

The linphon Reference

Version 0.1.0

Contents

1	Installation and import	1
1.1	Importing from Typst Universe	1
1.2	Local installation	1
2	Feature matrices	2
2.1	Binary/equipollent features	2
2.2	Privative/monovalent features	2
2.3	Categorial placeholders	2
2.4	Conditional submatrices (for if-then conditions)	2
2.5	Different types of feature values	3
2.6	Inline feature matrices	3
3	Cases / Option statements	4
3.1	Two-sided	4
3.2	One-sided	4
4	Rewrite rules	4
4.1	Context-free rules	4
4.2	Rules with contextual specification	4
5	Constraints	5
6	Custom brackets	5
7	Examples of rules	6
7.1	<i>l</i> -deletion	6
7.2	Tapping rule	6
7.3	SPE, Chapter 4, Rule 60 (p. 200)	6
7.4	SPE, Chapter 3, Rule 82 (p. 99)	7

1 Installation and import

1.1 Importing from Typst Universe

The easiest way to use linphon is to just import it from the Typst Universe with

```
#import "@preview/linphon:0.1.0"
```

1.2 Local installation

To install linphon locally, download the latest version from the linphon GitHub repository and unpack it inside Typst's data dir, as follows:

Linux \$XDG_DATA_HOME/typst/packages/local/

MacOS ~/Library/Application Support/typst/packages/local/

Windows %APPDATA%\typst\packages\local\

You can then import the local package with

```
#import "@local/linphon:0.1.0"
```

2 Feature matrices

2.1 Binary/equipollent features

Give binary/equipollent features as an array of (feature-value, feature-name).

```
#linphon.fmat(  
  ("+", "voice"),  
  ("-", "nasal"),  
  ("+", "coronal"),  
  ("-", "back"),  
)
```

$$\begin{bmatrix} + \text{ voice} \\ - \text{ nasal} \\ + \text{ coronal} \\ - \text{ back} \end{bmatrix}$$

2.2 Privative/monovalent features

Give privative/monovalent features as an array consisting only of the feature name (NB: a trailing comma is needed to unambiguously tell Typst that the parentheses denote an array, i.e. (feature-name,)). These will be typeset left-aligned.

```
#linphon.fmat(  
  ("consonantal",),  
  ("voice",),  
  ("coronal",),  
)
```

$$\begin{bmatrix} \text{consonantal} \\ \text{voice} \\ \text{coronal} \end{bmatrix}$$

2.3 Categorical placeholders

Simply give these as bare string/content arguments and they will be typeset centred within the feature matrix.

```
#linphon.fmat(  
  "C",  
  ("+", "nasal"),  
  ("+", "coronal"),  
)
```

$$\begin{bmatrix} \text{C} \\ + \text{ nasal} \\ + \text{ coronal} \end{bmatrix}$$

```
#linphon.fmat(  
  linphon.dash(length: 3.5em),  
  ("+", "nasal"),  
  ("+", "coronal"),  
)
```

$$\begin{bmatrix} \text{-----} \\ + \text{ nasal} \\ + \text{ coronal} \end{bmatrix}$$

2.4 Conditional submatrices (for if-then conditions)

You can pass an array of arrays to include a submatrix, which is set in angle brackets by default and typically used to indicate if-then conditions (e.g. if back, then also round), esp. across several feature matrices in a rule.

```
#linphon.fmat(  
  ("+", "syllabic"),  
  ("+", "high"),  
  ("-", "low"),  
  (  
    ("+", "back"),  
    ("+", "round"),  
  )  
)
```

$$\begin{bmatrix} + \text{ syllabic} \\ + \text{ high} \\ - \text{ low} \\ \left\langle \begin{array}{l} + \text{ back} \\ + \text{ round} \end{array} \right\rangle \end{bmatrix}$$

2.5 Different types of feature values

When given as feature values, the strings "+", "-", "plus", "minus", "plus.minus", "minus.plus", "m" (*marked*), "u" (*unmarked*), names of lower-case greek letters ("alpha", "beta", ..., "omega", incl. ".alt"-variants) and numerals will be set as though they had been written in math mode.

```
#linphon.fmat(  
  ("+", "F"),  
  ("-", "F"),  
  ("plus.minus", "F"),  
)
```

$$\begin{bmatrix} + F \\ - F \\ \pm F \end{bmatrix}$$

```
#linphon.fmat(  
  ("m", "F"),  
  ("u", "F"),  
  ("0", "F"),  
)
```

$$\begin{bmatrix} m F \\ u F \\ 0 F \end{bmatrix}$$

```
#linphon.fmat(  
  ("alpha", "F"),  
  ("beta", "F"),  
  ("gamma", "F"),  
)
```

