

תורת הקומפילציה

תרגיל בית 2 – ניתוח תחבירי top down
מתרגל אחראי לתרגיל: עומר כץ

הגשה בזוגות.

יש להפנות שאלות על התרגיל במייל ל-omerkatz@cs. בלבד.

לתרגיל ייפתח דף FAQ באתר הקורס. כל הבהרה שתופיע בו עד יומיים ממועד ההגשה תהווה הוראה מחייבת.

הנחיות כלליות

- בתרגיל זה תממשו מנתח תחבירי עבור דקדוק המשתמש באסימונים אשר זיהיתם בתרגיל הקודם.
- התרגיל ייבדק אוטומטית. **הקפידו למלא אחר ההוראות במדויק.** הבדיקה תתבצע על csl2.
- קראו את כל התרגיל עד סופו בטרם תחילת העבודה על פתרון.

להלן דקדוק במחלקה LL(1) המגדיר אובייקטי yamll (חלקיים) בהתבסס על האסימונים מתרגיל 1:

1. $S \rightarrow Structure$
2. $S \rightarrow Collection$
3. $Structure \rightarrow STARTSTRUCT SItems ENDSTRUCT$
4. $SItems \rightarrow Collection SItems$
5. $SItems \rightarrow \epsilon$
6. $Collection \rightarrow List$
7. $Collection \rightarrow Map$
8. $List \rightarrow LLIST LItems RLIST$
9. $LItems \rightarrow Item COMMA LItems$
10. $LItems \rightarrow \epsilon$
11. $Map \rightarrow LDICT MItems RDICT$
12. $MItems \rightarrow Key KEY Item COMMA MItems$
13. $MItems \rightarrow \epsilon$
14. $Key \rightarrow VAL$
15. $Key \rightarrow COMPLEXKEY Collection$
16. $Item \rightarrow STRING$
17. $Item \rightarrow INTEGER$
18. $Item \rightarrow REAL$
19. $Item \rightarrow Collection$
20. $Item \rightarrow TRUE$
21. $Item \rightarrow FALSE$
22. $Item \rightarrow VAL$

כך ש-S הוא המשתנה ההתחלתי בדקדוק.

מבנה התרגיל

עם התרגיל נתונים לכם מספר קבצים:

- הקובץ tokens.h מכיל אנומרציה (enum) של שמות האסימונים מהתרגיל הקודם.

- הקבצים `grammar.h` ו-`grammar.cpp` מגדירים ומאתחלים את הדקדוק המופיע בתחילת התרגיל, ומכילים פונקציות הדפסה שישמשו אתכם כדי לייצר את הפלט הנדרש.
- הקובץ `hw2.h` מכיל את הצהרות הפונקציות שעליכן לממש כדי לענות על הסעיפים השונים בתרגיל.

אין צורך לממש פונקציית `main`. נתונה לכם פונקציית `main` בקובץ `main.cpp`.

קראו **היטב** את התיעוד של הפונקציות בקבצים השונים.

שלב מקדים – הכנת המנתח הלקסיקלי

עדכנו את הפתרון שלכם לתרגיל 1 כך שכל ההדפסות שביצעתם בזיהוי אסימונים יוחלפו בהחזרת ערך האנומרציה של האסימון המתאים (לשם כך תצטרכו לבצע `include` לקובץ `tokens.h`). שנו את שם קובץ ה-`lexer.lex` ל-`lexer.lex`.

להערות לא מוגדר ערך אינומרציה. זו אינה טעות. מכיוון שהערות הן חסרות משמעות בדקדוק, אין צורך להעביר אותן לניתוח התחבירי.

סעיף א' – זיהוי משתנים אפיסים

המשתנה `grammar` (שמוגדר בקובץ `grammar.h` ומאותחל בקובץ `grammar.cpp`) מכיל את הדקדוק שמופיע בתחילת התרגיל. השתמשו בו כדי לזהות איזה משתנים (המוגדרים באנומרציה `nonterminal` בקובץ `grammar.h`) הם אפיסים ואיזה לא.

עליכם לממש אלגוריתם `fixed-point` לזיהוי המשתנים האפיסים בדקדוק. האלגוריתם ימומש בגוף הפונקציה `compute_nullable` המוצהרת בקובץ `hw2.h`.

בתום חישוב המשתנים האפיסים יש להדפיסם באמצעות הפונקציה `print_nullable`.

על מנת לבדוק סעיף זה יש להריץ את קובץ ההרצה עם הארגומנט `-nullable`.

סעיף ב' – חישוב פונקציית `first`

בסעיף זה עליכם לממש אלגוריתם `fixed-point` לחישוב פונקציית `first` לכל אחד מהמשתנים בדקדוק על פי האלגוריתם המופיע בשקף 17 בתרגול 2. האלגוריתם ימומש בגוף הפונקציה `compute_first` המוצהרת בקובץ `hw2.h`.

בתום חישוב פונקציית `first` יש להדפיס את ה-`first` של כל המשתנים באמצעות הפונקציה `print_first`.

על מנת לבדוק סעיף זה יש להריץ את קובץ ההרצה עם הארגומנט `-first`.

סעיף ג' – חישוב פונקציית `follow`

בסעיף זה עליכם לממש אלגוריתם `fixed-point` לחישוב פונקציית `follow` לכל אחד מהמשתנים בדקדוק על פי האלגוריתם המופיע בשקף 24 בתרגול 2. האלגוריתם ימומש בגוף הפונקציה `compute_follow` המוצהרת בקובץ `hw2.h`.

