אוטומטים ושפות פורמליות

ד"ר פרג' שיבאן
סמסטר א' - תשע"ו

חלק I

הגדרות בסיסיות

אלפבית (א"ב) - של שפה היא קבוצה <u>סופית</u> Σ שאבריה יקראו אותיות (או תווים).

מילה (או מחרוזת) **מעל א"ב Σ** - זוהי סדרה <u>סופית</u> של אותיות מ-Σ, שרשומה כשירשור ללא פסיקים בין אברי הסדרה.
מילה (אשר מסומנת ב-w):

$$a_i \in \Sigma,\ w = a_1a_2...a_n$$

למשל: אם $\Sigma = \{0,1\}$ אזי מילים מעל Σ הינן למשל: $0, 0111, 01010, ...$

האורך של מילה w מסומן $|\boldsymbol{w}|$ ומוגדר בתור מספר האותיות ב-w.

עבור $\Sigma \ni a$ ומילה w מעל Σ מסמנים ב-$\#_{\boldsymbol{a}}(\boldsymbol{w})$ את מספר המופעים של a ב-w.
אזי מתקיים:

$$\sum_{a\in\Sigma} \#_a(w) = |w|$$

מילה באורך אפס מסומנת ב-ε ונקראת **המילה הריקה**.
מסמנים ב-Σ^* את אוסף <u>כל</u> המילים מעל א"ב Σ.
חשוב לזכור:
Σ - סופית, Σ^* - <u>אינסופית</u>, אבל בת-מנייה.

עבור שתי מילים מ-Σ^*:
$w = a_1 \cdots a_n$ ו-$v = b_1 \cdots b_n$ אזי <u>**השירשור**</u> של w, v מסומן: $\boldsymbol{wv}$ ומוגדר ע"י:

$$wv = a_1 \cdots a_n b_1 \cdots b_n$$

<u>**שפה**</u> מעל א"ב Σ הינה אוסף מילים מ-Σ^*.

L שפה על Σ $\iff$ $L \subseteq \Sigma^*$.
נשים לב שמשפחת השפות מעל Σ <u>אינה</u> בת מנייה, היא מעוצמה $2^{\aleph_0}$.
שפות הן תת-קבוצות של Σ^*, לכן בין שפות מעל Σ מוגדרות פעולות איחוד $(L_1 \cup L_2)$, חיתוך $(L_1 \cap L_2)$ ומשלים: $L^C = \Sigma^* - L$.

בנוסף, מוגדרות בין שפות מעל Σ עוד שתי פעולות: <u>שירשור</u> <u>ואיטרציה</u>:
<u>שירשור</u>
עבור $L_1, L_2 \subseteq \Sigma^*$ השירשור L_1L_2 מוגדר ע"י:

$$L_1L_2 = \{uv \big| u \in L_1, v \in L_2\}$$

<u>איטרציה</u>
עבור שפה $L \subseteq \Sigma^*$ ועבור $n \geq 0$ שלם נגדיר L^n באינדוקציה:

$$\begin{aligned} L^0 &= \{\varepsilon\} \\ L^{n+1} &= L^n \cdot L \end{aligned}$$

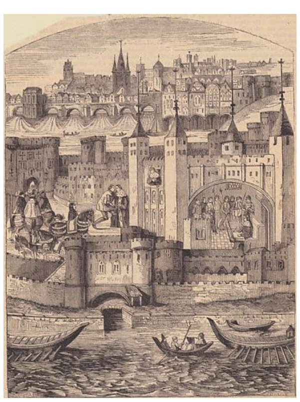
L^n = שפת כל השירשורים של n מילים מעל L.
<u>האיטרציה של L</u> (או: * של L) - מסומנת L^* ומוגדרת ע"י:

$$L^* = \bigcup_{n\geq 0} L^n = L^0 \cup L^1 \cup L^2 \cdots \cup L^n$$

L^* = קבוצת כל השירשורים של מספר כלשהו של מילים מתוך L (כולל ε - שירשור של אפס מילים מ-L).

0.1 כמה הערות על שפה ריקה ועל מילה ריקה

ישנו הבדל בין המילה הריקה לשפה הריקה. כלומר, אם $L = \emptyset$, אזי:
$\varepsilon \notin \boldsymbol{L}$, אלא אם זה הוגדר. אבל אם $\varepsilon \in L$ - אזי בהכרח L אינה ריקה, כלומר:
$L \neq \emptyset$.
כמו-כן, ע"פ ההגדרה, אם אנחנו צריכים לשרשר $L_1 \cdot L_2$ ואחת מהן היא שפה ריקה, אזי גם השירשור שלהן יהיה הקבוצה הריקה, כלומר:
$L_1 \cdot L_2 = \emptyset$.
כדאי לשים ♡ לדברים הללו.



חלק II
אוטומטיים סופיים

1 אוטומט סופי דטרמיניסטי (אס"ד)

1.1 הגדרה פורמלית

אוטומט סופי דטרמיניסטי (DFA) - זהו חמישה (דבר המכיל חמישה דברים):

$$\boldsymbol{A = \langle Q, \Sigma, \delta, q_0, F \rangle}$$

כאשר:

- Q - היא קבוצה **סופית** שאבריה נקראים **מצבים**.
- Σ - א"ב. זה הא"ב של האוטומט.
- δ - היא פונקציה $\delta : Q \times \Sigma \to Q$.
כלומר, לכל $q \in Q$ ולכל $a \in \Sigma$, δ מתאימה מצב שמסומן: $\delta(q,a)$. זה המצב שאליו יכנס האוטומט אם הוא נמצא במצב q וקורא את האות a.
- q_0 - הוא איבר מיוחד ב־Q שנקרא המצב ההתחלתי.
- F - היא קבוצה חלקית ל־Q [קבוצת מצבים] שאבריה נקראים **המצבים הסופיים**, או **המצבים המקבלים**.

1.2 הגדרות נוספות

יהי $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, DFA.
עבור מילה $w = a_1 \cdots a_n$ ב־Σ^*:
הריצה של A על w - היא סדרת מצבים: $q_0, q_1, q_2, ..., q_n$ כך ש־q_0 הוא המצב ההתחלתי של A, ולכל $1 \leq i \leq n$ מתקיים:

$$q_i = \delta(q_{i-1}, a_i)$$

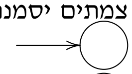
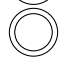
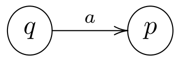
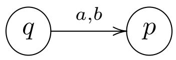
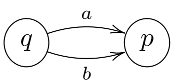
A **מקבל** את המילה w אם הריצה של A על w מסתיימת במצב מקבל (או סופי) (ז"א, $q_n \in F$).
אחרת, A **דוחה** את w.

השפה של A, מסומנת $L(A)$ ומוגדרת ע"י:

$$L(A) = \left\{ w \in \Sigma^* \middle| \odot \right\}$$

$\odot$ - A מקבל את w.

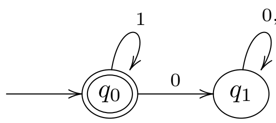
1.3 סימון גרפי

נציג DFA באמצעות גרפים מכוונים עם קשתות מסומנים באותיות מ־א"ב האוטומט.
צמתים יסמנו מצבים:
 - מצב התחלתי.
 - מצב סופי (או: מצב מקבל).
קשתות:  מסמן: $\delta(q,a) = p$.
 מקצרים כך: 
כמו־כן, במקום לכתוב את הא"ב על קשת, ניתן פשוט לכתוב Σ.

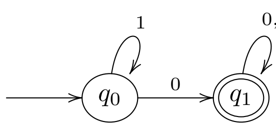
הערה: אומרים שאוטומט סופי דטרמיניסטי (DFA) A מקבל את השפה L (או: A מזהה את L) אם $L(A) = L$.

1.4 בור דוחה

אחד הדברים שכדאי להכיר הוא בור זוחה, כלומר מצב שממנו לא ניתן לצאת לשום מצב אחר.
למשל, אם אצלנו $\Sigma = \{0,1\}$ ו־L היא על המילים אשר מכילות 1 בלבד, אזי האוטומט שמקבל שפה זאת הינו:



נשים לב לכך שברגע שקיבלנו את האות "0" אנחנו נכנסים למצב שאין לנו שום דרך לצאת ממנו - כלומר, הגענו לבור דוחה.
כמו כן זה יכול להיות כמובן ההפך, למשל, אם ניקח את השפה המשלימה, כלומר, שפה שבה בכל מילה יש "0" אחד לפחות, אזי האוטומט שיקבל את השפה הינו:



וגם כאן, ברגע שבקלט היה "0" אחד - האוטומט מקבל את המילה והגענו לבור כזה (שנקרא הפעם "בור מקבל").

1.5 הרחבת δ

מבלי להיכנס להסברים, ניתן להרחב את δ שתהיה תקפה עבור מילה ולא רק עבור אות, כלומר:

$$\delta : Q \times \Sigma^* \to Q$$

ולכן הפונקציה תתנהג באופן הבא:

$$\begin{cases} \delta(q, \varepsilon) = q, & \forall q \in Q \\ \delta(q, wa) = \delta(\delta(q,w), a), & \forall q \in Q, \forall w \in \Sigma^*, \forall a \in \Sigma \end{cases}$$

נשים לב לכך ש:

$$\boxed{\boldsymbol{L(A)=\left\{w\in\Sigma^*\middle|\delta(q_0,w)\in F\right\}}}$$

או במילים:

$$\boxed{A \text{ מקבל את המילה } w \iff \delta(q_0,w)\in F}$$

תכונה נוספת של הפונקציה δ:

$$\boldsymbol{\delta(q,u\cdot v)=\delta(\delta(q,u),v)},\quad \forall q\in Q,\forall u,v\in\Sigma^*$$

1.6 שפה רגולרית

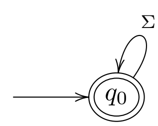
הגדרה 1.1 שפה $L\subseteq\Sigma^*$ נקראת **שפה רגולרית** אם קיים DFA A כך ש־$\boldsymbol{L(A)=L}$.

דוגמא לשפה לא רגולרית: $\Sigma=\{a,b\}$, $L=\left\{a^nb^n\middle|n\in\mathbb{N}\right\}$ או אותה שפה בכתיב אחר: $L=\left\{w\middle|\#_a(w)=\#_b(w)\right\}$.

1.7 משפחת השפות הרגולריות (מעל א"ב מסוים)

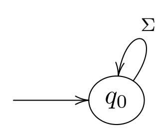
יהי Σ א"ב.

1. Σ^* - היא שפה רגולרית:



האוטומט הנ"ל מזהה את Σ^*.

2. $\emptyset$ היא שפה רגולרית:



אוטומט זה מזהה את $\emptyset$. באוטומט זה $F=\emptyset$.

1.8 אוטומט מכפלה

יהיו:

$$\begin{aligned}A_1 &= \langle Q_1,\Sigma,\delta_1,q_1,F_1\rangle\\ A_2 &= \langle Q_2,\Sigma,\delta_2,q_2,F_2\rangle\end{aligned}$$

שני DFA־ים מעל אותו א"ב Σ, **אוטומט מכפלה** של A_1 ו־A_2 זהו אוטומט (DFA) A:

$$A=\langle Q,\Sigma,\delta,q_0,F\rangle$$

שמקיים:

- $Q=Q_1\times Q_2=\left\{q\times r\middle|q\in Q_1,r\in Q_2\right\}$
- $\delta:Q\times\Sigma\to Q$
- $\delta((q,r),a)=(\delta_1(q,a),\delta_2(r,a))$
- $q=(q_1,q_2)$
- $F\subseteq Q$ (היא משתנה מאוטומט מכפלה אחד למשנהו). המשתמש קובע אותה לפי השפה שרוצים שהאוטומט יקבל.

ניתן לומר שבאיזשהו אופן, אוטומט מכפלה זה בעצם שילוב של שני אוטומטים שרצים במקביל וקבוצת המצבים המקבלים היא בהתאמה למה שאנחנו רוצים.
(דוגמא בהמשך...)

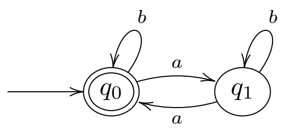
1.8.1 הרחבת δ באוטומט מכפלה

$$\delta:Q\times\Sigma^*\to Q$$

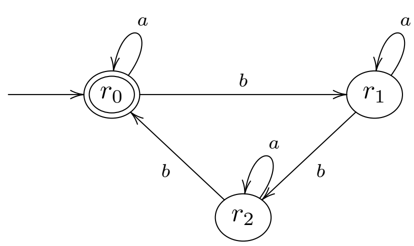
$$\begin{cases}\delta((q,r),\varepsilon) &= (q,r),\ \ \forall(q,r)\in Q\\ \delta((q,r),wa) &= \delta(\delta((q,r),w),a),\ \ \forall(q,r)\in Q,w\in\Sigma^*,a\in\Sigma\end{cases}$$

1.8.2 דוגמא לאטומט מכפלה

נסתכל בשני האוטומטים הבאים:
A_1:



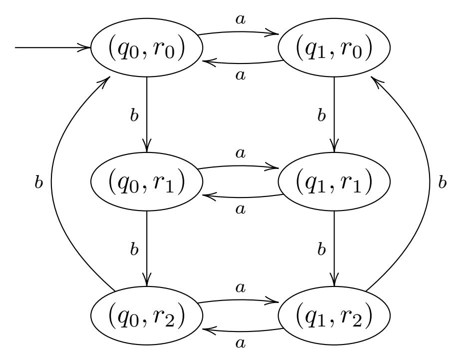
A_2:



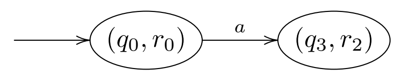
נשים לב לכך ש:

$$L(A_1)=\left\{w\in\{a,b\}^* \;\middle|\; 2 \mid \#_a(w)\right\}$$
$$L(A_2)=\left\{w\in\{a,b\}^* \;\middle|\; 3 \mid \#_b(w)\right\}$$

כלומר: $L(A_1)$ - מספר ה־a־ים זוגי, $L(A_2)$ - מספר ה־b־ים מתחלק ב־3 (תזכורת: $n \mid m$ פירושו ש־n מחלק את m).
כל אוטומט של A_1 ו־A_2 נראה כך: (ישנם לכל היותר שישה מצבים, היות ו־$|Q_1|\cdot|Q_2|=6$):



כיצד בונים את האוטומט מכפלה?
מתחילים מ־(q_0,r_0), המצב ההתחלתי של כל אוטומט, ואז בודקים כל אות בא"ב לאן היא מובילה למשל, אם נניח יש לנו בא"ב את האות a אזי בודקים לאן q_0 מעביר אותנו כאשר נותנים לו a ולאן r_0 מעביר אותנו שנותנים לו a, לדוגמא: $q_0\to q_3$ ו־$r_0\to r_2$ ואז מה שנקבל הוא:



ואחרי שעשינו ככה לכל $x\in\Sigma$ אנחנו יכולים ללכת ל־(q_3,r_2) [או מצב אחר אם ישנו] ולעשות את אותו הדבר בדיוק עבור כל $x\in\Sigma$.
וכך אנחנו ממשיכים הלאה עד שאנחנו מגיעים לכל המצבים או שאנחנו רואים שאין לאן להמשיך (לא חייב להיות שנגיע לכל $|Q_1|\cdot|Q_2|$ המצבים).
בצורה פורמלית יותר, אנחנו מתחילים מ־(q_0,r_0) ובודקים עבור כל $x\in\Sigma$ מהי: $\delta\left((q_0,r_0),x\right)$ ולאחר מכן ממשיכים לאיזשהו מצב אחר ש־(q_0,r_0) הוביל אותנו אליו.
לגבי קבוצת המצבים המקבלים באוטומט מכפלה - F - היא אינה קבועה! וניתן להגדיר אותה בהתאם לצורך. בהתאם לכך גם תוגדר שפה אוטומט המכפלה כמובן.
למשל, אם נבחר, לגבי אוטומט המכפה הנ"ל:

$$F=(F_1\times Q_1)\cup(Q_1\times F_2)=\left\{(q,r)\in Q \middle| q\in F_1\vee r\in F_2\right\}$$

אזי נקבל ששפת האוטומט (אוטומט המכפלה) היא:

$$L(A)=L(A_1)\cup L(A_2)$$

.

1.9 כמה משפטים נוספים

משפט 1.2 משפחת השפות הרגולריות מעל א"ב Σ סגורה לאיחוד וחיתוך.
כלומר, אם $L_1,L_2\subseteq\Sigma^*$ הן שפות רוגלריות אז כך גם $L_1\cup L_2$ ו־$L_1\cap L_2$.

מסקנה 1.3 משפחת השפות הרגולריות מעל א"ב Σ סגורה לאיחוד ולחיתוך של מספר סופי של שפות[1].

הערה 1.4 לכל $w\in\Sigma^*$ השפה $L=\{w\}$ היא שפה רגולרית (שפה בת מילה אחת).

מסקנה 1.5 כל שפה סופית מעל Σ היא שפה רגולרית (זה נכון, כי כי שפה סופית היא איחוד סופי של שפות שכל אחת מהן מכילה מילה אחת, ולכן היא רגולרית).

טענה 1.6 משפחת השפות הרגולריות מעל א"ב Σ סגורה להפרש ולהפרש סימטרי.
כלומר, אם $L_1,L_2\subseteq\Sigma^*$ הן שפות רגולריות אז כך גם: $L_1\Delta L_2=L_1\oplus L_2$ ו־$L_1\backslash L_2=L_1-L_2$ [2] [3].

אם קיימים באוטומט מצבים שאינם ניתנים להשגה, אזי ניתן להסיר מצבים אלה, יחד עם כל המעברים (החצים) שקשורים אליהם. מה שישאר הוא עדיין DFA שמזהה את אותה שפה של האוטומט המקורי.

[1] במקרה של מספר אינסופי של שפות רגולריות אזי זה אינו נכון.
[2] $L_1-L_2=L_1\backslash L_2=\left\{w\middle|w\in L_1\wedge w\notin L_2\right\}$
[3] $L_1\Delta L_2=L_1\oplus L_2=\left(w\middle|w\in L_1\oplus w\in L_2\right)$ - כלומר האיבר חייב להיות בקבוצה אחת בלבד!

