

LingTree User Documentation

H. Andrew Black

SIL Global

lingtree_support@sil.org

March 3, 2025

Version 1.6.0

Copyright © 2002-2025 SIL Global

Contents

1	Introduction	2
2	Keying a tree description	3
2.1	Basic tree description	4
2.2	Indicating node types	5
2.2.1	Indicating a non-terminal node	6
2.2.2	Indicating a lexical node	6
2.2.3	Indicating a gloss node	8
2.2.4	Indicating an “empty element” node	11
2.3	Indicating line types	12
2.3.1	Regular line	12
2.3.2	Indicating a triangle (a phrase)	12
2.3.3	Omitting a line	14
2.4	Indicating the node content	16
2.4.1	Indicating subscripts or superscripts	17
2.4.2	Indicating an abbreviation	25
2.4.3	Handling parentheses within a node	25
3	Formating a tree	27
3.1	Showing “flat” trees	28
3.2	Draw a tree right-to-left	30
3.3	Use column-oriented algorithm	31
3.4	Center each node over its daughters	32
3.5	Setting the font and color information	33
3.5.1	Setting custom font information for a node	35
3.6	Set keyboards	36
3.7	Tree spacing parameters	37
3.8	Background and line parameters	38
3.9	Draw vertical line with empty text	39
3.10	Saving format information for use with new tree diagrams	40

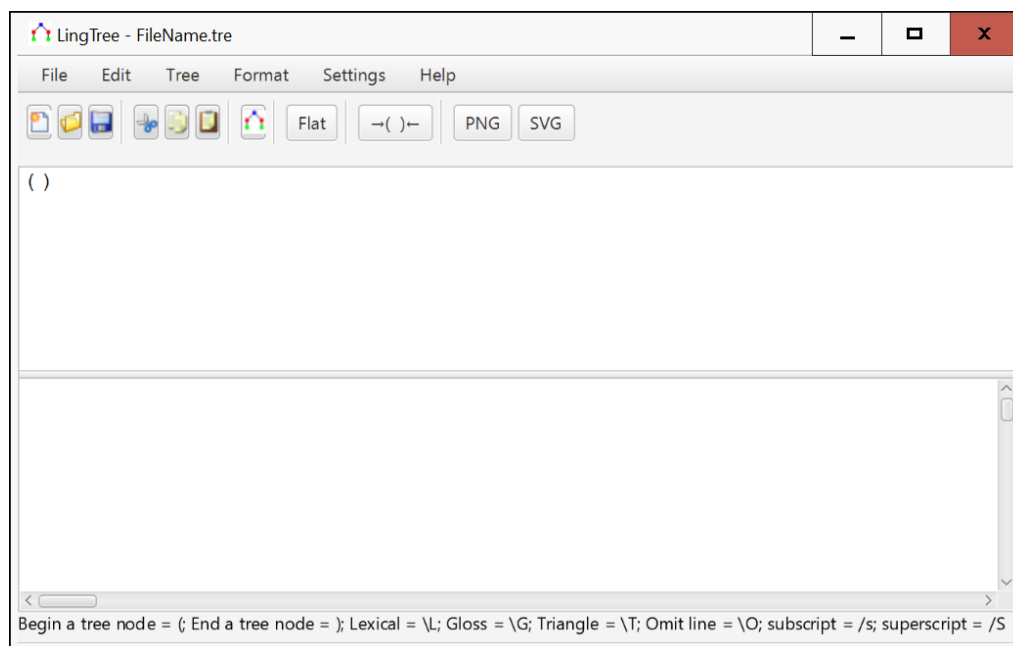
4	User convenience options	40
4.1	Font size for tree description symbols	40
4.2	Draw the tree as you type	40
4.3	Show matching parenthesis with arrow keys	41
4.4	Delay to use when showing matching parentheses	41
4.5	Show full file path	41
4.6	Change the interface language	41
4.7	Quick Reference Guide	42
5	Saving the tree	42
6	Error messages	43
7	Known problems	46
8	Support	47
	Index	48

1 Introduction

LingTree is a tool to produce linguistic tree diagrams. You key in a description of the tree, process that description, and LingTree shows what the tree looks like. The tree can be saved in up to two different file formats so that it can be included in papers or web pages.

LingTree looks like what is shown in (1).

(1)



The file name where the tree information is stored is shown in the upper left part of the window.¹ You key the description in the top pane, process the description, and then the tree appears in the bottom pane. See section 2 for information on keying a tree description and processing a tree.

¹You can have **LingTree** show the full path of the file name. See section 4.5.

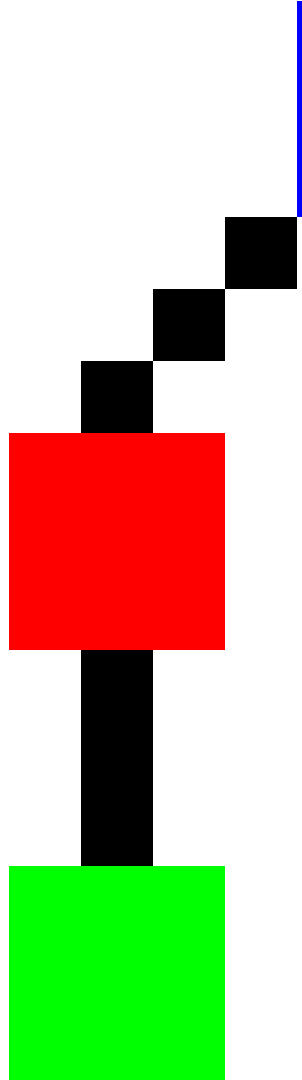
You can drag the edges of the window to make it larger. You can also drag the bar between the panes to make either the tree display or the description larger (or smaller).

2 Keying a tree description

You key in a tree description in the top pane. To show the corresponding tree in the bottom pane, you "process" the description in any one of four ways:

- by clicking the "process the tree description tool bar button" (
- by clicking on the **Tree** menu item and then **Process Description**;
- by typing **Ctrl-D** (holding the **Ctrl** key down while pressing the **D** key); or
- by using the **Settings** menu item / **Draw tree as you type** option. This will draw the tree for each keystroke you use in the tree description.² (See section 4.2.)

²When you key an opening parenthesis, it also automatically inserts a space and a closing parenthesis.



All four do the same processing of the tree description and are equivalent in effect.

2.1 Basic tree description

A tree description consists of a set of nodes enclosed within parentheses. The parentheses may be nested according to the tree structure. Whenever you key a closing parenthesis ')', the matching opening parenthesis will be selected briefly. Similarly, when you key an opening parenthesis '(', any matching closing parenthesis will be selected briefly. (If you have selected the **Settings** menu item / **Draw tree as you type** option, the closing parenthesis is inserted automatically when you key an opening parenthesis.) This is to help you match nested opening and closing parentheses. You can control how long the matching parenthesis is selected; see section 4.4.

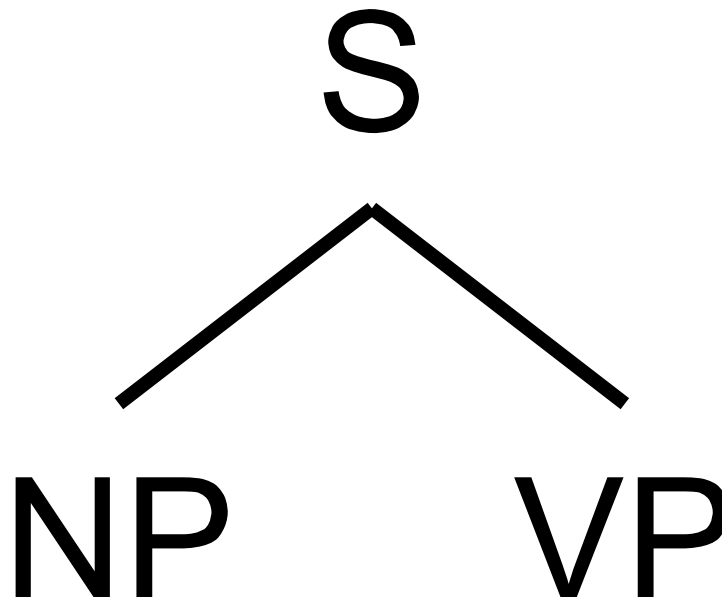
When keying a tree description, you can force a new line to occur by pressing the **Enter** key. You can also enter spaces and even tabs.³

Here are two examples of how to key trees.

When you key what is in (2), it shows up as in (3).

