HUMBOLDT-UNIVERSITÄT ZU BERLIN

LaTeX for Linguists

LaTeX 7: Math mode 2 & trees

Sebastian Nordhoff & Antonio Machicao y Priemer

www.linguistik.hu-berlin.de/staff/amyp

LOT 2019, Amsterdam

January 17, 2019

Contents

Non-exhaustive lists of symbols

Symbols you could need (the following lists are by no means exhaustive):

=	=	∼	\sim	∞	\infty
±	\pm	≈	\approx	∅	\emptyset
·	\cdot	⊂	\subset	□	\Box
×	\times	⊃	\supset	%	\%
∘	\circ	⊆	\subseteq	$	\$
∈	\in	∩	\cap	&	\&
∋	\ni	∪	\cup	#	\#
≠	\neq	∀	\forall	\	\backslash
≤	\leq	∃	\exists	...	\dots
≥	\geq	∧	\land	<	<
≪	\ll	∨	\lor	>	>
≫	\gg	¬	\lnot		

Table 1: Some non-specific symbols

$\rightarrow$	`\rightarrow`	$\Downarrow$	`\Downarrow`	$\{\}$	`\{\}`
$\leftarrow$	`\leftarrow`	$\mapsto$	`\mapsto`	$\mathcal{A}$	`\mathcal{A}`
$\leftrightarrow$	`\leftrightarrow`	$\rightsquigarrow$	`\leadsto`	$\mathfrak{A}$	`\mathfrak{A}`
$\Rightarrow$	`\Rightarrow`	$\xrightarrow[abc]{xyz}$	`\xrightarrow[abc]{xyz}`	$\mathbb{R}$	`\mathbb{R}`
$\Leftarrow$	`\Leftarrow`	$()$	`()`	$\aleph$	`\aleph`
$\Leftrightarrow$	`\Leftrightarrow`	$[]$	`[]`		

Table 2: Some arrows, brackets, fonts

α	`\alpha`	θ	`\theta`	ε	`\varepsilon`
γ	`\gamma`	ϕ	`\phi`	ϑ	`\vartheta`
δ	`\delta`	Γ	`\Gamma`	Φ	`\Phi`
ϵ	`\epsilon`	Δ	`\Delta`	φ	`\varphi`

Table 3: Some Greek letters and variants

$\tilde{a}$	\tilde{a}	$\notin$	\notin	$\widetilde{abc}$	\widetilde{abc}
$\bar{a}$	\bar{a}	$\dot{a}$	\dot{a}	$\overline{abc}$	\overline{abc}
$\vec{a}$	\vec{a}	$\ddot{a}$	\ddot{a}	$\overrightarrow{abc}$	\overrightarrow{abc}
$\hat{a}$	\hat{a}	$\doteq$	\ddot{=}	$\widehat{abc}$	\widehat{}

Table 4: Some combinations of symbols

Some lists of symbols for LaTeX:

- List of logic symbols (Wikipedia): `https://en.wikipedia.org/wiki/List_of_logic_symbols`
- LaTeX for Logicians: `http://www.logicmatters.net/latex-for-logicians/`
- The Great, Big List of LaTeX Symbols: Carlisle et al. (2001)
- The Comprehensive LaTeX Symbol List – Symbols accessible from LaTeX: Pakin (2017)

Draw the symbol and get the code:

- `http://detexify.kirelabs.org`

Set theory

```
$\{\textrm{a}\} \subset \{\textrm{a,e}\}$
```

(1) $\{\textrm{a}\} \subset \{\textrm{a,e}\}$

```
$\emptyset \subseteq \{\textrm{a,b}\}$
```

(2) $\emptyset \subseteq \{\textrm{a,b}\}$

```
$\# \{\emptyset, \textrm{a} \} = 2$
```

(3) $\#\{\emptyset, \textrm{a}\} = 2$

```
$\emptyset \in \{\emptyset, \textrm{a} \}$
```

(4) $\emptyset \in \{\emptyset, \textrm{a}\}$

```
$\emptyset \notin \{\textrm{a}\}$
```

(5) $\varnothing \notin \{\mathrm{a}\}$

```
If $|\textrm{A}| = n$ then $|\mathfrak{P}(\textrm{A})|=2^{n}$
```

(6) If $|\mathrm{A}| = n$ then $|\mathfrak{P}(\mathrm{A})| = 2^{n}$

```
$\{\textrm{a, e}\} \setminus \{\textrm{e, u}\} = \{\textrm{a}\}$
```