2 אוטומט סופי לא דטרמיניסטי (אסל"ד)

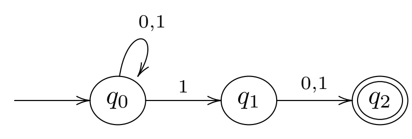
אסל"ד הוא חמישיה:

$$A=\langle Q,\Sigma,\delta,q_0,F\rangle$$

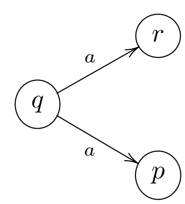
שבה: Q,Σ,q_0,F הם בדיוק כמו בהגדרת DFA אבל כאן:

$$\delta: Q\times\Sigma\to 2^Q$$

כאשר $2^Q=\mathcal{P}(Q)=\left\{\mathcal{P}\middle|\mathcal{P}\subseteq Q\right\}$ - קבוצת החזקה של Q.
כאן, בניגוד לאוטומט סופי דטרמיניסטי לכל מצב מותאמת קבוצה של מצבים.
עבור $q\in Q$ ו־$a\in\Sigma$: $\delta(q,a)\subseteq Q$, כלומר, $\delta(q,a)$ היא קבוצה של מצבים.
כשהאוטומט נמצא נמצא ב־q וקורא את a הוא יכול לעבור לכל אחד מהמצבים ב־$\delta(q,a)$ [האוטומט יכול לנחש את המשך החישוב].
ייתכן ש־ $\delta(q,a)=\emptyset$ - זה אומר שאם האוטומט נמצא ב־q וקורא a הוא נתקע ולא יכול להמשיך בחישוב.
למשל, הנה אסל"ד עבור $\Sigma=\{0,1\}$ כאשר השפה היא כל המילים שבהן האות האחת לפני אחרונה היא 1:



בגרף המייצג את האוטומט ממצב Q יכולים לצאת מספר קשתות (ייתכנו גם 0 קשתות) שמסומנות באותה אות קלט.
למשל:



משמעות הציור:
כשהאוטומט נמצא במצב q וקורא את a הוא יכול באופן לא דטרמיניסטי לעבור ל־pאו ל־r.
ייתכן שמ־q לא יוצאת קשת שמסומנת ב־a. במקרה כזה, אם האוטומט נמצא ב־q וקורא את a - האוטומט נתקע (ב־q) ולא יכול להמשיך בריצה שלו.
במקרה זה האוטומט **דוחה** את המילה **אפילו אם q הוא מצב סופי (מקבל).**

ריצה (או: מסלול חישוב) של A על מילה $w=a_1a_2\cdots a_n$ מ־Σ^* ($a_i\in\Sigma$) היא סדרה מצבים: $q_0,q_1,...,q_n$ (אורך סדרת המצבים: $|w|+1$) שמקיימת:
q_0 הוא המצב ההתחלתי ולכל $1\le i\le n$: $\boldsymbol{q_i\in\delta(q_{i-1},a_i)}$.
כמובן של־A יש מספר ריצות אפשריות על מילת קלט w. מספר זה יכול להיות 0.
[כלומר, על מילת קלט מסויימת האוטומט יכול להיתקע לפני שסיים את קריאת המילה].

אומרים שאסל"ד A **מקבל** את מילת הקלט w (כאשר $|w|=n$) אם קיימת ריצה $q_0,...,q_n$ של A על w כך ש־$q_n\in F$ (כלומר, ריצה של n פעמים שמסתיימת במצב סופי).
השפה של אסל"ד A מסומנת $L(A)$ומוגדרת להיות:

$$\boxed{\boldsymbol{L(A)=\left\{w\in\Sigma^*\middle|\text{⌽}\right\}}}$$

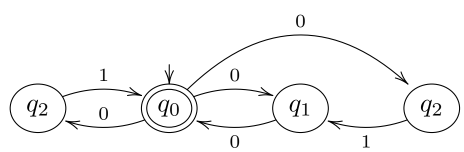
⌽ - A מקבל את w.

> כדאי גם לזכור את הכלל הבא: ייתכן שריצה על מילה בשפה תסתיים במצב לא מקבל או שהאוטומט יתקע, **אבל לא ייתכן שנריץ על האוטומט מילה שאינה בשפה והיא תסתיים, בריצה כלשהי, במצב מקבל** (אלא אם האוטומט נתקע ואז זה נחשב שהאוטומט דוחה את המילה).

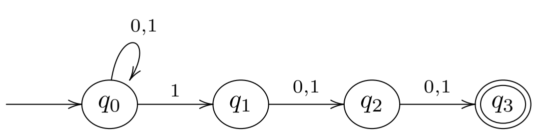
הערה 2.1 DFA הוא מקרה פרטי של NFA. DFA הוא בעצם NFA שבו $|\delta(q,a)|=1$ לכל $a\in\Sigma$ ולכל $q\in Q$.

2.1 דוגמאות לכמה אוטומטים מסוג NFA

דוגמא ראשונה:
כל האוטומטים בדוגמאות יהיו מעל הא"ב $\Sigma=\{0,1\}$
אוטומט שמזהה את השפה הבאה: L^* כאשר $L=\{00,01,010\}$



דוגמא שניה:
אוטומט שמזהה את כל המילים בשפה שבהן האותה השלישית מהסוף היא 1:



עבור קבוצת מצבים $P\subseteq Q$ ועבור $a\in\Sigma$ נגדיר:

$$\delta(P,a)=\bigcup_{q\in P}\delta(q,a)$$

(תזכורת: $\delta(q,a)$ היא קבוצה של מצבים לכל $q\in Q$ ולכל $a\in\Sigma$).
$\delta(P,a)$ = קבוצת המצבים שאליהם יכול האוטומט להגיע אם הוא נמצא באחד המצבים ב־P וקורא a.
למשל, בדוגמא השנייה:

$$\delta(\{q_0,q_1\},0)=\{q_0,q_2\}$$
$$\delta(\{q_0,q_1\},1)=\{q_0,q_1,q_2\}$$
$$\delta(\{q_3\},0)=\varnothing$$
$$\delta(\{q_1,q_3\},1)=\{q_2\}$$

ניתן להרחיב את δ גם עבור מילה: $\delta : Q \times \Sigma^* \to 2^Q$, ולכן $\delta(q,w)$ מוגדרת לכל $q \in Q$ ולכל $w \in \Sigma^*$:

$\delta(q,w)$ = קבוצת כל המצבים שהאוטומט יכול להגיע אליהם אם הוא מתחיל במצב q ומנסה לקרוא את w.

ייתכן ש־$\delta(q,w) = \varnothing$. זה קורה כאשר כל הריצות של aעל w נתקעות.

(כמו־כן, באותו אופן, ניתן להגדיר $\delta(P,w)$ עבור $P \subseteq Q$ ו־$w \in \Sigma^*$).

2.2 השפה של אוטומט סופי לא־דטמינסטי A

השאפה של אסל"ד A מוגדרת כך:

$$\boxed{L(A) = \left\{ w \in \Sigma^* \middle| \delta(q_0,w) \cap F \neq \varnothing \right\}}$$

במילים אחרות: עבור $w \in \Sigma^*$:

A מקבלת את $w \iff \delta(q_0,w) \cap F \neq \varnothing$.

2.3 אוטומט חזקה

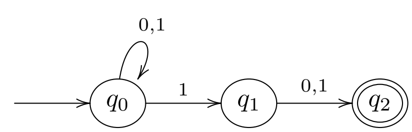
טענה 2.2 לכל אסל"ד A, קיים אס"ד B כך ש־$L(A) = L(B)$.

(אני לא אוכיח את הטענה, אלא רק אסביר איך בונים אוטומט כזה באמצעות דוגמא)

רק חשוב לזכור שכל מצב ב־B (האס"ד שנאחנו בונים) הוא בעצם קבוצת מצבים מ־A (יובהר בהמשך),

והמצב ההתחלתי של B הוא $\{q_0\}$.

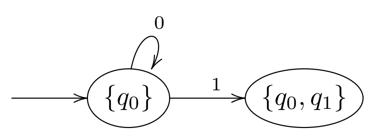
נניח ונתון לנו האוטומט הבא (מעל $\Sigma = \{0,1\}$):



אזי, בשביל לבנות את ה־DFA שלנו נעשה את הדבר הבא (וזה נכון לכל אוטומט NFA)

נתחיל מ־$\{q_0\}$ ונראה לאיזו קבוצת מצבים אנחנו נגיע אם נשלח לו כל אחת מהאותיות ב־Σ[4]:

$\delta(\{q_0\},0) = \{q_0\}, \ \delta(\{q_0\},1) = \{q_0,q_1\}$ ולכן, כך מתחילה הבניה:

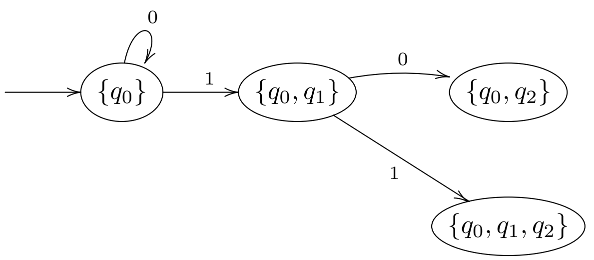


כעת, היות ו־0 מוביל אותנו אל q_0 אזי אין לנו מה לבדוק עבורו, אבל 1 הוביל אותנו ל־$\{q_0,q_1\}$ ולכן נבדוק כעת לאן קבוצת המצבים הזאת מובילה אותנו:

$\delta(\{q_0,q_1\},0) = \{q_0,q_2\}, \ \delta(\{q_0,q_1\},1) = \{q_0,q_1,q_2\}$

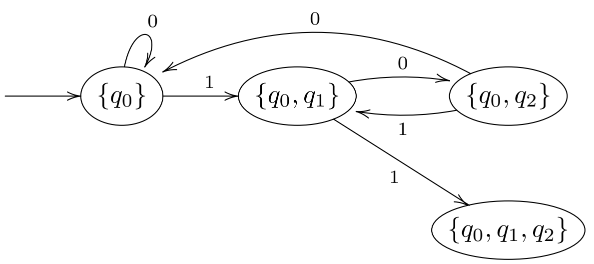
[4] מקודם הוסבר איך לחשב את זה: $\delta(P,a) = \bigcup_{q\in P} \delta(q,a)$.

ולכן הגרף יראה עכשיו:



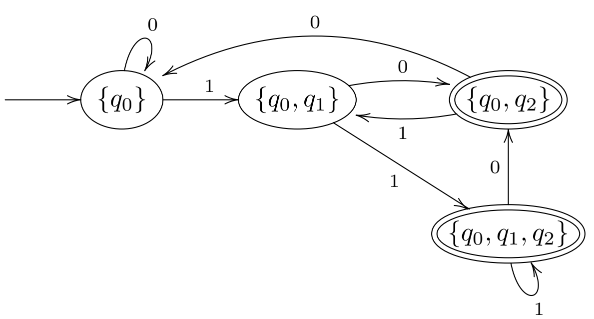
כעת נצטרך לבחור את אחד המצבים שעוד לא בחרנו ולבדוק לאיזו קבוצת מצבים כל אות בא"ב שולחת אותנו עבור אותו מצב (שהוא בעצם קבוצת מצבים):

$\delta(\{q_0,q_2\},0) = \{q_0\}, \ \delta(\{q_0,q_2\},1) = \{q_0,q_1\}$, כעת, היות וכבר יש לנו את המצבים הנ"ל בגרף, פשוט נמתח אליהם חצים:



כעת, נותר עוד מצב אחד שאותו לא בדקנו ונבדוק לאיזה קבוצת מצבים כל אות מהא"ב מביאה אותנו:

$\delta(\{q_0,q_1,q_2\},0) = \{q_0,q_2\}, \ \delta(\{q_0,q_1,q_2\},1) = \{q_0,q_1,q_2\}$ ולכן הגרף הוא:



כעת נוכל לראות שאין יותר מצבים לבדוק, כלומר שמצבים אחרים שלא נמצאים כאן בגרף אינם ניתנים להשגה (כאלה שלא נוכל להגיע אליהם לעולם מ־q_0) ולכן אין לנו סיבה לבדוק אותם ולרשום אותם בגרף.

הסבר למצבים המקבלים: היות ובגרף המקורי המצב המקבל היחיד הוא q_2 אזי כל קודקוד בגרף החדש שאחד מאיבריו הוא q_2 יהיה מצב מקבל. מה שאנחנו עושים זה שאנחנו הולכים לפי הכלל הבא: לכל מצב q בגרף החדש, אם $q \cap F \neq \varnothing$ אזי q הוא מצב מקבל...

ה־DFA B שבנינו נקרא **אוטומט החזקה** של NFA A. כעת, מהטענה האחרונה ומהעובדה ש־DFA הוא מקרה פרטי ל־NFA מתקבל משפט:

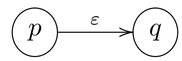
משפט 2.3 תהי $L \subseteq \Sigma^*$ שפה אזי:

L רגולרית $\iff$ קיים NFA A כך ש־$L(A) = L$.

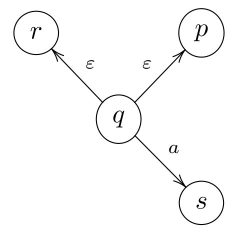
3 אוטומט סופי עם מסעי (מעברי) ε

3.1 הגדרה

אוטומט סופי עם מסעי ε זהו NFA שבנוסף להיותו NFA יש בגרף המייצג אותו קשתות המסומנות ב־ε (המילה הריקה):

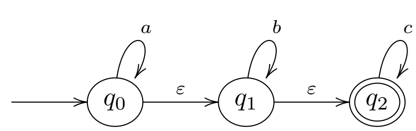


משמעות קשת (מעבר) כזה היא: שאם האוטומט נמצא ב־q הוא יכול לקפוץ למצב p מבלי לקרוא (או: "לצרוך") אות מהקלט.
למשל:



אם האוטומט נמצא ב־q וקורא a אזי האפשרויות הן:
ללכת ל־s ולעבור לאות הבאה, או: ללכת ל־p או r ולקרוא את a משם (כלומר, <u>קודם</u> לעשות את הקפיצה של ε ואז לקרוא את האות, ולא ההפך!).
כל שאר הפרטים במודל זה - זהים ל־NFA.

3.1.1 דוגמא



שפת אוטומט זה היא: $a^*b^*c^* = \left\{a^lb^mc^n \middle| l,m,n\geq 0\right\}$

3.2 הגדרה פורמלית

סימון: אם Σ הוא א"ב אזי נסמן: $\Sigma_\varepsilon = \Sigma\cup\{\varepsilon\}$.[5]
אוטומט אל דטרמיניסטי עם מסעי ε, בסימון εNFA זוהי חמישיה:

$$A = \langle Q,\Sigma,\delta,q_0,F\rangle$$

[5] $\{\varepsilon\}$ - זאת אינה אות, אלא המילה הריקה!

כאשר: Q,Σ,q_0,F הם בדיוק כמו בהגדרת NFA, אבל פה:

$$\delta : Q\times\Sigma_\varepsilon\to 2^Q$$

(ההבדל הוא שכאן מוגדרת גם $\delta(q,\varepsilon)$ עבור $q\in Q$. $\delta(q,\varepsilon)\subseteq Q$. ייתכן ש־$\delta(q,\varepsilon)=\varnothing$, כלומר אין מעברי ε שיוצאים מ־q.
$p\in\delta(q,\varepsilon)$ $\iff$ יש בגרף[6] מעבר כזה: $q \xrightarrow{\varepsilon} p$.

3.3 הרחבת δ למילים

3.3.1 סגור ε של q

עבור מצב $q\in Q$ נגדיר **סגור של ε** באופן הבא:
<u>$\mathrm{CL}_\varepsilon(q)$</u> = קבוצת כל המצבים ב־A שהאוטומט יכול להגיע אליהם ללא קריאת שום קלט, כלומר, ע"י שימוש במעברי ε.
למשל, בדוגמא שלמעלה: $\mathrm{CL}_\varepsilon(q_0)=\{q_0,q_1,q_2\}$,
$\mathrm{CL}_\varepsilon(q_1)=\{q_1,q_2\}$ ו־$\mathrm{CL}_\varepsilon(q_2)=\{q_2\}$.
עבור קבוצת מצבים $P\subseteq Q$ מסמנים:

$$\mathrm{CL}_\varepsilon(P)=\bigcup_{q\in P}\mathrm{CL}_\varepsilon(q)$$

כלומר - כל המצבים שניתן להגיע אליהם מאיזשהו $q\in P$ <u>ללא</u> קריאת אות קלט.
<u>הרחבת δ:</u>

$$\hat{\delta} : Q\times\Sigma^*\to 2^Q$$

מוגדרת כך:

$$\begin{cases}\hat{\delta}(q,\varepsilon)=\mathrm{CL}_\varepsilon(q)\\ \hat{\delta}(q,va)=\bigcup_{r\in\hat{\delta}(q,v)}\mathrm{CL}_\varepsilon(\delta(r,a)) & \forall v\in\Sigma^*, a\in\Sigma\end{cases}$$

כך $\hat{\delta}(q,w)$ מוגדרת לכל $q\in Q$ ולכל $w\in\Sigma^*$.
$\hat{\delta}(q,w)$ = קבוצת כל המצבים שניתן להגיע אליהם אם מתחילים במצב q וקוראים את w (מילה).
חשוב לשים לב לדבר הבא: <u>$\hat{\delta}(q,a)\neq\delta(q,a)$</u>! (עבור $a\in\Sigma$ ו־$q\in Q$).
הסבר: ב־$\hat{\delta}(q,a)$, ע"פ ההגדרה, מותר לנו להשתמש במעברי ε כי אנחנו מכללים את ε כחלק מהאפשרויות למעבר. לעומת זאת, ב־$\delta(q,a)$ אנחנו לא יכולים להשתמש במעברי ε, כי אנחנו מתייחסים כאן לכל המצבים שניתן להגיע אליהם מ־q ע"י קריאת <u>האות</u> a (ולא המילה כמו ב־$\hat{\delta}$). כזכור גם $\delta(q,\varepsilon)$ מוגדרת כאן.

3.4 הגדרת השפה של εNFA

השפה של A εNFA מוגדרת ע"י:

$$L(A)=\left\{w\in\Sigma^*\middle|\hat{\delta}(q_0,w)\cap F\neq\varnothing\right\}$$

[6] המייצג את האוטומט A.

3.5 בניית NFA מ־εNFA

בהינתן εNFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ נבנה NFA ללא מסעי ε: $B = \langle Q, \Sigma, \delta', q_0, F' \rangle$ באופן הבא:

$$\delta' : Q \times \Sigma \to 2^Q$$
$$\delta' (q, a) = \hat{\delta} (q, a)$$

כעת, לגבי F', נשים לב האם האוטומט A מקבל את ε (המילה הריקה):

$$F' = \begin{cases} F \cup \{q_0\} & \mathrm{CL}_\varepsilon \cap F \neq \varnothing \ (\iff \varepsilon \in L(A)) \\ F & \mathrm{CL}_\varepsilon \cap F = \varnothing \ (\iff \varepsilon \notin L(A)) \end{cases}$$

עבור מילה $\varepsilon \neq w \in \Sigma^*$: $\delta' (q, w) = \hat{\delta} (q, w)$ (עבור המילה הריקה ε זה נכון רק עבור q_0)
מכאן נובע ש־ $w \in L(A) \iff w \in L(B)$ (וכלל זה תקף גם עבור המילה הריקה).
כלומר, מתקיים: $L(A) = L(B)$.

הערה 3.1 נשים לב שב־A וב־B יש את אותם מהצבים ואותו מצב התחלתי.

משפט 3.2 לכל εNFA A קיים NFA (ללא מסעי ε) B, עם אותם מצבים ואותו מצב התחלתי שמקיים: $L(A) = L(B)$.

משפט 3.3 שפה $L \subseteq \Sigma^*$ היא רגולרית $\iff$ קיים εNFA A שמזהה את L.

