סיבוכיות – הרצאה 1

כל העולם הוא חישוב אחד גדול

22.02.11

Church-Turing Thesis

- כל שפה שניתנת לפתרון בעולם האמיתי (הפיזיקאלי) ניתנת לפתרון ע"י מכונת טיורינג.
- כל תהליך פיזיקאלי ניתן לסמלץ על מכונת טיורינג.

נסביר בהמשך ביותר פירוט.

חישוב

בצורה הפשוטה ביותר: יש מצב. יש כלל התקדמות.
נרצה לתאר גם את המצב וגם את כלל ההתדמות הכי פשוט שאפשר.

מכונת טיורינג

- סרט עבודה. יש לו התחלה ואין לו סוף.

במחשבים אמיתיים הזכרון סופי ויש גישה random access כלומר שניתן לגשת לכל תא בזכרון. במכונות טיורינג הזכרון (הסרט) לא חסום אבל הגישה היא סידרתית.

- ראש קורא שיכול לזוז ימינה או שמלה. זה מתאר מצב.
- כלל התקדמות שלנו יהיה אוטמוט. מה היתרון? הוא סופי.
  - סט מצבים S.
  - מצב התחלתי $q_{init} \in S$.
  - 2 מצבים מסיימים $q_{accept}, q_{reject} \in S$.
  - האלגוריתם: $\delta : \Sigma \times S \to \Sigma \times S \times \{L, R\}$

  וזה מתאר בדיוק מה המכונה עושה בכל מצב.

האלגוריתם הוא סופי. לתאר מכונת טיורינג אפשר ע"י מספר סופי של ביטים. כלומר, יש לנו אלגורתים סופי שמתאים לכל גודל של קלט.
הנקודה הזאת לא אוטומטית. יש מודלי חישוב בהם זה לא כך, לדוגמא כשבונים מעגל חומרה הוא טוב רק לגדול קלט ספציפי.
היינו רוצים להשוות בין שני מודלי חישוב:

1. מודל חישוב אוניפורמי – פותרים עם מכונות טיורינג. כשצצה בעיה נתכנן מכונת טיורינג שתפתור אותה..
2. מודל חישוב לא אוניפורמי – יהיה לנו אוסף של מעגלים C_n לכל n. לכל אורך קלט מותר לתכנן מעגל חומרה עבור גודל קלט נתון.

איזה מודל יותר חזק? המודל הלא אוניפורמי. יכול לפתור את בעיית העצירה. מצד שני מכונת טיורינג יכולה לקבל מכונת טיורינג אחרת ולסמלץ.

נחזור למכונות טיורינג.
הראש הקורא מתחיל על התא השמאלי של סרט העבודה. המצב הראשון הוא q_{init} הקלט $x_1, \ldots, x_n$ כתוב בתחילת סרט העבודה ואחריו יש סמבול מיוחד $\perp$. לשם נוחות נניח גם שכל שאר התאים בסרט העבודה מאותחלים ל- $\perp$.

הגדרה. מכונת טיורינג M עוצרת על קלט x אם כשנתחיל במצב ההתחלתי אז אחרי מספר סופי של צעדים נגיע למצב מסויים - q_{accept} או q_{reject}.

הגדרה. זמן הריצה במקרה של עצירה הוא מספר צעדי המכונה עד שהיא עוצרת.

הגדרה. M מקבלת (דוחה) את x אם M עוצרת על x ומגיעה למצב q_{accept} (q_{reject}).

הגדרה. שפה $L \subseteq \{0,1\}^*$ היא תת קבוצה של כל הקלטים האפשריים.

הגדרה. נאמר שמכונת טיורינג M פותרת שפה L אם לכל קלט x M עוצרת על x וכן M מקבלת $x \in L$ $Leftrightarrow$.

$x \in L$ המכונה עוצרת ועונה כן.
$x \notin L$ המכונה עוצרת ועונה לא.
אפשר להגדיר שפה שמקבלת כקלט M ו- x ומחזירה את התשובה הבאה: אם M עוצרת על x אז (M, x) בשפה ואחרת לא.
אפשר להגדיר מכונת טיורינג שמקבלת (M, x) (כאשר M היא כמובן מחורזת המייצגת מכונת טיורינג ו- x הוא קלט) ואם M עוצרת על x ומקבלת אז היא עונה כן, אם M עוצרת על x ודוחה אז היא עונה לא. אם M לא עוצרת אז המכונה שלנו לא תעצור.
מוכנה זו לא פותרת את השפה כי לא תעצור לכל קלט. זה בדיוק כמו מחשב שמריץ תוכנה.
היינו יכולים להגדיר מודל חישוב שיש בו מספר סרטי חישוב. טיורינג שם לב שמכונת טיורינג יכולה לסמלץ מכונת חישוב עם k סרטים. באופן דומה אלפבית $\{0,1,\perp\}$ יכולה לסמלץ אלפבית בכל גודל. למעשה אפשר לסמלץ כל קוד של כל שפה עילית.
האם יש משהו שמכונת טיורינג לא יכולה לחשב?
נגדיר שפה $Halt$: נאמר כי $(M, x) \in Halt$ אם מכונת טיורינג M עוצרת על הקלט x.
נוכיח שאף מכונת טיורינג לא יכולה לפתור את $Halt$.

