

מודלים חישוביים

דגשים, קווים מנחים, רעיונות מרכזיים

מיכל ארמוני

המחלקה להוראת המדעים

מכון ויצמן למדע

מודלים חישוביים

- שונה באופייה מיחידות אחרות
- עוסקת בבסיס ה**תיאורטי** של מדעי המחשב
- אופייה **מתמטי** ואינה מערבת תכנות

שאלת המוטיבציה

למה צריך את זה?!!
זה לא מדעי המחשב!!

- מה מקומו של הנושא בתוך מדעי המחשב?
- מה המוטיבציה לכלול את הנושא בתוכנית הלימודים בביה"ס התיכון?

שאלת המוטיבציה (2)

באתי ללמוד תכנות.

לא רוצה ללמוד מודלים חישוביים!

זה לא יתרום לי בעתיד.

זה לא יעשה אותי מתכנת טוב יותר!

גם זה מדעי המחשב!!

● התחום התיאורטי של מדעי המחשב

◆ תורת החישוביות: איזה בעיות ניתנות לפתרון?

◆ תורת הסיבוכיות: איזה בעיות ניתנות לפתרון באופן יעיל?

◆ קשרים בין בעיות (חישוביות, סיבוכיות)

תבניות חשיבה ייחודיות: הפשטה מתמטית,
הוכחות מתמטיות...

גם זה מדעי המחשב!!

● הזדמנות להתנסות בתבניות חשיבה מוכרות
בהקשר שונה

◆ הפשטה

◆ רדוקציה (קשרים בין בעיות)

התמודדות עם שאלת המוטיבציה (1) – בעזרת ההקדמה

למה מודל חישובי?

כדי לתת הגדרה מתמטית, פשוטה
ונקייה, המייצגת מכונות מסוג מסוים,
שבעזרתה אפשר לדון בשאלות העוסקות
במכונות אלו

התמודדות עם שאלת המוטיבציה (1) – בעזרת ההקדמה

קווים כלליים לדיון מקדים

- הצגת בעיית העצירה
- האם מחשב יכול לפתור את בעיית העצירה?
- מה משמעות השאלה?
- איך עונים על שאלה כגון זו?

שאלת המוטיבציה (2)

- מטרת לימוד מקצוע עיוני בתיכון, ברמה של 5 יח"ל: לערוך לתלמידים היכרות רחבה ומייצגת ככל שניתן עם התחום, על פניו השונים
- היכרות עם רובד תיאורטי ותבניות חשיבה פורמליות במדעי המחשב
- המטרה היא ללמד מדעי המחשב ולא תכנות
- לא הכשרה מקצועית
- חידוד החשיבה המתמטית, התיאורטית, הלוגית

ציר המודלים (הפשטה, הגדרות מתמטיות)

אוטומט סופי דטרמיניסטי



אוטומט סופי דטרמיניסטי לא מלא



אוטומט סופי לא דטרמיניסטי



אוטומט מחסנית



מכונת טיורינג

ציר השאלות התיאורטיות

- השוואה בין מודל חדש לקודם
- השוואה בין דטרמיניזם לאי-דטרמיניזם
- דיון במגבלותיו של מודל
- דיון בתכונות סגירות:
 - ◆ משלים
 - ◆ איחוד
 - ◆ חיתוך
 - ◆ שרשור
 - ◆ היפוך
 - ◆ חלקיות

טבלת מעברים (צורות ייצוג שונות)

● ניתן לתאר אוטומט על ידי

◆ תיאור גרפי

או

◆ טבלת מעברים + ציון מצב התחלתי + ציון מצבים מקבלים

הגדרת אוטומט סופי דטרמיניסטי

לאוטומט סופי דטרמיניסטי יש חמישה מרכיבים והם :

1. א"ב – כל אותיות הקלט האפשריות עבור האוטומט. מספר האותיות בא"ב זה חייב להיות סופי וגדול מ-0 (בל יטעה אתכם המושג "אותיות"; מדובר בסימנים כלשהם. ואולם, כפי שציינו בסוף הסעיף הקודם, מעתה נקרא לסימנים אלו בשם אותיות).
2. מצבים – כל המצבים שבהם יכול האוטומט להימצא. מספר המצבים חייב להיות סופי וגדול מ-0.
3. מצב התחלתי – המצב שממנו מתחיל האוטומט את מסלול החישוב על כל מילת קלט.
4. קבוצת מצבים מקבלים – קבוצה מתוך קבוצת המצבים, המכילה 0 מצבים או יותר.
5. פונקציית מעברים – לכל זוג של אות, פונקציה זו מתאימה מצב (אחד ויחיד) שאליו עובר האוטומט כאשר במצב זה נקראת אות זו.
כפי שכבר הזכרנו בסעיף הקודם, פונקציית המעברים היא דרך אחרת לתיאור טבלת המעברים

הגדרה פורמלית של אוטומט סופי דטרמיניסטי

אוטומט סופי דטרמיניסטי A נתון על-ידי

$$A = (\Sigma_A, Q_A, q_{0A}, F_A, \delta_A)$$

כאשר: $-\Sigma_A$ א"ב קלט

Q_A – קבוצה סופית לא ריקה של מצבים

$q_{0A} \in Q_A$ – מצב התחלתי,

$F_A \subseteq Q_A$ – קבוצת מצבים מקבלים,

$\delta_A: Q_A \times \Sigma_A \rightarrow Q_A$ – פונקציית מעברים,

הגדרת אוטומט סופי דטרמיניסטי

לאוטומט סופי דטרמיניסטי יש חמישה מרכיבים והם :

1. א"ב – כל אותיות הקלט האפשריות עבור האוטומט. מספר האותיות בא"ב זה חייב להיות סופי וגדול מ-0 (בל יטעה אתכם המושג "אותיות"); מדובר בסימנים כלשהם. ואולם, כפי שציינו בסוף הסעיף הקודם, מעתה נקרא לסימנים אלו בשם אותיות).
 2. מצבים – כל המצבים שבהם יכול האוטומט להימצא. מספר המצבים חייב להיות סופי וגדול מ-0.
 3. מצב התחלתי – המצב שממנו מתחיל האוטומט את מסלול החישוב על כל מילת קלט.
 4. קבוצת מצבים מקבלים – קבוצה מתוך קבוצת המצבים, המכילה 0 מצבים או יותר.
 5. פונקציית מעברים – לכל זוג של אות, פונקציה זו מתאימה מצב (אחד ויחיד) שאליו עובר האוטומט כאשר במצב זה נקראת אות זו.
- כפי שכבר הזכרנו בסעיף הקודם, פונקציית המעברים היא דרך אחרת לתיאור טבלת המעברים

פרק 3 – מילים ושפות פורמליות

- פרק תיאורטי
- המעבר קשה, המוטיבציה עלולה לרדת
- התמודדות אפשרית: ספירה גבוהה יותר
- ◆ מתכנת ← מפתח שפות תכנות וחוקר את תכונותיהן
- ◆ בניית אוטומטים ← חקר אוטומטים

סעיף 3.1 – רשימת הגדרות

תחום מתמטי חייב
להיות מוגדר היטב,
בצורה מדויקת.

אחרת ניתן למשל
להוכיח דברים שאינם
נכונים

- א"ב
- מילה
- אורך מילה
- מילה ריקה
- פעולות על מילים
- ◆ שרשור
- ◆ חזקה
- ◆ היפוך
- שפה פורמלית
- שפה ריקה
- פעולות על שפות
- ◆ שרשור
- ◆ חזקה
- ◆ היפוך

הוכחות אי-רגולריות

- הסעיף הקשה ביותר ביחידה כולה:

- ◆ הוכחה פורמלית

- ◆ בדרך השלילה

- ◆ שבתוכה חבויה שוב דרך השלילה...

- מוטיבציה:

- ◆ הגדרנו מודל חישובי – האם אפשר בעזרתו לעשות הכל? (ציר השאלות התיאורטיות)

מסגרת הוכחת אי-רגולריות בספר לתלמיד

נניח שהשפה L רגולרית.

לכן קיים אוטומט סופי A שמקבל את L .

נבחר קבוצת מילים אינסופית W ונראה:

על כל מילה בקבוצה W , האוטומט A מגיע למצב שונה.

\Leftarrow ל- A אינסוף מצבים, **בסתירה** להיותו אוטומט סופי.

מסגרת הוכחת אי-רגולריות בספר לתלמיד - המשך

הוכחת טענת העזר

על כל מילה בקבוצה W , האוטומט A מגיע למצב שונה.

נניח שיש בקבוצה W שתי מילים – w_1 ו- w_2 – שעליהן מגיע האוטומט A לאותו מצב q :

כעת נמצא מילה w המקיימת:

$$w_1 \cdot w \in L$$

$$w_2 \cdot w \notin L$$

מאחר ש- $w_1 \cdot w \in L$ אז A מקבל את $w_1 \cdot w$, כלומר, מהמצב q המילה w מובילה אותו למצב מקבל. אך אז בהכרח A מקבל גם את $w_2 \cdot w$.

סתירה להיותו אוטומט המקבל את L .

הוכחות אי-רגולריות - המשך

- לא עבור כל שפה לא רגולרית שיטת ההוכחה הזאת עובדת. \Leftarrow אם לא מצליחים להוכיח זה עוד לא אומר שהשפה לא רגולרית
- למרות שההוכחות דומות אחת לשנייה נדרשת מיומנות ביישום מסגרת ההוכחה \Leftarrow חשוב לתרגל
- רק הוכחה מלאה נחשבת הוכחה
- **זהירות בהיצמדות לתבנית !!!**

הוכחות אי-רגולריות - המשך

הלקח החשוב:

השפה מחייבת זיכרון אינסופי ולאוטומט זיכרון סופי



אבל גם למחשב יש זיכרון
סופי...

הפשטה

הוכחות אי-רגולריות – קשיים אופייניים

- הוכחות לא מלאות, המשתמשות בנימוקים אינטואיטיביים ("השפה מחייבת זיכרון אינסופי")
- היצמדות לתבנית בלי בחינת הפרטים \Leftarrow מצליחים ל"הוכיח" ששפות רגולריות הן לא רגולריות

דיוק בפרטים

למת הניפוח

משפט 4.1: למת הניפוח לשפות רגולריות

תהי L שפה רגולרית. אז קיים מספר טבעי n כך שכל מילה z ב- L , שאורכה לפחות n , ניתנת לפירוק בצורה $z=uvw$ באופן שמתקיימים התנאים האלה:

$$|uv| \leq n .1$$

$$1 \leq |v| .2$$

$$uv^i w \in L \text{ לכל } 0 \leq i .3$$

$\{a^n b^n \mid n \geq 0\}$ אינה שפה רגולרית הוכחה על ידי למת הניפוח:

נניח שהשפה רגולרית.

לכן, מלמת הניפוח, קיים מספר טבעי n שמקיים את תנאי הלמה.

נבחר את המילה $z = a^n b^n$.

על פי הלמה קיים פירוק $z = a^n b^n = uvw$ כך ש:

$$u = a^s$$

$$v = a^t, t \geq 1$$

$$w = a^{n-s-t} b^n$$

ומתקיים, לכל $i \geq 0$: $uv^i w \in L$.

בפרט, עבור $i = 0$:

$$uv^0 w = uw = a^s a^{n-s-t} b^n = a^{n-t} b^n$$

כלומר, $a^{n-t} b^n \in L$,

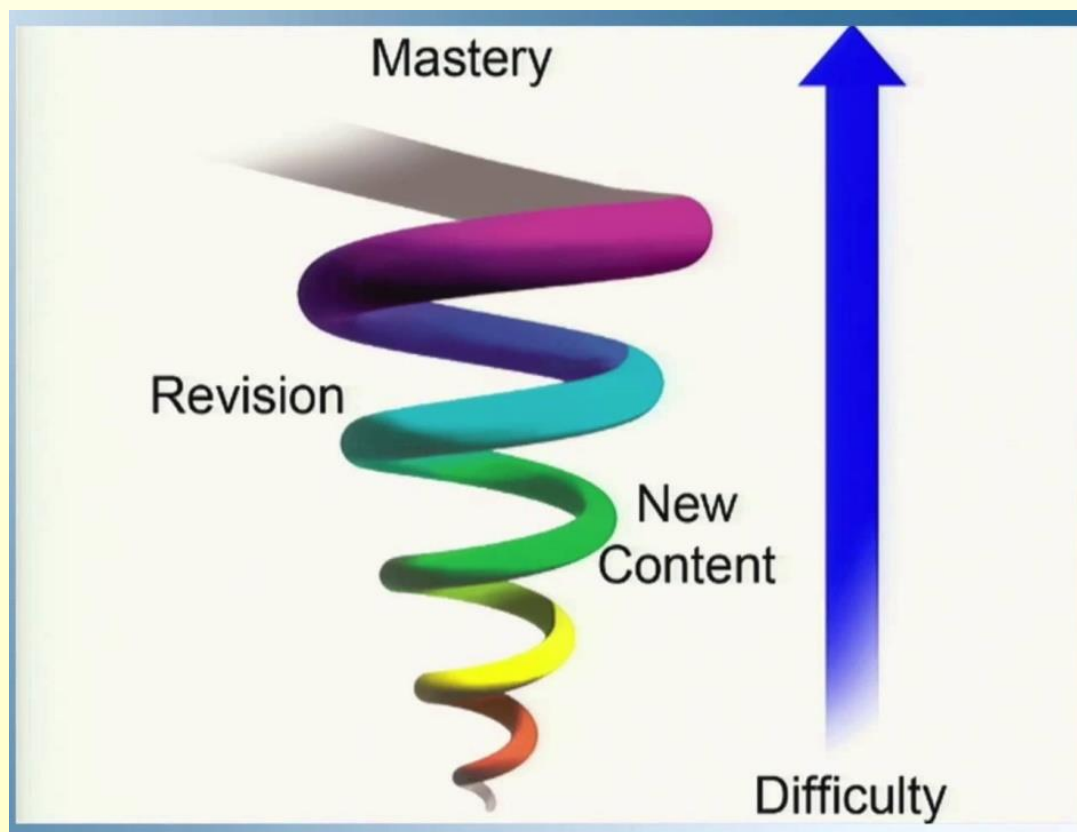
אבל, $n-t \neq n$ ולכן $a^{n-t} b^n \notin L$

סתירה!

Bruner (1960)

“Any subject can be taught effectively in some intellectually honest form to any child in any stage of development”

הוראה ספירלית (Bruner, 1960)





סעיף 3.3 – תכונות של משפחת השפות הרגולריות

מטרות

- היכרות עם שאלות תיאורטיות ועם אופי הדיון בהן
- כלי ל**הוכחה** ששפות נתונות הן רגולריות
- כלי ל**בניית** אוטומט עבור שפה רגולרית נתונה

סעיף 3.3 – תכונות הסגירות הנדונות

- | | | |
|---------------------|---|------------------|
| דוגמה נגדית |  | ● סגירות לחלקיות |
| הוכחה קונסטרוקטיבית |  | ● סגירות למשלים |
| הוכחה קונסטרוקטיבית |  | ● סגירות לחיתוך |
| הוכחה קונסטרוקטיבית |  | ● סגירות לאיחוד |

סוגי הוכחות

שימושים לתכונות סגירות - דוגמאות

- גם שאלות בנייה וגם שאלות הוכחה
- שאלות הוכחה שאינן מחייבות בניה הן שאלות תיאורטיות, ברמת הפשטה גבוהה יותר
 - ◆ הוכחה קיומית ולא קונסטרוקטיבית
 - ◆ יש בהן מרכיב טכני, ניתוח והסקה
 - ◆ אין הרבה עבודה טכנית
- ליבון סוגיית השפה הריקה מכיוון החיתוך (נורית רייך)

הוכחת תכונות סגירות – בעיות אופייניות

- המצאת "תכונות" חדשות (שפת החיתוך של שפה רגולרית עם שפה לא רגולרית היא רגולרית)
- התעלמות מהטכניקות המוצעות ובניית אוטומט "ישיר"

אי-ניצול תכונות סגירות

התעלמות מהטכניקות המוצעות ובניית
אוטומט "ישיר"

אי-ניצול תכונות סגירות

◆ להדגיש יתרונות:

■ בניית אוטומטים פשוטים מאוד ואחר כך הליך טכני, הפעלת אלגוריתם

■ במקרים מסוימים – שימוש בקיים

■ פחות מאמץ

■ סיכוי קטן יותר לטעויות

◆ להקביל לתכנות – מודולריות

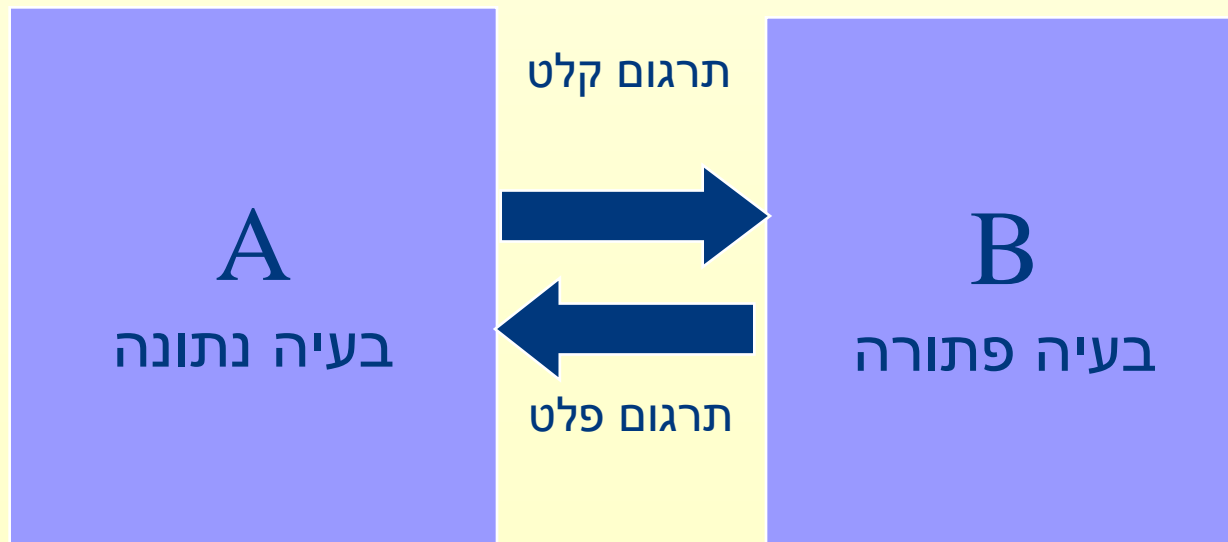
חינוך לחשיבה רדוקטיבית

פתרון בעיה על ידי רדוקציה

תרגום הבעיה הנתונה לבעיות
(בעיה) פשוטה יותר (או בעיות
שפתרון כבר ידוע) ושימוש
בפתרון של הבעיות האלו כדי
לפתור את הבעיה הנתונה.

