

## Remarks on Visualizing BNF Grammars using Interactive SVG

Visualization might raise critical questions. Here are some of them.

- is visualization more comprehensive than written text?
- is it precise?
- is it intuitive?
- does its interaction provide a better understanding?
- is it worthwhile (cost effectiveness)?
- is it more than a toy?
- what are the limits?
- is the technical base – Javascript, HTML plus embedded SVG - sufficient?

During the presentation of [interactive BNF diagrams](#) some of these questions will be answered.

Here is a short story instead of an abstract:

### EBNF

```
<alter_index> ::= "ALTER" "INDEX" <index_name> <on_clause> <op_clause>
  <on_clause> ::= "ON" <table_name> | ""
  <op_clause> ::= "ENABLE" | "DISABLE" | "INIT" "USAGE"

<alter_index> ::=
  ALTER INDEX <index name> [ON <table name>] {ENABLE | DISABLE | INIT USAGE}
```

Fig. 1: BNF as text versus EBNF in HTML

BNF (Backus Naur Form) is a meta language to define exactly the grammar of a context-free language. Often parsers are derived from the BNF. Most of the time an extended form (EBNF) is used for this purpose. The metalanguage can easily be adopted. It needs one page of explanation.

In brief: An EBNF grammar consists of production rules. A rule starts with its name followed by an assignment "::<=". The rule consists of clauses: Pure text - terminals, references to other rules - nonterminals, options enclosed in brackets "[ ]", alternatives separated by bars "|". repetitions indicated by three dots "...". nameless rules in curly braces "{}". That's all one has to know of the metalanguage used here.

EBNF is also used to document a language. Many times you will find it in the appendix of a handbook. And there it stays unused because it consists of rules in rules. This nesting forces you to thumb through the appendix. At the end you do not know where you came from. This is a big drawback.

### Highlighting and Links

Highlighting of metacharacters distinguishes them from language characters, no escapes or quotes are necessary any more. It helps if the grammar is in HTML format with links to rules. Then the thumb is replaced by the mouse. It's faster but not better.

## Syntax Diagrams

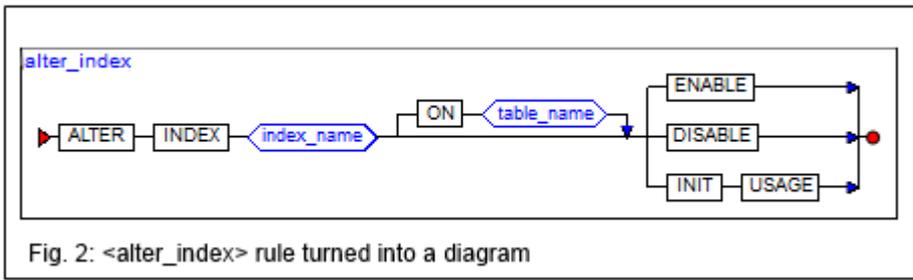


Fig. 2: <alter\_index> rule turned into a diagram

The transformation of a syntax rule into a diagram, known as railroad diagrams, improves the comprehension of a BNF notation. It is obviously easier to grasp the format of the diagram rule without losing precision. This is still a static approach. It could be part of a manual. See e.g. Oracle's DB documentation.

## Drop Down Diagrams

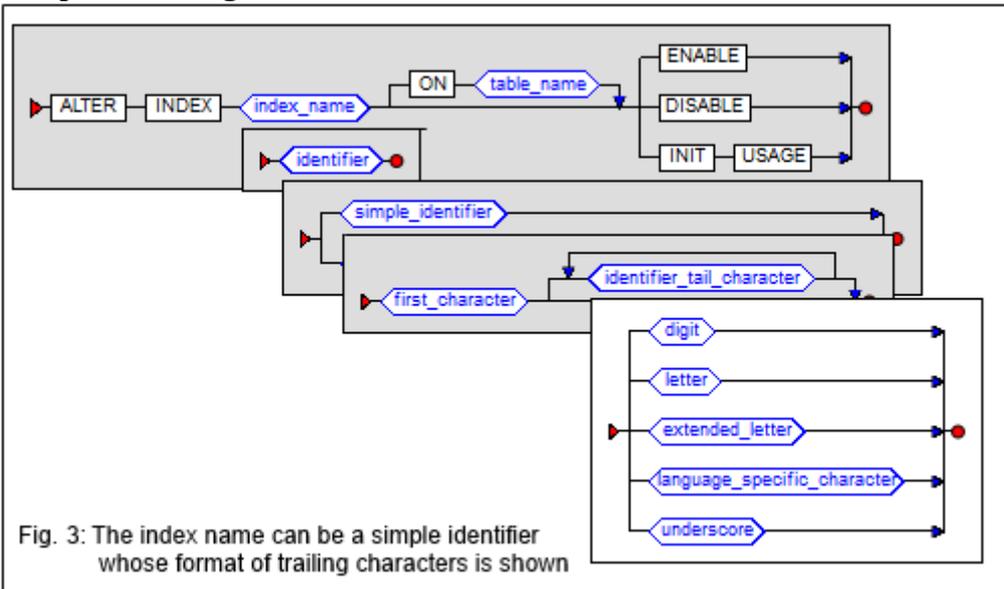


Fig. 3: The index name can be a simple identifier whose format of trailing characters is shown

Dynamics come in with the interactive exploded view of the subrules if they become clickable. One might call them drop down diagrams. The rules are visualized in their individual context. This approach could be implemented by using pictures of each rule in a grammar. The clicked subrule is placed directly under the rule reference.

