Wirth syntax notation

Wirth syntax notation (**WSN**) is a metasyntax, that is, a formal way to describe formal languages. Originally proposed by Niklaus Wirth in 1977 as an alternative to Backus–Naur form (BNF). It has several advantages over BNF in that it contains an explicit iteration construct, and it avoids the use of an explicit symbol for the empty string (such as <empty> or ε).[1]

WSN has been used in several international standards, starting with ISO 10303-21.[2] It was also used to define the syntax of EXPRESS, the data modelling language of STEP.

WSN defined in itself

```
SYNTAX     = { PRODUCTION } .
PRODUCTION = IDENTIFIER "=" EXPRESSION "." .
EXPRESSION = TERM { "|" TERM } .
TERM       = FACTOR { FACTOR } .
FACTOR     = IDENTIFIER
           | LITERAL
           | "[" EXPRESSION "]"
           | "(" EXPRESSION ")"
           | "{" EXPRESSION "}" .
IDENTIFIER = letter { letter } .
LITERAL    = """" character { character } """" .
```

The equals sign indicates a production. The element on the left is defined to be the combination of elements on the right. A production is terminated by a full stop (period).

- Repetition is denoted by curly brackets, *e.g.,* **{a}** stands for **ε | a | aa | aaa | ...**.
- Optionality is expressed by square brackets, *e.g.,* **[a]b** stands for **ab | b**.
- Parentheses serve for groupings, *e.g.,* **(a|b)c** stands for **ac | bc**.

We take these concepts for granted today, but they were novel and even controversial in 1977. Wirth later incorporated some of the concepts (with a different syntax and notation) into extended Backus–Naur form.

Notice that `letter` and `character` are left undefined. This is because numeric characters (digits 0 through 9) may be included in both definitions or excluded from one, depending on the language being defined, *e.g.*:

```
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" .
upper-case = "A" | "B" | … | "Y" | "Z" .
lower-case = "a" | "b" | … | "y" | "z" .
letter = upper-case | lower-case .
```

If `character` goes on to include `digit` and other printable ASCII characters, then it diverges even more from `letter`, which one can assume does not include the digit characters or any of the special (non-alphanumeric) characters.

Another example

The syntax of BNF can be represented with WSN as follows, based on translating the BNF example of itself:

```
syntax          = rule [ syntax ] .
rule            = opt-whitespace "<" rule-name ">" opt-whitespace "::="
                  opt-whitespace expression line-end .
opt-whitespace = { " " } .
expression      = list [ "|" expression ] .
line-end        = opt-whitespace EOL | line-end line-end .
list            = term [ opt-whitespace list ] .
term            = literal | "<" rule-name ">" .
literal         = """" text """" | "'" text "'" .
```

This definition appears overcomplicated because the concept of "optional whitespace" must be explicitly defined in BNF, but it is implicit in WSN. Even in this example, `text` is left undefined, but it is assumed to mean "`ASCII-character { ASCII-character }`". (`EOL` is also left undefined.) Notice how the kludge `"<" rule-name ">"` has been used twice because `text` was not explicitly defined.

One of the problems with BNF which this example illustrates is that by allowing both single-quote and double-quote characters to be used for a `literal`, there is an added potential for human error in attempting to create a machine-readable syntax. One of the concepts migrated to later metasyntaxes was the idea that giving the user multiple choices made it harder to write parsers for grammars defined by the syntax, so computer languages in general have become more restrictive in how a *quoted-literal* is defined.

References

1. Wirth, Niklaus (November 1977). "What Can We Do about the Unnecessary Diversity of Notations for Syntax Definitions?". *Communications of the ACM*. **20** (11): 822–823. doi:10.1145/359863.359883 (https://doi.org/10.1145%2F359863.359883).
2. "ISO 10303-21, Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure" (http://www.iso.org/iso/search/extendedsearchstandards.htm?from_js=true&type=adv&displayResult=true&published=on&formKeyword=&title=on&description=on&formISO_number=10303&formPart_number=21&formDocument_type=ALL&formSupp_type=&formICS=&formStage_code=&repost=1&formDate_stage=0&formOther_date=&formCommittee=&formSub_committee=&formSubmit=Search). International Organization for Standardization. 2002-01-24.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Wirth_syntax_notation&oldid=956147728"

This page was last edited on 11 May 2020, at 19:01 (UTC).