בתום חישוב פונקציית `follow` יש להדפיס את ה-`follow` של כל המשתנים באמצעות הפונקציה `print_follow`.

על מנת לבדוק סעיף זה יש להריץ את קובץ ההרצה עם הארגומנט `-follow`.

סעיף ד' – חישוב פונקציית select

בסעיף זה עליכם לממש אלגוריתם לחישוב פונקציית select לכל אחד מכללי הדקדוק על פי ההגדרה המופיעה בשקף 16 בתרגול 2. האלגוריתם ימומש בגוף הפונקציה compute_select המוצהרת בקובץ hw2.h.

בתום חישוב פונקציית select יש להדפיס את ה-select של כל הכללים באמצעות הפונקציה print_select.

על מנת לבדוק סעיף זה יש להריץ את קובץ ההרצה עם הארגומנט select-.

סעיף ה' – מימוש מנתח LL(1)

בסעיף זה תממשו מנתח LL(1) לדקדוק הנתון בתחילת התרגיל, בהסתמך על הסעיפים הקודמים ועל פי האלגוריתם המופיע בשקף 37 בתרגול 2.

את הפתרון לסעיף זה יש לממש בגוף הפונקציה parser המוגדרת בקובץ hw2.h.

על מנת לבדוק סעיף זה יש להריץ את קובץ ההרצה ללא ארגומנטים.

אופן פעולת המנתח

המנתח יבנה את רצף כללי הגזירה הנחוץ לבניית המילה המופיעה בקובץ הקלט. המנתח יקבל כקלט מ-stdin רצף תווים העונה להגדרות מתרגיל 1 של האסימונים המופיעים בדקדוק. השתמשו ב-yylex() על מנת לקרוא את רצף האסימונים (הקריאה הראשונה ל-yylex תחזיר את האסימון הראשון בקלט).

במהלך הניתוח, בכל הפעלת כלל גזירה (כחלק מפעולת predict) יש להדפיס את מספר כלל הגזירה בו השתמשתם (בהתאם למספור המופיע בתחילת התרגיל). למשל אם הופעל כלל הגזירה $S \rightarrow Collection$, מספרו 2 ולכן יש להדפיס:

2\n

במידה והניתוח מסתיים בהצלחה יש להדפיס את ההודעה:

Success\n

במידה ומתקבלת שגיאת ניתוח יש להדפיס את ההודעה:

Syntax error\n

ולסיים את ריצת המנתח מיד.

נתונות לכם שתי דוגמאות קלט/פלט עבור סעיף זה.

המלצות מימוש

- השתמשו ב-`std::vector<int>` למימוש המחסנית Q.
- השתמשו ב-`std::map<nonterminal, std::map<tokens, int>>` למימוש הטבלה M.

הערות לגבי הדפסות

פונקציות ההדפסה עבור סעיפים א'-ד' מקבלות `std::vector` המכיל את הערכים הצפויים.

עבור סעיפים א'-ג' ערכי ה-vector יהיו הערכים עבור איברי האנומרציה nonterminal, לפי הסדר. כלומר במקום ה-0 הערך עבור S, ובאופן כללי הערך עבור נטרמינל i יופיע במקום ה-i.

עבור סעיף ד' ערכי ה-vector יהיו הערכים עבור כללי הדקדוק, לפי הסדר שהוגדר בתרגיל. כלומר במקום ה-0 הערך עבור הכלל הראשון ($S \rightarrow Structure$), ובאופן כללי הערך עבור הכלל i יופיע במקום ה-($i-1$).

הוראות הגשה

אין לשנות את ה-Makefile.

יש להגיש קובץ אחד בשם ID1-ID2.zip, עם מספרי ת"ז של שני המגישים. על הקובץ להכיל:

- קובץ lexer.lex
- קובץ hw2.h
- את כל הקבצים הנדרשים לבניית המנתח, כולל קבצי `cpp/h` שסופקו כחלק מהתרגיל.

בנוסף יש להקפיד שהקובץ **לא יכיל** את:

- קבצי ההרצה
- קובץ הפלט של flex
- קובץ Makefile שסופק כחלק מהתרגיל

יש לוודא כי בביצוע `unzip` לא נוצרת תיקיה נפרדת – כלומר כי קבצי הפתרון נמצאים בתיקיה בה בוצע `unzip`. על המנתח להיבנות על השרת `cs12` ללא שגיאות באמצעות קובץ Makefile שסופק עם התרגיל. יש לוודא כי פורמט הפלט זהה לפורמט הפלט של הדוגמאות הנתונות. כלומר, ביצוע הפקודות הבאות:

```
unzip id1-id2.zip
cp path-to/Makefile
make
./hw2 < path-to/t1.in >& t1.res
diff t1.res path-to/t1.out
```

ייצור את קובץ ההרצה בתיקיה הנוכחית ללא שגיאות קומפילציה, יריץ אותם, והרצת `diff` תחזיר 0. שימו לב כי במידה והמנתח שלכם לא עובר את כל קבצי הבדיקה שסופקו מראש, לא תתאפשר הגשה חוזרת של התרגיל.

שימו לב כי באתר מופיע script לבדיקה עצמית לפני ההגשה בשם `selfcheck`. תוכלו להשתמש בו על מנת לוודא כי ההגשה שלכם תקינה.

בהצלחה!

