_2,i\right\rangle,\sigma\right)=\delta_1\left(q_1,\sigma\right)\times\delta_1\left(q_2,\sigma\right)\times\{j\}$ כאשר $j=\begin{cases}i & q_i\notin\alpha_i\\ 3-i & q_i\in\alpha_i\end{cases}$.
- $\alpha=Q_1\times\alpha_2\times\{2\}$.

הוכחה: $L\left(A\right)=L\left(A_1\right)\cap L\left(A_2\right)$. נראה כיוון אחד. נניח $w\in L\left(A\right)$. תהי $r=\left(\left\langle q_1^n,q_2^n,i_n\right\rangle\right)_{n\in\mathbb{N}}$ ריצה מקבלת של A על w. נראה כי $r_1=\left(q_1^n\right)_{n\in\mathbb{N}}$ ריצה מקבלת של A_1 על w, וכי $r_2=\left(q_2^n\right)_{n\in\mathbb{N}}$ ריצה מקבלת של A_2 על w. בסדרה r יש ∞ ביקורים ב־$Q_1\times\alpha_2\times\{2\}$, כלומר אינסוף אינדקסים j כך ש־$q_2^j\in\alpha_2$. לכן r_2 היא ריצה מקבלת. נראה כי בין כל שני אינדקסים $j_1<j_2$ כך ש־$\left\langle q_2^{j_1},q_2^{j_1},i_{j_1}\right\rangle\in\alpha$ וכן $\left\langle q_2^{j_2},q_2^{j_2},i_{j_2}\right\rangle\in\alpha$ קיים אינדקס $j_1<j_3<j_2$ כך ש־$q_1^{j_3}\in\alpha_1$. מהגדרת δ נובע $i_{j_1+1}=1$ כי אחרי ביקור ב־$q_2^j\in\alpha_2$ ו־$i_j=2$, עוברים ל־$i_{j+1}=1$. מאחר ו־$i_{j_2}=2$, קיים אינדקס j_3 כך ש־$i_{j_3+1}=2$, אולם $i_{j_3}=1$. מהגדרת δ מובטח כי $q_1^{j_3}\in\alpha_1$. ■

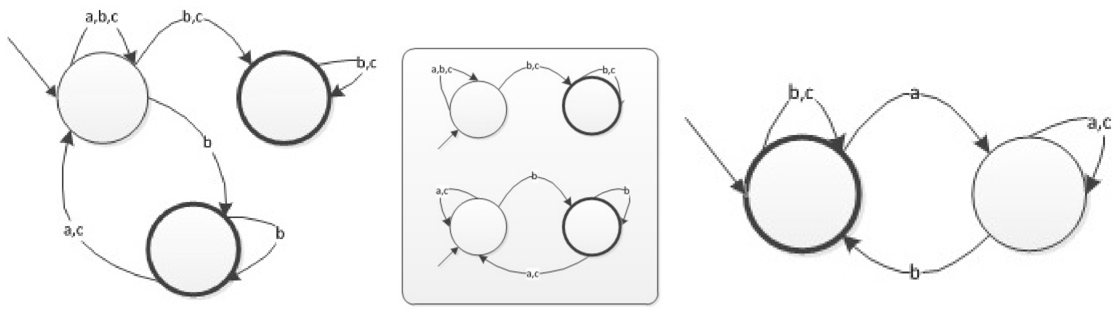
נשים לב כי $|A|=|Q_A|=O\left(|A_1|\,|A_2|\right)$ כאשר בחיבור $|A|=O\left(|A_1|+|A_2|\right)$. נראה כי בהשלמה המצב הרבה יותר חמור. מה לגבי סגור לאיחוד וחיתוך דטרמיניסטי? אוטומט המכפלה שראינו משמר דטרמיניזם, וכבר ראינו אותו. באיחוד, ניקח את אוטומט המכפלה עם $\alpha=\alpha_1\times Q_2\cup Q_1\times\alpha_2$ ואין צורך בעוד מרכיב למכפלה.

שבוע הבא נדבר על חיתוך. תרגיל אוטומט ל־אם יש אינסוף a יש אינסוף b (מעל $\Sigma=\{a,b,c\}$).

כ"ה אדר תשע"ג (שעור 2)

באוטומט על הלוח - $L\left(\mathcal{A}\right)=\{w\mid after\,every\,"a"\,there\,is\,"b"\}$. נסתכל על השפה $L':\infty a\rightarrow\infty b$, אזי $ac^{\omega}\in L\backslash L'$ כי $L'=\neg\infty a\vee\infty b$. אפשר ליצור אוטומט איחוד בין $\neg\infty a$ לבין ∞b, ואפשר לצמצם אותו.

אפשר לדבר על רמת החופש של אוטומט. יש ל־b שלוש אפשרויות ממצב ההתחלה. אם יש ∞a ופנינו ימינה לכיוון "יש מס' a סופי" היינו נתקעים, אבל אם יש מס' סופי של b, ופנינו למטה לכיוון "יש אינסוף b", יש לנו הזדמנות לתקן.



איור 4: אוטומטים לשפה $\infty a \to \infty b$

האוטומט מימין אינו מתאים לשפה, כי הוא דוחה את ac^{ω}. האוטומט האמצעי מסתכל על האיחוד של $\neg\infty a \vee \infty b$, והאוטומט משמאל מצמצם אותו. הדיון על האפשרויות עבור האות b מתייחס לאוטומט זה.

מה לגבי L תחת $\Sigma = \{a, b\}$? האם $\infty b =_{?} L$? כן, כי אם אין ∞b, בהכרח ∞a, והתנאי לא מתקיים.

ראינו שהשלמה של $L = \infty a$ אינה מתקבלת ע"י דואליזציה של תנאי הקבלה אפילו כשהאוטומט דטרמיניסטי. למה זה עובד במילים סופיות? כי שם מסתכלים על המצב האחרון, ואם נהפוך את המצבים, השלמנו את השפה. במילים אינסופיות, בתנאי הקבלה של בוקי אנחנו מבקרים אינסוף פעמים במצב מקבל. אם נהפוך, זה אומר שביקרנו אינסוף פעמים במצבים הלא מקבלים. זה לא אומר לגבי מס' הביקורים במצבים המקבלים המקוריים. לכן הגדרת $\alpha' = Q \backslash \alpha$ אינה משלימה את השפה.

ראינו אוטומט שישלים את השפה L. נרצה לבנות אוטומט שמשלים $\overline{NBW} \to DBW$.

משפט 1.1 בהינתן $\mathcal{A}$ DBW, יש $\mathcal{A}'$ NBW כך ש־$L(\mathcal{A}') = \Sigma^{\omega} \backslash L(\mathcal{A})$.

הרעיון: $\mathcal{A}'$ ינחש מתי $\mathcal{A}$ מפסיק לבקר ב־α. $\mathcal{A}' = \left\langle \Sigma, Q', \delta', q_0', \alpha' \right\rangle$ כאשר

- $Q' = (Q \times \{1\}) \cup ((Q \backslash \alpha) \times \{2\})$
- $q_0' = \langle q_0, 1 \rangle$
- $\alpha = (Q \backslash \alpha) \times \{2\}$

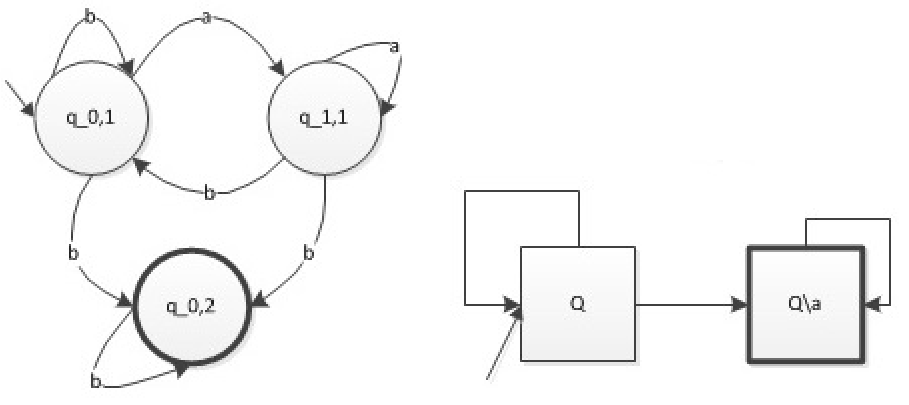
• יהי $s=\delta\left(q,\sigma\right)$, אזי

$$\begin{aligned}\delta'\left(\left\langle q,1\right\rangle ,\sigma\right) & = & \begin{cases}\left\{ \left\langle s,1\right\rangle ,\left\langle s,2\right\rangle \right\} & s\notin\alpha\\\left\{ \left\langle s,1\right\rangle \right\} & s\in\alpha\end{cases}\\\delta'\left(\left\langle q,2\right\rangle ,\sigma\right) & = & \begin{cases}\left\{ \left\langle s,2\right\rangle \right\} & s\notin\alpha\\\emptyset & s\in\alpha\end{cases}\end{aligned}$$

או בצורה יפה יותר לכתיבה:

$$\begin{aligned}\delta'\left(\left\langle q,1\right\rangle ,\sigma\right) & = & \left(\delta\left(q,\sigma\right)\times\left\{ 1\right\} \right)\cup\left(\left(\delta\left(q,\sigma\right)\backslash\alpha\right)\times\left\{ 2\right\} \right)\\\delta'\left(\left\langle q,2\right\rangle ,\sigma\right) & = & \left(\delta\left(q,\sigma\right)\backslash\alpha\right)\times\left\{ 2\right\} \end{aligned}$$

דוגמא: לאוטומט שראינו עם ∞a, (התחלה ב־q_0, $\alpha=\left\{ q_1\right\}$). האוטומט ניתן לצמצום למה שכבר ראינו.



איור 5: אוטומט ההשלמה

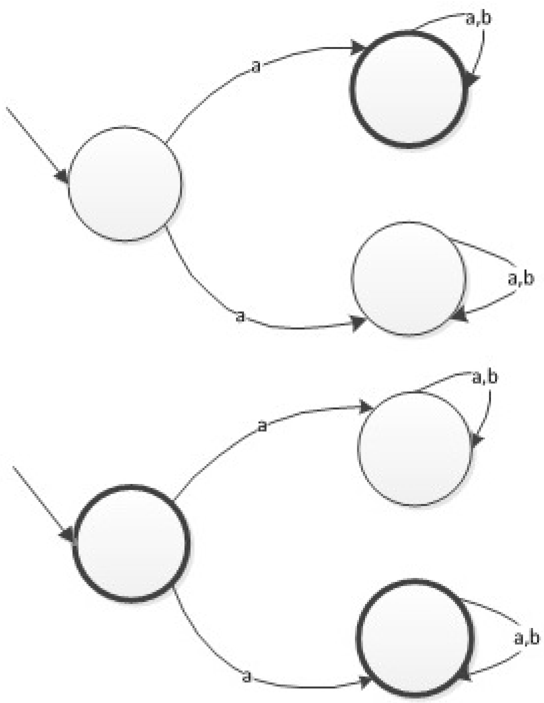
מימין תיאור של אוטומט כללי, שמחזיק "עותק וחצי" של האוטומט $\mathcal{A}$ המקורי. משמאל יישום של הבנייה עבור האוטומט מאיור 1 של $L=\infty a$.

טענה 1.2 הבניה נכונה - $L\left(\mathcal{A}'\right)=\Sigma^{\omega}\backslash L\left(\mathcal{A}\right)$

הוכחה: $L\left(\mathcal{A}'\right)\subseteq\Sigma^{\omega}\backslash L\left(\mathcal{A}\right)$: ריצה $\left\{ \left\langle q_n,c_n\right\rangle \right\}_{n=0}^{\infty}$ ב־$\mathcal{A}'$ מתאימה לריצה $\left\{ q_n\right\}_{n=0}^{\infty}$ ב־$\mathcal{A}$. תהי r' ריצה מקבלת על w ב־$\mathcal{A}$, הריצה r המתאימה לה ב־$\mathcal{A}$ היא הריצה היחידה של $\mathcal{A}$ על w, והיא דוחה.

$L(\mathcal{A}')\subseteq \Sigma^\omega \backslash L(\mathcal{A})$: תהי $w\notin L(\mathcal{A})$, הריצה r של $\mathcal{A}$ על w מבקרת ב־α רק מספר סופי של פעמים. ניתן לבנות ריצה מקבלת של $\mathcal{A}'$ על w. ■

איזה כיוון היה נופל אם $\mathcal{A}$ היה א"ד? כמו שאוטומט על מילים סופיות קשה להשלים. למשל יש אוטומט לשפה $a\left(a+b\right)^*$ שאם נהפוך את המצבים המקבלים נקבל $\varepsilon+a\left(a+b\right)^*$. באופן כללי, A' מקבל את w אם קיימת ריצה דוחה של A על w, ומה שאנו צריכים זה לקבל אם כל הריצות של A על w דוחות.



