חישוביות וסיבוכיות

מהי תורת החישוביות?

- תורת החישוביות היא תורה מתמטית המסווגת את הבעיות שניתן לפתור אותן בצורה חישובית.
- לשם כך, אנו נעזרים במודלים חישוביים, שהם תיאורים מתמטיים שמהווים הפשטה של מערכות חישוב
- בקורס זה נתמקד במודל חישובי בשם מכונת טיורינג.

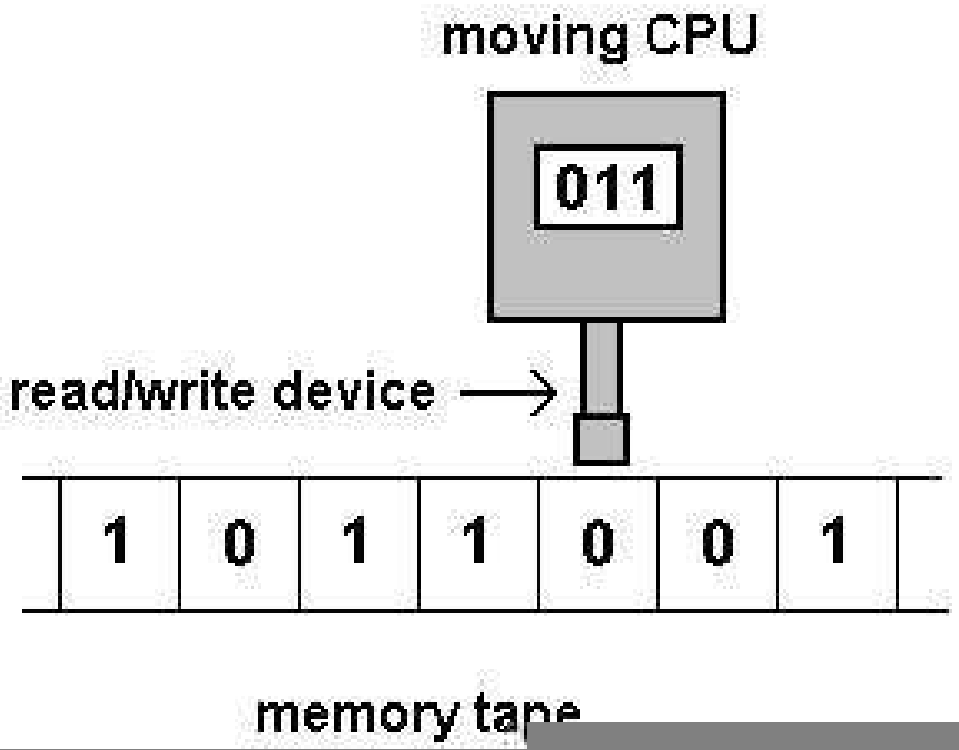
מושגי יסוד

- אלף-בית הוא קבוצה סופית לא ריקה של אותיות (למשל $\Sigma = \{0,1\}$, $\Gamma = \{a,b,c,d\}$).
- מחרוזת מעל א"ב Σ היא רצף סופי של אפס או יותר אותיות מהא"ב.
  המחרוזת הריקה מסומנת ε.
  קבוצת כל המחרוזות מעל Σ מסומנת Σ^*.
- שפה מעל א"ב Σ היא תת-קבוצה כלשהי של Σ^*.
- כל שפה מתארת בעיה חישובית – הבחנה בין מחרזות בשפה לאלו שאינן בשפה.

מכונת טיורינג

- מכונת טיורינג, באופן אבסטרטי מוכבת משלושה רכיבים:
  - סרט אינסופי שניתן לכתוב עליו אותיות מתוך א"ב סופי כלשהו.
  - ראש קורא/כותב שנע על גבי הסרט.
  - פונקציית מעברים (סופית) הקובעת לכל מצב של המכונה ולכל אות המופיעה על הסרט את המצב הבא אליו המכונה תעבור, את האות אותה תכתוב על הסרט ואת הכיוון אליו תזוז המכונה על הסרט.
- במצב ההתחלתי של המכונה הקלט כתוב על הסרט, כאשר שאר הסרט הוא ריק, כלומר מכיל סימני $\flat$ בלבד וראש המכונה נמצא בתחילת הקלט.
- המכונה מקבלת מחרוזת σ אם היא מגיעה למצב מקבל כאשר הסרט מכיל את המחרוזת σ בתחילת הריצה.
- המכונה דוחה מחרוזת σ אם היא מגיעה למצב דוחה או לא עוצרת כאשר הסרט מכיל את המחרוזת σ בתחילת הריצה.