(7) $\{\mathrm{a}, \mathrm{e}\} \setminus \{\mathrm{e}, \mathrm{u}\} = \{\mathrm{a}\}$

```
DeMorgan: $ \overline{[ \textrm{A} \cup \textrm{B} ]} =
[ \overline{\textrm{A}} \cap \overline{\textrm{B}} ] $
```

(8) DeMorgan: $\overline{[\mathrm{A} \cup \mathrm{B}]} = [\overline{\mathrm{A}} \cap \overline{\mathrm{B}}]$

Propositional Logic

```
DeMorgan's law:
$\lnot (P \lor Q ) \Leftrightarrow
(\lnot P \wedge \lnot Q)$

Biconditional law:
$(P \leftrightarrow P) \Leftrightarrow
((P \rightarrow Q) \wedge (Q \rightarrow P))$

Logical consequence:
$((p \rightarrow q) \wedge p) \Rightarrow q$
```

(9) DeMorgan's law: $\neg(P \vee Q) \Leftrightarrow (\neg P \wedge \neg Q)$

(10) Biconditional law: $(P \leftrightarrow P) \Leftrightarrow ((P \rightarrow Q) \wedge (Q \rightarrow P))$

(11) Logical consequence: $((p \rightarrow q) \wedge p) \Rightarrow q$

Quantifiers

```
$\exists x [$\textsc{woman}$(x)$ $\land$ \textsc{sleep}$(x)]$

$\forall x [$\textsc{woman}$(x)$ $\rightarrow$ \textsc{sleep}$(x)]$
```

(12) **Existential quantifier:** *A woman sleeps.*
$\exists x[\textsc{woman}(x) \land \textsc{sleep}(x)]$
↛ There is only one sleeper.

(13) **Universal quantifier:** *Every woman sleeps.*
$\forall x[\textsc{woman}(x) \rightarrow \textsc{sleep}(x)]$
↛ Only women are sleepers.

Meaning brackets

In order to use the meaning brackets $[\![ ]\!]$ you can

1. (using XeLaTeX) copy the Unicode symbol,
2. make an own command for the symbol to use the Unicode symbol,
3. use the package `MnSymbol`. It provides the meaning brackets a.o. symbols.

```
\usepackage{MnSymbol}
```

Meaning brackets can be used **only in math mode**:

```
$\lsem \alpha \beta \rsem = \lsem \beta \rsem (\lsem \alpha \rsem)$
```

(14) $[\![\alpha\beta]\!] = [\![\beta]\!]([\![\alpha]\!])$ [Function application]

Writing formulae

```
\ea $\lsem [_{\textrm{PP}}$\emph{in Amsterdam}$] \rsem (s')
    = \lambda P \lambda x [P(x) \land [x \textrm{ is in Amsterdam in } s']]$
\z
```

(15) $[\![[_{\mathrm{PP}}\textit{in Amsterdam}]]\!](s') = \lambda P \lambda x [P(x) \wedge [x \text{ is in Amsterdam in } s']]$

- *in Amsterdam*: object language
- s', x, P: variables
- is in Amsterdam: invariable predicate
- PP: Index

Trees

There are different packages for drawing trees:

- `qtree`
- `pstrees` (complex syntax, but more powerful than `qtree`)
- `tikz-qtree`
- `forest` (simple syntax, more powerful than `pstrees` and `qtree`, based on `tikz`)
- ...

Loading forest

```
\usepackage{forest}
```

`forest` provides many features for trees needed in linguistics.

These features can be loaded specifying the **option** `linguistics`.

```
\usepackage[linguistics]{forest}
```

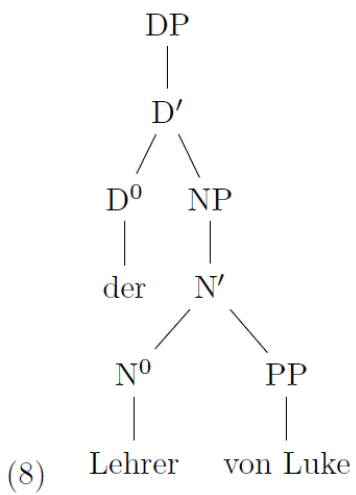


Fig. 1: without `linguistics`

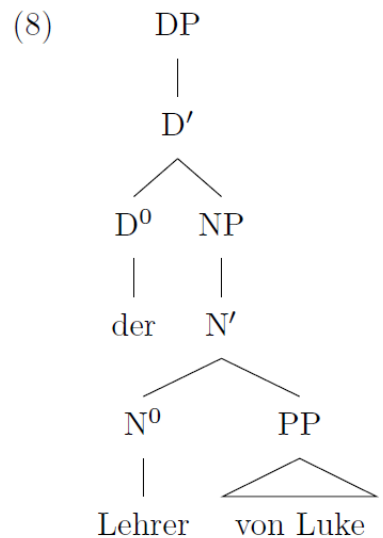


Fig. 2: with `linguistics`

`gb4e` re-defines some commands needed for `forest`. If you are using `gb4e`, you must load `forest` first and `gb4e` after.

