Em🤪tica

Entwurf einer Programmiersprache mit Instruktionen,
basierend auf dem Unicode Emoji-Zeichensatz

Modul FAC.BSc INF 2014.ZH1.FS18
Samuel Blattner
März 2018

Spezifikation der kontextfreien Grammatik

Allgemeine Definition

G = {Σ, N, S, P}

Terminale (Σ)

Die Menge der Terminale besteht aus dem Unicode-Zeichensatz. Dies schliesst zahlreiche Symbole (sog. «Emoticons» oder «Emojis») mit ein.

Σ = Menge der Zeichen des Unicode-Zeichensatz

Nicht-Terminale (N)

N = {instruction, assignment, conditional, function, loop, functioncall, value, condition, block, variable, string, number, operation, comparison, operator, comparator, built-in}

Startsymbol (S)

S = instruction

Produktionen (P)

instruction	—›	instruction ";" instruction
instruction	—›	conditional instruction
instruction	—›	function instruction
instruction	—›	conditional
instruction	—›	function
instruction	—›	assignment
instruction	—›	loop
instruction	—›	functioncall
instruction	—›	built-in
assignment	—›	value ➡️ variable
conditional	—›	❓ condition 🙂 block
conditional	—›	❓ condition 🙂 block 🙁 block
function	—›	⚡ variable 📤 varlist block
loop	—›	♻️ condition block | ♻️ variable range block
functioncall	—›	🌀 variable 📥 valuelist
value	—›	variable | string | number | operation | functioncall
condition	—›	comparison | operation

range —› ↔️ number ⏩ number | ↔️ number ⏭️ number
block —› 👉 instruction ✋

varlist —› ε | variable "," varlist
valuelist —› ε | value | value "," valuelist

variable —› RegEx: [A-Za-z_][A-Za-z0-9_]*
string —› 💬 Σ*💬
number —› RegEx: \d*\.?\d+

operation —› value operator value
comparison —› value comparator value
operator —› "+" | "-" | "*" | "/" | "^" | "%" |
comparator —› ">" | ">=" | "<" | "<=" | "==" | "!="

built-in —› 📢 value

Beispielprogramme

Hello World

```
💬 Hello World ➡️ myString;

⚡ shoutToTheWorld 📤 count 👉 📢 myString + 💬 for the💬 + count + 💬. time💬 ✋

♻️ counter ↔️ 0 ⏩ 100 👉
        🌀 shoutToTheWorld 📥 counter;
✋
```