איור 6: אוטומטים "משלימים" לשפות $a\left(a+b\right)^*$ ו־$\varepsilon+a\left(a+b\right)^*$

איך עושים השלמה במילים סופיות? מחרצנים את האוטומט. בפעולה הקודמת, עברנו מ־n מצבים ל־$2n$ מצבים בתהליך ההפיכה. באוטומט המכפלה שראינו היה לנו $O\left(n_1n_2\right)$.

משפט 1.3 $NBW>DBW$, כלומר קיימות שפות המזוהות עם אוטומט א"ד, ולא עם אוטומט דטרמיניסטי. more expressive

הוכחה: נתבונן $L=\neg\infty a$, ונראה $L\in NBW$ ו־$L\notin DBW$. את $L\in NBW$ כבר

ראינו. נניח בשלילה כי יש $\mathcal{A}$ DBW כך ש־$L(\mathcal{A}) = L$. יהי $n = |Q|$.

נתבונן במילה $w_1 = b^{\omega} \in L$. יש אינדקס i_1, כך ש־$\delta(b^{i_1}) := \delta'(q_0, b^{i_1}) \in \alpha$.

נתבונן במילה $w_2 = b^{i_1} a b^{\omega} \in L$. יש i_2 כך ש־$\delta(b^{i_1} a b^{i_2}) \in \alpha$.

נתבונן במילה $b^{i_1} a b^{i_2} a b^{\omega} \in L$, ופה ההוכחה מתפצלת. עם אינטואיציה טובה, יש סדרה אינסופית של מילים, וממנה אפשר לייצר מילה עם ∞a.

ניתן להמשיך ולהגדיר לכל $k \geq 1$ את המילה $w_k = b^{i_1} a b^{i_2} a b^{i_3} \ldots b^{i_k} a b^{\omega}$ כך שבריצה של $\mathcal{A}$ על w_k, יש ביקור ב־α לפני כל a.

עבור $k > n$, לפי עקרון שובך היונים יש $j_1 < j_2$ כך ש־$\delta(b^{i_1} a \ldots b^{i_{j_1}}) = \delta(b^{i_1} a \ldots b^{i_{j_2}}) = q \in \alpha$. נתבונן במילה $w = b^{i_1} a \ldots b^{i_{j_1}} (a b^{i_{j_1+1}} \ldots b^{i_{j_2}})^{\omega}$.

1. $w \notin L$ כי יש ב־w ∞a.
2. הריצה של $\mathcal{A}$ על w תבקר ∞ פעמים ב־q.

■

למה 1.4 (קניג konig) ־ בעץ עם ∞ קודקודים ודרגת פיצול סופית יש מסלול אינסופי.

ראינו שאי־דטרמיניזם מוסיף לכוח ההבעה. מכאן אפשר לעבור לבעיית ההכרעה: נתון אוטומט NBW, האם יש אוטומט DBW שקול לו? נראה שיש פתרון ב$LOGSPACE$, תלוי בייצוג של האוטומט.

בוקי בא עם ההגדרה של האוטומט ב62. לנדובר Landweger ב69 הגדיר עבור שפה $R \subseteq \Sigma^*$, את

$$\Sigma^{\omega} \supseteq \lim(R) := \{w \mid w\, has\, \infty\, prefixes\, in\, L\}$$

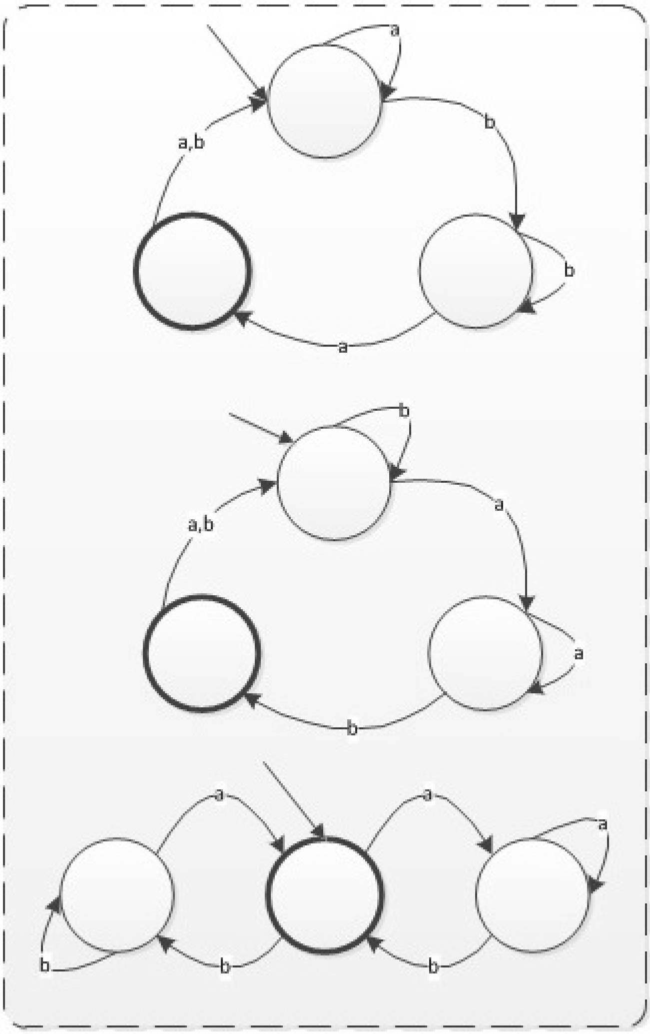
למשל עבור המילה $R = (a+b)^* a$, $\lim(R) = \infty a$. לשפה $R = (ab)^*$ יש $\lim(R) = (ab)^{\omega}$. גם עבור $R' = (ab)^* a$ נקבל $\lim(R') = (ab)^{\omega}$. לכן מ־R לגבול שלה יש שפה יחידה, אבל משפה אינסופית, ייתכנו כמה שפות שמייצרות אותה.

עבור $L = \neg\infty a$, איזו שפה R תקיים $\lim(R) = L$?

משפט 1.5 $L \in DBW$ אם"ם קיימת $R \subseteq \Sigma^*$ רגולרית כך ש־$\lim R = L$.

הוכחה: $\Leftarrow$: תהי R רגולרית, ו־$\mathcal{A}$ DFW עבור R, נתייחס ל־$\mathcal{A}$ כאוטומט בוקי. אזי $L_{\omega}(\mathcal{A}) = \lim(L_*(\mathcal{A}))$. הכיוון השני אותו דבר כמעט. ■

ראינו לשפה $\infty a \wedge \infty b$ 3 DBW שונים. אם נסתכל עליהם בתור DFW, הם מגדירים 3 שפות שונות.



איור 7: 3 אוטומטים DBW שונים לשפה $\infty a \wedge \infty b$

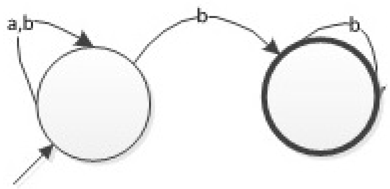
$L_{\omega}\left(\mathcal{A}_1\right)=L_{\omega}\left(\mathcal{A}_2\right)=L_{\omega}\left(\mathcal{A}_3\right)=\infty a \wedge \infty b$, אבל: $L_*\left(\mathcal{A}_1\right)=a^*b^*a$, $L_*\left(\mathcal{A}_2\right)=b^*a^*b$ ואלו $L_*\left(\mathcal{A}_3\right)=\left(aa^*b+bb^*a\right)^*$

בעיה: בהינתן $\mathcal{A}$ NBW, בנה $\mathcal{A}'$ NFW כך ש-$L\left(\mathcal{A}\right)=\lim\left(L\left(\mathcal{A}'\right)\right)$ או הכרע שאין כזה. אם היינו יודעים לחרצן אוטומטי בוקי, היינו יכולים לפתור את הבעיה.

אלג' 1: נגדיר את $\mathcal{A}'$ להיות $\mathcal{A}$ ונקווה שאם $L\left(\mathcal{A}\right) \in DBW$, מתקיים תמיד

$\lim L(\mathcal{A}') = L(\mathcal{A})$ (כמו בדרטמיניסטיים).

זה לא נכון. דוגמא נגדית: אנו מחפשים $\mathcal{A}$ כך ש־$L(\mathcal{A}) \in DBW$, אבל כשנתבונן על $\mathcal{A}$ כעל NFW, אז $\lim(L_*(\mathcal{A})) \neq L_\omega(\mathcal{A})$ (תרגיל). ראינו עבור האוטומט לשפה $\neg\infty a$ שזה לא עובד $L_*(\mathcal{A}) = (a+b)^* b$ והגבול הוא ∞b. אבל זו שפה שאין לה אוטומט דטרמיניסטי.



איור 8: האוטומט עבור $\neg\infty b$

$L_*(\mathcal{A}) = (a+b)^* b$, $L_\omega(\mathcal{A}) = \neg\infty b$, אולם $\lim(L_*(\mathcal{A})) = \infty b$

אלג' 2: חרצן את $\mathcal{A}$ ל־$\mathcal{D}$ DBW (טרם למדנו, אקספוננציאלי), $\mathcal{A}'$ המבוקש הוא $\mathcal{D}$כשמתייחסים אליו כ־DFW.

שאלה: האם הניפוח האקס' חיוני או שאפשר להחזיר אוטומט עם פחות מצבים. אין לנו חסם תחתון ידוע.

תזכורת: ניפוחים blow-up

ראינו בחישוביות כי $NFW \to DFW$ עולה 2^n כחסם עליון ותחתון. מסמנים $NFW \underset{2^n}{\overset{2^n}{\longrightarrow}} DFW$. החסם העליון היה subset construction, אם ננסה לעשות את זה לאוטומט NBW, נראה איפה אנחנו נופלים. איך מוכיחים שאי אפשר לעשות יותר טוב? מראים משפחה של שפות $L_1, L_2, \ldots \subseteq \Sigma^*$, כך של־$L_i$ יש NFW בגודל $O(i)$, אבל DFW המינימלי של L_i דורש לפחות 2^i מצבים.

צריך משפחה כי לכל פולינום p, קיים i כך ש־$2^i > p(i)$. בחישוביות הוכחנו לגבי המשפחה $L_i = (0+1)^* 0 (0+1)^i$. הא"ד מייצר שרשרת פשוטה, והדטרמיניסטי חייב לזכור את i האותיות האחרונות. היתה לנו הוכחה לחסם תחתון אקס' להשלמה, עבור המשפחה $L_n = \{ww \mid w \in \Sigma^n\}$. יש NFW בגודל $O(n^2)$ ל־$\overline{L_n}$ שינסה לנחש הפרה. לעומת זאת, NFW עבור L_n דורש לפחות $2^{O(n)}$ מצבים.

1.2 תנאי קבלה נוספים (חסר)

ר"ח אייר תשע"ג (שעור 4 ־ חסר שעור 3)

תזכורת: הגדרנו ריצה ו־$\inf(r)$. ראינו תנאי בוקי, רבין וסטריט. הראינו שלרבין וסטריט יש אינדקס מצבים וזוגות, ולאוטומט בוקי רק מצבים.

ראינו איך $Rabin(n,k) \to Buchi(n \cdot k)$ ו־$Streett(n,k) \to Buchi\left(n \cdot 2^k\right)$ עם חסם עליון ותחתון. ראינו $DBW < NBW$ ו־$DCW = \overline{DBW}$.

1.3 אוטומטים ולוגיקה מסדר שני

ב1962 התפתח The sequntial calculus: למשל האם קיימים x,y,z,n כך ש־$x^n + y^n = z^n$. ההבעה הלוגית היא

$$\exists x \exists y \exists z \exists n \text{ s.t. } x^n + y^n = z^n$$

לוגיקה פסוקית היא דלה בכך שיש לה קבוצה של אטומים $p,q,\ldots$ וכמתים $\neg, \wedge$ ואלגוריתם לבדיקה האם קיימת השמה ספיקה. כאן הדברים מסובכים יותר כי דומיין ההשמות הוא כל הטבעיים. לכן בעיית הספיקות לא בהכרח כריעה. איזו לוגיקה מוגדרת כך שבעיית הספיקות כריעה ואיזו לא. לבוקי היה שיא של בעיות לוגיות נרחבות שעדיין כריעות, וזה נעשה בעזרת אוטומטים.

בלוגיקה תמיד מדברים על המבנה. בהקשר של לוגיקה פסוקית יש לנו אטומים ואופרטורים בוליאנים, עם השמות T,F לאטומים.

כאן, המבנה $M = \langle \mathbb{N}, 0, +1, < \rangle$ מכיל איבר 0, פעולת העוקב, ויחס סדר. הנוסחאות הן למשל $\forall x \exists y\, y > x$. זה אומר שלכל איבר יש איבר גדול ממנו. או למשל

$$\forall x \forall y\ (x = y+1+1) \to \exists z\ (x < z \wedge z < y)$$

המתמטיקאים קיוו שאם יש פרוצדורה שמכריעה משפטים כאלו, הם יוכלו להוכיח משפטים נוספים. אם מצליחים להביע משפט בלוגיקה כריעה, אפשר להוכיח \ להפריך אותם.