יתרונות

- ניצול מה שכבר ידוע – מוביל לפתרונות פחות מורכבים וקלים יותר לפתרון



מאפיינים

- להתרכז במה שחשוב
- "מה" ולא "איך"
- התעלמות מפרטי ביצוע

שימוש ברדוקציה ב"יסודות מדעי המחשב"

◆ מציאת האיבר השכיח ביותר

■ רדוקציה לבעיית מיון

◆ מציאת מקסימום שני

■ רדוקציה לבעיית מציאת מקסימום

◆ שימוש בפעולות על מחרוזות

◆ פיתוח top-down

רעיונות יסודיים (Schwill, 1994)

- הקריטריון **האופקי** (רלבנטי להרבה תחומים של מדעי המחשב, בהקשרים שונים וצורות שונות.)
- הקריטריון **האנכי** (רמות אינטלקטואליות שונות)
 - ◆ הוראה ספירלית
- קריטריון **הזמן** (התפתחות היסטורית)
- קריטריון **המובן** (ניתן לייחס לחשיבה ושפה יומיומיות)

רדוקציה היא רעיון יסודי של מדעי המחשב

- הקריטריון **האופקי** (רלבנטי להרבה תחומים של מדעי המחשב, בהקשרים שונים וצורות שונות).
- הקריטריון **האנכי** (רמות אינטלקטואליות שונות)
 - ◆ הוראה ספירלית
- קריטריון **הזמן** (התפתחות היסטורית)
- קריטריון **המובן** (ניתן לייחס לחשיבה ושפה יומיומיות)

חינוך לחשיבה רדוקטיבית - דוגמאות

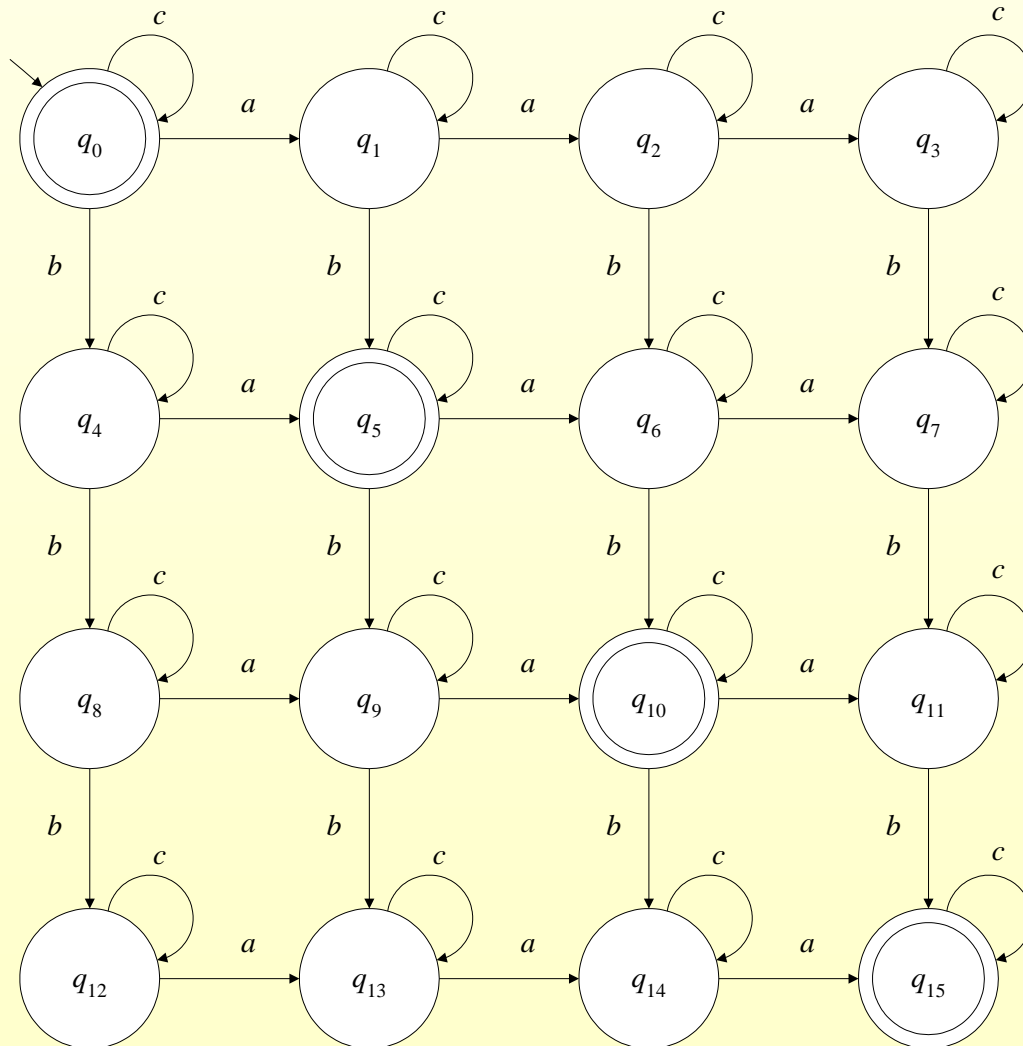
שאלה 1

נתבונן בשפה מעל הא"ב $\{a, b, c\}$ המכילה בדיוק את כל המילים שמקיימות לפחות אחד מבין שני התנאים הבאים:

1. מספר האותיות a במילה שווה למספר האותיות b במילה, והסכום של מספר האותיות a ומספר האותיות b הוא לכל היותר 6.

2. המילה מכילה את הרצף abc ומסתיימת ברצף bb .
האם שפה זו היא רגולרית? הוכח את תשובתך.

שאלה 1 – פתרון ישיר



שאלה 1 – פתרון רדוקטיבי ראשון

$$L_0 = \{w \mid \#_a(w) = \#_b(w) = 0\}$$

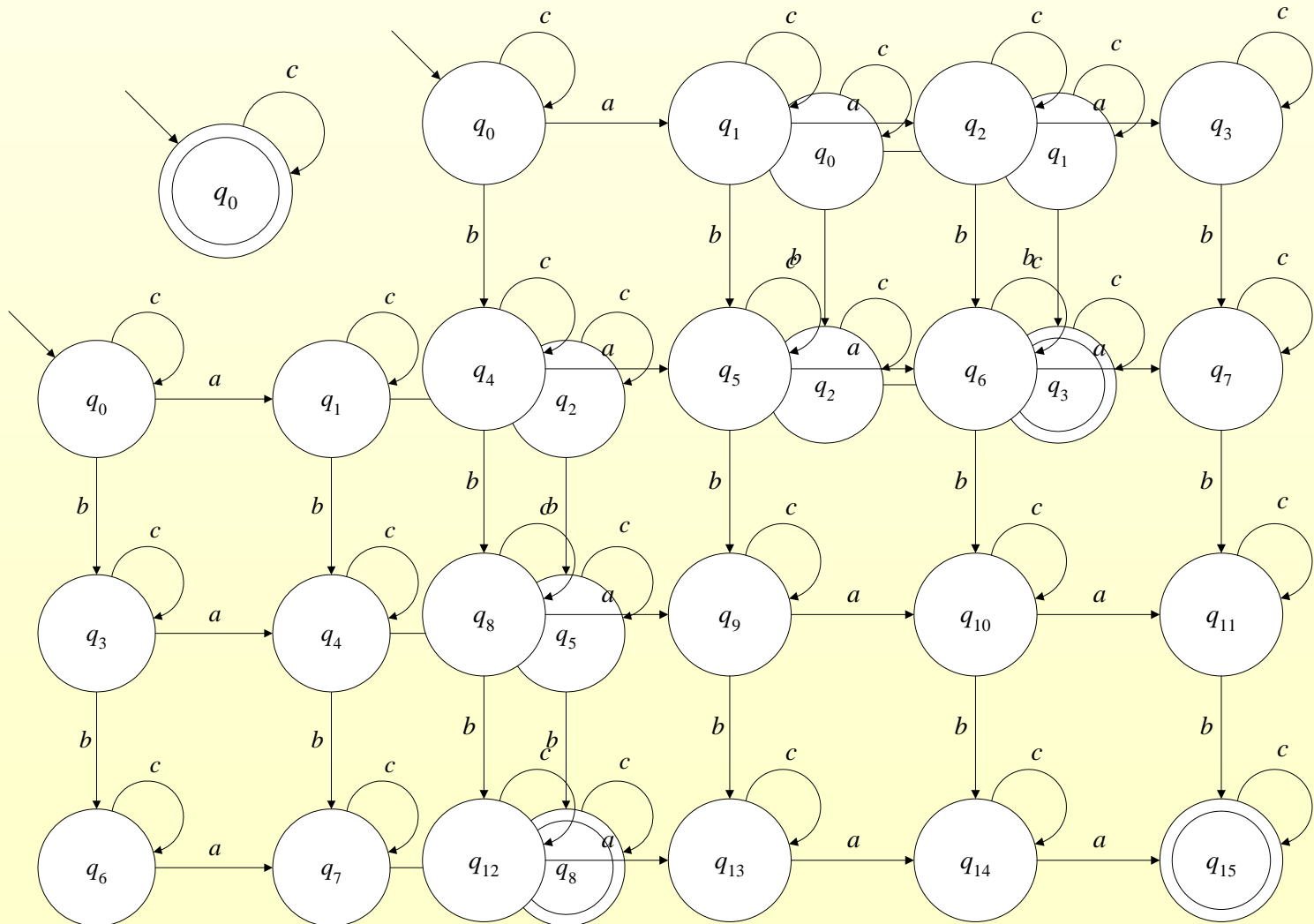
$$L_1 = \{w \mid \#_a(w) = \#_b(w) = 1\}$$

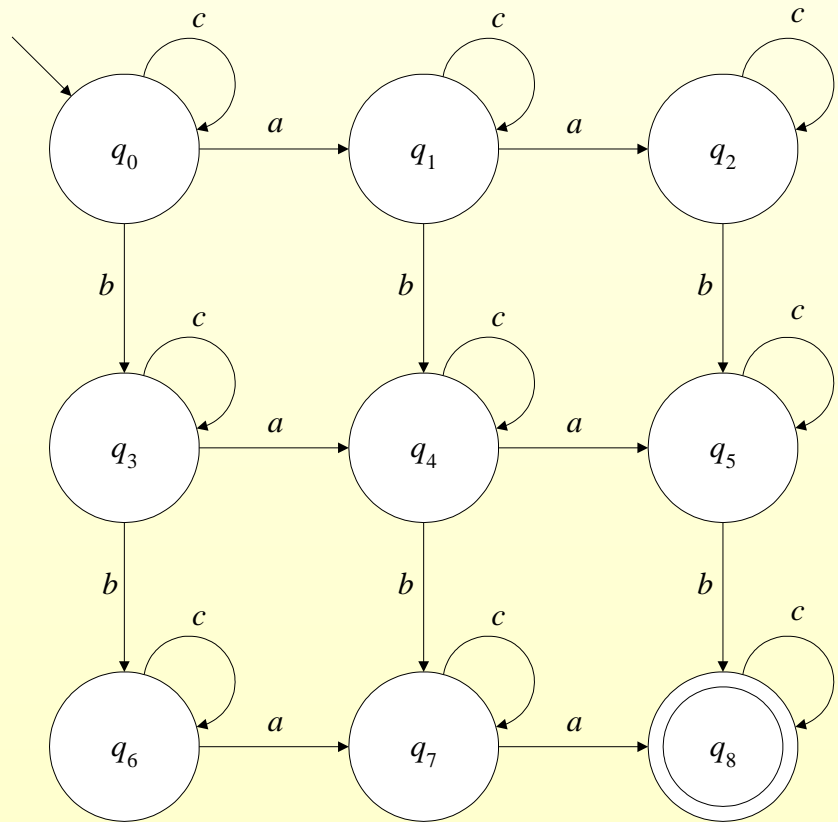
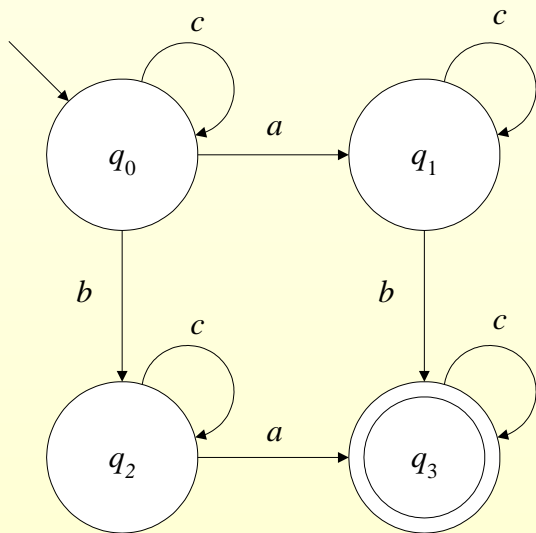
$$L_2 = \{w \mid \#_a(w) = \#_b(w) = 2\}$$

