Computational Engineering 2022-01-21

BM40A0202 Foundations of Computer Science

Olli-Pekka Hämäläinen

Exercise 6: From algorithm to machine language, translation process.

Tasks (1 p/task)

1. Newton's formula can be used to calculate the zero point (or zero points) of the function $f(x)$, the iterative formula being
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$
If the function is $f(x) = x^2 - N$, the same formula can be used to calculate the square root of N. Simplify the iteration formula to a form where only x_n and N occur (without exponents).

2. In symbolic machine language, present a program that computes the square root of N using the iteration formula simplified in the previous task.

3. In order for a compiled algorithm to run on a particular computer, it must be translated into a format that the computer can understand. In computing the need to translate languages into another language is also due to the large number of ways in which algorithms can be represented.
   (a) What are the two ways of running algorithms in a computer? What is the most obvious difference between them?
   (b) What are the three stages of a (programming language) compiler? What are the purposes of the steps? Which function(s) are often used in practice in addition to these three steps?
   (c) What are the common ways of storing information in the three key stages?

4. The parsing table for the following grammar

$$\begin{array}{lll lll} E & \rightarrow & TE' & T' & \rightarrow & *FT' \mid \epsilon \\ E' & \rightarrow & +TE' \mid \epsilon & F & \rightarrow & (E) \mid \mathbf{id} \\ T & \rightarrow & FT' & & & \end{array}$$

is (the start symbol is E) as follows:

Intermediate-	Input symbol					
symbol	**id**	+	*	(	)	$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow \mathbf{id}$			$F \rightarrow (E)$		

and then the parsing of the input expression $\mathbf{id} + \mathbf{id} * \mathbf{id}$ (parsing from the left) starts with the products

$$E \Rightarrow TE' \Rightarrow FT'E' \Rightarrow \mathbf{id}T'E' \Rightarrow \mathbf{id}E' \Rightarrow \mathbf{id} + TE' \Rightarrow ...$$

(a) Continue to parse (reduction) in the same way.

(b) Is the given sentence a permitted sentence in the language?

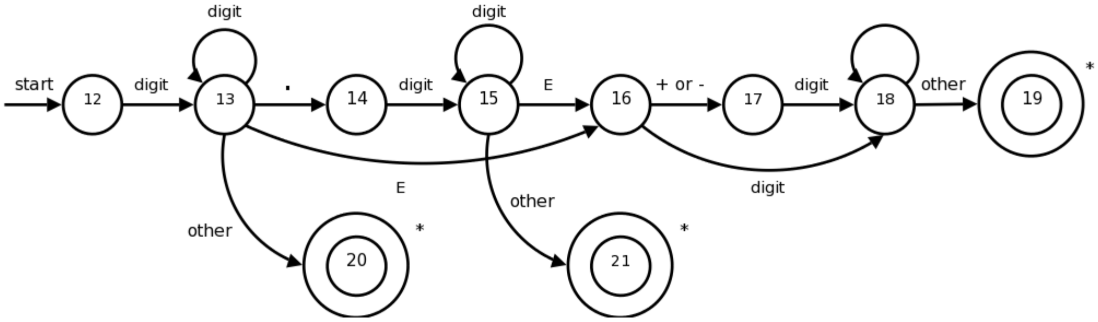
(c) What other sentences would be grammatical?

(d) What if the input is sentence $\mathbf{id} * \mathbf{id} + *$?

5. For parsing, a browser must be specified which returns the next textual subset (token) of the primary language program. The browser is implemented as a state machine that either recognizes the text subset or does not recognize the text subset.

The state machine is defined as

- digit = [0...9]
- other = any character other than [digit | . | E | + | -]



In modes 19, 20 and 21, the * sign indicates that the last character read belongs to the next token to be recognized.

a) Which tokens will the state machine recognise?

b) Which tokens it does not recognize?