מודלים חישוביים	19.5.2025

הרצאה 8: סיבוכיות

Based on previous iterations of this course, given by Nir Bitansky, Rotem Oshman, Iftach Haitner, and Omer Paneth.

מרצה: אורי שטמר

בשני השיעורים הקודמים דיברנו על בעיית החישוביות – ניסינו להבין איזה שפות ניתנות להכרעה ע"י מ"ט בזמן סופי. לא היה אכפת לנו בכמה זמן, רק האם אפשר לעשות את זה. גם לא היה אכפת לנו יותר מדי באיזה מ"ט מדובר (חד-סרטית, רב-סרטית, דטר', מטל"ד). לא היה אכפת לנו כי מבחינת הכרעה כל המכונות האלה הן שקולות ויכולות להכריע בדיוק את אותן שפות (הראינו זאת ע"י סומולציות).

מעכשיו עד סוף הקורס נתרכז בשאלת הסיבוכיות: נדון רק בבעיות כריעות ומה שנרצה להבין זה כמה "משאבים" צריך כדי לעשות את זה. כמה זמן? כמה זיכרון? אקראיות? היום נתחיל לדבר על משאב הזמן.

דוגמה לניתוח זמן ריצה:

<u>דוגמה:</u> $L = \{0^k\,1^k : k \geq 0\}$
כמה "זמן" (כלומר צעדים) נדרש למ"ט חד-סרטית על מנת להכריע את L ?

נתחיל עם מ"ט נאיבית המכריעה את L:

<u>M בהינתן קלט $x \in \{0,1\}^*$:</u>

- נסרוק את הסרט ואם 0 מופיע מימין ל- 1 אז נדחה. $O(n)$
- כל עוד הסרט מכיל גם 0 וגם 1: $O(n) \times$
  נסרוק את הסרט ונמחוק 0 יחיד ו- 1 יחיד $O(n)$
- נקבל אם"ם לא נותרו 0 או 1 על הסרט $O(n)$

סה"כ זמן ריצה על קלט באורך n: $O(n^2)$

מוסכמה: n תמיד יסמן את אורך הקלט

<u>הגדרה:</u> תהי $T: \mathbb{N} \to \mathbb{N}$ ותהי M מ"ט דטרמיניסטית. נאמר כי M רצה בזמן $T(n)$ אם לכל $n \in \mathbb{N}$ ולכל קלט x באורך n מתקיים ש- $M(x)$ מבצעת לכל היותר $T(n)$ צעדים (מעברי δ) בטרם עוצרת.

<u>הגדרה:</u> לכל פונקציה $T: \mathbb{N} \to \mathbb{N}$ נסמן:

$$DTime(T(n)) := \left\{ L(M) : \begin{array}{c} M \text{ מ"ט } \textbf{חד-סרטית} \\ \text{עם זמן ריצה } O(T(n)) \end{array} \right\}$$

כפי שנראה בהמשך, למרות שמ"ט חד-סרטית ורב-סרטית יכולות להכריע בדיוק את אותן שפות, זה קורה לא באותו זמן ריצה. כלומר זמן החישוב שלנו יהיה מאוד תלוי במודל בו אנחנו עובדים. לכן לצורך ההגדרה הזאת אנחנו חייבים לבחור מודל חישוב ספציפי ולדבוק בו. בלי סום סיבה מיוחדת אנחנו נבחר מ"ט חד-סרטית.

הסיבה שמגדירים מחלקות זמן עד כדי פקטור קבוע $O(T(n))$ ולא $T(n)$ היא על מנת שמוכל להתעלם משאלות לא מהותיות בנוגע למודל ספירת הזמן, למשל האם קריאה וכתיבה במסגרת מעבר δ צריכים להיספר בנפרד.

<u>דוגמה:</u> עבור $L = \{0^k\ 1^k : k \geq 0\}$ ראינו ש- $L \in DTime(n^2)$.
לפי הגדרה, ז"א שגם מתקיים $L \in DTime(T(n))$ לכל פונקציה $T(n) = \Omega(n^2)$

האם ניתן להכריע את L מהר יותר?