המעבר מ־εNFA ל־NFA (ללא מסעי ε) שקול, שמתואר בבניה הנ"ל נקרא: **הסרת מעברי (מסעי) ε.**

ניתן גם לעשות את המעבר באופן הבא: עבור כל מצב ב־A (εNFA) נתסכל על

3.5.1 תיאור תהליך הסרת מעברי ε

נתון εNFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$.
נסיר את מעברי (מסעי) ε באופן הבא:

1. עבור כל זוג מצבים (q, p), אם בגרף A יש מסלול מ־q ל־p **שכל** הקשתות בו מסומנות ב־ε, **מלבד אחת** המסומנת באות a, אזי מוסיפים קשת $q \xrightarrow{a} p$ לגרף (במידה וקשת כזאת אינה קיימת).
2. מוחקים את כל הקשתות שמסומנות ב־ε.
3. במידת הצורך משנים את F: אם $F \cap \mathrm{CL}_\varepsilon (q_0) \neq \varnothing$ אזי מוסיפים את q_0 ל־F.

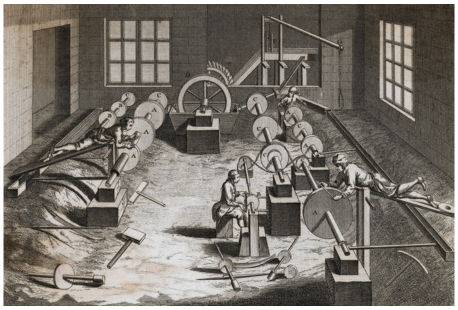
הערה: יכול להיות שאם נסיר את מעברי ε באמצעות האלגוריתם יצא לנו NFA שונה מזה שנעשה אם נסיר ע"פ ההגדרה. אבל זה בסדר כל עוד ה־DFA שלנו הוא אותו דבר. **לכן בכל מקרה תמיד כדאי לבדוק (אפילו חלקית ־ כי ברגע שאין התאמה אז אפשר להפסיק).**

3.6 פעולות רגולריות על שפות

פעולות רגולריות על שפות: איחוד (ולא חיתוך), שירשור ואיטרציה.
משפט: משפחת השפות הרגולריות סגורה לפעולות רגולריות.[7]

[7] הוכחה בדף נפרד.

חלק III
ביטויים רגולריים



4 הגדרה

יהי Σ א"ב.
ביטוי רגולרי (מעל Σ) מוגדר ברקורסיה באופן הבא:

1. לכל $a \in \Sigma$: a הוא ביטוי רגולרי.
2. ε (המילה הריקה) היא ביטוי רגולרי.
3. $\varnothing$ היא ביטורי רגולרי.
4. אם r ו־s הם ביטויים רגולריים אזי גם: $r + s, rs, r^*$ הוא ביטוי רגולרי.

למשל: 0, $(0+1)^*$ [כל המילים ב־Σ, שקול ל־Σ^*], $(1+0)^* 1$ [כל המילים שמסתיימות ב־1],....

השפה של ביטוי רגולרי r מסומנת $\boldsymbol{L(r)}$ ומוגדרת רקורסיבית באופן הבא:

1. $L(a) = \{a\}$ לכל $a \in \Sigma$.
2. $L(\varepsilon) = \{\varepsilon\}$.
3. $L(\varnothing) = \varnothing$.
4. אם r, s הם ביטויים רגולריים, אזי:
   (א) $L(r+s) = L(r) \cup L(s)$.
   (ב) $L(rs) = L(r) \cdot L(s)$.
   (ג) $L(r^*) = L(r)^*$

ביטויים רגולריים r, s שמקיימים $L(r) = L(s)$ נקראים **ביטויים רגולריים שקולים.**
למשל: $r = (0^*1)^*,\ s = \varepsilon + (0+1)^* 1$ ־ הם שקולים ־ ניתן לייצר מהם את אותן מילים.

הערה 4.1 נשמיט את ההבחנה בין ביטוי רגולרי לבין השפה שלו ונתייחס לביטוי רגולרי כמו לשפה שאותה הוא מגדיר.
על־כן, נרשום שקילות של ביטויים רגולריים $r = s$.

4.1 סדר פעולות

סדר הפעולות בביטויים רגולריים (כלומר, מה קודם למה מבחינת החשיבות בסדר יורד):

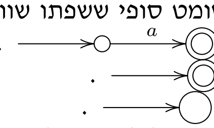
1. () - סוגריים.
2. $*$.
3. $\cdot$ (שירשור).
4. $+$.

5 משפט Kleene

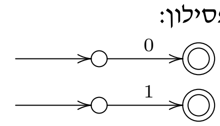
יהי Σ א"ב.
שפה $L \subseteq \Sigma^*$ היא שפה רגולרית $\iff$ קיים ביטוי רוגלרי r כך ש־$L(r) = L$.
(במילים אחרות, ביטוי רגולרי שקול לאוטומט סופי).

5.1 הפיכת ביטוי רגולרי לאוטומט

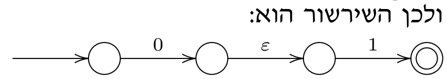
אוטומט סופי ששפתו שווה לשפת ביטוי רגולרי אטומי:



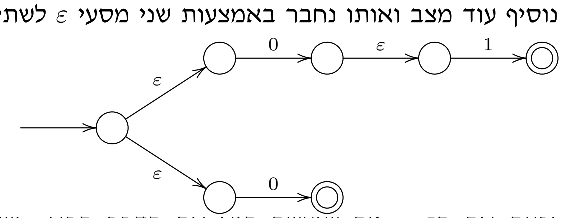
כעת, נוכל להפוך כל ביטוי רגולרי לאוטומט סופי:
למשל: ניקח את הביטוי הרגולרי $(01+0)^*$ ונהפוך אותו לאוטומט סופי:
נתחיל מליצור את השפה 01 ע"י שירשור השפות 0 ו־1 עם אפסילון:



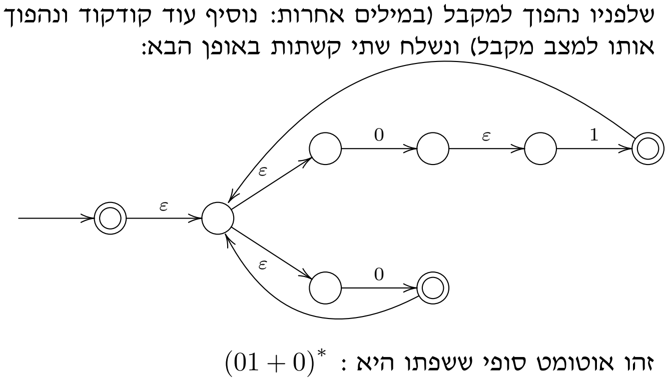
ולכן השירשור הוא:



כעת, את "+" נבטא באופן הבא:
נוסיף עוד מצב ואותו נחבר באמצעות שני מסעי ε לשתי השפות:



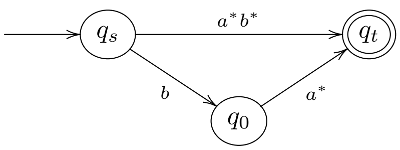
וכעת את ה־$*$. מה שנעשה הוא את הדבר הבא: נשים מעבר ε לקודקוד שנמצא אחרי מעבר ה־ε שמוביל לשפה ואת המצב שלפניו נהפוך למקבל (במילים אחרות: נוסיף עוד קודקוד ונהפוך אותו למצב מקבל) ונשלח שתי קשתות באופן הבא:



זהו אוטומט סופי ששפתו היא : $(01+0)^*$

5.2 הפיכת אוטומט DFA לביטוי רגולרי

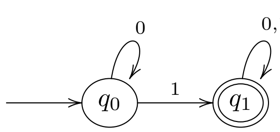
כדי לתאר הפיכת DFA לביטוי רגולרי נציג מודול חישובי חדש שנקרא <u>**GNFA**</u> (אוטומט מוכלל, G=Generalized).
$GNFA$ מוצג ע"י גרף מכוון, כמו NFA רק שהקשתות בו מסומנות בביטוים רגולרים .
דבר נוסף: יש מצב סופי יחיד - q_t, ומצב התחלתי יחיד q_s.
<u>אין</u> קשת שנכנסת ל־q_s <u>ואין</u> קצת שיוצאת מ־q_t.
למשל:



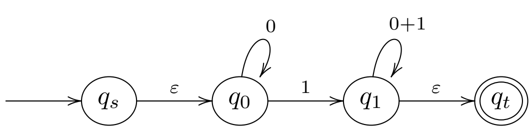
הסברים:
אם: $x \xrightarrow{r} y$ היא קשת ב־GNFA, אזי האוטומט יעבור מ־x ל־y אם הוא קורא מילה w שהיא משפת ביטוי רגולרי r, $L(r)$ - כלומר הוא יעבור מ־x ל־y אם הוא יקרא את הביטוי הרגולרי r.
GNFA אשר מקבל מילה w, יכול לעבור מ־q_s ל־q_t ע"י קריאת w.
שפת האוטומט היא קבוצת כל המילים שהאוטומט מקבל, דהיינו כל $w \in \Sigma^*$ כך ש־$\delta(q_s, w) = q_t$.
למשל, שפת ה־GNFA שלמעלה הינה: $\boldsymbol{a^*b^* + ba^*}$.
(כי בשביל להגיע מ־q_s ל־q_t אנחנו צריכים או a^*b^* או b ואז a^*).

5.2.1 הפיכת אוטומט DFA ל־GNFA

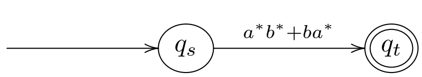
ניתן להפוך כל A DFA ל־GNFA בדרך הבאה:
נהפוך את כל המצבים המקבלים ב־A למצבים לא מקבלים ואת המצב ההתחלתי למצב לא התחלתי.
נוסיף שני מצבים חדשים:
q_s - מצב התחלתי.
q_t - מצב סופי.
נוסיף קשת מסומנת ב־ε מ־q_s למצב שהיה התחלתי ב־A ונוסיף קשת שמסומנת ב־ε מכל מבצ שהיה מצב סופי ב־A ל־q_t.
למשל: נתון לנו ה־DFA הבא:



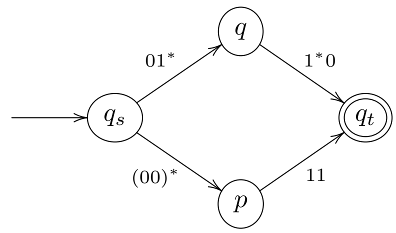
אזי אם נהפוך אותו ל־GNFA נקבל:



הערה: בדוגמא הראשונה של ה־GNFA נוכל להשמיט את q_0 ולקבל GNFA קטן יותר שמקבל את אותה השפה:



דוגמא נוספת ל־GNFA:



שפת אוטומט זה:

$$01^*1^*0 + (00)^*\,11 = 01^*0 + (00)^*\,11$$

5.2.2 הגדרה פורמלית ל־GNFA

GNFA זוהי חמישייה $A = \langle Q, \Sigma, \lambda, q_s, q_t\rangle$ כאשר:

- Q = קבוצה סופית של מצבים $|Q| \geq 2$.
- Σ = א"ב.
- $q_s \in Q$ ־ יקרא המצב ההתחלתי.
- $q_t \in Q$ ־ יקרא המצב המקבל.
- λ מוגדרת באופן הבא:

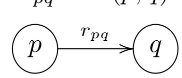
$$\lambda : (Q - \{q_t\}) \times (Q - \{q_s\}) \to R_\Sigma$$

כאשר R_Σ = קבוצת הביטויים הרוגלרים מעל Σ.

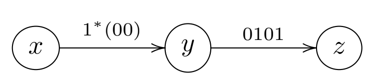
5.2.3 סימון קשתות ב־GNFA

לכל סוג סדור של מצבים (p,q) (מלבד (q_s,q_t)) ־ $\lambda(p,q)$ הוא ביטוי רגולרי. זהו הביטוי הרגולרי שמסמל את הקשת (או את הצלע) מ־p ל־q.
נסמן ביטוי זה ב־r_{pq}:

$$r_{pq} = \lambda(p,q)$$



ייתכן כמובן ש־ $\lambda(p,q) = r_{pq} = \varnothing$.
למשל, בדוגמא שלמעלה: $r_{q_sq} = 01^*, r_{qp} = \varnothing$
במקרה זה משמיטים את הצלע (אם ישנה).
עוד דוגמא:



במקרה הזה: $r_{xy} = 1^*00, r_{yz} = 0101, r_{zy} = \varnothing$.
ו־ $\boldsymbol{r_{xz} = \varnothing}$ ־ כי **אין קשת** (צלע) מ־x ל־z.

יהי A GNFA ותהי $w \in \Sigma^*$. אומרים ש־A מקבל את w אם קיימת סדרת מצבים ב־A: $q_s, q_1, q_2, ..., q_k, q_t$ (לשם פשטות, נסמן: $q_s = q_0$, ו־ $q_t = q_{k+1}$)
וקיים פירוק של w מהצורה: $w = w_1 w_2 \cdots w_{k+1}$ כך שלכל $1 \leq i \leq k+1$ מתקיים: $w_i \in \lambda(q_{i-1}, q_i)$ כלומר:

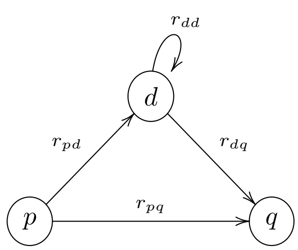
$$w_i \in r_{q_{i-1}q_i}$$

שפת האוטומט L היא קבוצת על המילים $w \in \Sigma^*$ ש־A מקבל.

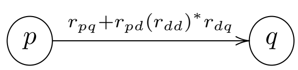
5.2.4 צימצום של GNFA

בההינתן GNFA $A = \langle Q, \Sigma, \lambda, q_s, q_t\rangle$ עם $|Q| > 2$ נוכל להקטין את מספר המצבים ב־1 ולהישאר עם GNFA A' שקול ל־A באופן הבא:

1. נבחר מצב $d \in Q - \{q_s, q_t\}$ (כלומר, $d \neq q_s, q_t$).
2. לכל זוג סדור של מצבים (p,q) מהמצבים שנשארו נעדכן את r_{pq}:



$$\Downarrow$$



כלומר, מה שאנחנו הוא מעדכנים פה הוא את r_{pq}:

$$\boxed{r_{pq} \leftarrow r_{pq} + r_{pd}\,(r_{dd})^*\,r_{dq}}$$

.

3. נשמיט מ־A את d ואת כל שאר הקשתות שנכנסות או יוצאות מ־d.

חשוב לזכור שמדובר בצעלות, כלומר, r_{pq} אינה מסילה בין p ל־q אלא קשת בין בני קודקודים.
תזכורת: אם אחד מהבאים היא קבוצה ריקה: r_{pd}, r_{dq} אזי: $r_{pd}\,(r_{dd})^*\,r_{dq} = \varnothing$.[8]

[8] אם אין ב־r_{dd} כלום (כלומר, אין מעגל), אזי נאמר ש־$r_{dd} = \varepsilon$.

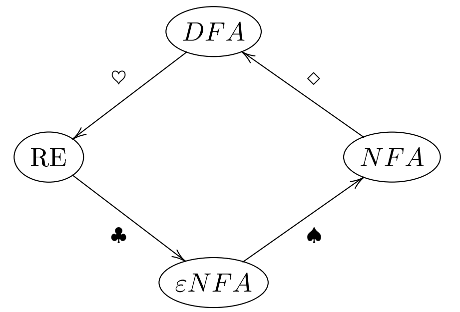
5.2.5 אלגוריתם הפיכת DFA לביטוי רגולרי

<u>קלט</u>: DFA A.
<u>פלט</u>: ביטוי רגולרי r כך ש־$L(r) = L(A)$.

1. הפוך את A ל־GNFA B.
2. כל עוד מספר המצבים ב־B גדול מ־2 ־ צמצם את B במצב אחד.
3. החזר את הביטוי הרגולרי $r_{q_s q_t}$ כפלט.

6 סיכום קצר

ראינו ארבעה סוגים שונים של ייצוג של שפות רגולריות:



♣ ־ הפיכת ביטוי רגולרי לאוטומט (סעיף 5.1).
♠ ־ הסרת מסעי ε (סעיף 3.5).
◇ ־ אוטומט החזקה (סעיף 2.3).
♡ ־ הפיכת אוטומט לביטוי רגולרי (סעיף 5.2).

7 למת הניפוח

7.1 הגדרה (ניסוח הלמה)

תהי $L \subseteq \Sigma^*$ שפה <u>רגולרית</u>.
אזי: קיים מספר טבעי n כך ש<u>לכל</u> מילה $w \in L$ עם $|w| \geq n$ קיים פירוק $w = xyz$ כאשר $x, y, z \in \Sigma^*$ ומתקיימות הדרישות הבאות:

1. $|xy| \leq n$.
2. $|y| > 0$ (כלומר: $y \neq \varepsilon$).
3. לכל $i \geq 0$ מתקיים: $xy^iz \in L$ ־ כלומר, ניתן לנפח את המילה.

הערה: למת הניפוח הינה תנאי הכרחי לשפה רגלורית, כלומר, כל שפה רגולרית מקיימת את תנאי הלמה, אבל לא כל שפה שמקיימת את תנאי הלמה היא רגולרית.
במילים אחרות ־ אם שפה מקיימת את תנאי הלמה ־ היא אינה בהכרח רגולרית.
תנאים אלו נקראים **תנאי הניפוח** (החל מהמילה "קיים" עד לתנאי 3).
הלמה אומרת שאם שפה L היא רגולרית אזי L מקיימת את תנאי הניפוח.
המספר n בתנאי הניפוח נקרא **קבוע הניפוח**.
ניתן להניח בה"כ ש־n שווה או גדול שווה למספר המצבים ב־DFA שמזהה את L.

השימוש העיקרי של הלמה הוא להראות אי־רגולריות של שפה L (ע"י כך שמראים ש־L לא יכולה לקיים את תנאי הניפוח, בד"כ זה יהיה התנאי השלישי).

7.2 דוגמא קצרה לשימוש בלמת הניפוח

נוכיח כי השפה:

$$L = \left\{ a^j b^j \middle| i = j,\ \ i, j \geq 0 \right\}$$

אינה רגולרית.
הוכחה:
נניח בשלילה כי השפה L אכן רגולרית, אזי קיים קבוע ניפוח $n \in \mathbb{N}$ כך שלכל $w \in L$ כך ש־$|w| \geq n$ מתקיימים שלושת התנאים, הנ"ל.
כעת, כל מה שעלינו למצוא זאת מילה אחת שלא תעמוד בתנאים. ברגע שמצאנו מילה כזאת, אזי הלמה אינה נכונה, כלומר, השפה אינה רגולרית. אבל בגלל שאנחנו לא יודעים מהו n נצטרך לכתוב את המילה בצורה תבניתית.
נקח את המילה: $w = a^n b^n$ ואכן $w \in L$ ע"פ הגדרת השפה ו־$|w| \geq n$, ולכן אם היא רגולרית היא צריכה לקיים את תנאי הניפוח.
לפי הלמה, ישנו פירוק $w = xyz$ כאשר $x, y, z \in \Sigma^*$ ומתקיים:

1. $|xy| \leq n$.
2. $|y| > 0$.
3. לכל $0 \leq i \in \mathbb{N}$ מתקיים: $xy^iz \in L$.