## Inner Expansion of Diagrams

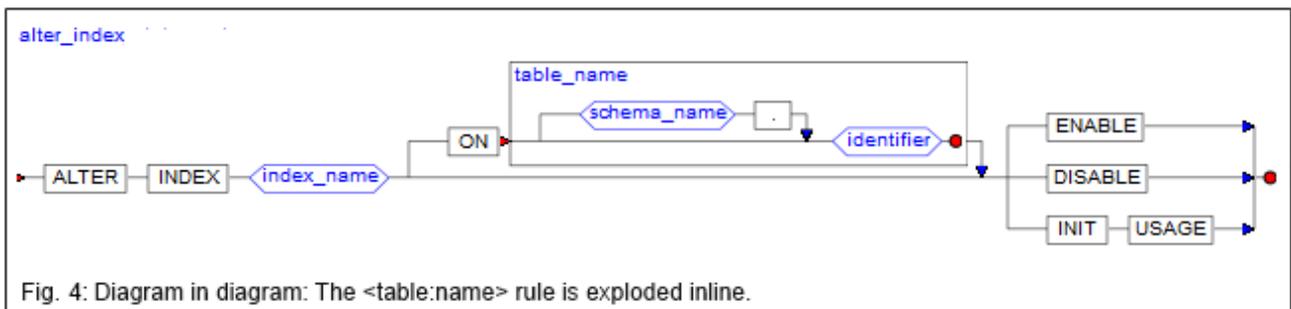


Fig. 4: Diagram in diagram: The <table:name> rule is exploded inline.

The best presentation will be an inline expansion of a rule. But here we stumble against the limits of conventional HTML. Because of the multitudinous combination of rules the diagrams have to be redrawn. The tool of choice for doing that is SVG.

One can see at a glance how the SQL statement should be written. Of course, nested diagrams can be drawn to arbitrary depth.

## Interactive Reduction of Complexity

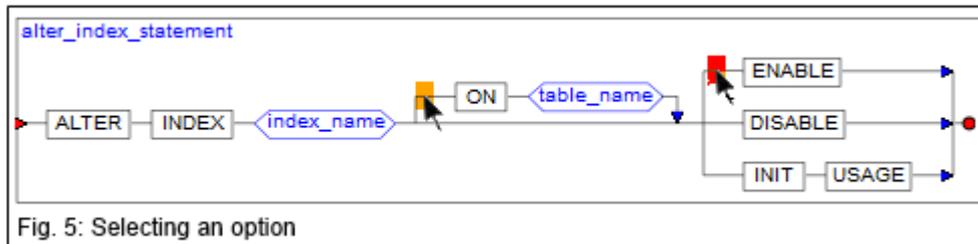


Fig. 5: Selecting an option

Instead of exploding rules, they can be collapsed by selecting alternatives or dropping options. As a reminder of a collapsed clause a colored rectangle will appear. There are hidden rectangles in front of options and alternatives. If the mouse moves over it, they will appear. If clicked, the option or alternative will be selected.



Fig. 6: With dropped option and selected alternative

An option can either be dropped or selected.



Fig. 7: With selected option and selected alternative

The colored rectangles – ocher for option and red for alternatives - are reminders of the reduction. If clicked, the original will be redrawn.

If only one line of clauses is left, a statement template will be printed, like:

```
ALTER INDEX <index_name> ON <table_name> DISABLE
```

## Redundancy Reduction

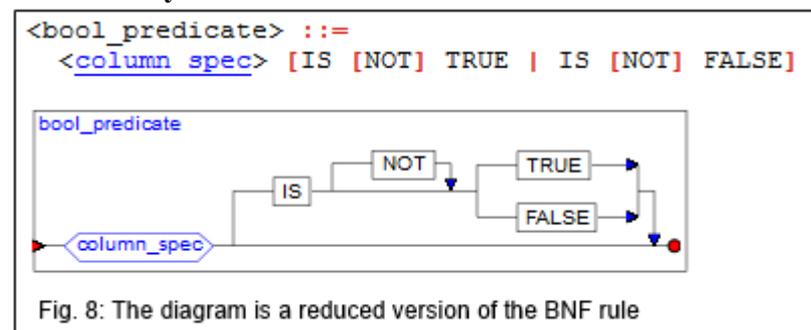


Fig. 8: The diagram is a reduced version of the BNF rule

A BNF and even an EBNF notation can be lengthy. Therefore an algorithm has been developed to reduce redundancy. Depending on the specified depth rules can be automatically reduced to their lowest term.

## Discussions

The above questions will be discussed during the interactive presentation. Especially, whether such an application is worthwhile and not just another toy.

The experience with published grammar diagrams of a licencefree database will be shown.

Some hundred BNFs on various languages were collected. The intention is to convert them to a standard BNF format, generate the diagrams and make them available to the public.