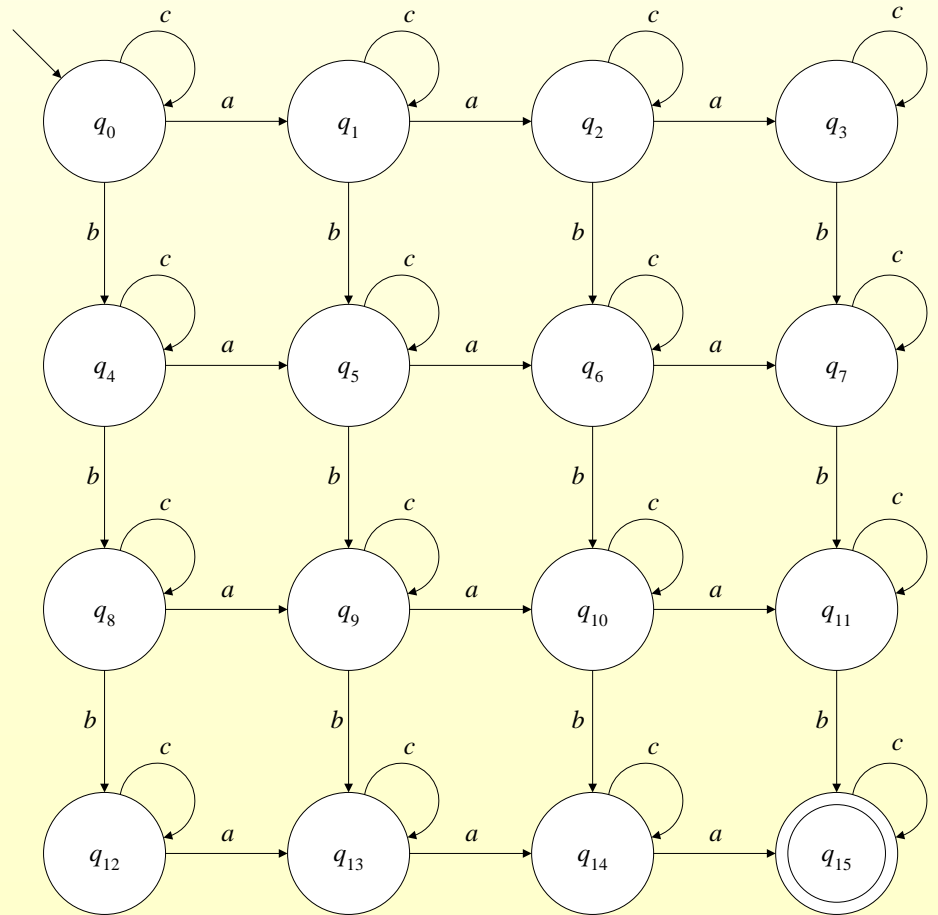
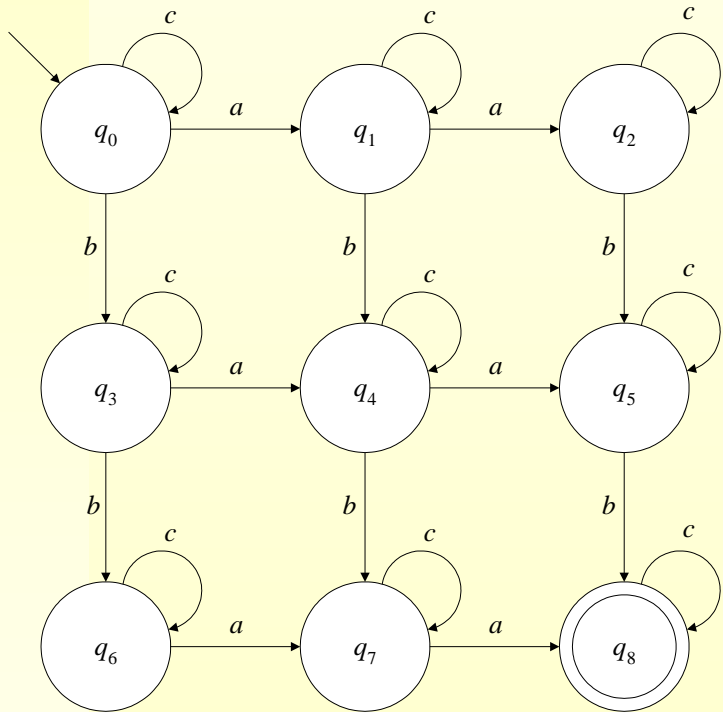
$$L_3 = \{w \mid \#_a(w) = \#_b(w) = 3\}$$

$$L = L_0 \cup L_1 \cup L_2 \cup L_3$$

פתרון ראשון – האוטומטים המתאימים







שאלה 1 – פתרון רדוקטיבי שני

$$L_0 = \{w \mid \#_a(w) = \#_b(w) = 0\}$$

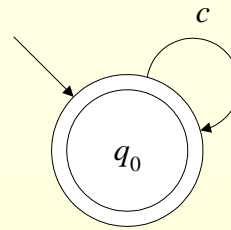
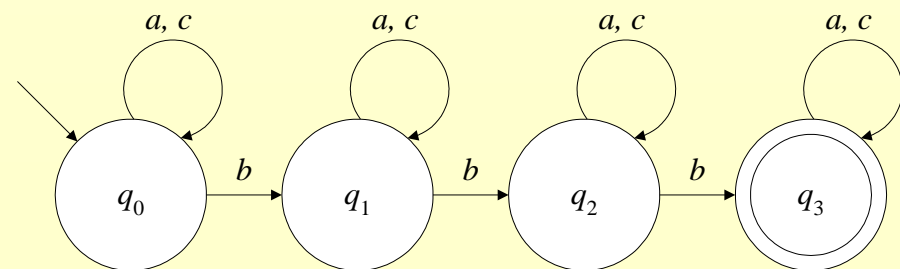
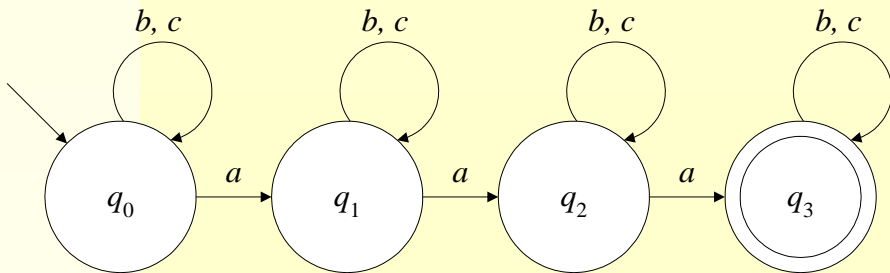
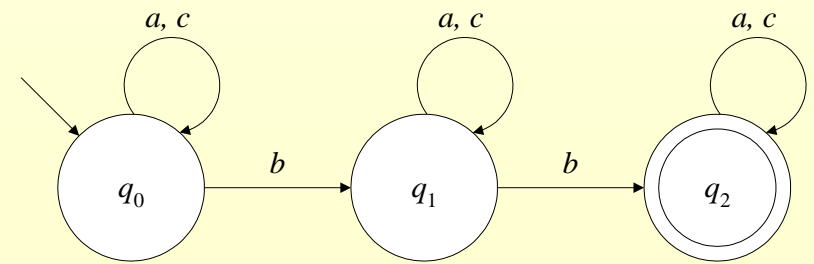
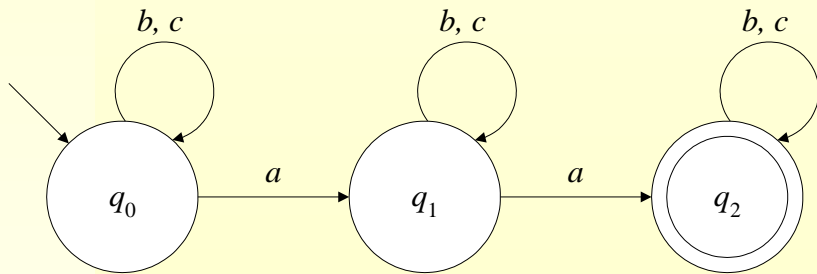
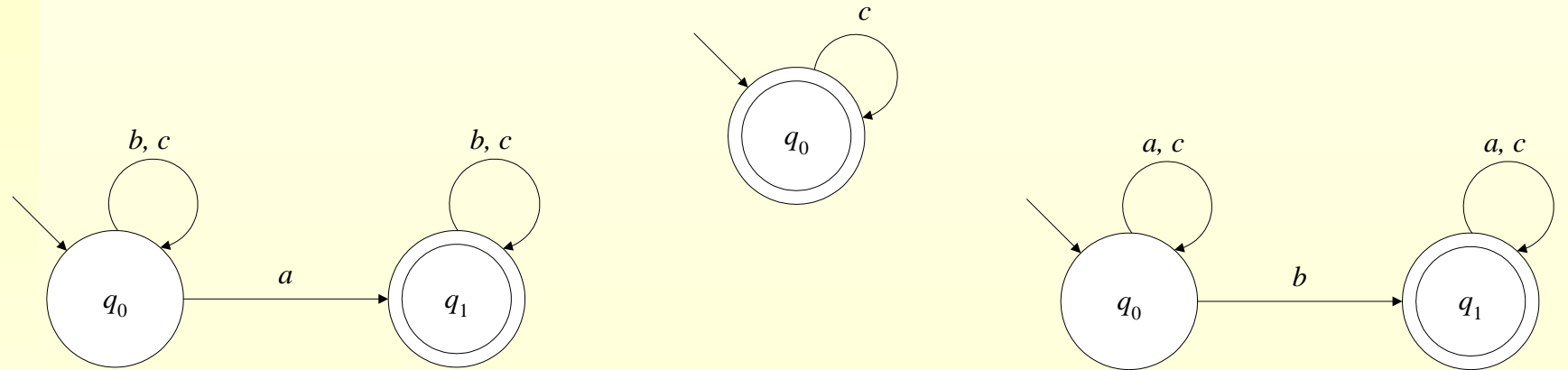
$$L_4 = \{w \mid \#_a(w) = 1\} \quad L_5 = \{w \mid \#_b(w) = 1\}$$

$$L_6 = \{w \mid \#_a(w) = 2\} \quad L_7 = \{w \mid \#_b(w) = 2\}$$

$$L_8 = \{w \mid \#_a(w) = 3\} \quad L_9 = \{w \mid \#_b(w) = 3\}$$

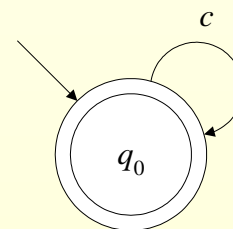
$$L = L_0 \cup (L_4 \cap L_5) \cup (L_6 \cap L_7) \cup (L_8 \cap L_9)$$

פתרון שני – האוטומטים המתאימים

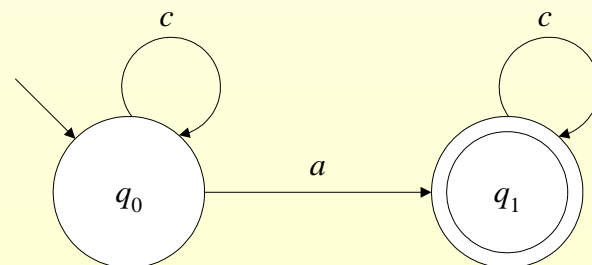


שאלה 1 – פתרון רדוקטיבי שלישי

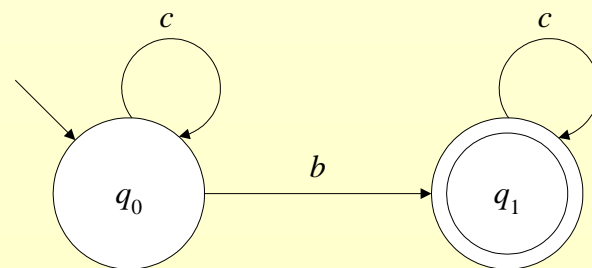
$$L_0 = \{w \mid \#_a(w) = \#_b(w) = 0\}$$



$$L_{10} = \{w \mid \#_a(w) = 1 \text{ and } \#_b(w) = 0\}$$



$$L_{11} = \{w \mid \#_a(w) = 0 \text{ and } \#_b(w) = 1\}$$



$$\begin{aligned}
L = & L_0 \cup (L_{10} \cdot L_{11}) \cup (L_{11} \cdot L_{10}) \cup \\
& (L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{11}) \cup (L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{11}) \cup \\
& (L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{10}) \cup (L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{11}) \cup \\
& (L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{10}) \cup (L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{10}) \cup \\
& (L_{10} \cdot L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{11}) \cup (L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{11}) \cup \\
& (L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{11}) \cup (L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{11} \cdot L_{10}) \cup \\
& (L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{11}) \cup (L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{11}) \cup \\
& (L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{10}) \cup (L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{11}) \cup \\
& (L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{10}) \cup (L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{10}) \cup \\
& (L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{11}) \cup (L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{11}) \cup \\
& (L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{10}) \cup (L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{11}) \cup \\
& (L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{10}) \cup (L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{10}) \cup \\
& (L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{10} \cdot L_{11}) \cup (L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{11} \cdot L_{10}) \cup \\
& (L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{11} \cdot L_{10} \cdot L_{10}) \cup (L_{11} \cdot L_{11} \cdot L_{11} \cdot L_{10} \cdot L_{10} \cdot L_{10})
\end{aligned}$$

עקרון ה-Trade-off

פתרונות מתוחכמים יותר (רמת חשיבה
רדוקטיבית גבוהה יותר) משרים
סיבוכיות בנייה נמוכה יותר

חינוך לחשיבה רדוקטיבית - דוגמאות

שאלה 2

יהיו L_1 ו- L_2 השפות הבאות מעל הא"ב $\{a, b\}$:
 L_1 היא שפת כל המילים המכילות אך ורק מספר זוגי של אותיות b .

L_2 היא השפה $\{ba^n \mid n \geq 0\}$.

מהן $L_1^2, L_1 \cdot L_2$? האם הן רגולריות? הוכח.

חינוך לחשיבה רדוקטיבית - דוגמאות

שאלה 3

נתבונן בשפה מעל הא"ב $\{a, b, c\}$ המכילה את כל המילים שמקיימות את שני התנאים הבאים:

1. אחד התנאים הבאים מתקיים:

א. החלק הראשון של המילה מכיל את הרצף aa והחלק השני של המילה מכיל את הרצף bb .

ב. המילה מסתיימת ברצף cc .

2. המילה מתחילה ברצף $aaba$.

האם שפה זו היא רגולרית? הוכח את תשובתך.

חינוך לחשיבה רדוקטיבית - דוגמאות

שאלה 4

נתבונן בשפה מעל האי"ב $\{a, b, c\}$ המכילה בדיוק את כל המילים שמקיימות את שני התנאים הבאים:

1. מקיימות לפחות אחד מבין שני התנאים הבאים:

א. מורכבות משני חלקים, כאשר בחלק הראשון של המילה מופיע הרצף ba ובחלק השני מופיע הרצף ab .

ב. מסתיימות ברצף abc .

2. מכילות את הרצף ba וגם את הרצף ab .

האם שפה זו היא רגולרית? הוכח את תשובתך.

חינוך לחשיבה רדוקטיבית - דוגמאות

שאלה 5

נתונה שפת כל המילים מהצורה $a^n b^m c^k$, $n, m, k > 0$,
כך שמספר האותיות a בכל מילה זוגי או שמספר
האותיות c בכל מילה הוא לפחות 3. האם שפה זו
רגולרית? הוכח את תשובתך.

חינוך לחשיבה רדוקטיבית - דוגמאות

שאלה 6

נתבונן בשפת כל המילים מעל הא"ב $\{a, b, c\}$
שמקיימות לפחות אחד מבין שני התנאים הבאים:

1. אחרי כל a מופיע bb וגם מספר האותיות a זוגי.
 2. המילה מורכבת משני חלקים כך שבחלק הראשון מספר האותיות a זוגי ובחלק השני מספר האותיות b זוגי.
-

חינוך לחשיבה רדוקטיבית - דוגמאות

שאלה 7

האם השפה $\{(ab)^n (ba)^m \mid n, m \geq 0\}$ מעל הא"ב $\{a, b\}$ היא רגולרית? הוכח את תשובתך.

חינוך לחשיבה רדוקטיבית - דוגמאות

שאלה 8

האם השפה $\{a^i b^j c^k \mid k = i \pmod{3}\}$ מעל הא"ב $\{a, b, c\}$ היא רגולרית? הוכח את תשובתך.