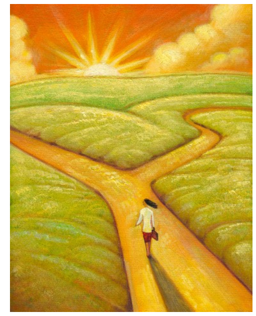
מ־1 ו־2 נקבל שקיימת חלוקה: $x = a^l, y = a^m, z = a^{n-l-m}b^n$ (נשים לב שאין אנחנו יכולים לבחור איך לחלק את המילה שבחרנו. החלוקה נובעת מתוך הלמה, ולכן אנחנו שמים משתנים).
כשאר (ע"פ הלמה): $l + m \leq n$ ו־$m > 0$. (יכול להיות ש־$l = 0, m = 1$ ואז, מלבד a אחד, כל ה־a־ים נמצאים ב־z).
כעת, אם נבחר $i = 0$ מה שנקבל הוא ש־$xz \in L$, אבל אם נשים לב: $xz = a^l \cdot a^{n-l-m} \cdot b^n = a^{n-m}b^n$.
אבל בגלל ש־$m > 0$ אזי $n - m > n$ מה שאומר ש־$xz \notin L$ והגענו לסתירה.
מכאן ש־L אינה רגורלית.
כאשר רוצים לסתור את זה ששפה היא רגולרית באמצעות הלמה מה שנעשה הוא שתמיד, כמו בדוגמא כאן, נחפש את ה־i בתנאי השלישי שיוציא את המילה מ־L וכך נראה כי היא אינה רגולרית. לכן האתגר כאן הוא בעצם: א. למצוא מילה שנוכל באמצעותה לסתור את רגולריות השפה. ב. למצוא את ה־i המתאים שעבורנו נראה כי המילה שבחרנו לא נמצאת בשפה (עבור i מסוים).

8 למה נוספת

יהי $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ אוטומט DFA עם n מצבים ($|Q| = n$) ותהי $L = L(A)$.
אזי:

1. $L \neq \varnothing \iff L$ מכילה מילה מאורך $n >$ ($|w| < n$).
2. L אינסופית $\iff L$ מכילה מילה w כך ש־$n \leq |w| \leq 2n$.

חלק IV
בעיות הכרעה לשפות רגולריות



בעיית הכרעה: בעיה שהתשובה עליה היא "כן" או "לא".
נציג שפות רגולריות ע"י DFA ונדון בבעיות ההכרעה הבאות:

1. בעיית השייכות: נתון DFA A ומילה w (מא"ב Σ של A), האם $w \in L(A)$?
2. בעיית הריקנות: נתון DFA A, האם $L(A) = \varnothing$?
3. בעיית הסופיות: נתון DFA, האם $L(A)$ אינסופית?
4. בעיית השקילות: נתונים שני DFA-ים A_1 ו-A_2 האם $L(A_1) = L(A_2)$?

9 פתרונות הבעיות שהוצגו

9.1 פתרון בעיית השייכות

קלט: DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ ומילה $w = a_1 a_2 \cdots a_n \in \Sigma^*$.
פלט: "כן" אם $w \in L(A)$, "לא" אחרת.

1. $q = q_0$.
2. עבור $i = 1, 2, ..., n$: $\delta(q, a_i) \to q$.
3. אם $q \in F$ החזר "כן. אחרת החזר "לא".

9.2 פתרון בעיית הריקנות

בודקים אם A מקבל איזושהי מילה w כאשר $|w| < n$ ו-$n = |Q|$
.
אם הוא מקבל, אזי מחזירים "כן", אחרת מחזירים "לא".

9.3 פתרון בעיית הסופיות

בודקים אם A מקבל מילה w כאשר $n \le |w| \le 2n$.
אם כן - מחזירים "כן", אם לא, מחזירים "לא".
(נכונות שני האלגוריתמים האחרונים נובעת מהלמה האחרונה שהראנו).

9.4 פתרונות יעלים לאלגוריתמים הנ"ל

נציג פתרונות יעלים לבעיות שהוצגו. לשם כך נשתמש ב-BFS.
ונכיר עוד שני מושגים חשובים: $R_0(q)$ ו-$R(q)$ כאשר $q \in Q$.

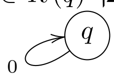
9.4.1 $R_0(q)$ ו-$R(q)$

יהי DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, עבור $q \in Q$ נסמן:

$$\boldsymbol{R_0(q) = \left\{ \delta(q, w) \,\middle|\, w \in \Sigma^* \right\}}$$

זוהי קבוצת כל המצבים שניתן להגיע אליהם מ-q באמצעות מילה כלשהי $w \in \Sigma^*$ (שזה כמובן כולל גם את המילה הריקה ולכן תמיד $q \in R_0(q)$), לכן, אם נתון לנו ש-$p \in R_0(q)$ אזי קיימת מילה $w \in \Sigma^*$ כך ש-$\delta(q, w) = p$.

$$\boldsymbol{R(q) = \left\{ \delta(q, w) \,\middle|\, w \in \Sigma^*,\ |w| > 0 \right\}}$$

זוהי קבוצת כל המצבים ב-Q שניתן להגיע אליהם מ-q אבל עם כל מילה מלבד המילה הריקה, כלומר, אם $p \in R(q)$ אזי קיימת מילה $\varepsilon \neq w \in \Sigma^*$ כך שמתקיים: $\delta(q, w) = p$.
חשוב לזכור כי ייתכן $q \in R(q)$, כי קיים מעגל מאורך חיובי[9] מ-q לעצמו. למשל:  אזי עבור $w = 0$ נגיע בחזרה ל-q.

$p \in R_0(q) \iff$ יש מסלול מ-q ל-p.
$p \in R(q) \iff$ יש מסלול מאורך חיובי מ-q ל-p.
תמיד נכון: $R_0(q) = R(q) \cup \{q\}$.
לא תמיד נכון: $R(q) = R_0(q) - \{q\}$ (במקרה שיש מעגל מאורך חיובי).

9.4.2 חישוב יעיל של $R(q)$ (או של $R_0(q)$)

בהינתן A ומצב q של A:

1. $R(q) = \left\{ \delta(q, a) \,\middle|\, \forall a \in \Sigma \right\}$.
2. לכל $a \in \Sigma$ ולכל $p \in R(q)$:
   (א) אם $\delta(p, a) \notin R(q)$ אזי הוסף את p ל-$R(q)$.
3. אם בצעד 2 התווסף ל-$R(q)$ מצב חדש - חזור לצעד 2.
4. החזר את $R(q)$.

חישוב $R_0(q)$ נעשה באופן ממש דומה (פשוט נוסיף את q עצמו ונחזיר).

9.4.3 פתרון יעיל לבעיית הריקנות

בהינתן DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ נחשב $R_0(q_0)$ ונבדוק:
אם $R_o(q_0) \cap F \neq \varnothing$ - אזי נחזיר "כן" - השפה אינה ריקה.
אם לא - נחזיר "לא" - השפה ריקה.

9.4.4 הסרת מצבים שאינם ניתנים להשגה ב-DFA

מחשבים $R_0(q_0)$ - משמיטים מהאוטומט כל מצב q שאינו ב-$R_0(q_0)$ יחד עם כל הקשתות שנכנסות לאותו מצב או יוצאות ממנו.

[9] אורך המעגל = מספר הקשתות.

9.4.5 פתרון יעיל לבעיית הסופיות

למה יהי DFA $A=\langle Q,\Sigma,\delta,q_0,F\rangle$ שכל מצביו ניתנים להשגה ותהי $L=L(A)$.
אזי: L אינסופית $\iff$ קיים מצב $q\in Q$ שמקיים $q\in R(q)$ ו־$R_0(q)\cap F\neq\varnothing$.
-

בהינתן DFA $A=\langle Q,\Sigma,\delta,q_0,F\rangle$:

1. נשמיט מ־A את המצבים שאינם ניתנים להשגה.
2. נחשב את $R(q)$ לכל $q\in Q$ ונבדוק אם $q\in R(q)$ ו־$R_0(q)\cap F\neq\varnothing$.
   (א) אם נמצא כזה נחזיר "כן" ונעצור.
3. נחזיר "לא".

9.4.6 פתרון בעיית השקילות

נתונים שני DFA־ים: $A_1=\langle Q_1,\Sigma,\delta_1,q_1,F_1\rangle$ = A_2 $\langle Q_2,\Sigma,\delta_2,q_2,F_2\rangle$.

1. נבנה אוטומט מכפלה A של A_1 ו־A_2 עם מצב התחלתי (q_1,q_2) וקבוצת מצבים סופיים: $F=(F_1-F_2)\cup(F_2-F_1)$.
2. נבדוק אם $L(A)$ ריקה.
   (א) אם כן נחזיר "כן".
   (ב) אם לא נחזיר "לא".

חלק V

אפיון אלגברי לשפות רגולריות

נדבר על יחסי שקילות על Σ^*.
למשל: נגדיר יחס שקילות R ו־S על Σ^* ע"י:
עבור $x,y\in\Sigma^*$ ($\Sigma=\{a,b\}$):

1. $xRy \iff |x|=|y|$.
2. $xSy \iff \#_a(x)=\#_a(y)$.

מחלקות השקילות של היחס R:
עבור $\alpha\in\Sigma$:

$$[\varepsilon],[\alpha],[\alpha^2],[\alpha^3],\ldots$$

כאן $[\varepsilon]=\{\varepsilon\}$ ולכל $i\geq 1$:

$$[\alpha^i]=\left\{w\in\Sigma^*\middle|\,|w|=i\right\}$$

מחלקות השקילות של היחס S:
עבור $\alpha\in\Sigma$:

$$[\varepsilon],[\alpha],[\alpha^2],[\alpha^3],\ldots$$

כאן: $[\varepsilon]=\left\{w\in\Sigma^*\middle|\#_\alpha(w)=0\right\}$
לכל $i\geq 1$:

$$[\alpha^i]=\left\{w\in\Sigma^*\middle|\#_\alpha(w)=i\right\}$$

10 עידון יחסי שקילות

יהיו R,S יחסי שקילות על Σ^*.
אומרים ש־R הוא **עידון** של S אם $R\subseteq S$.
(כלומר, עבור $x,y\in\Sigma^*$: $xRy\Rightarrow xSy$).
הערה: R הוא עידון של S $\iff$ כל **מחלקת** שקילות של S היא איחוד של **מחלקות** שקילות של R.

11 קבוצת מחלקות השקילות

קבוצת מחלקות השקילות של R ב־Σ^*־ מסומנת: Σ^*/R -

$$\Sigma^*/R=\left\{[x]\middle|x\in\Sigma^*\right\}$$

נשים לב שעבור $x,y\in\Sigma^*$:

$$xRy\iff[x]=[y]$$

(משמעות השוויון ב־Σ^*/R).
ולמשל, בשתי הדוגמאות שלמעלה:

$$\Sigma^*/R=\left\{[a^i]\middle|i\geq 0\right\}$$

11.1 הדרגה (האינדקס) של מחלקות שקילות (rank)

עוצמת הקבוצה Σ^*/R נקראת **הדרגה** (או האינדקס) של R ומסומנת: $\mathrm{rank}(R)$.
אם $\mathrm{rank}(R)<\infty$, אומרים ש־R הוא יחס מ**אינדקס סופי** על Σ^* (אחרת, Rהוא יחס מאינדקס אינסופי).
למשלת בשתי הדוגמאות שלמעלה גם R וגם S הן מאינדקס אינסופיץ
$\mathrm{rank}(R)$ = מספר מחלקות השקילות השונות של היחס R.

למה 11.1 יהיו R ו־S יחסי שקילות על Σ^*. אם $R\subseteq S$ (R עידון של S) אזי: $\mathrm{rank}(S)\leq\mathrm{rank}(R)$
(כי כל מחלקת שקילות של S היא איחוד של מחלקות שקילות של R).

למשל: R,S יחסי שקילות: $R=\bigcup_{i=1}^{8}r_i$ ו־$S=S_1\cup S_2$, אזי $R\subseteq S$ (עידון) אפשרי הינו:

S_2:		S_1:	
r_6	r_5	r_2	r_1
r_8	r_7	r_4	r_3
☻	$\star$	☺	☼

במקרה כאן ☼,☺ $\in S_1$, $\star$,☻ $\in S_2$ אך אף אחד מהללו אינו ב־R.
לעומת זאת - כל $x\in R$ נמצא ב־$S_1\cup S_2$.

12 יחס אינווריאנטי מימין

תהי R יחס שקילות על Σ^*.
אומרים ש־R **אינווריאנטי מימין** אם הוא מקיים, עבור כל $x,y,z\in\Sigma^*$:

$$xRy \Rightarrow xzRyz$$

בדוגמאות שלנו ממקודם - R,S[10] אינוריאנטים מימין.
דוגמא ליחס שאינו אינוריאנטי מימין:
$xRy \iff x=y \vee x=y^R$ (w^R - זאת המילה w ברוורס).
זהו יחס שקילות על $\{a,b\}^*$, אבל הוא אינו אינוריאנטי מימין, כי למשל:
$abRba$ אבל $aba\not{R}baa$.

13 יחס משמר שפה

יהי R יחס שקילות על Σ^* ותהי $L\subseteq\Sigma^*$ שפה.
אומרים ש־R **משמר** את L (או R **מעדן** את L) אם עבור $x,y\in\Sigma^*$:
$y\in L \Leftarrow x\in L\wedge xRy$.
אפשר גם לנסח זאת כך: $(y\in L\Leftrightarrow y\in L) \Leftarrow xRy$
זה קורה $\iff$ L היא איחוד של מחלקות שקילות של R.

14 שפה של מצב $q\in Q$ באוטומט

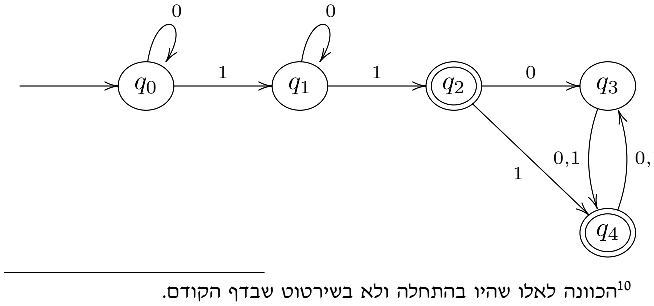
יהי $DFA\ A=\langle Q,\Sigma,\delta,q_0,F\rangle$ עבור $q\in Q$ מסמנים:

$$L(q)=\left\{w\in\Sigma^*\middle|\delta(q_0,w)=q\right\}$$

כלומר, אלו כל המילים ב־Σ^* שמעבירות את האוטומט מהמצב ההתחלתי למצב q.
$L(q)$ נקראת: **שפת המצב q.**
יתכן ש־$L(q)=\varnothing$ (זה קורה אםם q הוא מצב שאינו ניתן להשגה).
אם $p\neq q$ אזי $L(p)\cap L(q)=\varnothing$.
נשים לב כי:

$$\boldsymbol{L(A)=\bigcup_{q\in F}L(q)}$$

(תזכורת: $L(A)$ - שפת האוטומט, השפה שאותה האוטומט מקבל).
למשל, נסתכל באוטומט הבא:



[10] הכוונה לאלו שהיו בהתחלה ולא בשירטוט שבדף הקודם.

אזי שפות המצבים של האוטומט הנ"ל:

$$L(q_0)=0^*$$
$$L(q_1)=0^*10^*$$
$$L(q_2)=0^*10^*1$$
$$L(q_3)=0^*10^*1\left(0\left((0+1)(0+1)\right)^*+1(0+1)\left((0+1)(0+1)\right)^*\right)$$
$$L(q_4)=0^*10^*1\left(1\left((0+1)(0+1)\right)^*+0(0+1)\left((0+1)(0+1)\right)^*\right)$$

כעת, היות והמצבים המקבלים הינם q_2,q_4 אזי נסמן: $r_1=L(q_2)\,,\ r_2=L(q_4)$ אזי:

$$L(A)=r_1+r_2=L(q_2)\cup L(q_4)$$

15 שני יחסי שקילות חשובים R_A ו־R_L

15.1 יחס שקילות ראשון - R_A

יהי $DFA\ A=\langle Q,\Sigma,\delta,q_0,F\rangle$ מעל Σ. נגדיר יחס R_A על Σ^* באופן הבא:
עבור $x,y\in\Sigma^*$:

$$\boxed{\delta(q_0,x)=\delta(q_0,y) \iff xR_Ay}$$

כלומר, אם יש לנו שתי מילים ב־Σ^* אשר מביאות אותנו לאותו מצב אזי שתיהן נמצאות באותה מחלקת שקילות. כלומר, מחלקות השקילות של R_A הינן כל המילים אשר מביאות אותנו לאותו מצב, כלומר, במחלקה q יהיו כל המילים ב־Σ^* כך ש־$\delta(q_0,w)=q$ כאשר w זאת המילה.

1. זהו יחס שקילות על Σ^* (קל לבדוק).
2. זהו יחס אינווריאנטי מימין.
3. מחלקות השקילות של היחס R_A הינן:

עבור $x\in\Sigma^*$:

$$[x]=\left\{y\in\Sigma^*\middle|xR_Ay\right\}=\left\{y\in\Sigma^*\middle|\underbrace{\delta(q_0,x)}_{=q}=\delta(q_0,y)\right\}$$
$$=\left\{y\in\Sigma^*\middle|\delta(q_0,y)=q\right\}=L(q)$$

כלומר, מחלקות השקילות של היחס R_A הינן שפות המצבים של האוטומט.
קיבלנו:

$$[x]=L(q)$$
$$q=\delta(q_0,x)$$

לכן: מחלקות השקילות של היחס R_A הן שפות המצבים של Aשאינן ריקות.
אם <u>אין</u> ב־A מצבים שאינם ניתנים להשגה, אזי מחלקות השקילות של A הן שפות כל המצבים של A.

4. נסמן את מספר המצבים של A ב־n. אזי, על סמך מה שנאמר ב־3 אנחנו יודעים ש־

$$\text{rank}\,(A) \le n$$

ואם אין ב־A מצבים הניתנים להשגה, אזי:

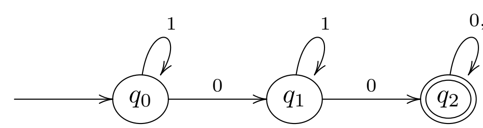
$$\text{rank}\,(A) = n$$

5. נסמן: $L = L(A)$ אז R_A משמר (מעדן) את L, כי L היא איחוד של מחלקות שקילות של היחס R_A כי:

$$L = L(A) = \bigcup_{q\in F} L(q)$$

$$\left(\Sigma^* = \bigcup_{q\in Q} L(q)\right)$$

<u>דוגמא:</u>
ניקח את A להיות:



מחלקות השקילות של היחס R_A הינן:

$$[\varepsilon] = L(q_0) = 1^*$$
$$[0] = L(q_1) = 1^*01^*$$
$$[00] = L(q_2) = 1^*01^*0\,(0+1)^*$$

הערה: אנחנו בוחרים לסמן את מחלקת השקילות במילה הכי קצרה שיש באותה מחלקה, לפיכך בחרנו את $[00], [0], [\varepsilon]$.