(2) (S (NP) (VP))

(3)



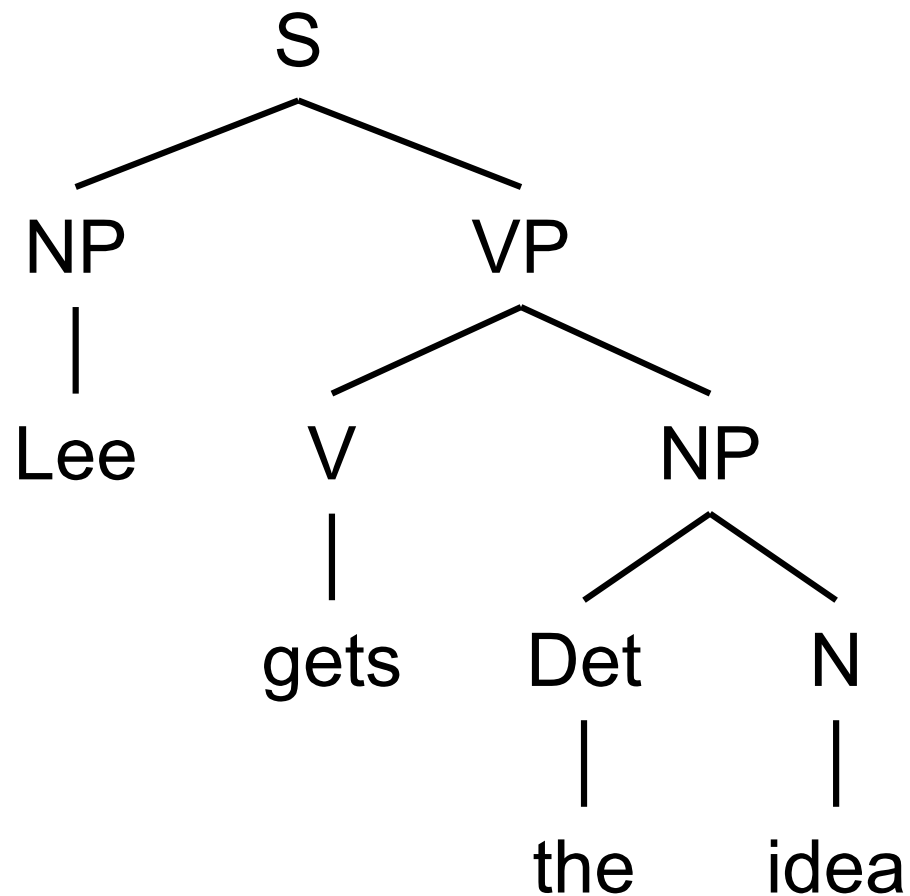
And if you key what is in (4)

(4) (S (NP (Lee)) (VP (V (gets)) (NP (Det (the)) (N (idea)))))

³Note that while you can use tabs, [LingTree](#) does not support setting tab stops to particular columns in the description pane.

it will produce what is in (5).

(5)



You may set font and color information for the nodes. See section 3.5. When you do, the font and color will show in the description pane.

2.2 Indicating node types

[LingTree](#) has four node types:

1. Non-terminal (2.2.1)
2. Lexical (2.2.2)
3. Gloss (2.2.3)
4. Empty element (2.2.4)

At most one node type can be indicated. If you key two or more types, an error message will be shown instead of the tree diagram.

2.2.1 Indicating a non-terminal node

The default type is what we call a non-terminal node. The thinking is that a typical full syntactic tree would have non-terminal nodes (for the syntactic constituents) and each branch would end with a lexical node and possibly also a gloss node.

Note that to indicate that a node is a non-terminal node, you do not have to do anything. Just key the parentheses around it.

You can control the formatting of non-terminal nodes. See section 3.5.

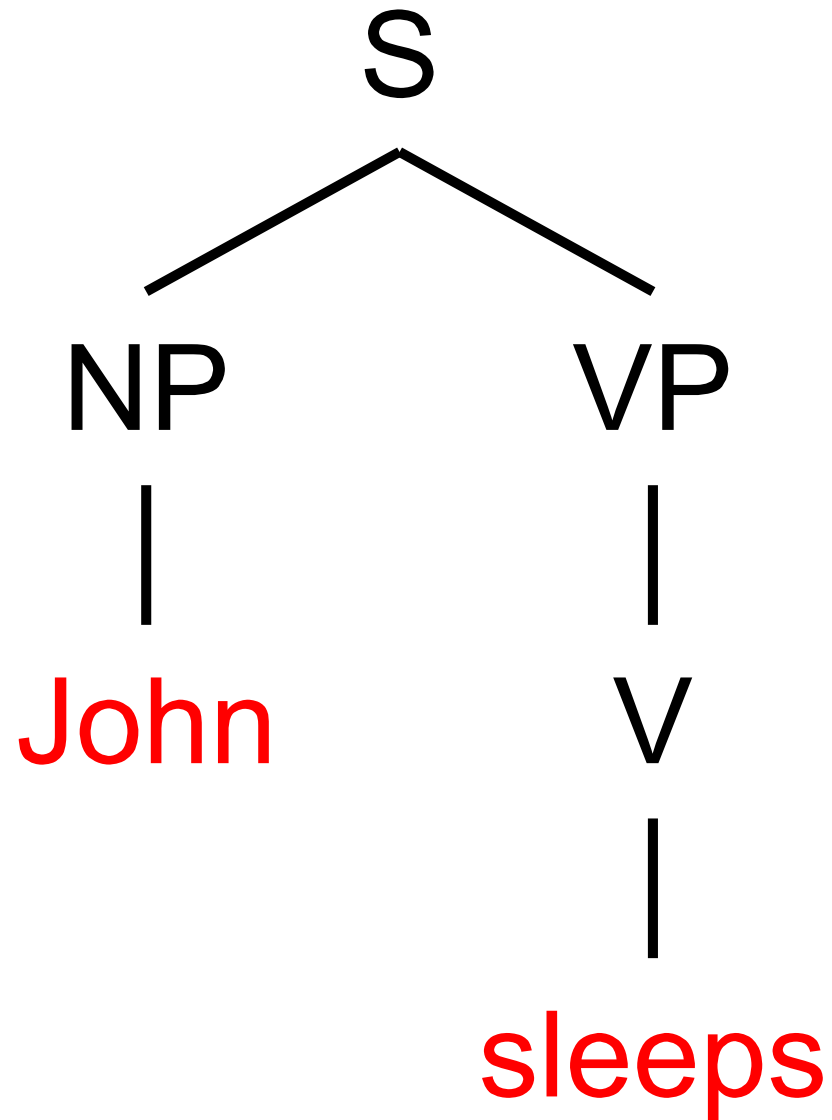
2.2.2 Indicating a lexical node

To mark a node in the tree as a lexical item, key `\L` before it (i.e., right after the opening parenthesis). While it is not required to include a space between the `\L` and the lexical content that follows it, it can make the description easier to read. For example, suppose we have set the parameters so that lexical nodes come out in red (see section 3.5). Then keying what is in (6)

```
(6)  (S (NP (\L John)) (VP (V (\L sleeps))))
```

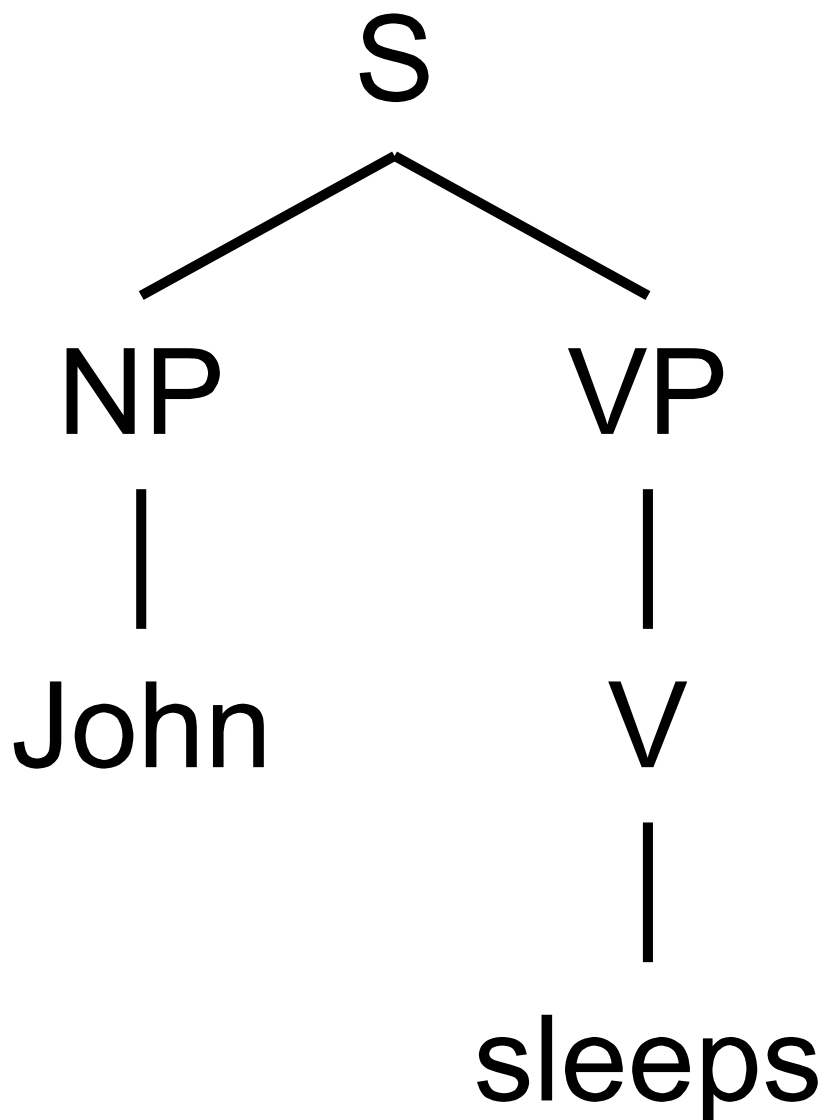
will produce what is in (7).