פרק 4 – מודלים נוספים של אוטומט סופי

מטרות

1. היכרות עם אופי העבודה התיאורטית
♦ משחק עם הגדרות פורמליות שצמחו על
רקע מציאותי, תוך התנתקות מהמציאות
2. הצגת מושג האי-דטרמיניזם
3. הצגת מושג ההשוואה בין כוח חישוב של
מודלים שונים

אוטומט סופי לא דטרמיניסטי

מוטיבציה להרחבה:

- המשחק התיאורטי
- במקרים מסוימים – הלך מחשבה טבעי (דוגמה פותחת)
- הוכחות קלות יותר לרגולריות (תכנון קל יותר)
- הוכחות קלות יותר של תכונות סגירות נוספות (סעיף 4.4)

אוטומט סופי לא דטרמיניסטי – בעיות הטמעה

● המודל אינו מציאותי, קשה לתלמידים להתמודד עימו

◆ הימנעות משימוש באי-דטרמיניזם

◆ קושי להתרגל לטכניקה

■ ניצול חלקי של אי-דטרמיניזם

■ איכות נמוכה של אי-דטרמיניזם

◆ הבנה "לא דטרמיניסטית" של המודל – עבודה ברמת הפשטה נמוכה

● "האנשת" האוטומט: סימטריה בין דחייה לקבלה

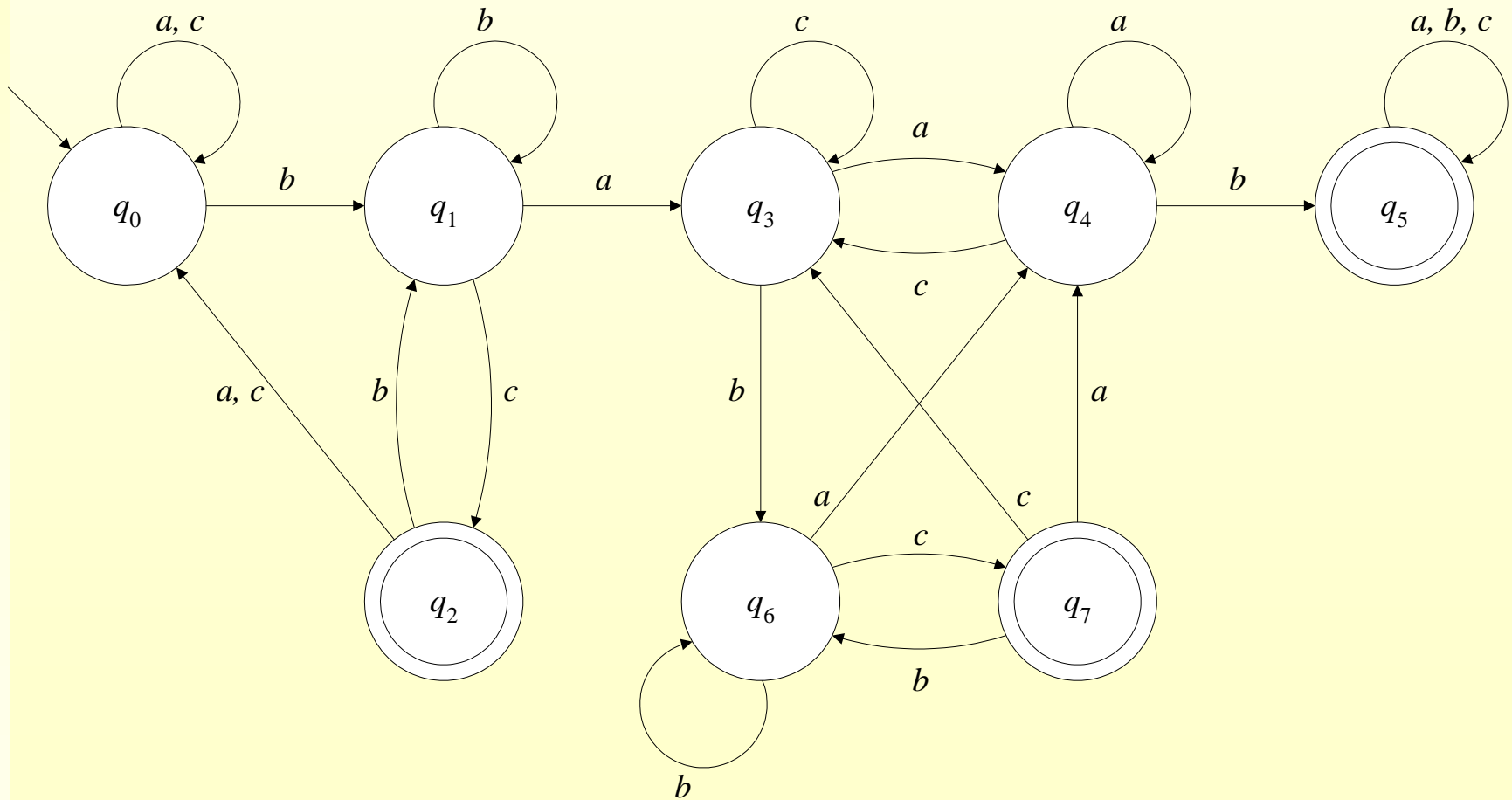
הימנעות משימוש באי-דטרמיניזם

בנה אוטומט המקבל את השפה מעל האי"ב $\{a, b, c\}$
המכילה בדיוק את כל המילים שמקיימות לפחות
אחד מבין שני התנאים הבאים:

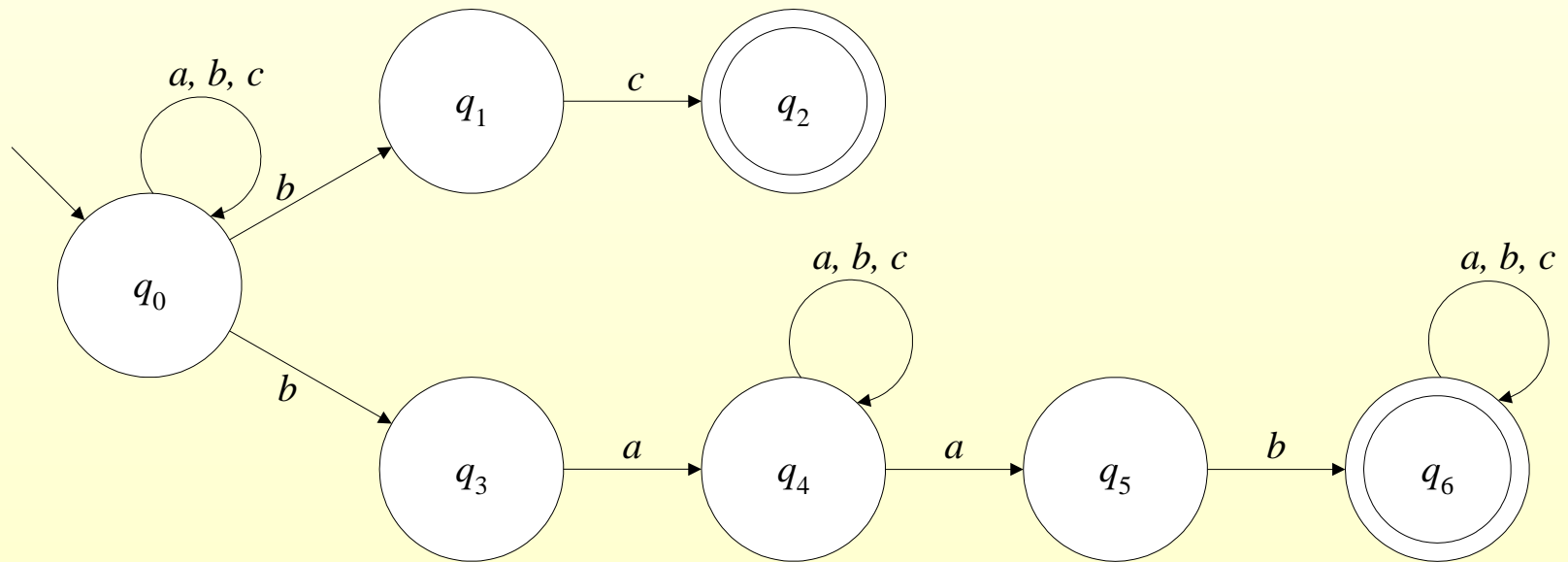
1. המילה מסתיימת ברצף bc .

2. המילה מורכבת משני חלקים, כאשר החלק הראשון מכיל את הרצף ba , והחלק השני מכיל את הרצף ab .

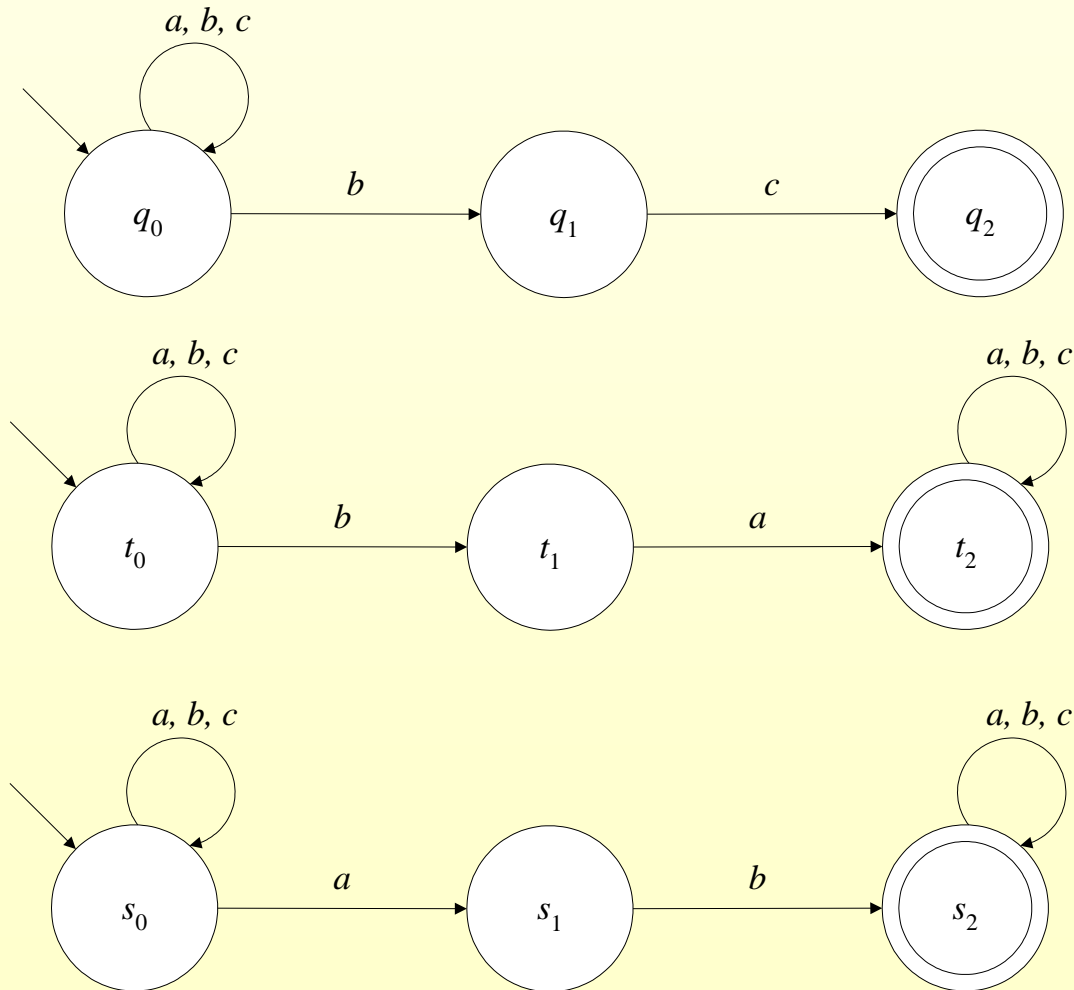
פתרון דטרמיניסטי ישיר



פתרון לא דטרמיניסטי ישיר

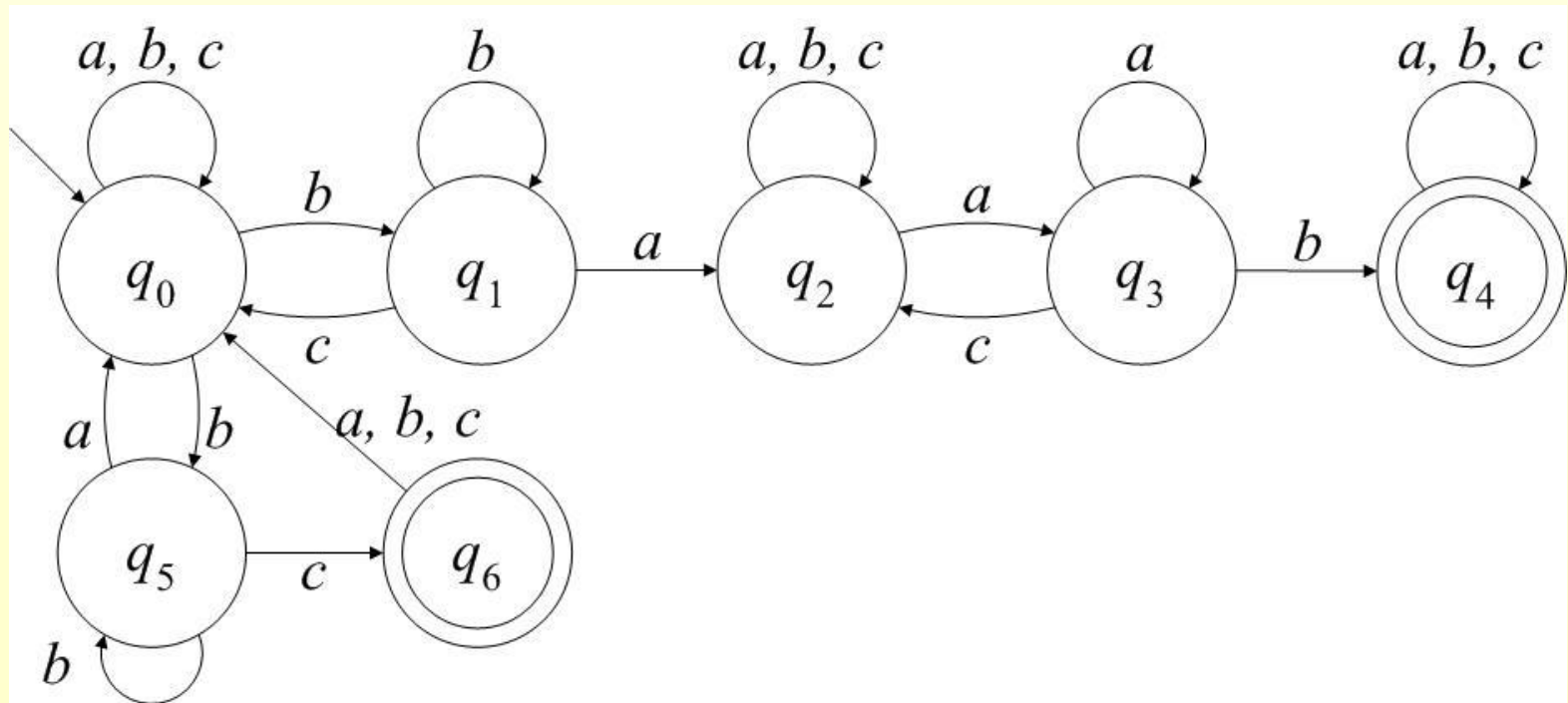


פתרון רדוקטיבי לא דטרמיניסטי



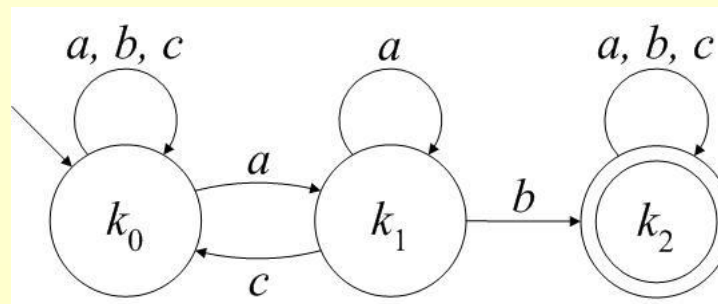
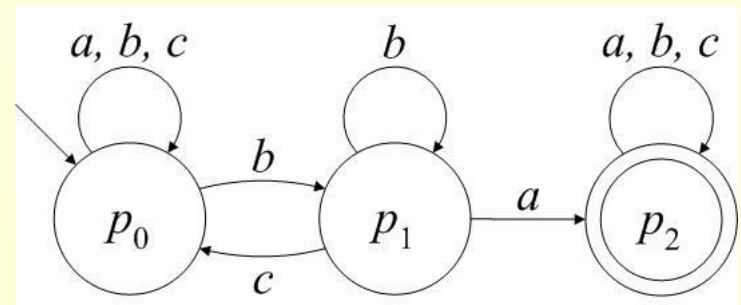
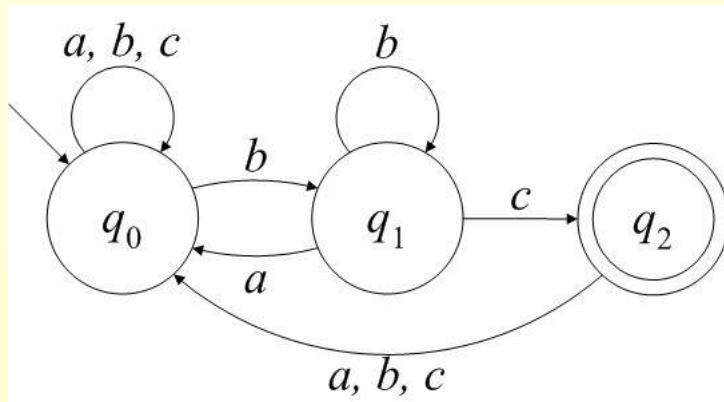
בעיות בהטמעת אי-דטרמיניזם – משפחה 1: פתרונות דטרמיניסטיים עם אי-דטרמיניזם לוקלי