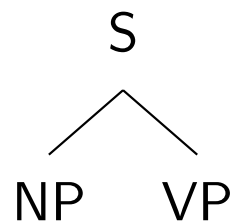
```
\usepackage[linguistics]{forest}

\usepackage{gb4e}
```

forest syntax

1. Use the `forest` **environment**.
2. Inside the `forest` environment, write the **bracket notation** for your tree.
3. Do **not** use **empty lines**!

```
\begin{forest}
[S [NP] [VP]]
\end{forest}
```

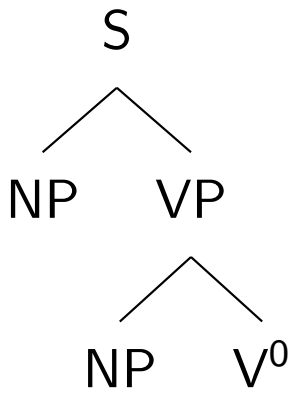
(16)



- Practice the bracket notation: `http://ironcreek.net/phpsyntaxtree/`

For bigger trees, it is useful – for the sake of clarity – not to write the bracket notation linearly.

```
\begin{forest}
[S
  [NP]
  [VP
    [NP]
    [V$^{0}$]
  ]
]
\end{forest}
```



vs.

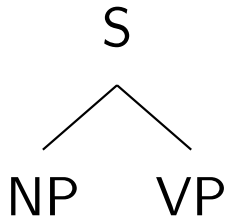
```
\begin{forest}
[S [NP] [VP [NP] [V$^{0}$]]]
\end{forest}
```

Trees in example environments

When using the option `linguistics`, you can embed the tree in an example environment.

```
\ea
\begin{forest}
[S [NP] [VP]]
\end{forest}
\z
```

(17)

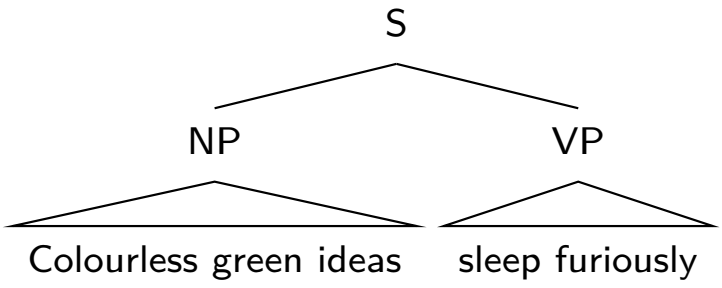


Abbreviating nodes

With the option `roof`, you can abbreviate nodes.

```
\ea
\begin{forest}
[S
  [NP [Colourless green ideas, roof]]
  [VP [sleep furiously, roof]]
]
\end{forest}
\z
```

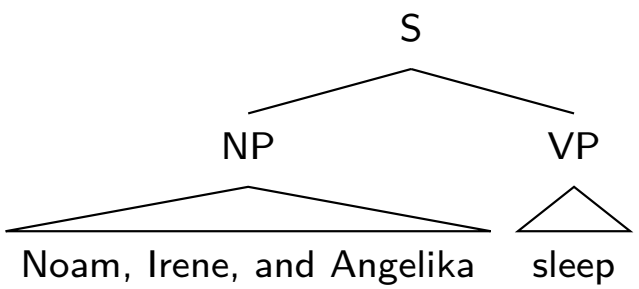
(18)



Take into account that options in `forest` (based on `TikZ`) are given by a **comma**. That means, you can use commas only when you **protect** them.

```
\ea
\begin{forest}
[S [NP [Noam{,} Irene{,} and Angelika,
    roof]] [VP [sleep, roof]]]
\end{forest}
\z
```

(19)