מכונת טיורינג -- איור



מכונת טיורינג -- הגדרה פורמאלית

- מכונת טיורינג היא שביעיה $M = (Q, q_0, q_A, q_R, \Gamma, \flat, \delta)$, כאשר:
  - Q היא קבוצת מצבים סופית.
  - $q_0 \in Q$ מצב התחלתי.
  - $q_A \in Q$ מצב מקבל.
  - $q_R \in Q$ מצב דוחה.
  - Γ - א"ב עבודה של המכונה.
  - $\flat \in \Gamma$ - הסימן הריק.
  - $\Sigma = \Gamma \setminus \{\flat\}$ - א"ב הקלט של המכונה.
  - $\delta : ((Q \setminus \{q_A, q_R\}) \times \Gamma) \mapsto (Q \times \Gamma \times \{L, R\})$ - פוקציית המעברים.

קונפיגורציות של מכונת טיורינג

- קונפיגורציה של מכונת טיורינג M היא שלישיה $c = (\alpha, q, i)$ כאשר:
  - $\alpha \in \Gamma^*$ היא תוכן הסרט מהסימן הראשון שאינו $\flat$ ועד הסימן האחרון שאינו $\flat$.
  - $q \in Q$ הוא המצב הנוכחי; ו-
  - $i \in \mathbb{I}$ הוא מיקום הראש מתחילת α.
- **קונפיגורציה** (α, q, i) **תקרא** קונפיגורציה מקבלת **אם** $q = q_A$, וקונפיגורציה דוחה **אם** $q = q_R$. **קונפיגורציה תקרא** קונפיגורציה סופית **אם היא מקבלת או דוחה.**
- קונפיגורצה התחלתית **של** מכונת טיורינג בהנתן **קלט** σ **היא** $c_0 = (\sigma, q_0, 0)$.
- צעד חישוב **של** מכונת טיורניג הוא מעבר מקונפיגורציה אחת לאחרת לפי פונקציית המעברים δ של המכונה.

חישוב במכונת טיורינג

- מסלול חישוב של מכונת טיורינג M עבור קלט σ הוא סדרה של קונפיגורציות המופרדות בצעד חישוב אחד מהקונפיגורציה התחלתית ועד לקונפיגורציה סופית (מסלול סופי) או ללא סיום (מסלול חישוב אינסופי).
- מכונת טיורינג M מקבלת קלט $\sigma \in \Sigma^*$ מסלול החישוב של M על σ מסתיים בקונפיגורציה מקבלת.
- השפה שלמכונת טיורינג M היא קבצות כל המחרוזות ב- Σ^* ש- M מקבלת.
- מכונת טיורינג M עוצרת על קלט σ אם מסלול החישוב של M על σ הוא סופי.
- אם מכונת טיורניג M מקבלת את השפה $L \subseteq \Sigma^*$ ועוצרת על כל קלט $\sigma \in \Sigma^*$, נאמר ש- M מכרינה את השפה L.

מכונת טיורינג - דוגמא

- קבוצת מצבים $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_A, q_R\}$.
- א"ב עבודה $\{0, 1, \flat\}$

מצב/אות קלט	0	1	$\flat$
q_0	$(q_1, \flat, R)$	$(q_2, \flat, R)$	$(q_A, \flat, R)$
q_1	$(q_1, 0, R)$	$(q_3, 1, R)$	$(q_5, \flat, L)$
q_2	$(q_2, 0, R)$	$(q_4, 1, R)$	$(q_R, \flat, R)$
q_3	$(q_1, 0, R)$	$(q_3, 1, R)$	$(q_R, \flat, R)$
q_4	$(q_2, 0, R)$	$(q_4, 1, R)$	$(q_5, \flat, L)$
q_5	$(q_6, \flat, L)$	$(q_6, \flat, L)$	$(q_A, \flat, R)$
q_6	$(q_6, 0, L)$	$(q_6, 1, L)$	$(q_0, \flat, R)$

קידוד של מכונת טיורינג

- הגדרה של מכונת טיורינג היא בעלת אורך סופי, לכן ניתן לתאר כל מכונת טיורינג על ידי מחרוזת ביטים בעלת אורך סופי.
- למשל, נקודד בצורה אונארית את מספר המצבים במכונה, ולאחר מכן נקודד את פונקציית המעברים לפי הסדר כאשר q_0 יהיה המצב הראשון ו- q_A ו- q_R יהיו המצבים האחרונים בקידוד. לכל מצב ואות קלט נרשום את מספר המצב שיש לעבור (מספר ביטים כ- $\log$ מספר המצבים), את אות הפלט שיש לכתוב $(0, 1, \flat)$ (שני ביטים) ואת הכיוון בו צריך לנוע (ביט אחד).
- נסמן ב- $\langle M \rangle$ את הקידוד של המכונה M.

