

אפריל 2022

תרגיל בית flex & bison ו-recursive descent parser סמסטר ב 2022

הגשה דרך moodle

אפשר להגיש בזוגות

מה צריך להגיש: קובץ zip הכולל את הקוד שאתם כתבתם ודוגמא לקלט עם הפלט המתאים.

יש לכתוב שתי גרסאות של תוכנית שתאורה יובא כאן. בגרסה אחת יש להשתמש ב-bison כדי לייצר את ה-parser. בגרסה השנייה יש לכתוב באופן ידני recursive descent parser. בשתי הגרסאות ניתן להשתמש ב-flex כדי לייצר את ה-lexer.

את הקבצים בגרסה עם ה-bison יש להגיש בתוך תיקיה בשם bottomUp את הקבצים בגרסה עם ה-recursive descent parser יש להגיש בתוך תיקיה בשם topDown (את שתי התיקיות יש לארוז בקובץ zip אחד). הגישו גם קובץ README עם הפרטים של שני המגישים (אפשר להגיש גם לבד). אם מגישים בזוג אז רק אחד או אחת מהזוג יגיש (עם הפרטים של שני המגישים).

התוכנית שתכתבו תקרא קלט הכולל מידע על נצחונות של שחקני טניס בטורנירים שונים. הקלט יופיע בקובץ ששמו יתן כ-command line argument.

היא תדפיס (ל-standard output) את שם השחקנית (להבדיל משחקן) שזכתה במספר הגבוה ביותר של טורנירים בווימבלדון.

התוכנות של flex & bison נמצאות ב-moodle בתיקיה על bison (אלו גרסאות שמיועדות ל-Windows. למי שעובד על Linux: אם תנסו לתת פקודה bison ל-shell ו-bison לא מותקן אצלכם -- סביר שתקבלו באופן אוטומטי הנחיות התקנה).

תאור הקלט

הקלט דומה למה שראינו בתרגיל הבית הראשון.

הקלט כולל רשימה של שחקני טניס. עבור כל שחקן או שחקנית מופיעים הנתונים הבאים: שם השחקן או השחקנית, המגדר (גבר או אשה), השנים בהן זכה בטורניר ווימבלדון, השנים בהם זכה באליפות אוסטרליה הפתוחה והשנים בהם זכה באליפות צרפת הפתוחה. שלושת הפרטים האחרונים הם אופציונליים. למשל בדוגמא שתובא בהמשך, עבור Simona Halep חסר החלק של אליפות אוסטרליה הפתוחה (שבה מעולם לא זכתה).

שנות הזכייה מופיעים כרשימה של שנים מופרדים ע"י פסיקים. בתור קיצור עשוי להופיע גם טווח של שנים למשל 1982-1987

זה קיצור של 1982, 1983, 1984, 1985, 1986, 1987

בקלט יכול להופיע גם whitespace (רווחים, טאבים ו- newlines).

הנה דוגמא לקלט:

**** Winners ****

```
<name> "Martina Navratilova" <Woman>
  <Wimbledon> 1978, 1979, 1982-1987, 1990
  <Australian Open> 1981,1983,1985
  <French Open> 1982, 1984

<name> 'Roger Federer' <Man>
  <Wimbledon> 2003-2007, 2009, 2012, 2017
  <Australian Open> 2004, 2006, 2007, 2010, 2017, 2018
  <French Open> 2009

<name> "Novak Djokovic" <Man>
  <Wimbledon> 2011, 2014, 2015, 2018, 2019, 2021
  <Australian Open> 2008, 2011-2013, 2015, 2016, 2019-
2021
  <French Open> 2016, 2021

<name> 'Simona Halep' <Woman>
  <Wimbledon> 2019
  <French Open> 2018
```

בדוגמא זו על התוכנית להדפיס:

Woman with greatest number of wins at Wimbledon: Martina
Navratilova (9 wins)

דקדוק לתאור הקלט (בפורמט של bison)

