236319 – שפות תכנות

תרגיל בית 2 – חלק יבש

תאריך הגשה: 4.5.2023

שם סטודנט	ת.ז
מורן עאמר	322730789
עדן סרחאן	324849256

שאלה 1:

נתון דקדוק של EmojiLang הנתון כ-EBNF:

```
<statements> = <statement> <statements> | <statement>
<statement> = ✍️<variable>⬅️<expression>
<statement> = 📖<expression>
<expression> = ↪️<variable>
<expression> = <term>
<expression> = <expression> <operation> <expression>
<variable> = 📗|📘|📙|📕|📔
<term> = 😇|🙂|😐|😕|☹️|☹️|😣
<operation> = ➕|➖|✖️|➗
```

1. רשמו את רשימת הטרמינלים, הלא טרמינלים וסימבול התחלתי של הדקדוק.

טרמינלים:

😇,🙂,😐,😕,☹️,☹️,😣,📗,📘,📙,📕,📔,➕,➖,✖️ and➗

also ✍️,⬅️,↪️and📖

לא טרמינלים:

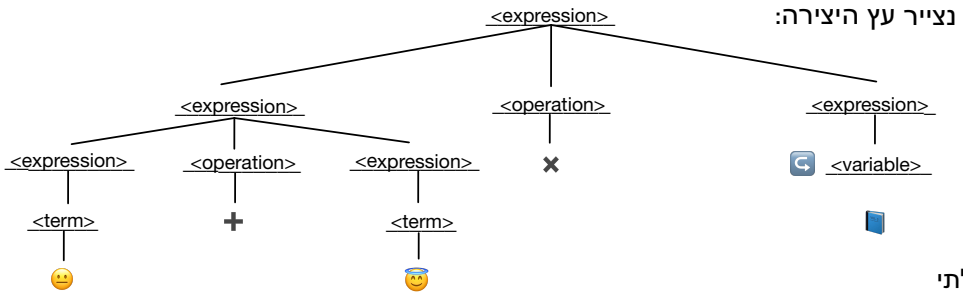
<statements> , <statement> , <expression> , <variable> , <term> , and <operation>

סימבולים התחלתיים:

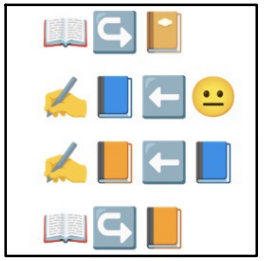
<statements>

2. עבור כל אחת מסדרות הטרמינלים קבעו האם היא שייכת לדקדוק:

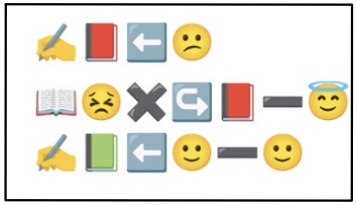
✗ לא שייך לדקדוק: כי הוא לא מתחיל בסימבול ההתחלתי של הדקדוק, נצייר עץ היצירה:



