מודלים חישוביים - הרצאה 6

שרון מלטר, אתגר 17

29 במאי 2024

1 מכונת טיורינג

ראינו בהרצאות הקודמות שאחת החולשות של אס"ד היא חוסר הזיכרון שלו. כעת נלמד על מודל חדש, הדומה לאס"ד, אך הוא בעל זיכרון דינמי.
במקום שכל תו יכול להיקרא רק פעם אחת ואז הוא נאבד לנצח, במ"ט אנו מקבלים את הקלט כסרט וניתן לעבור עליו קדימה ואחורה. הקלט מסתיים ברווח כך שניתן לזהות איכן הוא נגמר.
אנו חושבים על זיכרון המ"ט כך:
הזיכרון מנוהל כרשימה מקושרת של תאים, כך שבכל תא יש מצביע לתא הבא ולתא הקודם. זהו זיכרון דינמי שתומך בפעולות הבאות;

- זוז תא אחד שמאלה (לא אפשרי כאשר זהו התא הראשון)
- זוז תא אחד ימינה (אם אין תא מימין, נוצר תא חדש במקום זה)

בכל פעם שאנו נמצאים בתא מסוים, מותר לנו לקרוא את תוכנו ולשנות אותו.
במ"ט דטרמיניסטי, יש לזוז בכל צעד ימינה או שמאלה (זה לא מאוד משנה בפועל אבל כן מקל על הגדרות)
אנחנו נתעניין גם במ"ט דטרמיניסטיות וגם בלא דטרמיניסטיות.

דרך אחרת לחשוב על מבנה הזיכרון הוא טייפ סלילים שיכול לזוז שמאלה או ימינה.

1.1 מעבר במ"ט

יהי מעבר במ"ט ממצב q_1 למצב q_2 נניח שניתן לבצע מעבר זה אם קוראים את התו a ואז אנו רוצים להחליף אותו בתו b ולאחר מכן לעבור שמאלה או ימינה.
המעבר יסומן כך: $a \to b, L/R$

חשוב:

- הא"ב של הקלט לא חייב להיות גם ה"ב של הפלט. כלומר, ניתן לשנות את תווי הקלט לתווים שלא נמצאים בא"ב שלו.
- אם לא קיים מעבר אפשרי, המילה נדחית
- מספר קריאות תווי הקלט יכול להיות גדול מאורכו

בהמשך להערה האחרונה-
בכל מ"ט נרצה שמילים משפת המ"ט יעצרו במכונה ויתקבלו, שמילים שלא משפת המ"ט יעצרו במכונה ויידחו ושאף מילה לא תרוץ במכונה לנצח.

1.2 הגדרה פורמלית

מכונת טיורינג M היא שביעייה $M = (Q, \Sigma, \Gamma, q_0, \delta, q\_accept, q\_reject)$ כאשר;

- Q קבוצת מצבים סופית

- Σ א"ב הקלט (א"ב סופי)
- Γ א"ב הפלט (א"ב סופי)
- $q\_accept, q\_reject \in Q$, $q\_accept \neq q\_reject$ מצב מקבל ומצב דוחה
- δ פונקציית המעבר.
כאשר המכונה דטרמיניסטית מתקיים;

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

$$\delta(q, a) = (q', b, L/R)$$

וכאשר המכונה לא דטרמיניסטית מתקיים ישנן מספר אפשרויות לקריאת התו a מהמצב q;

$$\delta(q, a) = \{(q', b, L), (q'', c, R)...\}$$

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

כאשר P מסמנת את קבוצת החזקה של $Q \times \Gamma \times \{L, R\}$.