נתייחס למילה w מעל הא"ב Σ במבנה $M_w = \left\langle \mathbb{N}, 0, +1, <, (Q_\sigma)_{\sigma \in \Sigma} \right\rangle$ כאשר $\mathbb{N} \supseteq Q_\sigma := \{ i \mid w[i] = \sigma \}$. למשל במילה $(ab)^\omega$ נקבל Q_b אוסף המספרים הזוגיים.

כל דבר שנפתור במבנה הזה ייפתר במבנה הכללי.

את ∞a נבטא ע"י $\forall x \exists y,\ y > x \wedge y \in Q_a$.

לוגיקה מסדר ראשון FOL_Σ היא עם התחביר הבא:

- $terms$ - ביטויים.

- הקבוע 0.
- משתנים $x, y, \ldots$.
- הביטוי $t+1$ עבור הביטוי t.

• נוסחאות - $formulas$.

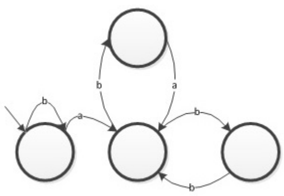
- $t = t', t < t', t \in Q_\sigma$ עבור ביטויים t, t' ופרדיקט Q_σ.
- $\neg\varphi_1, \varphi_1 \vee \varphi_2$ או $\exists x \left(\varphi_1 \left(x\right)\right)$ עבור נוסחאות φ_1, φ_2. (אפשר להרחיב גם לקשרים $\wedge, \rightarrow$, וכמת $\forall$). נסמן ב־$\varphi\left(x\right)$ את העובדה ש־x חופשי ב־φ, כלומר אינו בטווח של כמת.

איך נגדיר את השפה:

• אחרי כל a יש b? $\forall x,\, x \in Q_a \rightarrow \exists y,\, y > x \wedge y \in Q_b$.

• רק מספר סופי של aים? $\exists x \forall y \left(y > x \rightarrow \neg y \in Q_a\right)$.

• בין כל שני מופעים סמוכים של a יש מספר אי־זוגי של b: האם זו שפה רגולרית? כן, אפשר ליצור אוטומט. מסתבר שעם לוגיקה מסדר ראשון אי אפשר להביע



איור 9: האוטומט לשפה "בין כל שני מופעים סמוכים של a יש מספר אי זוגי של b

את השפה הזו. יש נוסחא בלוגיקה מסדר שני:

$$\forall x \forall y \left(x \in Q_a \wedge y \in Q_b \wedge x < y \wedge \forall z \left(z > x \wedge z < y \rightarrow \neg z \in Q_a\right)\right) \rightarrow$$
$$\rightarrow \exists X \left(x \in X \wedge \forall z \left(z \in X \leftrightarrow z + 1 \notin X\right) \wedge y \in X\right)$$

הקבוצה X היא כל האינדקסים שמסכימים עם x על הזוגיות.

בלוגיקה מסדר שני או לוגיקה מונדית מסדר שני, $Monadic\, SOL_\Sigma$ או $S1S_\Sigma$, הביטויים הם אותם ביטויים, אבל לנוסחאות נוסף

- $t \in X$ עבור ביטוי t ומשתנה קבוצות X (נוסחא אטומית).
- $\exists X\left(\varphi\left(X\right)\right)$ עבור נוסחא φ.

יש קשר יפה Descriptive complexity בין המחלקות עם הסיבוריות ללוגיקה איתה ניתן להביע אותם. למשל את הגרף 3 ־צביע ניתן לבטא ע"י

$$\begin{aligned}\exists Y\exists R\exists B,\forall v\forall u,\,E\left(v,u\right) & \rightarrow & \left(\left(v\in Y\rightarrow u\notin Y\right)\wedge\left(v\in R\rightarrow u\notin R\right),\ldots\right)\\ & \wedge & \forall v,\,v\in Y\vee v\in R\vee v\in B\end{aligned}$$

באופן כללי $NP=existentian\,SOL$ ו־$P=FOL+TC$. חשבו שדרך זה אפשר להוכיח את $P=_{?}NP$.

מה זה $monadic$? הפרדיקט $E\left(y,x\right)$ הוא בוליאני. בלוגיקה מונדית כל הפרדיקטים אונאריים.

נאמר ש־L גדירה ב־$S1S_{\Sigma}$ אם קיימת נוסחא φ ב־$S1S_{\Sigma}$ כך ש־$L=\left\{ w\mid M_{w}\models\varphi\right\}$.

משפט 1.6 (בוקי, 1962) L גדירה ב־$S1S_{\Sigma}$ אם"ם $L\in NBW$ (L ω־רגולרית).

מסקנה 1.7 $S1S_{\Sigma}$ כריעה. נבדוק האם לאוטומט יש מסלול העובר ב־α אינסוף פעמים.

הוכחה: $\Leftarrow$: יהי $\mathcal{A}=\left\langle \Sigma,Q,\delta,Q_{0},\alpha\right\rangle$ אוטומט NBW. נתאר נוסחא φ כך שלכל $w\in\Sigma^{\omega}$, $w\models\varphi$ אם"ם $w\in L\left(\mathcal{A}\right)$.

הנוסחא תאמר: "קיימת ריצה מקבלת על w". נשים לב כי ריצה של $\mathcal{A}$ על w משרה חלוקה של $\mathbb{N}$ ל־$|Q|$ קבוצות. למשל אם $Q=\left\{ 0,\ldots,n\right\}$, ריצה $r:\mathbb{N}\rightarrow Q$ משרה $Y_{0},\ldots,Y_{n}$ כך ש־$r\left(i\right)=j\Leftrightarrow i\in Y_{j}$.

$$\exists Y_{0}\ldots\exists Y_{n}$$

עם הפירוט הבא:

1. כל אינדקס $i\in\mathbb{N}$ נמצא במצב יחיד:

$$\forall i,\bigvee_{0\leq j\leq n}i\in Y_{j}\wedge\bigwedge_{0\leq j_{1}<j_{2}\leq n}\neg\exists i\,i\in Y_{j_{1}}\wedge i\in Y_{j_{2}}$$

נשים לב כי זו נוסחא סופית.

2. הריצה מתחילה במצב התחלתי:

$$\bigvee_{j\in Q_0} 0\in Y_j$$

3. הריצה עקבית עם δ. נסמן $\Delta\left(j_1,\sigma,j_2\right)$ אם $j_2\in\delta\left(j_1,\sigma\right)$.

$$\forall i,\quad \bigvee_{j_1,j_2,\sigma:\,\Delta(j_1,\sigma,j_2)} i\in Q_\sigma\wedge i\in Y_{j_1}\wedge i+1\in Y_{j_2}$$

(שוב, הקבוצה $\left\{\langle j_1,j_2,\sigma\rangle\mid j_1,j_2\in Q,\sigma\in\Sigma,\Delta\left(j_1,\sigma,j_2\right)\right\}$ סופית).

4. הריצה מקבלת:

$$\forall x\exists y\, y>x\wedge\bigvee_{j\in\alpha} y\in Y_j$$

זה הכיוון הקל, שפחות עניין את בוקי. הכיוון המעניין הוא להפוך נוסחא לאוטומט (שמאפשר להכריע את הנוסחא).

$\Leftarrow$ (אם L גדירה ב־$S1S_\Sigma$ אז L היא ω־רגולרית):

מאחר וכבר הוכחנו שאוטומטים סגורים לשלילה ולחיתוך, הטלה זה לא קשה, והחלק הבעייתי הוא הבסיס של הנוסחאות האטומיות. בוקי הציע תרגום לנוסחאות קנוניות. להוכחה שלושה צעדים:

1. מעבר מ־$S1S_\Sigma$ ל־$S1S$ (ללא הפרדיקטים Q_σ).

2. מעבר מ־$S1S$ לנוסחא ב־$S1S_{nf}$ (יש רק משתני קבוצות, ושני סוגים של נוסחאות אטומיות: $X_i\subseteq X_j$ ו־$succ\left(X_i,X_j\right)$ כלומר $X_i=\{x_i\},X_j=\{x_j\}$ וגם $x_j=x_i+1$).

3. תרגום מ־$S1S_{nf}$ אל NBW.

צעד ראשון:

נקודד את Σ ע"י משתנים $X_1,\ldots,X_n$ (משתני קבוצה בוליאניים) כאשר $n=\log|\Sigma|$. למשל עבור $\Sigma=\{a,b,c,d\}$, אזי X_1 מתאים ל־c,d ו־X_2 מתאים ל־b,d. את $dabab\,(ab)^\omega$ נקודד ע"י $X_1=\{0\}$ ו־$X_2=2\mathbb{N}$ (האי זוגיים). או למשל המילה המוגדרת ע"י $X_1=\{i:3\mid i\},X_2=\{i:2\mid i\}$, היא $(dabcba)^\omega$.

את כל המילים עם ∞d נבטא ע"י:

$$\forall x \exists y, y > x \wedge y \in X_1 \wedge y \in X_2$$

ואת $(a \vee b) \infty$ אפשר לבטא ע"י $y \notin X_1$.

הערה 1.8 אנחנו מאבדים את ההבדל בין ∞d ל־∞b בצורת הקידוד הזו ־ אין לנו בשום מקום קישור של הקידוד הספציפי עם האותיות, אבל נשים לב ש∞b גדירה אם"ם ∞d גדירה, וכנ"ל לגבי רגולריות, לכן ההוכחה היא לגבי L עד כדי פרמוטציה על Σ.

צעד שלישי:

יש לנו נוסחא ב־$S1S_{nf}$, עם משתנים $X_0, \ldots, X_n$ עם נוסחאות אטומיות $X_1 \subseteq X_2$, ו־$succ\left(X_i, X_2\right)$. האוטומט יהיה מעל הא"ב $\Sigma = \{0,1\}^{n+1}$. עבור $X_1 \subseteq X_2$ מורכב ממצב יחיד מקבל, עם קשת עצמית עבור כל $(x_0, \ldots, x_n) \in \{0,1\}^n$ חוץ מאלו בהם $x_1 = 1$ ו־$x_2 = 0$. (כשקוראים את האות $b_0 \ldots b_n$ במקום ה־i אנו יודעים כי $i \in X_j$ אם"ם $b_j = 1$).

את פרדיקט ה־$succ\left(X_i, X_j\right)$ נייצג ע"י אוטומט שקורא עד שהוא קורא את b עם $b_i = 1$, ולאחריה הוא קורא מילה עם $b_j = 1$, ומכאן ואילך הוא קורא רק מילים עם $b_i = b_j = 0$.

ח' אייר תשע"ג (שעור 5)

במעבר מ־SIS לNBW, יש לנו נוסחאות מעל המספרים הטבעיים.

כל מילה מעל $(0+1)^\omega$ מגדירה קבוצות $X_1, X_2, X_3 \subseteq \mathbb{N}$ כאשר נסתכל עליה כעל $\left((0+1)^3\right)^\omega$. באופן כללי כדי להגדיר t קבוצות מעל הטבעיים, נמדל בעזרת מילה $\left((0+1)^t\right)^\omega$.

נגדיר צורה נורמלית לנוסחאות $S1S$:

- נוסחאות אטומיות: $X_1 \subseteq X_2, Succ\left(X_1, X_2\right)$ עבור משתני קבוצות X_1, X_2.
(תזכורת: $Succ\left(X_1, X_2\right) = \mathbb{T}$ אם הם סינגלטונים של מספרים עוקבים).
- $\neg\varphi_1$, $\varphi_1 \vee \varphi_2$ ו־$\exists X\left(\varphi_1\right)$.

לא נדבר על איך ההמרה עובדת, זה שייך לקורס בלוגיקה, תהיה הפניה למאמר באתר.

דוגמא לנוסחא:

$$\exists X_1 \exists X_2 \exists X_2, Succ\left(X_1, X_2\right) \wedge Succ\left(X_2, X_3\right) \wedge Succ\left(X_2, X_3\right)$$

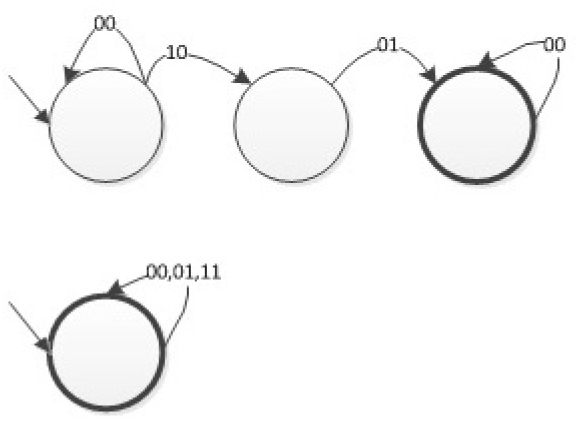
זוהי נוסחא שאינה ספיקה. המטרה שלנו היא לעשות תרגום מהצורה הנורמלית לאוטומטים.