נתאר מ"ט חד-סרטית M' שמכריעה את L מהר יותר:

<u>M בהינתן קלט $x \in \{0,1\}^*$:</u>

- נסרוק את הסרט ואם 0 מופיע מימין ל- 1 אז נדחה. $O(n)$
- כל עוד הסרט מכיל גם 0 וגם 1: $O(\log n)\times$
  - נסרוק את הסרט ואם המספר הכולל של 0 ו- 1 אי-זוגי אז נדחה $O(n)$
  - נסרוק את הסרט ונמחק כל 0 שני וכל 1 שני (החל מהראשון) $O(n)$
- נקבל אם"ם לא נותרו 0 או 1 על הסרט $O(n)$

סה"כ זמן ריצה: $O(n\log n)$

<u>מסקנה:</u> $L \in DTime(n\log n)$

האם ניתן להכריע את $L = \{0^k\ 1^k : k \geq 0\}$ אפילו מהר יותר? לא עם מ"ט חד-סרטית:

<u>משפט:</u> אם $L \in DTime(o(n\log n))$ אז L רגולרית.

לא נוכיח את המשפט.

תזכורת: $g(n) = o(f(n))$ אם לכל קבוע $c > 0$ קיים קבוע n_c כך שלכל $n \geq n_c$ מתקיים $g(n) < c \cdot f(n)$

<u>מסקנה מהמשפט:</u> אם יש לנו שפה שאפשר להכריע אותה בזמן למשל $n\log\log n$ אז היא רגולרית ואז בעצם אפשר להכריע אותה בזמן $O(n)$ ע"י האוטומט שלה...

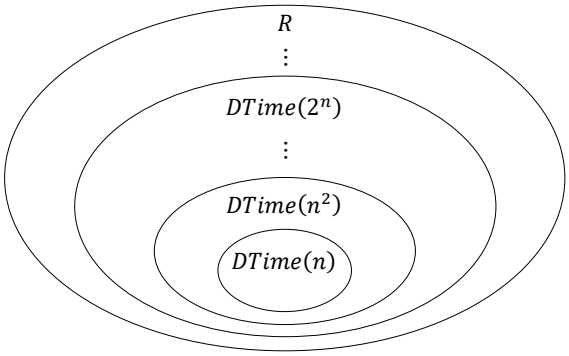
בהמשך נראה איך להכריע את L מהר יותר בעזרת מ"ט דו-סרטית.

היררכיית זמן

ראינו שפה ב- $DTime(n\log n)$ שאינה ב- $DTime\left(o(n\log n)\right)$.

עכשיו נראה משפט "היררכיית זמן" שמכליל את זה מאוד. בפרט, אנחנו נגלה שלכל מספר שלם $j>1$ יש שפה L כך ש- $L\in DTime\left(n^j\right)$ אבל $L\notin DTime\left(n^{j-1}\right)$. כלומר,
$$DTime(n)\subsetneq DTime(n^2)\subsetneq DTime(n^3)\subsetneq DTime(n^4)\subsetneq\cdots$$

בציור:



בפרט, ז"א שקיימת סדרה אינסופית של שפות שדורשות יותר ויותר זמן להכרעה.

למעשה אנחנו נראה אפילו משהו יותר חזק: לכל פונקציה $T:\mathbb{N}\to\mathbb{N}$ "מספיק נחמדה" ולכל פונקציה $t:\mathbb{N}\to\mathbb{N}$ המקיימת $t(n)\ll T(n)$ עם פער "מספיק גדול", קיימת שפה L כך ש-
$$L\in DTime\left(T(n)\right)\setminus DTime\left(t(n)\right)$$

נתחיל מלפרמל את המושג של פונקציה "מספיק נחמדה". בשיעור שעבר הגדרנו את המושג של <u>פונק' חשיבה</u> שהיא פונק' שיש מ"ט המחשבת אותה. היום זה לא יספיק לנו ונדרוש משהו יותר חזק:

<u>הגדרה:</u> פונקציה $T:\mathbb{N}\to\mathbb{N}$ היא <u>פונקציה חשיבה בזמן</u> אם קיימת מ"ט שבהינתן 1^n מחשבת את הקידוד הבינארי של $T(n)$ בזמן $O\left(T(n)\right)$.

