

תורת הקומפילציה

תרגיל 4

מתרגלת אחראית לתרגיל: הילה פלג hilap@cs

הגשה בזוגות בלבד

שאלות במייל. נא להוסיף HW4 בנושא ההודעה.

לתרגיל ייפתח דף FAQ באתר הקורס. יש להתעדכן בו. הנחיות שיופיעו בדף ה-FAQ עד יומיים לפני הגשת התרגיל מחייבות. שאלות המופיעות כבר בדף ה-FAQ לא יענו. התרגיל ייבדק בדיקה אוטומטית. **הקפידו למלא אחר ההוראות במדויק.**

כללי

בתרגיל זה עליכם לממש ניתוח תחבירי וסמנטי לשפת TEMPERATURE. בשפה זו המרות מובנות בין מידות טמפרטורה: Celsius (C), Fahrenheit (F), Kelvin (K)

להלן טבלת ההמרות בין המידות השונות:
(שימו לב כי הצורך בחישובי ההמרות יהיה בתרגיל בית 5 ולא בתרגיל הנוכחי.)

	from Celsius	to Celsius
Fahrenheit	$[^{\circ}\text{F}] = [^{\circ}\text{C}] \times \frac{9}{5} + 32$	$[^{\circ}\text{C}] = ([^{\circ}\text{F}] - 32) \times \frac{5}{9}$
Kelvin	$[\text{K}] = [^{\circ}\text{C}] + 273.15$	$[^{\circ}\text{C}] = [\text{K}] - 273.15$

מנתח לקסיקלי

יש לכתוב מנתח לקסיקלי שמתאים להגדרות הבאות:

תבנית	אסימון
(LP
)	RP
{	LC
}	RC
;	SC
=	ASSIGN
[+]	PLUS
[-]	MINUS
[*]	MULT
[/]	DIV
<=> = != > <	REL_OP
and	AND
or	OR
C	CELSIUS
F	FAHRENHEIT
K	KELVIN
int	INT

print	PRINT
input	INPUT
[a-zA-Z_]+	ID
\["^"]*\\"	STRING
true	TRUE
false	FALSE
if	IF
else	ELSE
while	WHILE
break	BREAK
not	NOT
bool	BOOL
[1-9][0-9]* 0	NUM
switch	SWITCH
case	CASE

הערה: ניתן לשנות את שמות האסימונים או להוסיף אסימונים נוספים, כל עוד המנתח הלקסיקלי מזהה את כל התבניות לעיל.

יש להניח כי carriage return לא ישמש כירידת שורה לבדו.

יש להתעלם מירידות שורה, טאבים (כך שלא תתקבל עליהם שגיאה לקסיקלית).

יש להתעלם מהערות מסוג C++ (הערות שורה) המיוצגות על ידי התבנית `"/"/["^"]*\\"`

תחביר

יש לכתוב מנתח תחבירי שיתאים לדקדוק הבא:

Prog → *Statements*

Statements → *Statements Statement*

Statements → *Statement*

Statement → *LC Statements RC*

Statement → *Type ID SC*

Statement → *Type ID ASSIGN Exp SC*

Statement → *ID ASSIGN Exp SC*

Statement → *PRINT ID SC*

Statement → *PRINT STRING SC*

Statement → *INPUT ID SC*

Statement → *IF LP Exp RP Statement*

Statement → *IF LP Exp RP Statement ELSE Statement*

Statement → *WHILE LP Exp RP Statement*

Statement → *BREAK SC*

Statement → *SWITCH LP Exp RP LC CasesList RC*

CasesList → *CaseStat CasesList*

CasesList → *CaseStat*

CaseStat → *CASE NUM TemperatureType:Statement BREAK SC*

$TemperatureType \rightarrow CELSIUS$
 $TemperatureType \rightarrow FAHRENHEIT$
 $TemperatureType \rightarrow KELVIN$

$Type \rightarrow TemperatureType$
 $Type \rightarrow INT$
 $Type \rightarrow BOOL$

$Exp \rightarrow LP\ Exp\ RP$
 $Exp \rightarrow Exp\ PLUS\ Exp$
 $Exp \rightarrow Exp\ MINUS\ Exp$
 $Exp \rightarrow Exp\ MULT\ Exp$
 $Exp \rightarrow Exp\ DIV\ Exp$
 $Exp \rightarrow ID$
 $Exp \rightarrow NUM$
 $Exp \rightarrow NUM\ TemperatureType$
 $Exp \rightarrow TRUE$
 $Exp \rightarrow FALSE$
 $Exp \rightarrow NOT\ Exp$
 $Exp \rightarrow Exp\ OR\ Exp$
 $Exp \rightarrow Exp\ AND\ Exp$
 $Exp \rightarrow Exp\ REL_OP\ Exp$

