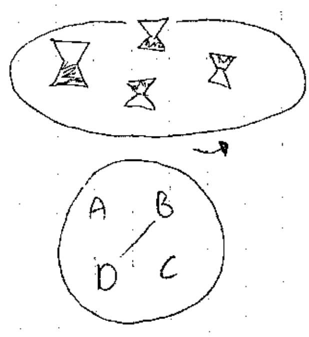
12.1.09

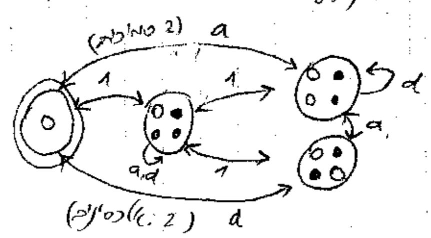
מודלים חישוביים - שיעור 11



בשיעור הקודם למדנו על אוטומטים דטרמיניסטיים ואוטומטים לא דטרמיניסטיים.

נסתכל על בעיה בה יש מגש מסתובב ו-4 כוסות בסדר לא ידוע (הפוכה או לא הפוכה). הברמן יכול להפוך כוס, שתיים בבת אחת סמוכות, או להפוך 2 כוסות באלכסון. הברמן לא יודע איזה כוס ביחס לאחרות הוא הפך. האם ניתן להגיע למצב בו כל הכוסות באותו כיוון?

נתאר את האוטומט הלא-דטרמיניסטי



להלן הגרף המתאר למעלה את הסיפור.

השאלה היא האם יש תוכנית שמכל מצב התחלתי (ולא ידוע) הברמן יגיע למצב הסופי.

כלומר, השאלה: $\exists w. \forall q. \exists x,y.\ w=xy \wedge x\in L(q)$.
במילים, השאלה האם קיים w כך שניתן לחלק אותו לרישא וסיפא כך שהרישא שייך לשפה של q עבור כל q.

הבעיה $x\in L(A)$ כריעה כי יש מספר סופי של מסלולים. גם אם היה מספר אינסופי של מסלולים (אם היה ε לדוגמא), הבעיה הייתה כריעה כיוון שניתן לתרגם את האוטומט הלא-דטרמיניסטי לאוטומט דטרמיניסטי ולבדוק.

$L(A) \stackrel{?}{=} \emptyset$ היא בעיה כריעה, כי ניתן לבדוק אם יש מצב מקבל ולוודא שקיים מסלול ממצב התחלה למצב המקבל.

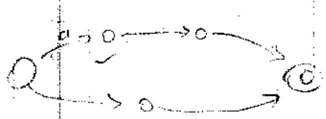
טוב כדאי לזכור!

בשבוע שעבר דיברנו גם על <u>למת הניפוח</u>, אם מילה גדולה ממספר המצבים, אז יש חלק במילה שניתן "לנפח". כלומר $w=xyz$ ב-$L(A)$ כך ש-w מספיק ארוכה, אזי גם $xy^iz \in L(A)$.

מלמת הניפוח ניתן להסיק כי $L(A)=\emptyset$ אמ"מ $L(A)$ לא מכילה מילה באורך קטן מ-$|Q_A|$. כלומר יש מספר סופי של שאלות שאנחנו יכולים לברר את הבעיה (בהינתן גודל האוטומט).

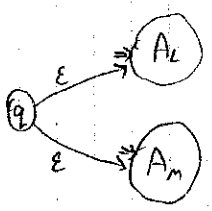
ראינו כי יש שפות כריעות, כמו שפת הפלינדרומים, שאינן רגולריות כיוון שאין אוטומט שמקבל אותן.

גם $|L|<\infty$ (כלומר באורך סופי), אז בודקים שיש תוכנית של מכונה לזיהוי אשר מקבלת אותה. כלומר, אם <u>השפה סופית אז היא גם רגולרית</u>. הסיבה לכך היא שניתן לבנות אוטומט עם מסלול לכל אחת מהמילים שבשפה.



נשים לב שיש גם הרבה שפות אינסופיות שיש להם אוטומט שמקבל אותן, כלומר הן רגולריות. בדוגמה אנו רואים את השפה $\{a^i \mid i\in\mathbb{N}\}$.