(7)



If you do not key the `\L`, then it will look like what is in (8).

(8)



That is, every node is considered to be a non-terminal node.

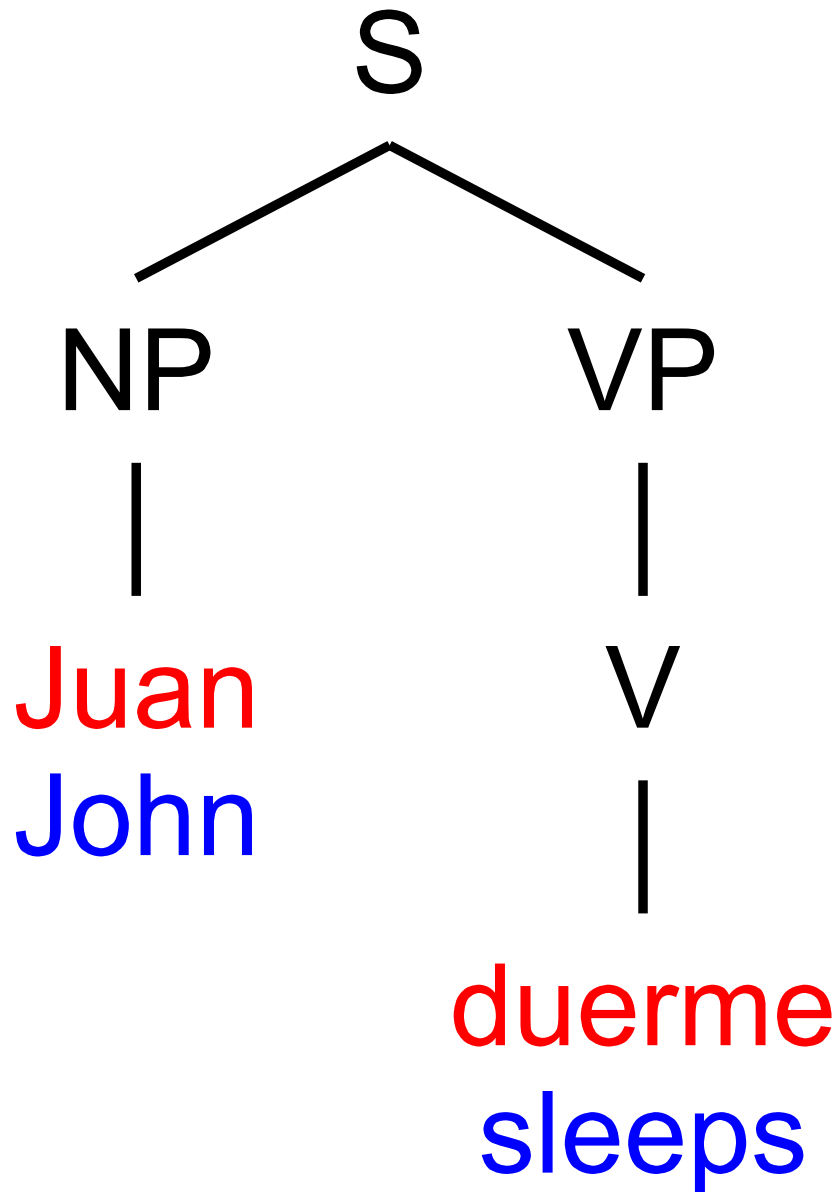
2.2.3 Indicating a gloss node

To mark a node in the tree as a gloss node, key `\G` before it (i.e., right after the opening parenthesis). While it is not required to include a space between the `\G` and the gloss content that follows it, it can make the description easier to read. If you are going to mark glosses, you should also mark lexical nodes. For example, suppose we have set the parameters so that lexical nodes come out in red and gloss nodes come out in blue (see section 3.5). Then keying what is in (9)

```
(9) (S (NP (\L Juan (\G John))) (VP (V (\L duerme (\G
sleeps)))))
```


produces what is in (10).

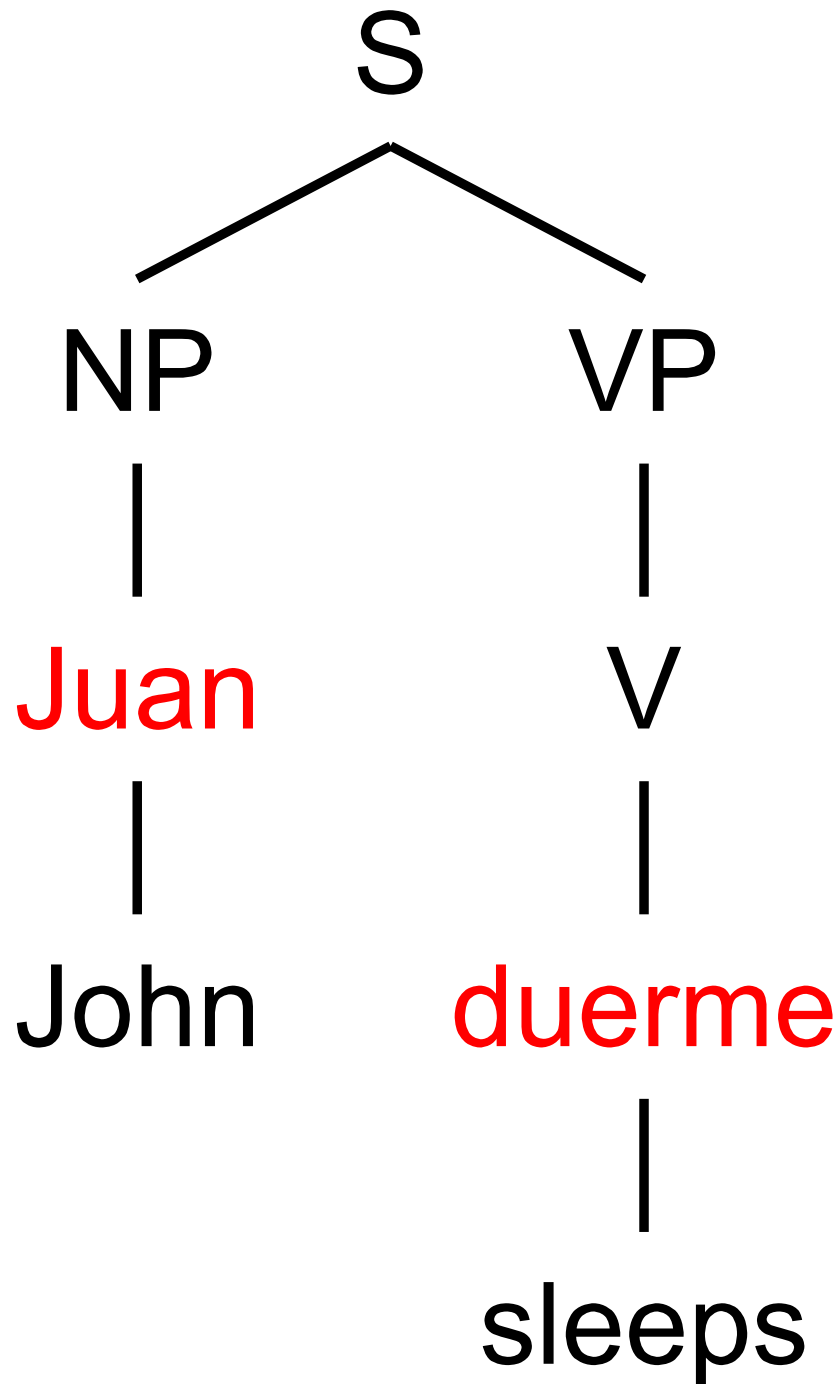
(10)



Notice that there are no lines between the lexical nodes and their respective glosses and that the glosses are relatively close to the lexical node.

If you do not key the **\G** , then it would look like what is in (11).

(11)



Notice that there is a line between the lexical node and their daughter node. The blue color also does not show for the daughter nodes.

