Nestable pragmat delimiters in Algol 68

GNU68-2026-002 (draft)

by Jose E. Marchesi

Copyright © 2026 Jose E. Marchesi.

You can redistribute and/or modify this document under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Foreword

The following specification has been released under the auspices of the GNU Algol 68 Working Group, and has been scrutinized to ensure that

a. it is strictly upwards-compatible with Algol 68,
b. it is consistent with the philosophy and orthogonal framework of the language, and
c. it fills a clearly discernible gap in the expressive power of that language.

The source of this document can be found at `https://git.sr.ht/~jemarch/gnu68`.

The informal description of this proposal introduces the proposed new language features, providing a rationale and usage examples.

The formal definition of this proposal uses the existing formalism and conventions of the Revised Report, and it is expressed as modifications to the Report.

Finally, the implementation notes of this proposal describe ways in which the features added by this specification can be implemented. No implementer should feel committed to do things as described there; the same language facilities may well be implementable in other ways, more suitable to specific implementations.

1 Informal Description

Pragmats in Algol 68

The Algol 68 strict language defines two styles for pragmats, each style using a different delimiter symbol:

`bold pragmat symbol`	**pragmat**
`style i pragmat symbol`	**pr**

In both cases the same symbol marks the beginning and end of the pragmat: a bold pragmat starts with **pragmat** and it must be ended by another **pragmat**; a `style i` pragmat starts with **pr** and ends with a **pr**. These delimiters are not ideal, for several reasons.

First, using the same symbol for starting and ending a pragmat makes it necessary to inspect the whole source in order to determine, given a delimiter, whether it begins or finishes the construct. For programmers this can be confusing and error prone, for programs such as text editors wanting to highlight a pragmat, or to skip over it, this can be very inefficient.

Pragmats using that kind of delimiter cannot be nested. This may not seem like a significant limitation at first. Other constructs, such as comments, are often nested, but not annotations for the compiler. However, the ability of doing so may be useful in order to combine several pragmats without having to resort to other language constructs.

Nestable pragmats in Algol 68

The shortcomings described in the previous section motivate the addition of nestable pragmat delimiters to Algol 68.

Representations of pragmats are defined for the *bold* and *i* styles, both of which use symmetric delimiters:

`bold pragmat begin symbol`	**order**
`bold pragmat end symbol`	**redro**
`style i pragmat begin symbol`	`[[`
`style i pragmat end symbol`	`]]`

In Algol 68 the contents of several contiguous pragmats (ignoring possibly intermixed comments) are combined via simple concatenation into a string that is then interpreted as a single pragmat.

For example, the following pragmats:

```
[[un]] [[used]]
```

are equivalent to:

```
[[unused]]
```

When nested pragmats are found, they are expanded into the contents of the containing pragmat, including both delimiters and contents. This makes it possible for pragmats to operate on other pragmats. For example, the following pragmats may be a no-op:

```
[[unused]] [[ignore [[unused]]]]
```

2 Formal Description

New symbols have been invented for opening and closing pragmats:

```
9.4.1.h
  bold pragmat begin symbol                  ORDER
  bold pragmat end symbol                    REDRO
  style i pragmat begin symbol               [[
  style i pragmat end symbol                 ]]
```

Syntax of pragments is expanded:

```
9.2.1

b) PRAGMENT{a} : ... ; STYLE nestable pragmat{d}.
d) STYLE nestable pragmat{b} :
     STYLE pragmat begin symbol{94h,-},
       STYLE PRAGMENT item{c} sequence option,
       STYLE pragmat end symbol{94h,-}.
```

3 Implementation Notes

The new pragmats are introduced as an addition to the existing pragmats, requiring rather straightforward changes to the lexical analyzer.

It is recommended for implementations offering stropping regimes based on reserved words to only recognize pragmats with symmetric delimiters, and of these only the brief style.

Since pragmats can be concatenated, it is advisable to compilers to use a pragmat language in which such concatenation will lead to a valid sequence of orders.