דוגמה 1 – פתרון ישיר עם פירוק חבוי ל-3 שפות



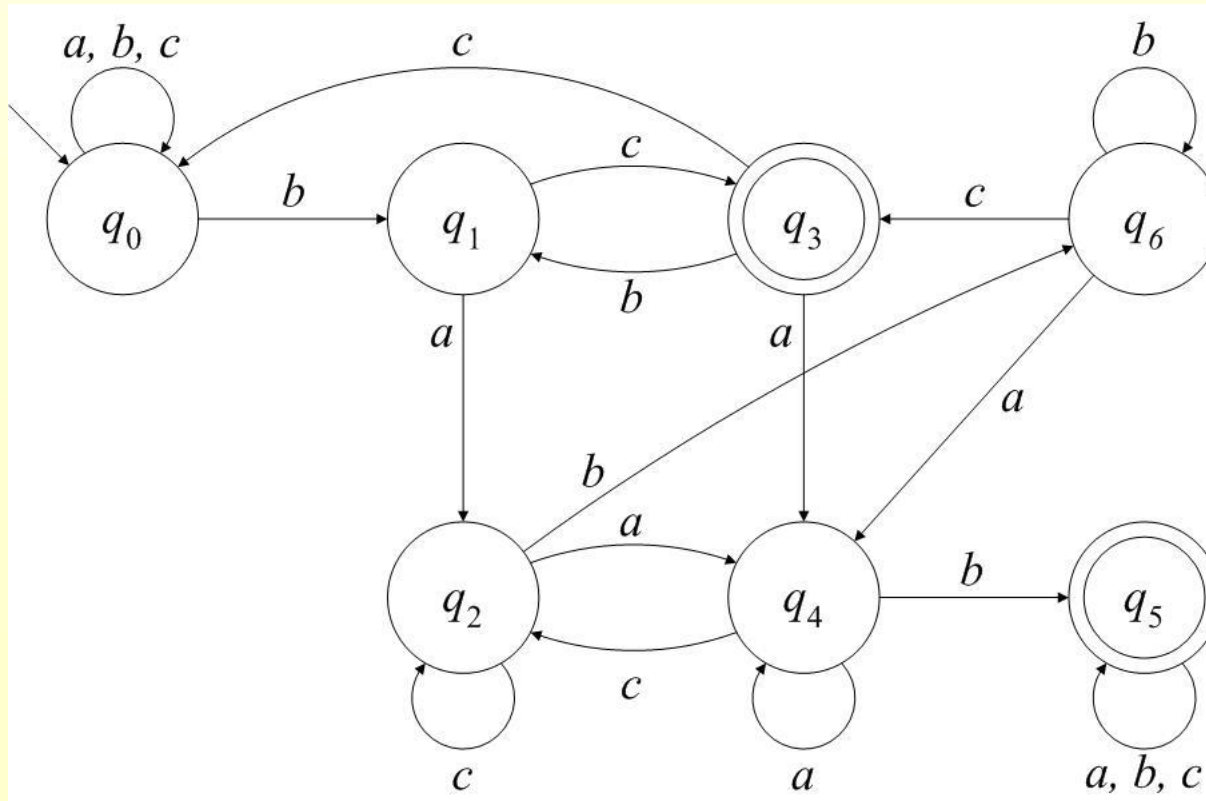
בעיות בהטמעת אי-דטרמיניזם – משפחה 1: פתרונות דטרמיניסטיים עם אי-דטרמיניזם לוקלי

דוגמה 2 – פירוק מלא ל-3 שפות בסיס



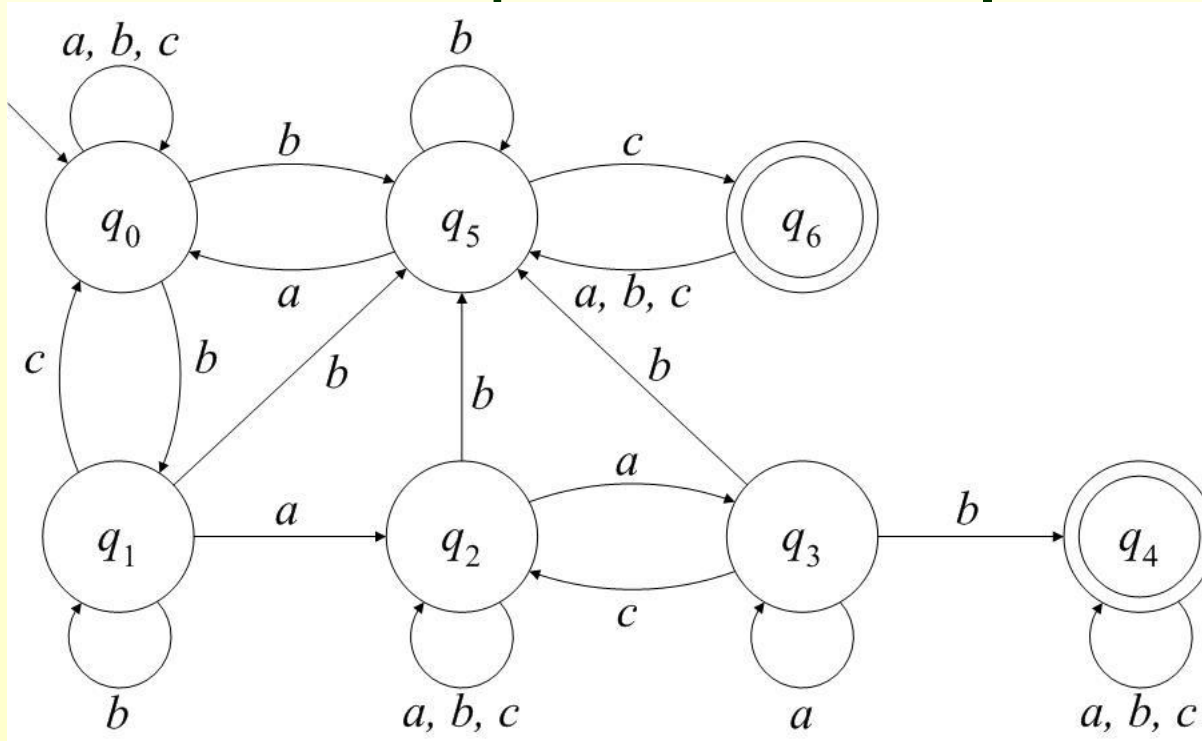
בעיות בהטמעת אי-דטרמיניזם – משפחה 1: פתרונות דטרמיניסטיים עם אי-דטרמיניזם לוקלי

דוגמה 3 – פתרון ישיר



בעיות בהטמעת אי-דטרמיניזם – משפחה 1: פתרונות דטרמיניסטיים עם אי-דטרמיניזם לוקלי

דוגמה 4 – פתרון ישיר עם פירוק חבוי ל-3 שפות

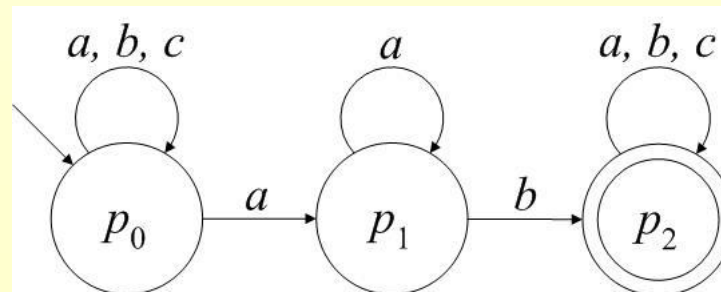
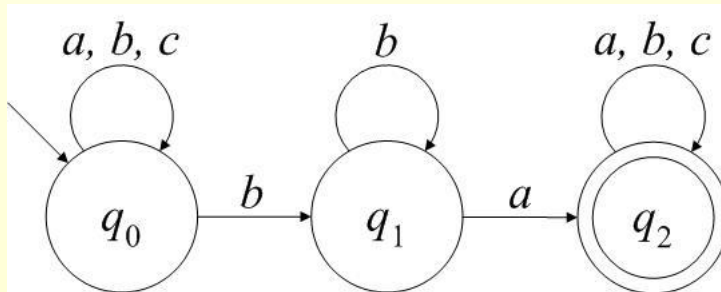


משפחה 1 - סיכום

- אוטומט דטרמיניסטי בבסיסו
- מעברים אי-דטרמיניסטיים מיותרים
- תופעה חוזרת: מעבר אי-דטרמיניסטי מיותר שהוא לולאה עצמית עם כל אותיות הא"ב במצב ההתחלתי

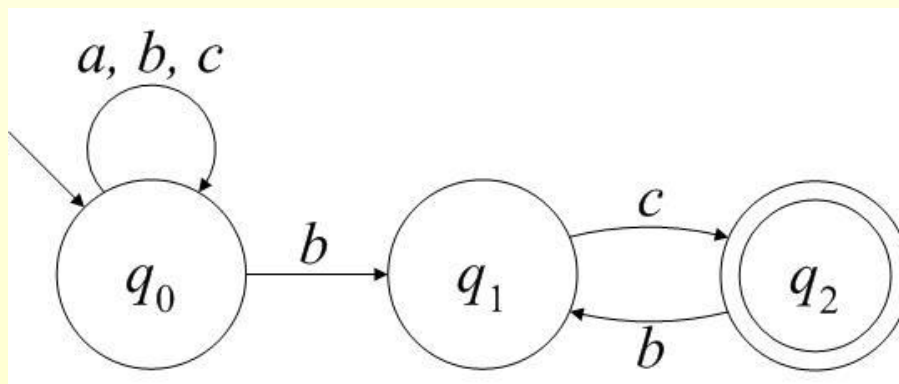
בעיות בהטמעת אי-דטרמיניזם – משפחה 2: פתרונות אי-דטרמיניסטיים עם סממנים דטרמיניסטיים

דוגמה 1 – פירוק מלא לשלוש שפות בסיס



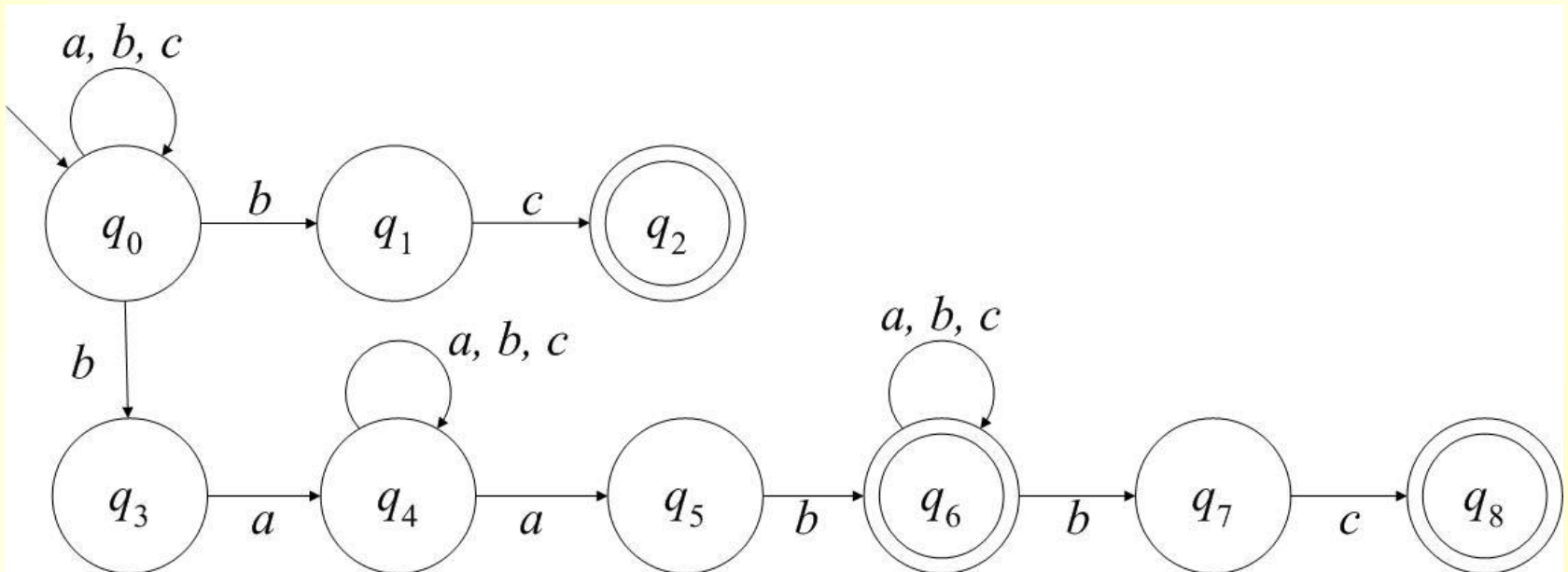
בעיות בהטמעת אי-דטרמיניזם – משפחה 2: פתרונות אי-דטרמיניסטיים עם סממנים דטרמיניסטיים

דוגמה 2 – פירוק מלא לשלוש שפות בסיס



בעיות בהטמעת אי-דטרמיניזם – משפחה 2: פתרונות אי-דטרמיניסטיים עם סממנים דטרמיניסטיים

דוגמה 3 – פתרון ישיר

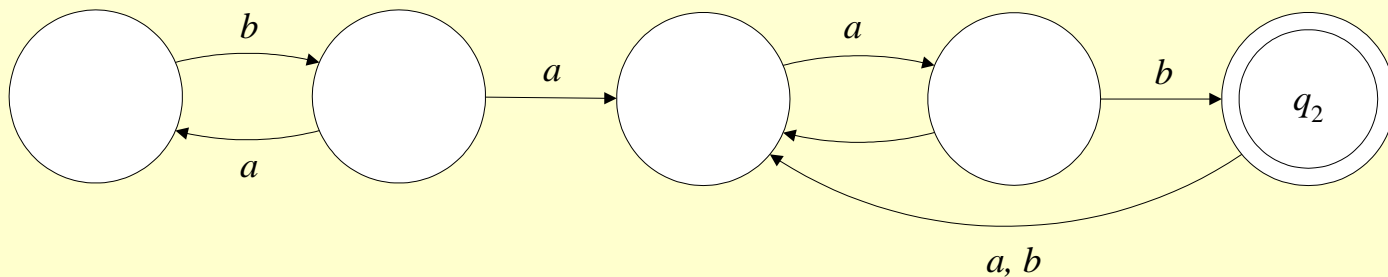
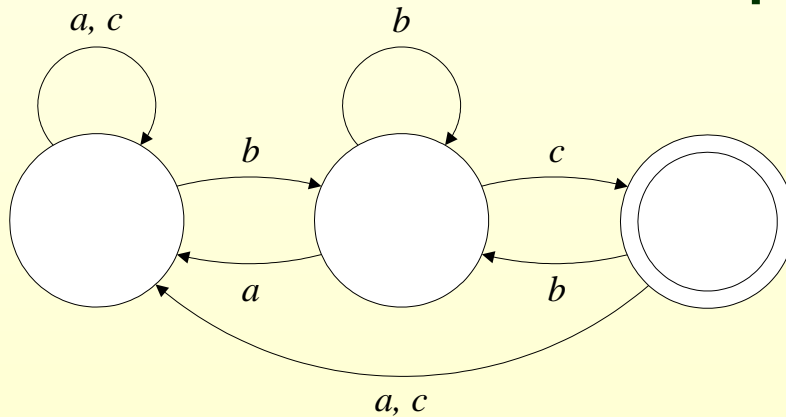


משפחה 2 - סיכום

- אוטומט אי-דטרמיניסטי או כמעט אי-דטרמיניסטי
- דטרמיניזם לוקאלי

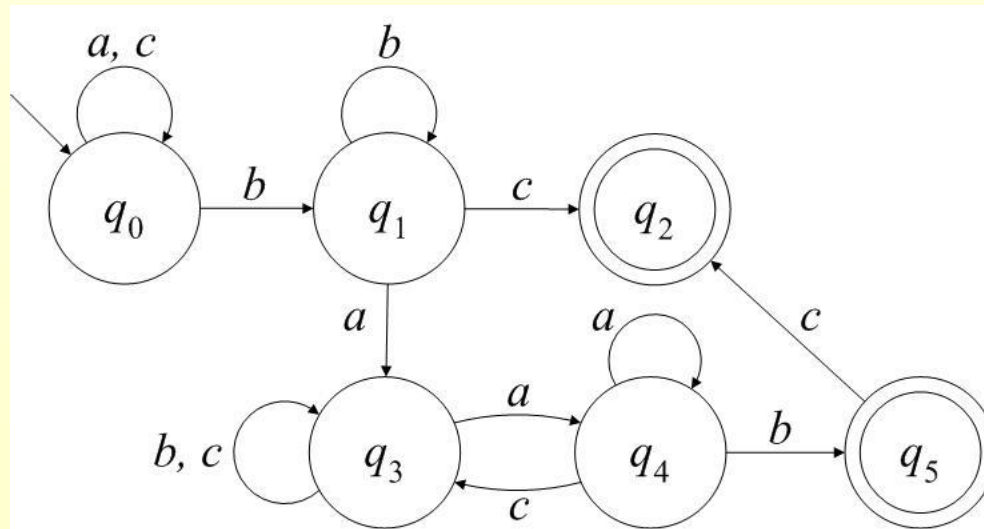
בעיות בהטמעת אי-דטרמיניזם – משפחה 3: פתרונות תהליכיים: מאי-דטרמיניזם אל דטרמיניזם