היה לנו גם דרך לסלק את הפרדיקטים של התווים ב־Σ. אפשר לעשות זאת ע"י קבוצה מאפיינית לכל $\sigma \in \Sigma$. ראינו דרך לעשות ע"י $\log|\Sigma|$ קבוצות מאפיינות, זה קצת בלבל, אבל גם זה אפשרי.

משפט 1.9 בהינתן נוסחא φ מעל $X_1, \ldots, X_t$, ניתן לבנות NBW שמקבל מילה ב־$\left((0+1)^t\right)^\omega$ אם"ם היא מספקת את φ.

הוכחה: בסיס: עבור $\varphi = X_1 \subseteq X_2$, האוטומט מעל $\left((0+1)^2\right)^\omega$ לא מוכן לקבל אותיות "10" כי אם קיימות אותיות כאלו, יש איבר שנכנס ל־X_2 אבל לא ל־X_1. אם יש לנו $X_i \subseteq X_j$ מעל t קבוצות, האוטומט יקרא תווים ב־$\{0,1\}^t$, ולא יקבל תווים בהם $\sigma_i = 1, \sigma_j = 0$.

עבור $Succ\left(X_1, X_2\right)$, אם היינו רוצים לומר ש־$X_1$ הוא סינגלטון, היינו צריכים לקרוא כל הזמן 0, מתישהו לקרוא 1, ומאז לקרוא רק 0. אם לא קראנו 1, או קראנו יותר מפעם אחת, אנחנו דוחים. אם מדובר על משתנה X_i מתוך t משתנים, צריך להסתכל על כל התווים האפשריים למשתנים האחרים. כשאנחנו מוסיפים את ה־$Succ$, יש לנו דרישה לקרוא בהתחלה "00", מתישהו נקרא 10 (קוראים X_1), מיד אחריו 01 (קוראים X_2), ולאחר מכן קוראים רק 00. שוב, אם יש יותר משני משתנים, כל חץ מסמן את כל 2^{t-2} האפשרויות למשתנים האחרים.



איור 10: האוטומטים עבור $X_1 \subset X_2, Succ\left(X_1, X_2\right)$

עבור $\neg\varphi$, זה נובע מסגור להשלמה, $\varphi_1 \vee \varphi_2$ נובע מסגור לאיחוד, $\exists X\varphi$ נובע מסגור להטלה שנראה:

תהי L שפה מעל א"ב $\Sigma_1 \times \Sigma_2$. נניח $\Sigma_1 = \{0,1\}$ ו-$\Sigma_2 = \{a,b\}$ המוגדרת ע"י אוטומט $\mathcal{A}$. עבור מילה $w_1 \in \Sigma_1^\omega$ ו-$w_2 \in \Sigma_2^\omega$ נגדיר את הרכבת המילים $w_1 \oplus w_2 \in (\Sigma_1 \times \Sigma_2)^\omega$. פעולת ההטלה מוגדרת באופן הבא:

$$L\mid_{\Sigma_1} = \{w_1 \mid \exists w_2 \in \Sigma_2^\omega, w_1 \oplus w_2 \in L\}$$

ברור כי אוטומט סגור להטלה, כי נסיר מעל הדרישות של החיצים את החלק ששייך לתווים של Σ_2 (האוטומט יהיה א"ד). בהינתן $\mathcal{A}$ מעל $\Sigma_1 \times \Sigma_2$ נקבל את $\mathcal{A}\mid_{\Sigma_1}$ מעל Σ_1 ע"י פונקציית המעברים

$$\delta'(q,\sigma_1) = \bigcup_{\sigma_2 \in \Sigma_2} \delta(q,(\sigma_1,\sigma_2))$$

לעניינו, עבור נוסחא $\exists X_1 \varphi(X_1,X_2)$, ויש לנו אוטומט מעל $\left((0+1)^2\right)^\omega$ המתאים ל-$\varphi(X_1,X_2)$, נטיל אותו על $(0+1)^\omega$. ■

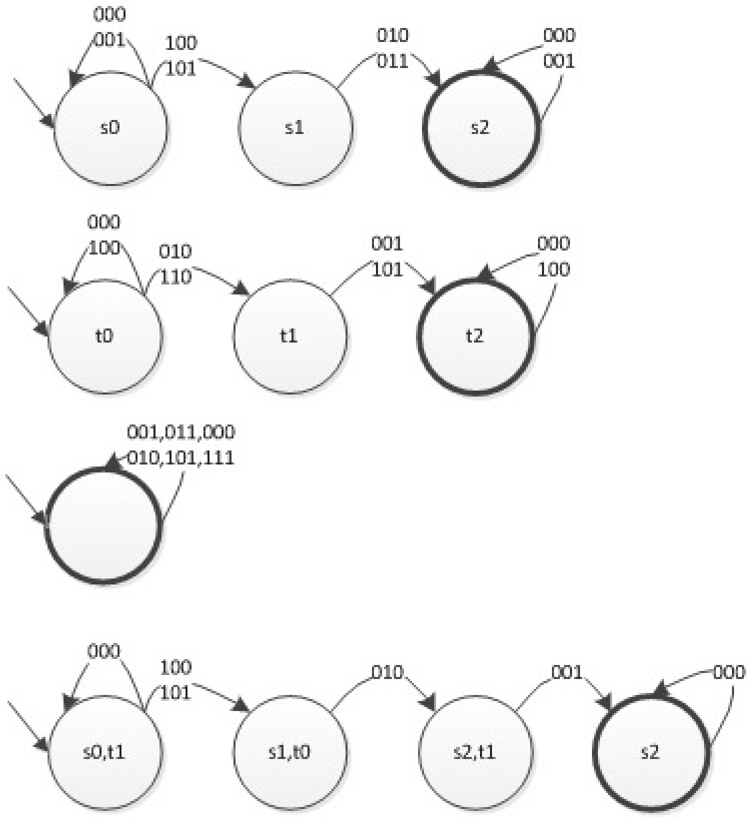
■

דוגמא: $\underbrace{X_1 \subseteq X_3}_{\theta_3} \wedge \underbrace{\mathrm{Succ}(X_2,X_3)}_{\theta_2} \wedge \underbrace{\mathrm{Succ}(X_1,X_2)}_{\theta_1}$, הא"ב הוא $\left((0+1)^3\right)^\omega$. בחיתוך של $\theta_1 \wedge \theta_2$ עם θ_3, נקבל חיתוך ריק במעבר מ-$\langle s_0,t_0\rangle$ ל-$\langle s_1,t_0\rangle$ כי θ_3 לא מקבל את $.100$

2 בעיות הכרעה

אנחנו מכירים את $PTIME, NP, EXPTIME, NLOGSPANCE, PSPACE$- אלו בעיות הכרעה של מ"ט שעוצרות בזמן סופי. יש מחלקות שמעניין אותנו הזמן, ויש כאלו שמעניין אותנו השטח. בעיית הישיגות: אם יש לנו גרף מכוון ואנחנו מעוניינים לדעת האם יש מסלול בין s,t זו בעיה $NLOGSPACE - complete$. על סרט העבודה שומרים את הקודקוד הנוכחי ומספר הצעדים (דורש $O(\log n)$ מקום), ועל הפלט תרשום את הקודקוד הראשון, וכל פעם תנחש קודקוד נוסף באופן א"ד.

אנחנו יודעים כי $TIME(f(n)) \subseteq SPACE(f(n))$. בכיוון ההפוך $SPACE(f(n)) \subseteq TIME\left(2^{f(n)}\right)$ (אפשר לדמות את מכונת המקום, ולמנות את הקונפיגורציות השונות).



איור 11: אוטומטים עבור $Succ\left(X_1, X_2\right) \wedge Succ\left(X_2, X_3\right) \wedge X_1 \subset X_3$

בשורה הראשונה אוטומט עבור θ_1. בשורה השניה עבור θ_2 ובשורה השלישית עבור θ_3. בשורה הרביעית מופיע אוטומט המכפלה עבור $\theta_1 \wedge \theta_2$, וקל לראות כי אוטומט המכפלה שלו עם האוטומט ל־θ_3 הוא ריק.

משפט סאביץ' - $NSPACE\left(s\left(n\right)\right) \subseteq SPACE\left(s^2\left(n\right)\right)$ כלומר $NPSPACE = PSPACE$.

הבעיות שיעניינו אותנו: בהינתן NBW $\mathcal{A}$,

- האם $L(\mathcal{A}) \neq \emptyset$? (בעיית הריקנות).
- האם $L\left(\mathcal{A}\right) = \Sigma^{\omega}$ (בעיית האוניברסליות)?
- בהינתן $\mathcal{A}_1, \mathcal{A}_2$, האם $L\left(\mathcal{A}_1\right) \subseteq L\left(\mathcal{A}_2\right)$ (בעיית ההכלה)?

בעזרת בעיית ההכלה ניתן לפתור את בעיית הריקנות ובעיית האוניברסליות.

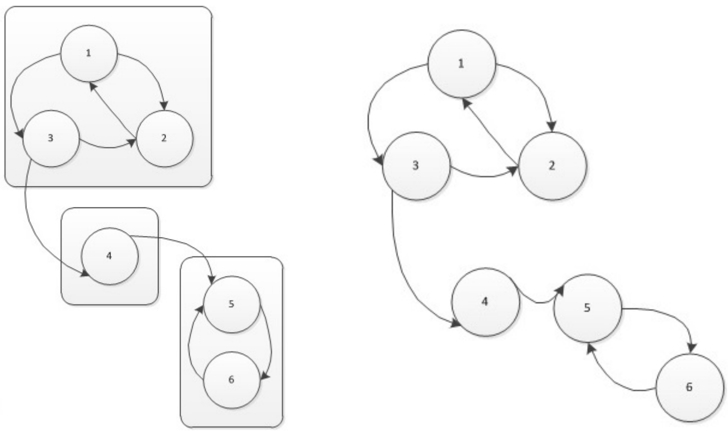
2.1 בעיית הריקנות

נתחיל מבעיית הריקנות: ניתנת לפתרון ב־$NLOGSPACE$, ננחש מצב $q \in \alpha$ כך ש־q ישיג מ־Q_0 ע"י u ומעצמו ע"י v (ישיגות בגרף).

נכונות: אם מצאנו מבנה כזה, אזי $uv^\omega \in L(\mathcal{A})$. בכיוון השני, אם $w \in L(\mathcal{A})$, יש מצב $q \in \alpha$ שהריצה על w עוברת בו ∞ פעמים, ולכן הוא ישיג מ־Q_0 ומעצמו.

כמו כן זה ניתן לפתרון בזמן לינארי: יהי $G = \langle V, E \rangle$, נאמר ש־$C \subseteq V$ הוא רכיב קשיר היטב (Strongly Connected Component) אם לכל $v_1, v_2 \in C$ יש מסלול מ־v_1 ל־v_2 $(E^*(v_1, v_2))$. C הינו רכיב קשיר היטב מקסימלי (Maximal SCC) אם לכל $\emptyset \neq S \subseteq V \backslash C$, $C \cup S$ אינו SCC.

בגרף לדוגמא $\{4\}, \{5,6\}, \{1,2,3\}, \{1,2\}, \{2\} : SCC$. יש לנו חלוקה ל־$\{1,2,3\}, \{4\}, \{5,6\}$ של $MSCC$, וזה משרה DAG של גרף בלי רכיבים קשירים.



איור 12: גרף לדוגמא

$Tarjan$ 75': יש חלוקה יחידה ל־$MSCC$ וניתן למצוא אותה בזמן לינארי.

כעת נפתור את בעיית הריקנות של NBW: נחפש S $MSCC$ שהוא לא טרוויאלי (יש בו לפחות קשת אחת) כך ש־$S \cap \alpha \neq \emptyset$, והוא ישיג מ־$Q_0$.

מה לגבי תנאי קבלה נוספים?

עבור NGW - יש כמה קבוצות $\alpha = \{F_1, \ldots, F_k\}$, ואנו רוצים $\inf(r) \cap F_i \neq \emptyset$ לכל $i \in [k]$. הסיבוכיות עדיין לינארית: מחפשים מסלול $Q_0 \to F_1 \to F_2 \to \ldots \to F_k \to F_1$. זה עדיין לינארי כי ה־$MSCC$ הלא טרוויאלי S צריך לקיים $S \cap \{F_i\} \neq \emptyset$ לכל $i \in [k]$.

עבור NCW ־ דורשים $\inf(r) \cap \alpha = \emptyset$. זה אותה סיבוכיות כמו $NCW \to NBW$: בודקים האם קיים רכיב קשירות שאינו מכיל איברים ב־α (נפעיל $MSCC$ על $\langle Q \backslash \alpha, E_\delta \rangle$), ונחפש כזה שישיג (בגרף המקורי) מ־Q_0.

ט"ו אייר תשע"ג (שעור 6)

טעות שכיחה בתרגילים, בשאלה האחרונה על תנאי $prompt$: עבור $\alpha \subseteq Q$, r מקבלת אם קיים k כך שמבקרים ב־α בצפיפות k. הבלבול ־ הk אינו פרמטר של האוטומט. חלק הלכו בכיוון של "קיים k כך שr מקבלת אם מבקרים בα בצפיפות k. זה שינוי סדר הכמתים.