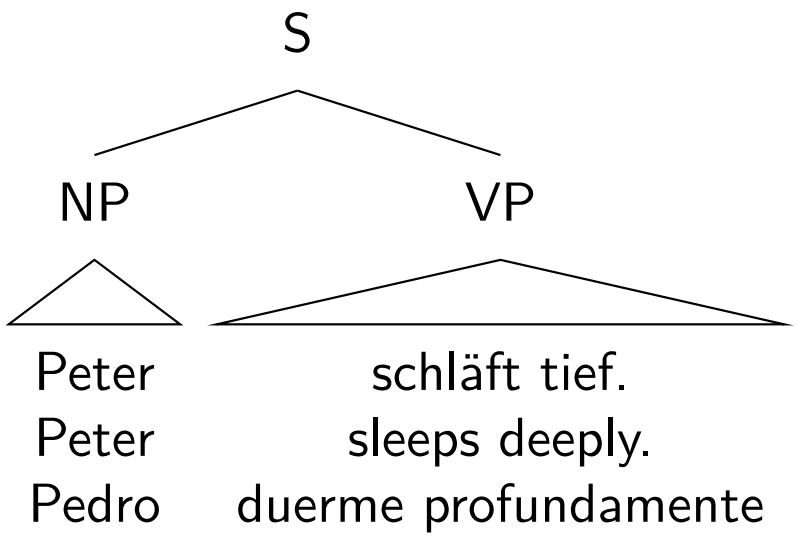


Glossing or translating

With `\\`, you can add **glosses or translations** to your tree.

```
\begin{forest}
[S
  [NP
    [Peter \\ Peter \\ Pedro, roof]
  ]
  [VP
    [schläft tief. \\ sleeps deeply. \\
          duerme profundamente, roof]
  ]
]
\end{forest}
```

(20)



Sub- and superscript

The characters `^` and `_` are used in **math mode** for sub- and superscript, respectively.

```
$x^1$

$x_1$
```

(21) x^1

(22) x_1

The **default scope** of `^` and `_` is only one character (23), use `{ }` to **expand** it, see (24).

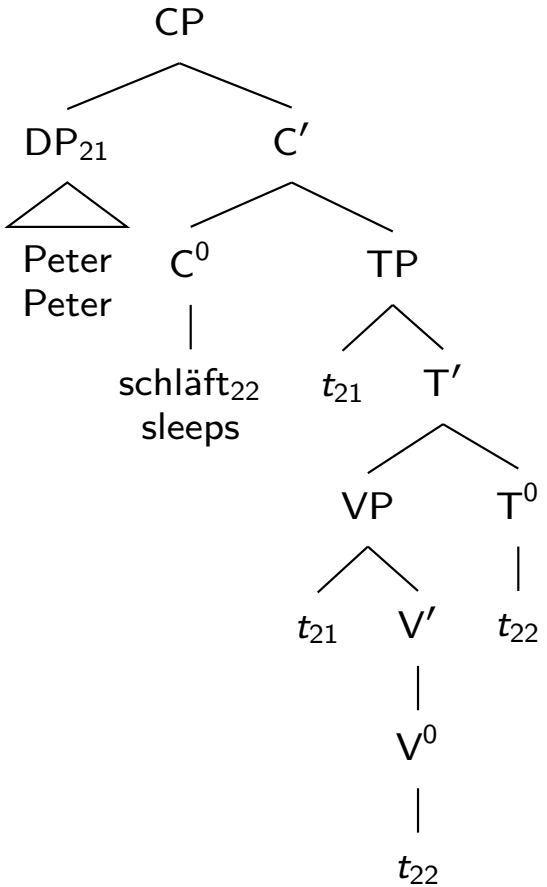
```
\ea X$^1$ Y$^21$ X$_1$ Y$_21$ \label{ex:SubSup1}

\ex X$^{1}$ Y$^{21}$ X$_{1}$ Y$_{21}$ \label{ex:SubSup2}
\z
```

(23) $X^1\ Y^21\ X_1\ Y_21$

(24) $X^1\ Y^{21}\ X_1\ Y_{21}$

Tree with sub- and superscripts

```
[CP
  [DP$_{21}$ [Peter \\ Peter, roof]]
  [C$^{\prime}$
    [C$^{0}$ [schläft$_{22}$ \\ sleeps
        ]]
    [TP
      [$t_{21}$]
      [T$'$
        [VP
          [$t_{21}$]
          [V$^{\prime}$
            [V$^{0}$ [$t_{22}$]]
          ]
        ]
        [T$^{0}$ [$t_{22}$]]
      ]
    ]
  ]
]
```