15.2 יחס שקילות שני - R_L

עבור שפה $L \subseteq \Sigma^*$ נגדיר יחס R_L על Σ^* באופן הבא:
עבור: $x, y \in \Sigma^*$

$$\boxed{\forall z \in \Sigma^* : \ (xz \in L \Leftrightarrow yz \in L) \iff xR_L y}$$

כלומר: x, y נמצאים באותה מחלקת שקילות אםם קיימת מילה $z \in \Sigma^*$ כך שגם $xz \in L$ וגם $yz \in L$ (כמובן שיכולות להיות מספר מילים כאלו).
ננסח את הגדרת R_L בדרך נוספת:
<u>הגדרה:</u> עבור שפה $L \subseteq \Sigma^*$ ומילה $x \in \Sigma^*$, ההקשר הימני של x ב־L מסומן: $\text{rc}_L(x)$ ומוגדר ע"י:

$$\text{rc}_L(x) = \left\{ z \in \Sigma^* \middle| xz \in L \right\}$$

כעת, קל לראות עבור $x, y \in \Sigma^*$:

$$\text{rc}_L(x) = \text{rc}_L(y) \iff xR_L y$$

(זוהי הגדרה שקולה ליחס שלמעלה).
R_L הוא יחס שקילות ויחס אינווריאנטי מימין.

15.3 מספר דוגמאות

15.3.1 דוגמאות לשפות סופיות

ניקח: $\Sigma = \{a, b\}$ ו־$L = \{a, ba, baa\}$.
נחשב את מחלקות השקילות של R_L:
בטור הראשון (השמאלי) יהיו מחלקות השקילות של היחס R_L שאיחוד שלהם הוא בעצם על המילים ב־Σ^*, מה שיבדיל ביניהן הוא הסיפא, כלומר ה־rc_L - המילה $z \in \Sigma^*$ שתגרום לאותה מילה שבטור השמאלי להיות בשפה.

$\text{rc}_L(x)$	x **(כל המילים ב־Σ^*)**
$a+ba+baa$	ε
ε	$a+baa$
$a+aa$	b
$\varepsilon+a$	ba
ε	baa **(יש כבר מחלקה כזאת, היכן ש־a)**
$\varnothing$	**כל שאר המילים**

לכן מחלקות השקילות הינן (תזכורת, לוקחים את המילה הכי קצרה מכל מחלקה [הטור השמאלי]):

$$\begin{array}{ccccc} [\varepsilon] & [a] & [b] & [ba] & [aa] \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow \\ \{\varepsilon\} & \{a, baa\} & \{b\} & \{ba\} & \oplus \end{array}$$

$\oplus$ - כל שאר המילים .
מה שקובע לנו את מחלקות השקילות זה הטור הימני, למשל, הסיבה ש־a ו־baa באותה מחלקת שיקילות היא לשניהם אפשר להוסיף רק ε כדי שהם יהיו בשפה, והיות a זאת המילה הקצרה אותה בחרנו שתייצג את המחלקה.
למשל המחלקה $[b] = \{b\}$ כי היא המילה היחידה ב־Σ^* שאם נוסיף לה $z = a + aa$ היא תהיה בשפה.
או: $[ba] = \{ba\}$ מאותה סיבה כנ"ל - זאת המילה היחידה שאם נוסיף לה $z = \varepsilon + a$ היא תהיה בשפה.
ה־rc_L (הטור הימני) עוזר לנו להפריד בין חלקות השקילות.
<u>חשוב לזכור:</u>
מה שמבדיל בין מחלקות השקילות זאת אותה סיפא (z, המילה שאנחנו מוסיפים) שאנחנו מוסיפים ל־x כדי שהיא תיכנס לשפה.
למשל: a נמצאת באותה מחלקת שקילות עם baa כי לשתיהן אם נוסיף **רק** ε המילה תישאר בשפה, אחרת לא.
לעומת זאת, למילה ba אנחנו יכולים להוסיף, גם ε וגם a $(z = a + \varepsilon)$ ובשני המקרים היא תישאר בשפה, לכן היא לא חלק ממחלקת השקילות $[a]$, כי לה בנוסף ל־ε את a.
מספיק שיש מילה <u>**אחת**</u> שונה ב־rc_L בשביל להפריד בין מחלקות השקילות!
למשל, ההבדל בין מחלקת המחלקות $[a], [ba]$ היא המילה a: ב־$[a]$ נוכל להוסיף <u>רק</u> ε, ולעומת זאת ב־$[ba]$ אפשר להוסיף או את ε או את a וכל המילים שבמחלקת השיקילות הנ"ל תישאר בשפה.

15.3.2 דוגמאות לשפות אינסופיות

$\Sigma = \{0, 1\}$, $L = 10^*1^*$.
נחשוב על שפה זאת כ־$L_1 = 10^* + 10^*11^*$ (ההבדל כאן הוא סוף המילה - אפשרות אחת היא כל המילים שמסתיימות ב־0 בשפה L אפשרות נוספת היא כל המילים שמתסיימות ב־1 ב־L. $(L = L_1)$.

נחשב את מחלקות השקילות של היחס R_L:

x **(כל המילים ב־Σ^*)**	$\mathrm{rc}_L(x)$
ε	$10^*+10^*11^*$
10^*	$0^*+0^*11^*$
10^*11^*	1^*
$0(0+1)^*+10^*11^*0(0+1)^*$	$\varnothing$

על־כן ישנן 4 מחלקות שקילות של היחס R_L:

$$\begin{array}{cccc} [\varepsilon] & [1] & [11] & [0] \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow \\ \{\varepsilon\} & \{10^*\} & \{10^*11\} & \left\{0(0+1)^*+10^*11^*0(0+1)^*\right\} \end{array}$$

חשוב לשים לב לכך שבין כל מחלקת שקילות מפרידה לפחות מילה אחת (כלומר, מסתכלים על ה־$\mathrm{rc}_L(x)$ ובודקים שיש מילה אחת לפחות שמפרידה בין המחלקות - אם אין מילה כזאת - אזי מדובר באותה מחלקת שקילות).
לגבי איך בונים את המילים של המחלקות (x): צריך למצוא אילו מילים יתנו לנו מחלקות שקילות שונות. למשל: ε ו־0^* לא בהכרח יניבו לנו מחלקות שקילות שונות (כי $\{\varepsilon\}\in 0^*$), לכן ניתן למשל לשים 00^* כדי ליצור את ההבדל.

דוגמא נוספת:
$\Sigma=\{1,0\}$, $L=(0+1)^*1(0+1)$ - זאת השפה שמקבלת את כל המילים שהאות האחת לפני אחרונה (השנייה מהסוף) היא "1".
נחשב את מחלקות השקילות של היחס R_L:

x **(כל המילים ב־Σ^*)**	$\mathrm{rc}_L(x)$
$\varepsilon+0+(0+1)^*00$	$(0+1)^*1(0+1)$
$1+(0+1)^*01$	$0+1+(0+1)^*1(0+1)$
$(0+1)^*10$	$\varepsilon+(0+1)^*1(0+1)$
$(0+1)^*11$	$\varepsilon+0+1+(0+1)^*1(0+1)$

ולכן, מחלקות השקילות של היחס R_L הינן:
$[11],[10],[1],[\varepsilon]$ (לוקחים את המילה הכי קצרה מכל ביטוי).

15.3.3 דוגמא לשפה אינסופית אם אינסוף מחלקות שקילות

$\Sigma=\{a,b\}$, $L=\left\{w\in\Sigma^*\middle|\#_a(w)=\#_b(w)\right\}$.
עבור מילה $x\in\Sigma^*$:

$$\begin{aligned}\mathrm{rc}_L(x) &= \left\{z\in\Sigma^*\middle|\#_a(x)+\#_a(z)=\#_b(x)+\#_b(z)\right\}\\ &= \left\{z\in\Sigma^*\middle|\#_a(z)-\#_b(z)=\#_b(z)-\#_a(z)\right\}\end{aligned}$$

לכן, מחלקות השקילות של R_L הינן:
$\ldots,[b^3],[b^2],[b],[\varepsilon],[a],[a^2],[a^3],\ldots$
כאשר:

$$[\varepsilon] = \left\{x\in\Sigma^*\middle|\#_a(x)=\#_b(x)\right\}$$

ולכל $i\geq 1$:

$$\begin{aligned}[a^i] &= \left\{x\in\Sigma^*\middle|\#_a(x)-\#_b(x)=i\right\}\\ [b^i] &= \left\{x\in\Sigma^*\middle|\#_b(x)-\#_a(x)=i\right\}\end{aligned}$$

בדוגמא זאת R_A היא מיחס (מדרגה) אינסופי.

למה 15.1 תהי Lשפה רגולרית מעל Σ, ויהי $A=\langle Q,\Sigma,\delta,q_0,F\rangle$ DFA שמזהה את L אזי:

$$R_A\subseteq R_L$$

מסקנה מנתוני הלמה:

$$\mathrm{rank}(R_A)\leq\mathrm{rank}(R_L)$$

16 משפט מייהיל־נרוד

(Myhill–Nerode theorem)
יהי Σ א"ב ותהי $L\subseteq\Sigma^*$ שפה. אזי:

$$\boxed{L \text{ רגולרית} \iff \mathrm{rank}(R_L)<\infty}$$

משפט זה עוזר לנו להוכיח ששפה היא רגולרית.
אם אנחנו מראים שישנם מספר סופי של מחלקות שקילות R_L - אזי השפה היא רגולרית.

16.1 בניית אוטומט DFA מינימלי

אחרי שגילינו כי האוטומט סופי ומצאנו את מחלקות השקילות, ניתן לבנות אוטומט DFA שמזהה את השפה L עם מספר מצבים מינימלי באופן הבא:
נתון לנו כי $\mathrm{rank}(R_L)<\infty$ - כלומר מדובר בקבוצה סופית:
אזי נבנה:

$$A_L=\langle Q,\Sigma,\delta,q_0,F\rangle$$

- $Q={}^{\Sigma^*}/_{L}$ - קבוצת מחלקות השקילות של R_L (קבוצה סופית).
- $q_o=[\varepsilon]$ - מחלקת השקילות של ε.
- $F=\left\{[x]\middle|x\in L\right\}$ - כלומר, כל מחלקות השקילות שהנציג שלהם x ב־L $(x\in L)$.
- פונקציית המעברים:

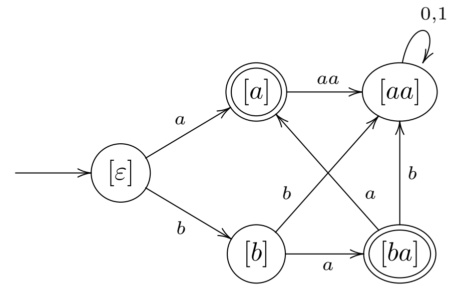
$$\begin{gathered}\delta:Q\times\Sigma\to Q\\ \delta([x],a)=[xa]\end{gathered}$$

וזה תקף גם עבור מילה:

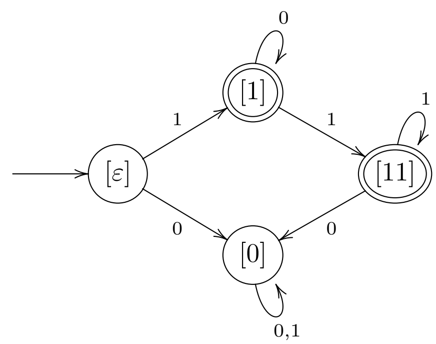
$$\begin{gathered}\forall w\in\Sigma^*,[x]\in Q:\\ \delta([x],w)=[xw]\end{gathered}$$

נראה את האוטומטים <u>המינימליים</u> לדוגמאות של השפות שהיו מקודם (ע"פ מחלקות השקילות של היחס R_L):

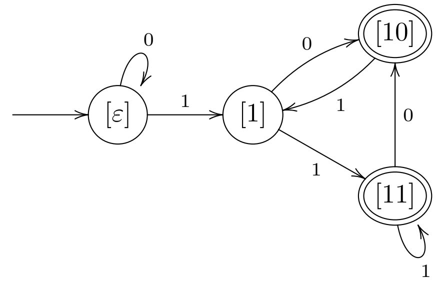
:$L = \{a, ba, baa\}$



:$L = 10^*1^* = 10^* + 10^*11$



:$L = (0+1)^* \, 1 \, (0+1)$



16.2 דוגמא להוכחה באמצעות משפט מייהיל־נרוד ששפה אינה רגולרית

נוכיח כי השפה $L = \left\{0^n 1^n \middle| n \geq 0\right\}$ <u>אינה</u> רגולרית.

הוכחה: מספיק להראות שיש מספר אינסופי של מחלקות שקילות של היחס R_L, ואכן:

לכל $i \neq j$: $a^i \not{R_L} a^j$ (כי $b^i \in \mathrm{rc}_L\left(a^i\right)$ אבל $b^i \notin \mathrm{rc}_L\left(a^j\right)$.

כלומר, b^i היא מילה מפרידה בין שתי מחלקות השקילות $\left[a^i\right]$ ו־$\left[a^j\right]$).

לכן: $[\varepsilon], [a], \left[a^2\right], \left[a^3\right], \ldots$ הן מחלקות שקילות <u>שונות</u> ביחס R_L, לכן R_L הוא מאינדקס אינסופי ולכן ע"פ משפט מייהיל־נרוד L אינה רגולרית.

חלק VI
מינימיזציה של אוטומט DFA

מקודם בנינו אוטומט מינימלי לשפה L ע"י כך שבנינו את האוטומט A_L, כעת נראה דרך קצת פחות מסובכת לבניית אוטומט שיזהה את L עם מספר מצבים זהה לזה של A_L.

17 מצבים ניתנים להפרדה

יהי $DFA \; A = \langle Q, \Sigma, \delta, q_0, F\rangle$ כלשהו, כאשר $\boldsymbol{p, q \in Q}$.

נגדיר יחס E על Q[11] באופן הבא:

$$\boldsymbol{: \forall z \in \Sigma^*}$$
$$\boldsymbol{(\delta(p,z) \in F \Leftrightarrow \delta(q,z) \in F) \iff : pEq}$$

נשים לב שהמשמעות היא שעבור <u>כל</u> מילה $z \in \Sigma^*$ אין זה משנה אם אנחנו ב־p או ב־q - או שהיא תתקבל משניהם או שלא.

אם pEq אומרים ש־<u>p שקול ל־q</u>. (p ו־q שקולים).

E הוא יחס שקילות על Q שמחלק אותו למחלקות שקילות.

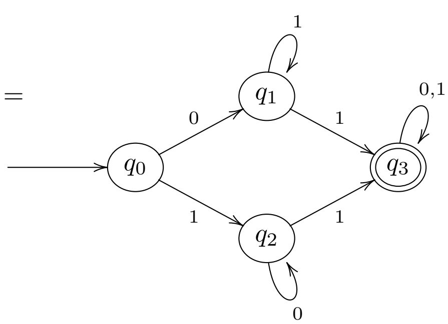
אם p ו־q אינם שקולים אומרים ש־p ו־q **ניתנים להפרדה.**

p־וq ניתנים להפרדה $\iff$ <u>קיימת</u> מילה $w \in \Sigma^*$ כך ש־$\delta(p,w) \in L \wedge \delta(q,w) \notin L$ (או להפך).

מילה w כזאת נקראת **מילה מפרידה.**

דוגמא:

$A =$



$q_1 E q_2$ - כי עבור $z \in \Sigma^*$:

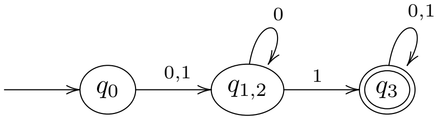
$(\delta(q_1,z) \in F \Leftrightarrow \delta(q_2,z) \in F) \iff z \in 0^*1(0+1)^*$.

$q_0 \not{E} q_1$ - מילה מפרידה "1" (למשל).

$q_0 \not{E} q_2$ - מילה מפרידה "1" (למשל).

$q_1 \not{E} q_3$ - מילה מפרידה "ε" (למשל).

ניתן לצמצם את שני המצבים השקולים ב־A למצב אחד ולקבל אוטומט שקול ל־A עם מספר מצבים קטן יותר:



שני האוטומטים שקולים - השפה שלהם היא: $(0+1)\,0^*1\,(0+1)^*$.

[11] חשוב לשים לב לכך שזהו יחס על מצבים ולא על מילים.

18 בניית אוטומט הצימצום של A

בהינתן אוטומט DFA $A = \langle Q, \Sigma, \delta, q_0, F\rangle$, נבנה אוטומט A' הנקרא **"הצמצום של A"** באופן הבא:

$$A' = \langle Q', \Sigma, \delta', q_0', F'\rangle$$

כאשר:

- $Q' = {}^{Q}/_{E}$ - קבוצת מחלקות השקילות של היחס E. $Q' = \left\{ [q] \middle| q \in Q \right\}$.
- $q_0' = [q_0]$.
- $F' = \left\{ [q] \middle| q \in F \right\}$ (הערה: לא כל המצבים המקבלים חייבים להיות באותה מחלקת שקילות). כלומר כל מחלקות השקילות של F.
- לגבי δ':

$$\delta' : Q' \times \Sigma \to Q'$$
$$\delta'([q], a) = [\delta(q, a)], \quad (\forall a \in \Sigma)$$

וכמובן שניתן להרחיב את δ' גם עבור מילה $w \in \Sigma^*$: $\delta'([q], w) = [\delta(q, w)]$.

הגדרה 18.1 אוטומט DFA שאין בו מצבים שונים שקולים נקרא **אוטומט מצומצם**. אוטומט הוא מצומצם אסם מתקיים: עבור כל $p, q \in Q$: $p = q \Leftarrow pEq$.

18.1 אלגוריתם לזוהי מצבים שקולים ב-DFA

נתון $A = \langle Q, \Sigma, \delta, q_0, F\rangle$
שלב מקדים: עבור כל שני מצבים שונים $p, q \in Q$ נסמן:

$$\Delta(p, q) = \varnothing$$

1. לכל $p \neq q$: אם $p \notin F \wedge q \in F$ (או ההפך) אזי $\Delta(p, q) = \varepsilon$.
2. עבור כל שני מצבים <u>שונים</u> p, q שעבורם $\Delta(p, q) = \varnothing$ <u>ולכל</u> $a \in \Sigma$ בודקים האם:

$$\Delta(\delta(p, a), \delta(q, a)) \neq \varnothing$$

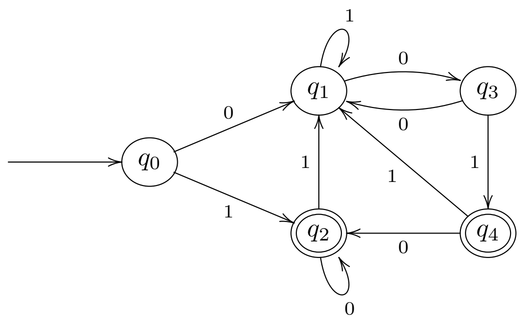
כלומר - זה מסומן.

(א) אם כן: נסמן $\Delta(p, q) = a$ - ונעבור לזוג הבא (נפסיק לחפש עבור הזוג הזה).