הוכחה. נניח ש- M פותרת את $Halt$. נבנה מכונה A שעושה את הדבר הבא:
בהינתן קלט x נפרש את x כמכונת טיורינג. נריץ את M על x (למעשה שואלים האם x עוצר על הקלט x). אם כן, נכנס ללולאה אינסופית, אם לא נעצור.
האם A עוצרת על A?
אם A עוצרת על A, נקרא ל- M על הקלט (A, A), M פותרת את בעיית העצירה ולכן M תעצור בזמן סופי ותחזיר כן ולכן A תכנס ללולאה אינסופית וזו סתירה. מצד שני, אם A לא עוצרת על A אז M תחזיר לא ולכן ולכן A תעצור. בסתירה.
לכן אין M הפותרת את בעיית העצירה. □

למרות שאין מכונת טיורינג שפותרת את בעיית העצירה אפשר לפתור אותה עם סדרת מעגלים.

הגדרה. נאמר ששפה L נפתרת ע"י מעגל C_n אם לכל קלט x באורך n. $C_n(x) = true$ אם $x \in L$ ו- $C_n(x) = false$ אם $x \notin L$.

זה מודל חישוב לא אוניפורמי. לכל קלט יש "מתכון" אחר.

טענה. כל שפה L אפשר לפתור ע"י סדרת מעגלים C_n.

הוכחה. לכל n נכתוב את טבלת האמת המתאימה ל- 2^n הקלטים האפשריים. ואז פשוט נממש מעגל שיפתור באמצעות הטבלה. □

מכונות טיורינג, לעומת זאת, מוגבלות כי אנחנו דורשים אלגוריתם שיש לו ייצוג סופי וגם שיעבוד לכל קלט.
Church-Turing Thesis $\Leftarrow$ כל בעיה שאפשר לפתור בעולם האמיתי אפשר לפתור ע"י מכונת טיורינג.

הגדרה. מכונת טיורינג M רצה בזמן $T(n)$ אם היא עוצרת לכל קלט ומספר צעדי ה- δ על כל קלט $x \in \{0,1\}^*$ עד ש- M עוצרת הוא לכל היותר $T(n)$.

הגדרה. $Time(T(n))$ - אוסף כל השפות שיש מכונת טיורינג שפותרת אותן בזמן $O(T(n))$.

נרצה להראות שאם $T(n) >> t(n)$ אז $Time(t(n)) \subsetneqq Time(T(n))$
ראינו כבר מכונת טיורינג A שמקבלת כקלט M ו-x כך שאם M עוצרת על x נענה מה ש-M עונה ואחרת נעשה משהו. אפשר לעשות סימלוץ כזה בזמן שהוא ריבועי בזמן ריצת M על x.

משפט. לכל $t(n) \geq n$, אז $T(n) = O(t^2(n)) \Rightarrow Time(t(n)) \subsetneqq Time(T(n))$.

אבחנה. מכונת טיורינג A שמקבלת קלט (M,x) ועונה את התשובה ש-M מחזירה על x. אז $A \in Time(t(n)^2)$ כאשר $M \in Time(t(n))$. מדוע? כי היא תעצור אחרי לכל היותר $t(n)$ צעדי סימולציה. מצד שני, אפרנו שיקח לכל היותר $t(n)^2$ לסמלץ $t(n)$ צעדי המכונה המסומלצת.

הוכחה. נגדיר מכונה B: בהינתן קלט (M,x) נסמלץ את M על x $t(n)$ צעדים. אם M עצרה אז B תחזיר את ההפך. אחרת נדחה.
מהאבחנה $B \in Time(T(n))$. מצד שני האם B שייכת ל- $Time(t(n))$? אם כן, אז יש מכונת טיורניג A שתמיד עוצרת בזמן $t(n)$ ועושה בדיוק את מה ש-B עושה.
נריץ את B עם הקלט (A,A). B תסמלץ את A על הקלט (A,A) למשך $t(n)$ צעדים. אבל $A \in Time(t(n))$ ולכן במסגרת מספר צעדים זה הסימלוץ של A יסתיים, נקבל את התשובה של A על A ונענה הפוך. אז B על (A,A) עונה:

$$B(A,A) \overset{1}{=} \neg A(A,A) \overset{2}{=} \neg B(A,A)$$

1. כי תמיד עוצר.
2. כי שם A ו- B מחזירים אותו הדבר.