דוגמה 1 – נסיגה במהלך בנייה ישירה



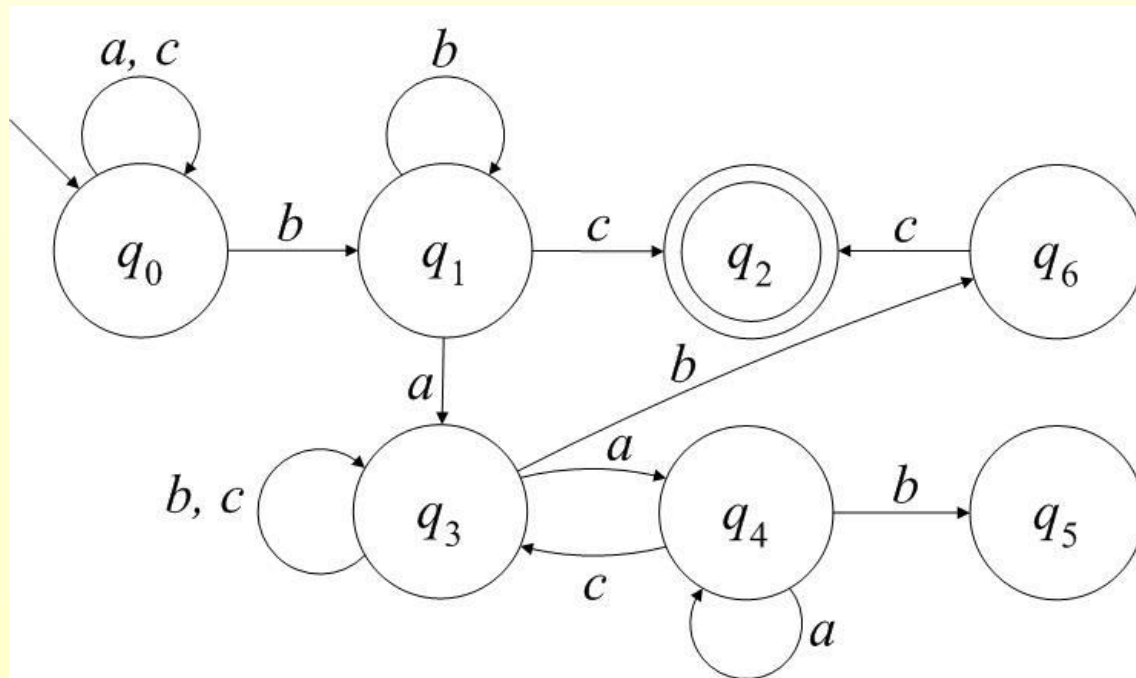
בעיות בהטמעת אי-דטרמיניזם – משפחה 3: פתרונות תהליכיים: מאי-דטרמיניזם אל דטרמיניזם

דוגמה 1 – נסיגה במהלך בנייה ישירה - המשך



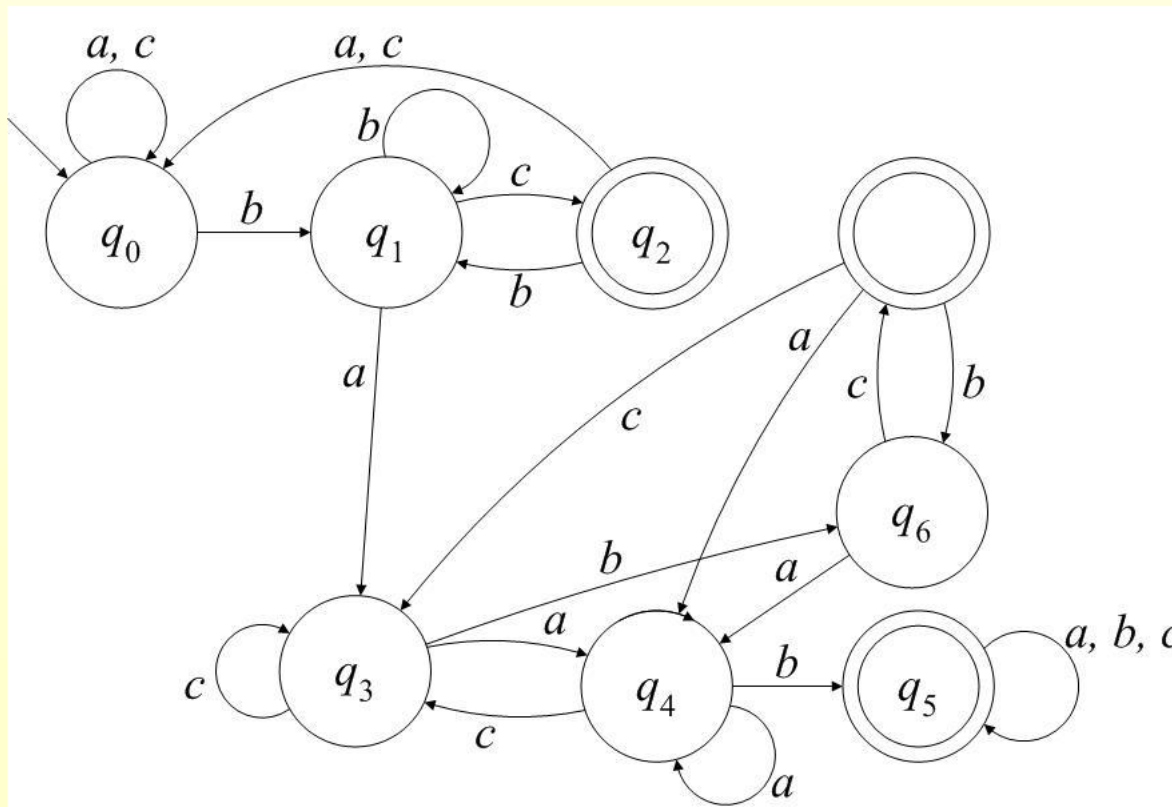
בעיות בהטמעת אי-דטרמיניזם – משפחה 3: פתרונות תהליכיים: מאי-דטרמיניזם אל דטרמיניזם

דוגמה 1 – נסיגה במהלך בנייה ישירה - המשך



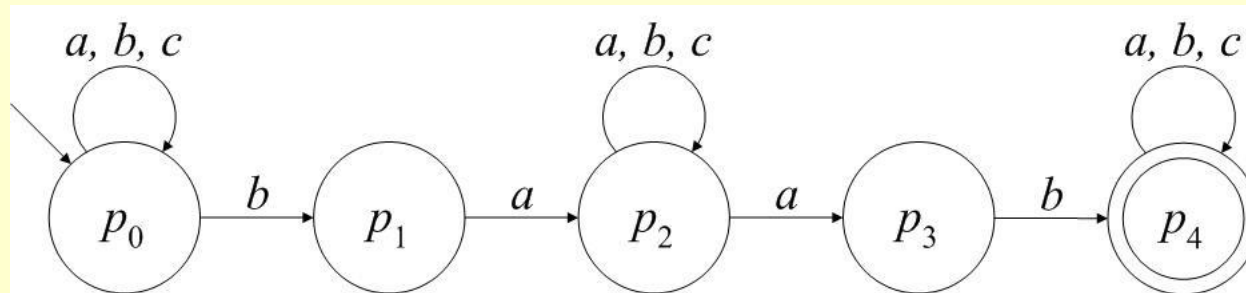
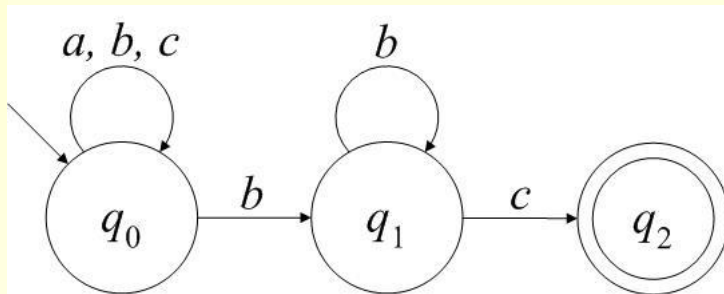
בעיות בהטמעת אי-דטרמיניזם – משפחה 3: פתרונות תהליכיים: מאי-דטרמיניזם אל דטרמיניזם

דוגמה 1 – נסיגה במהלך בנייה ישירה - המשך



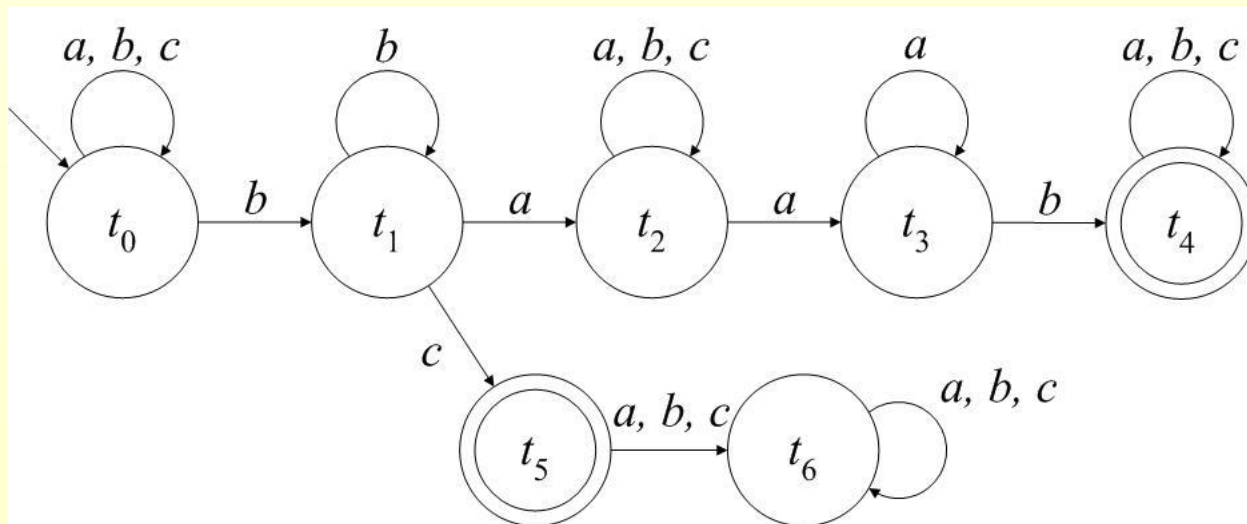
בעיות בהטמעת אי-דטרמיניזם – משפחה 3: פתרונות תהליכיים: מאי-דטרמיניזם אל דטרמיניזם

דוגמה 2 – נסיגה במהלך ביצוע רדוקציה



בעיות בהטמעת אי-דטרמיניזם – משפחה 3: פתרונות תהליכיים: מאי-דטרמיניזם אל דטרמיניזם

דוגמה 2 – נסיגה במהלך ביצוע רדוקציה

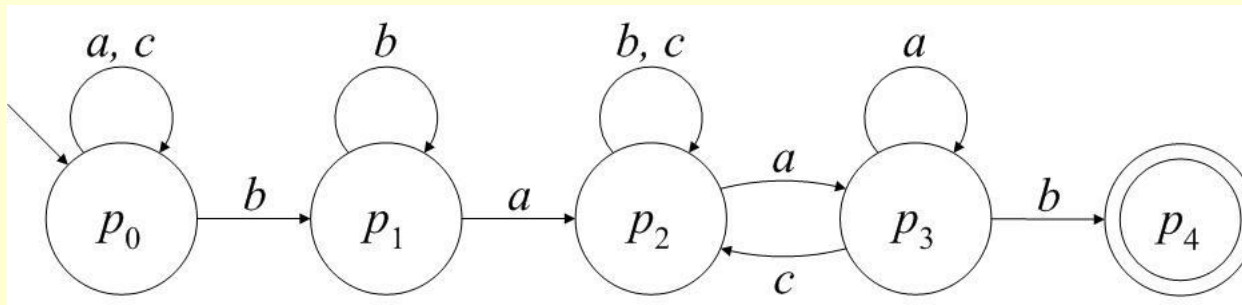
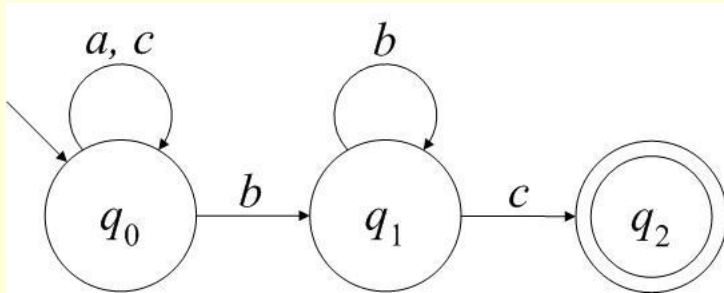


משפחה 3 - סיכום

- מעבר דרך אוטומט לא דטרמיניסטי נכון או כמעט נכון
- החלפתו באוטומט דטרמיניסטי (באופן מפורש או באופן סמוי)

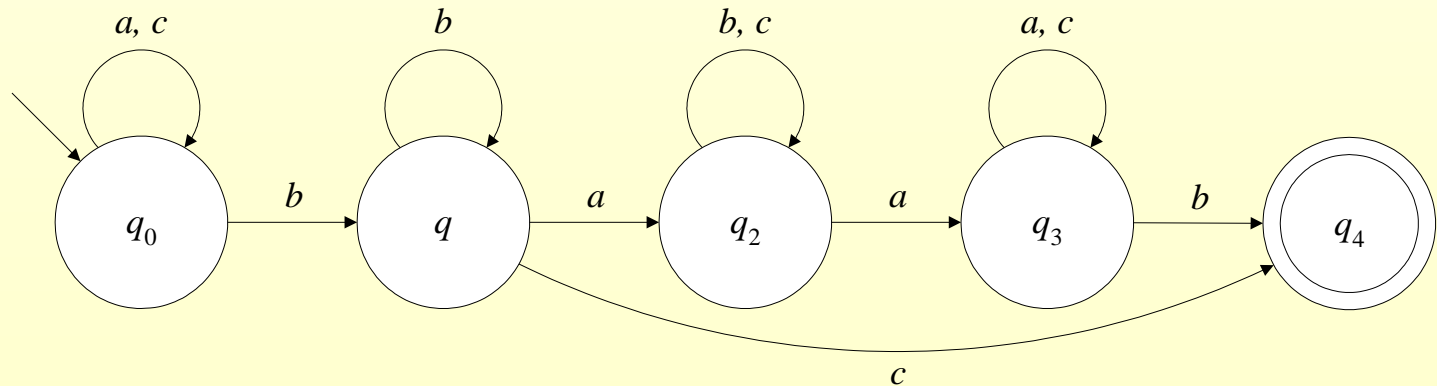
בעיות בהטמעת אי-דטרמיניזם – משפחה 4: פתרונות דטרמיניסטיים לא מלאים

דוגמה 1 – פירוק לשתי שפות בסיס



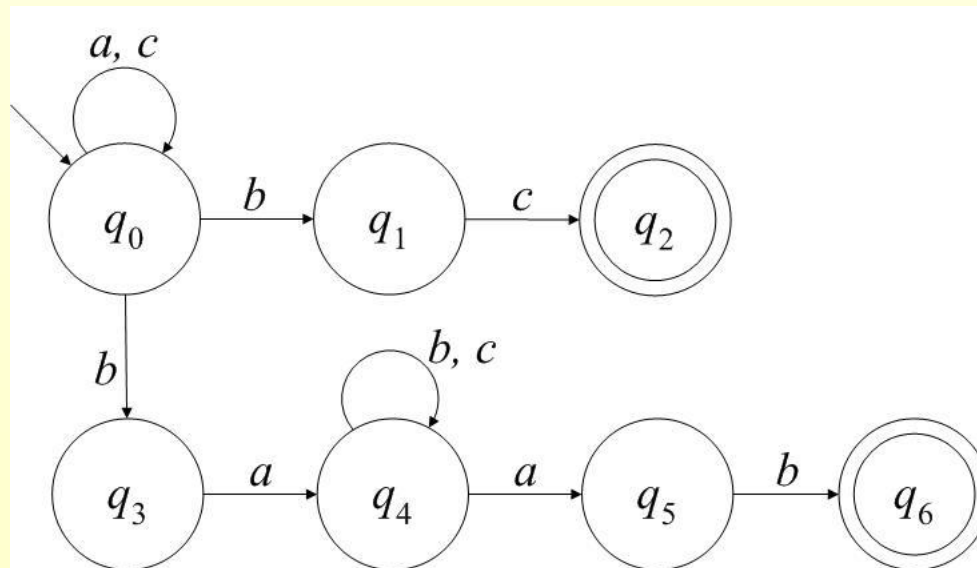
בעיות בהטמעת אי-דטרמיניזם – משפחה 4: פתרונות דטרמיניסטיים לא מלאים

דוגמה 2 – פתרון ישיר עם עקבות פירוק



בעיות בהטמעת אי-דטרמיניזם – משפחה 4: פתרונות דטרמיניסטיים לא מלאים

דוגמה 3 – פתרון ישיר



משפחה 4 - סיכום

- נטייה הפוכה לזו שהודגמה במשפחה 1 – חסרון מעברים לולאתיים חיוניים
- ייתכן כי תלמידים "מוכנים" לקבל את המודל הדטרמיניסטי הלא-מלא אך קשה להם להתמודד עם המודל הלא דטרמיניסטי

אי-דטרמיניזם – מבט מחקרי

מחקרים קודמים

- קשיים בהבנת האופי הלא דטרמיניסטי של מערכות חישוב מקבילי (Ben-David Kolikant, 2004a, 2004b)
- קשיים בלמידת מודלים חישוביים אי-דטרמיניסטיים (Armoni & Gal-Ezer, 2006, 2007)
- ◆ העדפה של תכנונים דטרמיניסטיים
- ◆ ניצול חלקי של כוח התיאור של מודלים לא דטרמיניסטיים

המחקר (1)

- מטרה ראשונית: לבדוק אפקטיביות של אסטרטגיה דידקטית להוראת אוטומטים לא-דטרמיניסטיים (השפעה על הנטייה לתכנונים אי-דטרמיניסטיים ואיכות האי-דטרמיניזם).
- אוכלוסיה: סטודנטים למדעי המחשב במכללה אקדמית הלומדים קורס חובה בתחום.
- שלב מקדים: ראיונות עם 6 סטודנטים שכבר סיימו את הקורס באותה מכללה אצל אותה מרצה.