Arrows

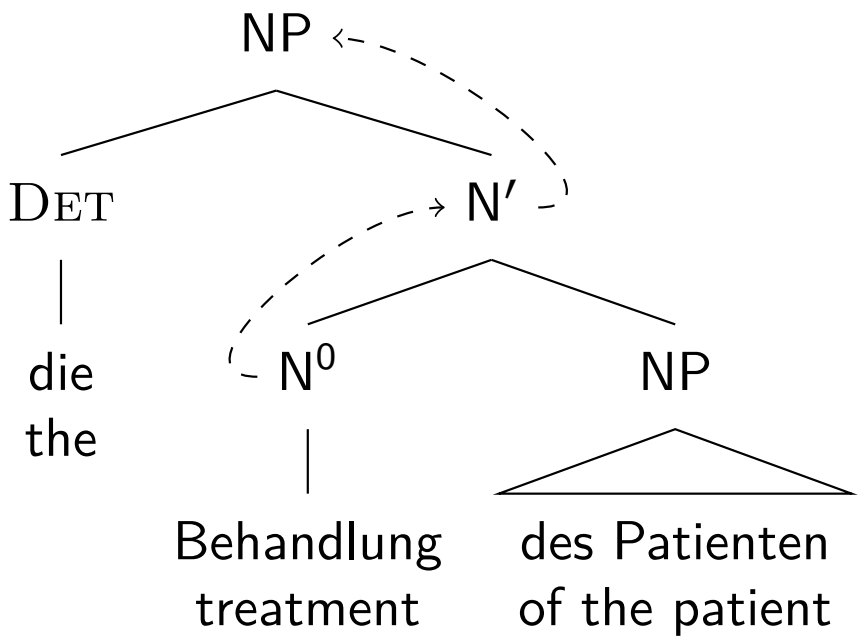
Arrows/lines **from node to node** (e.g. for movement, projection, etc.) can be drawn easily.

Give the nodes a **name** (command: `, name=`) and draw an arrow with the following command:

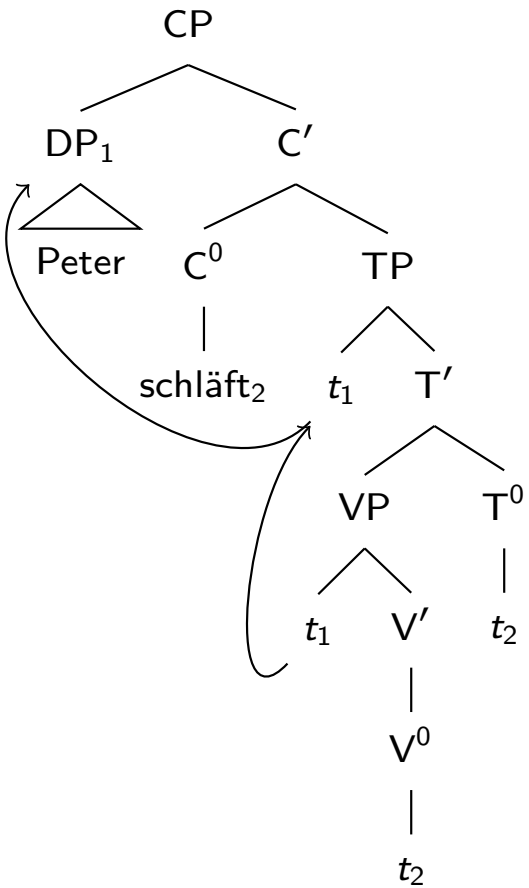
```
\draw[X] (Y) to[out=V, in=W] (Z);

\draw[->] (T10) to[out=south west, in=south west](T11);
```

- X: type of arrow/line (`->` `<-` `<->` `-`)
- Y: name of start node
- Z: name of end node
- V: starting position of the arrow at the start node (`south`/`north` + `east`/`west`)
- W: end position of the arrow at the end node (`south`/`north` + `east`/`west`)
- ;: end of the command

```
[NP, name=N2
  [\textsc{Det} [die \\ the]]
  [N$'$, name=N1
    [N$^0$, name=N0 [Behandlung \\ treatment]]
    [NP [des Patienten \\ of the patient, roof]]
  ]
]
\draw[->,dashed] (N0) to[out=west,in=west] (N1);
\draw[->,dashed] (N1) to[out=east,in=east] (N2);
```



```
[CP
  [DP$_{1}$, name=T12 [Peter, roof]]
  [C$^{\prime}$
    [C$^{0}$ [schläft$_{2}$, name=T22]]
    [TP
      [$t_{1}$, name=T11]
      [T$^{\prime}$
        [VP
          [$t_{1}$, name=T10]
          [V$^{\prime}$
            [V$^{0}$ [$t_{2}$, name=T20]]
          ]
        ]
        [T$^{0}$ [$t_{2}$, name=T21]]
      ]
    ]
  ]
]
\draw[->] (T10)
to[out=south west, in=south west](T11);
\draw[->] (T11)
to[out=south west, in=south west](T12);
```