מה ההבדל בין <u>פונק' חשיבה בזמן</u> לבין ההגדרה של <u>פונק' חשיבה</u> מהשיעור הקודם?

1. עכשיו אנחנו מדברים רק על פונקציות $T:\mathbb{N}\to\mathbb{N}$ שממפות מספר למספר. בשיעור שעבר לא בהכרח חשבנו על התחום והטווח של הפונק' החשיבות שראינו כמספרים.
2. כדי שפונק' T כזאת תהיה <u>חשיבה בזמן</u> צריך להתקיים: (א) יש מ"ט שמחשבת אותה, כלומר מ"ט שבהינתן קלט n מחזירה לנו את $T(n)$. ובנוסף (ב) המ"ט הזאת מחשבת את $T(n)$ "מהר". כמה מהר? תוך לכל היותר $O\left(T(n)\right)$ צעדים.

שימו לב: בהגדרה האחרונה אנחנו משתמשים ב $T(n)$ פעמיים: כדי ש- T תהייה חשיבה בזמן צריכה להיות מ"ט שהפלט הרצוי שלה הוא $T(n)$ ובנוסף $O(T(n))$ גם חוסם את זמן הריצה שלה.

למה זה הגיוני להשתמש ב- $T(n)$ פעמיים ככה? בהמשך נבנה מכונות שרצות בזמן $T(n)$ ולפעמים נהיה צריכים להניח שהמכונה יודעת כמה צעדים מותר לה לעשות, כלומר יודעת לחשב את הערך של $T(n)$ במסגרת תקציב זמן הריצה שיש לה. אינטואיטיבית, אם למכונה מותר לרוץ $T(n)$ צעדים, אבל כדי לחשב את הערך של $T(n)$ לוקח למשל $T^2(n)$ זמן, אז המכונה הזאת שאמורה לרוץ בזמן $T(n)$ אפילו לא יכולה להבין מזה $T(n)$ במסגרת תקציב זמן הריצה שיש לה...

<u>עוד הערות:</u>

① באנגלית פונק' כאלה נקראות *time-constructible*

② כל פונק' חשיבה בזמן שאיננה קבועה מקיימת $T(n)=\Omega(n)$ כי צריך $\Omega(n)$ זמן כדי לקרוא את הקלט.

③ למשל, $T(n)=\lfloor \log n \rfloor$ או $T(n)=\lfloor n^{0.9} \rfloor$ אינן חשיבות בזמן (קטנות מידי...)

④ כל פונקציה "סבירה" שהיא לפחות $n\log n$ תהיה חשיבה בזמן. כל פונקציות הזמן שנעבוד איתן יהיו חשיבות בזמן. למשל $\dots, n^k, 2^{\lfloor\sqrt{n}\rfloor}, 2^{\lfloor\log^2 n\rfloor}$

⑤ אינטואיציה: בזמן $O(n\log n)$ ניתן להמיר קלט 1^n לייצוג הבינארי שלו (באורך $\log n$ ביטים). לאחר מכן נוכל לבצע פעולות חשבון "סבירות" בזמן פולינומי בזמן $\log n$.

⑥ דוגמה לפונק' חשיבה שכנראה אינה חשיבה בזמן:

$$f(n)=\begin{cases} 2n & \text{, הייצוג הבינארי של } n \text{ הוא קידוד מ"ט שעוצרת תוך } 2^n \text{ זמן} \\ 2n+1 & \text{, אחרת} \end{cases}$$

<u>משפט היררכיית הזמן:</u> לכל $T(n)$ חשיבה בזמן ולכל פונקציה $t(n)=o(T(n)/\log T(n))$ מתקיים:

$$DTime(t(n)) \subsetneq DTime(T(n))$$

אנחנו נניח ש- T,t הן פונק' מונוטוניות, כלומר $T(n+1)\geq T(n)$.

<u>מסקנה:</u> לכל זוג קבועים $1\leq c<d$ מתקיים:

$$DTime(n^c) \subsetneq DTime(n^d)$$

רעיון הוכחת משפט היררכיית הזמן: נסתכל על בעיה מהסגנון:

בהינתן קידוד $\langle M,x\rangle$, האם $M(x)$ עוצרת תוך T צעדים?

