# סיכום מקיף: אוטומטים ושפות פורמליות

גרסה פרקטית למבחן | מחולק לפי צבעים ונושאים

# 1. מבוא והגדרות בסיס

BASICS

## אינטואיציה מרכזית

**הכל עניין של זיכרון.**

*אנחנו מודדים כמה "זיכרון" המחשב צריך כדי לדעת אם המילה חוקית.*

*זיכרון הכי חלש =*

**רגולרית**

*(אס"ד).*

*זיכרון מחסנית =*

**ח"ה**

*(PDA).*

## הגדרות ודגשים טכניים

- **א"ב ( $\Sigma$ ):** קבוצה סופית של תווים. למשל $\{0,1\}$ .
- **מילה ריקה ( $\varepsilon$ ):** מילה באורך 0. שים לב: היא שונה מ- $\emptyset$ (הקבוצה הריקה).
- **השפה האוניברסלית ( $\Sigma^*$ ):** אוסף כל המילים האפשריות (קבוצה אינסופית).
- **שפה ( $L$ ):** תת-קבוצה כלשהי של $\Sigma^*$ .

## גישה לפתרון תרגילים (פעולות)

**<u>ניתוח פעולות</u>**

- **שרשור ( $L_1L_2$ ):** לוקחים מילה מ- $L_1$ ומדביקים לה מילה מ- $L_2$ .
  דוגמה: אם $L_1=\{a\}$ ו- $L_2=\{b, bb\}$ , השרשור הוא $\{ab, abb\}$ .
- **כוכבית ( $L^*$ ):** בחירה ושרשור של מילים מהשפה, 0 פעמים או יותר. תמיד מכיל $\varepsilon$ .
- **הפרש שפות ( $A \setminus B$ ):** כל המילים שנמצאות ב- $A$ , פחות אלו שנמצאות גם ב- $B$ .

⚠️ **מלכודות במבחן:**

1. $\emptyset^* = \{\varepsilon\}$ (ולא קבוצה ריקה!).
2. השפה $\Sigma^*$ היא אינסופית, אבל כל מילה בתוכה היא סופית.

# 2. סיווג שפות (ההיררכיה)

## צ'ק ליסט לזיהוי מהיר

### 1. רגולרית (זיכרון סופי)

חפש:

- מגבלה סופית ("מכיל 01", "עד 1000 תווים").
- מודולו (זוגי/אי-זוגי, מתחלק ב-3).
- בלוקים ללא קשר כמותי ( $a^* b^*$ ).

### 2. חסרת הקשר - ח"ה

חפש:

- תלות אחת של כמויות ( $a^n b^n$ ).
- סימטריה / פלינדרום ( $ww^R$ ).
- השוואה כמותית ( $\#a = \#b$ ).

### 3. לא ח"ה (קשה מדי)

חפש:

- שלוש תלויות ( $a^n b^n c^n$ ).
- תלות צולבת ( $a^n b^m c^n d^m$ ).
- העתקה ( $ww$ ) - דורש לזכור הכל.

[הדבק כאן: תמונה של דיאגרמת ההיררכיה (חומסקי)]

⚠️ **דגש קריטי:** זה ששפה $L_1$ מוכלת ב- $L_2$ (שהיא רגולרית), לא אומר ש- $L_1$ רגולרית.

דוגמה: $\{a^n b^n\} \subseteq \Sigma^*$ . הראשונה ח"ה, השנייה רגולרית.

# 3. אוטומטים סופיים (אס"ד / אסל"ד)

MACHINES

## אינטואיציה

*טיול על גרף. כל אות לוקחת אותך למצב הבא. המכונה "זוכרת" רק באיזה עיגול היא נמצאת כרגע. אם המילה נגמרה במצב מקבל (עיגול כפול) - היא התקבלה.*

**[הדבק כאן: ציור כללי של אוטומט]**

סימונים: חץ התחלה, מצבים (עיגולים), מעברים (חצים), מצב מקבל (עיגול כפול)

## ההבדלים הטכניים

**אס"ד (DFA - דטרמיניסטי):**

- מכל מצב יוצא חץ אחד **בדיוק** לכל תו.
- חייב להיות שלם (אם נתקעים ← הולכים למצב "מלכודת" Trap).
- אין ניחושים.