3. אם בביצוע האחרון של צעד 2 סומן לפחות זוג מצבים חדש, אז חזור ל-2.
4. הכרז על כל זוג מצבים שונים p, q שעבורם $\Delta(p, q) = \varnothing$ כמצבים שוקלים ועצור.

18.1.1 דוגמא לשימוש באלגוריתם

ניקח את האוטומט הבא:



מיישמים את האלגוריתם בכך שבונים טבלה שמכילה תא אחד לכל זוג מצבים שונים: (תאים עם הריבוע השחור זה תאים שכבר יש לזוג שלהם תא בטבלה, וכל תא ריק פירושו $\varnothing$ כאשר $\Delta(p, q)$ = תוכן התא).

q_1		■	■	■
q_2			■	■
q_3				■
q_4				
	q_0	q_1	q_2	q_3

כעת נשים לכל זוג מצבים כך ש-$p \in F$ ו-$q \notin F$ (או ההפך) ε בתא:

q_1		■	■	■
q_2	ε	ε	■	■
q_3				■
q_4	ε	ε		ε
	q_0	q_1	q_2	q_3

נתחיל מ-(q_0, q_1): ניקח $a = 0$ ונראה ש-$\Delta(\delta(q_0, 0), \delta(q_1, 0)) = \Delta(q_1, q_3) = \varnothing$.
לכן ע"פ האלגוריתם נצטרך להמשיך לחפש: $\Delta(\delta(q_0, 1), \delta(q_1, 1)) = \Delta(q_2, q_1) = \varepsilon$.
זהו! התא אינו ריק - ולכן $\Delta(q_0, q_1) = 1$:

q_1	1	■	■	■
q_2	ε	ε	■	■
q_3				■
q_4	ε	ε		ε
	q_0	q_1	q_2	q_3

כעת נעבור ל-(q_0, q_3): נשים לב כי:
$\Delta(\delta(q_0, 0), \delta(q_3, 0)) = \Delta(q_1, q_1)$ - והגענו לאותו מצב לכן נשאיר את זה ריק.
$\Delta(\delta(q_0, 1), \delta(q_3, 1)) = \Delta(q_2, q_4) = \varnothing$ - גם כאן הגענו למצב ריק. לכן נשאיר את $\Delta(q_0, q_3) = \varnothing$ (במצב ריק).

כעת נעבור ל-(q_1,q_3):
$\Delta\left(\delta\left(q_1,0\right),\delta\left(q_3,0\right)\right)=\Delta\left(q_3,q_1\right)=\varnothing$ - ולכן נמשיך הלאה...
$\Delta\left(\delta\left(q_1,1\right),\delta\left(q_3,1\right)\right)=\Delta\left(q_1,q_4\right)=\varepsilon$ ולכן $\Delta\left(q_1,q_3\right)=1$
ונעדכן את הטבלה:

q_1	1	■	■	■
q_2	ε	ε	■	■
q_3		1		■
q_4	ε	ε		ε
	q_0	q_1	q_2	q_3

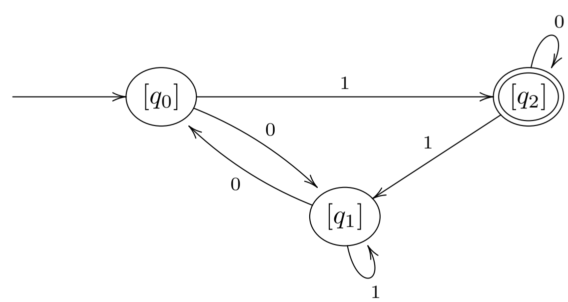
נעבור ל-(q_2,q_3):
$\Delta\left(\delta\left(q_2,0\right),\delta\left(q_3,0\right)\right)=\Delta\left(q_2,q_1\right)=\varepsilon$ ולכן $\Delta\left(q_2,q_3\right)=0$.

q_1	1	■	■	■
q_2	ε	ε	■	■
q_3		1	0	■
q_4	ε	ε		ε
	q_0	q_1	q_2	q_3

כעת, נסתכל על הזוג (q_2,q_4):
נשים לב כי בשני המקרים נקבל $\varnothing$ (כי עבור 0 נגיע בשניהם ל-q_2 ועבור 1 נגיע בשניהם ל-q_1).
ולכן אנחנו נשאיר את הטבלה כמו שהיא.
הטבלה הסופית הינה:

q_1	1	■	■	■
q_2	ε	ε	■	■
q_3		1	0	■
q_4	ε	ε		ε
	q_0	q_1	q_2	q_3

ולכן: q_0Eq_3, q_2Eq_4, כלומר מחלקות השקילות של היחס E על Q הינן:
$\left[q_0\right]=\left\{q_0,q_3\right\},\left\{q_1\right\}=\left[q_1\right],\left\{q_2,q_4\right\}=\left[q_2\right]$.
כעת, נבנה את אוטומט הצמצום של A:



הערה 18.2 ייתכן שב-A ישנם מצבים שאינם ניתנים להשגה, ולכן יתכן שגם ב-A' (אוטומט הצמצום של A) ישנם מצבים שאינם ניתנים להשגה. לכן, מינימיזציה של אוטומט A DFA מורכבת משני שלבים:
א. הסרת מצבים שאינם ניתנים להשגה ב-A.
ב. צמצום האוטומט שנשאר.

מינימיזציה של אוטומט A DFA נותנת A' DFA שמזהה את L והוא מינימלי מבין כל האוטומטים שמזהים את A.
למעשה זה נותן את A_L.

19 איזומורפיזם בין אוטומטים

__הגדרה:__ יהיו

$$A_1=\left\langle Q_1,\Sigma,\delta_1,q_1,F_1\right\rangle$$
$$A_2=\left\langle Q_2,\Sigma,\delta_2,q_2,F_2\right\rangle$$

שני DFA-ים.
אומרים ש A_1 **איזומופרי ל-A_2** ורושמים $A_1\cong A_2$ אם קיימת פונקציה חח"ע ועל $f:Q_1\to Q_2$ ששומרת על המצב ההתחלתי, שומרת על מצבים סופיים ושומרת על פונקציית המעברים, כלומר, פונקציה שמקיימת:

1. $f\left(q_1\right)=q_2$.
2. עבור $q\in Q_1$:
$f\left(q\right)\in F_2 \iff q\in F_1$.
3. $\delta_2\left(f\left(q\right),a\right)=f\left(\delta_1\left(q,a\right)\right)$, לכל $q\in Q_1$ ולכל $a\in\Sigma$.

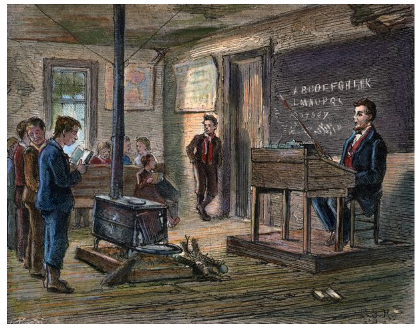
19.1 משפט יחידות האוטומט המינימלי

יהי DFA $A=\left\langle Q,\Sigma,\delta,q_0,F\right\rangle$ מצומצם שכל מצביו ניתנים להשגה ותהי $L=L\left(A\right)$, אזי:

$$\boldsymbol{A\cong A_L}$$

סוף הנושא של אוטומטים!!

חלק VII
דקדוקים חסרי הקשר ושפות חסרות הקשר



20 דקדוקים חסרי הקשר (CFG)

20.1 הגדרות

דקדוק חסר הקשר זוהי רביעיה: $G=\langle V,\Sigma,P,A\rangle$ כאשר:

- V - היא קבוצה סופית לא ריקה שערכיה נקראים **משתנים דקדוקיים** או: משתנים, נהוג לסמן את אברי הקבוצה ב: $A,B,C,\dots$.
- Σ - היא קבוצה סופית, לא ריקה וזרה ל-V שאריה נקראים **סימנים טרמינליים** או: אותיות. נהוג לסמן את אברי הקבוצה ב: $a,b,c,\dots$.
- P - היא קבוצת **כללי הגזירה** (או: כללי השכתוב של G), זוהי קבוצה חלקית לקבוצה $V\times(V\cup\Sigma)^*$, כלומר, מדובר בזוגות סדורים (A,α) כאשר $A\in V$ ו-$\alpha\in(V\cup\Sigma)^*$. את כלל הגזירה (A,α) נהוג לרשום בצורה הבאה: $\boldsymbol{A\to\alpha}$.
- S - הינו איבר מיוחד ב-V אשר נקרא **המשתנה ההתחלתי** של G.

לגבי P:
אם נניח $V=\{S,A\},\Sigma=\{a,b,c\}$ אזי אברי P יכולים להיות:

$$\begin{array}{ccc}(S,aSa) & \Leftrightarrow & S\to aSa\\ (A,AaS) & \Leftrightarrow & A\to AaS\\ (S,aAbc) & \Leftrightarrow & S\to aAbc\\ (S,a) & \Leftrightarrow & S\to a\end{array}$$

כאשר יש לנו יותר משתי אפשרויות עבור איבר כלשהו ב-V, אזי נהוג לציין זאת ב-"או", למשל, עבור הדוגמא הנ"ל:

$$S\to aSa\big|aAbc\big|a$$

והקו המפריד מסמל "או" (כלומר, אנחנו יכולים לבחור רק אחת מהאפשרויות).

- איבר של הקבוצה $V\cup\Sigma$ נקרא **סימן** בדקדוק, ובאופן כללי מסמנים אותו ב: $X,Y,Z,\dots$.
- מילה ב-$(V\cup\Sigma)^*$ נקראת **תבנית פסוקית** (למשל: $aAbSA$) ומילה ב-Σ^* נקראת **מילה טרמינלית**. (למשל: $abaac$).
נסמן תבניות פסוקיות ב: $\alpha,\beta,\gamma,\dots$.
נסמן מילים טרמינליות ב: $u,v,w,\dots$.
כמובן שכל סימן וכל מילה טרמינלית הם תבניות פסוקיות.

20.1.1 הגדרות גזירה

- תהיינה $\varphi,\psi\in(V\cup\Sigma)^*$. אומרים ש-$\psi$ **ניתנת לגזירה ישירה מ-φ** בדקדוק G ורושמים:

$$\varphi\underset{G}{\Rightarrow}\psi$$

אם $\varphi=\beta\boldsymbol{A}\gamma$ ו-$\psi=\beta\boldsymbol{\alpha}\gamma$ ו- $\boldsymbol{A\to\alpha}$ הוא כלל גזירה ב-P.
(מדובר כאן על מעבר אחד בלבד, כמו-כן, אם ידוע הדקדוק, כי יש רק אחד למשל, אזי אפשר להשמיט את ה-G מהסימן הנ"ל).

- אומרים ש-ψ **גזירה מ-φ**[12] בדקדוק G ורושמים:

$$\varphi\underset{G}{\Rightarrow}^*\psi$$

אם קיימת סדרה סופית של תבניות פסוקיות $\varphi_1,...,\varphi_n\in(V\cup\Sigma)^*$ כך ש-$\varphi=\varphi_0$, $\varphi_n=\psi$ ולכל $0\le i<n$:

$$\varphi_i\underset{G}{\Rightarrow}\varphi_{i+1}$$

סדרת תבניות פסוקיות כזאת נקראת **סדרת גזירה (או: גזירה) מאורך n** בדקדוק G.

- נאמר ש-ψ **גזירה** (או: ניתנת לגזירה) **מ-φ ב-n צעדים** ונרשום:

$$\varphi\underset{G}{\Rightarrow}^n\psi$$

אם קיימת סדרת גזירה מאורך n: $\varphi_1,\dots,\varphi_n$ כך ש-$\varphi=\varphi_1$ ו-$\psi=\varphi_n$.

- נשים לב שלפי ההגדרות הנ"ל:
  - ◄ $\varphi\underset{G}{\Rightarrow}^0\psi\iff\varphi=\psi$.
  - ◄ $\varphi\underset{G}{\Rightarrow}^1\psi\iff\varphi\underset{G}{\Rightarrow}\psi$.
  - ◄ $\varphi\underset{G}{\Rightarrow}^n\psi\iff\varphi\underset{G}{\Rightarrow}^*\psi$ עבור $n\ge 0$.
- תבנית פסוקית $\phi\in(V\cup E)^*$ היתנת לגזירה מהמשתנה ההתחלתי S (של הדקדוק G) כלומר: $S\underset{G}{\Rightarrow}^*\phi$ נקראת: **תבנית פסוקית של G**.
- **מילה טרמינלית של G** היא מילה $w\in\Sigma^*$ אשר מקיימת: $S\underset{G}{\Rightarrow}^*w$.

[12] או: ψ ניתנת לגזירה מ-φ

- **השפה** של דקדוק חסר הקשר $G=\langle V,\Sigma,P,S\rangle$ מסומנת ב־$L(G)$ ומוגדרת ע"י:

$$\boxed{L(G)=\left\{w\in\Sigma^*\middle|S\underset{G}{\Rightarrow}^* w\right\}}$$

- שאומרים על דקדוק G שהוא **יוצר** את L אזי הכוונה היא ש־$L(G)=L$.
- שפה L כאשר $L\subseteq\Sigma^*$ נקראת **שפה חסרת הקשר (CFL)** אם קיים דקדוק חסר הקשר $G=\langle V,\Sigma,P,S\rangle$ שיוצר אותה.
- אומרים שהדקדוקים G_1,G_2 הם **דקדוקים שקולים** אם $L(G_1)=L(G_2)$ ־ כלומר, הם יוצרים את אותה השפה.
- אומרים ש־$G_1=\langle V_1,\Sigma_1,P_1,S_1\rangle$ הוא **דקדוק חלקי** ל־$G_2=\langle V_2,\Sigma_2,P_2,S_2\rangle$ אם:
  - $V_1\subseteq V_2$.
  - $\Sigma_1\subseteq\Sigma_2$.
  - $P_1\subseteq P_2$.
  - $S_1=S_2$.

  כמובן שבמקרה זה מתקיים: $L(G_1)\subseteq L(G_2)$.

20.2 תכונות סגירות של שפות חסרות הקשר

יהיו: $G_1=\langle V_1,\Sigma,P_1,S_1\rangle$ ו־$G_2=\langle V_2,\Sigma,P_2,S_2\rangle$ שני דקדוקים חסרי הקשר. נסמן: $L(G_1)=L_1$, $L(G_2)=L_2$.

בה"כ נניח כי $V_1\cap V_2=\varnothing$[13].
כעת נתאר בניות של דקדוקים חסרי הקשר שיוצרים את השפות הנ"ל: $L_1\cup L_2$, L_1L_2, L_1^*.
בכל שלושת הסעיפים ה־S שנגדיר ב־V הוא סימן חדש, כלומר, הוא סימן ספציפי לדקדוק עצמו.

20.2.1 $L_1\cup L_2$

$$G=\langle V_1\cup V_2\cup\{S\},\Sigma,P_1\cup P_2\cup\{S\to S_1,S\to S_2\},S\rangle$$

כלומר, P בדקדוק הנ"ל הינו:

$$S\to S_1\big|S_2$$
$$P_1\to ...$$
$$P_2\to ...$$

דקדוק זה יוצר את $L_1\cup L_2$.

[13] אם ישנו סימן שמופיע בשתי הקבוצות פשוט נחליף אותו באחת מהקבוצות כך שהתנאי יתקיים.

20.2.2 L_1L_2

$$G=\langle V_1\cup V_2\cup\{S\},\Sigma,P_1\cup P_2\cup\{S\to S_1S_2\},S\rangle$$

כלומר, P בדקדוק הנ"ל הינו:

$$S\to S_1S_2$$
$$P_1\to ...$$
$$P_2\to ...$$

דקדוק זה יוצר את L_1L_2.

20.2.3 L_1^*

$$G=\langle V_1\cup\{S\},\Sigma,P_1\cup\{S\to S_1S,S\to\varepsilon\},S\rangle$$

כלומר, P בדקדוק הנ"ל הינו:

$$S\to S_1S\big|\varepsilon$$
$$P_1\to ...$$

דקדוק זה יוצר את L_1^*.

משפט 20.1 משפחת השפות חסרות ההקשר מעל א"ב Σ, סגורות לאיחוד, שירשור ואיטרציה (כלומר, לפעולות הרגולריות).

20.3 דוגמאות לשפות חסרות הקשר

נסתכל על שתי דוגמאות לשפות חסרות הקשר:
בשני המקרים: $\Sigma=\{a,b\}$.
דוגמה ראשונה ־ כל המילים שמסתיימות ב־a, כלומר ־ $L=\left\{w\in L\middle|w=(a+b)^*a\right\}$:

$$G=\langle\{S\},\{a,b\},\{S\to a,\ S\to aS,\ S\to bS\},S\rangle$$

כלומר, כללי הגזירה בשפה הינן:

$$S\quad\to\quad aS\big|bS\big|a$$

נגזור למשל את המילה: $bbaa$ באמצעות סדרת כללי גזירה:

$$S\Rightarrow b\boldsymbol{S}\Rightarrow bb\boldsymbol{S}\Rightarrow bba\boldsymbol{S}\Rightarrow bbaa$$

בכל פעם השתמשנו בכלל גזירה ושמנו אותו במקום ה־S.
<u>דוגמא שנייה</u> ־ כל המילים מהצורה ־ a^nb^n עבור $\mathbb{N}\ni n\geq 0$:

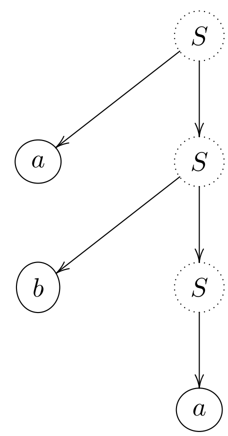
$$G=\langle\{S\},\{a,b\},\{S\to\varepsilon,\ S\to aSb,\},S\rangle$$

$$S\quad\to\quad aSb\big|\varepsilon$$

21 עצי גזירה

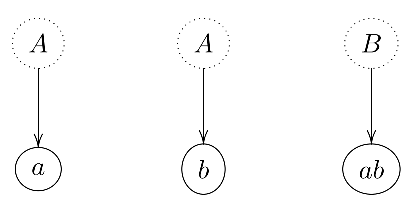
ניתן לגזור מילה בדקדוק חסר הקשר לא רק באמצעות סדרת גזירה אלא למשל באמצעות עץ.

אם למשל נרצה לגזור את המילה aba מהשפה הראשונה אזי נוכל ליצור עץ כזה:



נשים לב באופן כללי להבדל בין שני המקרים:

אם יש לנו את כללי הגזירה הבאים: $A \to a|b$ ו-$B \to ab$ אזי עצי הגזירה יראו כך:



חשוב לשים לב שבמקרה של A אנחנו צריכים לבחור איזה עץ אנחנו רוצים משתי האפשרויות.