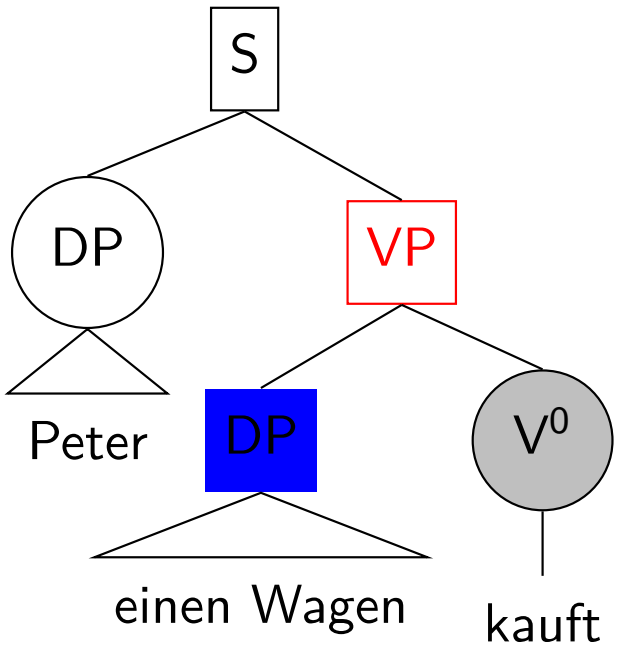


Marking nodes

Some options:

- `draw`: square
- `circle, draw`: circle
- `red`: marking node with red
- `fill=X`: fill background of node with colour *X*
- `circle, draw, fill=lightgray`: circle around node, background in grey

```
[S, draw
  [DP, circle, draw
    [Peter, roof]]
  [VP, draw, red
    [DP, fill=blue
      [einen Wagen, roof]]
    [V$^{0}$, circle, draw, fill=lightgray
      [kauft]]
  ]
]
```



Syllabic structures

The `forest` offers the style `GP1` for syllabic structures.

```
\begin{forest} GP1, [
[$\sigma$
  [O
    [[C[\textipa{l}]]]
  ]
  [R  [N
        [V[\textipa{a}]]
      ]
  ]]
  [$\sigma$
    [O [ [ C[\textipa{t}] ] ] ]
    [R
      [N [V [\textipa{E}] ] ]
      [K [C [\textipa{\c{c}}] ] ]
]]
]
\end{forest}
```

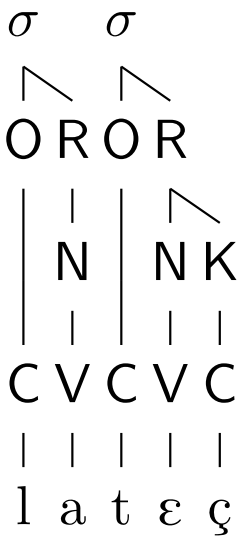


Fig. 3: Two syllables with GP1

```
\begin{forest} GP1
[ [$\sigma$
    [O
      [[C[\textipa{S}]]]
      [[C[\textipa{t}]]]
      [[C[\textipa{\textscr}]]]
    ]
    [R
      [N
        [V[\textipa{U}]]
      ]
      [K
        [C[\textipa{m}]]
        [C[\textipa{\t{pf}}]]
        [C[\textipa{s}]]
        [C[\textipa{t}]]
      ]
    ]
  ]
]
\end{forest}
```

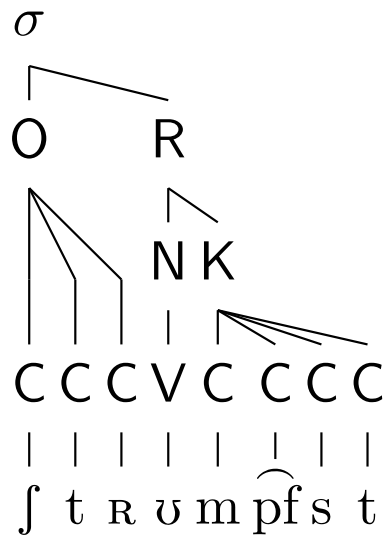


Fig. 4: Complex syllable with GP1

Without using GP1, you can draw your syllabic structures with forest. You will need the (TikZ) commands , phantom and , tier=word.

```
\begin{forest}
[,phantom
  [$\sigma$
    [O
      [x, tier=word[\textipa{f}]]
      [x, tier=word
        [\textipa{\textscr }]
      ]
    ]
    [R
      [N
        [x, tier=word
          [\textipa{E}]
        ]
      ]
      [K [x[\textipa{\c{c}}]]]]
      ]
]
\end{forest}
```

