

**מגישים:**

יונדב קזז 207068973

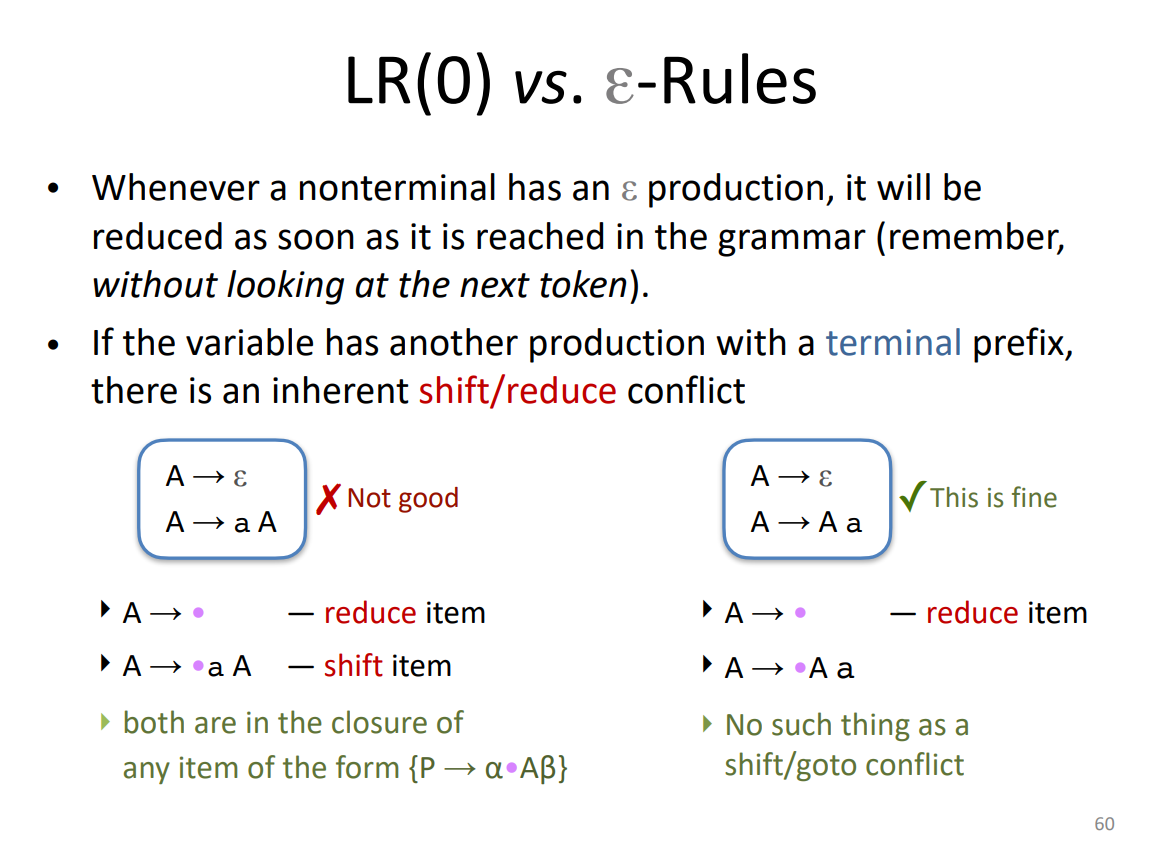
לירן כהן 209043470

**שאלה 1 – Parsing**

1. הדקדוק אינו LL(1), יש קונפליקט:

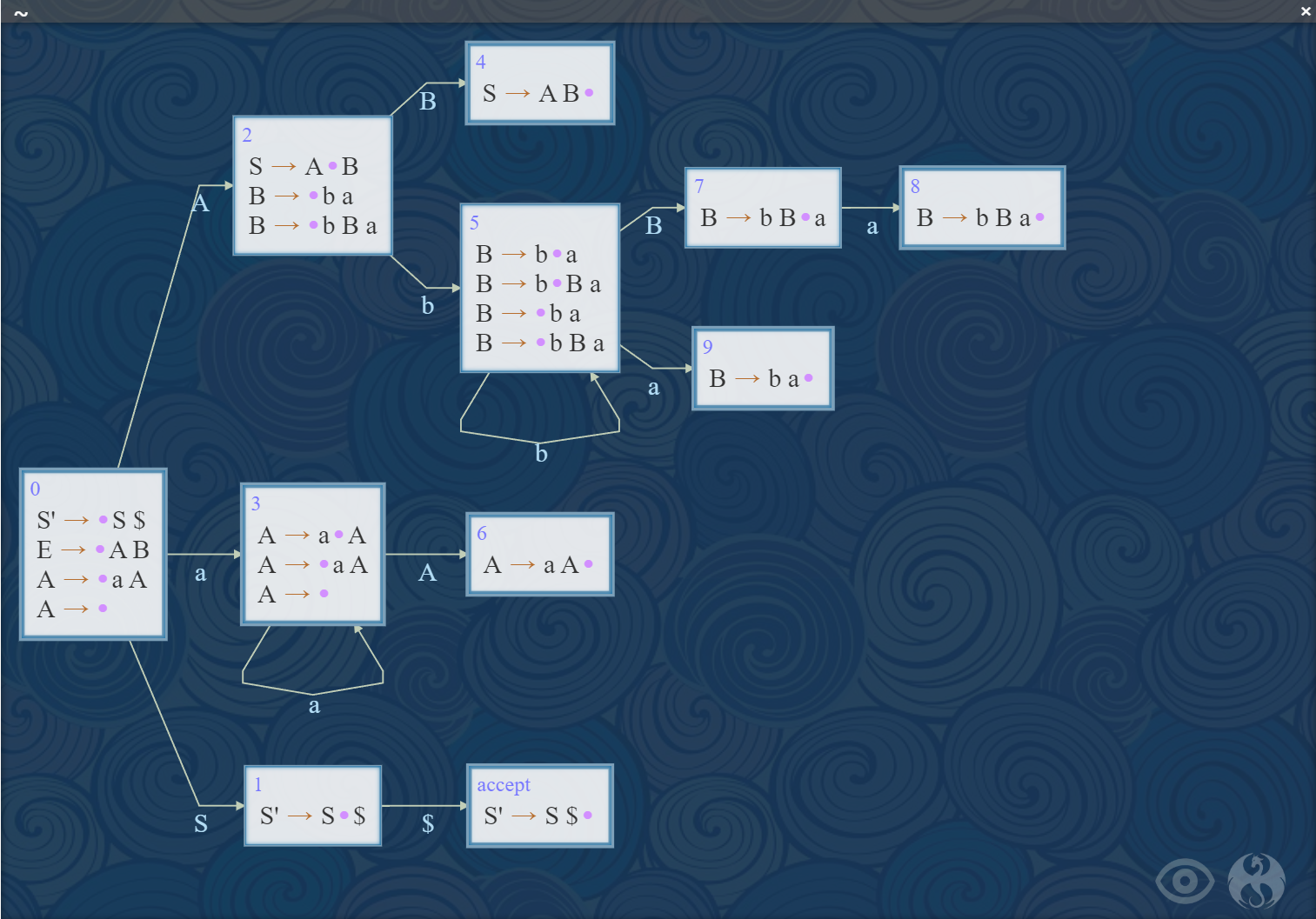
|  |  |  |  |
| --- | --- | --- | --- |
|  | a | b | $ |
| S |  |  |  |
| A |  |  |  |
| B |  |  |  |

1. הדקדוק אינו LR(0), יש קונפליקט shift-reduce מכלל האפסילון, כפי שנלמד בהרצאה.  
   בכל היתקלות עם המשתנה A, יהיה קונפליקט האם לעשות reduce עבור כלל האפסילון, או shift עבור a:



1. הדקדוק SLR. נראה שאין קונפליקטים ע"י בניית האוטומט והטבלה.  
   נמספר את הכללים:

אוטומט:



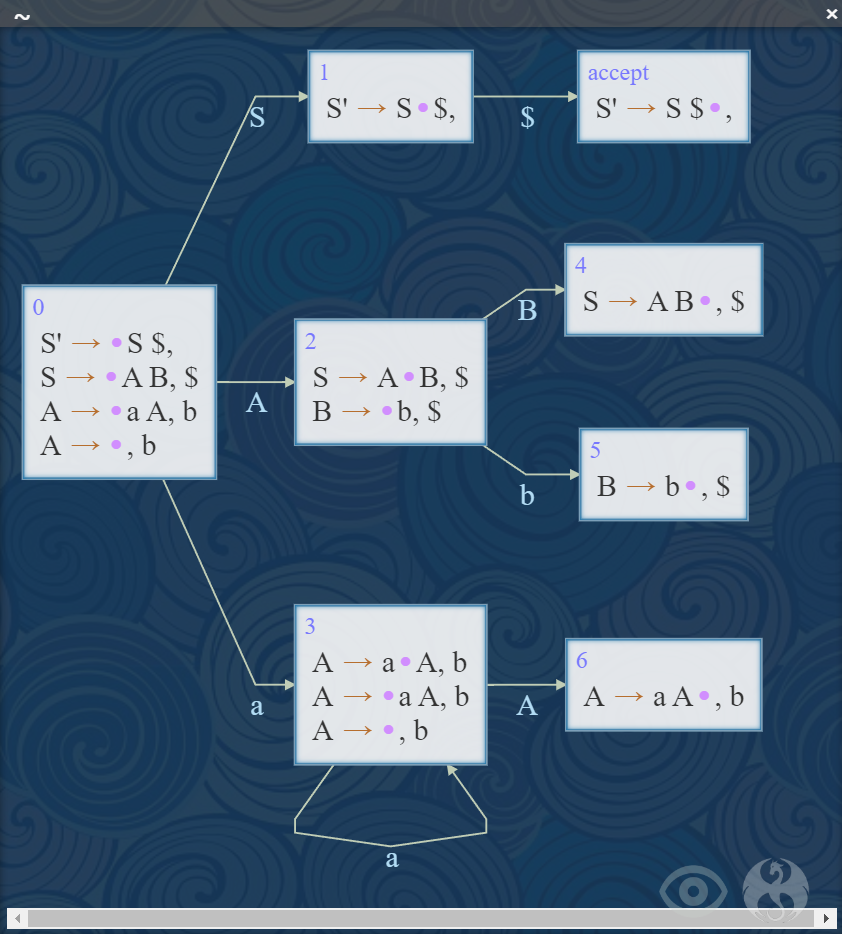
טבלת מצבים:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SLR actions | | | | goto | | |
|  | a | b | $ | S | A | B |
| 0 | s3 | r3 |  | 1 | 2 |  |
| 1 |  |  | acc |  |  |  |
| 2 |  | s5 |  |  |  | 4 |
| 3 | s3 | r3 |  |  | 6 |  |
| 4 |  |  | r1 |  |  |  |
| 5 | s9 | s5 |  |  |  | 7 |
| 6 |  | r2 |  |  |  |  |
| 7 | s8 |  |  |  |  |  |
| 8 | r5 |  | r5 |  |  |  |
| 9 | r4 |  | r4 |  |  |  |

אין קונפליקטים לכן הדקדוק SLR.

