Name

NASSI – Typeset Nassi-Schneiderman diagrams in LaTeX

Date

September 10, 1996

Synopsis

`\STRUCT{struct name}{structure purpose}{statements}`

For a description of the statements see section Macros

Description

The `Nassi` macros enable the user to create Nassi-Schneiderman diagrams in a LaTeX document. The macros can be used in any type of document, within all the standard LaTeX environments.
An NS-diagram in this context is described in macros resembling a pseudo code. The diagram itself is defined as a structure and is build from statements. Macros are available for all standard programming statements (if, while, action etc.).
The Language Sensitive Editor for LaTeX, LSA, is extended to support the creation of NS-diagrams with the `Nassi` macros.
The macros needed to create a Nassi-Schneiderman diagram in your LaTeX document will be made available to you when you include the option `Nassi` in the `\documentstyle` command at the start of your LaTeX file, as in the following example:

```
\documentstyle[11pt,Nassi]{article}
```

An NS-diagram in the context of the `Nassi` macros is called a structure. The drawing of the diagram is based on units. A unit can be seen as the box in which an action is described. Both the structure and the unit have an attribute, the width. The layout of the diagram is determined from the value of these attributes. Both can be changed by the user through macros which must be used before the actual structure is described. They have the following effect:

- If both attributes have their default values, the unit width is 60 points (there are 72 points to an inch). The width of the structure is determined by the number of units on one line, while each unit has a minimum width of 60 points. The maximum width of a unit is the structure width.
- If the unit width is specified by the user the same rules apply but the minimum width of a unit is the specified width.
- If the structure width is specified by the user, the diagram will have this width, while the width of each unit is determined depending on their contents and the number of units on one line. The maximum unit width is still the structure width.

It is recommended that users do not change both attributes for the same structure.
By default the text in the diagrams is typeset in the same size as the text in the document. This will be to large. Before starting a structure it is adviced to set the desired text size. For the default settings of structure and unit width `\scriptsize` is recommended.
When you run into errors like `overfull hbox[]` you may either have to use a smaller text size like `\tiny` or give a proper hyphenation directions for the indicated text.
Do not forget to reset the text size to `normalsize` after the structure.

Macros

The available macros fall into four categories:

1. Width settings of structure and unit. They must appear before the structure.
2. Definition of the structure itself.
3. Statements. They must appear within the structure definition.
4. Options. They may appear before as well as within the structure definition.

Width setting macros are:

- `\unitwidth=size`
  Specifies the minimum width of a unit. Size must be given as a standard LaTeX length.
- `\nassiwidth=size`
  Specifies the total width of the diagram. Size must be given as a standard LaTeX length. For a diagram with the same width as the normal text the command `\nassiwidth=\textwidth` can be used.

Structure definition macro is:

- `\STRUCT{struct-name}{struct-description}{statements}%`
  Specifies the complete diagram. The structure name and description, which can be a short description of its purpose, will be put above the actual diagram. The statements of the structure, described below, must be specified as part of the `\STRUCT` command.

Statement macros are:

- `\ACTION{action}%`
  Specifies a normal statement.
- `\PROC{proc-name}{proc-description}%`
  Specifies a special kind of action, a procedure call. The proc-name will be put behind the proc-description text in brackets.
- `\ACCEPT{entry-name}{statements}\ENDACCEPT%`
  Specifies the `ADA` accept statement. The entry-name will be followed by a separate box containing the statements.

  This statement is not available for flow structures.

- `\IF{condition}\THEN{statements}\ELSE{statements}\ENDIF%`
  Specifies an if statement with a condition and a collection of statements in the `THEN` and the `ELSE` part. Neither the `THEN` nor the `ELSE` part is optional. If one of them is not needed an empty statement, "`{}`", must be used.
- `\REPEAT{statements}\UNTIL{test}%`
  Specifies a loop statement with a collection of statements and an end-condition.
- `\WHILE{condition}{statements}\ENDWHILE%`
  Specifies a loop statement with a start-condition and a collection of statements.
- `\CASE{case-item}{when-statements}\ENDCASE%`
  Specifies a case statement with a case-item and a collection of when-statements. The case statement can also be used to represent the `ADA` select statement by specifying `select` as the case-item.
- `\WHEN{condition}{statements}%`
  Part of a case statement. Specifies a condition and a collection of statements. When the case statement is used as an `ADA` select statement the condition represents the `gard`, `delay` or `else` part.

