



Syntax Diagrams

...if you can't avoid them...

Frank Wegmann, DITA-OT Day 2024, Helsinki



You don't want to hear this...

Extend your coffee / tea break instead!

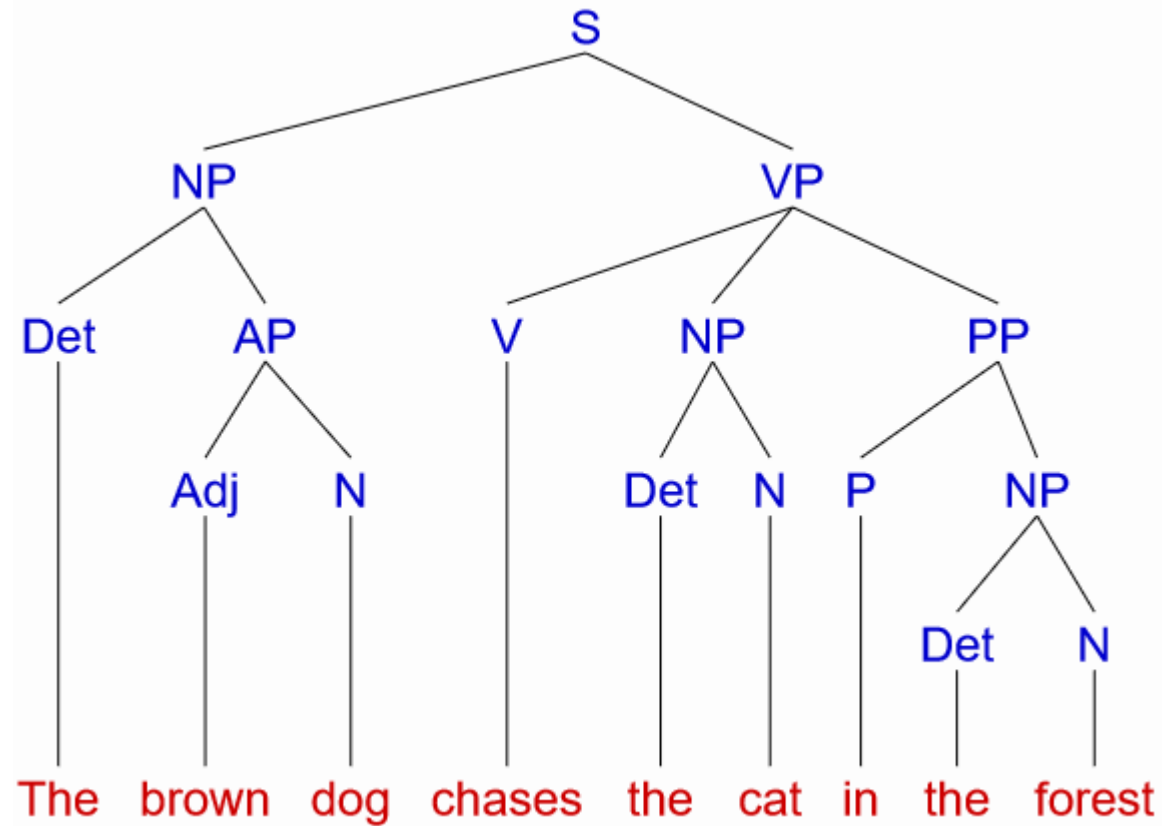


Kiitos!

Syntax Trees for Natural Languages

Phrase structure grammar (N. Chomsky)

S	Sentence
NP	Noun Phrase
AP	Adjective Phrase
VP	Verb Phrase
PP	Prepositional Phrase
Det	Determiner
N	Noun
Adj	Adjective
P	Preposition
V	Verb



