

תורת הקומפילציה

תרגיל 2

מתרגל אחראי: שי גנדלמן

הגשה בזוגות

שאלות לגבי התרגיל יש לשאול בפיאצה, במידה והשאלה לא נשאלה בעבר. שאלות שנשאלו בעבר לא יענו.

תיקונים והערות חשובות יסומנו בצהוב בדף, הודעה תפורסם באתר הקורס במידה והקובץ יעודכן. כל עדכון שיתבצע עד יומיים לפני הגשת התרגיל מחייב.

התרגיל יבדק בבדיקה אוטומטית, הקפידו למלא אחר ההוראות במדויק.

כללי

בתרגיל זה עליכן לממש ניתוח תחבירי לשפת FanC, הכוללת פעולות אריתמטיות, פונקציות, מבני בקרה, והמרות מובנות מ-byte (בית אחד) ל-int (4 בתים).

מנתח זה ימשש אתכם גם בתרגילים הבאים ולכן המלצתנו היא שתממשו ותגישו תרגיל זה על אף שהוא מגן.

מנתח לקסיקלי

יש לכתוב מתנח לקסיקלי המתאים להגדרות ההבאות:

אסימון	תבנית
VOID	void
INT	int
BYTE	byte
B	b
BOOL	bool
AND	and
OR	or
NOT	not
TRUE	true
FALSE	false
RETURN	return
IF	if
ELSE	else
WHILE	while
BREAK	break
CONTINUE	continue
SWITCH	switch

case	CASE
default	DEFAULT
:	COLON
;	SC
,	COMMA
(LPAREN
)	RPAREN
{	LBRACE
}	RBRACE
=	ASSIGN
== != < > <= >=	RELOP
+ - * /	BINOP
[a-zA-Z][a-zA-Z0-9]*	ID
0 [1-9][0-9]*	NUM
"([\n\r\"\\] \\[rnt\\])+"	STRING

ניתן לשנות את שמות האסימונים ואף להוסיף אסימונים נוספים, כל עוד המנתח הלקסיקלי מזהה את כל התבניות לעיל. יש להתעלם מרווחים, ירידות שורה משני הסוגים (\n,\r) וטאבים כך שלא תתקבל עליהם שגיאה לקסיקלית. יש להתעלם מהערות שורה (הערות ב-C) המיוצרות על ידי התבנית `//[^\r\n]*(\r|\n|\r\n)?`

תחביר

יש לכתוב מנתח תחבירי המתאים לדקדוק הבא:

1. $Program \rightarrow Funcs$
2. $Funcs \rightarrow \epsilon$
3. $Funcs \rightarrow FuncDecl Funcs$
4. $FuncDecl \rightarrow RetType ID LPAREN Formals RPAREN LBRACE Statements RBRACE$
5. $RetType \rightarrow Type$
6. $RetType \rightarrow VOID$
7. $Formals \rightarrow \epsilon$
8. $Formals \rightarrow FormalsList$
9. $FormalsList \rightarrow FormalDecl$
10. $FormalsList \rightarrow FormalDecl COMMA FormalsList$
11. $FormalDecl \rightarrow Type ID$
12. $Statements \rightarrow Statement$
13. $Statements \rightarrow Statements Statement$
14. $Statement \rightarrow LBRACE Statements RBRACE$
15. $Statement \rightarrow Type ID SC$
16. $Statement \rightarrow Type ID ASSIGN Exp SC$
17. $Statement \rightarrow ID ASSIGN Exp SC$

18. *Statement* → *Call SC*
19. *Statement* → *RETURN SC*
20. *Statement* → *RETURN Exp SC*
21. *Statement* → *IF LPAREN Exp RPAREN Statement*
22. *Statement* → *IF LPAREN Exp RPAREN Statement ELSE Statement*
23. *Statement* → *WHILE LPAREN Exp RPAREN Statement*
24. *Statement* → *BREAK SC*
25. *Statement* → *CONTINUE SC*
26. *Statement* → *SWITCH LPAREN Exp RPAREN LBRACE CaseList RBRACE*
27. *Call* → *ID LPAREN ExpList RPAREN*
28. *Call* → *ID LPAREN RPAREN*
29. *ExpList* → *Exp*
30. *ExpList* → *Exp COMMA ExpList*
31. *Type* → *INT*
32. *Type* → *BYTE*
33. *Type* → *BOOL*
34. *Exp* → *LPAREN Exp RPAREN*
35. *Exp* → *Exp BINOP Exp*
36. *Exp* → *ID*
37. *Exp* → *Call*
38. *Exp* → *NUM*
39. *Exp* → *NUM B*
40. *Exp* → *STRING*
41. *Exp* → *TRUE*
42. *Exp* → *FALSE*
43. *Exp* → *NOT Exp*
44. *Exp* → *Exp AND Exp*
45. *Exp* → *Exp OR Exp*
46. *Exp* → *Exp RELOP Exp*
47. *Exp* → *LPAREN Type RPAREN Exp*
48. *CaseList* → *CaseDecl CaseList*
49. *CaseList* → *CaseDecl*
50. *CaseList* → *DEFAULT COLON Statements*
51. *CaseDecl* → *CASE NUM COLON Statements*

