Ada Compiler
CS355:Compiler Design

Anshu Avinash
anshuavi@iitk.ac.in

Pranjal Singh
spranjal@iitk.ac.in

Atique Firoz
atiquef@iitk.ac.in

Parth Tripathi
partht@iitk.ac.in

January 18, 2013

Ada is a strongly typed, modular, object oriented, concurrent, readable and expressible high-level computer programming language [1]. In this project we would be creating a compiler for a subset of Ada language. We would be implementing following features [2]:

- Type System: Ada's type system is governed by four principles: Strong typing, Static typing, Abstraction and Name equivalence.
  - type and subtypes: Creation of new types and subtypes with following features:
    - constrained and unconstrained types
    - dynamic types

  Following types would be in standard package:
  - Signed Integers
  - Unsigned Integers
  - Enumerations:
    - Operators: $<, <=, =, /=, >=, >$
    - Attributes: Pos, Val, Image, Value
    - Enumeration Literals: Character and Boolean as enumeration literals
    - subtype
  - Floating Point
  - Ordinary and Decimal Fixed Point
  - Array:
    - Allow creation of arrays with:
      - with known subrange
      - with unknown subranges
      - with aliased elements
    - Multi-Dimensional Arrays
    - Operations with Arrays: Assignment, Concatenate
    - Attributes: First, Last, Length, Range
    - Null Arrays
  - Record: A record is a composite type that groups one or more fields.
    Support for Null Record, Record with Values, Discriminated Record, Variant Record, Union, Tagged, Abstract Tagged, with Aliased Elements and Limited.
  - Access Type: Access types in Ada are what other languages call pointers. There are following Access types:
    - Pool Access

    - General Access: Access to Variable and Access to Constant
    - Anonymous Access
    - Access to subprogram

- Conditionals: These are blocks of code that will only execute if a particular expression (the condition) is true. They are categorized into:
  - **if-else**
  - **case:** The case statement is used to compare one specific variable against several constant expressions. If one of the several cases are not satisfied then the statement given in *others* is executed.
- Unconditionals: They let you change the flow of the program without a condition. They are categorized into:
  - **goto:** Transfers the control to the statement after the specified *label*.
  - **return:** It ends a function and returns to the calling procedure or function. For *procedures* just return is used whereas for *functions* return is followed by a *value*.
- Loops:Categorized into:
  - **for**
  - **while**
  - **until**
  - **exit**
  - **array:** This loop iterates over every element of the array specified.
- Input/Output: ADA has 5 independent libraries for Input and Output operations.
  - **Text:** It provides support for line and page layout but the standard is free form text.
  - **Direct:** It is used for random access files which contain only elements of one specific type.
  - **Sequential:** It can be used to choose between definite and indefinite element types but one has to read and write the elements one after the other.
  - **Storage:** It allows to store one element inside a memory buffer. The element needs to be a definite subtype.
  - **Stream:** It allows to mix objects from different element types in one sequential file.
- Exception: Ada has modules which raise an error when certain conditions are not satisfied and another module which does corresponding error-handling.
  - **Predefined:** They are included in *Standard* package. Some of them are: *Constraint_Error, Program_Error, Storage_Error, Tasking_Error*
  - **Input/Output:** These exceptions raised by subprograms of the predefined package *Ada.Text_IO*. Some of them are *End_Error, Data_Error,Mode_Error,Layout_Error*
  - **Raising exceptions:** The *raise* statement explicitly raises a specified exception.

References

[1] *Ada Information ClearingHouse: News and resources for Ada programming language.* URL: `http://www.adaic.org`.

[2] *Ada Programming: WikiBook.* URL: `en.wikibooks.org/wiki/Ada_Programming`.