1. כפי שלמדנו בכיתה, כל דקדוקי SLR מוכלים בכל דקדוקי LR(1). בשילוב עם הסעיף הקודם נקבל כי הדקדוק הוא LR(1).
2. נמספר את הכללים:

אוטומט:



טבלת מצבים:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| LR(1) actions | | | | goto | | |
|  | a | b | $ | S | A | B |
| 0 | s3 | r3 |  | 1 | 2 |  |
| 1 |  |  | acc |  |  |  |
| 2 |  | s5 |  |  |  | 4 |
| 3 | s3 | r3 |  |  | 6 |  |
| 4 |  |  | r1 |  |  |  |
| 5 |  |  | r4 |  |  |  |
| 6 |  | r2 |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| מחסנית | פעולה | קלט |
| (0,) | s3 | ab$ |
| (0,)(3,a) | r3 | b$ |
| (0,)(3,a)(goto(3,A),A) |  |  |
| (0,)(3,a)(6,A) | r2 | b$ |
| (0,)(goto(0,A),A) |  |  |
| (0,)(2,A) | s5 | b$ |
| (0,)(2,A)(5,b) | r4 | $ |
| (0,)(2,A)(goto(2,B)) |  |  |
| (0,)(2,A)(4,B) | r1 | $ |
| (0,)(goto(0,S)) |  |  |
| (0,)(1,S)) | acc | $ |

|  |  |  |
| --- | --- | --- |
| מחסנית | פעולה | קלט |
| (0,) | r3 | bb$ |
| (0,)(goto(0,A)) |  |  |
| (0,)(2,A) | s5 | bb$ |
| (0,)(2,A)(5,b) | error | b$ |

**שאלה 2 – DFA**

1. נגדיר:

|  |  |  |
| --- | --- | --- |
| Statement | Kill(B) | Gen(B) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. *נגדיר מעברים בעזרת* טבלת Kill/Gen. נגדיר בנוסף את הפעולות והקבוצות הבאות:

*נעדיף להשתמש באפשרות השנייה של ה-*replace *כי כך נדע ש-*c *לא יכול להיות ב-*x*.   
עבור תתי-חישובים בתוך בלוק נניח לשם הפשטות שהתוצאה שלהם נמצאת ב-*in *של הבלוק הזה.*

1. *נמספר את הבלוקים לפי מספרי השורות בהם. (לדוגמה, שורות 2-3 הן בלוק 1). מתקיים:*

*(since x is unknown, it can have every character)*

*נריץ את האנליזה:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. *הרעיון של סת' לא מוכיח את ה-*assert *בשורה 7, כיוון שלפי הגדרתו הוא רק מראה ש-*s***יכול*** *להיות בתוך* y*, אבל לא מראה ש-*s***חייב*** *להיות בתוך* y *תמיד.*
2. *כעת נגדיר:*
3. הסמנטיקה תשתנה עבור פעולות ה-replace והכפל, כי עכשיו צריכים לעקוב אחר התווים שבוודאות יהיו לאחר ההחלפה, ולא תווים שאולי יתווספו.  
   עבור הפעולות האחרות הסמנטיקה תהיה זהה. השינויים יהיו:

|  |  |  |
| --- | --- | --- |
| Statement | Kill(B) | Gen(B) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

נעדיף להשתמש באפשרות הראשונה של replace כי כך נדע שכל התווים של y חייבים להיות ב-x.

1. כעת פונקציית המעברים תהיה:

*(since x is unknown, there's no character it must have)*

נריץ את האנליזה:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

כעת ניתן להוכיח את ה-assert בשורה 7 – אנחנו יודעים שבכניסה לבלוק 4, התו 's' חייב להיות  
בתוך המחרוזת y ולכן ה-find לא יחזיר -1.

1. נרצה להשתמש באנליזה של סת' – כיוון שהיא מראה את כל התווים שעלולים להיות בכל מחרוזת (אנליזת may), כל תו שלא יימצא באנליזה עבור מחרוזת כלשהי מובטח לא להיות במחרוזת זו. כלומר, כשנריץ את האנליזה של סת' על bar נוכל לבדוק האם ב-in של הבלוק האחרון (שורה 17) מתקיים ש-(y,'s') שייך לו. אם לא, נדע שאכן אין את האות s במחרוזת y בשלב זה ונוכיח שה-assert מתקיים.