הערות:

1. הדקדוק כפי שמוצג כאן אינו חד משמעי ב-Bison. יש להפכו לחד משמעי תוך שימור השפה. בעיה לדוגמה שיש לפתור https://en.wikipedia.org/wiki/Dangling_else. יש לפתור בעיה זו ללא שינוי הדקדוק, אלא באמצעות מתן עדיפות לכללים או אסוציאטיביות מתאימה לאסימונים.
2. יש להקפיד על מתן עדיפויות ואסוציאטיביות מתאימים לאופרטורים השונים. יש להשתמש בטבלת העדיפויות כאן:
<https://introcs.cs.princeton.edu/java/11precedence>
3. אין צורך לבצע שינויים בדקדוק, פרט לשם הבדלה בין האופרטורים השונים.

קלט ופלט המנתח

קובץ ההרצה של המנתח יקבל את הקלט מ-stdin.

יש להיעזר בקובץ output.hpp המצורף לתרגיל על מנת לייצר פלט הניתן לבדיקה אוטומטית.

על המנתח להדפיס את כללי הגזירה על פי סדר ביצוע פעולות ה-reduce. בעת ביצוע reduce לכלל יש לקרוא לפונקציה printProductionRule(ruleno) עם מספר הכלל שנגזר.

לתרגיל מסופקים 2 דוגמאות עם קבצי קלט פלט מתאימים. יש לבדוק שפורמט ההדפסה שלכם זהה לדוגמאות. הבדלי פורמט יגרמו לכישלון הבדיקות האוטומטיות.

טיפול בשגיאות

בקובץ הקלט יכולות להיות שגיאות לקסיקליות ותחביריות. **על המנתח לסיים את ריצתו מיד עם זיהוי שגיאה**, כלומר בנקודה העמוקה ביותר בעץ הגזירה שבה ניתן לזהותה.

ניתן להניח כי הקלט מכיל שגיאה אחת לכל היותר.

על מנת לדווח על שגיאות יש להשתמש בפונקציות בקובץ output.hpp:

- שגיאה לקסיקלית – errorLex(lineno)
- שגיאה תחבירית – errorSyn(lineno)

בכל השגיאות הנ"ל lineno הוא מספר השורה בה מופיעה השגיאה.

במקרה של שגיאה, הפלט של המנתח יהיו הכללים שנעשה להם reduce והשגיאה שהתגלתה (כמו שניתן לראות בדוגמה t2)

הערה – כדי להשתמש במשתנים גלובלים של flex ב-bison (לדוגמה yylineno), יש להוסיף בראש קובץ ה-parser.hpp את השורה:

```
extern int yylineno;
```

הוראות הגשה

שימו לב כי קובץ ה-makefile מאפשר שימוש ב-STL. אין לשנות את ה-makefile.

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

1. קובץ flex בשם scanner.lex המכיל את כללי הניתוח הלקסיקלי
2. קובץ בשם parser.hpp המכיל את המנתח
3. את כל הקבצים הנדרשים לבניית המנתח, כולל *.output שסופקו כחלק מהתרגיל.

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

- קובץ ההרצה
- קבצי הפלטי של flex ו-bison
- קובץ ה-makefile שסופק כחלק מהתרגיל

יש לוודא כי בביצוע unzip לא נוצרת תקיה נפרדת על המנתח להיבנות על השרת csComp ללא שגיאות וללא הזהרות באמצעות קובץ makefile שסופק על התרגיל. באתר הקורס מופיע קובץ zip המכיל קבצי בדיקה לדוגמה. יש לוודא כי פורמט הפלט זהה לפורמט הפלט של הדוגמאות הנתונות. כלומר ביצוע הפקודות הבאות:

```

unzip <id1>-<id2>.zip                # unzip to current working directory (CWD)
cp <path-to>/makefile-hw2 ./makefile  # copy & rename the makefile-hw2 to CWD
cp <path-to>/hw2-tests.zip .          # copy the test zip
unzip hw2-tests.zip                  # unzip to CWD
make                                # compile the HW
./hw2 < hw2-tests/t1.in 2>&1 > t1.res  # run on t1.in and dump stdin and stderr to t1.res
diff t1.res hw2-tests/t1.out          # check the diff, should print nothing

```

ייצור את קובץ ההרצה בתקייה הנוכחית ללא שגיאות קומפילציה, יריץ אותו, ו-diff יחזיר 0.

הגשות שלא יעמדו בדרישות לעיל יקבלו ציון 0 ללא אפשרות לבדיקה חוזרת.

בדקו היטב שההגשה שלכן עומדת בדרישות הבסיסיות הללו לפני ההגשה עצמה.

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

בתרגיל זה (כמו בתרגילים אחרים בקורס) **יבדקו העתקות**. אנא כתבו את הקוד שלכן בעצמכן.

בהצלחה 😊.