בהתאם למוסכמה של bison -- אסימונים כתובים כאן באותיות גדולות,
ומשתנים כתובים באותיות קטנות. (גם תווים המוקפים בגרש בכל צד כמו
למשל ' - ' הם אסימונים).
שימו לב שבסוף כל כלל גזירה (או מספר כללי גזירה המופרדים ע"י |) מופיע
נקודה פסיק בהתאם לפורמט של bison.

```
input: TITLE list_of_players;

list_of_players: list_of_players player;

list_of_players: player;

player: NAME PLAYER_NAME GENDER
      optional_wimbledon optional_australian_open
```

```

optional_french_open;

optional_wimbledon: WIMBLEDON list_of_years |
    %empty;
optional_australian_open: AUSTRALIAN_OPEN list_of_years |
    %empty;

optional_french_open: FRENCH_OPEN list of years |
    %empty;

list_of_years: list_of_years ',' year_spec;
list_of_years: year_spec;

year_spec: YEAR | YEAR '-' YEAR ;

```

הערות

עליכם להחליט באיזה ערכים סמנטיים להשתמש. אין להשתמש במשתנים גלובליים (כדי לתרגל את השמוש בערכים סמנטיים).

בתרגיל הראשון הקלט היה כמו כאן ושם התבקשתם לקבוע לבד מה הם סוגי האסימונים. בתרגיל הזה יש להשתמש באסימונים שמופיעים בדקדוק הנ"ל ולא באסימונים שהגדרתם בתרגיל הראשון.

תזכורת: הכנת תוכנית בעזרת flex & bison (ההערות מתייחסות ל-Windows ול-Linux)

נניח שברשותנו קובצי קלט ל-flex ול-bison שהכנו בעזרת text editor (למשל Notepad++). נקרא לקבצים tennis.lex ו-tennis.y.

נריץ את הפקודות הבאות בחלון המריץ command prompt (או shell אם עובדים על Linux).

```

flex tennis.lex
1. מריצים את flex

```

```

bison -d tennis.y
2. מריצים את bison עם האופציה -d
bison יצור שני קבצים: tennis.tab.c ו-tennis.tab.h (את השני הוא יצור בגלל האופציה -d).

```

הערה: אין חשיבות לסדר שבו מבצעים את שני הצעדים הראשונים כלומר ניתן להריץ קודם את bison.

הערה: אם עובדים על Windows הקפידו להעתיק את התיקיה data לתיקיה שבה אתם עובדים (התיקיה data נמצאת ב-moodle ביחד עם קובץ ההרצה של bison).

3. יש לקמפל את קובצי ה-C ש-flex & bison יצרו עבורנו. (כמובן שאם התוכנית שלנו כוללת קבצים נוספים יש לקמפל גם אותם). לצורך כך ניתן להשתמש בכל קומפיילר לשפת C.

אם נשתמש בקומפיילר gcc (קומפיילר פופולרי של GNU) הפקודה היא:
gcc -o tennis.exe lex.yy.c tennis.tab.c

כאן האופציה -o מציינת את שם הקובץ שהוא התוצר של הקומפילציה (במקרה זה שם הקובץ הוא tennis.exe. אם עובדים על Linux לא צריך סיומת .exe).

4. נכין קובץ טקסט שנקרא לו tennis_input.txt ובו נכתוב קלט לדוגמא למשל

```
<name> "Martina Navratilova" <Woman>
  <Wimbledon> 1978, 1979, 1982-1987, 1990
  <Australian Open> 1981,1983,1985
  . . .
```

נריץ את הפקודה

```
tennis.exe tennis_input.txt
```

על Linux זה יראה כך:

```
./tennis tennis_input.txt
```

והפלט יהיה לדוגמא

```
Woman with greatest number of wins at Wimbledon: Martina
Navratilova (9 wins)
```

דוגמאות לתוכניות שהוכנו בעזרת flex & bison:

בתיקיה של bison באתר הקורס יש דוגמאות. בנוסף לכך אפשר להסתכל בפתרונות של בחינות (בתיקית הבחינות) מהשנים האחרונות. השאלה הראשונה בכל בחינה עוסקת ב-flex & bison.

הערות לגבי ה-recursive descent parser

ב-moodle יש דוגמא לתוכנית עם recursive descent parser (היא נמצאת בתיקיה על top down parsing). זו התוכנית שראינו בכיתה שעוסקת ברשימת שירים (ביניהם של מדונה).

הדקדוק שהובא למעלה (בפורמט של bison) אינו LL(1). אבל למרות זאת

לא קשה לכתוב עבורו recursive descent parser. גם בדוגמא עם מדונה שראינו,
הדקדוק המקורי לא היה LL(1).

בהצלחה!