אינטואיטיבית, אם יש לי תקציב זמן ריצה $t\approx T$ אז אני יכול לסמלץ את ריצת $M(x)$ בעזרת מ"ט אוניברסלית למשך T צעדים ולראות אם עוצרים. אבל אם יש לי רק תקציב $t\ll T$ אז תהיה לי בעיה לעשות את זה.

בשביל לפרמל את הרעיון הזה אנחנו צריכים לחדד כמה דברים לגבי מ"ט אוניברסליות. ספציפית: כמה זמן לוקח לי לסמלץ T צעדים של מ"ט M? האם זה בדיוק T? בערך T?

הערה: מומלץ לראות את ההקלטה מהכיתה לפני קריאת הוכחת משפט היררכיית הזמן כאן.

לפני ההוכחה:

- בהרצאה 6 הוכחנו שקיימת מ"ט אוניברסלית U שיכולה לסמלץ כל מ"ט אחרת בהינתן קידוד $\langle M,x\rangle$. לא שמנו על זה את האצבע בהרצאה 6, אבל בבניה שהראינו התקיים שזמן הריצה של $U(\langle M,x\rangle)$ היה לינארי בזמן הריצה של $M(x)$.
- כדי לפשט את הבניה, בהרצאה 6 השתמשנו ב 3 סרטים. אפשר להפוך את U שראינו למ"ט חד-סרטית תוך כדי שזמן הריצה נשאר לינארי.

משפט: קיימת מ"ט אוניברסלית חד-סרטית U כך שלכל מ"ט חד-סרטית M וקלט x, אם $M(x)$ עוצרת תוך t צעדים אז $U(\langle M,x\rangle)$ עוצרת תוך $c_M\cdot t$ צעדים לכל היותר, כאשר c_M תלוי בקידוד $\langle M\rangle$ בלבד.

כלומר, זמן הריצה של U לינארי בזמן הריצה של M, עד כדי קבוע שתלוי בקידוד של M (אבל לא בקידוד של x)

- בבניה שנראה אנחנו צריכים מ"ט אוניברסלית עם תכונה נוספת: בנוסף לקלט $\langle M,x\rangle$ אנחנו נגדיר לה גם "תקציב צעדים" והיא תצטרך לסמלץ את $M(x)$ למשך $\bar{t}$ צעדים בלבד. לצורך כך, המ"ט האוניברסלית הזאת צריכה לתחזק איזשהו מונה, וזה מגדיל את זמן הריצה שלה במשהו כמו $\log\bar{t}$. אז למרות שהמ"ט הזאת מסמלצת רק $\bar{t}$ צעדים של $M(x)$, הסימלוץ הזה יקח משהו כמו $\bar{t}\cdot\log\bar{t}$ צעדים. נסמן את המ"ט הזאת כ- U_{timer}

משפט: קיימת מ"ט חד-סרטית U_{timer} כך שבהינתן קלטים $t\geq 0$ בייצוג בינארי וקידוד $\langle M,x\rangle$ מתקיים:

- אם $M(x)$ מקבלת תוך t צעדים אז $U_{\text{timer}}(t,\langle M,x\rangle)$ מקבלת
- אם $M(x)$ דוחה או לא עוצרת תוך t צעדים אז $U_{\text{timer}}(t,\langle M,x\rangle)$ דוחה

בנוסף, $U_{\text{timer}}(t,\langle M,x\rangle)$ עוצרת תוך $c_M\cdot t\log t$ צעדים, כאשר c_M תלוי בקידוד $\langle M\rangle$ בלבד.

נשים לב ש- U_{timer} עוצרת בזמן $O(t\log t)$, אבל הקבוע שה $O()$ מסתיר תלוי בקידוד של המ"ט M. כדי להתחמק מזה, נוכל להפעיל את U_{timer} לא ישירות על M אלא על U "הרגילה" כלומר ללא טיימר. נקבל:

מסקנה: עבור קלטים $t\geq 0$ בייצוג בינארי וקידוד $\langle M,x\rangle$, כאשר מפעילים את $U_{\text{timer}}(t,\langle U,\langle M,x\rangle\rangle)$:

- אם $M(x)$ מקבלת תוך t/c_M צעדים אז $U(\langle M,x\rangle)$ מקבלת תוך t צעדים ואז U_{timer} מקבלת
- אם $M(x)$ דוחה או לא עוצרת תוך אם t/c_M צעדים אז $U(\langle M,x\rangle)$ דוחה או לא עוצרת תוך t צעדים ואז U_{timer} דוחה.

