בשבוע שעבר התחלנו לדבר על שפות לא כריעות. הגדרנו:

$$A_{TM}=\{(P,w)|P(w)=1\}$$

$$H=\{(P,w)|P\downarrow w\}$$

השפות האלו לא ניתנות להכרעה, אבל ניתנות לזיהוי.

הגדרה

$$E=\{P|L(P)=\emptyset\}$$

כלומר שפה המורכבת מתוכניות שלא מאשרות אף מילה. דוגמאות:

$$P(x)\{while(1)\}$$

טענה

E לא כריעה

הוכחה

בשלילה, נניח E כריעה. תהי D_E תוכנית שמכריעה את E (D מהמילה Decide).
נבנה (בעזרת D_E) תוכנית $D_{\overline{A_{TM}}}$ שמכריעה את $\overline{A_{TM}}$:
$\underline{D_{\overline{A_{TM}}}(P,w)}$

1. כתוב את הקוד של התוכנית הבאה:

   $Q(x)$"

   1. הרץ את התוכנית P על הקלט w והחזר את התשובה שמתקבלת."

2. $\mathrm{return}(D_E(Q))$

דוגמה

```
P=''P(y){if y==''gila'' then return(1);while(1)}''
w=''yossi''
```

$D_{\overline{A_{TM}}}$ תיצור את מחרוזת הבאה:

```
Q(x)
{
    return(P(''yossi''));
}
P(y){if y==''gila'' than return(1);while(1);}''
```

ואז תשלח את המחרוזת לD_E.

מתקיים

אם $P\left(w\right)=1$ $\Leftarrow$ Q מאשרת כל מילה $\Leftarrow$ $L\left(Q\right)=\Sigma^*$ $\Leftarrow$ $Q\notin E$
אם $\overline{A_{TM}}$ $\Leftarrow$ $(P,w)\in\overline{A_{TM}}$ $\Leftarrow$ $P\left(w\right)\neq 1$ $\Leftarrow$ Q לא מאשרת שום מילה $\Leftarrow$ $L\left(Q\right)=\emptyset$ $\Leftarrow$
$Q\in\overline{A_{TM}}$.
קיבלנו

$$(P,w)\in\overline{A_{TM}}\Leftrightarrow Q\in E$$

כלומר יצרנו מכונה שמכריעה את $\overline{A_{TM}}$, אבל כבר הוכחנו ש$\overline{A_{TM}}$ לא כריעה $\Leftarrow D_E$ לא קיימת.

הוכחנו שE לא כריעה, אבל ההוכחה לא הייתה עובדת אם היינו משתמשים במכונה שמזהה את A. האם E ניתנת לזיהוי?

הוכחנו

$L\wedge\overline{L}$ ניתנות לזיהוי $\Leftrightarrow$ L כריעה.

מסקנה

אם L לא כריעה ו$\overline{L}$ ניתנת לזיהוי, אזי L לא ניתנת לזיהוי.

טענה

E לא ניתנת לזיהוי.

הוכחה

נראה ש$\overline{E}$ ניתנת לזיהוי. נבנה $A_{\overline{E}}$שמזהה את $\overline{E}$. $\overline{E}=\{P|L\left(P\right)\neq\emptyset\}$. $A_{\overline{E}}$ לא דט':
$A_{\overline{E}}$

1. בחר באופן ל"ד מילה w.
2. $\mathrm{return}(P\left(w\right))$

מתקיים

$L\left(P\right)=\emptyset$ $\Leftarrow$ $A_{\overline{E}}\left(P\right)$ לעולם לא תחזיר "1".
$L\left(P\right)\neq\emptyset$ $\Leftarrow$ קיימת בחירה של w שתגרום לתוכנית $A_{\overline{E}}\left(P\right)$ להחזיר "1" $\Leftarrow$ קיים חישוב שמחזיר "1" ■

הערה

$A_{\overline{E}}$ מזהה את $\overline{E}$, אבל לא מכריעה אותה, שכן אם היא לא תגיע ל$P\left(w\right)=1$ היא תמשיך לחפש עד אינסוף.

עוד דוגמה

$$EQ=\{(P_1,P_2)|L(P_1)=L(P_2)\}$$

כלומר שפת התוכניות שמאשרות את אותן מילים(כלומר מזהות את אותן שפות)

דוגמה

$$f(x)\{while(i);\}$$

$$g(x)\{if\,x=="yossi"\,return(0);else\,return(0)\}$$

טענה

EQ לא ניתן לזיהוי.

הוכחה

בשלילה נניח EQ ניתנת לזיהוי. תהי A_{EQ} תוכנית שמזהה את EQ. נבנה A_E שמזהה את E:

$\underline{A_E(Q)}$

1. ייצר את התוכנית P_2:

$$"P_2(x)\{return(0);\}"$$

2. $\text{return}(A_{EQ}(Q,P_2))$

נשים לב ש$L(P_2)=\emptyset$ ולכן $L(Q)=\emptyset \Leftrightarrow L(Q)=L(P_2) \Leftrightarrow A_{EQ}(Q,P_2)=1$.

$\Leftarrow$ A_E מזהה את E בסתירה!

$\Leftarrow$ A_{EQ} לא קיימת. ■

רדוקציה

יש שתי עולמות - בעולם אחד יש את כל התוכניות $\{Q\}$. בעולם אחרת יש את כל הזוגות של תוכניות - $\{(P_1,P_2)\}$. יש שתי תת קבוצות: $E\subseteq\{Q\}$ ו$EQ\subseteq\{(P_1,P_2)\}$.