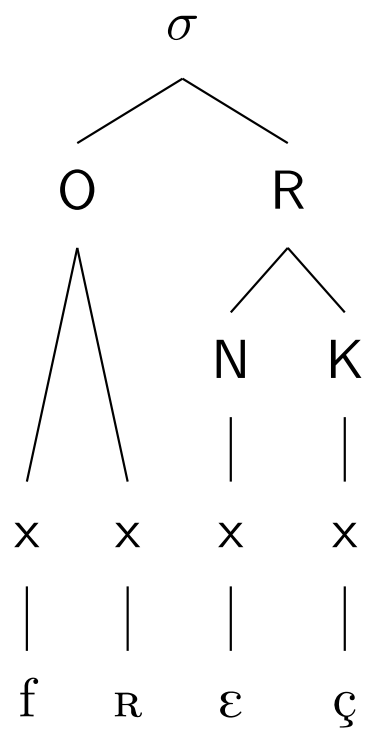


Fig. 5: Syllable without GP1

```
\begin{forest}
[,phantom
  [$\sigma$
    [O
      [x, tier=word [\textipa{f}]]
      [x, tier=word[\textipa{K}]]]
    [R
      [N
        [x, tier=word[\textipa{\
             textopeno}]]]
      [K
        [x[\textipa{s}]]]]
  ]
  [$\sigma$
    [O
      [x, tier=word [\textipa{t}]]]
    [R
      [N [x[\textipa{I}]]]
      [K[x[\c{c}]]]]
  ]
]
\end{forest}
```

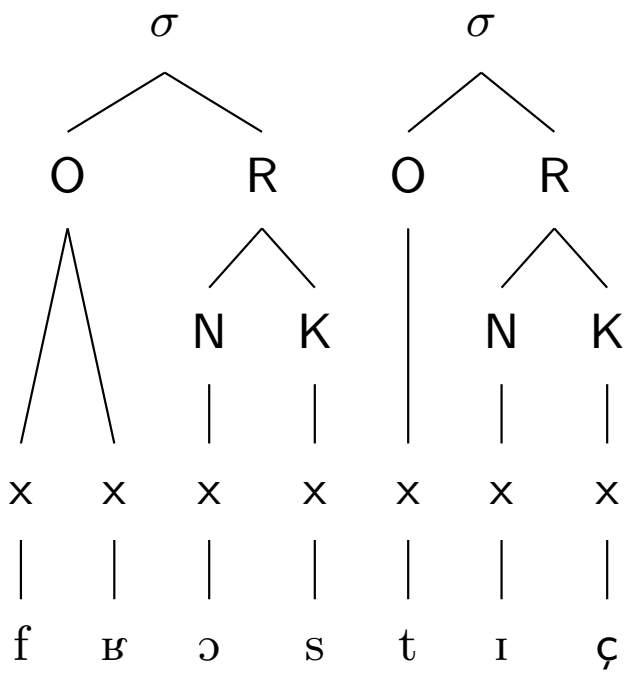


Fig. 6: Two syllables

```
\begin{forest}
[,phantom
  [$\sigma$
    [O
      [x, tier=word
        [\textipa{P}]]]
    [R
      [N
        [x, tier=word
          [\textipa{\t{aU}}, name=aU] ]
        [x, name=x]
      ]
      [K
        [x [\textipa{x}] ] ] ]
  ]
]
{\draw[black] (aU.north)--(x.south);}
\end{forest}
```

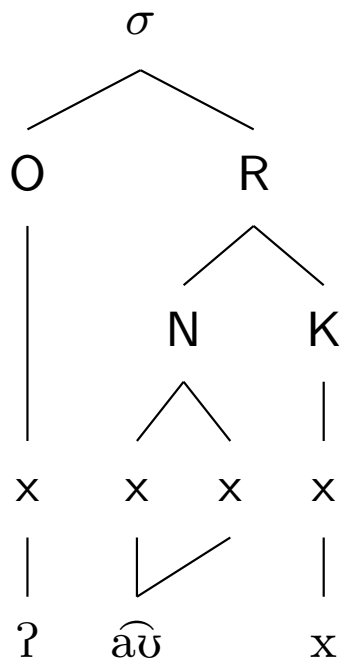


Fig. 7: Diphthongs and long vowels

```
\begin{forest}
[,phantom
  [$\sigma$
    [O [x, tier=word [\textipa{t}] ]
    ]
    [R
      [N [x, tier=word [\textipa{I}] ]
          ]
      [K [x, name=x [\textipa{k}] ] ]
    ]
  ]
  [$\sigma$
    [O, name=onset]
    [R
      [N [x [\textipa{@}] ] ]
      [K [x [\textipa{n}] ] ] ]
    ]
  ]
\draw[black] (x.north)--(onset.south);
\end{forest}
```