שימי לב ♡
לפי ההגדרות שהצגנו;

- אין מעברי ϵ במ"ט
- צריך להחליט מה קורה כאשר המ"ט עומד על התו השמאלי ביותר בסרט ורוצה לעבור שמאלה

נשאר לנו רק להגדיר את $L(M)$.
מדוע הגדרה זו מסובכת יותר? ישנן דקויות שאנחנו לא רגילים אליהן, בגלל לולאות אינסופיות.
נלמד שני מושגים חדשים:

אחד עבור מ"ט לשפה L כך שלכל $w \in L$ מסלול החישוב של M על w מסתיים בקונפיגיורציה מקבלת, לכל $w \notin L$ מסלול החישוב של M על w מסתיים בקונפיגיורציה דוחה ואף מילה לא נכנסת ללופ אינסופי. במקרה זה אומרים שהמ"ט M מכריעה את השפה L.
ואחד עבור מ"ט שיכולה להיכנס ללולאה אינסופית. השפה שלה תוגדר כך; $L(M)$ היא שפת כל המילים שעוצרות במכונה ומתקבלות בה. במקרה זה אומרים שמ"ט M מקבלת את השפה L.
כלומר: מילים שרצות במכונה בלופ אינסופי אינן מתקבלות, גם אם הן נשארות במצב מקבל.

1.3 דוגמה

יהי א"ב הקלט $\Sigma = \{0\}$ ושפה של מילים מהצורה $\{0^{2^n} : \quad n \geqslant 0\}$

האבחנה הראשונה שתעזור לנו היא שמספר הוא חזקה של 2 אם ניתן לחלק אותו ב־2 ולקבל מספר טבעי זוגי עד שמקבלים 1.
כיצד נחלק מספר במ"ט? ־ נעבור על טייפ הקלט ובכל פעם נמחק את האפסים האי־זוגיים (נרשום x במקומם).
כך מחקנו מחצית מאיברי המספר. אם נקבל בסוף אפס יחיד, הקלט שייך לשפה.
נעשה זאת בכך שנחליף ב־ x כל אפס שנמצא במקום אי־זוגי, כאשר האפסים לא מופרדים או מופרדים ע"י x ־ים.

2 התזה של צ'רץ' וטיורינג

זהו משפט יסוד שקבעו אלן טיורינג ואלונזו צ'רץ', שאומר כך;

- כל מה שניתן לחישוב ניתן לחישוב על ידי מכונת טיורינג
- כל מה שניתן לחישוב על ידי מחשב שיהיה בעתיד ניתן לחישוב על ידי מכונת טיורינג
- כל בעיה שניתן לפתור בצורה שיטתית על ידי אלגוריתם ניתן לפתור בעזרת מכונת טיורינג

כלומר, אם אנחנו מאמינים בתזה של צ'רץ' וטיורינג, אנו מאמינים שמכונת טיורינג היא מחשב. מה שמעלה את השאלה - אילו בעיות מחשבים יכולים לפתור? כלומר, אילו שפות מ"כ יכולה לקבל?

2.1 נכונות התזה

אבל לפני כן, נצטרך להשתכנע בתזה.
או במילים אחרות, נצטרך להשתכנע בכך ש-

- יש קומפיילר שלוקח כל פונקציה בשפת C (שמקבלות כקלט מחרוזת ומחזירות ערך בוליאני כן/לא) ומתרגם אותה למ"ט
- כל פונקציה שניתן לכתוב כשפה עילית ניתן לממש במ"ט
- כל מחשב שאנו מכירים, ניתן לסמלץ בתוכנה בשפה עילית

נשים לב ששלושת טענות אלה נבנות אחת על השנייה.

עם זאת, דיברנו גם על מחשבים שיהיו בעתיד. כיצד אנו יודעים שהם שקולים למ"ט?
כלומר, מה עם כל אלגוריתם שיהיה בעתיד?

בשביל להפיג את הפחד תחילה נציין שהתזה נוצרה בשנת 1936 ומאז פרסומה לא הייתה תקופה בה טען מתמטיקאי שהיא שגויה בעקבות סיבה כזו או אחרת.
כמו כן, כל הרעיון של אלגוריתם הוא הרכבה של פקודות ברורות ומוגדרות היטב מסט סופי של פעולות מותרות, וזה אפשרי בשפת עלית או במ"ט.
אבל אז מה שונה באס"ד ובא"מ? - שני מודלים אלה באו מתחום החומרה, מ"ט מזוהה יותר עם תחום התוכנה.