הערות

1. הדקדוק כפי שהוגדר כאן אינו חד משמעי ב-Bison. יש להפכו לחד-משמעי תוך שימור השפה.
בעיה לדוגמא שיש לפתור: http://en.wikipedia.org/wiki/Dangling_else
2. יש להקפיד על מתן עדיפויות ואסוציאטיביות מתאימים לאופרטורים השונים. יש להשתמש בטבלת העדיפויות כאן: <http://introcs.cs.princeton.edu/java/11precedence>
3. עבור כלל הדקדוק $Statement \rightarrow BREAK\ SC$ יש לבצע בדיקה שהוא מתגלה רק בתוך לולאת while. אחרת יש לעצור עם שגיאת UnexpectedBreak, ולצאת מהתכנית. ראו סעיף "טיפול בשגיאות" בהמשך.

טבלאות סמלים

בשפת TEMPERATURE יש קינון סטטי של scopes – כל משתנה מוגדר ב-scope שבו הוכרז, ובכל הצאצאים של אותו scope. **אסור להכריז על משתנה שכבר מוגדר באותו scope** ואסור להשתמש במשתנה שלא מוגדר. שימוש במשתנה הוא כל מופע פרט להכרזה שלו.

קטע הקוד הבא חוקי מבחינה תחבירית:

```
int a;
int a;
```

וגם:

```
int a;
b=6;
```

לא נרצה לאפשר אותם בשפת TEMPERATURE. לכן יש לנהל טבלאות סמלים. יש להשתמש בטבלאות הסמלים כדי לבצע את הבדיקות הבאות:

1. בכל הכרזה על משתנה יש לוודא שמשתנה בשם זה לא מוגדר ב-scope הנוכחי או באחד האבות שלו.
2. בכל שימוש במשתנה יש לוודא שהוא מוגדר.
- לכל משתנה יש לשמור בטבלה את שמו, מיקומו היחסי והטיפוס שלו.

הערות

1. ניתן להגדיר בתוך ה-scope של בלוק בענפי if, else או while (מדובר בבלוק לכל דבר, שיתבצע או לא בהתאם לזמן הריצה).
2. אם בתוך ענף ה-if/else או בלולאת ה-while יש פקודה בודדת (ללא פתיחת בלוק) היא תחשב כ-scope חדש אם הפקודה מהווה הכרזה על משתנה חדש.
3. בפקודת switch כל הכרזת case מתנהגת כענף נפרד.

בדיקת טיפוסים

יש לממש את בדיקות הטיפוסים הבאות:

1. אין המרה בין bool לאף טיפוס אחר.
2. טיפוס טמפרטורה ניתן להמרה בטיפוס טמפרטורה אחר בלבד.
3. אין המרה בין int לאף טיפוס אחר.
4. טיפוסים האופרנדים של פעולה לוגית הם bool.
5. ביטויי התנאי במשפטי if, while הם מטיפוס bool.
6. מותר לבצע חיבור וחסור של טיפוסים טמפרטורה בינם לבין עצמם. בקביעת הטיפוס של פעולת חיבור או חיסור, תתבצע המרה לאחד האופרנדים אם שני האופרנדים אינם מאותו טיפוס, ותוצאת הביטוי תיקבע לטיפוס האופרנדים אחרי ההמרה.
- בתת עץ $Exp \rightarrow Exp1 \text{ OP } Exp2$ הדורש המרה, הטיפוס יקבע לפי הטיפוס של $Exp1$.
7. אסור לבצע פעולת כפל בין שני ביטויים מטיפוס טמפרטורה. מותר לבצע כפל בין טיפוס טמפרטורה לבין int וטיפוס התוצאה יהיה זהה לטיפוס של אופרנד הטמפרטורה המעורב.
8. מותר לבצע פעולת חילוק בין שני טיפוסים טמפרטורה. תוצאת הביטוי תהיה מטיפוס int. מותר לחלק טיפוס טמפרטורה ב-int. תוצאת הביטוי תהיה טיפוס אופרנד הטמפרטורה המעורב. אסור לחלק int בטיפוס טמפרטורה.
9. מותר לבצע פעולות חיבור\חיסור\כפל\חילוק בין שני אופרנדים מטיפוס int. טיפוס התוצאה הוא int.
10. מותר לבצע השמה של ביטוי מטיפוס טמפרטורה לתוך משתנה מטיפוס טמפרטורה בלבד. אין חובה שהמשתנה יהיה מאותו טיפוס טמפרטורה כמו הביטוי – אם הטיפוס אינו זהה תתבצע ההמרה המובנית.
11. מותר לבצע פעולות השוואה בין שני אופרנדים מטיפוס טמפרטורה (שוב: אין חובה ששני האופרנדים יהיו מאותו טיפוס טמפרטורה, יש המרה מובנית), או בין שני אופרנדים מטיפוס int.
12. בפקודות input ו-print טיפוסים האופרנדים הם int או טיפוסים טמפרטורה.
13. ביטוי מספרי הנגזר מהכלל $Exp \rightarrow NUM \text{ TemperatureType}$ יהיה מטיפוס הטמפרטורה שצוין בתכנית הקלט, ואילו ביטוי הנגזר מהכלל $Exp \rightarrow NUM$ יהיה מטיפוס int.
14. במבנה switch הנגזר מהכלל $Statement \rightarrow SWITCH LP Exp RP LC CasesList RC$ יהיה טיפוס הביטוי EXP טיפוס טמפרטורה.