בנוסף, U_{timer} עוצרת תוך $c_U\cdot t\log t=O(t\log t)$ צעדים לכל היותר.

הוכחת משפט היררכיית הזמן:

תהי T פונק' חשיבה בזמן.

נגדיר מ"ט בשם *Flip* אשר בהינתן קידוד $\langle M,0^k\rangle$ מסמלצת את ריצת M על 0^k במשך לכל היותר $T(n)$ צעדים והופכת את ההחלטה שלה. פורמלית,

Flip בהינתן קלט w באורך n:

1. נחשב את $\bar{t}(n) = T(n)/\log(T(n))$ % אנחנו מסמנים את זה ב $\bar{t}$ כי זה חסם עליון על t
2. אם $w = \langle M, 0^k \rangle$ עבור מ"ט M ו- $k \in \mathbb{N}$ אז נמשיך. אחרת נדחה.
3. נריץ את $U_{\text{timer}}(\bar{t}(n), \langle U, \langle M, 0^k \rangle \rangle)$ ונקבל אם"ם U_{timer} דוחה

<u>טענה 1:</u> $L(Flip) \in DTime(T(n))$

צעד 1: דורש $O(T(n))$ כי T חשיבה בזמן.
בזמן $O(T(n))$ ניתן לחשב את $T(n)$ בבינארי. זה מיד מספר לנו גם את $\lfloor \log(T(n)) \rfloor$ לפי המיקום של ה-1 "הגדול" ביותר בייצוג הבינארי. למשל עבור 9 מתקיים 1001 ולכן הלוג (בבסיס 2) הוא 3 (ערך שלם תחתון). כל שנותר הוא לחלק בין 2 מספרים ביצוג בינארי, מה שלוקח זמן פולינומי באורך הייצוג, כלומר פולינומי ב $\log(T(n))$ ולכן זניח ביחס ל $T(n)$.

צעד 2 דורש $O(n)$ זמן (עבור קידוד סביר)

צעד 3 דורש $O(\bar{t}(n) \cdot \log(\bar{t}(n))) = O(T(n))$ זמן לפי התכונות של U_{timer}

<u>טענה 2:</u> $L(Flip) \notin DTime(t(n))$
נקבע מ"ט A הרצה בזמן $O(t(n))$.
עבור פרמטר k נסתכל על הקלט $w = \langle A, 0^k \rangle$. זהו קלט באורך $n = |w| = w = \Theta(|\langle A \rangle| + k)$.
נראה כי עבור k מספיק גדול מתקיים ((*))

$$w = \langle A, 0^k \rangle \notin L(Flip) \quad \Leftrightarrow \quad w = \langle A, 0^k \rangle \in L(A)$$

(מה שבפרט מראה ש- $L(Flip) \neq L(A)$)

כדי להראות את ((*)), כל מה שאנחנו צריכים להראות זה שעבור k מספיק גדול מתקיים ש- $U(\langle A, 0^k \rangle)$ עוצרת תוך $\bar{t}(n)$ צעדים, כי אז

- אם $\langle A, 0^k \rangle \in L(A)$ אז בצעד 3 של Flip נקבל ש- U_{timer} עוצרת ואומרת "כן" ולכן Flip אומרת "לא" ולכן $\langle A, 0^k \rangle \notin L(Flip)$
- אם $\langle A, 0^k \rangle \notin L(A)$ אז בצעד 3 של Flip נקבל ש- U_{timer} עוצרת ואומרת "לא" ולכן Flip אומרת "כן" ולכן $\langle A, 0^k \rangle \in L(Flip)$

ולמה שעבור k מספיק גדול יתקיים ש- $U(\langle A, 0^k \rangle)$ תעצור תוך $\bar{t}(n)$ צעדים?