21.1 הגדרות פורמליות

הגדרה 21.1 יהי $G = \langle V, \Sigma, P, S \rangle$ דקדוק חסר הקשר. **עץ גזירה** ב-G זהו עץ מכוון[14] המקיים את הדרישות הבאות אשר נסמנו ב-T:

1. כל צומת בעץ מסומן בסימן של הדקדוק $X \in V \cup \Sigma$ או ב-ε.
2. השורש של העץ מסומן ב-S (המשתנה ההתחלתי של G).
3. הסימן של כל צומת פנימי (קודקוד שאינו עלה) הוא איבר של V (משתנה דקדוקי).
4. כל צומת שמסומן ב-ε הוא צומת יחיד (כלומר, לא ייתכן כי לאותו מוצת יהיה אח).
5. לכל צומת פנימי המסומן ב-A $(A \in V)$: אם הבנים של A הם $X_1, X_2, \cdots, X_n$ אזי $A \to X_1 X_2 \cdots X_n \in P$ (הוא כלל גזירה).

כמה הערות:

- כל צומת בעץ שמסומן ב-ε או בסימן טרמינלי הוא עלה. אבל עלה יכול להיות מסומן בסימן שאינו טרמינלי.

21.1.1 חזית העץ

- המילה המתקבלת ע"י שירשור סימני **העלים** של T לפי הסדר הטבעי שלהם (משמאל לימין) נקראת **חזית העץ**.
- חזית העץ היא תבנית פסוקית, כלומר, מילה ב-$(V \cup \Sigma)^*$ וזוהי גם תבנית פסוקית ב-Gהניתנת לגזירה מ-S.
- אם חזית העץ אינה מכילה משתנה דקדוקי, אזי היא מילה טרמינלית $w \in \Sigma^*$ ו-$w \in L(G)$.
- עץ שהחזית שלו היא מילה טרמינלית נקרא <u>עץ גזירה מלא</u>.
- לכל מילה $w \in L(G)$ קיים עץ גזירה יחיד T שהחזית שלו היא המילה w. לכן, עבור מילה טרמינלית $w \in \Sigma^*$:
$w \in L(G) \iff$ קיים עץ גזירה מלא בדקדוק G שילת החזית שלו היא w.

21.1.2 חד משמעיות ורב משמעיות

- ייתכן שבדקדוק מסוים יש יותר מעץ גזירה לאותה מילה, לכן דקדוק חסר הקשר G נקרא דקדוק **חד־משמעי** אם לכל מילה $w \in L(G)$ קיים עץ גזירה יחיד T שחזיתו היא המילה w. אחרת הדקדוק נקרא **רב־משמעי**.

21.1.3 גזירות קנוניות

הגדרה 21.2 יהי $G = \langle V, \Sigma, P, S \rangle$ דקדוק חסר הקשר ותהיינה $\varphi, \psi \in (V \cup \Sigma)^*$ ונניח כי $\varphi \underset{G}{\Rightarrow} \psi$ (כלומר ניתן להגיע ל-ψ מ-φע"י גזירה אחת ב-G) אזי לפי ההגדרה: $\varphi = \chi \boldsymbol{A} \rho$ ו-$\psi = \chi \boldsymbol{\alpha} \rho$ כאשר $A \to \alpha \in P$ ו-$\chi, \rho \in (V \cup \Sigma)^*$.
אומרים ש-$\varphi \underset{G}{\Rightarrow} \psi$ היא גזירה **<u>שמאלית</u> ביותר** אם $\chi \in \Sigma^*$,
ואומרים ש-$\varphi \underset{G}{\Rightarrow} \psi$ היא גזירה **<u>ימנית</u> ביותר** אם $\rho \in \Sigma^*$.

[14] מידע נוסף על עצים ועל עצים מכוונים בפרט, ניתן למצוא בסיכום של "מבוא לתאוריה של מדעי המחשב" (שנה א').

22 דקדוקים רגולריים

22.1 דקדוק לינארי ימני

הגדרה 22.1 דקדוק לינארי ימני הוא דקדוק חסר הקשר $G = \langle V, \Sigma, P, S \rangle$ שכל כללי הגזירה בו הם אחת מהצורות הבאות:

עבור: $A, B \in V$ ו־$a \in \Sigma$:

1. $A \to aB$.
2. $A \to a$.
3. $S \to \varepsilon$ - מהסוג הזה יכול להיות רק כלל אחד כמובן (וזה אומר ש־$\varepsilon \in L(G)$).

משפט 22.2 יהי Σ א"ב.
שפה $L \subseteq \Sigma^*$ היא שפה רגולרית $\iff$ קיים דקדוק לינארי ימני G כך ש־$L = L(G)$.

למשל:
ניקח את השפה הבאה:
$L = \left\{ w \in (0+1)^* \middle| w = 1(0+1)^* 0 \right\}$ כלומר, כל המילים שמתחילות ב־1 ומסתיימות ב־0 כאשר $\Sigma = \{0,1\}$.
דקדוק רגולרי ימני של השפה הנ"ל:

$$S \to 1A$$
$$A \to 0A | 1A | 0$$

מסקנה 22.3 כל שפה רגולרית היא שפה חסרת הקשר.

23 למת הניפוח לשפות חסרות הקשר

למת הניפוח לשפות חסרות הקשר מאוד דומה ללמת הניפוח הקודמת (בחלק 3), אבל עם שינוי קל בפירוק של המילה ובניסות.
<u>ניסוח הלמה:</u>
תהי L שפה חסרת הקשר מעל א"ב Σ. אזי קיים מספר טבעי n (קבוע הניפוח) כך שלכל מילה $w \in L$ המקיימת $|w| \geq n$ יש פירוק מהצורה $w = xuvzy$ כאשר $x, u, z, v, y \in \Sigma^*$ ומתקיימות שלושת הדרישות הבאות:

1. $|uzv| \leq n$.
2. $|uv| > 0$.
3. לכל $i \geq 0$ מתקיים: $xu^i zv^i y \in L$.

בשביל לזכור איך לפרק את המילה כדאי לזכור את הדבר הבא:
יש לנו $x - y$, בתוך ה־$x - y$ יש לנו $u - v$, כלומר $xu - vy$ וכעת בתוך כל אלה יש לנו z, סה"כ: $xuzvy$.
(אחרי שקצת מתרגלים את זה זוכרים את זה טוב...)
וכמו בלמת הניפוח הקודמת הרעיון כאן הוא להראות ששפה אינה ח"ה בכך שאנחנו מראים שהכלל השלישי לא מתקיים בעוד שני הכללים האחרים מתקיימים.
אנחנו בוחרים מילה כלשהי בשפה (מילה ספציפית! עם n כמו בלמת הניפוח הקודמת) ואז מניחים בשלילה שהשפה ח"ה ומראים כיצד, עבור i מסוים, זה סותר את הדרישה השלישית.

24 פישוט דקדוקים חסרי הקשר

24.1 הסרת סימנים מיותרים

הגדרה 24.1 יהי $X \in V \cup E$.
אומרים ש־X **גוזר מילה טרמינלית** (בדקדוק G) אם קיימת מילה $w \in \Sigma^*$ כך ש־$X \underset{G}{\Rightarrow}^* w$.
אומרים ש־X הוא **סימן ניתן להשגה** (בדקדוק G) אם קיימות $\alpha, \beta \in (V \cup \Sigma)^*$ כך ש־$S \underset{G}{\Rightarrow}^* \alpha X \beta$ (כלומר, X מופיע באיזו תבנית פסוקית של הדקדוק)

הערה 24.2 לפי ההגדרה - כל סימן טרמינלי גוזר מילה טרמינלית. כמו־כן S הוא תמיד סימן הניתן להשגה. אם ב־G הסימן ההתחלתי S אינו גוזר מילה טרמינלית אזי: $L(G) = \varnothing$.

לכן, באלגוריתם הבאים נציג את הדרך להסיר אותם סימנים מיותרים אשר יפשטו לנו את הדקדוק.

24.2 אלגוריתם להסרת סימנים שאינם גוזרים מילה טרמינלית

נזכור שכל סימן טרמינלי גוזר מילה טרמינלית לכן מדובר באלגוריתם הזה רק על משתנים דקדוקיים שאינם גוזרים מילה טרמינלית.
למשל, אם יש לנו כלל גזירה כזה $A \to aA | bA$ (כאשר $a, b \in \Sigma$ ו־$A \in V$) אזי A הוא סימן שאינו ניתן להשגה מכיוון שהוא אינו גוזר מילה טרמינלית.
<u>קלט:</u> דקדוק ח"ה $G = \langle V, \Sigma, P, S \rangle$.
<u>פלט:</u> הכרזה ש־$L(G)$ (שפת הדקדוק) היא ריקה <u>או</u> הצגת $G' = \langle V', \Sigma, P', S \rangle$ שכל סימן בו גוזר מילה טרמינלית אשר מקיים: $L(G') = L(G)$.

1. חישוב V':
   (א) $V' = \left\{ A \in V \middle| \exists w \in \Sigma^*,\ A \to w \in P \right\}$ (כלומר, בשלב הזה אנחנו מכניסים ל־V' את כל המשתנים הדקדוקיים ($\in V$) אשר גוזרים מילה טרמינלית).
   (ב) אם $B \to \alpha \in P$ כאשר $B \in V \backslash V'$ ו־$\alpha \in (V' \cup \Sigma)^*$ צרף את B ל־V' (כלומר, לכל $B \notin V'$ אבל $B \in V$ אם הוא מפנה למילה שמורכבת מאותיות מ־Σ או משתנים דקדוקיים מ־V' אזי נצרף את B ל־V').
   (ג) נחזור על (ב) כל עוד מתווסים איברים חדשים ל־V'.
   (ד) לבסוף, נבדוק האם $S \in V'$:
      i. לא - מכריזים ש־$L(G) = \varnothing$ ועוצרים את רוצת האלגוריתם (לא ממשיכים לשלב הבא).
      ii. כן - ממשיכים לשל הבא:
2. חישוב P':
   (א) נסיר מ־P את כל כללי הגזירה שמופיע בהם משתנה $B \in V \backslash V'$ כלומר:
   $P' = \left\{ A \to \alpha \in P \middle| A \in V',\ \alpha (V' \cup \Sigma)^* \right\}$.
   (ב) החזר את $G' = \langle V', \Sigma, P', S \rangle$.

24.3 אלגוריתם להסרת סימנים שאינם ניתנים להשגה

למשל: עבור $\Sigma = \{a, b, c\}$

$$S \to aS|Aa$$
$$A \to a|Ab|b$$
$$B \to aA|c$$

אזי B הוא מצב שאינו ניתן להשגה (כי לא ניתן להגיע אליו בשום סדרת גזירה) וכמו כן c הוא סימן טרמינלי שאינו ניתן להשגה (כי לא ניתן לגזור שום מילה שמכילה אותו).

<u>הקלט:</u> דקדוק ח"ה $G = \langle V, \Sigma, P, S\rangle$.

<u>הפלט:</u> דקדוק ח"ה $G' = \langle V', \Sigma', P', S\rangle$ כך שכל סימן $X \in V' \cup \Sigma'$ ניתן להשגה ו־$L(G) = L(G')$.

1. בניית V' ו־Σ':
   (א) $V' = \{S\}$, $\Sigma = \varnothing$.
   (ב) אם $A \to \alpha \in P$ ו־$A \in V'$ צרף ל־V' כל משתנה דקדוקי המופיע ב־α ואינו ב־V' וצרף ל־Σ' כל סימן טרמינלי ב־α שאינו ב־Σ'.
   (כלומר, עבור כל $A \in V'$ ו־ $\alpha \in (V \cup \Sigma)^*$ כך ש־ $A \to \alpha \in P$ אנחנו רואים אילו סימנים ב־α אינם ב־$V' \cup \Sigma'$ ונצרף אותם ל־V' ו־Σ' בהתאם).
   (ג) חוזרים על צעד (ב) כל עוד מתווספים איברים חדשים ל־V'.

2. בניית P' ו־G':
   (א) $P' = \{A \to \alpha \in P | A \in V'\}$.
   (ב) מחזירים כפלט את $G' = \{V', \Sigma', P', S\}$.

24.4 אלגוריתם להסרת כללי ε

הרעיון של אלגוריתם זה הוא להסיר כללים מהצורה $A \to \varepsilon$ כאשר $A \in V$.

ניתן להחליף כל כלל ε בכללים אחרים כך שהדקדוק ישאר אותו דקדוק (מבחינת השפה) רק ללא כללים מהצורה הנ"ל.

כמובן שבשפה החדשה המילה הריקה אינה כלולה.

באופן פורמלי: בהניתן דקדוק חסר הקשר $G = \langle V, \Sigma, P, S\rangle$ שיש בו כללי ε נבנה דקדוק ח"ה $G' = \langle V, \Sigma, P', S\rangle$ שאין בו כללי ε כך ש־$L(G') = L(G) - \varepsilon$.

<u>חשוב לשים לב שאם $\varepsilon \notin L(G)$ אזי: $L(G') = L(G)$.</u>

24.4.1 אלגוריתם למציאת משתנים הניתנים לאיפוס

לפני שניגש לאלגוריתם נצטרך להשתמש באלגוריתם נוסף אשר יתן לנו קבוצה: $n(G) \subseteq V$ ־ אשר אלו משתנים הניתנים לאיפוס.

הגדרה 24.3 יהי $G = \langle V, \Sigma, P, S\rangle$ דקדוק ח"ה. משתנה $A \in V$ נקרא משתנה **ניתן לאיפוס** אם $A \underset{G}{\Rightarrow}^* \varepsilon$. כלומר אם קיימת סדרת גזירה מ־A שבסופה נגזור את המילה הריקה.

$$n(G) = \left\{A \in V \middle| A \underset{G}{\Rightarrow}^* \varepsilon\right\}$$

נסמן ב־$n(G)$ את קבוצת המשתנים הניתנים לאיפוס בדקדוק G.

<u>הקלט:</u> דקדוק ח"ה $G = \langle V, \Sigma, P, S\rangle$.

<u>הפלט:</u> קבוצת כל המשתנים הניתנים לאיפוס $n(G)$.

1. $n(G) = \{A \in V | A \to \varepsilon \in P\}$ (בשלב זה מוסיפים את כל המשתנים הניתנים לאיפוס בצעד אחד).
2. אם $B \to \alpha \in P$ כאשר $B \in V$ ו־$\alpha \in n(G)^+$ (כל הסימנים ב־α הם משתנים הניתנים לאיפוס), צרף את B ל־$n(G)$ (במידה והוא לא שם).
3. נחזור על צעד 2 עד שלא יתווספו משתנים חדשים ל־$n(G)$.

<u>**כעת נראה את האלגוריתם להסרת כללי ε:**</u>

<u>קלט:</u> דקדוק ח"ה $G = \langle V, \Sigma, P, S\rangle$ שיש בו כללי ε.

<u>פלט:</u> דקדוק ח"ה $G' = \langle V, \Sigma, P', S\rangle$ ללא כללי ε אשר מקיים: $L(G') = L(G) - \varepsilon$.

1. נחשב את $n(G)$ לפי האלגוריתם הקודם.
2. $P' = \varnothing$.
3. לכל כלל גזירה $A \to X_1X_2\cdots X_n$ שאינו כלל ε, נצרף ל־P' את כללי הגזירה $A \to Y_1Y_2\cdots Y_n$ המקיימים את שלושת התנאים הבאים:
   (א) אם $X_i \in \Sigma$ או $X_i \in V - n(G)$ אז $X_i = Y_i$.
   (ב) אם $X_i \in n(G)$ אז $X_i = Y_i$ או $Y_i = \varepsilon$.
   (ג) $Y_1Y_2\cdots Y_n \neq \varepsilon$.
4. נחזיר את $G' = \langle V, \Sigma, P', S\rangle$.

הסבר לכלל השלישי:

נניח כי $V = \{A, B, C, D, E\}$, $\Sigma = \{a, b\}$

ו־$n(G) = \{B, C, E\}$, אזי, אם יש לנו כלל מהצורה:

$$A \to BCaD$$

נכניס ל־P' את הכלל:

$$A \to BCaD|CaD|BaD|aD$$

הסבר: עבור כל $X \in n(G)$ אנחנו פעם אחת מתייחסים אליו כאל ε או כאל מה שהוא ושמים ב־P' את כל האפשרויות, במקרה שלמעלה: בכלל הראשון ־ $BCaD$ לא התייחסנו לאף $X \in n(G)$ כאל כלל ε, בכלל השני ־ CaD התייחסנו ל־B כאל ε ואל C לא, בכלל השלישי בדיוק ההפך, ובכלל האחרון התייחסנו גם אל B וגם אל C כ־ε. כך, גם בכללים אחרים אנחנו צריכים לכסות את כל האפשרויות.

או אם למשל הכלל היה:

$$A \to BCE$$

אזי הכלל שהיינו מכניסים ל־P' הוא:

$$A \to BCE|CE|BE|BC|B|C|E$$

ובגלל כלל (ג) אנחנו לא נתייחס לשלושתם כאל ε כי אז נקבל כלל מהצורה $A \to \varepsilon$ ואת הכללים הללו אנחנו רוצים להסיר...

24.5 אלגוריתם להסרת כללי יחידה

בדקדוק ח"ה $G = \langle V, \Sigma, P, S \rangle$ כל גזירה מהצורה $A \to B$ כאשר $A, B \in V$ נקרא **<u>כלל יחידה</u>**.
לכל דקדוק ח"ה קיים דקדוק שקול ללא כללי יחידה שמתקבל ע"י שינוי כללי הגזירה בלבד.

הגדרה 24.4 יהי דקדוק ח"ה $G = \langle V, \Sigma, P, S \rangle$. זוג סדור (A, B) כאשר $A, B \in V$ נקרא **<u>זוג יחידה</u>** אם Bניתן לגזירה מ־A <u>תוך שימוש בכללי יחידה בלבד</u> (כמובן שישנה אפשרות שמספר הצעדים יהיה 0 ולכן, לכל $A \in V$ (A, A) הוא זוג יחידה).

הערה 24.5 אם בדקדוק אין כללי ε אזי (A, B) הוא זוג יחידה אם ורק אם $A \underset{G}{\Rightarrow}^* B$ (כי בהעדר כללי ε אין אפשרות אחרת לגזירת B מ־A מלבד השימוש בללי יחידה).

24.5.1 אלגוריתם לחישוב קבוצת זוגות היחידה $u(G)$

לפני שניגש לאלגוריתם להסרת כללי יחידה נצטרך לחשב את את קבוצת זוגות היחידה בדקדוק G: $u(G) \subseteq V \times V$.
<u>קלט:</u> דקדוק ח"ה $G = \langle V, \Sigma, P, S \rangle$.
<u>פלט:</u> $u(G)$.

1. $u(G) = \{(A, A) | A \in V\}$.
2. אם $(A, B) \in u(G)$ ו־$B \to C \in P$ נצרף את (A, C) ל־$u(G)$ (אם הוא אינו שם).
3. נחזור על צעד 2 כל עוד מתווספים איברים חדשים ל־$u(G)$.
4. נחזיר את $u(G)$.

כעת ניגש לאלגוריתם להסרת כללי היחידה:
<u>קלט:</u> דקדוק ח"ה $G = \langle V, \Sigma, P, S \rangle$.
<u>פלט:</u> דקדוק ח"ה $G' = \langle V, \Sigma, P', S \rangle$ שאין בו כללי יחידה ומתקיים: $L(G') = L(G)$.
נסמן: $A \to \alpha\star$ פירושו ש־$A \to \alpha$ <u>אינו</u> כלל יחידה.