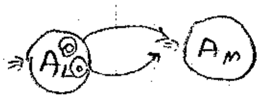
<u>שאלה:</u> בהינתן 2 שפות רגולריות L, M, האם $L\cup M$ היא גם רגולרית?



<u>תשובה:</u> כן. ל-2 השפות יש אוטומט מקבל, נוכל להוסיף מצב נוסף חדש כך שיעביר למצב ההתחלה של כל אחת מהן ע"י ε (ציור).

כלומר, <u>איחוד של שפות רגולריות</u> היא שפה רגולרית.

<u>שאלה:</u> בהינתן 2 שפות רגולריות, האם השרשור של 2 השפות היא גם שפה רגולרית?



<u>תשובה:</u> כן. ל-2 השפות יש אוטומט מקבל. נוכל לחבר את המצב סופי של האוטומט הראשון למצב ההתחלה של האוטומט השני באמצעות ε.

כלומר, <u>שרשור של שפות רגולריות</u> היא שפה רגולרית.

<u>שאלה:</u> בהינתן שפה רגולרית, האם השפה המשלימה גם רגולרית?

<u>תשובה:</u> כן. נוכל להפוך באוטומט של השפה המקורית את המצבים הלא-מקבלים למצבים מקבלים ולהפך, נצטרך לעשות זאת באוטומט <u>הדטרמיניסטי</u>.

כלומר, <u>משלים של שפה רגולרית</u> היא שפה רגולרית.

מכאן, עפ"י משפט דה-מורגן, גם <u>חיתוך של שפות רגולריות</u> היא שפה רגולרית.

הבניה של החיתוך היא עולה $2^{m\cdot L}$, כלומר זו בניה יקרה. יש עוד דרך, גם בניה ישירה במחיר $m\cdot L$ ולכן הבניה משתלמת (m ו-L הם גודל האוטומטים המקוריים).

כדי לבדוק חיתוך, אנו רוצים בעצם לבדוק במקביל את w בשני האוטומטים (ריצה במקביל כי אסור לחזור לתחילת w). כדי לעשות זאת, ניצור אוטומט חדש שמצב לכל זוג של מצבים אפשרי (זוג מצב באוטומט החדש). כך נוכל לרוץ בצורה מדויקת במקביל על שניהם.

באופן כללי, ונשים לב שהמעבר מאוטומט לא דטרמיניסטי לאוטומט דטרמיניסטי עולה 2^n, כלומר עלות מעריכית.

האם $L(A)=L(B)$ הוא בעיה כריעה בהינתן 2 שפות רגולריות? כן, כי ניתן לבדוק זאת כך:

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

← וזאת ניתן לבדוק כי ראינו סגירות של שפות רגולריות.

רוצים למצוא את האוטומט הקטן ביותר לשפה A. (ננסה לעשות זאת בשיטת המינימיזציה הברוטלית) יש לנו פרדיקט שקילות בין שפות, נוכל לבדוק את כל האוטומטים לפי סדר הגודל שלהם ונבדוק שקילות ל-A, אם נמצא, נחזיר אותו. אחרת נחזיר את A.

כלומר, <u>בעיית מציאת אוטומט מינימלי לשפה רגולרית היא כריעה</u>.

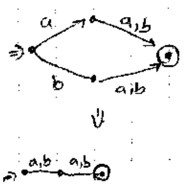
12.1.09 3

מודלים חישוביים - שיעור 11

אך האם יש שיטה יעילה יותר למציאת אוטומט מינימלי?

<u>מהו מצב נחוץ?</u> מצב שאפשר להגיע אליו ממצב התחלה ולהגיע ממנו למצב סיום.

ניתן למחוק את כל המצבים הלא-נחוצים. זה עדיין לא אומר שהגענו לאוטומט מינימלי לשפה A. (דוגמה בצד).