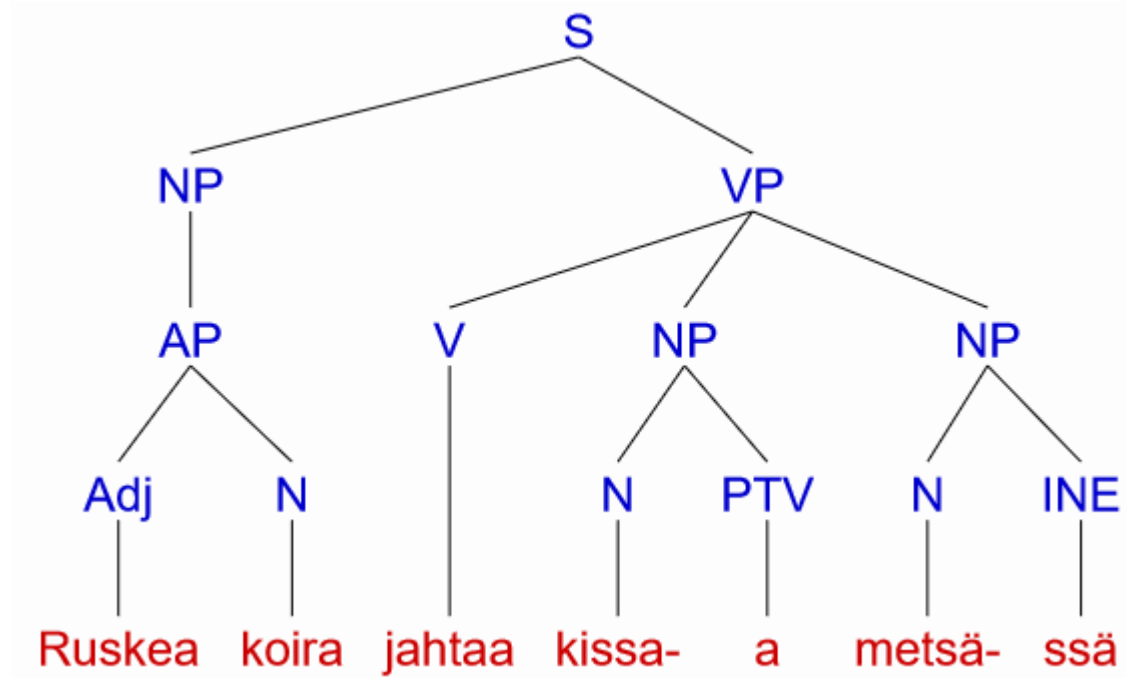
Grammatical categories

Lexical entries

Syntax Trees for Natural Languages

Another language – another grammar...

S	Sentence
NP	Noun Phrase
AP	Adjective Phrase
VP	Verb Phrase
N	Noun
Adj	Adjective
V	Verb
PTV	Partitive
INE	Inessive



Grammatical categories

Lexical entries

Backus-Naur Form

Grammar rules

Set of productions

$\langle S \rangle ::= \langle NP \rangle \langle VP \rangle$

$\langle NP \rangle ::= \langle Det \rangle \langle AP \rangle$

$\langle NP \rangle ::= \langle Det \rangle \langle N \rangle$

$\langle AP \rangle ::= \langle Adj \rangle \langle N \rangle$

$\langle VP \rangle ::= \langle V \rangle$

$\langle VP \rangle ::= \langle V \rangle \langle NP \rangle$

$\langle VP \rangle ::= \langle V \rangle \langle NP \rangle \langle PP \rangle$

$\langle PP \rangle ::= \langle P \rangle \langle NP \rangle$

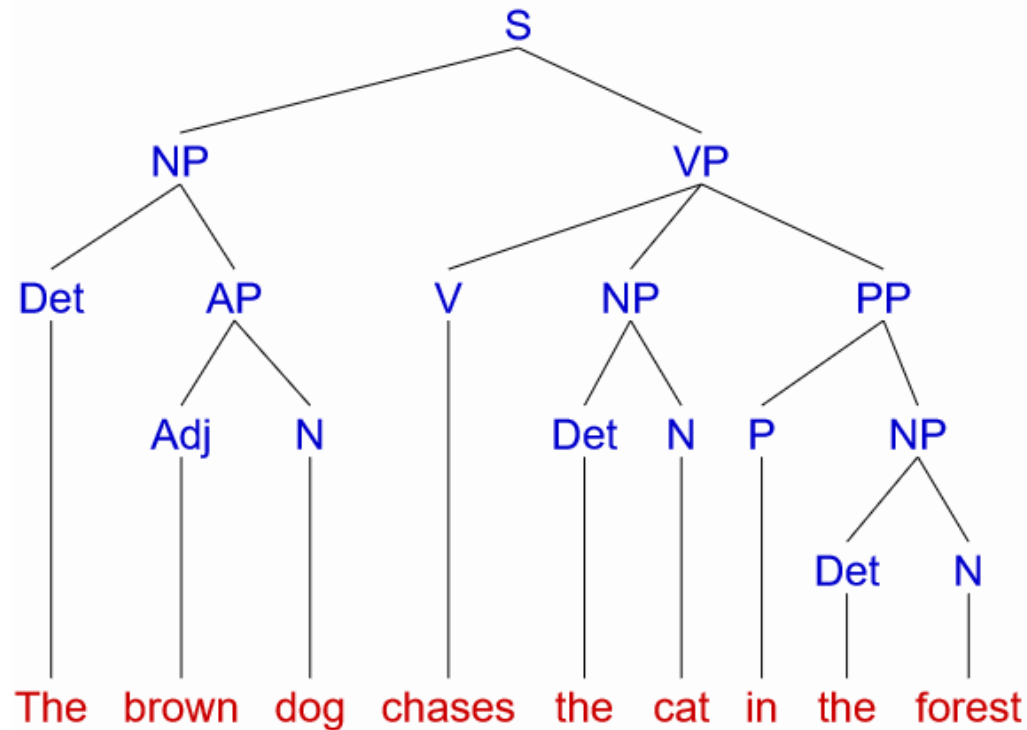
$\langle N \rangle ::= \text{"dog"} \mid \text{"cat"} \mid \text{"forest"}$