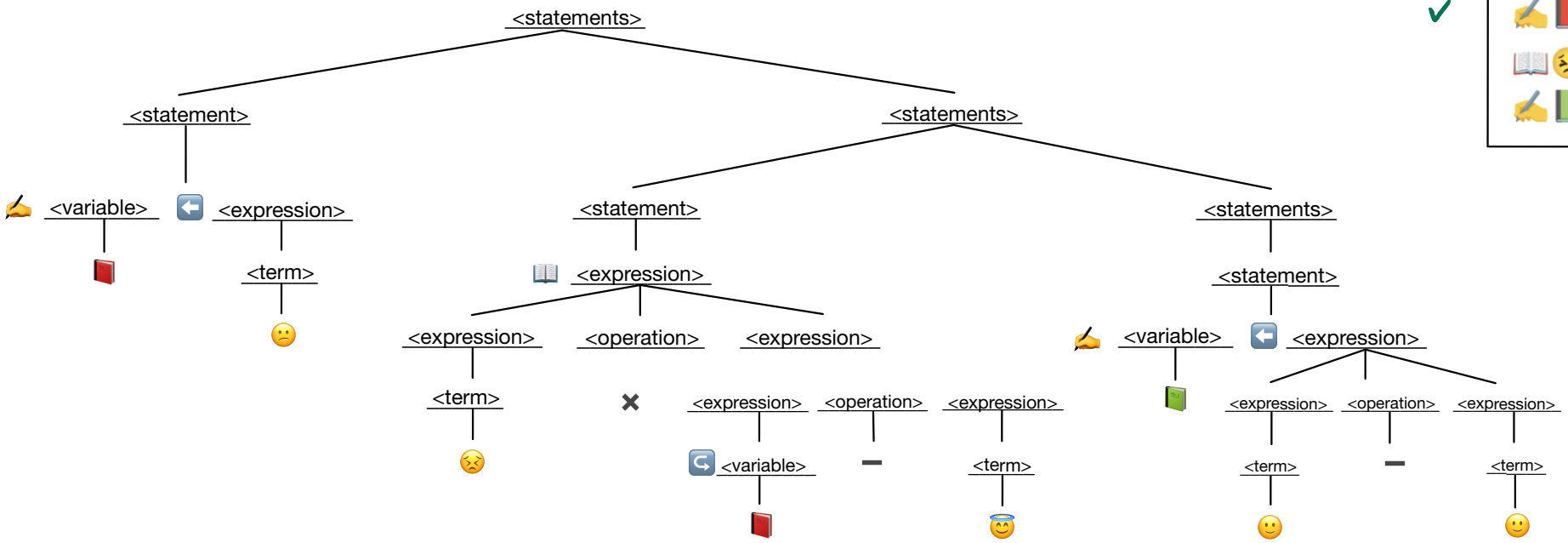
כפי שרואים, הסדרה מתחילה ב-<expression> שזה אינו סימבול התחלתי

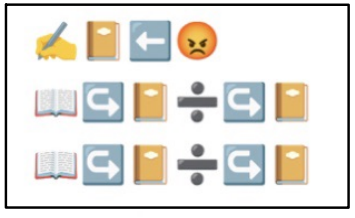


✗ לא שייך לדקדוק, האימוג'יז 📖 ו- ✍️ הם סימבולים ייחודיים והם מאתחלים רק לא-טרמינלים מסוג <statement>, כיוון ששני הביטויים הם <statement> לכן לפי חוקי הדקדוק הצמד שלהם הוא מסוג <statements>.
כיוון שיש רק דרך יחידה להתחיל <statement> באימוג'י ✍️ לפי החוקים, לכן תת-הביטוי ✍️📙📘⬅️ חייב להיות מצורת החוק שני.
כלומר 📘 הוא <expression>, אבל הוא גם כן <variable> והדרך היחידה ש<expression> יכיל <variable> אם יש 🔁 לפניו (לפי החוקים) שזה לא קיים בתת הביטוי.



✓





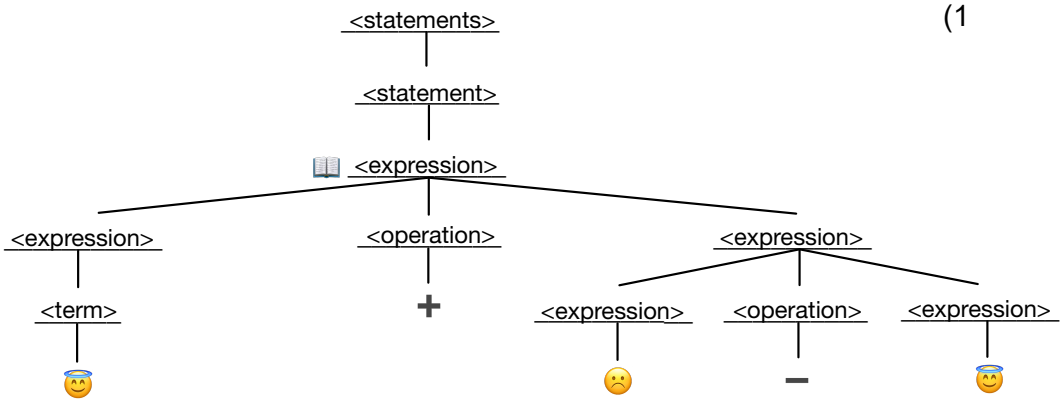
✗ לא שייך לדקדוק: הסימבול 😡 אינו בדקדוק לכן סדרת הטרמינלים אינה חוקית