כריעות

- שפה L נקראת כריעה אם קיימת מכונת טיורינג שמכריעה אותה.
  כלומר, אם קיימת מכונת טיורינג שתמיד עוצרת, ומסיימת
  בקונפיגורציה מקבלת עבור קלטים ב- L ובקונפיגורציה דוחה עבור
  קלטים שאינם ב- L.
- נסמן ב- R את קבוצת השפות הכריעות.

מודל ה- RAM

- מכונת טיורינג שקולה למודל של מכונה בעלת גישה אקראית לזיכרון בעל גודל אינסופי עם תוכנה בעלת אורך סופי.
- מודל זה הוא אבסטרקציה טובה למערכות מחשב שאנו מכירים.
- שקילות המודלים היא במובן שקבוצת השפות הכריעות על ידי מכונת טיורינג שווה לקבוצת השפות הכריעות על ידי RAM.

בעיית העצירה

- בעיית העצירה הנה בעיה בסיסית באימות של תכונה.
- הבעיה היא: בהינתן קידוד של מכונת טיורינג M וקלט σ להכריע האם M עוצרת בהנתן הקלט σ.
- כלומר, השפה HP היא שפת כל הקידודים של מכונות וקלט $\langle M, \sigma \rangle$ כך ש- M עוצרת על σ.
- פתרון (שגוי): נדמה את הריצה של M על σ. הבעיה היא שאם M אינה עוצרת על σ גם המכונה המדמה שלנו לא תעצור, ואז היא לא תכריע את השפה המבוקשת.
- בעיית העצירה הנה בעיה שאינה ניתנת להכרעה.
- הוכחה: נניח בשלילה שקיימת מכונת טיורינג M_{HP} המכריעה את HP. נבנה מכונת טיורינג M' באופן הבא:

– M' תקבל כקלט קידוד של מכונת טיורינג ותעתיק אותו פעמיים, ואז תנוע לתחילת הקידוד.

- לאחר מכן, M' תפעל בדיוק כמו M_{HP} (נעתיק את כל מצבי M_{HP}) עד שזו תגיע לאחד מהמצבים q_A או q_R .
  * אם M_{HP} הגיעה ל- q_A, המכונה M' תיכנס ללולאה אינסופית.
  * אם M_{HP} הגיעה ל- q_R, המכונה M' תעבור למצב המקבל q'_A

כעת, נבחן את ריצת M' על הקלט $\langle M' \rangle$, כלומר על הקידוד של עצמה.

- אם M' עוצרת על הקלט $\langle M' \rangle$ אזי M_{HP} תגיע למצב q_A על $\langle M', M' \rangle$ ולכן M' תיכנס ללולאה אינסופית על הקלט $\langle M' \rangle$, בסתירה להנחה.
- אם M' לא עוצרת על הקלט $\langle M' \rangle$ אזי M_{HP} תגיע למצב q_R על $\langle M', M' \rangle$ ולכן M' תעצור ותקבל את הקלט $\langle M' \rangle$, בסתירה להנחה.

הגענו לסתירה בכל המקרים ולכן מתקבל ש- M_{HP} לא יכולה להתקיים ולכן $HP \notin R$.

רדוקציה

- רדוקציה היא דרך להמיר בעיה אחת לבעיה אחרת. הרעיון הוא שאם קיימת רדוקציה מבעיה א' לבעיה ב', הרי שאם נוכל לפתור את בעיה ב', נוכל לפתור גם את בעיה א'. או בשלילה, אם בעיה א' הינה בלתי כריעה, גם בעיה ב' תהיה כזו.
- מכונת טיורינג M מחשבת פונקציה $f : \Sigma^* \to \Sigma^*$ אם היא עוצרת על כל קלט σ ובקונפיגורציה הסופית שלה עבור קלט σ, תוכן הסרט שאינו $\flat$ הוא $f(\sigma)$.
- מכונת טיורינג M המחשבת את הפונקציה f_M מהווה רדוקציה משפה $L_1 \subseteq \Sigma^*$ לשפה $L_2 \subseteq \Sigma^*$ אם מתקיים: $L_1 = \{\sigma | f_M(\sigma) \in L_2\}$.
- אם קיימת רדוקציה משפה L_1 לשפה L_2 (סימון $L_1 \leq L_2$):
  - אם שפה L_2 הינה כריעה, הרי שגם לפה L_1 הינה כריעה.
  - אם שפה L_1 הינה בלתי-כריעה, הרי שגם שפה L_2 הינה בלתי-כריעה.