2.2.4 Indicating an “empty element” node

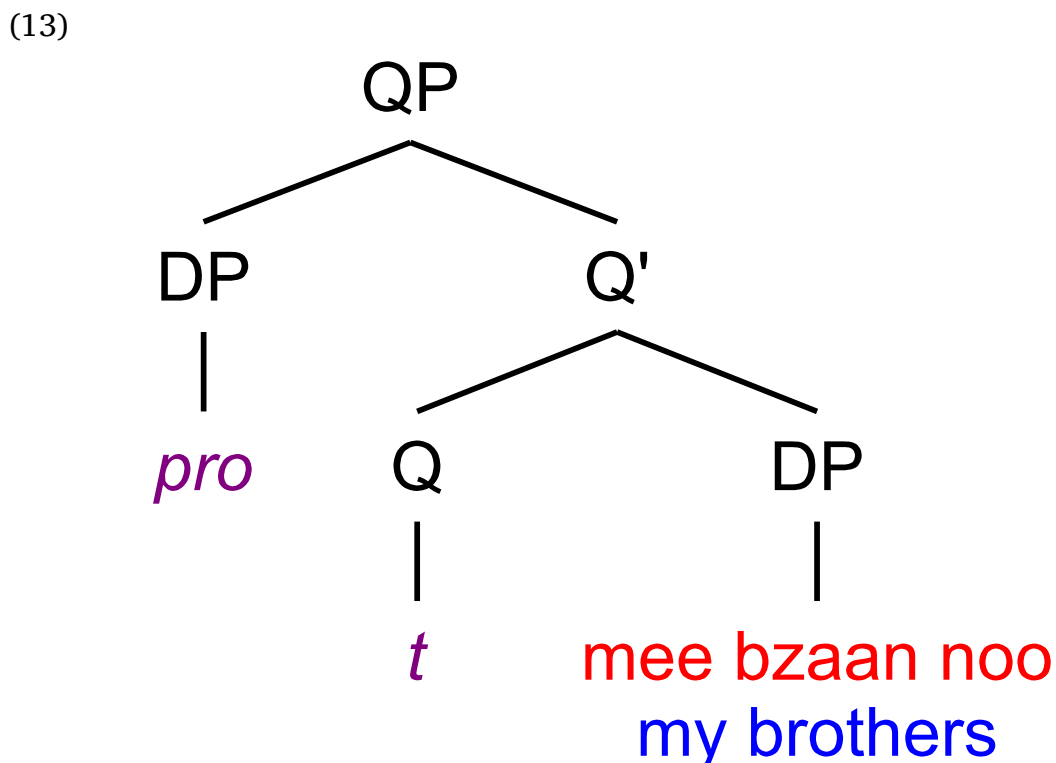
In some syntactic theories which show movement, the convention is to indicate a “trace” of where the moved constituent used to be. You can use an empty element node for traces.

Also, some syntactic theories want to show “empty pronouns” where a non-overt pronoun is considered to occur in the tree structure. You can use an empty element for an empty pronoun node.

To mark a node in the tree as an empty element node, key `\E` before it (i.e., right after the opening parenthesis). While it is not required to include a space between the `\E` and the empty element content that follows it, it can make the description easier to read. Conventionally, if you have an empty element, then there is no gloss daughter node. For example, suppose we have set the parameters so that empty element nodes come out in purple and italic (see section 3.5). Then keying what is in (12)

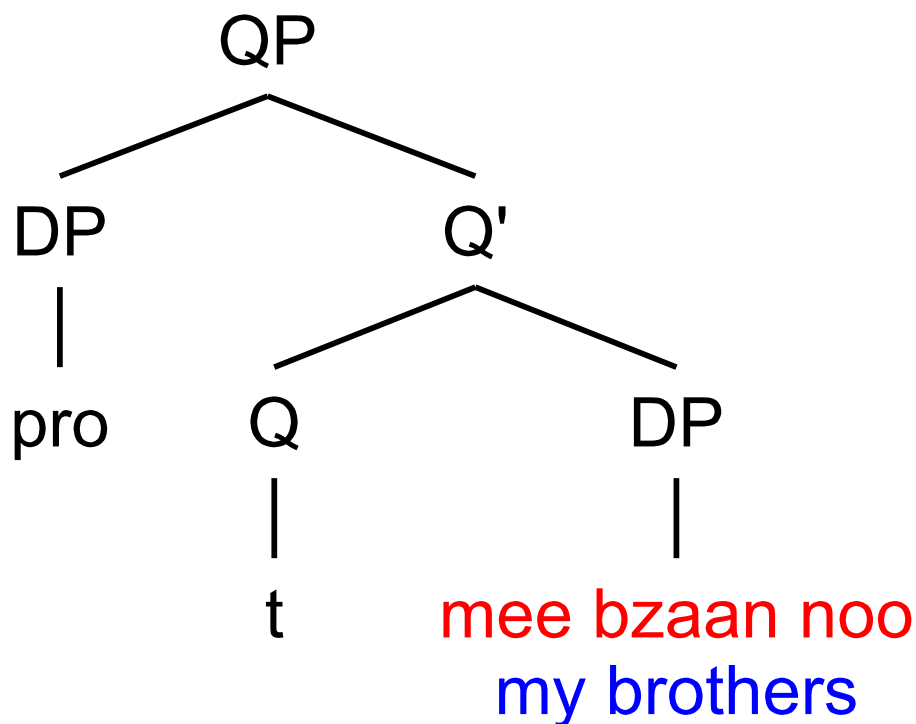
(12) (QP (DP (\E pro)) (Q' (Q (\E t)) (DP (\L mee bzaan noo
(\G my brothers)))))

produces what is in (13).



If you do not key the `\E` , then it would look like what is in (14).

(14)



Notice that the empty pronoun and trace nodes are treated as non-terminal nodes.

2.3 Indicating line types

Besides the node types, there are three line types to control what kind of line should appear over a node:

1. Regular line (2.3.1)
2. Triangle (2.3.2)
3. Omit a line (2.3.3)

At most one line type can be indicated. If you key two or more types, an error message will be shown instead of the tree diagram.

2.3.1 Regular line

The default is to produce a regular line over a node. You do not have to do anything for this to happen. We have already seen examples where a line occurs over nodes.

2.3.2 Indicating a triangle (a phrase)

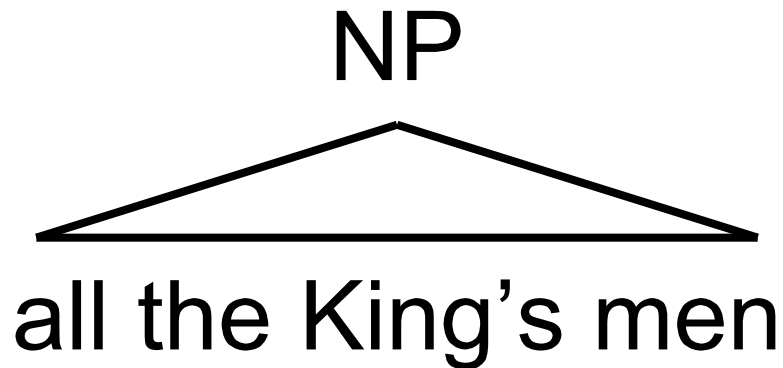
To mark a node as a phrase by using a triangle above it, key `\T` before it (i.e., right after the opening parenthesis). While it is not required to include a space

between the `\T` and whatever follows it, it can make the description easier to read. For example, keying what is in (15)

(15) (NP (`\T` all the King's men))

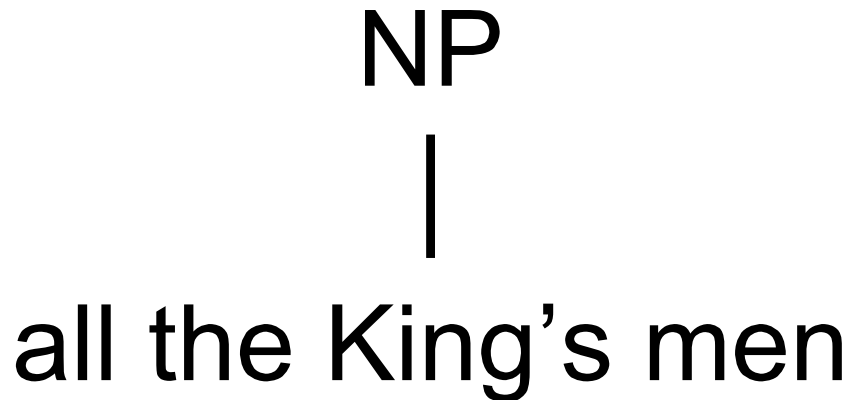
will produce what is in (16).

(16)



If you do not key the `\T`, then it will look like what is in (17).