- אנחנו מניחים ש- $A(0^k)$ עוצרת תוך $O(t(n))$ צעדים.
- לכן, כפי אמרנו, $U(\langle A, 0^k \rangle)$ עוצרת תוך $O(c_A \cdot t(n))$ צעדים
- מכיוון שהתיאור של A לא תלוי באורך הקלט שלו, אז c_A לא תלוי ב k (או ב n). לכן כאשר מגדילים את k, כלומר מגדילים את n, נקבל ש $c_A \cdot t(n)$ קטן אסימפטוטית מתקציב זמן הריצה $\bar{t}(n)$ ולכן $U(\langle A, 0^k \rangle)$ עוצרת תוך $O(t(n))$ צעדים.

תלות זמן הריצה במודל החישוב

הפער של $\log n$ במשפט היררכיית הזמן נובע מהתקורה של סימולציה אוניברסלית ע"י מ"ט חד-סרטית. התקורה תלויה במודל. בפרט, עבור מ"ט רב-סרטית, ניתן לסמלץ בזמן לינארי ובפרט להוכיח היררכיית זמן הדוקה יותר. כעת נראה דוגמה נוספת לכך.

<u>דוגמה:</u> עבור השפה $L = \{0^k\ 1^k : k \geq 0\}$ ראינו כי
$L \in DTime(n \log n)$ אבל $L \notin DTime(o(n \log n))$

נתאר מ"ט <u>דו-סרטית</u> שמכריעה את L מהר יותר:

M' בהינתן קלט $x \in \{0,1\}^*$:	
• נסרוק את הסרט ואם 0 מופיע מימין ל- 1 אז נדחה.	$O(n)$
• נעביר את כל ה-0-ים מסרט הקלט לסרט השני	$O(n)$
• נעבור במקביל על שני הסרטים ונמחוק במקביל 0 מסרט אחד ו- 1 מהסרט השני	$O(n)$
• נקבל אם"ם לא נותרו 0 או 1 בסרטים	$O(n)$
	$O(1)$

סה"כ זמן ריצה: $O(n)$

<u>מסקנה:</u> קיימות שפות שניתן להכריע עם מ"ט דו-סרטית מהר יותר מכל מ"ט חד-סרטית.

אבל כמה יותר מהר?

בהרצאה 5 הוכחנו את המשפט הבא: לכל מ"ט רב-סרטית M הרצה בזמן $T(n) \geq n$ קיימת מ"ט חד-סרטית M' המסמלצת אותה ורצה בזמן $O(T^2(n))$.

כלומר התקורה של מ"ט חד-סרטית על פני מ"ט רב-סרטית היא לכל היותר $O(T^2(n))$. אפשר להראות שהתקורה על פני מודל ה RAM היא לכל היותר $O(T^3(n))$. באופן כללי, נוכל לחכות כל מודל דטרמיניסטי סביר (לדוגמה פייתון) בעזרת כל מודל אחד בתקורה פולינומית (כלומר $O(T^c(n))$ עבור $c > 0$ קבוע). המצב שונה עבור חיקוי של מודלים לא-דטרמניסטיים בעזרת מודלים דטרמיניסטיים.

זמן ריצה לא דטרמיניסטי

<u>תזכורת:</u> הגדרנו קונפיגורציה של מ"ט $c = uqv$.
עבור מטל"ד N וקלט x סימנו ב $T_{N,x}$ את עץ החישוב של N על x.
לכל קונפיגורציה c_a בעומק d בעץ יש "כתובת" $\alpha \in \{1,2,\dots,C_N\}^d$

- $N(x)$ מקבלת אם יש ב- $T_{N,x}$ עלה מקבל
- $N(x)$ דוחה אם $T_{N,x}$ סופי ללא עלים מקבלים
- אחרת, $N(x)$ לא עוצרת

נכליל את ההגדרה של זמן ריצה גם למטל"ד:

<u>הגדרה:</u> תהי $t:\mathbb{N}\to\mathbb{N}$ פונקציה ותהי N מטל"ד. N רצה בזמן $t(n)$ אם לכל $n\in\mathbb{N}$ ולכל x באורך n, עץ הקונפיגורציות $T_{N,x}$ הוא בעומק $t(n)$ לכל היותר.

