

COMP712 Programming Languages

Interpreter Implementation Project

This document specifies a programming language called AJS (A JavaScript) which is a subset of ECMAScript 2018. The goal of the project is to implement an interpreter for AJS using Racket with the help of the **SLLgen** lexer and parser generator and the **Datatype** macro extension introduced by D. Friedman and M Wand in their book *Essentials of Programming Languages*, third edition.

1. Syntax

The syntax is written in a variant of BNF described in A Human Engineered Variant of BNF, ACM SIGPLAN Notices, Vol. 15, issue 10, Oct. 1980, pp. 57-62. In the grammar below, **bold** font is used for keywords, *italics* for syntactic variables, ϵ for nothing, $x \dots$ for zero or more repetitions of x .

program	::=	statement...	Program
statement	::=	const name = expression ;	Constant declaration
		function name (names) block	Function declaration
		return expression ;	Return statement
		if-statement	Conditional statement
		block	Block statement
		expression statement	Expression statement
names	::=	name (, name)...	Name list
if-statement	::=	if (expression) block	
		else (block if-statement)	Conditional statement
block	::=	{ statement }	Block statement
expression	::=	number	Primitive number
		true false	Primitive Boolean
		string	Primitive string
		null	Primitive list
		name	Name expression
		expression binary-operator expression	Binary operator combination
		unary-operator expression	Unary operator combination
		expression binary-logical expression	Logical composition
		expression (expressions)	Function application
		(name (names)) => expression	Lambda expression
		(name (names)) => block	Lambda expression
		expression ? expression : expression	Conditional expression
		(expression)	Parenthesized expression
binary-operator	::=	+ - * / % === !==	
		> < >= <=	Binary operator
Unary-operator	::=	! -	Unary operator
Binary-logical	::=	&&	Logical composition
Expressions	::=	ϵ expression (, expression)...	Argument expressions

Restrictions

- (a) Return statements are only allowed in bodies of functions.
- (b) There cannot be a newline character between return and expression in return statements.
- (c) There cannot be any newline characters between (name | (parameters)) and => in function definition expressions.
- (d) Implementations of AJS are allowed to treat function declaration as syntactic sugar for constant declaration. Programmers need to ensure that functions are not called before their corresponding function declaration is evaluated.

Names

Name (identifier) must start with `_`, `$` or a letter and contain only `_`, `$`, letters or digits. Keywords are not allowed as names. Keywords include any of the words in bold in the syntax rules above.

Numbers

Only decimal numbers are used, with an optional decimal point. Scientific notation is not allowed. Example numbers are therefore, 1234, -3456, and 567.8.

Strings

String is a possibly empty sequence of characters within double-quotes. For example, "This is a string." Newline and other special characters can be represented as follows:

- Newline: `\n`
- Single quote: `\'`
- Double quote: `\"`

Comments

Single-line comments start with `//` and therefore any characters until the next newline character is ignored.

Library Function

There is no need to provide any sort of library function other than the arithmetic and Boolean operators shown in the syntax.