ראיונות (שלב מקדים)

כל הסטודנטים תפסו אוטומטים לא
דטרמיניסטיים כמכונות **עקביות** – תמיד
נותנות אותה תשובה לאותו קלט

ציטוטים

- "[המכונה] תמיד תיתן אותה תשובה בכל מקרה [דחייה או קבלה]."
- "זה רץ במקביל."
- "זה מתפצל: בעץ של כל המסלולים הוא עושה גם את המסלול הזה וגם את המסלול הזה."
- "הוא לא מנחש. הוא בודק את כל המסלולים."
- Backtracking

תהליך ההוראה הקיים

- "פתיחה" של אי-דטרמיניזם (בדיקת כל המסלולים האפשריים)
- מטבע קסם

שאלות רקע

- באיזו גישה כדאי ללמד (שיקולים קוריקולריים):
האם לעקוף את המאפיינים האי-דטרמיניסטיים,
או להתמודד עימם?
- האם סטודנטים יכולים להפנים מודל לא
דטרמיניסטי ומכונות לא עקביות?

המחקר (2)

- סקירה וניתוח של ההתפתחות ההיסטורית של המושג אי-דטרמיניזם במדעי המחשב, ושל הדרך שבה הוא מוצג בספרי לימוד מובילים

(The concept of nondeterminism: its development and implications for teaching, Armoni & Ben-Ari, 2008)

מהמלצות המאמר

- הוראה מפורשת של אי-דטרמיניזם יכולה להוביל לראייה רחבה של המושג, בהקשריו השונים, כרעיון יסודי וכללי במדעי המחשב.
- התמודדות מפורשת עם אי-דטרמיניזם מזמנת צורך בהפשטה (עוד רעיון יסודי).

אי-דטרמיניזם כרעיון יסודי במדעי המחשב

- תכנות מקבילי ומבוזר
- מערכות הפעלה
- תקשורת
- שפות תכנות א"ד
- מודלים חישוביים

אי-דטרמיניזם כמבנה תכנותי יזום

(guarded command, Dijkstra, 1975)

```
if  
{  
  x >= y : max := x  
  y >= x : max := y  
}
```

- $X = 5, y = 2$
- $X = 3, y = 6$
- $X = 4, y = 4$

● העשרת כוח הביטוי

אי-דטרמיניזם כרעיון יסודי במדעי המחשב

התמודדות
עם מצב קיים

● תכנות מקבילי ומבוזר

● מערכות הפעלה

● תקשורת

הגדרה מתוך
בחירה

● שפות תכנות א"ד

● מודלים חישוביים

אי-דטרמיניזם כרעיון יסודי במדעי המחשב

אי-דטרמיניזם הוא

אי-דטרמיניזם הוא

אי-דטרמיניזם (דייקסטרא)

● יש הבדלים (מטרה, הגדרת קבלה)

אבל

● יש מאפיין משותף משמעותי וחשוב: בחירה

חופשית

הפשטה

- התעלמות מפרטים
- הכללה

מוכרת מזמן כרעיון יסודי מרכזי של
מדעי המחשב

אי-דטרמיניזם והפשטה

- להתרכז במה שחשוב (max)
- "מה" ולא "איך"
- התעלמות מפרטי ביצוע

מאפיינים של רדוקציה

- להתרכז במה שחשוב
- "מה" ולא "איך"
- התעלמות מפרטי ביצוע

גם רדוקציה היא ביטוי של הפשטה

הוראה מפורשת של אי-דטרמיניזם

● לעבוד עם אוטומטים לא דטרמיניסטיים כ"מכונות עם בחירה"

◆ לא צפויות

◆ לא עקביות

◆ מגדירות שפה ולא מתפקדות ככלי הכרעה

◆ החלטות נלקחות ברמה גבוהה מרמת הביצוע

השלבים

- ניתוח תהליך ההוראה
- תכנון התערבות
- בדיקת השפעת ההתערבות – small-scale

תהליך ההוראה הקיים באותה מכללה

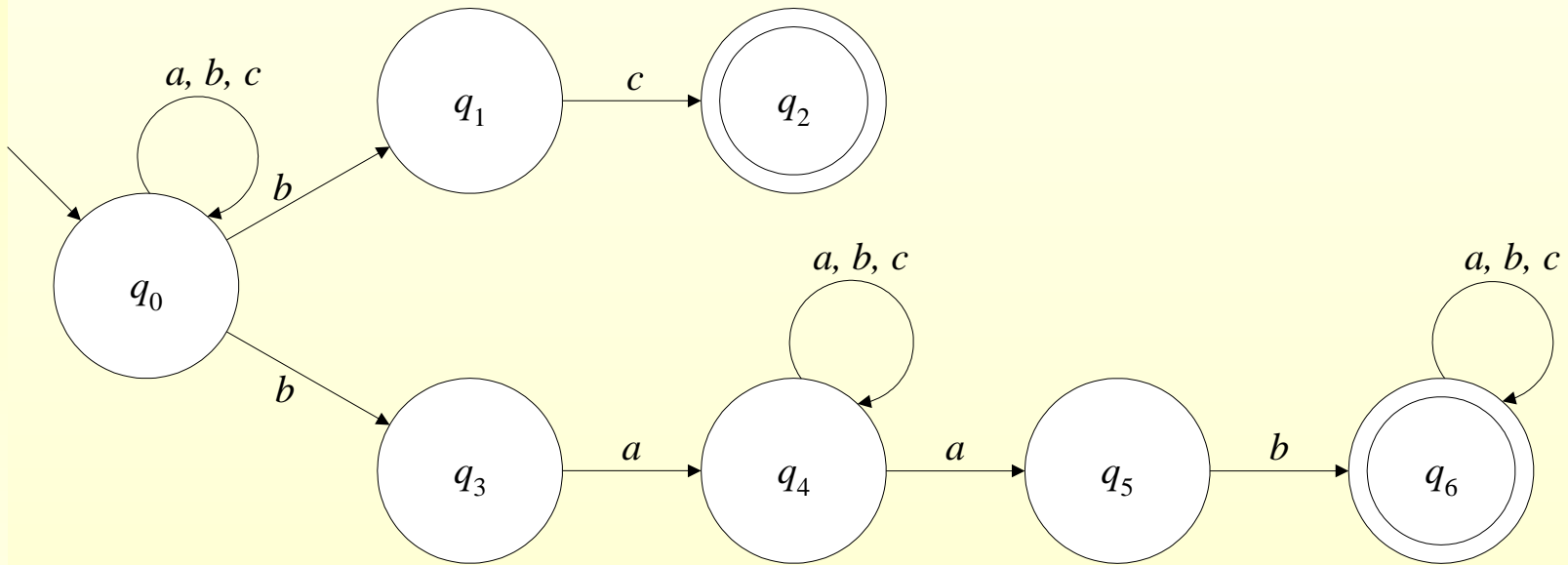
- אוטומטים לא דטרמיניסטיים כיישויות אבסטרקטיות, אמצעי מתמטי להגדרת שפות רגולריות.
- אף פעם לא הוזכרו מקביליות או backtracking או כל שיטת הרצה אחרת
- ניחושים הוזכרו בכיתה לעיתים קרובות

אז למה בכל זאת?

השערה:

בדוגמאות הראשונות הוצגו כל המסלולים
האפשריים, בסדר סטנדרטי (שיכול להזכיר עץ
חישוב)

דוגמה



מילת קלט: baabbc

?

התערבות (מינימלית)

- שינוי אופן הצגת הדוגמאות
 - ◆ רק חלק ממסלולי החישוב האפשריים
 - ◆ באופן בלתי תלוי (תמיד מהמצב ההתחלתי)
 - ◆ בסדר אקראי

בדיקת השפעת ההתערבות

ניסוי: מכללה א'

ביקורת: מכללה ב'

- שתי מרצות מנוסות
- קורס חובה שנה ג'
- מכללות פרטיות
- סילבוס דומה
- 4 שעות הוראה שבועיות (הרצאות ותרגולים משולבים לפי הצורך)

תחילת הסמסטר

- שאלון ידע מקדים –

חשיפה דומה לאי-דטרמיניזם בשתי הקבוצות

סיום הסמסטר

- שאלון שני –

image of nondeterministic automata

- ראיונות (שבעה סטודנטים ממכללה א')

השאלון השני

התייחס למשפטים הבאים. לכל משפט ציין האם אתה מסכים איתו או לא. הסבר בקצרה.

א. בכל פעם שאוטומט לא דטרמיניסטי מסוים A מופעל על אותו מילת קלט w הוא ייתן אותה תשובה.

ב. בכל פעם שאוטומט לא דטרמיניסטי A מופעל על מילת קלט $w \in L(A)$ הוא ייתן תשובה חיובית.

ג. בכל פעם שאוטומט לא דטרמיניסטי A מופעל על מילת קלט $w \notin L(A)$ הוא ייתן תשובה שלילית.

ד. כאשר אוטומט לא דטרמיניסטי A מופעל על מילת קלט $w \in L(A)$ ייתכן שייתן תשובה שלילית.

ה. כאשר אוטומט לא דטרמיניסטי A מופעל על מילת קלט $w \notin L(A)$ ייתכן שייתן תשובה חיובית.

ו. כאשר אוטומט לא דטרמיניסטי A מופעל על מילת קלט w אפשר לסמוך על תשובתו כדי לדעת אם w שייכת ל- $L(A)$.

ז. כדי לקבוע אם מילת קלט w שייכת לשפה $L(A)$ של אוטומט לא דטרמיניסטי A מספיק להפעיל את A על w פעם אחת.

ח. כדי לקבוע אם מילת קלט w שייכת לשפה $L(A)$ של אוטומט לא דטרמיניסטי מספיק להפעיל את A על w כמה פעמים.

ט. אוטומט לא דטרמיניסטי הוא מכונה שאפשר להשתמש בה כדי לקבוע שייכות של מילת קלט לשפה פורמלית.

- סמן את המשפט שאיתו אתה מסכים. הוסף הסבר קצר:
כאשר אוטומט לא דטרמיניסטי A מופעל על מילת קלט w :
- א. הוא מבצע את כל מסלולי החישוב האפשריים.
 - ב. מבצע מסלול חישוב כלשהו
 - ג. מבצע מסלול חישוב אחד ואם $w \in L(A)$ אז זה בהכרח מסלול מקבל.
 - ד. אחרת:
-

תוצאות: ציון אי-עקביות ממוצע

ציונים חיוביים: דימוי של ישות לא עקבית
ציונים שליליים: דימוי של ישות עקבית

- קבוצת הניסוי (מכללה א'): 3.29
- קבוצת הביקורת (מכללה ב'): -1.95

עקביות של הדימוי

| דימוי מעורב ולא עקבי | דימוי עקבי של מכונות לא-עקביות | דימוי עקבי של מכונות עקביות | |
|----------------------|--------------------------------|-----------------------------|----------|
| 13 | 6 | 1 | מכללה א' |
| 6 | 3 | 12 | מכללה ב' |

- הסטודנטים ממכללה ב' (ביקורת): יותר בטוחים (בדימוי של מכונות עקביות)
- הסטודנטים ממכללה א' (ניסוי): יותר מבולבלים.

ראיונות בסיום הסמסטר

- כל שבעת המרואיינים ראו אוטומט אי-דטרמיניסטי כישות עם היבט של חוסר עקביות
- אף אחד משבעת המרואיינים לא חשב על אוטומט אי-דטרמיניסטי כמכונה שבודקת את כל האפשרויות
- דימוי לא עקבי – אצל סטודנט אחד

ציטוטים

- "יכול להיות שאני אגיע למצב לא מקבל, אבל גם יכול לקרות שאגיע למצב מקבל [על אותה מילה]."
- "דברים שיכולים להתקבל לא יתקבלו אם אני לא בוחר את המסלול הנכון."
- "האחריות של האוטומט הוא רק לבצע את המעברים שלו נכון. הוא לא צריך, הוא לא צריך לקבל החלטה." – רמה גבוהה מרמת הביצוע
- "לפעמים הוא אומר "כן", לפעמים הוא אומר "לא" ... אין לי הרבה מה לעשות עם המערכת הזאת."

דיון

- יותר סטודנטים ממכללה ב' הפגינו דימוי של מכונות עקביות
 - במכללה א': תזוזה לכיוון דימוי של מכונות לא עקביות, שהביאה לדימויים עקביים של מכונות לא עקביות אבל גם לדימויים מעורבים ולא עקביים
- האם זה השינוי שלו קיווינו?

דיון

- הכיוון רצוי.
- ההתערבות היתה מינימלית ולא עימתה באופן מפורש את הסטודנטים עם האופי הלא עקבי של האוטומטים
- ייתכן כי התערבות פחות מינורית, עם עימות מפורש, יכולה להגדיל את עוצמת השינוי, באותו כיוון.
- הוראה ספירלית של אי-דטרמיניזם, בהקשרים שונים, בהתאם להיותו רעיון יסודי של מדעי המחשב.