(17)

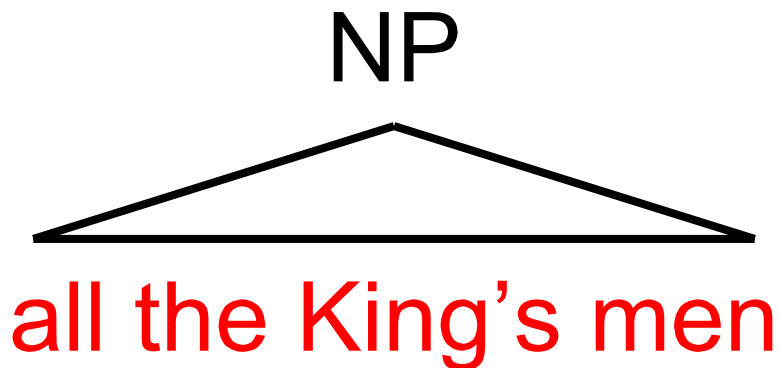


You may key both a triangle and either a lexical item or an empty element for a given node. The order in which you key the `\T` and either the `\L` or the `\E` is not crucial. Either order will work. If you key what is in (18),

(18) (NP (`\T \L` all the King's men))

it will look like what is in (19).

(19)



On the other hand, if you key the `\T` and `\L` in the other order, as in (20), it will also look exactly like it does in (19).

(20) `(NP (\L \T all the King's men))`

You may find that it is easier to read and think about if you key the `\T` first since it has to do with what occurs over the node, while the `\L` or `\E` has to do with what is in the node.

2.3.3 Omitting a line

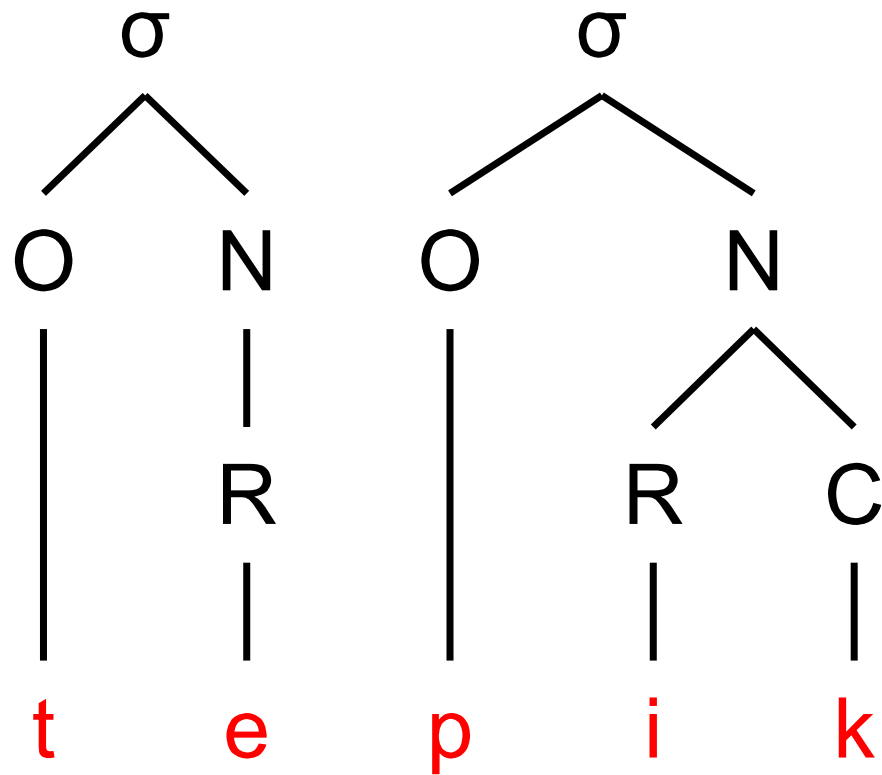
To not have a line show above a node, key `\O` before it (i.e., right after the opening parenthesis). While it is not required to include a space between the `\O` and whatever follows it, it can make the description easier to read. For example, keying what is in (21)⁴

(21) `((\O σ (O (\L t)) (N (R (\L e)))) (\O σ (O (\L p)) (N (R (\L i)) (C (\L k)))))`

will produce what is in (22) (assuming that the "flat" structure parameter is also chosen; see 3.1).

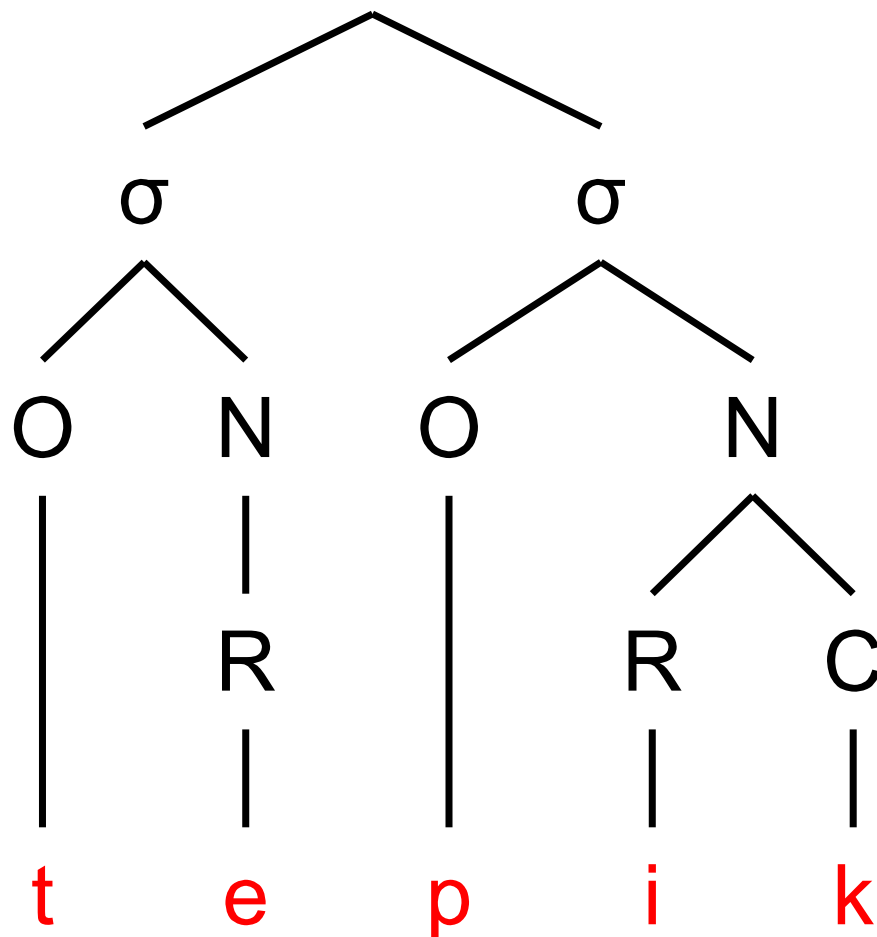
⁴In this particular tree diagram, "O" is an abbreviation for "Onset."

(22)



If you do not key the $\backslash o$ symbols, then it will look like what is in (23).

(23)



The difference between (22) and (23) is the unlabeled node at the top.

You may combine the omit lines command with either a lexical item or an empty element for a given node. The order in which you key the \O and either the \L or the \E is not crucial. Either order will work. You may find that it is easier to read and think about, though, if you key the \O first since it has to do with what occurs over the node, while the \L or \E has to do with what is in the node.

2.4 Indicating the node content

In the vast majority of cases, you merely key the content of the node. The only exceptions are for inserting subscripts and/or superscripts, indicating abbreviations and for keying parentheses that will occur as content.

2.4.1 Indicating subscripts or superscripts

You can include simple subscripts and superscripts within a tree node.

