

Jan 16, 04 10:25

stdin

Page 1/3

Vpf  
NAME vpf(1)

vpf - View Parse Forest - a parse forest browser

SYNOPSIS  
vpf [ file ]

## DESCRIPTION

vpf is a graphical browser for parse forests with or without feature structures. It is similar to the *Xmfed* program which is part of the *Alep* grammar development environment. It has no editing functionality, however, and the file format (see FILE FORMAT) is slightly different.

When vpf is started, it displays the first parse forest from the input file (or from standard input if no argument was specified). vpf is a mouse-controlled program. When the middle button is pressed, vpf will display the previous sentence. When the right button is pressed, vpf will display the next sentence. The same functionality is provided by the "<" and ">" buttons. The number of the current sentence is displayed beneath these buttons. When the mouse pointer is moved over the feature structure, the selected part of the feature structure will be highlighted. This feature is turned off if the *highlighting* button is clicked. Pressing the left button of the mouse will implode this structure (or explode it if it was previously imploded). Imploded parts of the feature structure are represented by a small crossed box.

Structure sharing is represented by tags (small black boxes). Shared structures are only displayed at one location at a time and a tag is displayed at the other locations. Pressing the middle button while a tag is selected will cause the corresponding shared structure to be now displayed at this location.

## FILE FORMATS

The input file contains a sequence of parse forests. The simplified definition of the parse forest format in Backus-Naur form is:

```

NLIST: N NLIST
      | <empty>
N:     IFLAG N
      | HFLAG N
      | DFLAG N
      | RFLAG
      | '(' N N NLIST ')'
      | '[' NLIST '['
      | '{' NLIST '}'
      | '|' NLIST '|'

```

Jan 16, 04 10:25

stdin

Page

SunOS 5.6 Last change: October 1999 1  
Vpf vpf(1)

```

| '<' NLIST '>'
| STRING '=' N
| STRING

```

The tokens are recognised by regular expressions:

```

IFLAG: #i
DFLAG: #[0-9]+=
RFLAG: #[0-9]+
STRING: "([^\]|(\.))*"
        '([^\]|(\.))*'
        ([/A-Za-z0-9_$$%?;:.-]|(\.))+

```

An IFLAG marks the following node as imploded. An HFLAG marks a node as hidden (invisible). A DFLAG assigns a tag (number) to the following node and an RFLAG creates a link to the node with the respective tag. An X node with two coreferring Y daughters is created by:

(X #1=Y #1)

Nodes between parentheses are printed as a tree. Nodes between curly brackets are printed in horizontal order enclosed in curly brackets. Nodes between square brackets are printed in vertical order enclosed in square brackets. Nodes between '^' and '\$' and between angle brackets are printed without brackets in horizontal order and vertical order, respectively. A sequence of STRING '=' N triples enclosed by square brackets is printed as an aligned stack of feature value pairs.

## EXIT STATUS

vpf returns 0 unless some error occurs.

## SEE ALSO

vpf

## AUTHOR

Helmut Schmid (email: schmid@ims.uni-stuttgart.de)