1. נחשב את $u(G)$ ע"פ האלגוריתם שלמעלה.
2. $P' = \{A \to \alpha | A \to \alpha\star,\ (A, B) \in u(G),\ B \to \alpha \in P\}$.
3. נחזיר את $G' = \langle V, \Sigma, P', S \rangle$.

הסבר ל־2:
נניח ויש לנו את הדקדוק הבא:

$$S \to a | A$$
$$A \to Bb | c$$
$$B \to a$$

אזי $u(G) = \{(S, S), (A, A), (B, B), (S, A)\}$ ויש לנו את כלל היחידה $S \to A$,
אזי נשים לב כי $A \to Bc$ ו־$A \to c$ אינם כללי יחידה, שניהם ב־P ו־$(S, A) \in u(G)$, ולכן הכללים שנצרף הם:
$\boldsymbol{S} \to Bc$ ואת $\boldsymbol{S} \to c$ - כלומר, במקום ה־A שמנו את S...
במילים פשוטות יותר, מה שבעצם אנחנו עושים כאן הוא זה:
אם יש לנו כלל יחידה $A \to B$ אזי אנחנו מסתכלים כל הכללים שאינם כללי יחידה מהצורה $B \to \alpha$ ו"מדביקים" אותם ל־A במקום B באותו כלל $A \to B$.

לכן, אם יש לנו משהו כזה:

$$S \to A$$
$$A \to B$$
$$B \to a$$

אחרי שנריץ את האלגוריתם מה שנקבל הוא את הדקדוק הבא: $S \to a$.
(אנחנו פשוט מדביקים ב"שרשרת").

24.6 הסדר הנכון לביצוע האלגוריתמים

1. הסרת כללי ε.
2. הסרת כללי יחידה.
3. הסדרת כללים מיותרים.

אבל אפשר לעשות את זה בצורה יותר פשוטה ונוחה לביצוע:

1. הסרת סימנים מיותרים.
2. הסרת כללי ε.
3. הסרת סימנים מיותרים.
4. הסרת כללי יחידה.
5. הסרת סימנים מיותרים.

הדרך הנ"ל (השנייה) תהפוך את כל התהליך לפשוט ונוח יותר (למרות שהיא קצת יותר ארוכה).

25 הצורה הנורמלית של חומסקי (CNF)

הגדרה 25.1 דקדוק חסר הקשר $G = \langle V, \Sigma, P, S \rangle$ שכל כללי הגזירה בו הם באחת מהצורות הבאות:

1. $A \to BC$.
2. $A \to a$.

כאשר $A, B, C \in V$ ו־$a \in \Sigma$.
נקרא דקדוק **<u>בצורה הנורמלית של חומסקי</u>**.

משפט 25.2 לכל שפה חסרת הקשר L <u>שאינה</u> מכילה את ε, קיים דקדוק חסר הקשר $G = \langle V, \Sigma, P, S \rangle$ בצורה הנורמלית של חומסקי כך ש־$L = L(G)$.

25.1 בניית CNF מדקדוק ח"ה

הניתן דקדוק ח"ה $G = \langle V, \Sigma, P, S \rangle$ שמקיים $\varepsilon \notin L(G)$:

1. בלי הגבלת הכלליות נניח כי בדקדוק אין כללי ε ואין כללי יחידה (במידה ויש - נסיר אותם ע"פ האלגוריתמים שלמעלה).
2. לכל סימן טרמינלי $a \in \Sigma$:

   (א) נוסיף ל־V משתנה חדש שנסמנו S_a.

   (ב) נוסיף ל־P את הכלל הבא: $S_a \to a$.

(ג) לכל כלל גזירה ב־P מהצורה $A \to X_1 \cdots X_k$ עם $k \geq 2$ נחליף כל מופע של a בכלל גזירה זה ב־S_a.

נסמן את הדקדוק המתקבל ב־$G' = \langle V', \Sigma, P', S \rangle$.
שלב שני:
לכל כלל גזירה ב־P' מהצורה $A \to X_1 \cdots X_k$ עם $k \geq 3$:

1. נוסיף $k-2$ משתנים חדשים ל־V' ונקרא להם $Y_1, Y_2, ..., Y_{k-2}$ (לכל כלל גזירה משתנים אחרים!)
2. נחליף את הכלל $A \to X_1 \cdots X_k$ ב־ $k-1$ הכללים הבאים:
$A \to X_1 Y_1$, $Y_1 \to X_2 Y_2$, $Y_2 \to X_3 Y_3$....
עד שלבסוף נגיע ל־$Y_{k-3} \to X_{k-2} Y_{k-2}$,
$Y_{k-2} \to X_{k-1} X_k$.

נסמן את הדקדוק המתקבל אחרי ביצוע השלב הזה ב־$G'' = \langle V'', \Sigma, P'', S \rangle$ ונחזיר אותו.
כלומר אם יש לנו דקדוק ובו יש כלל גזירה מהצורה:

$$A \to BCDa$$

אזי בשלב הראשון הוא יהפוך להיות:

$$A \to BCDS_a$$
$$S_a \to a$$

ולאחר־מכן בשלב השני הוא יהיה:

$$A \to BY_1$$
$$Y_1 \to CY_2$$
$$Y_2 \to DS_a$$
$$S_a \to a$$

וקיבלנו דקדוק בצורה הנורמלית של חומסקי.

26 משפטים בקשר לתכונות סגירות של שפות חסרות הקשר

משפט 26.1 משפחת השפות חסרות ההקשר, מעל א"ב Σ, סגורה לאיחוד, לשירשור, ולאיטרציה (כלומר, היא סגורה לפעולות רגולריות).

משפט 26.2 משפחת השפות חסרות ההקשר **אינה** סגורה לחיתוך ולמשלים.

26.1 משפט חשוב על חיתוך שפה רגולרית ושפה חסרת הקשר

חיתוך של שפה **רגולרית** עם שפה **חסרת הקשר** נותן שפה **חסרת הקשר**.
כלומר:
אם L_1 רגולרית ו־L_2 חסרת הקשר (או ההפך כמובן), אזי:
$L_1 \cap L_2$ היא שפה **חסרת הקשר**.

27 פתרון בעיות הכרעה בבעיות רגולריות

בכל המקרים נתון לנו דקדוק ח"ה $G = \langle V, \Sigma, P, S \rangle$.

27.1 בעיית הריקנות

יש להכיע האם השפה $L(G)$ אינה ריקה.
פתרון: מסירים מ־G את המשתנים שאנים גוזרים מילה טרמינלית (לפי האלגוריתם שראינו למעלה) אם המתשנה ההתחלתי S נשאר בדקדוק ־ השפה לא ריקה ונחזיר "כן", אחרת נחזיר "לא".

27.2 בעיית הסופיות

יש להכריע האם השפה $L(G)$ היא אינסופית (כלומר, כוללת אינסוף מילים).
פתרון: בה"כ נניח כי ב־Gאין סימנים מיותרים, כללי ε או כללי יחידה. נבנה גרף מכוון שהצמתים בו הם המשתנים הדקדוקיים של G (אברי V) והקשתות הן כל הזוגות הסדורים (A, B) אשר מקיימים: $A, B \in V$ וקיים בדקדוק כלל מהצורה $A \to \alpha B \varphi$ כאשר $\alpha, \varphi \in (V \cup \Sigma)^*$.
נבדוק אם יש מעגל מעגל בגרף שבנינו ־ אם כן ־ השפה היא אינסופית ונחזיר "כן", אחרת נחזיר "לא".

חלק VIII
אוטומט מחסנית (PDA)

הרעיון באוטומט מחסנית הוא שיש לנו אוטומט NFA כולל מסעי ε אבל הפעם גם יש לו מחסנית עם מספר אינסופי של תאים שממנה הוא יכול כל פעם או לדחוף מילה או לקרוא אות אחת (ואז הוא מסיר אותה).

28 הגדרה פורמלית

אוטומט מחסנית (PDA) זוהי שביעייה $M=\langle Q,\Sigma,\Gamma,\delta,q_0,\perp,F\rangle$ כאשר:

- Q - קבוצה סופית של **מצבים**.
- Σ - **א"ב הקלט** (קבוצה סופית).
- Γ - קבוצה סופית של **א"ב המחסנית**. באוטומט מחסנית אנחנו יכולים לדחות למחסנית א"ב שהוא שונה מא"ב של של השפה (Σ).
- q_0 - **המצב ההתחלתי**. $q_0\in Q$.
- $\perp$ - איבר הנקרא **תחתית המחסנית**. $\perp\in\Gamma$ והוא מסמל את תחתית המסחנית. כלך עוד לא נגענו במחסנית ולא קראנו שום אות קלט - הדבר היחיד שיש במחסנית זה את $\perp$.
- F - **קבוצת המצבים המקבלים**. $F\subseteq Q$ץ
- δ- **פונקצית המעברים**: $\delta: Q\times\Sigma_\varepsilon\times\Gamma\to 2^{Q\times\Gamma^*}$, כאשר $\Sigma_\varepsilon=\Sigma\cup\{\varepsilon\}$. אנח מניחים כי הערכים של הפונקציה δ הן תת-קבוצות סופיות של $Q\times\Gamma^*$. (על הפונקצהי הזאת יוסבר קצת בהמשך).

29 סימונים

את אברי Σ (אותיות הקלט) נסמן ב- $a,b,c....$.
את אברי Σ_ε (אות קלט או ε) נסמן ב- $\sigma,\tau,...$.
את אברי Γ (א"ב המחסנית) נסמן ב- $X,Y,Z,...$.
מילים ב- Σ^* יסומנו ב-$w,u,v,...$.
מילים ב-Γ^* יסומנו ב-$\alpha,\beta,\gamma,...$.

30 הערות ופירושים

30.1 מבנה המחסנית

המסחנית ההתחלתית נראית כך:

$\perp$					

כל פעם שהאוטומט קורא אות מהקלט הוא מסיר את האות שנמצאת בראש המחסנית (תמיד זאת תהיה האות שנמצאת בראש המחסנית וזאת תמיד תהיה אות אחת בלבד).
נניח שהחלטנו לדחוף למחסנית את האות האות A (וזה כמובן אחרי קריאה של אות קלט או ε, הסברים נרחבים יותר על כך יהיו בהסברים על פונקצית המעברים δ), אזי היא המחסנית תיראה כך:

A					

מכיוון שהוא קודם כל הסיר את האות $\perp$ הוא הכניס את האות A.
למחסנית ניתן לדחוף גם מילים מ-Γ^*, לכם אם נרצה שהסימן $\perp$ ישאר נצטרך לדחוף קודם כל אותו ואז את את מה שאנחנו רוצים (חשוב לציין, שלפעמים אנחנו לא נרצה להשאיר סימן ה-$\perp$ ולכן לא חייבים תמיד שהוא יהיה בסוף המילה), לכן, אם נדחוף למחסנית $A\perp$ - אזי המחסנית תיראה כך:

A	$\perp$				

ואם עכשיו נדחוף BC אזי המחסנית תיראה כך:

B	C	$\perp$			

כי קראנו אות קלט כלשהי (או ε) ואז הסרנו את A ונשארנו רק עם $\perp$ וכעת האוטומט דחף את C למחסנית (והזיז את $\perp$ אחד קדימה) ואז האוטומט הכניס את B ודחף את $C\perp$ אחד קדימה.
כמו-כן, ניתן גם לדחוף למחסנית את ε - ואז זה אומר שהוא רק מסיר את האות שבראש המחסנית ולא מוסיף שום אות במקום.

דבר נוסף שחשוב לזכור - האוטומט יכול להתקדם רק עם אם יש לפחות אות אחת מ-Γ, **אם המחסנית ריקה - האוטומט לא יכול להתקדם**, במידה ויש עוד אותיות בקלט אזי האוטומט נתקע.
(לגבי השאלה אם הוא יקבל את המילה או לא - זה תלוי בשפה ועל זה ידובר בהמשך).

הערה 30.1 המחסנית היא בגודל אינסופי, כלומר אנחנו יכולים לדחוף אליה כמה אותיות שנרצה (או כמה מילים שנרצה, באיזה אורך שנרצה).

30.2 פונקצית המעברים δ

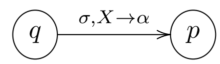
כפי שניתן לראות פונקציית המעברים מוגדרת כך:

$$\boldsymbol{\delta : Q\times\Sigma_\varepsilon\times\Gamma\to 2^{Q\times\Gamma^*}}$$

כלומר, אנחנו עוברים מהשלשה (q,σ,X) כאשר $q\in Q$, $\sigma\in\Sigma_\varepsilon$ ו-$X\in\Gamma$ לזוג סדור (p,α):

$$\delta\left(q,\sigma,X\right)\to\left(p,\alpha\right)$$

הסבר:
אם אנחנו נמצאים במצב q האות הבאה בקלט היא σ (שזה יכולה להיות אות ב־Σ או מילה ריקה - מעבר ε [בהמשך יהי הסבר לזה]) ובראש המחסנית נמצאת האות X אזי אנחנו נעבור למצב p ונדחוף למחסנית את המילה α (וזה אחרי שהסרנו את האות שנמצאת בראש המחסנית).
ניתן להביא את זה קצת יותר טוב באמצעות דיאגרמה:



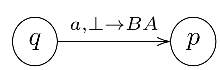
המעבר מ־q ל־p מתאפשר אם רק בשני תנאים:

1. האות הבאה שהאוטומט עומד לקרוא היא האות $a\in\Sigma$ או ε - ואז לא משנה מה האות הבאה אם בכלל ישנה.
2. בראש המחסנית יש את האות X.
3. (אנחנו כמובן במצב q).

אם אנחנו מבצעים את המעבר הזה אזי אנחנו:

1. קוראים עוד אות ממילת הקלט.
2. עוברים למצב p.
3. מסירים את האות שנמצאת בראש המחסנית ובמקומה שמים את α.

יכולים להיות כמה מעברים מ־q ל־p כאשר כל אחד תלוי באות אחרת (בדיוק כמו ב־NFA) או תלוי באות שנמצאת בראש המחסנית.
אם למשל אנחנו במצב q, האות הבאה במילה היא a ובראש המחסנית יש את האות B, אבל המעבר היחיד שיש לנו הוא:



אזי לא נוכל להמשיך וניתקע....
לשם שנוכל להמשיך אחד מהכללים של המעבר מ־q ל־p חייב להיות מהצורה: $a, A\to\cdots$ או במקום a יכול להיות ε.
כמו־כן, ייתכן כי $\delta(q,\sigma,X)=\varnothing$ - וזה בעצם כל ה־δ שלא ניתן להמשיך מהם הלאה (במקרה שלנו כל - δ אחרת ממה שכתוב)ץ

30.3 השפה של M

השפה של אוטומט מחסנית $M=\langle Q,\Sigma,\Gamma,\delta,q_0,\perp,F\rangle$ מתחלקת לשתי שפות:

1. $L_f(M)$ - זאת השפה כזאר מסתיימת מילת הקלט והאוטומט נמצא במצב מקבל. לא משנה לנו אם המחסנית ריקה או מלאה - אם מילת הקלט הסתיימה ואנחנו במצב מקבל - המילה התקבלה. חשוב עם זאת לזכור שאם המילה לא נגמרה והאוטומט ריק - אזי אנחנו נתקעים.
2. $L_e(M)$ - זאת השפה של כל המילים שמתקבלות ע"י כך שהמילה נגמרת והמחסנית ריקה ו<u>לא</u> משנה באיזה מצב אנחנו מקבל או לא מקבל.

כמובן שלא בהכרח $L_f(M)=L_e(M)$ ובהמון מקרים - $L_f(M)\neq L_e(M)$.

31 החישוב של אוטומט מחסנית

כעת נתאר באופן פורמלי את הדרך שבה אוטומט מחסנית עובד ואת האופן שבו הוא מקבל או דוחה מילה.
נתייחס כאן לאוטומט מחסנית $M=\langle Q,\Sigma,\Gamma,\delta,q_0,\perp,F\rangle$.

31.1 קונפיגורציה של M (או: תיאור רגעי)

קונפיגורציה (תאירו רגעי) C של M היא שלשה $C=(q,v,\alpha)$ כאשר:
$q\in Q$, $v\in\Sigma^*$ ו־$\alpha\in\Gamma^*$. נסמן ב־$c(M)$ את כל הקונפיגורציות של M:

$$c(M)=\left\{(q,v,\alpha)\,\middle|\,q\in Q,\ v\in\Sigma^*,\ \alpha\in\Gamma^*\right\}$$

המשמעות של $C=(q,v,\alpha)$ היא כזאת:

- q - המצב שבו האוטומט נמצא כרגע.
- v - יתרת מילת הקלט (מה שהאוטומט עוד לא קרא ממילת הקלט).
- α - תוכן המחסנית (האות הראשונה ב־α מופיעה בראש המחסנית).

הקונפיגורציה הנוכיחת של האוטומט מכילה את כל האינפורמציה הדרושה להמשך החישוב.
כלומר, ניתן להפסיק את החישוב ואז להמשיך אותו מהקונפיגורציה הנוכחית אחרי כל פרק זמן (מה שכבר היה פחות מעניין אותנו, והקונפיגורציה הנוכחית הוא כמו "סימנייה").

31.1.1 קונפיגורציה התחלתית C_0^w

בהינתן מילת קלט $w\in\Sigma^*$ **הקונפיגורציה ההתחלתית** של M על w תסומן ב־$\boldsymbol{C_0^w}$ והיא מוגדרת ע"י:

$$C_0^w=(q_0,w,\perp)$$

31.1.2 היחס $\models_M$

נגדיר יחס $\boldsymbol{\models_M}$ על הקבוצה $c(M)$ באופן הבא:
עבור: $C_1=(q_1,v_1\alpha_1)$ ו־$C_2=(q_2,v_2,\alpha_2)$ ב־$c(M)$,
אומרים ש־ $\boldsymbol{C_2}$ **(היא) קונפיגורציה עוקבת ל־$\boldsymbol{C_1}$**, ורושמים $C_1\models_M C_2$ אם האוטומט M יכול בצעד אחד (כלומר, ע"י קריאת אות אחת ממילת הקלט או ע"י ביצוע מעבר ε אחד) לעבור מקונפיגורציה C_1 ל־C_2 (באופן חוקי, לא בהכרח דטרמיניסטי לפי פונקציית המעברים).
באופן פורמלי: $\boldsymbol{C_1\models_M C_2}\iff$

1. $\alpha_1=X\beta$ ו־$\alpha_2=\alpha\beta$ כאשר $\alpha,\beta\in\Gamma^*$, $X\in\Gamma$.
2. $v_1=\sigma v_2$ כאשר $\sigma\in\Sigma_\varepsilon$.
3. $(q_2,\alpha)\in\delta(q_1,\sigma,X)$ (כי כמו שהוסבר למעלה ל־$\delta(q_1,\sigma,X)$ יש כמה אפשרויות, כלומר, מדובר בקבוצה של זוגות סדורים).

מעבר מקונפיגורציה לקונפיגורציה עוקבת הוא **צעד חישוב** של אוטומט מחסנית. נשים לב שאם C_2 היא קונפיגורציה עוקבת ל־C_1 אז בהכרח: $|v_2|=|v_1|-1$ או $|v_2|=|v_1|$ (אם עשינו מעבר ε).

נשים לב

תוכן עניינים