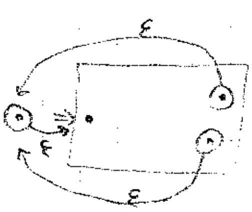
כיצד נבדוק אילו של מצבים כפולים? לכל אוטומט, יש מצבים מקבלים. עבור כל קודקוד אחר באוטומט q, נגדיר את $L(q)$ להיות שפת ההמשך של q, כלומר המילים שמ-q יגיעו למצב סופי מקבל. אם נמצא q' כך ש $L(q)=L(q')$, אזי יש כפולות. לכן, ניתן לעבור על כל המצבים בגרף ולמצוא מחלקות שקילות ע"פ שפת ההמשך, ולאחד מצבים אלו. (נשים לב שדרך נוחה ויעילה היא לסמן דווקא ע"י חיפוש מצבים לא שקולים כי אז המצבים שמובילים אליהם בנפרד גם לא שקולים)

לאחר ביצוע 2 פעולות אלו, נקבל אוטומט מינימלי לשפה.

2 אוטומטים מינימלים לאותה השפה הם זהים עד כדי שמות המצבים.

כאשר משווים 2 אוטומטים מינימלים, לכל מצב באוטומט הראשון, יש מצב מקביל אחד בדיוק באוטומט השני. הסיבה לכך היא שאלו אמורים להיות אוטומטים עבור אותה שפה ומינימלים ולכן לא יכולים להיות 2 מצבים שקולים באותו אוטומט (כי היינו מאחדים) ולא יכולים להיות מצבים שנמצאים באוטומט אחד ולא בשני (כי אז אזי שזה מצב לא נחוץ והיינו מורידים אותו, אחרת השפות אינן זהות).

האם שפות רגולריות סגורות גם ל L^*?



כן, ניתן להוסיף חיצים ממצבי הסיום למצב ההתחלה עם ε וע"י כך לבצע את הבדיקה של חלוקה למילים בשפה. כדי לאפשר את המילה הריקה, (נשנה) את מצב ההתחלה להיות מצב חדש (מקבל) שממנו נעבור למצב ההתחלה המקורי בהינתן ε.

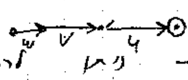
<u>ביטוי רגולרי</u> בנוי מאות ב- $\Sigma, \varepsilon, \emptyset, R+R', R\cdot R', R^*$.

לכל ביטוי רגולרי יש אוטומט סופי ולהפך. לכך שלכל ביטוי רגולרי יש אוטומט סופי, ראינו.

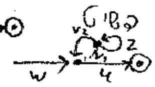
עתה נראה כיצד ניתן לתרגם כל אוטומט סופי לביטוי רגולרי.

יהי $\Sigma=\{a,b\}$, אזי $(a+b)^*$ היא שפת כל המילים מעל a,b.
השפה a^*+b^* היא שפת המילים המכילות רק a או רק b.
השפה $(a^*b)^*a^* = (a+b)^*$.

ננסה "לתרגם" כל מבנה באוטומט לחלק בביטוי. לדוגמא, מעגל פשוט מקביל לביטוי v^*. אך יכול להיות שבתוך המעגל, יש מעגל נוסף, ניתן לפרק את v ל-3 חלקים ואז לקבל $(v_1 z^* v_2)^*$.
נשים לב שמסלול פשוט



ניתן לכתוב כ wvu. לכן, האוטומט



ניתן לרשום כביטוי $w(v_1 z^* v_2)^* u$

בפעולת התרגום מתבססת ע"י שימוש $L(q,r,S)$, שפת כל המילים w שמתחילות במצב q, מסתיימות במצב r ועוברות רק דרך המצבים שבקבוצה S.

לדוגמא $L(q,r,\emptyset)$ מסמן שחייבת להיות קשת ישירה בין q ו-r. כלומר נקבל:

$$L(q,r,\emptyset) = \begin{cases} a+b+c & q \xrightarrow{a,b,c} r \\ \emptyset & \text{אין חץ בין } q \text{ ל-} r \\ \varepsilon & q=R \\ a+b+c+\varepsilon & \circlearrowleft_{a,b,c} \quad q=r \end{cases}$$

נשתמש בזה כדי לבדוק הכללה:

$$L(q,r,S\cup\{s\}) = \overbrace{L(q,r,S)}^{\text{במקרה שלא עברנו דרך צומת } s} + \overbrace{L(q,s,S) + L(s,s,S)^* + L(s,r,S)}^{\text{מקרים בהם עוברים דרך } s}$$