

The linphon Reference

Feature matrices

Binary/equipollent features

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

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

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

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",),  
)
```

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

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"),  
)
```

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

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"),  
  )  
)
```

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

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"),  
)
```

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

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()  
)
```

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

Cases / Option statements

Two-sided

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

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

```

    Case 2
  ][...]
Y

```

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

```

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

Custom brackets

Possible delim values: "(", ")", "{", "}", "[", "]", "<", ">", "/", "|", and none.

```

#linphon.cases(
  delim: ("(", ")")
)[Case 1][Case 2][...]

```

$$\left(\begin{array}{l} \text{Case 1} \\ \text{Case 2} \\ \dots \end{array} \right)$$

```

#linphon.cases(
  delim: ("}", "{")
)[Case 1][Case 2][...]

```

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

```

#linphon.cases(
  delim: ("|", ">")
)[Case 1][Case 2][...]

```

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

```

#linphon.cases(
  delim: ("|", ">")
)[Case 1][Case 2][...]

```

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

Rewrite rules

Context-free rules

```

#linphon.rule[
  Input
][
  Output
]

```

Input → Output

```

#linphon.rule[CCV][CəCV]

```

CCV → CəCV

Rules with contextual specification

```

#linphon.rule[
  Input
][
  Output
]

```

Input → Output / Context

```

][
  Context
]

```

```

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

```

$$\emptyset \rightarrow \text{ə} / \text{C} ___ \text{CV}$$

Constraints

```

#linphon.constraint[
  Illicit stuff
]

```

*Illicit stuff

```

#linphon.constraint[VCCV]

```

*VCCV

```

#linphon.constraint[
  #linphon.fmat(
    ("+", "syll"), ("+", "high")
  )
  $("C"_"0"V"_"0")_0$
  #linphon.fmat(
    ("+", "syll"), ("-", "high")
  )
]

```

$$\begin{bmatrix} + \text{syll} \\ + \text{high} \end{bmatrix} (\text{C}_0 \text{V}_0)_0 \begin{bmatrix} + \text{syll} \\ - \text{high} \end{bmatrix}$$

Examples of rules

l-deletion

```

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

```

$$l \rightarrow \emptyset / \text{C} ___ \#$$

Tapping rule

```

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

```

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

```

#linphon.fmat(
  "+", "syll"),
  "-", "stress")
)
]

```

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

```

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

```

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

```

#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"),

```

$$\begin{aligned}
 (60) \quad & (a) \quad \left\{ \begin{matrix} \text{æ} \\ \text{u} \end{matrix} \right\} \rightarrow [+ \text{tense}] / \left[\begin{matrix} \text{---} \\ + \text{stress} \end{matrix} \right]_{nge} \\
 & (b) \quad \text{ɔ} \rightarrow \bar{\text{ɔ}} / \text{---CV}[- \text{seg}]
 \end{aligned}$$

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

```

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

```