הוכחת אי-כריעות בעזרת רדוקציה

- נתונה השפה הבאה: $U = \{\langle M, \sigma\rangle | \sigma$ מקבלת את הקלט $M\}$.
- נוכיח ש- U בלתי-כריעה בעזרת רדוקציה מ- HP ל- U.
- נבנה מכונת טיורינג M_T המקבלת קידוד $\langle M, \sigma\rangle$ ומבצעת את העיבוד הבא על מכונה M לקבלת מכונה M':
  - מוסיפה ל- M עוד מצב q_x.
  - משנה את כל המעברים ל- q_R לעבור ל- q_x.
  - מוסיפה לפונקציית המעברים עבור q_x מעבר חזרה ל- q_x והזזת הסרט ימינה עבור כל אות קלט.
  - M_T מסיימת את החישוב בכתיבת $\langle M', \sigma\rangle$ על הסרט.
- פעולה זו הינה פשוטה לחישוב וניתנת לביצוע על ידי מכונת טיורינג.
- M מהווה רדוקציה על מ- HP ל- U משום שעבור קלט $\langle M, \sigma\rangle$, M מקבלת את σ אמ"ם M' עוצרת על σ.

סיבוכיות

מהי תורת הסיבוכיות?

- כמו שתורת החישוביות עוסקת בשאלה מה ניתן לחשב, עוסקת תורת הסיבוכיות בשאלה מה ניתן לחשב בזמן סביר.
- בפרט באמצעות תורת הסיבוכיות אנו עונים לשאלה כמה קשה לחשב בעיה מסויימת, ולאו דוקא לשאלה האם הבעיה קשה לחישוב.
- זמן חישוב של מכונת טיורינג M על קלט σ (סימון $t_M(\sigma)$) הוא מספר צעדי החישוב שעושה מכונה M על קלט σ עד להגעה לקונפיגורציה סופית, או ∞ אם M אינה עוצרת על σ.
- תהי $f : \mathbb{N} \to \mathbb{N}$ פונקציה. נאמר שמכונת טיורינג M עובדת בזמן $O(f(x))$ אם קיימים קבועים $a, b \in \mathbb{R}$ כך שלכל $\sigma \in \Sigma^*$, $t_M(\sigma) \leq a \cdot f(|\sigma|) + b$.

זמן פולינומי

- נאמר שמכונת טיורינג M עובדת בזמן פולינומי אם קיים $n \in \mathbb{N}$ כך ש- M עובדת בזמן $O(x^n)$.
- נסמן ב- P את קבוצת כל השפות אשר קיימת מכונת טיורינג שעובדת בזמן פולינומי ומכריעה אותן.
- נשים לב ש- $P \subseteq R$, משום שאנו דורשים מהשפות ב- P שיהיו כריעות.
- כמו כן, ידוע כי $P \neq R$. כלומר, ישנן בעיות אשר ניתנות לפיתרון, אך לא ניתן לפתור אותן בזמן פולינומי.

רדוקציה פולינומית

- מכונת טיורנג M מהווה רדוקציה פולינומית משפה $L_1 \subseteq \Sigma^*$ לשפה $L_2 \subseteq \Sigma^*$ אם היא רדוקציה מ- L_1 ל- L_2 והיא עובדת בזמן פלינומי.
- אם קיימת רדוקציה פלינומית משפה L_1 לשפה L_2 (סימון $L_1 \leq_P L_2$):
  - אם $L_2 \in P$, הרי שגם $L_1 \in P$.
  - אם $L_1 \notin P$, הרי שגם $L_2 \notin P$.