מיקום המשתנים בזיכרון
נניח שכל משתנה הוא בגודל 1. אזי עבור הקוד הבא:

```
int x;  
{  
    C y;  
}  
bool z;
```

ה- offsets יהיו:

x	y	z
0	1	1

קלט ופלט המנתח
קובץ ההרצה של המנתח יקבל ארגומנט בודד ובו נתיב לקובץ הקלט.
יש להיעזר בקובץ output.hpp המצורף לתרגיל על מנת לייצר פלט הניתן לבדיקה אוטומטית.
בסוף כל scope, כולל ה-scope הגלובאלי, המנתח ידפיס את המשתנים שהוגדרו ב-scope זה ל-
stdout לפי הסדר הבא:

1. קריאה לפונקציה endScope()
 2. עבור כל משתנה שהוגדר ב-scope על פי סדר ההכרזה על המשתנים בקוד יש לקרוא לפונקציה printVar(id,offset,type) עם שם המשתנה, המיקום בזיכרון, והטיפוס. מחרוזת הטיפוס צריכה להיות זהה לשם האסימון שהוגדר לטיפוס בחלק הלקסיקלי בתיאור התרגיל.
 3. שימו לב לבצע זאת בסוף כל scope לפי ההגדרה בפרק טבלת הסמלים של תיאור התרגיל.
- ניתן קובץ פלט לדוגמא. יש לבדוק שהפורמט שהודפס זהה אליו. הבדלי פורמט יגרמו לכשלון הבדיקות האוטומטיות.

טיפול בשגיאות
בקובץ הקלט יכולות להיות שגיאות לקסיקליות, תחביריות וסמנטיות. על המנתח לסיים את ריצתו מיד עם זיהוי השגיאה הראשונה. ניתן להניח כי הקלט מכיל שגיאה אחת לכל היותר.

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

errorLex(lineno)	שגיאה לקסיקלית
errorSyn(lineno)	שגיאה תחבירית
errorUndef(lineno, id)	שימוש במשתנה שלא הוגדר
errorDef(lineno, id)	ניסיון להגדיר משתנה שכבר מוגדר
errorMismatch(lineno)	אי התאמה של טיפוסים
errorUnexpectedBreak(lineno)	פקודת break שאינה חלק מלולאה

כאשר id הוא שם המשתנה ו-lineno הוא מספר השורה של השגיאה.

הדרכה והערות

מומלץ לעבוד לפי סדר הפעולות הבא:

1. כתבו מנתח לקסיקלי ותחבירי ללא כללים סמנטיים
2. בדקו שהמבנה התחבירי של השפה נאכף ואין אף קונפליקט
3. הגדירו את YYSTYPE וממשו טבלת סמלים

4. בצעו בדיקות סמנטיות

הוראות הגשה

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

- את כל הקבצים הדרושים לבניית המנתח
- ללא קובץ ההרצה או קבצי הפלט של flex ו-bison
- Makefile המריץ את flex ו-bison ובונה קובץ הרצה בשם hw4

יש לוודא כי בביצוע unzip לא נוצרת תיקיה נפרדת. יש לוודא כי פורמט הפלט זהה לפורמט הפלט של הדוגמה הנתונה. כלומר, ביצוע הפקודות

```
unzip
make
./hw4 path-to/t1.in > out.txt
diff out.txt path-to/t1.out
```

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

על המנתח להיבנות על השרת t2 ללא שגיאות.

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

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

בהצלחה!