$\langle Det \rangle ::= \text{"a"} \mid \text{"an"} \mid \text{"the"}$

$\langle Adj \rangle ::= \text{"brown"}$

$\langle P \rangle ::= \text{"in"}$

$\langle V \rangle ::= \text{"chase"}$



Start symbol (S)

Non-terminal symbols

Terminal symbols
(tokens)

“Railroad Diagrams” from EBNF

Software AG Tamino XML Server documentation (XQuery reference)

FLWORExpr ::= (ForClause | LetClause)+ WhereClause? OrderByClause? 'return' Expr

Tamino XML Server 10.15 | XQuery 4 Reference Guide | Syntax Constructs in Alphabetical Order | FLWORExpr

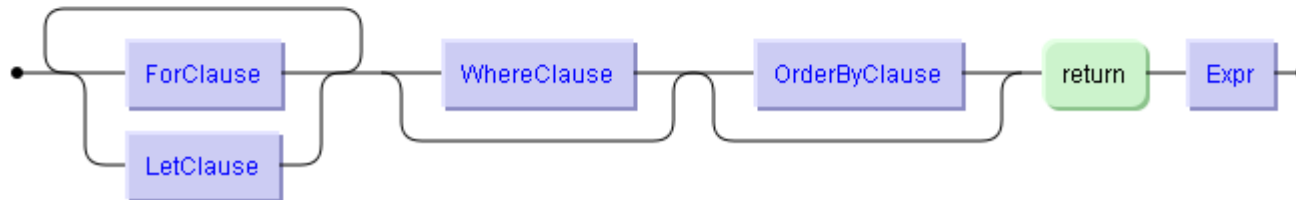
FLWORExpr

Iterate over sequences of items.

[[Syntax](#) | [Description](#) | [Examples](#) | [Related Syntax Constructs](#)]

Syntax

FLWORExpr

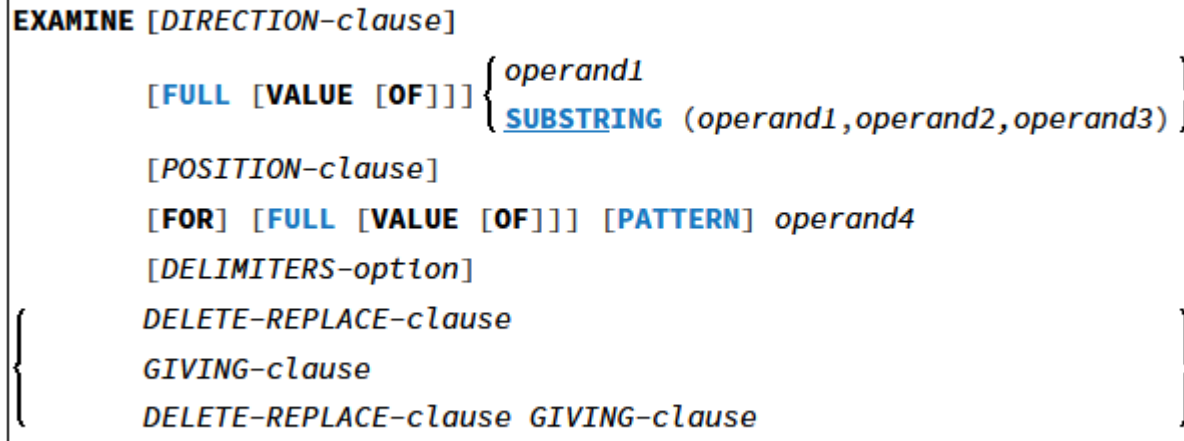


<https://documentation.softwareag.com/webmethods/tamino/ins10-15/webhelp/ins-webhelp/xqueryref/ref-FLWORExpr.htm#ref-FLWORExpr>

Syntax Diagram – “UNIX Style”

Software AG Natural for Mainframes documentation (Natural language reference)

Syntax 1 - EXAMINE



For an explanation of the symbols used in the syntax diagram, see [Syntax Symbols](#).