נתחיל בשכנוע.
ניקח אדם אשר לא רגיל לתכנת. כיצד נשכנע אותו בתזה?
אם לקחנו אס"ד והוספנו לו מחסנית / סרט, כלומר הוספנו לו זיכרון, הוא יהיה יותר חזק ויוכל לבצע יותר פעולות. אז התהייה הראשונה שיש לענות עליה היא האם אנו בטוחים שלא ניתן לחזק מ"ט בעזרת הוספת רכיבים. נפריך זאת כך;

יהי מ"ט שיש לה CPU ו- k סרטים. נוכיח שהוספת סרטים לא תועיל למ"ט.
הקלט ניתן על הסרט הראשון וצעד (או מעבר) פועל כך: מסתכלים על המצב והתו הנוכחי בסרטים. כפונקציה על מה שראינו מבצעים מעבר. משנהים את המצב ולכל ראש (מה שמצביע על התו הנוכחי בסרט) כותבים תו וזזים.

<u>טענה:</u> לכל מ"ט M עם מספר סרטים קיימת מ"ט M' חד-סרטית שמסמלצת אותה. כך שכל מילה $w \in L(M)$ שעוצרת ומתקבלת ב- M גם תיעצר ותתקבל ב- M', אם w עוצרת ונדחית ב- M היא תיעצר ותידחה גם ב- M' ואם היא נכנסת ללופ אינסופי ב- M היא תיכנס גם ללופ אינסופי ב- M'. כל עוד שלושת התכונות האלה מתקיימות, המ"ט השנייה מסמלצת את המקורית.

<u>הוכחה:</u>
כל אחד מכם יכול לפתור תוכנית בשפת C שמסמלצת ריצה של מ"ט k סרטית ואז לקמפל למ"ט.
נראה כיצד בונים מכונה כזו.
עבור מ"ט M נגדיר "קונפיגיורציה" שהיא תמונת מצב רגעית של החישוב (על מנת לדעת כיצד להמשיך את ההרצה אם היא הופסקה באמצע) ניתן לומר שמסלול חישוב הוא סדרה של קונפיגיורציות.

במ"ט שלנו הקונפיגיורציה תכיל מצב, מה כתוב על הסרט (עד המקום הראשון שממנו והלאה יש קלט) ואיפה נמצא הראש.

הערה: אילו היינו מגדירים פורמלית ריצה (מסלול חישוב) של מכונת טיורינג, היינו אומרים שלמ"ט דטרמיניסטית יש מסלול חישוב סופי או אינסופי של מצבים ובמ"ט לא־דטרמיניסטית יש מספר מסלולי חישוב.

בשביל לסמלץ את המ"ט ה־ k־ סרטית, נשמור סרט יחיד, שבו מופרדים באמצעות התו המיוחד $\#$ תכני הסרטים השונים.
בשביל זאת, בהתחלה נשמור את מילת הקלט בחלק הראשון של הסרט היחיד, ואז $k-1$ האשטאגים שיפרידו בין ההצגות של שאר הסרטים ואחרי כל האשטאג יהיה רווח שיסמן את סופו הנוכחי.
בכל צעד של ה־ k־ סרטית, יסומלצו צעדים של החד־סרטית. ניתן להוכיח באינדוקציה שבצעד ה־ i, כאשר כל הצעדים הקודמים סומלצו נכון, יבוצע צעד מקביל במ"ט החד־סרטית.
הצעד המקביל הינו פשוט לשנות את הקטע בסרט היחיד שמסמלץ סרט מסוים מבין k הסרטים כפי שמשנים את אותו סרט נפרד.
אך ניתן לשים לבעיה נוספת: כיצד נוסיף תאים לסרטים המסומלצים?

ניעזר בפרוצדורה של $shift-right$, שמטרתה להוסיף תא מימין והקוד שלה מורכב כך;

$$\delta(q_{in},a)=(q_a,\$,R)$$

$$\delta(q_a,b)=(q_b,a,R)$$

$$\delta(q_a,\sqcup)=(q',a,L)$$

$$\delta(q',a)=(q',a,L)$$

$$\delta(q',\$)=(q_{out},\$,R)$$

(שימי לב $\heartsuit$־ התו $\$$ הוא תו מיוחד שמציין את התחלת הקלט.)

