Algo 60
Version 8.11.1.10

December 31, 2023
1 Implementation

The “Algol 60” language for DrRacket implements the language defined by the “Revised Report on the Algorithmic Language Algol 60,” edited by Peter Naur.
2 Including Algol 60 Programs

Although Algol 60 is mainly provided as a DrRacket language, include-algol supports limited use of Algol 60 programs in larger programs.

```
(receive algol60/algol60)  package: algol60

(include-algol path-string)
```

Includes the Algol 60 program indicated by path-string as an expression in a Racket program. The included Algol 60 program is closed (i.e., it doesn’t see any bindings in the included context), and the result is always #<void>.

```
(literal-algol string ...)
```

Evaluates the Algol 60 program indicated by the literal strings as an expression in a Racket program. The Algol 60 program is closed (i.e., it doesn’t see any bindings in the included context), and the result is always #<void>.

This is generally useful when combined with the at-exp reader, e.g.,

```
#lang at-exp racket
@literal-algol{
  begin
    println ('hello world')
  end
}
```
3 Language

The DrRacket and include-algol implementation departs from the Algol 60 specification in the following minor ways:

- Strings are not permitted to contain nested quotes.
- Identifiers cannot contain whitespace.
- Argument separators are constrained to be identifiers (i.e., they cannot be keywords, and they cannot consist of multiple identifiers separated by whitespace.)
- Numbers containing exponents (using the “10” subscript) are not supported.

Identifiers and keywords are case-sensitive. The boldface/underlined keywords of the report are represented by the obvious character sequence, as are most operators. A few operators do not fit into ASCII, and they are mapped as follows:

```
times          *
quotient       div
exponential    ~
less or equal  <=
greater or equal  =>
not equal      !=
equivalence    ==
implication    =>
and             &
or              |
negation        !
```

In addition to the standard functions, the following output functions are supported:

```
prints(E)  prints the string E
printsln(E) prints the string E followed by a newline
printn(E)  prints the number E
printnln(E) prints the number E followed by a newline
```

A prompt in DrRacket’s interactions area accepts whole programs only for the Algol 60 language.