<https://documentation.softwareag.com/natmf/9.2.2/en/webhelp/natmf-webhelp/sm/examine.htm>

```
<table frame="none" role="syntax">
<tbody>
<tr>
<td colspan="2" style="text-align: center; vertical-align: middle;">
<math display="block">
\left. \begin{array}{l}
\text{[FULL [VALUE [OF]]] } \\
\text{SUBSTRING (operand1, operand2, operand3)}
\end{array} \right\}



<tr>
<td colspan="2" style="text-align: center; vertical-align: middle;">
<math display="block">
\left. \begin{array}{l}
\text{DELETE-REPLACE-clause} \\
\text{GIVING-clause} \\
\text{DELETE-REPLACE-clause GIVING-clause}
\end{array} \right\}



</tbody>
</table>
```

Representation of Syntax in DITA

- Do you want any semantics in your syntax markup?
- Do you want an accurate XML representation of the structure of the grammar?
- Do you want to have shiny graphics produced from your DITA code?

There is `<codeblock>` and there is a dedicated syntax diagram domain...

Elements in the Syntax Diagram Domain

Element	Meaning
<delim>	A character that marks the beginning or end of a section within a syntax diagram.
<fragment>	Contains a labeled subpart of the syntax within a syntax diagram.
<fragref>	Mechanism for referencing a syntax fragment within the same syntax diagram.
<groupchoice>	Provides a set of choices between groups of pieces of syntax.
<groupcomp>	Groups a set of pieces of syntax as a single unit.
<groupseq>	Specifies the sequence of groups with pieces of syntax.
<kwd>	Identifies a keyword within a syntax diagram or phrase.
<oper>	Identifies an operator within a syntax definition.
<repsep>	Identifies a character that indicates that a group of syntax elements can (or should) be repeated in a syntax diagram.
<sep>	A character that separates pieces of syntax in a syntax diagram.
<synblk>	A syntax block organizes small pieces of a syntax definition into a larger piece.
<synnote>	Provides additional information within a syntax diagram.
<synnoteref>	Mechanism for referencing a syntax note within the same syntax diagram.
<synph>	A syntax phrase is a small group of pieces of syntax.
<syntaxdiagram>	Represents the syntax of a command, function call, or programming language statement.
<var>	Identifies a variable within a syntax diagram or phrase.

Example

dibabuild - internal command-line utility

<codeblock>

```
dibabuild [OPTIONS] { -d DITAMAP | -p PROJECT }
```

bare-bones <codeblock>

</codeblock>

<codeblock>

```
<cmdname>dibabuild</cmdname> [OPTIONS] {  
  <option>-d</option> <parmname>DITAMAP</parmname> |  
  <option>-p</option> <parmname>PROJECT</parmname> }
```

semantically enriched <codeblock>

</codeblock>

```
<synph>dibabuild [OPTIONS] { <kwd>-d</kwd> <var>DITAMAP</var> |  
  <kwd>-p</kwd> <var>PROJECT</var> }
```

syntax phrase <synph>

</synph>

Example

dibabuild - internal command-line utility

```
dibabuild [OPTIONS] { -d DITAMAP | -p PROJECT }
```

<syntaxdiagram>

<groupseq>

<kwd>dibabuild</kwd>

<groupseq importance="optional">

<var>OPTIONS</var>

</groupseq>

<groupchoice importance="required">

<groupseq>

<oper>-</oper><kwd>d</kwd> <var>DITAMAP</var>

</groupseq>

<groupseq>

<oper>-</oper><kwd>p</kwd> <var>PROJECT</var>

</groupseq>

</groupchoice>

</groupseq>

</syntaxdiagram>

Rendering

Pure HTML 5

dibabuild

This command builds output from DITA, Lightweight DITA and/or Markdown sources.

Using `<codeblock>`

```
dibabuild [OPTIONS] { -d DITAMAP | -p PROJECT }
```

Using `<synph>`:

```
dibabuild [OPTIONS] { -d DITAMAP | -p PROJECT }
```

Using `<syntaxdiagram>`:

```
dibabuild [ OPTIONS ] { - d DITAMAP | - p PROJECT }
```

Using `<syntaxdiagram>`:

```
dibabuild [ OPTIONS ] { - d DITAMAP | - p PROJECT }
```

Options

```
[ { - E ENGINE | -- engine = ENGINE } ]
```

```
[ { - F FILTER (explicit id) | -- filter = FILTER } ]
```

```
[ { - V | -- version } ]
```

¹ use this file as .ditaval file

Using Robert's DITA-OT plugins for syntax diagrams

dibabuild

This command builds output from DITA, Lightweight DITA and/or Markdown sources.