איפה זה בא לידי ביטוי: היינו צריכים להוכיח שאין אוטומט עבור ∞a, בגישה השניה, הניחו שיש אוטומט עבור ∞a, עם פרמטר k, לקחו $\left(b^{k+1}a\right)^\infty$, ומהריצה שלה אפשר לפמפם את b^k. לפי תשובה 1, היה צריך להסתכל על $ababbabbba\ldots$, כלומר לכל k, יש רצף של $b^{|\alpha|k+1}$.

לפרטים נוספים ־ מאמר prompt Buchi automata באתר של המרצה.

בעיות הכרעה: דיברנו על ריקנות. ראינו אלגוריתם פשוט עבור בוקי (מחפשים מסלול מעגלי לקודקוד ב־α שישיג מQ_0). זה נמצא ב־$NLOGSPACE$ ולינארי. לNGB אנחנו רוצים מעגל דומה, רק דרך כל F_i. ראינו כי $NRW_{(n,k)} \to NBW_{(n \cdot k)}$, לכן בעזרת הרדוקציה אפשר לפתור בזמן פולינומי. כמו כן ב־$NLOGSPACE$ כי אנו מנחשים מסלול שזר ל־B_i וחותך את G_i. סטריט יותר מסובך, כי הרדוקציה היא ל־$NBW_{\left(n \cdot 2^k\right)}$ והיא אקספוננציאלית.

נשים לב כי הבעיה היא $NLOGSPACE-hard$: יש רדוקציה מישיגות אליה ־ בהינתן גרף s,t, נוסיף ל־t חץ שיוצא לקודקוד חדש שיהווה בור מקבל, נתייג את הקשתות, ונסמן את $Q_0 = \{s\}$, ויש לנו אוטומט שבעיית הריקנות שלו זהה לבעיית הישיגות.

2.2 בעיית האוניברסליות

אוניברסליות: $L(\mathcal{A}) = \Sigma^\omega$? הצעה ראשונה: בהינתן $\mathcal{A}$ NBW, נבדוק ריקנות של $\overline{\mathcal{A}}$. הבעיה היא ש־$\left|\overline{\mathcal{A}}\right| = 2|\mathcal{A}|$ אם $\mathcal{A}$ הוא DBW ואז האלגוריתם הזה הוא פולינומיאלי. אבל אם $\mathcal{A}$ הוא NBW, $\left|\overline{\mathcal{A}}\right| = 2^{O(n \log n)}$, וזה נותן לנו אלגוריתם $EXPTIME$ לבנות את $\overline{\mathcal{A}}$ ולבדוק ריקנות. אפשר גם ב־$PSPACE$ אם נבצע זאת on the fly. מנחשים קודקוד, וממשיכים ממנו הלאה. ניתן להראות כי זו בעיית $PSPACE-hard$, ע"י רדוקציה מבעיית ALL_{NFA} (בעיית האוניברסליות של NFA)

- תרגיל. יש גם רדוקציה גנרית הבאה:

בהינתן מ"ט דטרמיניסטית $T = \langle \Gamma, Q, \delta, q_0, q_{acc}, q_{rej} \rangle$ שפועלת בשטח פולינומיאלי, וקלט $x \in \Gamma^*$, נבנה NBW $\mathcal{A}_{T,x}$ כך ש־$\mathcal{A}_{T,x}$ אוניברסלי אס"ם T דוחה את x. נזכור כי

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

כעת נרצה לבנות אוטומט. הרעיון: $\mathcal{A}_{T,x}$ מקבלת מילה w אס"ם w אינה קידוד של ריצה חוקית של t על x או w קידוד של ריצה דוחה של T על x.

אכן, $\mathcal{A}_{Tx}$ אוניברסלית אס"ם T דוחה את x. כי אם $\mathcal{A}_{Tx}$ אוניברסלית, כל מילה היא אינה קידוד חוקי של ריצה של T על x, או שהיא ריצה דוחה. לכן אין ריצה מקבלת של T על x. בכיוון ההפוך אם T דוחה את x, אין ריצה מקבלת של T על x ולכן $\mathcal{A}_{Tx}$ תקבל כל מילה.

מימוש: $\Sigma = \Gamma \cup (\Gamma \times Q) \cup \{\#\}$. קונפיגורציה של T תקודד ע"י $\# \gamma_1 \gamma_2 \ldots (\gamma_i, q) \ldots \gamma_{s(n)}$.
אם $x = x_1 x_2 \ldots x_n$, הקונפיגורציה ההתחלתית תיראה $\# (x_1, q_0)\, x_2 x_3 \ldots x_n \_\ \_\ \ldots\ \_$.
אם $\delta(q_0, x_1) = (q_1, c, R)$, הקונפיגורציה הבאה תקודד ע"י $\# c\,(x_2, q_1)\, x_3 \ldots$. אם שתי קונפיגורציות עוקבות $\# \gamma_1 \gamma_2 \ldots (\gamma_i, q_j)\, \gamma_{i+1} \ldots \gamma_{s(n)} \# \gamma_1 \gamma_2 \ldots (\gamma_{i-1}, q_x)\, c, \gamma_{i+1} \ldots$ הן נבדלות בחלון בגודל 3 סביב הראש הקורא. ניתן להגדיר פונקציה

$$\text{next} : \Sigma^3 \to \Sigma$$

כך שאם בקונפיגורציה הנוכחית יש חלון בגודל 3 שמקודד ע"י $\sigma_{i-1}, \sigma_i, \sigma_{i+1}$, בקונפיגורציה העוקבת, האות האמצעית בחלון המתאים היא $\text{next}(\sigma_{i-1}, \sigma_i, \sigma_{i+1})$.

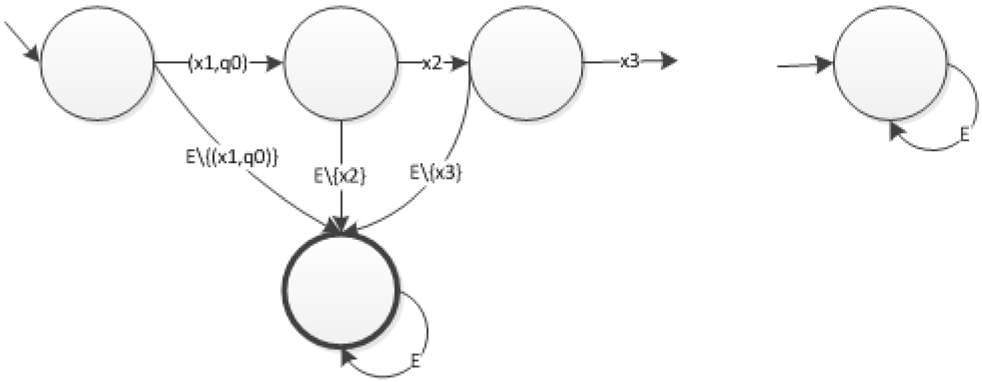
הגדרת next:

- עבור $\gamma_1, \gamma_2, \gamma_3 \in \Gamma$, $\text{next}(\gamma_1, \gamma_2, \gamma_3) = \gamma_2$.
- עבור $\gamma_1, \gamma_2, \gamma_3 \in \Gamma$ ו־$q \in Q$,

$$\begin{aligned}
\text{next}(\gamma_1, \gamma_2, (q, \gamma_3)) &= \begin{cases} (\gamma_2, q') & \delta(q, \gamma_3) = (q', \gamma', L) \\ \gamma_2 & \delta(q, \gamma_3) = (q', \gamma', R) \end{cases} \\
\text{next}(\gamma_1, (\gamma_2, q), \gamma_3) &= \gamma' \text{ s.t. } \delta(\gamma_2, q) = (q', \gamma', \Delta')
\end{aligned}$$

באופן דומה ניתן להבדיל בין # וכד'.

האוטומט $\mathcal{A}_{Tx}$ יפעל כך - יקבל מילה אם:



איור 13: הרכיב המקבל אם הקונפיגורציה ההתחלתית אינה נכונה

1. יש הפרה של הקונפיגורציה ההתחלתית (מילים שלא מתחילות ב־(x_1, q_0) הולכות למצב מקבל, או אם האות הבאה היא לא x_2 וכן הלאה), הרכיב שמזהה את זה הוא בגודל $s(n) + 2$.

2. יש הפרה של next. הרכיב מתחיל ומחכה להפרה. הפרה היא קריאה של $\sigma_1, \sigma_2, \sigma_3$, התקדמות עוד $s(n) - 1$ צעדים, ואז קריאה של $\sigma \in \Sigma \setminus \{\text{next}(\sigma_1, \sigma_2, \sigma_3)\}$. זה רכיב בגודל $O(s(n))$, ויש צורך ב־Σ^3 רכיבים כאלו.

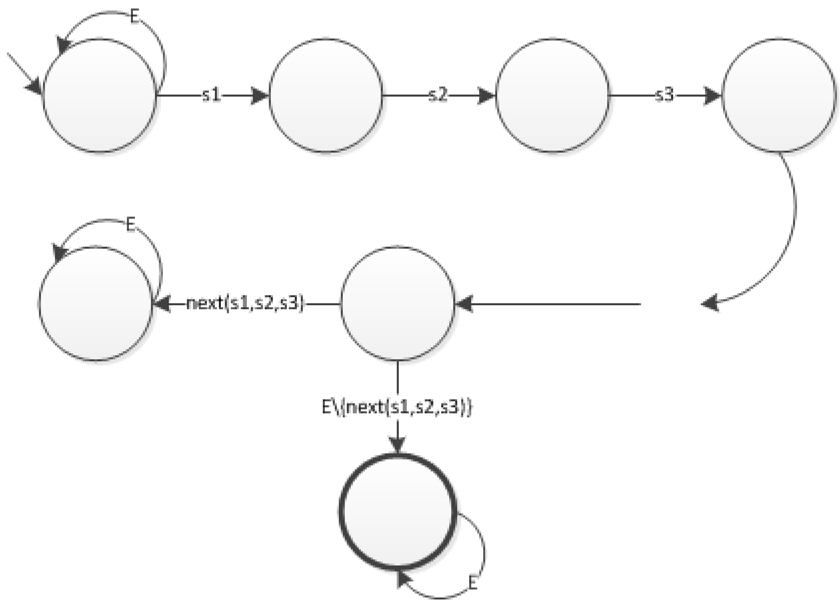
3. יש בה קונפיגורציה דוחה ־ נקבל אם קראנו תו $\sigma \in \Gamma \times \{q_{rej}\}$.

הריצה היחידה שלא תתקבל באיחוד של כל הרכיבים היא הקידוד של הריצה המקבלת של $T(x)$. נשים לב כי $\overline{L(\mathcal{A}_{Tx})} = \{w \mid w \text{encode infinite accept run of } T(x)\}$ כאשר ריצה מקבלת נכנסת ללולאה של קונפיגורציות זהות המתאימות ל־next.

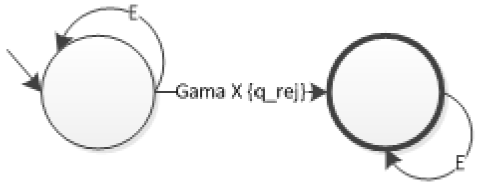
2.3 בעיית הכלת השפות

אנו רוצים לדעת האם $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$. זו בעיה קשה לפחות כמו אוניברסליות וריקנות.

אלגוריתם: נבנה NBW $\mathcal{A}'$ עבור $L(\mathcal{A}_1) \cap \overline{L(\mathcal{A}_2)}$, ואז $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2) \Leftrightarrow L(\mathcal{A}') = \emptyset$. נשים לב כי $|\mathcal{A}'| = |\mathcal{A}_1 \times \exp(\mathcal{A}_2)|$. ניתן לבנות את $\mathcal{A}'$ on the fly ולכן הבעיה היא ב־$PSPACE$.



איור 14: הרכיב המקבל הפרה של next



איור 15: הרכיב המקבל קונפיגורציה דוחה

עבור $\Sigma=\{a\}$, עבור NBW יתקיים $L(\mathcal{A})=\emptyset$ או $L(\mathcal{A})=\{a^\omega\}=\Sigma^\omega$. לעומת זאת עבור NFW, $L(\mathcal{A})$ מגדיר תת־קבוצה של הטבעיים

$$\{k \mid k\in L(\mathcal{A})\}$$

זה נובע מכך שבעיית החברות (מקבלת $w,\mathcal{A}$ ומכריעה האם $w\in L(\mathcal{A})$). עבור $w=vu^\omega$ איך בודקים האם $w\in L(\mathcal{A})$? אפשר להגדיר $L(\mathcal{A}_w)$ שמקבלת רק את w, ולקבוע כי $w\in L(\mathcal{A})\Leftrightarrow L(\mathcal{A}_w)\cap L(\mathcal{A})\neq\emptyset$.