הפשטה

מוכרת מזמן כרעיון יסודי מרכזי של
מדעי המחשב

● התעלמות מפרטים

● הכללה

הפשטה: הכללה

- מציאת פתרונות לבעיות
- ◆ פתרון אחד לכל המקרים
- תרגום של "פתרון" לצורה כללית
- ◆ כפל ארוך
- תכנות

◆ General statements משפטים כלליים

- $\text{number} \leftarrow \text{read}(x)$
- $\text{while index} < \text{lim do index} \leftarrow \text{index} + 1$

הפשטה: הכללה

- מציאת פתרונות לבעיות
 - ◆ פתרון אחד לכל המקרים
- תרגום של "פתרון" לצורה כללית
 - ◆ כפל ארוך
- תכנות
 - ◆ משפטים כלליים
 - ◆ מבנים כלליים
- $\langle \text{variable} \rangle \leftarrow \langle \text{value} \rangle$ $\text{number} \leftarrow \text{read}(x)$
- $\text{if } \langle \text{boolExp} \rangle \text{ then } \langle \text{statement1} \rangle \text{ else } \langle \text{statement2} \rangle$

הפשטה

מוכרת מזמן כרעיון יסודי מרכזי של
מדעי המחשב

● התעלמות מפרטים

● הכללה

● תנועה בין רמות:

◆ רמות של גודל

◆ רמות של משמעות

רמות של גודל

● שפות מכונה

◆ פעולות מאוד מפורטות ופרימיטיביות

◆ על ביטים

◆ שמכילים 0 או 1

רמות של גודל

- שפות מכונה

- ◆ פעולות מאוד מפורטות ופרימיטיביות

- ◆ על ביטים

- ◆ שמכילים 0 או 1

- שפות סף

- ◆ הוראות מפורטות מרמה נמוכה

- ◆ על בתים ומילים

- ◆ שמכילים ערכים הקסה-דצימליים

רמות של גודל

- שפות מכונה

- ◆ פעולות מאוד מפורטות ופרימיטיביות

- ◆ על ביטים

- ◆ שמכילים 0 או 1

- שפות סף

- ◆ הוראות מפורטות מרמה נמוכה

- ◆ על בתים ומילים

- ◆ שמכילים ערכים הקסה-דצימליים

- שפות עיליות

- ◆ משפטים שדומים לשפה טבעית

- ◆ על ישויות נתונים מופשטות

ככל שהרמה גבוהה יותר, כך הפעולות וישויות הנתונים
עליהן פועלים מופשטים יותר

רמות של משמעות

- "[T]his oscillation between “the representation” and “what it stands for” is an intrinsic part of the programmer’s game” (Dijkstra, 1975)
- “[C]omputer scientists ... must create abstractions of real-world problems that can be represented and manipulated inside a computer“ (Aho & Ullman, 1992)



רמות של משמעות

- "[T]his oscillation between “**the representation**” and “**what it stands for**” is an intrinsic part of the programmer’s game” (Dijkstra, 1975)
- “[C]omputer scientists ... must create abstractions of real-world problems that can be represented and manipulated inside a computer“ (Aho & Ullman, 1992)



רמות של משמעות

הבעיה: מיון רשימת מספרים

bubbleSort (A: list of sortable items)

פתרון:

```
n = length(A)
```

```
repeat
```

```
  swapped = false
```

```
  for i = 1 to n-1 do
```

```
    if A[i-1] > A[i] then
```

```
      swap(A[i-1], A[i])
```

```
      swapped = true
```

```
until not swapped
```


רמות של משמעות

הבעיה: מציאת החציון

פתרון:

findMedian (A: a list of sortable items)

n = length (A)

sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$

רמות של משמעות

הבעיה: מציאת החציון

פתרון:

findMedian (A: a list of sorted items)

n = length (A)

sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$

bubbleSort (A: list of sortable items)

n = length(A)

repeat

swapped = false

for i = 1 to n-1 do

if $A[i-1] > A[i]$ then

swap(A[i-1], A[i])

swapped = true

until not swapped

רמות של משמעות

findMedian (A: a list of sorted items)

n = length(A)

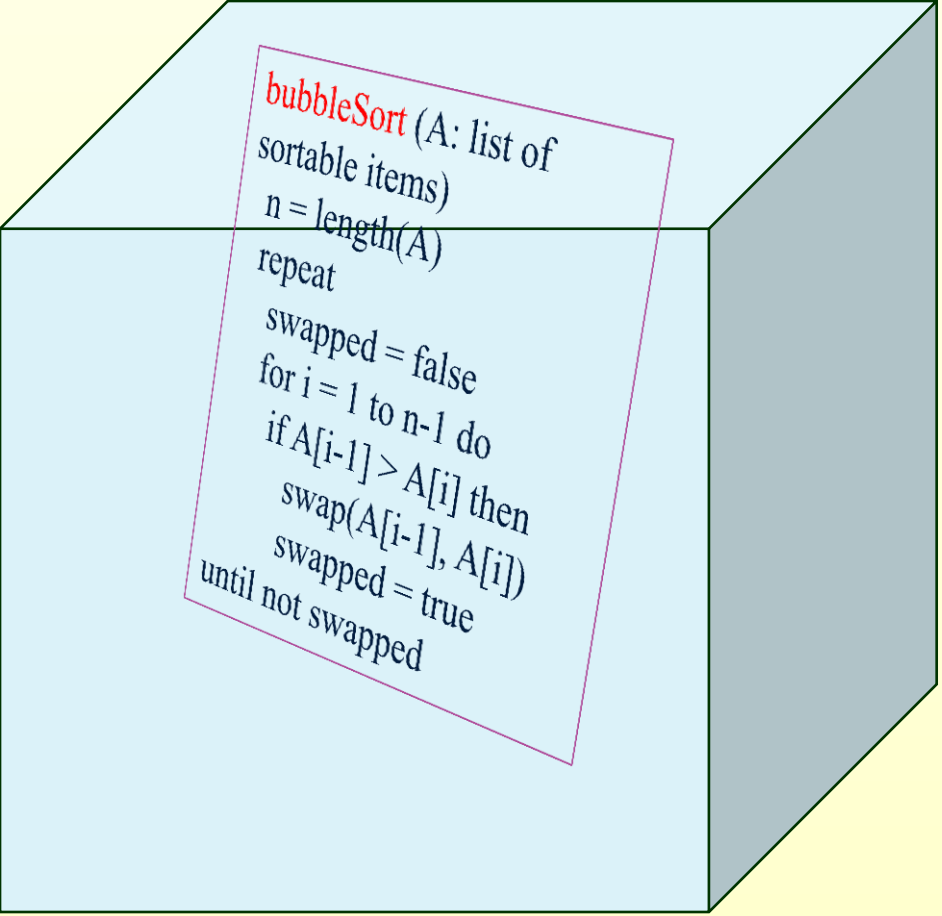
sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$



```
bubbleSort (A: list of
sortable items)
n = length(A)
repeat
  swapped = false
  for i = 1 to n-1 do
    if A[i-1] > A[i] then
      swap(A[i-1], A[i])
      swapped = true
  until not swapped
```

רמות של משמעות

findMedian (A: a list of sorted items)

n = length (A)

sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$



Sort a list of
sortable items

רמות של משמעות

mergeSort (A: list of sortable items)

n = length of A

if $n \leq 1$ then

return A

left = left half of A

right = right half of A

left = mergeSort (left)

right = mergeSort (right)

return merge (left, right)

Sort the list A

if n is odd

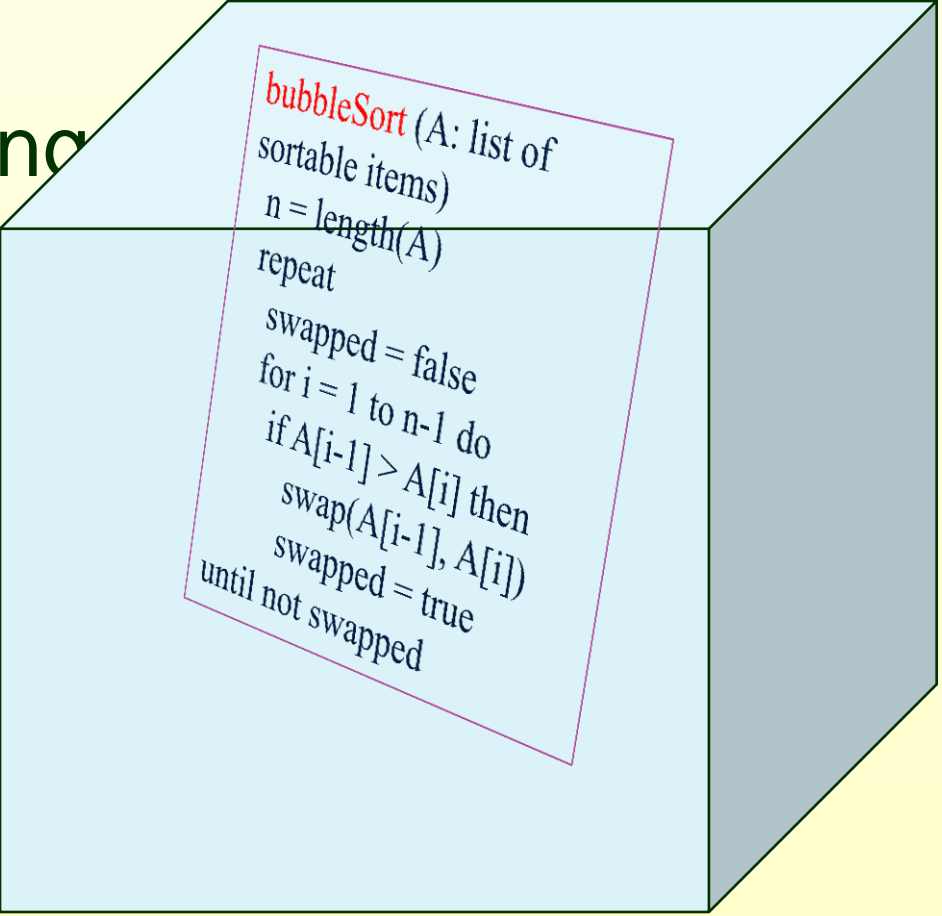
return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$

Finding

of sort



```
bubbleSort (A: list of
sortable items)
n = length(A)
repeat
  swapped = false
  for i = 1 to n-1 do
    if  $A[i-1] > A[i]$  then
      swap( $A[i-1]$ ,  $A[i]$ )
      swapped = true
  until not swapped
```

רמות של משמעות

findMedian (A: a list of sorted items)

n = length (A)

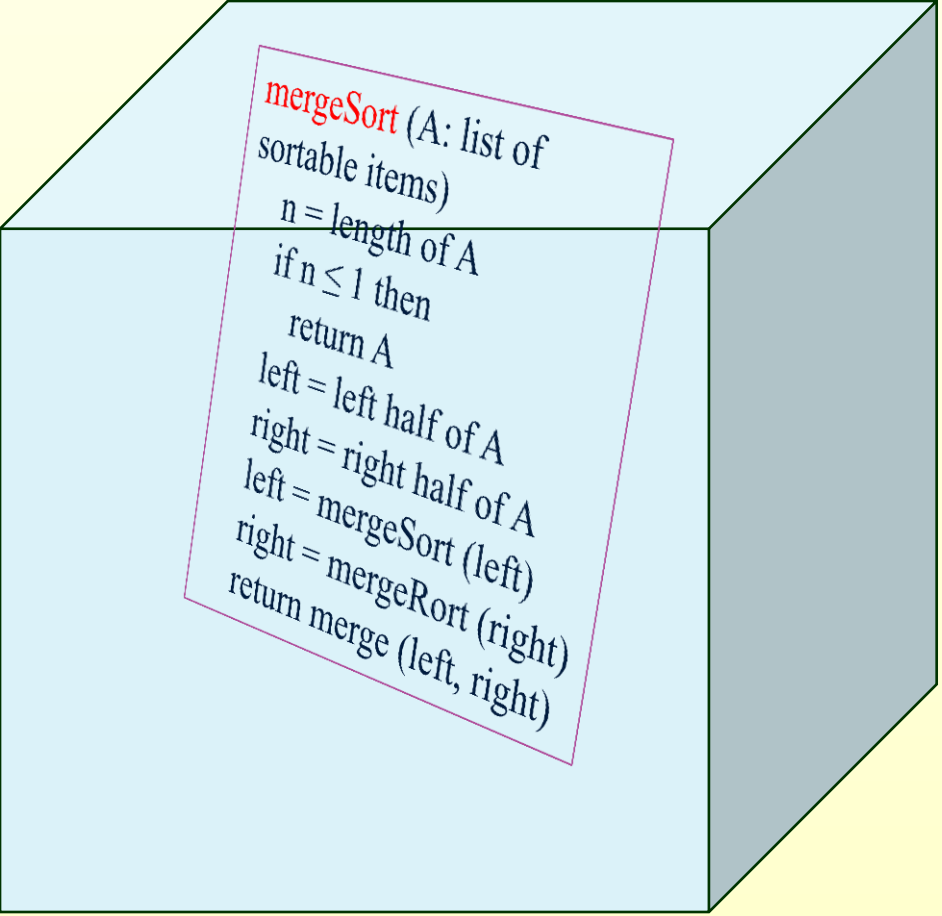
sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$



```
mergeSort (A: list of
sortable items)
n = length of A
if n ≤ 1 then
return A
left = left half of A
right = right half of A
left = mergeSort (left)
right = mergeSort (right)
return merge (left, right)
```

רמות של משמעות

findMedian (A: a list of sorted items)

n = length (A)

sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$



Sort a list of
sortable items

רמות של משמעות

הבעיה: מציאת החציון

פתרון:

findMedian (A: a list of sortable items)

n = length (A)

sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$

רמות של משמעות

הבעיה: מציאת החציון

findMedian (A: a list of sortable items)

n = length(A)

sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$

bubbleSort (A: list of sortable items)

n = length(A)

repeat

swapped = false

for i = 1 to n-1 do

if $A[i-1] > A[i]$ then

swap(A[i-1], A[i])

swapped = true

until not swapped

רמות של משמעות

findMedian (A: a list of sorted items)

n = length (A)

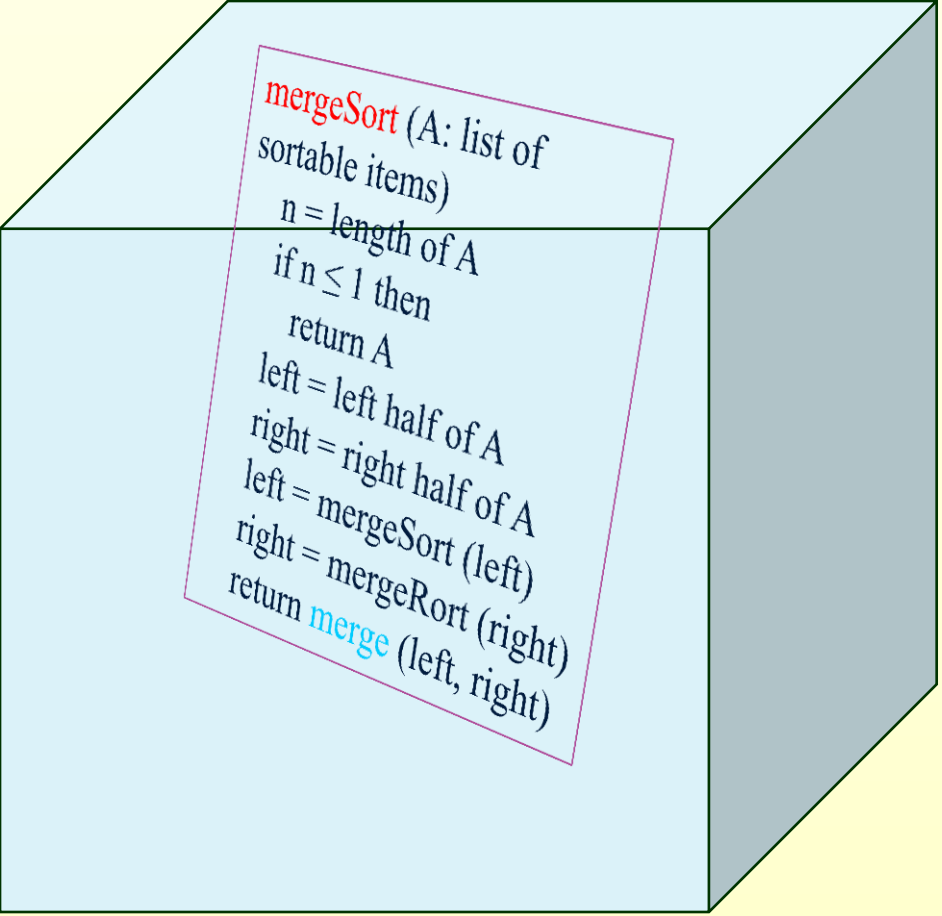
sort the list A

if n is odd

return $A[(n+1)/2]$

if n is even

return the average of $A[n/2]$ and $A[n/2+1]$



```
mergeSort (A: list of
sortable items)
n = length of A
if n ≤ 1 then
return A
left = left half of A
right = right half of A
left = mergeSort (left)
right = mergeSort (right)
return merge (left, right)
```

רמות של משמעות

- שפות מכונה
- שפות סף
- שפות עיליות

points ← points + 3

רמות של משמעות

- שפות מכונה

- שפות סף

mov (points, points1)

add (points, points2)

- שפות עיליות

points \leftarrow points1 + points2

רמות אבסטרקציה

לנוע באופן חופשי ומודע בין רמות אבסטרקציה

- "[T]his **oscillation** between “the representation” and “what it stands for” is an intrinsic part of the programmer’s game” (Dijkstra, 1975)
- Thinking like a computer scientist [...] requires **thinking in multiple levels of abstraction.**“ (Wing, 2006)
- "they [natural computer scientists] are individuals who can **rapidly change levels of abstraction, simultaneously** seeing things “in the large” and “in the small”" (Knuth, in Hartmanis, 1994)