Using `<codeblock>`

```
dibabuild [OPTIONS] { -d DITAMAP | -p PROJECT }
```

Using `<synph>`:

```
dibabuild [OPTIONS] { -d DITAMAP | -p PROJECT }
```

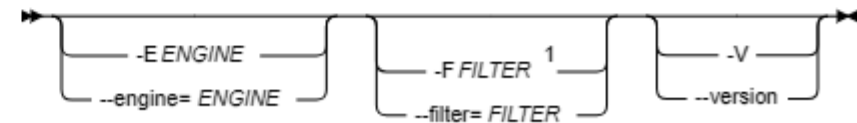
Using `<syntaxdiagram>`:



Using `<syntaxdiagram>`:



Options



Notes:

¹ use this file as .ditaval file

Advanced Usage

Fragments and syntax blocks

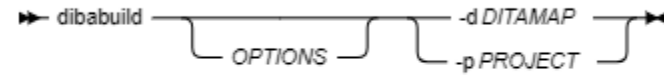
```
<syntaxdiagram>
  <groupseq>
    <kwd>dibabuild</kwd>
    <!-- Mandatory arguments -->
  </groupseq>
  <fragment>
    <title>Options</title>
    <groupchoice importance="optional">
      <groupcomp>
        <oper>-</oper><kwd>E</kwd><sep>#xA0;</sep>
        <var>ENGINE</var>
      </groupcomp>
      <groupcomp>
        <oper>--</oper><kwd>engine</kwd><sep>=</sep>
        <var>ENGINE</var>
      </groupcomp>
    </groupchoice>
  </fragment>
  <!-- More options -->
  <synblk conkeyref="syntax_blocks/loggingOpts"/>
</syntaxdiagram>
```

External file

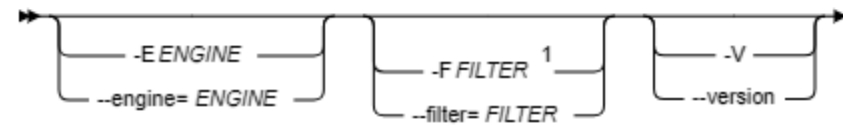
```
<syntaxdiagram>
  <synblk id="loggingOpts">
    <title>Logging options</title>
    <fragment>
      <groupseq importance="optional">
        <oper>-</oper><kwd>z</kwd><sep>#xA0;</sep>
        <var>DEBUGLEVEL</var>
        <oper>--</oper><kwd>debug</kwd><sep>=</sep>
        <var>DEBUGLEVEL</var>
      </groupseq>
      <groupseq importance="optional">
        <oper>--</oper><kwd>verbose</kwd>
      </groupseq>
    </fragment>
  </synblk>
</syntaxdiagram>
```

Rendering

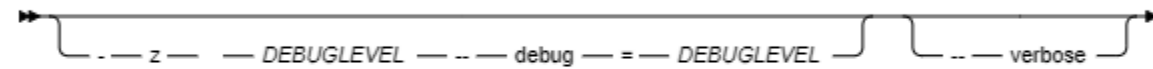
Using `<syntaxdiagram>` and `<synblk>`:



Options



Logging options



Notes:

¹ use this file as `.ditaval` file

Conclusio

“I wouldn’t want to use any of it.”

(Simon Bate in a blog post (2013))

Purism or Pragmatism?

Your choice...

References

ANDERSON, R. (2019, October 11): *DITA-OT plugins for SVG Syntax diagrams*. Retrieved from <https://github.com/robander/svg-syntaxdiagrams/>.
Version 2.0.

BATE, S. (2013, January 8): *Perplexed by complex syntax: understanding syntax diagrams in DITA*. Retrieved from <https://www.scriptorium.com/2013/01/perplexed-by-complex-syntax-understanding-syntax-diagrams-in-dita/>.

CHOMSKY, N. (1957): *Syntactic Structures*. Den Haag : Mouton.

EISENBACH, Andre & Mei (2022, December 15): *Dynamic JavaScript version of phpSyntaxTree*. Retrieved from <https://github.com/int2str/jssyntaxtree/>. Version 1.2.

RADERMACHER, G. (2024, January 6): *Railroad Generator*. Retrieved from <https://github.com/GuntherRademacher/rr/>. Version 2.1.

BRAY, T. ET AL. (2008, November 26) (eds.): Notation. In: *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C recommendation. Retrieved from <https://www.w3.org/TR/xml/#sec-notation>.

Contact me on LinkedIn or by email: Frank.Wegmann@softwareag.com