כ"ב אייר תשע"ג (שעור 7)

שבוע הבא יש כנס על שפות ממושקלות. אם עד כה דברנו על $L:\Sigma^*\to\{0,1\}$, אפשר לדבר על $L:\Sigma^*\to\mathbb{R}$, ועל אוטומט עם עלות של ריצה. יש גם הגדרה של עלות ריצה אינסופית (ממוצע). אפשר אוטומט א"ד שלא ידוע אם יש לו אוטומט דטר' שקול. זה לא כריע (אם אפשר לחרצן). נרצה לפרמל איכות - חישוב הוא טוב אם הוא מגיע בסוף ל־1 ולא ל־0.

3 אוטומטים מעל עצים

3.1 הגדרות - עצים, אוטומטים וריצה

נדבר על עץ מושרש, כאשר הקודקודים הם איברים ב־Σ.

נרצה למשל להגדיר את העצים שבמסלול השמאלי שלהם יש ∞a.

הגדרה 3.1 עבור קבוצה סופית D של כיוונים, נאמר ש־$T\subseteq D^*$ סגורה לרישות אם $x\cdot d\in T$ עבור $x\in D^*$ ו־$d\in D$ אז $x\in T$. אם $x\cdot d$ במבנה העץ T ו־d הוא ההתפצלות האחרונה, גם הרישא $x\in T$.

נשים לב כי אם $|D|=1$ אז העץ הוא שרוך. $T=D^*$ הוא העץ המלא.

כעת נרצה לסמן את הקודקודים באותיות של הא"ב Σ:

הגדרה 3.2 $\Sigma-labeled\,D\,tree$ הוא זוג $\langle T,\tau\rangle$ כך ש־$T\subseteq D^*$ הוא $D-tree$, ו־$\tau:T\to\Sigma$ ממפה כל קודקוד לאות בא"ב Σ.

השורש של T: הצומת ε. מסלול ב־T הוא $\pi\subseteq T$ כך ש־$\varepsilon\in\pi$, לכל $x\in\pi$, x עלה (לא קיים $d\in D$ כך ש־$xd\in T$) או קיים $d\in D$ יחיד כך ש־$x\cdot d\in\pi$, כמו כן

π סגור לרישות. בעץ המלא מפתה לזהות מסלול π עם קבוצת הכיוונים ב־D^{ω} ־ יש התאמה חח"ע ביניהם, אבל מסלול זו קבוצה של קודקודים.

אם יש לי $\{a,b\}-labeled\ \{0,1\}-tree$, יש לי את העץ עם המסלולים, ו־τ שמתייגת את הצמתים, אוטומט בוקי מעל עצים (NBT) יהיה

$$\mathcal{A} = \langle D, \Sigma, Q, Q_0, \delta, \alpha \rangle$$

נניח ש־$|D| = 2$. אזי $\delta : Q \times \Sigma \to 2^{Q^2}$. למשל $\delta(q, \alpha) = \{\langle q_1, q_2 \rangle, \langle q_3, q_4 \rangle\}$. ריצה של $\mathcal{A}$ על עץ $\langle T, \tau \rangle$ היא $\langle T, r \rangle$ שהיא $Q-labeled\, D-tree$ ־ מסמנים כל קודקוד במצב מ־Q ($r : T \to Q$). כך ש־

1. הריצה מתחילה במצב התחלתי: $r(\varepsilon) \in Q_0$.
2. הריצה מכבדת את פונקציית המעברים: לכל $x \in T$,

$$\langle r(x \cdot 0), r(x \cdot 1) \rangle \in \delta(r(x), \tau(x))$$

(כאשר $D = \{0,1\}$).

עבור ריצה $\langle T, r \rangle$ ומסלול $\pi \subseteq T$, נגדיר $\inf(r \mid \pi)$ כקבוצת המצבים ש־r מבקרת בהם ∞ פעמים על המסלול π.

$$\inf(r \mid \pi) := \{q \in Q \mid |\{x \in \pi \mid r(x) = q\}| = \infty\}$$

כעת נאמר ש־$\langle T, r \rangle$ מקבלת אם $\inf(r \mid \pi) \cap \alpha \neq \emptyset$ לכל מסלול $\pi \subseteq T$.

לדוגמא: $D = \{0,1\}$ (ככה תמיד עד שנאמר אחרת), ו־$\Sigma = \{a, c\}$. אנחנו רוצים NBT עבור

$$L = \{\langle T, \tau \rangle \mid (*)\}$$

כאשר $(*)$: יש ב־T מסלול שמופיע בו הרצף cc.

אחד החסרונות של אוטומטים מעל עצים זה שהם קשים לציור. נתאר את פונקצית המעברים: אם נאמר $\delta(q_0, a) = \{\langle q_0, q_0 \rangle\}$ זה אומר שממשיכים לחפש את cc גם בתת־העץ השמאלי וגם בתת־העץ הימני. זה לא מה שאנחנו רוצים. נבצע ניחוש לא־דטרמיניסטי:

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

כאשר q_{acc} הוא "בור מקבל", כלומר $\delta\left(q_{acc},\sigma\right)=\left\{\left\langle q_{acc},q_{acc}\right\rangle\right\}$ לכל $\sigma\in\Sigma$, ו־ $q_{acc}\in\alpha$.

$$\begin{aligned}\delta\left(q_0,c\right)&=\left\{\left\langle q_1,q_{acc}\right\rangle,\left\langle q_{acc},q_1\right\rangle\right\}\\ \delta\left(q_1,c\right)&=\left\{\left\langle q_{acc},q_{acc}\right\rangle\right\}\\ \delta\left(q_1,a\right)&=\delta\left(q_0,a\right)\end{aligned}$$

נדגים ריצה של האוטומט לכל מיני עצים:

יש קשר בין אוטומט מעל עצים ותורת המשחקים. זה משחק בין האוטומט שאומר לאן הוא מתקדם, ושחקן נגדי שמקשה עליו איך הוא ממשיך מכאן ואילך.

דוגמא: NBT עבור ־ קיים מסלול עם cc ובכל המסלולים יש ∞a.

באופן כללי, אנחנו יודעים איך יראה אוטומט מעל מילים כאלו. צריך לפתוח את זה לעץ ־ יש פתיחות אוניברסליות ־ לכל המסלולים, ופתיחות של "קיים מסלול". נשים לב כי את הבדיקה של ∞a אנחנו יכולים לדחות לשלב מאוחר יותר.

נגדיר את המצבים:

- q_0 ־ בודק הכל.
- q_1 ־ בודק הכל, ראינו c ראשון.
- q_2 ־ בודק ∞a בכל מסלול, קראנו a.
- q_3 ־ בודק ∞a בכל מסלול, קראנו c.

כעת קל לכתוב את פונקצית המעברים:

$$\begin{aligned}\delta\left(q_0,a\right)&=\left\{\left\langle q_0,q_2\right\rangle,\left\langle q_2,q_0\right\rangle\right\}\\ \delta\left(q_0,c\right)&=\left\{\left\langle q_0,q_3\right\rangle,\left\langle q_1,q_3\right\rangle\right\}\\ \delta\left(q_1,a\right)&=\delta\left(q_0,a\right)\\ \delta\left(q_1,c\right)&=\left\{\left\langle q_3,q_3\right\rangle\right\}\\ \delta\left(q_2,a\right)&=\delta\left(q_3,a\right)=\left\{\left\langle q_2,q_2\right\rangle\right\}\\ \delta\left(q_2,c\right)&=\delta\left(q_2,c\right)=\left\{\left\langle q_3,q_3\right\rangle\right\}\end{aligned}$$

ו־$\alpha=\{q_2\}$.

הגדרה 3.3 נאמר ש־$\langle T,\tau\rangle$ הוא עץ Σ, D רגולרי, אם לכל $\sigma\in\Sigma$, השפה $\tau^{-1}(\sigma)\subseteq D^*$ היא רגולרית.

עץ רגולרי ניתן לתאור ע"י אוטומט $\Sigma - transducer \mid D$ (משרן). אם $\tau : D^* \to \Sigma$, באוטומט דטרמיניסטי כל מילה ב־D^* מגיע למצב כלשהו באוטומט, וניתן להשרות ממנה את τ.

3.2 תכונות סגור של אוטומטים מעל עצים

3.2.1 סגור לחיתוך

עבור $\mathcal{A}_1,\mathcal{A}_2\in NBT$, האם קיים $\mathcal{A}$ כך ש־$L(\mathcal{A}) = L(\mathcal{A}_1)\cap L(\mathcal{A}_2)$? ב־$NBW$ היה לנו אוטומט מכפלה עם שני עותקים. נגדיר גם כאן:

$$\mathcal{A} = \left\langle \Sigma, Q_1\times Q_2\times\{0,1\}, \delta, Q_1^0\times Q_2^0\times\{1\}, \alpha\right\rangle$$

כאשר $\delta : (Q_1\times Q_2\times\{1,2\})\times\Sigma\to 2^{(Q_1\times Q_2\times\{1,2\})^2}$

$$\begin{aligned}
\delta(\langle q_1,q_2,i\rangle,\sigma) &= \{\langle\langle s_1,s_2,i_1\rangle,\langle t_1,t_2,i_2\rangle\rangle \mid (1),(2),(3)\}\\
(1) &: \langle s_1,t_1\rangle\in\delta_1(q_1,\sigma)\\
(2) &: \langle s_2,t_2\rangle\in\delta_2(q_2,\sigma)\\
(3) &: i_k = \begin{cases} i_k & q_k\notin\alpha_k\\ 3-i_k & q_k\in\alpha_k\end{cases}
\end{aligned}$$

ו־$\alpha = Q_1\times\alpha_2\times\{1\}$.

האוטומט המושרה על DBT הוא דטרמיניסטי גם הוא.

3.2.2 סגור לאיחוד

איחוד של NBT הוא פשוט. בהנחה ש־Q_1, Q_2 זרים, אפשר להניח אותם זה ליד זה (כמו באיחוד א"ד של מילים). הבעיה היא בדטרמיניזם.

במודל הדטרמיניסטי - $Q_0 = \{q_0\}$, ו־$\delta : Q\times\Sigma\to Q^D$. כמו אוטומט מעל מילים, זה משרה ריצה יחידה. עם אוטומט המכפלה (בלי שני עותקים)

$\alpha = (Q_1\times\alpha_2)\cup(\alpha_1\times Q_2)$ מה יקרה כאן? מסתבר שזה לא סגור לאיחוד.

טענה 3.4 אין DBT עבור "יש a בצומת 0 או בצומת 1"$=L$. האוטומט הא"ד יקיים

$$\begin{aligned}\delta\left(q_0,\sigma\right) &= \left\{\left\langle q_a,q_{acc}\right\rangle,\left\langle q_{acc},q_a\right\rangle\right\}\\ \delta\left(q_a,a\right) &= \left\{\left\langle q_{acc},q_{acc}\right\rangle\right\}\\ \delta\left(q_a,c\right) &= \emptyset\, or\ \left\{\left\langle q_{rej},q_{rej}\right\rangle\right\}\end{aligned}$$

נשים לב כי השפה היא איחוד של שתי שפות ב־DBT.

הוכחה: נניח בשלילה כי $\mathcal{A}$ הוא DBT עבור L. עבור העץ עם $\tau_1\left(x\right)=\begin{cases}a & x=0\\ c & o.w.\end{cases}$.
נניח כי $\delta\left(q_0,c\right)=\left\langle q_1,q_2\right\rangle$. $\left\langle T,\tau_1\right\rangle\in L$, ולכן $\mathcal{A}$ עם מצב התחלתי q_2 מקבל את העץ שכולו c. בעץ τ_2 עם a בקודקוד 1. הריצה המקבלת עליו חייבת לעבור עם $r\left(0\right)=q_1,r\left(1\right)=q_2$. לכן גם $\mathcal{A}$ עם מצב התחלתי q_1 מקבל את העץ הקבוע c. לכן האוטומט $\mathcal{A}$ יקבל גם את $\tau_3=c$. כי משני הכיוונים הריצה הנמשכת מקבלת. אבל $\left\langle T,\tau_3\right\rangle\notin L$. ■

כ"ט אייר תשע"ג (שעור 8)

תזכורת: ראינו עץ ל"קיים b איפה שהוא בעץ". $\delta\left(q_0,a\right)=\left\{\left\langle q_0,q_{acc}\right\rangle,\left\langle q_{acc},q_0\right\rangle\right\}$ ו־$\delta\left(q_{acc},\sigma\right)=\delta\left(q_0,b\right)=\left\langle q_{acc},q_{acc}\right\rangle$ כאשר $\alpha=\left\{q_{acc}\right\}$. ראינו שאין אוטומט DBT עבור שפות מהסוג הזה. זה נבע מכך שהיינו צריכים לנחש מסלול.

עבור שפה $L\subset\Sigma^\omega$ נגדיר derivable language of L ע"י

$$\operatorname{der}\left(L\right):=\left\{\left\langle T,\tau\right\rangle\mid\forall\pi\subseteq T,\tau\left(\pi\right)\in L\right\}$$

כלומר כל העצים בהם כל המסלולים הם מילים בשפה.