בפרט, מטל"ד שרצה בזמן $t(n)$ תמיד עוצרת כי עץ החישוב תמיד סופי.

<u>הגדרה:</u> תהי $t:\mathbb{N}\to\mathbb{N}$. נסמן:

$$NTime(t(n)) \coloneqq \left\{ L(M) : \begin{matrix} M \text{ מטל"ד חד-סרטית} \\ \text{עם זמן ריצה } O(t(n)) \end{matrix} \right\}$$

<u>טענה:</u> כל מטל"ד בעלת זמן ריצה $t(n)$ ניתנת לסימולציה ע"י מ"ט (דטרמיניסטית) בעלת זמן ריצה $2^{O(t(n))}$.

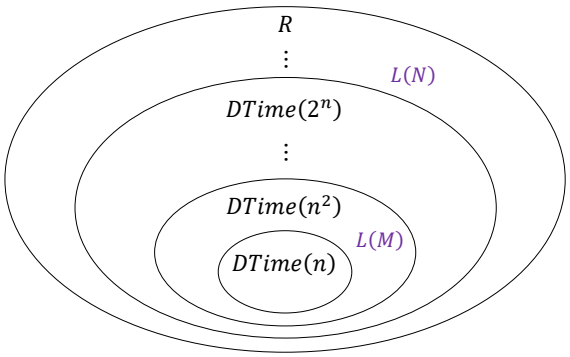
<u>הוכחה:</u> נובע ישירות מהסימולציה שראינו בהרצאה 5.

תזכורת: בהינתן מטל"ד N בנינו מ"ט M המסמלצת אותה ע"י כך שסרקנו את עץ החישוב "DFS-style". אם העומק של העץ הוא $t(n)$ אז הגודל שלו הוא $2^{t(n)}$.

אז מודלים סבירים דטר' שקולים עד כדי תקורה פולינומית, בעוד שבין מודלים דטר' ולא-דטר' יש פער אקספוננציאלי. משערים שזה אינהרנטי. זה הופך את המודל הלא-דטר ללא פיזיבילי. בקורס הזה ובתאוריה באופן כללי: "פיזבילי" = פולינומי.

שאלה: תהי M מ"ט דטר' רב-סרטית שרצה בזמן $O(n)$ ותהי N מטל"ד שרצה בזמן $O(N)$. מהן המחלקות הקטנות ביותר בהן נוכל למקם את השפות המתאימות $L(M), L(N)$ בציור מעמוד 3?

תשובה:



המחלקות *P* ו- *NP*

באופן פרקטי יש הבדל גדול בין אלג' שרץ בזמן $O(n)$ לבין $O(n^2)$ או $O(n^3)$. אך מעתה בקורס שלנו נעדיף להתייחס לכל האלגוריתמים בזמן פולינומי (כלומר $O(n^c)$ עבור קבוע c כלשהו) כמחלקה אחת.
הייתרונות בכך הם:

1. אין חשיבות למודל החישוב (כל עוד הוא דטרמיניסטי)
2. ניתוח זמן הריצה פשוט יותר (למשל אם האלג' מבצע מס' פולינומי של צעדים וכל צעד דורש זמן פולינומי, אז האלג' כולו פולינומי)
3. פחות רגיש לקידוד הקלט (למשל ניתן לקודד גרף ע"י מטריצה או ע"י רשימת שכנויות)
4. זמן פולינומי נחשב כפיזבילי. בפועל, לאלג' פולינומיים לבעיות מעניינות יש קבוע c קטן יחסית

<u>הגדרה:</u>

$$P \coloneqq \bigcup_{c\in\mathbb{N}} DTime(n^c)$$

כלומר אוסף כל הבעיות שניתנות להכרעה בזמן פולינומי ע"י מ"ט דטרמיניסטית.

נראה דוגמאות לשפות ב- P

<u>הגדרה:</u>

$$PATH = \{\langle G,s,t\rangle : t \text{ ל } s \text{ מ מסלול עם } G \text{ גרף מכוון}\}$$

<u>טענה:</u> $PATH \in P$

אלג' נאיבי: נעבור על כל המסלולים. זמן ריצה $n! \approx n^n$ כאשר n הוא מספר הצמתים ב G.

