

The Programming Language *Geizela* (Reference Manual)

1 Lexical Issues

1.1 Identifiers

An identifier is a sequence of letters, digits, underscores (_) and primes (') starting with a letter. Uppercase letters are distinguished from lowercase ones. In the following presentation, we use *id* to range over identifiers and *lab* to range over labels, which are either identifiers or sequences of digits.

1.2 Comments

There are three forms of comments:

- A comment may start with `*` and end with `*`.
- A comment may start with `//` and end with the end-of-line (EOL) symbol.
- A comment may start with `////` and end with the end-of-file (EOF) symbol.

2 Types

The grammar for types is given in Figure 1.

2.1 Built-in Types

The named types *bool*, *char*, *int*, and *string* are predefined.

2.2 Tuple Types

A tuple type is described as a sequence of types enclosed in `'()`. For instance, the following is an example of tuple type:

```
'(bool, char, '(int, string))
```

2.3 Record Types

A record type is described as a sequence of fields enclosed in `'{}`, where each field is described as *lab* : *ty*. For instance, the following is an example of record type:

```
'{1:bool, 2:char, 3:'(int, string)}
```

$atty ::= tyid$	type identifier
$\quad \quad \quad ' (tyseq)$	tuple type
$\quad \quad \quad ' \{ tyrow \}$	record type
$\quad \quad \quad \&atty$	reference type
$\quad \quad \quad @atty$	array type
$\quad \quad \quad (ty)$	
$ty ::= atty$	
$\quad \quad \quad atty \rightarrow ty$	function type: one argument
$\quad \quad \quad (tyseq) \rightarrow ty$	function type: multiple arguments
$tyseq ::= \epsilon$	
$\quad \quad \quad ty \{ , ty \}$	type sequence
$tyrow ::= \epsilon$	
$\quad \quad \quad lab : ty \{ , lab : ty \}$	type row

Figure 1: The grammar for types in Geizela

2.4 Reference Types

We use the symbol $\&$ followed by a type to denote a reference type. For instance, here is an example of reference type: $\& \ '(\text{int}, \text{int})$

2.5 Array Types

We use the symbol $@$ followed by a type to denote an array type. For instance, here is an example of array type: $@ \ '(\text{int}, *int)$

2.6 Function Types

As usual, we use \rightarrow to form function types. For instance, $\text{int} \rightarrow \text{int}$ is a function type for functions from integers to integers; $(\text{string}, \text{string}) \rightarrow \text{bool}$ is for functions that take two strings to produce a boolean value.

3 Expressions

4 Declarations

5 Standard Library

6 Sample Programs