---

**אסל"ד (NFA - לא דטרמיניסטי):**

- מותר כמה חצים לאותה אות (ניחוש).
- מותר מעברי `ε` (דילוג חינם).
- אם אין חץ מתאים, המכונה נתקעת (והמסלול מת).
- **קבלה:** אם קיים לפחות מסלול אחד שמצליח.

## גישה לפתרון תרגילים

**<u>סוג 1: בניית אוטומט</u>**

- **שאלות מודולו (מתחלק ב-3):** צייר מעגל של 3 מצבים ( $q_0, q_1, q_2$ ). כל אות מקדמת צעד.

- **"מכיל את רצף abc":** בנה שרשרת $q_0 \to q_1(a) \to q_2(b) \to q_3(c)$. במצב האחרון נשארים לנצח.
- **"האות השלישית מהסוף היא a":** השתמש באסל"ד! תמיד תהיה בהתחלה, וגם תנחש שהסוף הגיע (מעבר עם a לשלושת המצבים האחרונים).

**סוג 2: המרה מאסל"ד לאס"ד**

1. בנה טבלה: עמודות = תווים, שורות = קבוצות מצבים.
2. מצב ההתחלה החדש = $\varepsilon\text{-closure}(\{q_0\})$ (כל מה שאפשר להגיע אליו עם אפסילון).
3. מלא את הטבלה: לאן מגיעה כל קבוצה? (איחוד המצבים).

[הדבק כאן: דוגמה לתרגיל המרה מאסל"ד לאס"ד (טבלה + שרטוט)]

⚠️ **כלל ברזל למבחן:** רוצים לבנות **משלים** לשפה?
1. בונים אס"ד (חייב להיות דטרמיניסטי ומלא!).
2. הופכים: מקבל ↔ לא מקבל.

**אסור לעשות משלים על אסל"ד!**

# 4. ביטויים רגולריים (Regex)

REGEX

## אינטואיציה

כתיבת השפה כנוסחה. `*` = לולאה, `+` = בחירה (או), שרשור = רצף.

## תבניות זהב לפתרון

- **מתחיל ב-a:** $a\Sigma^*$
- **מסתיים ב-bb:** $\Sigma^* bb$
- **מכיל את aba:** $\Sigma^* aba \Sigma^*$
- **מספר זוגי של a:** $(b^* \ a \ b^* \ a \ b^*)^*$

  הסבר: ה-b חופשיים, ה-a חייבים לבוא בזוגות בתוך הבלוק.
- **לפחות a אחד:** $\Sigma^* a \Sigma^*$

## מעבר מאוטומט לביטוי

**שיטת הניפוי (State Elimination):**

1. מוסיפים מצב התחלה חדש ומצב סוף חדש.
2. בוחרים מצב למחיקה ( $q$ ).
3. יוצרים "גשר" (ביטוי רגולרי) לכל מסלול שעבר דרך $q$ .
4. חוזרים עד שנשאר רק המעבר מההתחלה לסוף.

[הדבק כאן: דוגמה פתורה של מעבר מאוטומט לביטוי רגולרי]

⚠️ **שים לב להבדל:**

$(ab)^*$ = {ε, ab, abab, ababab...}

$a^*b^*$ = {ε, a, b, aa, ab, bb, aaa...}

# 5. למת הניפוח (הוכחת אי-רגולריות)

PROOF

## אינטואיציה

*אם השפה רגולרית ← יש לה אוטומט סופי. אם ניקח מילה ענקית, חייבים לעבור באותו מצב פעמיים ← נוצר*

**מעגל**

*. הלמה אומרת: "אם יש מעגל, אפשר לרוץ עליו שוב ושוב (לנפח) ולהישאר בשפה". אם ניפחנו ויצאנו מהשפה ← סתירה ← לא רגולרית.*

**החלוקה** `z = uvw` **:**

- `u` : החלק לפני המעגל.
- `v` : המעגל עצמו (מה שמנפחים). חובה `|v| ≥ 1` .
- `w` : החלק אחרי המעגל.
- תנאי חשוב: `|uv| ≤ n` (המעגל קרוב להתחלה).