אלג' משופר: BFS, DFS. זמן ריצה פולינומי.

<u>הערה לגבי קידוד הקלט:</u> לגרף עם n קוד' כל קידוד סביר של הקלט יהיה פולינומי ב n ונוכל לעבור ממנו לקידודים סטנדרטיים בזמן פולינומי ב n. לכן אין צורך לקבוע קידוד מסויים.

כשדיברנו על מ"ט *M* שהקלט שלה הוא מ"ט אחרת *M'*, רצינו להדגיש את ההבדל בין המ"ט שאנחנו מריצים *M* לבין המ"ט שניתנת לנו בצורה מקודדת כחלק מהקלט *M'* ולכן הקפדנו לרשום *<M'>*.

כשאנחנו מדברים על בעייה שהקלט שלה הוא גרף או מספר אז זה הרבה פחות מבלבל – ברור שהגרף מקודד איכשהו (רשימת או מטריצת שכנויות) ולכן בהרצאות נרשה לעצמנו לכתוב G במקום $\langle G\rangle$. באופן פורמלי, בעיית הכרעה היא תמיד שפה מעל Σ^* כלשהו, כלומר הקלטים שלנו הם תמיד מחרוזות מעל א"ב כלשהו, ולכן בפועל כמעט תמיד יש איזשהו קידוד ברקע.

<u>הגדרה:</u>

$$PRIME = \{p \in \mathbb{N} : p \text{ is prime}\}$$

קיים אלג' פולינומי ולכן $PRIME \in P$
האלג' הפולינומי מסובכך ולא נראה אותו.

<u>עוד הערה לגבי קידוד הקלט:</u> המחלקה P אמנם מאפשרת לנו להיות פחות רגישים לקידוד, אבל חשוב לציין שיש מקרים בהם אופן קידוד הקלט משפיע <u>מאוד</u> על סיבוכיות הזמן של הבעיה ולעיתים אפילו משפיע על היות הבעיה ב P או לא.

דוגמה: אלגוריתם נאיבי ל PRIME. בהינתן מספר p: עבור $i \in \{2, \dots, p\}$ וחפש מחלק.

זמן ריצה $\widetilde{\Theta}(p)$. האם האלג' פולינומי באורך הקלט? תלוי בקידוד: אם p מקודד באונרית אז כן. אם p מקודד בבינארית אז האלג' אקספוננציאלי.

כברירת מחדל, כאשר נדבר על בעיות עם מבנה (מספרים, גרפים וכו') אז נניח כי הקלט נתון בקידוד קצר. למשל מספרים בקידוד בינארי/דצימלי, גרפים מקודדים כמטריצה או רשימת שכנויות. כל עוד אפשר לעבור בין קידודים בזמן פולינומי אז לא יהיה אכפת לנו באיזה קידוד משתמשים (בניגוד נניח לקורס "אלגוריתמים").

<u>שאלה:</u> האם יש שפה שהיא מחוץ ל P? כן, למשל HALT היא אפילו מחוץ ל R....

האם יש שפות ב- $R \setminus P$?
כן, ממשפט היררכיית הזמן נובע שקיימת בעייה ב $DTime\left(n^{\log n}\right)$ שהיא לא ב P.

למה? כי לכל קבוע c מתקיים ש $n^{\log n}$ הוא הרבה יותר גדול אסימפטוטית מ n^c והפער הזה מספיק גדול כדי "לאכול" גם את הפקטור של הלוג הדרוש למשפט ההיררכייה.

<u>הגדרה:</u>

$$NP \coloneqq \bigcup_{c \in \mathbb{N}} NTime(n^c)$$

כלומר אוסף כל הבעיות שניתנות לפתרון בזמן פולינומי ע"י <u>מטל"ד</u>.

<u>אבחנה:</u> $P \subseteq NP$. אז למשל $PATH \in NP$.

מסתבר שחלק גדול מאוד מהבעיות האלגוריתמיות שמעניינות אותנו נמצאות ב- NP. בהמשך נראה שאלו בדיוק הבעיות שאולי לא ניתן לפתור באופן יעיל אבל ניתן לזהות פתרון באופן יעיל.