Options are:

- `\setiftext{left-hand}{right-hand}%`
  Specifies the text to be put on the left-hand and right-hand sides of the condition of an if statement. Defaults are "Y" and "N" for the left-hand and the right-hand respectively. The scope for these settings depends on the position of the command within the document.
  - If the command occurs before a `\STRUCT` statement, the settings remain valid for all `\IF` statements in all following structures until they are reset by a `\setiftext` command with the same scope. (Remember to put the `\setiftext` command **after** any font size changing commands).
  - If the command occurs inside a `\STRUCT` statement but outside any other commands, the settings will remain valid for all `\IF` statements in the same structure until they are reset by a `\setiftext` command with the same scope.
  - If the command occurs inside a command within a structure, the settings will remain valid for all `\IF` statements within the command (for instance an if statement within the then or else part of another if statement) until they are reset by a `\setiftext` command with the same scope.

  It is allowed to 'nest' the `\setiftext` commands, thus changing the settings for a particular part of a structure or a document.
- `\underlinewhentrue%` and `\underlinewhenfalse%`
  Specifies whether a line is drawn underneath an `\IF`, `\WHILE`, `\REPEAT`, `\ACCEPT` or a nested `\CASE` statement. To be compatible with previous releases of Nassi the default setting is `\underlinewhenfalse%`. The scope for this settings depends on the position of the command within the document. See `\setiftext` for more information.
  This option is not available for flow structures.

Note that each line within the structure, including the command itself, *must* be concluded by a %.
The text appearing (between the braces) in the structure definition and in all statements, including the `\setiftext` macro is free but must adhere to the standard LaTeX syntax. Every text is considered a paragraph in the LaTeX context.

Files

The file `TEX$INPUTS:NASSI.STY` specifies the Nassi-Schneiderman macros.

See also

The pages on LaTeX, LSA and FLOW in this section of the manual.

Bugs

- The `Nassi` option is mutually exclusive with the `Flow` option because the same macro names are used.
- `Nassi` has a limit to the number of statement macro's in one structure definition. For example a maximum of 26 IF macro's can be used. If one of these limits is reached, LaTeX shows some curious error messages, like "You cannot use = as a prefix", which have no relation at all to the real problem.
- Within the `\ACCEPT` statement part at least one statement is required. As dummy the `\ACTION{null}` is recommended.
- If `\underlinewhentrue%` is specified and the concerning `\WHEN` statement is the tallest of all `\WHEN` statements in the `\CASE` statement a double line will show. By specifying `\underlinewhenfalse%` at the end of this `\WHEN` statement, this problem can be prevented.

Example

The following is an example of a structure with all possible statements. The resulting diagram is shown below.

```
\scriptsize
\STRUCT{structure name}{structure purpose}{%
  \ACTION{initial statement}%
  \PROC{proc name}{procedure purpose}%
  \ACCEPT{entry name}{%
    \ACTION{critical part}%
  }%
  \ENDACCEPT%
  \IF{condition to test}%
  \THEN{%
    \ACTION{true action 1}%
    \ACTION{true action 2}%
```

```
    }%
    \ELSE{%
      \ACTION{false action}%
    }%
    \ENDIF%
    \REPEAT{%
      \ACTION{statement to repeat}%
    }%
    \UNTIL{end condition}%
    \WHILE{start condition}{%
      \ACTION{statement to do}%
    }%
    \ENDWHILE%
    \CASE{case item}{%
      \WHEN{condition 1}{%
        \ACTION{statement to do}%
      }%
      \WHEN{condition 2}{%
        \ACTION{statement 1 to do}%
        \ACTION{statement 2 to do}%
      }%
      \WHEN{condition 3}{%
        \ACTION{statement to do}%
      }%
    }%
    \ENDCASE%
    \CASE{select}{%
      \WHEN{gard 1}{%
        \ACCEPT{entry name 1}{%
          \ACTION{statement to do}%
        }%
        \ENDACCEPT%
      }%
      \WHEN{}{%
        \ACCEPT{entry name 2}{%
          \ACTION{critical statement to do}%
        }%
        \ENDACCEPT%
        \ACTION{non critical statement to do}%
      }%
      \WHEN{}{%
        \ACTION{delay 10.0}%
      }%
      \WHEN{else}{%
        \ACTION{else actions}%
      }%
    }%
    \ENDCASE%
  }%
  \normalsize
```

structure name — structure purpose