[הדבק כאן: ציור "הבלון" הממחיש מסלול עם מעגל v באמצע]

## האלגוריתם המלא להוכחה

1. **הנחה:** נניח בשלילה ש- `L` רגולרית. יהי `n` קבוע הניפוח.
2. **בחירה (קריטי):** נבחר מילה `z` שתלויה ב- `n` ושייכת לשפה.
   *טיפ:* לשפת $a^nb^n$ נבחר $z = a^nb^n$ .
3. **פירוק:** תהי חלוקה כללית `z = uvw` המקיימת:
   - `|uv| ≤ n`
   - `|v| ≥ 1`

4. **ניתוח:** לפי החלוקה, `v` חייב להיות מורכב רק מ- `a` (כי `uv` מוכל ב- `n` האותיות הראשונות).
5. **הניפוח (סתירה):** נבחר `i` (למשל `i=2`). המילה החדשה $z' = u\ v^2\ w$ מכילה יותר `a` מ- `b` ← לא בשפה ← סתירה!

## דוגמה מלאה לפתרון (להעתקה)

**הוכח ש- $L = \{a^n b^n\}$ לא רגולרית.**

1. נניח בשלילה שרגולרית, יהי `n` הקבוע.
2. נבחר $z = a^n b^n$ . (ברור ש- $z \in L$ ואורכה $2n \geq n$ ).
3. תהי חלוקה $z=uvw$ , המקיימת את תנאי הלמה.
4. בגלל ש- $|uv| \leq n$ , החלק `uv` נופל כולו בתוך רצף ה- `a` .
לכן `v` מכיל רק `a` (כלומר $v=a^k$ עבור $k \geq 1$ ).
5. נבחר `i=2` (ננפח). המילה שתתקבל: $a^{n+k} b^n$ .
קיבלנו מילה שבה מספר ה- `a` גדול ממספר ה- `b` . מילה זו אינה ב- `L` . סתירה להנחה.

⚠️ **מה אסור לעשות?**
אסור לבחור את `n` (הוא נתון).
אסור לבחור חלוקה ספציפית ל- `u,v,w` (חייבים להראות סתירה לכל חלוקה).

# 6. משפט נרוד ו-Rank

## יחס שקילות ו-Rank

**יחס שקילות ( $R_L$ ):** שתי מילים $x, y$ הן שקולות אם "יש להן אותו עתיד". כלומר, לכל סיומת $z$ , או ששתיהן מתקבלות או ששתיהן נדחות.

**Rank (דרגה):** מספר מחלקות השקילות של השפה.
זהו בדיוק **מספר המצבים ב-DFA המינימלי** של השפה.

## מילה מפרידה

> *כדי להוכיח ששתי מילים שונות ( $x, y$ ) חייבות להגיע למצבים שונים, צריך למצוא מילה $z$ שתפריד ביניהן (אחת תתקבל ואחת לא).*

## דוגמה וגישה לתרגיל

**שאלה: מה ה-Rank של שפת המילים באורך זוגי מעל $\{a\}$ ?**

האינטואיציה: האוטומט צריך לזכור רק שני דברים: "כרגע זוגי" או "כרגע אי-זוגי". לכן ה-Rank יהיה 2.

**הוכחה פורמלית:**

- ניקח מילה באורך זוגי: $x = aa$ .
- ניקח מילה באורך אי-זוגי: $y = a$ .
- נחפש מפריד $z$ : המילה הריקה $\varepsilon$ תעבוד.
  $x \cdot \varepsilon = aa$ (זוגי → בשפה).
  $y \cdot \varepsilon = a$ (אי-זוגי → לא בשפה).
- מסקנה: $x$ ו- $y$ לא שקולות. יש לפחות 2 מחלקות.
- כל מילה אחרת שקולה לאחת מאלה. לכן **Rank = 2**.

[הדבק כאן: דוגמה למציאת מפריד לשפה L (למשל שארית חלוקה ב-3)]

# 7. אוטומט מחסנית (PDA)

PDA

## אינטואיציה

*כמו אוטומט רגיל, אבל עם "מחסנית פנקייקים" בצד. אפשר לשים (PUSH) או לאכול מלמעלה (POP). זה נותן יכולת לספור ולבצע התאמות (LIFO).*

[הדבק כאן: שרטוט של מעבר PDA (קלט, שליפה, דחיפה)]

## איך בונים PDA? (הגישה לפתרון)

**דוגמה: השפה** $a^n b^n$

1. **מצב 1 (דחיפה):** כל עוד מגיע a, דחוף A למחסנית.
2. **מעבר:** כשמגיע b ראשון (או בניחוש באסל"ד), עבור למצב 2.
3. **מצב 2 (שליפה):** על כל b שמגיע, שלוף A מהמחסנית.
4. **קבלה:** אם הקלט נגמר והמחסנית ריקה (או הגענו לסימן תחתית) - קבל.