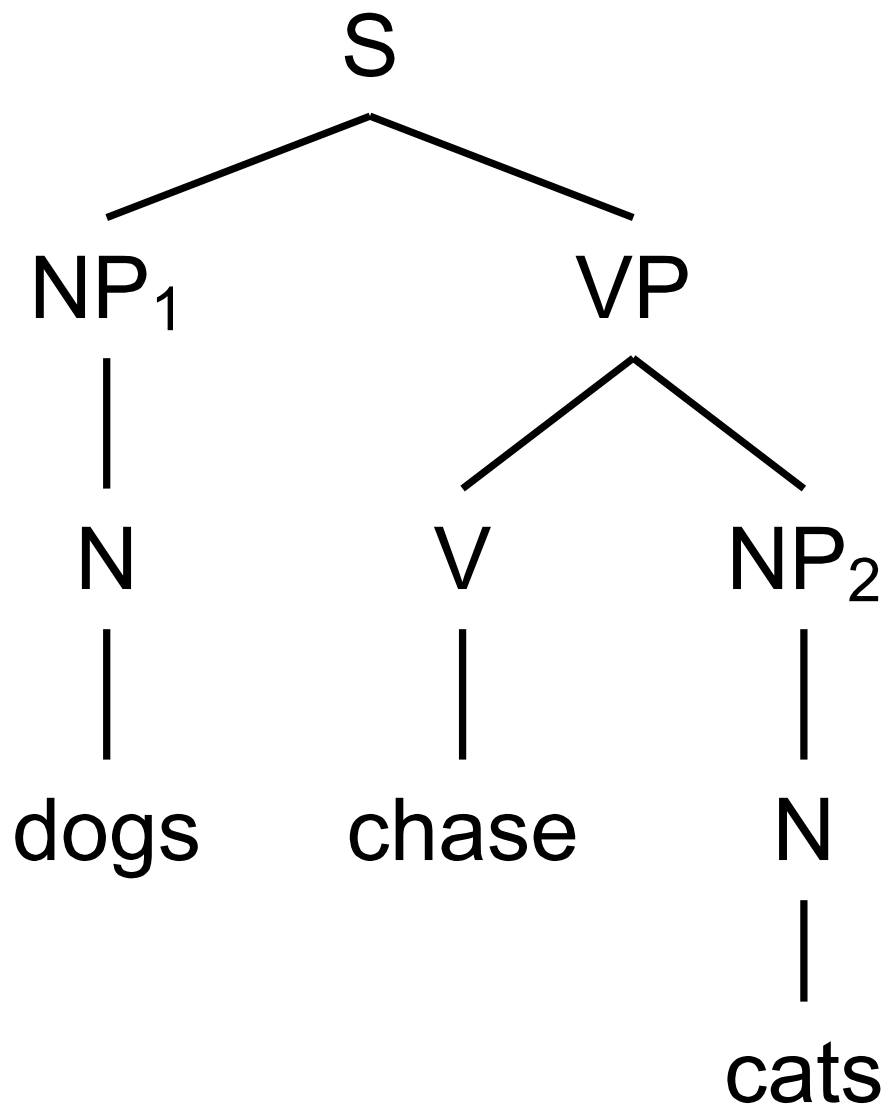
2.4.1.1 Subscript

To have a subscript show up at the end of some text, key **/s** before the subscript text. Note that this has a forward slash, not a backslash. It also uses a lowercase **s**, not uppercase (uppercase is used for a superscript). You do not have to include a space between the **/s** and whatever follows it. For example, keying what is in (24)

(24) (S (NP/s1 (N (dogs))) (VP (V (chase)) (NP/s2 (N (cats)))))

will produce what is in (25).

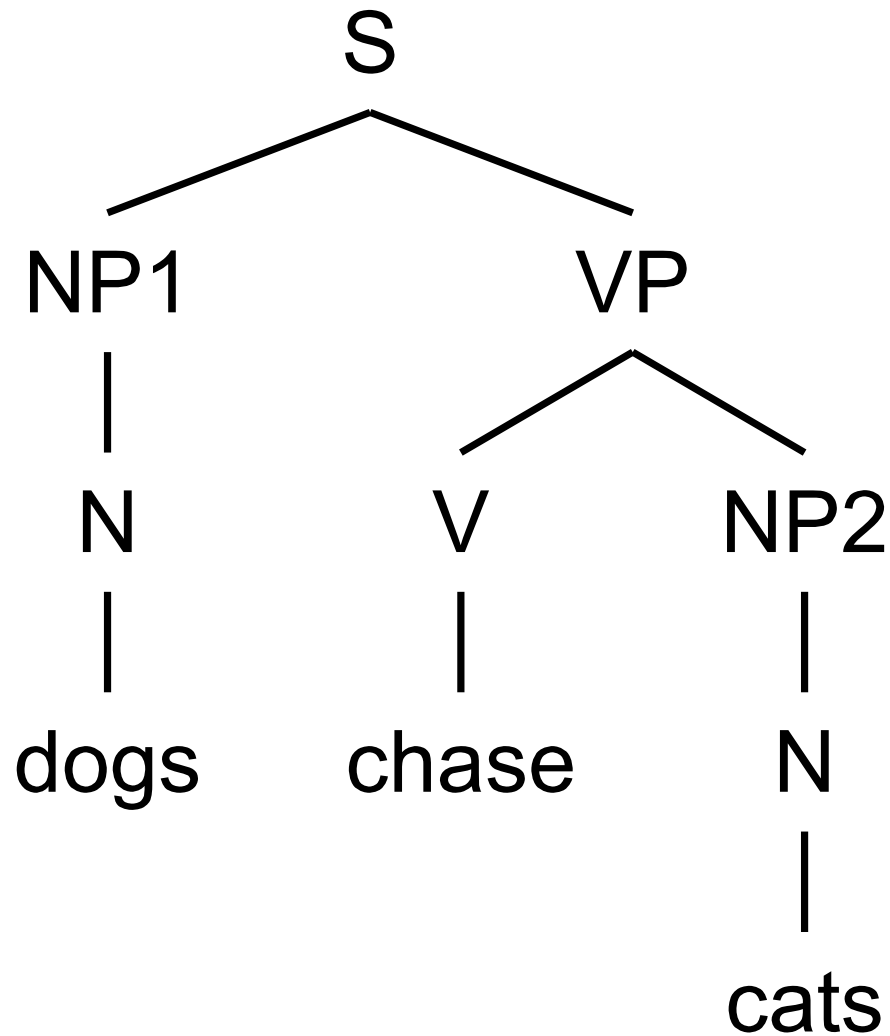
(25)



Note that the subscript will use the same font family and color that the node it is in uses. So if the node is a non-terminal, then it uses the non-terminal font family and color. Similarly for lexical, gloss, and empty node types. The subscript text also is always in regular style (i.e., neither bold nor italic). To get an italic subscript, see section 2.4.1.2.

If you do not key the /**s**, then it will look like what is in (26).

(26)



2.4.1.2 Subscript in *italic*

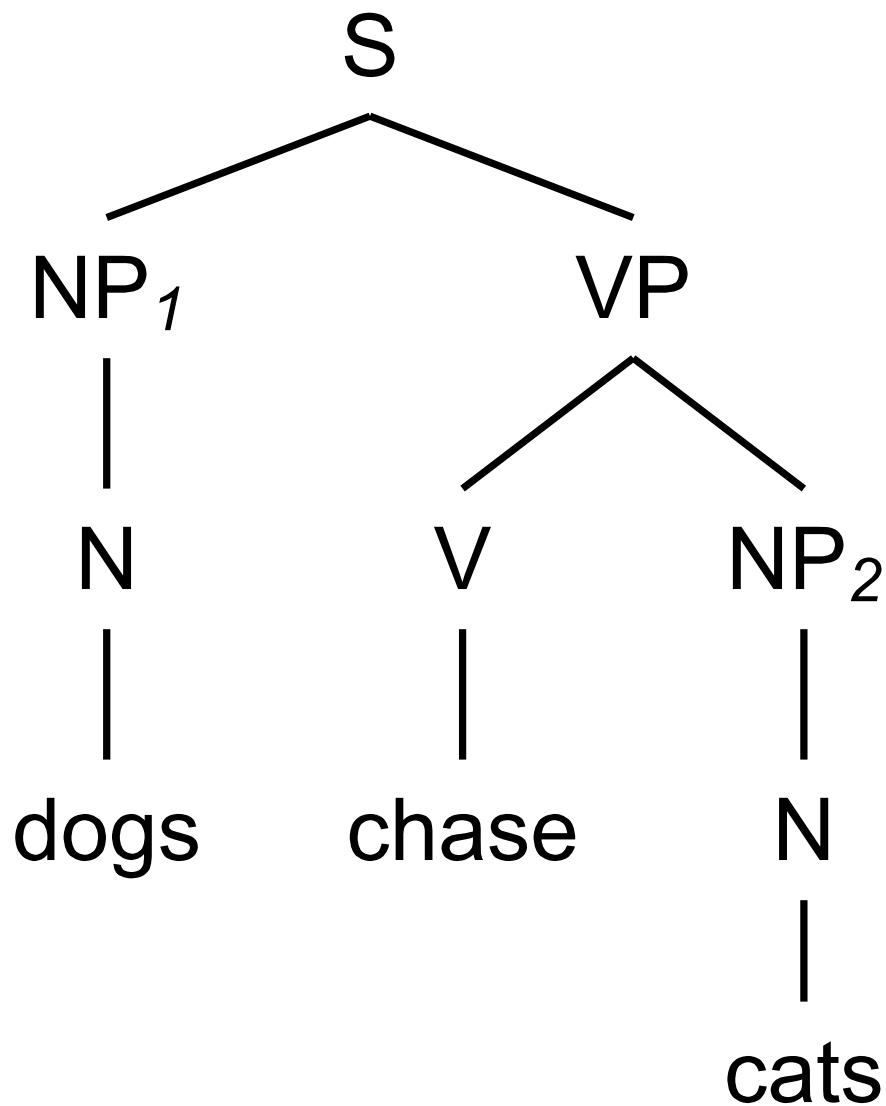
Keying the */s* before the subscript text results in the subscript coming out in regular style (i.e., neither bold nor italic). If you need a subscript to be italic, use */_*, instead. The only difference between */s* and */_* is that the first is rendered in regular style and the latter is rendered in italic.