סתירה. □

הערה. אם מכונת טיורינג פותרת שפה אז היא תמיד עוצרת. אז M מגדירה את השפה $\{x \in \{0,1\}^* | \text{M accpets x}\}$

פורמלית לא היינו בסדר כי אמרנו שמכונת טיורינג שייכת למשפחת שפות. המשמעות היא כפולה: שהמכונה עוצרת בזמן $t(n)$ וכן שמגדירה שפה שניתן לפתור ב- $t(n)$. כמובן שייתכן שאפשר לפתור בזמן קצר יותר.
מוסר ההשכל מכל סיפור הוא שאם למכונה יש מספיק זמן לסמלץ מכונה ולהגיד את ההפך אז היא לא יכולה להיות במחלקת הזמן. הטיעון הוא למעשה חזק יותר ונרצה להרחיב אותו.

הגדרה. מכונת אורקל היא מכונת טיורינג M שיש לה סרט עבודה, מצבים S, אלפבית Σ וגם סרט שני ומצב מיוחד שנקרא לו q_{querry}. הסמניטיקה היא כמו קודם, מלבד זה שהכשהמכונה נמצאת במצב q_{querry} אז מופעלת שאילתא על מה שרשום בסרט השאילתא ומתקבלת תשובה של כן או לא.

כלומר המכונה יכולה לרוץ כמו קודם אבל גם לדוגמא, לשאול האם המכונה M תעצור על x. אנחנו לא יודעים מאיפה התשובה מגיעה אבל היא מקבלת תשובות.
עם התשובה אפשר לעשות מה שנרצה: לכתוב לסרט, לעבור בהתאם למצב אחר וכו'. כמובן ששאילתא היא צעד דלתא ולכן עלותה 1.

הגדרה. תהי $O \subseteq \{0,1\}^*$ שפה (ייתכן אינה קריאה). נגיד ששפה L שייכת למחלקה $Time^O(t(n))$ אם קיימת מכונת אורקל M כך שאם התשובה לשאילתא y לפי שייכות ל-O (כלומר $y \in O$ התשובה היא כן, $y \notin O$ התשובה היא לא) מקבלת את L בזמן $t(n)$.

דוגמא.

$$Halt \in Time^{Halt}(n)$$

כי צריך להעתיק לסרט. אפשר גם להגיד $Halt \in Time^{Halt}(1)$. בכל מקרה לא ניתן לעשות את זה עם מכונת טיורינג.

כלומר אורקל אומר שבהינתן פתרון של בעית אורקל באיזה זמן נוכל לפתור בעיה (אולי כתלוי בזמן שלוקח לאורקל).
ראינו שאם בזמן $T(n)$ אפשר לסמלץ מכונות טיורינג שרצות בזמן $t(n)$ אז

$$Time(t(n)) \subsetneqq Time(T(n))$$

ואמרנו שאם $T(n) \geq n$ אז אפשר לסמלץ בזמן $T(n) = t^2(n)$.
באותה צורה לכל אורקל O אפשר לסמלץ מכונת אורקל ל-O שרצה בזמן $t(n)$ עם מכונת אורקל ל-O בזמן $T(n)$.

מסקנה. לכל אורקל O:

$$Time^O(t(n)) \subsetneqq Time^O(T(n))$$

הפרזה *relativise*.

הגדרה. יהיו A, B מחלקות חישוב. אפשר להגדיר A^O, B^O מחלקות חישוב - כל שפה שאפשר לפתור ע"י מ"ט השייכת ל-A או ל-B בהתאמה שגם מותר לה לבצע שאילתות לאורקל O. לעיתים מעבר כזה הוא עדין. בעיקר עם בעיות זכרון.
נניח $A \subseteq B$ (כל שפה שאפשר לפתור ע"י מכונה מ-A אפשר לפתור גם ע"י מכונה מ-B).
נאמר כי ההכלה $A \subseteq B$ היא $Relativise$ אם לכל אורקל O $A^O \subsetneqq B^O$.

בהמשך נראה לדוגמא שקיים אורקל O כך ש- $P^O = NP^O$ וקיים O עבורו $P^O \subsetneqq NP^O$.
לכן לא נוכל לפתור $P = NP$ ע"י סימלוץ מתוחכם.