SICPy §1

Khooi Xin Zhe, Ang Jun Jie

19 April, 2021

A SICPy[1] program is a *module*, defined using Backus-Naur Form[2] as follows:

SICPy §1

module	::=	*statement* ...	statement sequence
statement	::=	*name* = *expression*	assignment statement
	|	*function*	function declaration
	|	**return** *expression*	return statement
	|	*if-statement*	conditional statement
	|	*expression*	expression statement
function	::=	**def** *name* (*parameters*) :	
		statement ...	function declaration
parameters	::=	ϵ | *name* [, *name*] ...	function parameters
if-statement	::=	**if** *expression* :	
		statement ...	
	[[	**elif** *expression* :	
		statement ...] ...	
		else :	
		statement ...]	conditional statement
expression	::=	*number*	primitive number expression
	|	**True** | **False**	primitive boolean expression
	|	*string*	primitive string expression
	|	*name*	name expression
	|	*expression binary-operator expression*	binary operator combination
	|	*unary-operator expression*	unary operator combination
	|	*expression* (*expressions*)	function application
	|	**lambda** *name* [, *name*] ... : *expression*	lambda expression
	|	*expression* **if** *expression* **else** *expression*	conditional expression
	|	(*expression*)	parenthesised expression
binary-operator	::=	+ | - | * | / | % | ==	
		| > | < | >= | <= | **and** | **or**	
unary-operator	::=	**not** | + | -	
expressions	::=	ϵ | *expression* [, *expression*] ...	argument expressions

---

[1]SICPy is an adaptation of Source - the official language of the textbook Structure and Interpretation of Computer Programs, JavaScript Adaptation.

[2]We adopt Henry Ledgard's BNF variant that he described in A human engineered variant of BNF, ACM SIGPLAN Notices, Volume 15 Issue 10, October 1980, Pages 57-62. In our grammars, we bold and underline keywords, [ ] for optional syntaxes, italics for syntactic variables, ϵ for nothing, $x|y$ for x or y, and x... for zero or more repetitions of x.