מכונת טיורינג לא-דטרמיניסטית

- נתאר לעצמנו מודל חישובי פלאי, אשר יכול לבצע מספר רב של חישובים במקביל.
- מכונת טיורינג לא-דטרמיניסטית, דומה למכונת טיורינג רגילה, אלא שפונקציית המעברים יכולה לעבור לקבוצה של פעולות (מצב,פלט,תנועת הסרט).
- המכונה הלא-דטרמיניסטית יכולה לבצע מסלולי חישוב רבים ושונים עבור כל קלט.
- מכונת טיורינג לא-דטרמיניסטית M מקבלת קלט σ אם קיים מסלול חישוב של M על σ המסתיים בקונפיגורציה מקבלת.
- מכונת טיורינג לא-דטרמיניסטית M דוחה קלט σ אם כל מסלול חישוב של M על σ מסתיים בקונפיגורציה דוחה או לא עוצר.

מכונת טיורינג לא-דטרמיניסטית -- דוגמא

- נתאר מכונת טיורינג לא-דטרמיניסטית לבחינה האם מספר הוא פריק (לא ראשוני):
  - המכונה מתחילה בלכתוב על הסרט, לאחר הקלט, מספר ספרות הזהה לאורך הקלט בצורה לא דטרמיניסטית.
  - המכונה מחלקת את המספר הכתוב בקלט באמצעות חילוק ארוך במספר אשר נכתב בצורה לא דטרמיניסטית.
  - אם המספר מתחלק ללא שארית, המכונה עוברת למצב מקבל. אחרת, המכונה עוברת למצב דוחה.
- המכונה הזו עובדת, משום שלמספר פריק תמיד ימצא מספר בו הוא מתחלק, ולכן יהיה לפחות מסלול חישוב אחד בו המכונה תקבל. עבור קלט ראשוני, המכונה תדחה. בכל מסלול חישוב, כנדרש.

שקילות למכונה דטרמיניסטית

- **משפט:** לכל מ"ט אי-דטרמיניסית M קיימת מ"ט דטרמיניסטית M' שמקבלת אותה שפה. בנוסף, אם כל מסלולי החישוב של M סופיים, אזי M' תמיד עוצרת.
- **הוכחה:** תהי M מ"ט אי-דטרמיניסטית. נבנה מ"ט דטרמיניסטית שקולה M'.
- נשים לב שניתן לתאר את כל מסלולי החישוב של M כעץ של קונפיגורציות.
- המכונה M' תבצע סריקת BFS של עץ הקונפיגורציות, ותעצור ותקבל אם היא מוצאת קונפיגורציה מקבלת, ותעצור ותדחה אם היא סיימה לסרוק את העץ.
- המכונה הנ"ל שקולה ל- M משום שהיא מקבלת אמ"ם קיים ל- M מסלול מקבל. כמו כן, אם כל מסלולי החישוב של M סופיים, אזי גם עץ הקונפיגורציות שלה סופי ולכן M' תעצור תמיד.

זמן פולינומי לא-דטרמיניסטי

- זמן חישוב של מכונת טיורינג לא-דטרמיניסטת M על קלט σ (סימון $t_M(\sigma)$) הוא מספר צעדי החישוב המקסימלי שעושה מכונה M על קלט σ עד להגעה לקונפיגורציה סופית בין כל מסלולי החישוב שלה, או ∞ אם יש ל-M מסלול חישוב בו היא אינה עוצרת על σ.
- המושג של עבודה בזמן פולינומי מוגדר באופן זהה גם עבור מכונת טיורינג לא-דטרמיניסטית.
- נסמן ב- NP את קבוצת כל השפות אשר קיימת מכונת טיורינג לא-דטרמיניסטי שעובדת בזמן פולינומי ומכריעה אותן.
- נשים לב ש- $NP \subseteq R$, משום שאנו דורשים מהשפות ב- NP שיהיו כריעות.
- כמו כן, ידוע כי $NP \neq R$. כלומר, יש שפות שהן כריעות, אך אין מ"ט לא-דטרמיניסטית שעובדת בזמן פולינומי ומכריעה אותן.

הקשר בין P ל- NP

- נשים לב כי $P \subseteq NP$, משום שכל מ"ט דטרמיניסטית היא גם מ"ט לא-דטרמיניסטית המקבלת אותה שפה באותו מספר צעדי חישוב.
- השאלה האם $P = NP$ או $P \subsetneq NP$ היא שאלה פתוחה במדעי המחשב.
- כלומר, לא ידוע האם יש תועלת בחישוב לא-דטרמיניסטי, מבחינת סוג הבעיות שניתן לפתור בזמן פולינומי.
- שפה L תיקרא NP-קשה אם לכל שפה $L' \in NP$ מתקיים $L' \leq_P L$. כלומר, שפה L קשה לפחות כמו כל שפה ב- NP.
- שפה L תיקרא NP-שלמה אם $L \in NP$ והיא NP-קשה.