נסביר באופן מילולי את הפרוצדורה:
תחילה, מהמצב הראשון, q_{in} נעים ימינה כל עוד נקרא התו a ובכל פעם כותבים במקומו $\$$. כאשר נתקלים בתו b כותבים דווקא a ועדיין נעים ימינה.
ממשיכים כך עד שנתקלים בתו $\sqcup$ שכאמור מייצג את סוף המילה. כותבים במקומו את התו a ונעים שמאלה.
ממשיכים לנוע שמאלה עד שמגיעים לתו $\$$ וממנו ממשיכים ימינה שוב.

בסה"כ בכל צעד של M, M' מבצע $O(v,t)$ צעדים.
ישנם t צעדים כאלה ולכן M' רצה ב־ $O(k\cdot t\cdot t)=O(kt^2)$ צעדים כאשר המ"ט ה־ k־ סרטית רצה t צעדים.
אם אנחנו חושבים ש־ k קבוע אזי בסך הכל מבצעים $O(t^2)$ צעדים.

בסך הכל קיבלנו שניתן לסמלץ מ"ט k־ סרטית בעזרת מ"ט חד־סרטית. ניתן להוכיח שמ"ט זו צריכה $O(n^2)$ צעדים.

3 סגירויות של שפות כריעות ומתקבלות

הגדרנו מקודם את המושגים "מ"ט מכריעה את השפה L" ו "מ"ט מקבלת שפה L".
הנושא הבא שלנו הוא ניסיון להבין אילו שפות הן כריעות ואילו מתקבלות.

שימי ♡
תהי L שפת המ"ט M. אם M מכריעה את השפה, אזי $L(M)$, קבוצת המילים שעוצרות ומתקבלות ב־ M, מקיימת $L(M) = L$. כמובן שבמקרה זה גם M מקבלת את L.

קודם כל, נבדוק מה הן הסגירויות של שפות כריעות ושפות ניתנות לקבלה.

- משלים
שפות כריעות סגורות תחת משלים, מכיוון שניתן להחליף בין $qaccept$ ל־ $qreject$.
אך השלמה לא סגורה עבור שפות מתקבלות מכיוון שמילה שנכנסת ללולאה אינסופית ולא תתקבל תעשה זאת גם אם נחליף בין המצב המקבל לדוחה.
אזי שפות כריעות סגורות תחת משלים ושפות מתקבלות לא.

- חיתוך
אם L_1, L_2 כריעות, אזי יש מ"ט M_1, M_2 שמכריעות את L_1, L_2 ואפשר גם לבנות מ"ט M שמכריעה את $L_1 \cap L_2$. נריץ את M_1 על מילה ואחרי שסיימה נריץ אותה ב־ M_2. נקבל אותה במ"ט החיתוך רק אם היא התקבלה בשניהם.
בשפות מתקבלות, מאחר שכל המילים שמתקבלות עוצרות, ניתן להריץ כל מילה בשני מ"ט שמקבלים שתי שפות וכך לדעת האם היא מתקבלת בשניהם. עם זאת, כאשר נבדוק מילה שלא שייכת לאחת מהן ייתכן שניכנס ללולאה אינסופית (מצב זה חוקי, אך לא כדאי)
אזי שפות כריעות ומתקבלות סגורות תחת חיתוך.

- איחוד
בשפות כריעות נפעל כמו עם חיתוך, אך נקבל כל מילה שמתקבלת באחד משני המ"ט.
כאן ישנו מצב יותר טריקי, מכיוון שאם מריצים מילה בשני מ"ט שמקבלים שתי שפות שונות, ייתכן שהיא תיכנס ללואה אינסופית באחת מהן. לכן פשוט נבנה מ"ט דו־סרטית שמקבלת מילה כאשר היא מתקבלת בסרט אחד (כמו באוטומט מכפלה)
אזי שפות כריעות ומתקבלות סגורות תחת איחוד.

יש לקח חשוב שניתן ללמוד מהדוגמאות שהצגנו:
אם אנו לא יודעים אם שפה היא כריעה או מתקבלת, כאשר נשתמש במ"ט שלה, ייתכן שניכנס ללולאה אינסופית. וכפי שצוין, לא נוכל לזהות כאשר זה קורה.