For example, keying what is in (27)

(27) (S (NP/_1 (N (dogs))) (VP (V (chase)) (NP/_2 (N (cats)))))

will produce what is in (28).

(28)



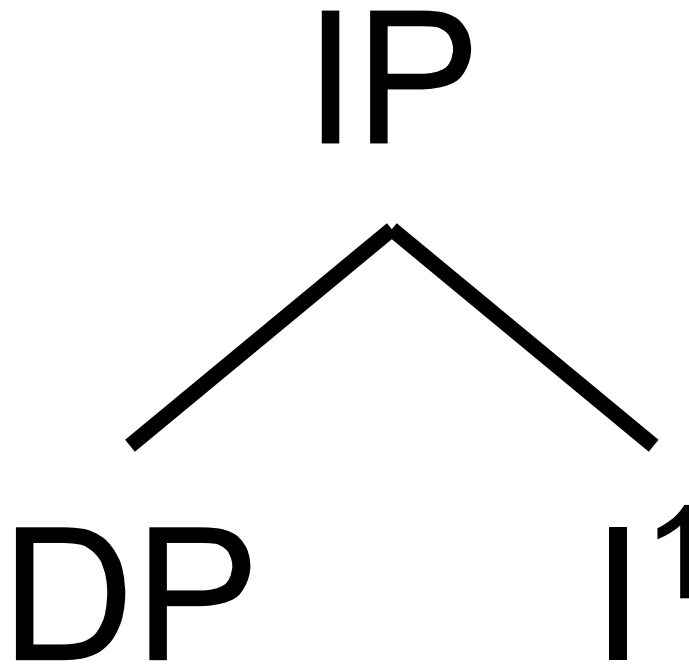
2.4.1.3 Superscript

To have a superscript show up at the end of some text, key `/s` before the superscript text. Note that this has a forward slash, not a backslash. It also uses an uppercase **s**, not lowercase (lowercase is used for a subscript). You do not have to include a space between the `/s` and whatever follows it. For example, keying what is in (29)

(29) (IP (DP) (I/S1))

will produce what is in (30).

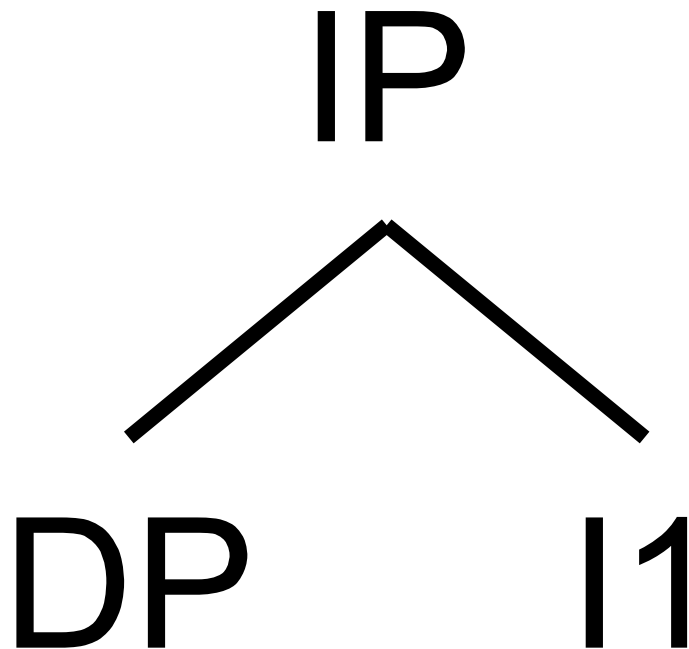
(30)



Note that the superscript will use the same font family and color that the node it is in uses. So if the node is a non-terminal, then it uses the non-terminal font family and color. Similarly for lexical, gloss, and empty node types. The superscript text also is always in regular style (i.e., neither bold nor italic). To get an italic superscript, see section 2.4.1.4.

If you do not key the /*S*, then it will look like what is in (31).

(31)



2.4.1.4 Superscript in italic

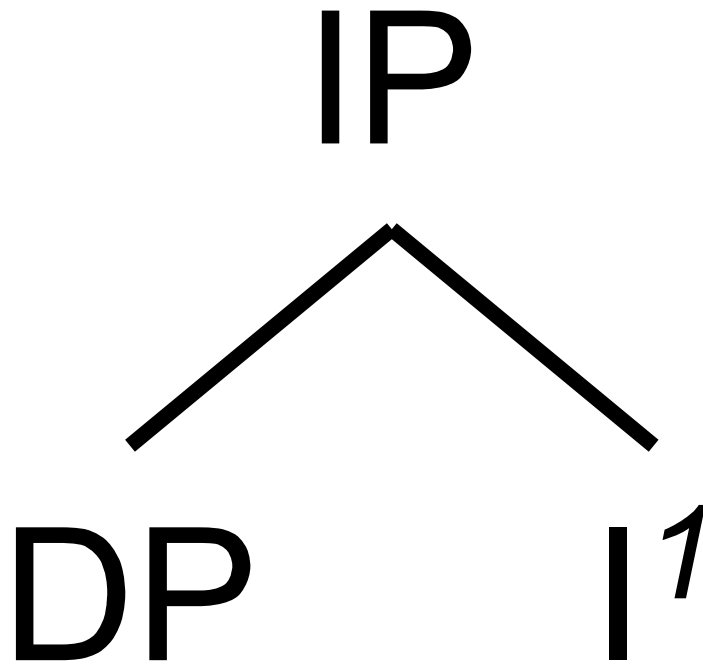
Keying the /s before the superscript text results in the superscript coming out in regular style (i.e., neither bold nor italic). If you need a superscript to be italic, use /[^], instead. The only difference between /s and /[^] is that the first is rendered in regular style and the latter is rendered in italic.

For example, keying what is in (32)

(32) (IP (DP) (I/[^]1))

will produce what is in (33).

(33)



2.4.1.5 Combining subscripts and superscripts can be done

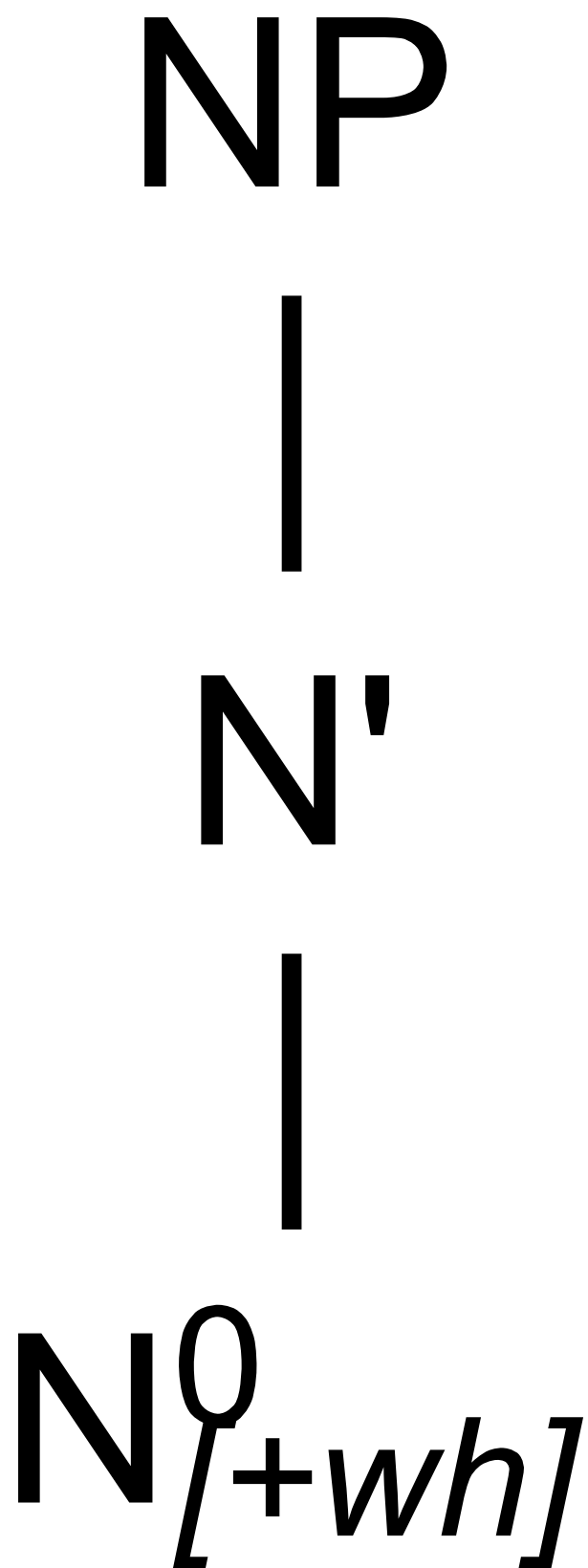
Unlike the older version of [LingTree](#), you can have both a subscript and a superscript on the same node. It does not matter which order you key the subscript and superscript. Either will work.