3. האם הדקדוק מכיל ambiguities? כן

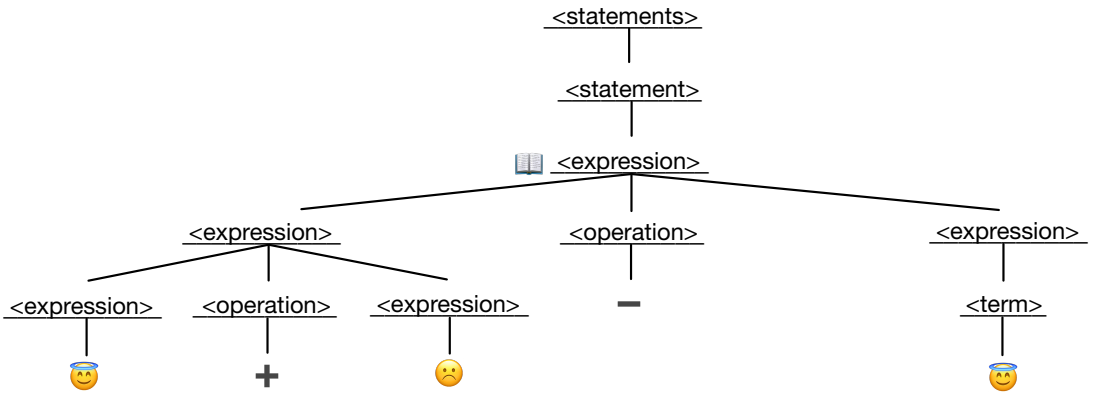
נתבונן בסדרת הטרמינלים הבאה: 📖😇+☹️−😇

נצייר שני עצי יצירה שונים:

1)



2)



מתקיים כי שני עצי היצירה שונים אבל שניהם יוצרים את אותה הסדרה, כלומר יש יותר מדרך אחת ליצור אותה סדרת הטרמינלים ולכן לפי הגדרה יש דו-משמעות (ambiguity).

שאלה 2:

עבור כל אחת מהקריאות הבאות הסבירו בקצרה למה היא מחזירה שגיאה ב-SML:

```
1.   "h" ^ #"C";
2.   84 / 2;
3.   fun f x = if x = 0 then x else false;
4.   fun f x = if x = #"a" then x ^ "b" else x ^ "c";
5.   5 - (-3);
6.   Math.sqrt 9;
7.   sin 3.14;
8.   val if = 5;
9.   String.sub ("hello", 5);
10.  fun sqrt_of_int x:int = Math.sqrt (real x);
```

(סליחה מראש שהתשובה כתובה באנגלית)

1. the ^ operator is of type string*string -> string, whereas #"C" is a char, since there aren't any automatic conversions in SML then this line of code gives an error because ^ cannot merge string and char.
2. the / function is of type real*real -> real, both of the given values (84 and 2) are integers and not reals. since there aren't any automatic conversions in SML then this line of code gives an error because / cannot be applied on integers.
3. functions in SML should return values of the same type, in the given code x is compared to 0 therefore x is an int, but the function returns either x or false, one of which is an integer whereas the second is a Boolean.
4. in the condition we can see that x is compared to #"a", since #"a" is a char therefore x is also a char, but then we see that the operator ^ is used between x and "b" or x and "c" which are strings, just like the error explained in (1.), operator ^ cannot merge char and string.
5. in SML, to write "minus 3" as in "3 bellow zero" we must use ~ and not -, - is a function that get two arguments, so the given code gives an error because – is used incorrectly (instead we should use ~ or add 0 before the -3).
6. Math.sqrt is of type real->real, whereas 9 is an integer , since there aren't any automatic conversions in SML then this line of code gives an error.
7. sin is not an identifies function in the standard library of SML.
8. "if" is a reserved keyword in SML, it cannot be used to name values and variables.
9. since counting in a string starts from zero (aka the letter "h" is in index 0 aand the letter "o" is in index 4), the given index (5) is out of range in "hello" string.
10. Math.sqrt returns a real, where as the function sqrt_of_int returns int (because of type constraints) , there is a clash of the returned type, and since there aren't any automatic conversions in SML then this line of code gives an error