

Homework: ArithLang

Learning Objectives:

1. Write programs in Arithlang
2. Understand and extend Arithlang interpreter

Instructions:

1. Total points: 50 pt
2. Early deadline: Sept 18 (Wed) 6:00 pm, Regular deadline Sept 20 (Fri) 6:00 pm
3. Download tutorial.zip and tutorial.pdf from Canvas
4. Set up the programming project following the instructions in the tutorial
5. How to submit:
 - For questions 1–3, you can write your solutions in latex or word and then convert it to pdf; or you can submit a scanned document with legible handwritten solutions. Please provide the solutions in one pdf file.
 - For questions 4 and 5, please submit your solutions in two different zip files with all the source code files (just zip the complete project's folder).
 - Submit two zip files and one pdf file to Canvas under Assignments, Homework 2

Questions:

1. (4 pt) Write two Arithlang programs that compute 21, containing three or more operators.
2. (4 pt) Get familiar with the syntax of Arithlang: construct parse trees for the following programs in ArithLang
 - (a) $(* (- 120 (+ 4 2)) (/ 6 2))$
 - (b) $(+ 43 (* 6 6 6) (- 100 17))$
3. (6 pt) Get familiar with the ArithLang source code:
 - (a) What are the purposes of the AST and Evaluator classes?
 - (b) Use an example ArithLang program to explain how the visitor pattern works to compute the value of this program?
 - (c) How the visitor pattern is implemented in the Evaluator and AST classes?

4. (10 pt) Extend ArithLang to support the power operation, (pow num1 num2), e.g.,
 (pow 3 2) = 9
 (pow (pow 2 3) 2) = 64
 (pow -3 2 5) = 59049
 (pow 10) = error
 (pow 10 -2) = error
5. (26 pt) Implement an interpreter for AbstractLang described as follows. You can modify from the ArithLang code.
- This language contains only five terminals, e, o, u, *, +
 - e represents even numbers, o represents odd numbers and u represent unknown values
 - There are two operators * and + that can be applied on the terminals, their syntactic rules are similar to * and + in ArithLang, except that each operator only can take two operands
 - The semantics of AbstractLang are defined by the following rules (the first column in the table represents the first operand and the first row in the table represents the second operand of the operation):

+	e	o	u
e	e	o	u
o	o	e	u
u	u	u	u

*	e	o	u
e	e	e	u
o	e	o	u
u	u	u	u

e.g.,

(+ e e) = e
 (* (+ e o) e) = e
 (+ e o e) = error
 (+ 3 4) = error
 (* 2 e) = error
 (+ a b) = error