⚠️ **דגש חשוב:** בניגוד לאוטומטים סופיים: **PDA דטרמיניסטי חלש יותר מ-PDA לא דטרמיניסטי.**

למשל, השפה $ww^R$ (פלינדרום) דורשת ניחוש ולכן היא לא דטרמיניסטית.

# 8. דקדוקים (CFG)

CFG

## אינטואיציה ועצי גזירה

מערכת של חוקי גזירה (החלפה). מתחילים מ- `S` ומחליפים משתנים עד שנשארות רק אותיות.

- **עץ גזירה:** ציור שמראה איזה חוק הופעל בכל שלב.
- **אמביגואיות (דו-משמעות):** אם למילה אחת יש **שני עצי גזירה שונים** (באותו דקדוק).
- **דקדוק ליניארי:** בכל צד ימין יש לכל היותר משתנה אחד → שקול לשפה רגולרית!

[הדבק כאן: תמונה של עץ גזירה למילה aaabbb]

## טריקים לבניית דקדוק

- **רקורסיה באמצע ( $a^n b^n$ ):** $S \to aSb \mid \varepsilon$
- **איחוד ( $L_1 \cup L_2$ ):** $S \to S_1 \mid S_2$
- **שרשור ( $L_1 L_2$ ):** $S \to S_1 S_2$
- **פלינדרום:** $S \to aSa \mid bSb \mid \varepsilon$
- **סוגריים מאוזנים:** $S \to (S) \mid SS \mid \varepsilon$

# 9. העמקה: שפות חסרות הקשר (ח"ה)

DEEP DIVE

## מהי שפה ח"ה?

*שפה שניתן ליצור ע"י דקדוק חסר הקשר (CFG) או לקבל ע"י אוטומט מחסנית (PDA). היא מייצגת מבנים היררכיים (כמו קוד תכנות, ביטויים חשבוניים, סוגריים).*

## ההבדל בין דטרמיניסטי ללא-דטרמיניסטי

- **ח"ה דטרמיניסטית (DCFL):** שפות שאפשר לקבל ע"י DPDA (אוטומט מחסנית דטרמיניסטי).
  *דוגמה:* $a^n b^n$ (ברור מתי עוברים מ-a ל-b, אין ניחושים).
- **ח"ה לא דטרמיניסטית:** שפות שדורשות "ניחוש". DPDA לא מספיק להן.
  *דוגמה קלאסית:* פלינדרומים $ww^R$ (האוטומט לא יודע איפה האמצע בלי לנחש).

**היררכיה פנימית:** רגולריות ⊂ ח"ה דטרמיניסטיות ⊂ ח"ה.

## מהאינוח"ה?

- **שפות התלויות ב-3 גורמים:** $a^n b^n c^n$ (המחסנית יכולה להשוות a ל-b, אבל אז היא ריקה ולא ניתן להשוות ל-c).
- **שפות העתקה:** $ww$ (דורש זיכרון של רצף אקראי שלם ושליפתו באותו סדר, אבל מחסנית הופכת סדר).
- **תלות צולבת:** $a^n b^m c^n d^m$ .

# 10. תכונות סגירות

CLOSURE

## טבלת סגירות מסכמת

- **רגולריות:** סגורות להכל (איחוד, חיתוך, משלים, שרשור, כוכבית).
- **ח"ה (CFL):**
  ✓ סגורות ל: איחוד, שרשור, כוכבית, היפוך.
  ✗ **לא סגורות** ל: חיתוך, משלים.

**טריקים נפוצים למבחן**

- **חיתוך ח"ה עם רגולרית → נשאר ח"ה.**
  (שימושי להוכחה ששפה אינה ח"ה ע"י חיתוך וסתירה).
- **חיתוך שתי שפות ח"ה → לרוב לא ח"ה.**
  דוגמה: $L_1=\{a^n b^n c^*\}$ חיתוך עם $L_2=\{a^* b^n c^n\}$ נותן $a^n b^n c^n$.

[הדבק כאן: טבלת סגירות מלאה מהמחברת]