4 מ"ט אוניברסלית

עד כה, דיברנו על מ"ט בעזרת רכיבי חומרה (סרט, CPU, ראש חץ...) אך כפי שנאמר, מדובר במודל מעולם התוכנה. מעבר לכך, לפי התזה של צ'רץ' וטיורינג, תוכנית מחשב שקולה למ"ט.
כפי שאנו יודעים, תוכנית מחשב יכולה לקבל כקלט תוכנית מחשב אחרת. ומתי תוכניות מחשב ייִרצו לקבל כקלט תוכנית מחשב?

- כאשר מערכות הפעלה רוצות להריץ אותן
- כאשר נרצה לבדוק אותן (אולי נרצה לבדוק שהיא לא יכולה להיכנס ללואה אינסופית)

אפשר לקודד מ"ט בתור מחזרוזת (מעל הא"ב $\Sigma = \{0,1\}$) אנחנו נבחר איזשהו קידוד של $M = (Q, \Sigma, \Gamma, \delta, q_0, qaccept, qreject)$ נקודד כל אחד משבעת האיברים באופן בינארי (אנו מקודדים תוכניות כמחרוזות כדי שיהיה אפשר להריץ אותן) נסמן את המחרוזת המתקבלת מקידוד אל מ"ט ע"י $< M >$.

מ"ט אוניברסלית היא מ"ט u שמקבלת בתור קלט מחרוזת $x =< M, w >$;
היא מצפה לקבל קידוד של מ"ט M ומחרוזת w. על הקלט $< M, w >$ (קידוד של M ושל) המ"ט מסמלצת את ריצה M על w.

- אם M עוצרת ומקבלת את w אז u עוצרת ומקבלת את $< M, w >$

- אם M עוצרת ומקבלת את w אז u עוצרת ומקבלת את $< M, w >$
- אם M נכנסת ללולאה אינסופית על w אז u נכנסת ללולאה אינסופית על $< M, w >$

שאלה ראשונה שיכולה לעלות מהצגת המ"ט האוניברסלית, היא למה דווקא $< M, w >$ הוא הסימון לקלט.
לא עדיף לסמן ב־ $w < M >$ למשל, כדי לייצג את הפעולה שנעשית על המילה המקודדת?
כשאנו כותבים $< M, w >$ אנו מתכוונים לכך שמי שמסתכל במחרוזת יודע איפה נגמר M ואיפה מתחיל w, מה שלא בא לידי ביטוי בסימון כמו $w < M >$.

השאלה הבאה שאמורה לעלות בראש, היא האם בכלל קיימת מ"ט אוניברסלית?
אך זוהי שאלה טריוויאלית, מכיוון שאנו יודעים שמעבדים קיימים.
לכן מעכשיו כשנדבר על מ"ט M ונריץ עליה את המילה w נניח שניתן למ"ט u את הקלט $< M, w >$.

סיכום:
מלצייר דיאגרמות של מ"ט אנחנו עכשיו מרשים לעצמנו;

1. לכתוב תוכנית ב־ C
2. לכתוב פסאודו קוד (כי אפשר לתרגם אותו למ"ט)
3. כאשר אנו מקבלים כקלט מ"ט (או תוכנית מחשב בשפת C) ניתן להריץ אותה באמצעות מ"ט אוניברסלית

4.1 $HALT$ ו־ $ACCEPT$

לפי הגדרת המ"ט האוניברסלית, היא לא תהיה כריעה עבור כל שפה.
לכן ישנן שתי הגדרות נוספות;
$ACCEPT$ היא $L(u)$, שזוהי קבוצת ה־ $< M, w >$ שבהם M היא מ"ט ו־ w היא מילה שמתקבלת ב־ M.
כמו כן $HALT$ היא קבוצת ה־ $< M, w >$־ים כך ש־ M היא מ"ט ו־ w היא מילה שעוצרת בה.

האם $HALT$ כריעה?
זוהי שאלה הרת גורל, מכיוון שאם כן, אז כל תוכנית ניתן להריץ במ"ט שמכריעה את $HALT$ וכך לדעת האם היא תגרום ללולאה אינסופית. כך נוכל לדעת מראש על כל בעיה של לולאה אינסופית.
בהמשך נוכיח שגם $HALT$ וגם $ACCEPT$ אינן כריעות.