דוגמא: $L=\left(ab^*\right)^\omega$, אז $\operatorname{der}\left(L\right)$ הוא כל העצים שבכל המסלולים יש ∞ aים.

אם באוטומט DBW עבור L, $\delta\left(q,a\right)=q_1,\delta\left(q,b\right)=q_0$ לכל $q\in\left\{q_0,q_1\right\}=Q$, באוטומט $\mathcal{A}_T$ יהיה לנו $\delta\left(q,a\right)=\left\langle q_1,q_1\right\rangle,\delta\left(q,b\right)=\left\langle q_0,q_0\right\rangle$.

טענה 3.5 עבור DBW $\mathcal{A}$, $L\left(\mathcal{A}_t\right)=\operatorname{der}\left(L\left(\mathcal{A}\right)\right)$ כאשר $\mathcal{A}_t$ הוא כמו $\mathcal{A}$, אולם $\delta_t\left(q,\sigma\right)=\left\langle\delta\left(q,\sigma\right),\delta\left(q,\sigma\right)\right\rangle$.

הוכחה: אם $\left\langle T,\tau\right\rangle\in L\left(\mathcal{A}_t\right)$, נתבונן בריצה $\left\langle T,r\right\rangle$ של $\mathcal{A}_t$ על $\left\langle T,\tau\right\rangle$. יהי $\pi\subseteq T$ מסלול, $r\left(\pi\right)$ היא ריצה חוקית מקבלת של $\mathcal{A}$ על $\tau\left(\pi\right)$. לכן כל המסלולים ב־$\left\langle T,\tau\right\rangle$ מסומנים במילה שתתקבל ע"י $\mathcal{A}$.

אם $\langle T,\tau\rangle \in \mathrm{der}\left(L\left(\mathcal{A}\right)\right)$, לכל $\pi \subseteq T$, $\tau\left(\pi\right) \in L\left(\mathcal{A}\right)$ כלומר כל המסלולים ב־T מסומנים במילה שתתקבל ע"י $\mathcal{A}$. נראה שהריצה של $\mathcal{A}_t$ על $\langle T,\tau\rangle$ מקבלת. נשים לב כי באוטומט דטרמיניסטי, מילים עם אותה רישא, הריצה עליהם מתחילה אותו דבר. מתקיים כי הריצה של $\mathcal{A}$ על כל מסלול היתה מקבלת. ■

הגדרת $\mathcal{A}_t$ ל־NBW $\mathcal{A}$:

$$\delta_t\left(q,\sigma\right) = \delta\left(q,\sigma\right) \times \delta\left(q,\sigma\right)$$

לדוגמא: בשפה של $L = \left(a+b\right)^* b^\omega$ (מספר סופי של a). האוטומט "מנחש" מתי מתחיל זנב של b. הטבלה היא

$$\delta\left(q_0,a\right) = \left\{q_0\right\}, \delta\left(q_0,b\right) = \left\{q_0,q_1\right\}, \delta\left(q_1,a\right) = \emptyset, \delta\left(q_1,b\right) = \left\{q_1\right\}$$

טבלת המעברים עבור δ_t תהיה

δ_t	a	b
q_0	$\langle q_0,q_0\rangle$	$\langle q_0,q_1\rangle, \langle q_0,q_0\rangle, \langle q_1,q_0\rangle, \langle q_1,q_1\rangle$
q_1	$\emptyset$	$\langle q_1,q_1\rangle$

קל לראות כי $L\left(\mathcal{A}_t\right) \subseteq \mathrm{der}\left(L\left(\mathcal{A}\right)\right)$ באופן זהה להוכחה הדטרמיניסטי ־ ריצה מקבלת על $\left(T,\tau\right)$ ב־$\mathcal{A}_t$ משרה ריצות מקבלות של $\mathcal{A}$ על כל המסלולים.

המשימה היא למצוא עץ שבכל מסלול שלו יש רק מספר סופי של a, ובכל זאת אינו מתקבל: נבנה עץ בו המסלול השמאלי ביותר הוא b^ω, וכל סטיה ימינה בשלב i נותנת לנו a בודד וזנב של b. העץ הזה לא יתקבל, כי המסלול הכי שמאלי חייב להיות כל הזמן ב־q_0, כדי לתמוך בa שיגיע מההתפצלויות הימניות שלו. לכן $L\left(\mathcal{A}_t\right)$ הוא כל העצים שכל מסלוליהם מגיעים לתת־עץ שכולו b.

החסרון של אוטומט א"ד מעל עצים הוא שהוא צריך לנחש מעל כל העתידים האפשריים. במצב כזה, א"ד לא מוסיף לנו הרבה.

ראינו כי $DFW \xrightarrow[2^n]{2^n} NFW$. מעל $\Sigma_n = [n] \cup \{\#\}$ כאשר

$$L_n = \left\{\sigma_1,\ldots,\sigma_k\#\sigma_{k+1} \mid \sigma_i \in [n], \sigma_{k+1} \in \left\{\sigma_1,\ldots,\sigma_k\right\}\right\}$$

למשל $122\#4 \notin L_4$ אולם $122314\#3 \in L_4$.

אנו יודעים כי:

1. יש NFW קטן ־ האוטומט ינחש את האות האחרונה, הרכיב שמנחש את הסיומת 1, נשאר במצב ההתחלתי בקריאת $2,\ldots,n$. אם רואה 1 הולך למצב הבא, ושם נשאר עד שיראה $\#1$ ויקבל ($\#2$ תתקע ברכיב הזה). זהו אוטומט עם $3n+1$ מצבים.
2. כל DFW חייב להיות עם 2^n מצבים לפחות.

נקבל כי $\operatorname{der}(L_n)$ הוא כל העצים שכל המסלולים שלהם ב־L_n. אם יש עץ עם שני מסלולים שנגמרים במספרים שונים, אני לא אצליח לנחש את $r(\varepsilon)$ נכון (נשים לב כי טבלת המעברים היא דטרמיניסטית, ואי הדטרמיניזם הוא בכך ש־Q_0 הוא בגודל n).

כ"א סיון תשע"ג (שעור 9)

4 אוטומטים ומשחקי And-Or

4.1 הגדרות

יש לנו גרף מכוון, עם תיוג $t:V\to\{\wedge,\vee\}$. בכל קודקוד אחד השחקנים בוחר לאן ללכת. יש כמה תנאי ניצחון אפשריים. הראשון הוא בסיסי ־ האם שחקן האו יכול להכריח את שחקן הוגם להגיע ממצב אחד למצב אחר.

אסטרטגיה עבור $\vee$: $f_{OR}:V^*V_{OR}\to V$ ובאופן דומה עבור $\wedge$.

בהינתן אסטרטגיה f_{OR} של $\vee$, ו־f_{AND} של $\wedge$, ומצב $v\in V$, יש מסלול $outcome\left(v,f_{OR},f_{AND}\right)$ שאותו יסרוק המשחק שמתחיל מ־v ומתקדם לפי f_{OR},f_{AND}.

אסטרטגיה f היא חסרת זיכרון memoryless strategy אם היא תלויה רק במצב הנוכחי (ולא בכל ההיסטוריה).

עבור תנאי ניצחון הוא חסר זיכרון אם לכל אסטרטגית ניצחון עבורו, יש גם אסטרטגית ניצחון חסרת זיכרון.

תנאי ניצחון generalized Buchi אינו חסר זיכרון. למשל המשחק הבא (ציור) ותנאי הקבלה $\alpha=\left\{\left\{s_1\right\},\left\{s_2\right\}\right\}$ האסטרטגיה

$$f_{OR}\left(w\cdot v\right)=\begin{cases}s_1 & w=u\cdot s_2\\ s_2 & o.w.\end{cases}$$

4.2 מאוטומט עצים למשחק

בהינתן $\mathcal{A} = \langle \Sigma, Q, \delta, Q_0, \alpha \rangle$ ($D = \{0,1\}$) נבנה משחק $G_\mathcal{A}$ כך ששחקן ה־OR ינצח ב־$G_\mathcal{A}$ אם"ם $L(\mathcal{A}) \neq \emptyset$.

- $V_{OR} = Q$, $V_{AND} \subset Q \times Q$.
- $E(q, \langle s_1, s_2 \rangle) \Leftrightarrow \exists \sigma, \langle s_1, s_2 \rangle \in \delta(q, \sigma)$ וכן $E(\langle s_1, s_2 \rangle, q) \Leftrightarrow (q = s_1 \vee q = s_2)$.
- תנאי הניצחון הוא α.

המשמעות של המשחק, מתחילים מהשורש, שחקן $\vee$ בוחר איזה מצבים לתת לשני הבנים, שחקן $\wedge$ יכול לדרוש ממנו שיראה איך הוא ממשיך מאחת האפשרויות.

דוגמא:

$$\begin{aligned} \delta(q,a) &= \{\langle q,q \rangle\} \\ \delta(q,e) &= \{\langle q,s \rangle, \langle s,q \rangle\} \\ \delta(q,b) &= \delta(s,\sigma) = \{\langle s,s \rangle\} \end{aligned}$$

הגרף כמו בציור. החיצים על הקשתות של שחקן OR אומרות מהי σ שיוצרת את הקשת הזו, החיצים על הקשתות של שחקן AND אומרות האם $q = s_1$ או $q = s_2$.

לשחקן OR יש אסטרטגית ניצחון ללכלת ל־$\langle s,s \rangle$, ומשם ניכנס למעגל.

באופן פורמלי אפשר להוכיח שתי טענות:

טענה 4.1 אם ל־$\vee$ יש אסטרטגית ניצחון (מאחד הקודקודים ב־Q_0) אז $L(\mathcal{A}) \neq \emptyset$.

הוכחה: נבנה עץ וריצה מקבלת עליו. בהינתן $f_{OR} : V^* V_{OR} \to V$ נשים לב כי ההסטוריה האפשרית היא מסוג $(V_{OR} \cdot V_{AND})^*$, וכי היעד הוא V_{AND}. לכן $f_{OR} : (Q \cdot (Q \times Q))^* \cdot Q \to Q \times Q \, (\times \Sigma)$ כאשר Σ נוספה כי כל מעבר לתיוג $Q \times Q$ נעזר ב־$\sigma \in \Sigma$ כדי לבנות את העץ: $f_{OR}(q_0) = \langle q_1, q_2, \sigma \rangle$ אז $\tau(\varepsilon) = \sigma$. אח"כ אם f_{AND} בוחר באחד הצדדים, ו־$f_{OR}(q_0, \langle q_1, q_2 \rangle, q_1) = \langle q_3, q_4, \theta \rangle$, נתייג $\tau(0) = \theta$, וכדומה.

■

טענה 4.2 אם $L(\mathcal{A}) \neq \emptyset$, יש ל־$\vee$ אסטרטגית ניצחון.

הוכחה: אם יש לנו ריצה מקבלת על עץ מסויים, אפשר לבנות ממנה אסטרטגית ניצחון.

■

4.3 הכרעת משחקים

בהינתן משחק G, מצא את קבוצת הקודקודים W שמהם יש לשחקן ה־$\vee$ אסטרטגית ניצחון (יש f_{OR} כך שלכל f_{AND} מתקיים ש$outcome\left(u, f_{OR}, f_{AND}\right)$ מקיים את תנאי הניצחון אם"ם $u \in W$).

נשים לב שאם לכל מצב יש אסטרטגיה מנצחת משלו, אפשר להרכיב מכולם אסטרטגיה משותפת ־ לפי ההתחלה של הסדרות.

4.3.1 פתרון משחקי ישיגות

נראה שאם יש אסטרטגיה, יש גם אסטרטגיה חסרת זיכרון, שמשמעותה פשוט הסרת חלק מהקשתות (והשארת קשת בודדת לכל קודקוד של שחקן OR), וזה עד לניצחון, אם כל המסלולים בגרף שנשאר הם טובים (אסטרטגיה רעה אם קיים מסלול שלא מגיע ל־α).

אלגוריתם פולינומיאלי לחישוב W (מצבי הניצחון של $\vee$) ־ האלגוריתם יפעל בצורה איטרטיבית:

1. $W_0 = T$ (T הוא המצבים אליהם אנחנו רוצים להגיע).
2. $W_{i+1} = \{v \in V_{OR} \mid \exists u \in W_i,\ E\left(v, u\right)\} \cup \{v \in V_{AND} \mid \forall u\, E\left(v, u\right),\ u \in W_i\} \cup W_i$

W_i הוא אוסף המצבים שיש אסטרטגיה להגיע אליהם בתוך i צעדים. מספר האיטרציות הוא הקוטר של הגרף. הבנייה הזו גם נותנת לנו אסטרטגיה חסרת זיכרון. נשים לב שעל כל קשת עוברים פעם אחת, ואפשר מימוש ליניארי אם שמים מונים לכל צומת על מספר הקשתות היוצאות.

זוהי בעיה קלאסית שהיא $PTIME - complete$.