נניח שיש פונקציה $R:\{Q\}\to\{(P_1,P_2)\}$ שמקיימת $R(Q)\in EQ \Leftrightarrow Q\in E$ - כלומר הפונקציה שומרת על שייכות לקבוצה.

נניח שאנחנו יודעים למצוא את EQ - D_{EQ}. אם יש לנו תוכנית ב$\{Q\}$, אנחנו יכולים להפעיל את הפונקציה $R(Q)$, כדי להעביר את הפונקציה ל$\{(P_1,P_2)\}$, ושם נוכל לבדוק אם $R(Q)\in EQ$, ולפי זה לדעת אם $Q\in E$.

הגדרה

תהיינה $A \subseteq X$, $B \subseteq Y$ שפות. רדוקציה מA לB הינה פונקציה $R : X \to Y$ כך שלכל $x \in X$:

$$\boxed{R\left(x\right) \in B \Leftrightarrow x \in A}$$

טענה(לא נכונה)

תהיינה $A \subseteq X$, $B \subseteq Y$ שפות. נניח שB כריעה ושישנה רדוקציה מA לB. אזי A כריעה.

הוכחה

תהי D_B תוכנית שמכריעה את B. נבנה D_A שמכריעה את A:

$D_A\left(x\right)$
{

1. y← $R\left(x\right)$
2. return($D_B\left(y\right)$)

}

בעיה!!!

יש לנו פונקציה, אבל זה לא מספיק - כי לא בטוח שאפשר לחשב אותה.

כלומר, לא מספיק שישנה רדוקציה, צריך רדוקציה ניתנת לחישוב, ואז ההוכחה נכונה:

טענה(הפעם נכונה)

תהיינה $A \subseteq X$, $B \subseteq Y$ שפות. נניח שB כריעה ושישנה רדוקציה ניתנת לחישוב מA לB. אזי A כריעה.

מסקנה

אם A לא כריעה וישנה רדוקציה ניתנת לחישוב מA לB, אזי B לא כריעה.
(/ניתנת לזיהוי)
זה נותן לנו:

מתכון להוכחה ששפה B לא ניתנת להכרעה(/זיהוי)

1. בחר שפה A לא ניתנת להכרעה(/זיהוי)
2. מצא רדוקציה חישובית מA לB.

דוגמה

$$S_1 = \{Q || L(Q)| = 1\}$$

ז"א Q מאשרת מילה אחת ויחידה.

טענה

S_1 לא ניתנת לזיהוי.

הוכחה

נסתכל ב-$\overline{H}$, לא ניתנת לזיהוי. נבנה רדוקציה ניתנת לחישוב מ-$\overline{H}$ ל-S_1. רוצים: $R: \{(P,w)\} \rightarrow \{Q\}$ כך ש $P \uparrow w \Leftrightarrow |L(Q)|$.

```
Q(x)
{
    1. if x=="dana" return(1);
    2. P(w);
    3. return(1);
}
P(...)
⋮
```

$(P,w) \in \overline{H} \Leftrightarrow P \uparrow w \Leftrightarrow Q$ מאשרת את "dana" $\Leftarrow |L(Q)| = 1$

$(P,w) \notin \overline{H} \Leftarrow P \downarrow w \Leftarrow Q$ מאשרת כל מילה (או בשורה 1 או בשורה 3) $\Leftarrow$
$L(Q) = \Sigma^* \Leftarrow Q \in S_1$

נשים לב: אנחנו לא מריצים את Q, אנחנו רק כותבים אותה.

יצרנו רדוקציה ניתנת לחישוב מ-$\overline{H}$ ל-S_1, ולכן מכיוון שאנחנו לא יכולים לזהות את $\overline{H}$, אנחנו גם לא יכולים לזהות את S_1

עוד דוגמה

$$SUB = \{(Q_1, Q_2) | L(Q_1) \subsetneq L(Q_2)\}$$

טענה

SUB לא ניתנת לזיהוי.

הוכחה

ברדוקציה מ$\overline{A_{TM}}$ משלים.

$\overline{A_{TM}}$ לא ניתנת לזיהוי. נראה רדוקציה חישובית מ$\overline{A_{TM}}$ לSUB.

כלומר נרצה $R:\{(P,w)\}\to\{(Q_1,Q_2)\}$ כך ש $P(w)\neq 1 \Leftrightarrow L(Q_1)\subsetneq L(Q_2)$

$$Q_1 = "Q_1(x)\{return\,(P(w));\}"$$

$$Q_2 = "Q_2(x)\{return\,(1);\}"$$

נשים לב ש$L(Q_2)=\Sigma^*$. מתקיים:

אם $(P,w)\in\overline{A_{TM}} \Leftarrow P(w)\neq 1 \Leftarrow L(Q_1)=\emptyset \Leftarrow L(Q_1)\subsetneq L(Q_2) \Leftarrow$ $(Q_1,Q_2)\in SUB$

מצד שני, אם $(P,w)\notin\overline{A_{TM}} \Leftarrow P(w)=1 \Leftarrow L(Q_1)=\Sigma^* \Leftarrow L(Q_1)=L(Q_2)$ $\Leftarrow (Q_1,Q_2)\notin SUM$

כנדרש.