$$\begin{bmatrix} \alpha F \\ \beta F \\ \gamma F \end{bmatrix}$$

```
#linphon.fmat(  
  ("-alpha", "F"),  
  ("+beta", "F"),  
  ("-sigma.alt", "F"),  
)
```

$$\begin{bmatrix} -\alpha F \\ +\beta F \\ -\varsigma F \end{bmatrix}$$

2.6 Inline feature matrices

These can be specified in the same manner as the tabular feature matrices above, but will be set as normal text instead. They are also breakable across lines between features.

```
#linphon.fmat-inline(  
  ("+", "voice"),  
  ("-", "nasal"),  
  ("+", "coronal"),  
)
```

$$[+ \text{voice}, - \text{nasal}, + \text{coronal}]$$

```
#linphon.fmat-inline(  
  ("+", "syllabic"),  
  (  
    ("alpha", "back"),  
    ("alpha", "round"),  
  ),  
  ("-", "high"),  
  linphon.dash()  
)
```

$$[+ \text{syllabic}, \langle \alpha \text{ back}, \alpha \text{ round} \rangle, - \text{high}, ____]$$

3 Cases / Option statements

3.1 Two-sided

```
X
#linphon.cases[
  Case 1
][
  Case 2
][...]
Y
```

$$X \left\{ \begin{array}{l} \text{Case 1} \\ \text{Case 2} \\ \dots \end{array} \right\} Y$$

3.2 One-sided

```
X
#linphon.cases-left[
  Case 1
][
  Case 2
][...]

```

$$X \left\{ \begin{array}{l} \text{Case 1} \\ \text{Case 2} \\ \dots \end{array} \right.$$

```
#linphon.cases-left[
  Case 1
][
  Case 2
][...]
X
```

$$\left. \begin{array}{l} \text{Case 1} \\ \text{Case 2} \\ \dots \end{array} \right\} X$$

4 Rewrite rules

4.1 Context-free rules

```
#linphon.rule[
  Input
][
  Output
]
```

Input → Output

```
#linphon.rule[CCV][CəCV]
```

CCV → CəCV

4.2 Rules with contextual specification

```
#linphon.rule[
  Input
][
  Output
][
  Context
]
```

Input → Output / Context

```
#linphon.rule[
  #sym.emptyset
][
  ə
][
  C#linphon.dash()C
]
```

∅ → ə / C ____ CV

5 Constraints

<code>#linphon.constraint[</code> <code>illicit stuff</code> <code>]</code>	\ast Illicit stuff
<code>#linphon.constraint[VCCV]</code>	\ast VCCV
<code>#linphon.constraint[</code> <code>#linphon.fmat(</code> <code>("+", "syll"), ("+", "high")</code> <code>)</code> <code>\$("C"_0"V"_0)_0\$</code> <code>#linphon.fmat(</code> <code>("+", "syll"), ("-", "high")</code> <code>)</code> <code>]</code>	$\ast \begin{bmatrix} + \text{ syll} \\ + \text{ high} \end{bmatrix} (C_0 V_0)_0 \begin{bmatrix} + \text{ syll} \\ - \text{ high} \end{bmatrix}$

6 Custom brackets

It is possible to specify custom delimiters (brackets) for both feature matrices and case/option statements. The `linphon.fmat()` and `linphon.fmat-inline()` functions take both a `delim` and a `submatrix-delim` argument, while `linphon.cases()` takes only a `delim` argument.

Possible `delim` values are always an array specifying a left and a right bracket as a pair, each of which can be one of: `"(", ")", "{", "}", "[", "]", "<", ">", "/", "|"`, or to not set a bracket at all, `none`.

<code>#linphon.fmat(</code> <code>delim: ("[" , none),</code> <code>submatrix-delim: ("(" , " "),</code> <code>("-", "syllabic"),</code> <code>("+", "nasal"),</code> <code>("-", "voice"),</code> <code>(</code> <code>("alpha", "coronal"),</code> <code>("-alpha", "dorsal")</code> <code>)</code> <code>)</code>	$\left[\begin{array}{l} - \text{syllabic} \\ + \text{nasal} \\ - \text{voice} \\ \left(\begin{array}{l} \alpha \text{ coronal} \\ -\alpha \text{dorsal} \end{array} \right) \end{array} \right $
<code>#linphon.cases(</code> <code>delim: ("(", ")")</code> <code>)[Case 1][Case 2][...]</code>	$\left(\begin{array}{l} \text{Case 1} \\ \text{Case 2} \\ \dots \end{array} \right)$
<code>#linphon.cases(</code> <code>delim: ("}", "{")</code> <code>)[Case 1][Case 2][...]</code>	$\left\{ \begin{array}{l} \text{Case 1} \\ \text{Case 2} \\ \dots \end{array} \right\}$
<code>#linphon.cases(</code> <code>delim: ("/", "/")</code> <code>)[Case 1][Case 2][...]</code>	$\left/ \begin{array}{l} \text{Case 1} \\ \text{Case 2} \\ \dots \end{array} \right/$
<code>#linphon.cases(</code> <code>delim: (" ", ">")</code> <code>)[Case 1][Case 2][...]</code>	$\left \begin{array}{l} \text{Case 1} \\ \text{Case 2} \\ \dots \end{array} \right\rangle$

