CS 4620
assigned: 09/9/2022
due: 09/23/2022 at 5pm

12 points

Course Project Deliverable 1

The course project provides a context for you to study concepts related to compilation more deeply. This is the first of 5 project deliverables:

Deliverable 1 Extend the parser to support SIP

Deliverable 2 Extend the front end to support SIP

Deliverable 3 Extend the semantic analyses to support SIP

Deliverable 4 Extend the code generator to support SIP

Deliverable 5 Extend the optimizer

The project is solved in teams of two, unless you want to do it by yourself.

Deliverable 1 : Problem Statement

You are to extend the tipc parser to support the constructs in SIP – an extension of the TIP language which is described at `https://cs.au.dk/~amoeller/spa/`. SIP introduces the following extensions to TIP:

- Boolean type with constants `true` and `false`, unary operator `not`, and non-short circuiting binary operators `and` and `or`.
- Array type with constructors `[E1, ..., En]` and `[E1 of E2]`, a unary prefix length operator `#`, and a binary array element reference operator `E1[E2]`. Note that it is permissable to have an empty array in the first of these type constructors, e.g., `[]`; the length of such an array is `0`, i.e., `#[] == 0`.
- Arithmetic operators are extended with `%`, the modulo operator, and `-`, the arithmetic negation operator.
- Relational operators are extended with `>=`, `<`, and `<=`.
- Ternary conditional expression operator `E1 ? E2 : E3` which implements an *if then else* expression.
- Increment and decrement statements `E++;` and `E--;`.
- For loop using an iterator-style `for (E1 : E2) S`.
- For loop using a range-style `for (E1 : E2 .. E3 by E4) S`, where the `by E4` increment specification is optional and defaults to 1.

The precedence and associativity of operators in SIP follows the definition of operators in C.

In extending the parser, you must not introduce any errors in parsing TIP programs, i.e., programs written without any of the SIP extensions. You may find, however, that illegal TIP programs become legal SIP programs due to the extensions. If you find tests that fail for TIP, but should pass for SIP then you may update those tests.

You are to provide a CATCH2 parser unit test, named `SIPParserTest`, that thoroughly exercises the functionality of your parser. You are expected to follow the model of `TIPParserTest` and provide tests that: determine that syntactically and lexically correct programs are parsed correctly, determine that operator precedence is enforced in the structure of the parse tree, determine that lexically incorrect programs are rejected, and determine that syntactically incorrect programs are rejected. The unit test `ParserHelper` class has methods that you can use to construct such tests.

Deliverable 1 : Accessing the Project, Solution Expectations and Grading

To initiate work on this deliverable, join the classroom and assignment using the following link: [redacted]. You will be prompted to create a team, or join one. If you are creating the team, name the team the last names of the two members lowercase, separated by an underscore, in alphabetical order, e.g. Hossain and Phair should be hossain_phair. Once you join a team, a link to your repo will appear on the screen of the form [redacted], where team_name is your team name, e.g., hossain_phair. From here, you can clone and interact with the repository like any github repository where you have write access.

You are expected to modify the parser, add SIP parser unit testing, and to provide a brief overview of your work. The description of your work should in a file `SOLUTION1.md` that you add to the root directory of your project. You should briefly describe any tricky aspects of your solution, any design alternatives you considered and how you decided among them, and your approach to testing,

We will access the latest project commit prior to 5pm Eastern Time on Sept. 23, 2022 to assess your solution. You may continue your work on the project after that time, but it will not be considered for grading this deliverable. Grading will be based on the following rubric:

Quality of Solution (7pts) The completeness and correctness of your solution when considering both the SIP extensions and the preservation of TIP support.

Quality of Testing (4pts) The completeness of testing of your solution.

Quality of Description (1pt) The completeness and quality of the description of your solution.