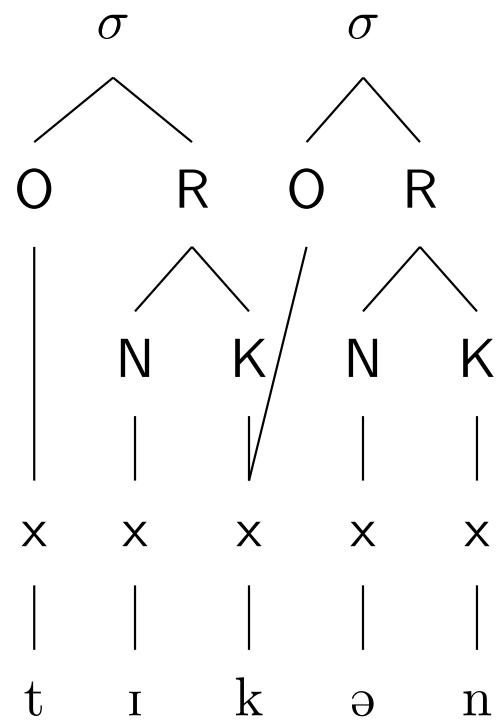


Fig. 8: Ambisyllabic consonant

Further features

- `forest` is a very powerful package. Check the package documentation (Živanovi, 2017) to see all of its benefits.
- Also, check the *Quick start guide* for linguists (Vanden Wyngaerd, 2016).

Exercise

Go to
`https://github.com/langsci/latex4linguists/blob/master/4-1.md`
and
`https://github.com/langsci/latex4linguists/blob/master/4-2.md` and follow the instructions of **all blocks** in your `.tex` file.

Internet sources I

- Link: Akzente und Sonderzeichen in LaTeX.
  `https://de.wikibooks.org/wiki/LaTeX/_Akzente_und_Sonderzeichen`
  [Access: 10/10/2017]
- Link: Detexify
  `http://detexify.kirelabs.org`
  [Access: 08/12/2017]
- Link: LaTeX/Special Characters.
  `https://en.wikibooks.org/wiki/LaTeX/Special_Characters`
  [Access: 02/01/2019]
- Link: List of logic symbols – Wikipedia
  `https://en.wikipedia.org/wiki/List_of_logic_symbols`
  [Access: 08/12/2017]
- Link: LaTeX for Logicians:
  `http://www.logicmatters.net/latex-for-logicians/`
  [Access: 08/12/2017]

Internet sources II

- Link: The Comprehensive LaTeX Symbol List – Symbols accessible from LaTeX (Pakin, 2017):
  `https://ctan.org/tex-archive/info/symbols/comprehensive/`
  [Access: 08/12/2017]
- Link: The Great, Big List of LaTeX Symbols (Carlisle et al., 2001):
  `https://www.rpi.edu/dept/arc/training/latex/LaTeX_symbols.pdf`
  [Access: 08/12/2017]

Literature I

Carlisle, David, Scott Pakin & Alexander Holt. 2001. The great, big list of LaTeX symbols. Handbook. https://www.rpi.edu/dept/arc/training/latex/LaTeX_symbols.pdf.

Freitag, Constantin & Antonio Machicao y Priemer. 2015. LaTeX-Einführung für Linguisten. Manuscript. https://www.linguistik.hu-berlin.de/de/staff/amyp/latex-einfuehrung.

Knuth, Donald E. 1986. *The TeX book*. Boston: Addison-Wesley.

Kopka, Helmut. 1994. *LaTeX: Einführung*, vol. 1. Bonn: Addison-Wesley.

Machicao y Priemer, Antonio. 2018. Kopf. In Stefan Schierholz & Pál Uzonyi (eds.), *Grammatik: Syntax* (Wörterbücher zur Sprach- und Kommunikationswissenschaft 1.2), Berlin: De Gruyter. https://www.researchgate.net/publication/325046855_Kopf_Pre-Print.

Pakin, Scott. 2017. The comprehensive LaTeX symbol list – symbols accessible from LaTeX. Handbook. https://ctan.org/tex-archive/info/symbols/comprehensive/.

Vanden Wyngaerd, Guido. 2016. Forest quickstart guide for linguists. Manuskript. https://ling.auf.net/lingbuzz/003391.

Živanovi, Sao. 2017. Forest: a pgf/tikz-based package for drawing linguistic trees v2.1.5. *CTAN: Comprehensive TeX Archive Network* https://ctan.org/pkg/forest.