7 Examples of rules

7.1 *l*-deletion

```
#linphon.rule[
  \
][
  #sym.emptyset
][
  C #linphon.dash() \#
]
```

$$1 \rightarrow \emptyset / \mathbb{C} _____\#$$

7.2 Tapping rule

```
#linphon.rule[
  #linphon.fmat(
    ("- ", "cont"),
    ("+", "cor"),
    ("- ", "nas")
  )
][
  #linphon.fmat(("+", "tap"))
][
  #linphon.fmat(
    ("+", "syll"),
    ("+", "stress")
  )
  #linphon.dash()
  #linphon.fmat(
    ("+", "syll"),
    ("- ", "stress")
  )
]
```

$$\begin{bmatrix} - \text{cont} \\ + \text{cor} \\ - \text{nas} \end{bmatrix} \rightarrow [+ \text{tap}] / \begin{bmatrix} + \text{syll} \\ + \text{stress} \end{bmatrix} \text{---} \begin{bmatrix} + \text{syll} \\ - \text{stress} \end{bmatrix}$$

7.3 SPE, Chapter 4, Rule 60 (p. 200)

Note that this is not the best way to set the example numbers (60a,b), it would be much better instead to use a package such as `eggs` for that.

```
#grid(
  columns: (1.5cm, 1fr)
)[
  $lr((60), size: #250%)$
][
  (a) ~ #linphon.rule[
    #linphon.cases[æ][u]
  ][
    #linphon.fmat-inline(
      "+", "tense"),
    )
  ][
    #linphon.fmat(
      (linphon.dash-horizon(
        length: 3.5em
      )),
      ("+", "stress")
    )
    _nge_
  ]
  (b) ~ #linphon.rule()[ɔ][ɔ]
```

(60)

(a) $\begin{Bmatrix} \text{æ} \\ \text{u} \end{Bmatrix} \rightarrow [+ \text{tense}] / \left[\begin{array}{c} \text{---} \\ + \text{ stress} \end{array} \right] nge$

(b) $\text{ɔ} \rightarrow \bar{\text{ɔ}} / \text{---CV}[- \text{seg}]$

```

        #linphon.dash-
horizon()CV#linphon.fmat-
inline("-", "seg"))
    ]
]

```

7.4 SPE, Chapter 3, Rule 82 (p. 99)

Note that this is not the best way to set the example number (82), it would be much better instead to use a package such as eggs for that.

```

#grid(
  columns: (1.2cm, 1fr)
)[
  $lr((82), size: #250%)$
][
  #linphon.rule()[
    V
  ][
    #linphon.fmat-inline(
      ("1", "stress")
    )
  ][
    \[_X_#linphon.dash-
horizon()$"C"_0$
    (
      #linphon.fmat(
        ("-", "tense"),
        "V"
      )
      $"C"_0^1$
      #linphon.fmat(
        ("alpha", "voc"),
        ("alpha", "cons"),
        ("-", "ant")
      )_(thin 0)$
    )
  ]
#h(6.75em)
$slash.big$
#linphon.dash-horizon()
#sym.angle.l
#linphon.cases()[
  #linphon.fmat(
    delim: ("<", ">"),
    ("+", $"C"_0$),
  )
  #linphon.fmat(
    ("-", "stress"),
    ("-", "tense"),
    ("V")
  )
]
#linphon.cases(
  delim: ("<", ">"),
)[
  #linphon.fmat-inline("-",
"seg"))
]
$"C"_0$

```

$$\begin{aligned}
 (82) \quad V &\rightarrow [1 \text{ stress}] / [X \text{---} C_0 \left(\begin{bmatrix} - & \text{tense} \\ & V \end{bmatrix} C_0^1 \begin{bmatrix} \alpha \text{ voc} \\ \alpha \text{ cons} \\ - \text{ ant} \end{bmatrix}_0 \right) \\
 & / \text{---} \left\langle \begin{bmatrix} \langle + C_0 \rangle \\ \langle - \text{ stress} \\ - \text{ tense} \\ V \\ \langle [- \text{ seg}] \rangle C_0 \frac{1}{V} \end{bmatrix} \right\rangle C_0 \rangle]_{\langle N \langle A \rangle \rangle}
 \end{aligned}$$

```

#math.attach(math.limits("V"), t:
"1")
]
$"C"_0$#sym.angle.r
\|#sub[
$angle.l
"N" angle.l "A" angle.r
angle.r$
]
]

```