Agda Reference Manual

Various Authors

February 22, 2012

1 Lexical Matters

1.1 Source Files

Agda code is written in plain text files consisting of Unicode characters encoded in UTF-8. The tokens that may appear in source files are literals (§1.3), names (§1.4), and keywords (§1.5).

Tokens are separated by whitespace and organised into blocks. Whitespace includes comments (§1.2). For more details about layout, see §1.6.

Agda programs are organised into modules, with one top-level module per file. For more details, see §5.

1.2 Comments

Agda has both single-line and multi-line comments.

A single-line comment is two hyphens (`--`) followed by a sequence of any characters except newline. The comment ends with a newline.

A multi-line comment is any sequence of characters delimited by `{-` and `-}`. Multi-line comments may be nested, and must be nested properly.

Comments may not appear in string literals. (Apparent comments in string literals are part of the string.)

1.3 Literals

Numbers

Characters A character literal is given by

`'`*c*`'`

where *c* is any character except backslash (`\`) or single-quote (`'`). The backslash literal is written `'\\'` and the single-quote literal `'\''`.

Strings A string literal is given by

`"`*s*`"`

where *s* is a sequence of normal or escaped characters characters. A normal character, in this context, is any Unicode character except backslash (`\`) or double-quote (`"`). Escaped characters and their meanings are listed below.

Character	Meaning
`\\`	backslash
`\"`	double-quote
`\n`	newline
`\r`	carriage return
`\t`	tab

(Unescaped newlines, carriage returns, and tabs may also occur in strings.)

1.4 Names

Name Parts A *name part* is a string of printable characters not containing any of the following characters.

```
_;."(){}@
```

(Unqualified) Names A *name* is is a sequence of one or more name parts separated by underscores (`_`).

Qualified Names A *qualified name* is a sequence of one or more unqualified names separated by dots (`.`).

1.5 Keywords

Keywords are name parts that have a special meaning and can only appear in certain contexts.

The following keywords may not appear in names (though they may appear as substrings of name parts).

`=`	`forall`	`private`
`|`	`hiding`	`public`
`->`	`import`	`quoteGoal`
`:`	`in`	`quoteTerm`
`?`	`infix`	`quote`
`\`	`infixl`	`record`
$\to$	`infixr`	`renaming`
$\forall$	`let`	`rewrite`
λ	`module`	`Set`n ($n \simeq [0-9]^*$)
`abstract`	`mutual`	`syntax`
`codata`	`open`	`unquote`
`constructor`	`postulate`	`using`
`data`	`primitive`	`where`
`field`	`Prop`	`with`

The following keywords may appear in names.

```
to
```

1.6 Layout

2 Type Theory

2.1 Core Syntax

2.2 Typing Rules

2.3 Normalisation

3 Extended Syntax

3.1 Mixfix Operators

3.2 Fixity Declarations

3.3 Implicit Arguments

3.4 Instance Arguments

4 Declarations

4.1 Data

4.2 Constants

4.3 Records

4.4 Postulates

5 Modules

6 Totality

7 Relevance

8 Practical Matters

8.1 Emacs Interface

8.2 Pragmas

8.3 Auto

8.4 Compilation

9 Standard Library