האם $HALT$ ו־ $ACCEPT$ כן מתקבלות;
u מקבלת את $ACCEPT$ כאשר כש־ M נכנסת ללולאה אינסופית על w אז u נכנסת ללולאה אינסופית על $< M, w >$ וכשהיא עוצרת ומקבלת מילה גם u מקבלת אותה.
באופן דומה, $HALT$ מתקבלת כאשר כש־ M נכנסת ללולאה אינסופית על w אז u נכנסת ללולאה אינסופית על $< M, w >$ וכשהיא עוצרת על מילה גם u עוצרת עבורה.

5 דוגמאות

5.1

תהי השפה L שפת הקידודים $< M >$ כך שלכל אחד מהם קיימת מילה שעוצרת ומתקבלת בהם w המקיימת $|w| \leqslant 1,000$. נוכיח ש־ L מתקבלת.
נבנה מ"ט עבור L, אשר נקרא לה N;
נבנה מ"ט k־ סרטית, כך ש־ k הוא מספר המילים הקיימות שאורכן קטן־שווה ל־ $1,000$. נכתוב אותה בצורת פסאודו־קוד שמקבל את הקלט x, כאשר $x =< M >$.

הפסאודו־קוד:

1. נעבור על כל המילים שאורכן i מ־ $i = 1$
2. כל מילה נריץ למשך i צעדים.
   אם בסוף (או לפני) אותם צעדים היא עוצרת ומתקבלת ב־ M, נקבל את $< M >$.

3. $i++$

4. חזור לשלב 1

שימי ♡
מאחר שיש לכל היותר i צעדים, תמיד תחזור אלינו השליטה ולא ניכלא ללולאה אינסופית. ה־ i המקסימלי יכול להיות גדול ככל שנרצה.

אנו רוצים להוכיח ששפת הקידודים שמתקבלים בפסאודו־קוד והשפה L שקולות.

<u>הוכחה:</u>
תחילה נוכיח ש־ $L(N) \subset L$
קידוד של מ"ט M מתקבל רק אם קיימת מילה באורך קטן־שווה ל־ $1,000$ שמתקבלת בה. לכן $< M >\in L$ כנדרש.

כיוון שני: נוכיח ש־ $L \subset L(N)$.
כלומר, אנו רוצים להוכיח שאם $w \in L$ אזי; w מתקבלת ב־ N.
יהי $< M >\in L$ ותהי $w' \in L(M)$ מילה באורך של לכל היותר $1,000$ כך שהיא נעצרת ומתקבלת ב־ M לאחר i^* צעדים (וזהו המספר המינימלי של צעדים למילה זו)
לכל $i < i^*$ כאשר מריצים את N על M ובכניסה ללולאה יש לנו i, נגיע לשלב שבו מקדמים את i באחד. יגיע הרגע שבו $i = i^*$, אז אחת המילים שנשקלות היא w' והיא מתקבלת ב־ M לאחר i^* צעדים, כך ש־ $< M >$ תתקבל ב־ N, כנדרש.

5.2

תהי L שפת קידודי המ"ט שמקבלות מילה כלשהי. כלומר;

$$L = NOT - EMPTY = \{< M >: M \neq \varnothing\}$$

נוכיח ש־ $NOT - EMPTY$ ניתנת לקבלה, נעשה זאת כמובן בעזרת בניית מ"ט N שמקבלת אותה.
לא נוכל לבנות מ"ט באותה הדרך שעשינו קודם, מכיוון שקבוצת כל המילים שנצטרך לבדוק איננה סופית (לא מדובר במילים עד אורך מסוים)
במקום זאת, הפסאודו־קוד יכול להיראות כך;

1. $i = 1$

2. נריץ את כל המילים $|w| \leqslant i$ למשך i צעדים. אם M עצרה וקיבלה בשבילם, נקבל את $< M >$.

3. $i++$

4. חזור לשלב 2

נוכיח ש־ $L(N) = NOT - EMPTY$

קל להוכיח את ש־ $L(N) = NOT - EMPTY$ מאחר שכל מ"ט שמקבלת מילה כלשהי לאחר מספר כלשהו של צעדים מתקבלת וכל המ"ט שמקבלות בהכרח מקבלות מילה כלשהי.