NAME

vwg − VW-grammar sentence generator

SYNOPSIS

vwg [**-d**... **-l**... **-L**] [file]

DESCRIPTION

The program *vwg* reads a two-level (van Wijngaarden) grammar and produces **all** its terminal productions; a production is "terminal" if it contains no metanotions and no hyperrule applies. If there are infinitely many of these, this will take infinite time, but given that, they'll all be there. The production process is "fair" in the sense that any given terminal production will appear in finite time.

The grammar of the input is:

input: metarule-sequence, point-symbol, hyperrule-sequence, point-symbol,
starting-production-sequence, point-symbol.

metarule: metaletter, colon-symbol, colon-symbol, meta-alternative-series, point-symbol.
meta-alternative: letter-sequence.
letter: metaletter; small-mark.

hyperrule: hypernotion, colon-symbol, hyper-right-hand-side, point-symbol.
hyper-right-hand-side: hyperalternative-series; plus-symbol; minus-symbol.
hypernotion: hyperletter-sequence.
hyperletter: primed-metaletter; small-mark.
primed-metaletter: metaletter; primed-metaletter, prime-symbol.
hyperalternative: hypernotion-list.

starting-production: hyperalternative, point-symbol.

Here a "series" is separated by semicolon-symbols, a "list" by comma-symbols and a "sequence" by nothing, as usual in two-level grammars. A "metaletter" is a capital letter; a "small-mark" is any printable character except a capital letter, a point-symbol, semicolon-symbol, a comma-symbol, a colon-symbol or a prime-symbol. Comment starts at a hash-mark (#) and continues until end-of-line. Layout is allowed everywhere.

Blind alleys and symbol rules

The Algol 68 Report implies that the specific terminal production of the start notion that it is interested in is a sequence of "symbols"; any other notion is a blind alley and any terminal production containing one is to be rejected. It then says (1.1.3.1.f) that a "symbol" is a notion ending in `symbol`; this (extra-grammatical) rule is a "positive symbol rule".

The program *vwg* defines a terminal production as a sequence of notions that do not contain metanotions and do not match any hyperrule left-hand side; that is, notions that cannot continue to produce any more. Such terminal productions often contain many blind alleys and to single out the "real" terminal productions of the grammar, positive and/or negative symbol rules can be added to the hyperrule section of the grammar.

Symbol rules look like hyperrules, and have a `+` or `-` as their only hyperalternative. A terminal production is reported if all its notions match the positive symbol rule, if present, and no notion matches a negative symbol rule, if present. Only one positive symbol rule is allowed. Negative symbol rules can increase efficiency by elimination productions with blind alleys early. Examples for the Algol 68 Report (Section 1.3.1) could be (in the notation of the A68 Report)

```
ALPHA symbol : + .
where false : - .
unless true : - .
```

Options

There are two groups of command line options, one starting with −**l** and one starting with −**d.**

The −**l**-option sets the level of error messages:

W suppress warnings;

E suppress warnings and error messages.

The −**d**-option is used for debugging, mainly of the program itself, but occasionally for a grammar; it may be followed by a combination of the following letters:

G display the grammar;

I display the input;

M show memory requests;

P print intermediate results;

S display substitutions;

T trace the parser.

In principle *vwg* cannot handle grammars with left-recursive bound metanotions (a metanotion is bound when it occurs in the left hand side of a hyperrule), but the −**L**-option causes the program to use a cancellation parser that can handle left recursion, indirect (hidden) left recursion, but not mutual recursion.

Example:

```
# A grammar which produces aˆn bˆn cˆn for all n
# The metarules:
N :: n N ; .
A :: a ; b ; c .
.
# the hyperrules:
NnA : A-symbol, NA.
A : .
.
# The starting symbol(s)
Na, Nb, Nc.
.
```

AUTHOR

Dick Grune, Mathematisch Centrum, Amsterdam; dick@dickgune.com.

BUGS

Bound metanotions cannot be left-recursive, but free ones can. See, however, option **-L**.

Metanotions can only be one letter long, but the restriction on the number of primes on a metanotion has been lifted.

The debugging options produce tons of output.