משפט קוק

- נשאלת השאלה האם קיימת שפה שהיא NP-שלמה, כלמר שניתן לפתור אותה באמצעות מ"ט לא-דטרמיניסטית, אך היא קשה יותר מכל בעיה אחרת ב- NP.
- כדי להראות שבעיה NP-שלמה נצטרך להראות מ"ט לא-דטרמיניסטית שמכריעה אותה בזמן פולינומי, ולהראות רדוקציה פולינומית מכל בעיה ב- NP לבעיה זו.

- בעיית ספיקות הפסוקים (SAT) היא הבעיה הבאה: בהיתנן פסוק בתחשיב הפסוקים, האם קיימת השמה למשתני הפסוק אשר תגרום לפסוק לקבל ערך אמת **TRUE**?
- מ"ט לא-דטרמיניסטית לפתרון בעיה זו פשוט תנחש השמה ותנסה להציב אותה בפסוק. השמה מספקת אחת מספיקה כדי להבטיח מסלול חישוב בו המכונה תקבל.
- קיום רדוקציה מכל בעיה ב- NP נובעת מהאפשרות לתרגם את המכונה הלא-דטרמיניסטית הפותרת את הבעיה והקלט שלה לפסוק בלוגיקה שהוא ספיק אמ"ם המכונה מקבלת את הקלט בזמן פולינומי.
- מכאן נובע ש- SAT היא NP-שלמה.

עוד בעיות NP-שלמות

- בעיית הסוכן הנוסע -- נתון גרף של n ערים ועלות הנסיעה בין כל 2 ערים. יש להכריע האם קיים מסלול מעגלי העובר בין כל הערים בעלות של לא יותר מ- k.
- בעיית המעגל ההמילטוני -- מקרה פרטי של בעית הסוכן הנוסע כאשר המרחקים בין כל זוג ערים הם 1 או אינסוף ו- k הוא מספר הערים.
- בעיית התכנות בשלמים 0/1 -- הכרעה האם קיימים ערכי $x_1, x_2, \ldots, x_n \in \{0,1\}$ המקיימים את האי-שוויונים הבאים:

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & \leq & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & \leq & b_2 \\ \vdots & \vdots & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & \leq & b_m \end{array}$$

עבור ערכי a_{ij} נתונים.

סיבוכיות זיכרון

- לפעמים מעניינת אותנו, בנוסף לזמן החישוב, כמות הזיכרון הנדרשת לחישוב.
- מכונת טיורינג M עובדת בזיכרון $O(m)$ אם קיימים קבועים $a, b \in \mathbb{R}$ כך שלכל $\sigma \in \Sigma^*$, $m_M(\sigma) \leq a \cdot f(|\sigma|) + b$, כאשר $m_M(\sigma)$ הוא מספר המיקומים שונים שהראש של המכונה במהלך ריצת M על σ.
- נאמר שמכונת טיורינג M עובדת בזיכרון פולינומי אם קיים $n \in \mathbb{N}$ כך ש- M עובדת בזיכרון $O(x^n)$.
- ההגדרות הנ"ל נכונות עבור מכונות טיורינג דטרמיניסטיות ואי-דטרמיניסטיות כאחד.

מחלקות סיבוכיות זיכרון ומשפט Savitch

- מחלקת השפות המתקבלות על ידי מכונות טיורניג דטרימינסטיות שעבודות בזיכרון פולינומי נקראת $PSPACE$.
- מחלקת השפות המתקבלות על ידי מכונות טיורניג לא דטרימינסטיות שעבודות בזיכרון פולינומי נקראת $NPSPACE$.
- נשים לב להכלות הבאות:

$$P \subseteq NP \subseteq PSPACE \subseteq NPSPACE$$

משום שמכונה דטרמיניסטית היא מקרה פרטי של מכונה לא דטרמיניסטית, וכי הסימולציה של מכונה לא-דטרמיניסטית ע"י מכונה דטרמיניסטית דורשת זיכרון פולינומי בלבד.

- באופן מפתיע, Savitch (1970) הוכיח ש-
$PSPACE = NPSPACE$, כלומר כל בעיה בניתן לפתור בזיכרון פולינומי על ידי מכונה לא דטרמיניסטית, ניתן לפתור בזיכרון פולינומי גם על ידי מכונה דטרמיניסטית