

### תרגיל בית 3 בשפות תכנות (236319) - חלק יבש

שאלה 1:

סעיף א:

עבור מספרים בייצוג עשרוני נקבל את הביטוי הרגולרי הבא:

```
([\d]+[_]?[\d]?\.\.[\d][eE]?[+-]?[\d]?[_?][\d]?[_?][\d])
| ([\d]+[_]?[\d]*[eE]?[+-]?[\d]+[_]?[\d]*)
| (\.\.[\d]+[_]?[\d]*[eE]?[+-]?[\d]+[_]?[\d]*?)
```

עבור מספרים בייצוג הקסה-דצימלי, נקבל את הביטוי הרגולרי הבא:

```
(0[xX][_]?(([d]|([A-F]|[a-f])) ([_]? ([\d]|([A-F]|[a-f])))*)\.\.([\d]|([A-F]|[a-f])) ([_]?([\d]|([A-F]|[a-f])))*)
| (0[xX][_]?(([d]|([A-F]|[a-f])) ([_]?([\d]|([A-F]|[a-f])))*)[pP][+-]?(([d])([_]?[\d]))*)
| (0[xX]\.(([d]|([A-F]|[a-f])) ([_]?([\d]|([A-F]|[a-f])))*)[pP][+-]?(([d])([_]?[\d]))*)
| (0[xX]\.(([d]|([A-F]|[a-f])) ([_]?([\d]|([A-F]|[a-f])))*)[pP][+-]?(([d])([_]?[\d]))*)
```

לבסוף עבור כל ה-floating point literals שפה go נקבל:

```
(([\d]+[_]?[\d]*\.\.[\d][eE]?[+-]?[\d]*[_]?[\d])
| ([\d]+[_]?[\d]*[eE]?[+-]?[\d]+[_]?[\d]*)
| (\.\.[\d]+[_]?[\d]*[eE]?[+-]?[\d]+[_]?[\d]*))
| ((0[xX][_]?(([d]|([A-F]|[a-f])) ([_]? ([\d]|([A-F]|[a-f])))*)\.\.([\d]|([A-F]|[a-f])) ([_]?([\d]|([A-F]|[a-f])))*)[pP][+-]?(([d])([_]?[\d]))*)
| (0[xX][_]?(([d]|([A-F]|[a-f])) ([_]?([\d]|([A-F]|[a-f])))*)[pP][+-]?(([d])([_]?[\d]))*)
| (0[xX]\.(([d]|([A-F]|[a-f])) ([_]?([\d]|([A-F]|[a-f])))*)[pP][+-]?(([d])([_]?[\d]))*)
```

\* הערכה: כתבנו את הביטוי בלבד (הוא עובר את הדוגמאות המסופקות במדריך go על ידי בדיקה באתר <https://regex101.com> . ניסינו לעשות מספר איטרציות עם chatgpt אך נראה שככל חוקרי הבינה למניהם לא עשו עבודה ככ טובה בلمידת ביטויים רגולריים....).

סעיף ב:

שפה התכנות go משתמשת במיללים שמוראות כמו בשפת התכנות C שלא ניתן להשתמש בהן כمزחים, שמות של משתנים, שמות של פונקציות וכו' .. בשפה קיימים גם מזחים מוגדרים מראש אך המשמעות שלהם קבועות ומהוות חלק מפרט שפת Go.

## Operators and punctuation

The following character sequences represent operators (including assignment operators) and punctuation [[Go 1.18](#)]:

```
+ & += &= && == != ( )
- | -= |= || < <= [ ]
* ^ *= ^= <- > >= { }
/ << /= <<= ++ = := , ;
% >> %= >>= -- ! ... . :
&^ &^= ~
```

## Keywords

The following keywords are reserved and may not be used as identifiers.

```
break default func interface select
case defer go map struct
chan else goto package switch
const fallthrough if range type
continue for import return var
```

Punctuation	Names			
	Names of types	Names of values	Names of operators	Control flow keywords
See detailed list above (from the Go programming language specification <a href="https://go.dev/ref/spec#Identifiers">https://go.dev/ref/spec#Identifiers</a> )	Struct, const, var, package, map, interface, func, type, chan	default	Import, select, defer, go	If, else, break, case, continue, switch, goto, return, fallthrough, range, for

**שאלה 2:**

**סעיף א:**  
:Match  
אני חושב שזו דוגמה יפה:

```
datatype piece = king | queen | rook | bishop | knight | pawn
;
fun value king = 1.0
value queen = 9.0 |
value rook = 5.0 |
value bishop = 3.0 |
value knight = 4.0 |
;
val nice = pawn
;
value nice;
```

Div:

```
5 div 0;
```

Bind:

```
val "x" = "y" ;
```

**סעיף ב:**

הטיפוס של כל אחד מהם הוא: Div :SML exception:  
הראשון טריוויאלי handle statement אבל במקרה זהה היא לא משנה את הטיפ של הביטוי  
השני מכיל פונקציית `handle` statement לא בפניהם הינה פשוטות תופסת את השגיאה ומעלה אותה עוד פעם  
השלישי מעלה את השגיאה `explicitly`.

**סעיף ג:**

```
fun safediv x y = (SOME (x div y) handle Div => NONE);
```

שאלה 3:

exception InvalidSExp

fun cons a b = CONS (a, b)

fun car (CONS (a, \_)) = a  
| car (ATOM \_) = raise InvalidSExp

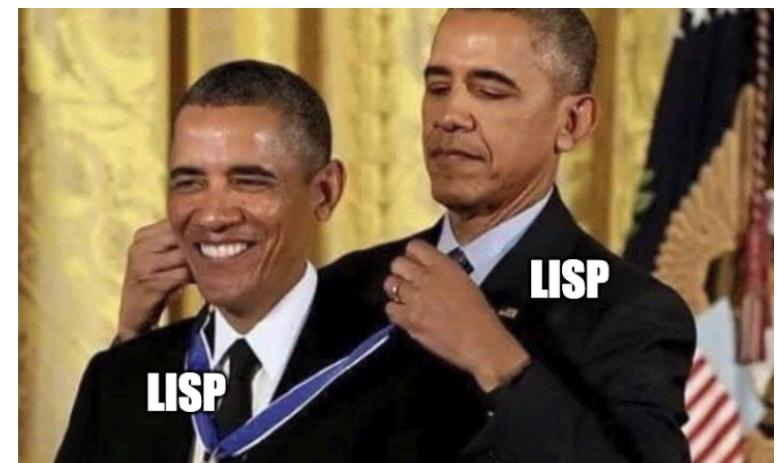
fun cdr (CONS (\_, b)) = b  
| cdr (ATOM \_) = raise InvalidSExp

fun toString (ATOM (SYMBOL a)) = a  
| toString (ATOM (NIL)) = "NIL"  
| toString (CONS (a,b)) = "(" ^ toString(a) ^ " . " ^ toString(b) ^ ")"



finished hw3\_q1.sml. i hate my life.

 bbdaria committed yesterday



אנחנו מאשר צאטגיפטי לא יודע ביטויים רגולריים  
וגם אנחנו לא יודעים ביטויים רגולריים



פותח מטלה חדשה בשפות