4.3.2 פתרון משחק בוקי

באוטומט שראינו, $7, 8$ הם הקודקודים היחידים שמהם אי אפשר להשיג. בבעיה הנוכחית, אנחנו רוצים להבטיח שיהיו ביקורים אינסופיים בקודקודים המבוקשים, והם האידיאלים עבור שחקן ה־$\wedge$. אפשר לדבר על המשחק הדואלי ־ שחקן ה־$\wedge$ רוצה להגיע אליהם, וזה אותו הדבר, רק שהפעם זה משחק $co - Buchi$.

הסיבוכיות היא עדיין בעיה פתוחה, היה פעם מאמר שהתיימר לפתור בזמן לינארי, אולם הוא לא התפרסם בז'ורנל.

נגדיר את $nonempty-reach\left(S\right)$ כקבוצת המצבים שמהם יש לשחקן ה־$\vee$ אסטרטגיה להגיע ל־S במסלול לא ריק. ניתן לחשב את $NER\left(S\right)$ בזמן לינארי כמו קודם, רק שמתחילים מ־$W_1 = \{v \in V_{OR} \mid \exists u \in S,\ E\left(v,u\right)\} \cup \{v \in V_{AND} \mid \forall E\left(v,u\right), u \in S\}$.

כעת, נגדיר $B_0 = NER\left(T\right)$, אלו המצבים שמהם נגיע לפחות פעם אחת ל־T.

ו־$B_{i+1} = NER\left(T \cap B_i\right)$ - המצבים שמהם נוכל להבטיח הגעה ל־T לפחות $i+1$ פעמים (באינדוקציה על i).

בדוגמא שלנו, $B_0 = NER\left(\{2,6\}\right) = \{1,2,3,4,5\}$. $B_1 = NER\left(\{2\}\right) = \{1,2,4,5\}$. מכאן $B_2 = NER\left(\{2\}\right)$ והגענו לנקודת שבת. זו גם הוכחה לקיום אסטרטגיה חסרת זיכרון.

הסיבוכיות היא $O\left(|V| \cdot |E|\right)$. זו בעיה פתוחה אם אפשר לבצע הכל באיטרציה אחת.

4.3.3 פתרון משחקי רבין

ראינו כי יש $NRW_{n,k} \to NBW_{nk}$. עבור $\alpha = \{\langle G_i, B_i \rangle\}_{i=1}^{k}$. משחקי רבין גם הם חסרי זיכרון, ולכן הם שייכים ל־NP.

בהינתן עד (אסטרטגיה חסרת זיכרון), האם קיים מסלול שמקיים את תנאי $Streete$ הדואלי. ריקנות של $Streete$ זה פולינומיאלי.

נראה $NP-hardness$ ע"י רדוקציה מ־SAT ('88):

בהינתן $\theta = C_1 \wedge \ldots \wedge C_m$ כאשר $C_i = l_i^1 \vee l_i^2 \vee l_i^3$ ו־$L_i^j \in \{x_p, \neg x_p\}$.

$V_{AND} = \{\theta, x_1, \neg x_1, \ldots, x_n, \neg x_n\}$, $V_{OR} = \{C_1, \ldots, C_m\}$.

הקשתות הן מסוג:

- $E\left(\theta, C_i\right)$ לכל $i \in [m]$,
- $E\left(C_i, l_i^j\right)$ לכל $j \in [3]$.
- מכל x_p ל־C_i כך ש־$\neg x_p$ ב־C_i.
- מכל $\neg x_p$ ל־C_i כך ש־x_p ב־C_i.
- חוגים עצמיים ל־x_p, $\neg x_p$.

- $.\alpha = \bigcup_{1\leq i\leq n} \{\langle\{x_i\},\{\neg x_i\}\rangle,\langle\{\neg x_i\},\{x_i\}\rangle\}$

אם θ ספיקה בהשמה כלשהי, נסמן את קודקודי הליטרלים הטובים לפי ההשמה, וכל פעם ש־$\wedge$ שואל אותי על C_i כלשהו, נלך לאחד מהם. נבקר רק בכאלו שמתאימים להשמה (אינסוף פעמים), ולא לאלו שלא מתאימים להשמה.

אם יש לי אסטרטגית ניצחון,

5 אוטומטים מתחלפים alternating automata

כ"ח סיון תשע"ג (שעור 9)

5.1 הגדרות

הסמטניקה של אי דטרמיניזם: $\delta(q,a)=\{q_1,q_2,q_3\}$ שמשמעותו $q_1\vee q_2\vee q_3$. אפשר באוטומט מתחלף לתת כל דרישה בוליאנית כמו $\delta(q,a)=q_1\vee(q_2\wedge q_3)$. $\vee$ מציין אי דטרמיניזם, ו־$\wedge$ מציין ריבוי דרישות (אוניברסליות). המשמעות של הביטוי הקודם היא שהסיפא מתקבלת מ־q_1, או גם מ־q_2 וגם מ־q_3.

עבור קבוצה X נסמן ב־$B^+(X)$ את קבוצת הנוסחאות הבוליאניות החיוביות עם אטומים מ־X ($\theta = x \mid \theta\wedge\theta \mid \theta\vee\theta \mid true \mid false$)

עבור נוסחא $\theta\in B^+(X)$ וקבוצה $Y\subset X$ נאמר כי Y מספקת את θ אם ההשמה $f_Y : X\to\{T,F\}$ כך ש־ $f_Y(x)=\begin{cases}T & x\in Y\\ F & x\notin Y\end{cases}$. למשל עבור $X=\{x_1,x_2,x_3,x_4\}$ ו־$\theta = x_1\wedge(x_2\vee x_3)$ מספקות את θ הקבוצות $\{x_1,x_2\}$ ו־$\{x_1,x_3\}$, וכן כל הקבוצות המכילות אותן (סגורות מעלה).

אוטומט מתחלף הוא

$$\mathcal{A}=\langle\Sigma,Q,q_0,\delta,\alpha\rangle$$

כאשר $\delta : Q\times\Sigma\to B^+(Q)$.

ריצה של $\mathcal{A}$ על מילה $w=w_1w_2\ldots$ היא עץ $Q-labeled\,\mathbb{N}-tree$. הוא לא יהיה עץ שלם (דרגת הפיצול בכל קודקוד היא סופית). הוא יהיה $\langle T_r,r\rangle$ כאשר $T_r\subset\mathbb{N}^*$, כך ש־

1. $\varepsilon \in T_r$ ו־$q_0 - r(\varepsilon)$.
2. יהי $x \in T_r$, נתבונן ב־$\theta = \delta\left(r(x), w_{|x|+1}\right)$. קיימת קבוצה $Y = \left\{s_0, \ldots, s_{|Y|-1}\right\} \subseteq Q$ כך ש־
   (א) Y מספקת את θ,
   (ב) $\{x \cdot 0, x \cdot 1, \ldots, x \cdot (|Y|-1)\} \subset T_r$.
   (ג) לכל $0 \leq i \leq |Y|-1$, $r(x \cdot i) = s_i$.

דוגמא: $\delta(q_0, a) = q_1 \wedge (q_2 \vee q_3)$. ריצות אפשריות על $a(bc)^\omega$. אנחנו מתחילים מ־$r(\varepsilon) = q_0$. משם אפשר להמשיך ל־$\{x_1, x_2\}, \{x_1, x_3\}$ או $\{x_1, x_2, x_3\}$ (דרגת פיצול 2 או שלוש ־ אלא אם כן נדרוש קבוצה מינימלית).

המשמעות של $\delta(q, a) = true$ היא שאין צורך שיהיו ל־q בנים בעץ. אין ריצה שמשתמשת ב־$\delta(q, a) = false$ (היה אפשר להשתמש בבור מקבל ובור דוחה במקום זה).

$\langle T_r, r\rangle$ מקבלת אם כל המסלולים האינסופיים מבקרים ב־α אינסוף פעמים (מסלולים שהסתיימו ב־$true$ הם גם מקבלים).

מילה w מתקבלת אם קיימת ריצה מקבלת על w.

דוגמא: $\Sigma_n = [n]$, $L_n = \{w \mid \forall i \in \Sigma, ii \in w\}$ (כלומר הרצף ii מופיע ב־w). אוטומט NBW עבור L_n הוא אקספוננציאלי ב־n (צריך לזכור אלו רצפים כבר היו). ABW דורש $2n+1$ מצבים, כאשר q_0 הוא מצב התחלתי, q_{2i-1}, q_{2i} אורבים ל־ii.

$$\begin{aligned}
\delta(q_0, i) &= q_{2i} \wedge \bigwedge_{j \neq i} q_{2j-1} \\
\delta(q_{2i-1}, i) &= q_{2i} \\
\delta(q_{2i}, i) &= true \\
\delta(q_{2i}, j) = \delta(q_{2i-1}, j) &= q_{2i-1}\,(if\ j \neq i)
\end{aligned}$$

אם $E(xists)$ מציין א"ד, ו־$\forall(ll)$ מציין אוניברסליות, $\emptyset$ מציין דטרמיניזם $E+A$ מציין מתחלפים.

5.2 כוח ההבעה של אוטומטים מתחלפים

5.2.1 $AFW \to NFW$.

תנאי הקבלה של AFW הוא שכל הקודקודים ברמה הסופית הם מצבים מקבלים (ב־α). ניתן לבצע את זה בצורה דומה לחרצון. מאוטומט $E+A$ עוברים לאוטומט E.
אם $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, \alpha \rangle$ נייצר $\mathcal{A}' = \left\langle \Sigma, 2^Q, \delta', \{q_0\}, \alpha' \right\rangle$ כאשר

$$\delta'(S,\sigma) = \left\{ T \mid T\, satisfy \bigwedge_{s\in S} \delta(s,\sigma) \right\}$$

ו־$\alpha' = 2^{\alpha}$. אם נרצה $AFW \to DFW$ אפשר לבצע חרצון של NFW, ומרחב המצבים יהיה 2^{2^Q}. נראה חסם תחתון של $2Exp$.
משפחה $L_1, L_2, \ldots$ של שפות כך ש־

1. ל־L_n יש AFW בגודל $O(n)$.
2. DFW המינימלי עבור L_n צריך 2^{2^n} מצבים.

$\Sigma = \{0,1,\#,\$\}$. $L_n = (0+1+\#)^* \#w\# (0+1+\#)^* \$w$ כך ש־$w \in (0+1)^n$.
למשל

$$01\#\#0110\#000\#01110\#010\#\#0\$000 \in L_3$$

אוטומט יהיה דאבל־אקספוננט כי בבואו לקרוא את ה־\$ ה־DFW צריך לזכור תת קבוצה של וקטורים ב־$(0+1)^n$.

אוטומט מתחלף ב־$O(n^2)$: כל פעם שנראה $\#$ ננחש האם זו המילה שהולכת להופיע בסוף. בניחוש הנוכחי, אם נסמן $w = \sigma_1 \ldots \sigma_n$, הוא ישלח עותק שיבדוק את האות הראשונה פה ואחרי ה־\$. עותק נוסף יבדוק את האות השניה. כל עותק כזה דורש $O(n)$ מצבים, ואנחנו צריכים $O(n)$ עותקים.

איך אפשר לרדת ל־$O(n)$? אפשר להשתמש בא"ב עשיר $\Sigma \times [n]$. יש לנו מקביליות חסרת שיתוף פעולה בין העותקים. אפשר לחשוב על מודלים חזקים יותר של תיאום בין העותקים ־ עותקים שיכולים לפזול הצידה. דרך לבדיקה היא כמה מצבים נחוצים כדי לזהות את השפה $L_n = \{a^n\}$. עבור DFW יש צורך ב־$n+1$ מצבים. עבור NFW אותו הדבר. עבור UFW אפשר לבצע עם $\sqrt{n}$ מצבים, ועבור AFW אפשר לפי משפט השאריות הסיני בעזרת $\log n$ מצבים (מאמר באתר הבית של המרצה).

קל לראות כי AFW סגור לשלילה. עבור AFW $\mathcal{A}$ אפשר לעבור ל־$\tilde{\mathcal{A}} = \left\langle \Sigma, Q, \tilde{\delta}, q, \tilde{\alpha} \right\rangle$ כאשר $\tilde{\delta}$ ממירים $\wedge, \vee$ ו־T, F והופכים את תנאי הקבלה (ההוכחה הפורמלית עוברת דרך שריגים, אבל קל להשתכנע אינטואיטיבית).

אין NCW עבור ∞aים (ראינו כי אין DBW עבור $\neg\infty a$). נשים לב כי $NCW = \overline{UBW}$.

ננסה למצוא UCW עבור ∞a. זה אותו אוטומט של NBW עבור $\neg\infty a$ עם מעבר לבור דוחה מהמצב המקבל. נקבל $\delta(q_0, a) = q_0$, $\delta(q_0, b) = q_0 \wedge q_1$. $\delta(q_1, b) = q_1$ ו־$\delta(q_1, a) = T$. לכן התחלפות מוסיפה לכוח הבעה של $Co - Buchi$.

\לדח