For example, if you key what is in (34)

(34) (NP (N' (N/S0/_ [+wh])))

the result will look like (35).

(35)



2.4.2 Indicating an abbreviation

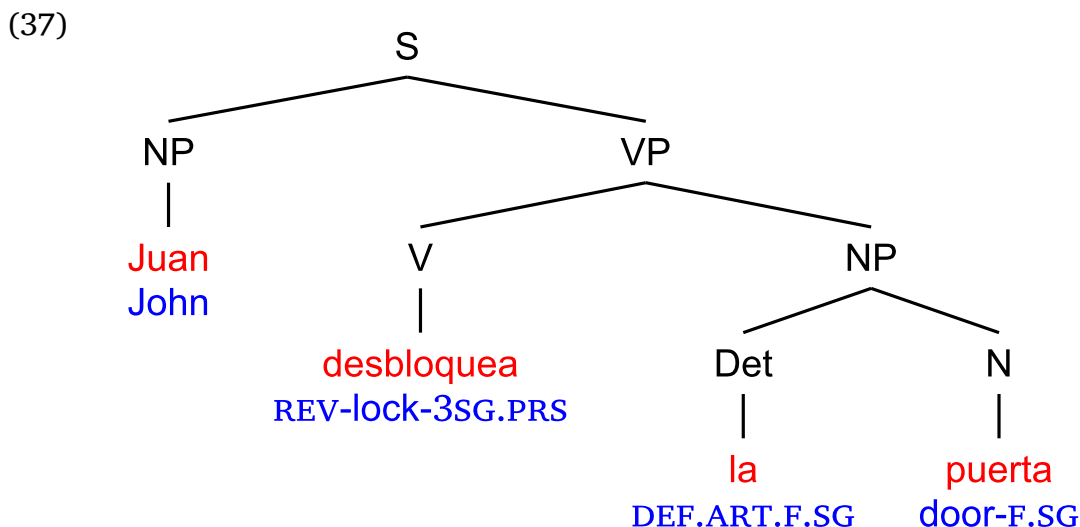
Since it is conventional to use small caps for abbreviations within glosses, you can set the font for abbreviations to use a small caps font (see section 3.5) and then demarcate an abbreviation within a node. While it is expected that you would only use abbreviations within a gloss node, *LingTree* does not enforce this. You can demarcate an abbreviation within any node type.

To indicate an abbreviation within the text of a node, key **/a** before the abbreviation and key **/A** after it. You can have any number of abbreviations within the text of a node.

For example, keying what is in (36)

```
(36) (S (NP (\L Juan (\G John)))
      (VP (V (\L desbloquea (\G /arev/A-lock-/a3sg.prs/A)))
        (NP (Det (\L la (\G /adef.art.f.sg/A)))
          (N (\L puerta (\G door-/af.sg/A))))))
```

produces what is in (37).



Notice that the abbreviations are in small caps and, unlike subscripts and superscripts, may appear anywhere within the text of the node.

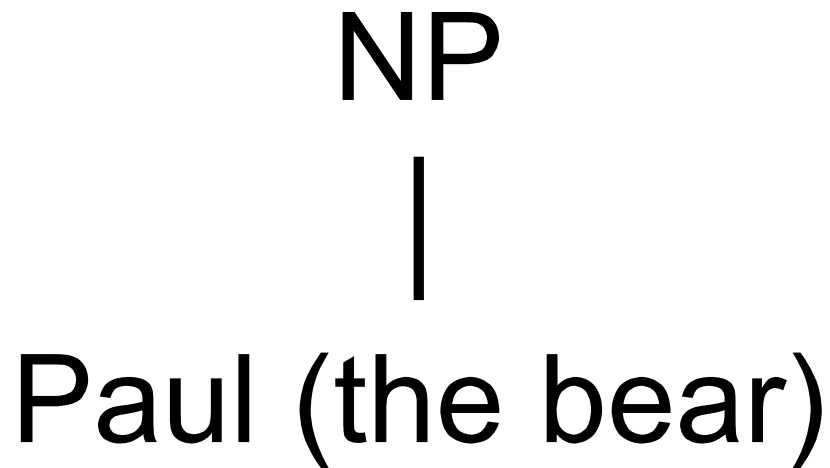
2.4.3 Handling parentheses within a node

If you need to put an opening or closing parenthesis in your tree, quote the parenthesis by using a backslash immediately before it. For example, if you key what is in (38)

```
(38) (NP (Paul \(the bear\)))
```

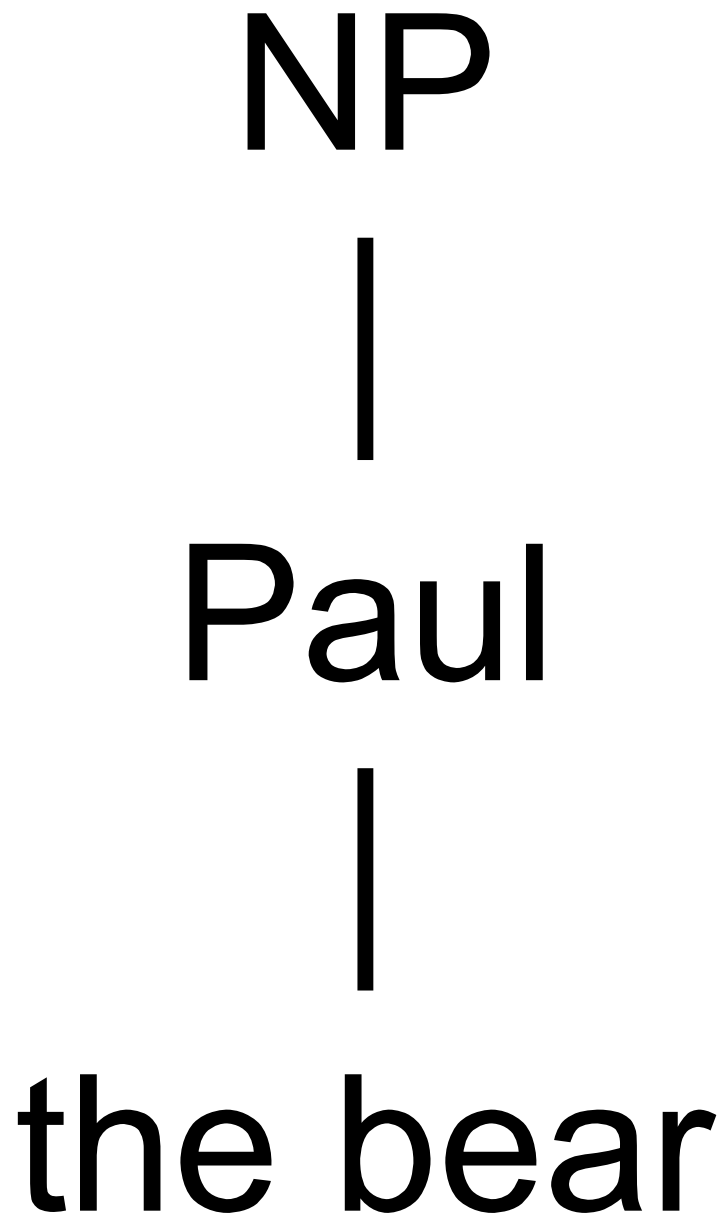
it will come out as in (39).

(39)



If you do not quote the parentheses with a backslash, then it will look like what is in (40) because the parentheses are interpreted as a new tree subnode.

(40)



3 Formating a tree

You can adjust the positioning of the tree and its nodes as well as choose the fonts, sizes, colors and keyboards for the non-terminal, lexical, gloss and empty nodes (and abbreviations). You can also set a background color and determine the color and thickness of the lines in the trees. In addition, you can have the tree

show all of its lexical and gloss items at the same vertical position (which we refer to as using a “flat” view of the tree).

All of these parameters are available on the **Format** menu item.

Please note that these settings are associated with each individual tree diagram. Thus, you can set them to make a given tree diagram look its best.

3.1 Showing “flat” trees

The **Use flat tree** item in the **Format** main menu controls how the lexical and gloss nodes are displayed. It will have a check mark before it if flat trees are to be drawn. There also is a button on the toolbar for controlling flat trees. Example (41) indicates what this button looks like in its two states.

(41) **Show as flat** **Show regularly**

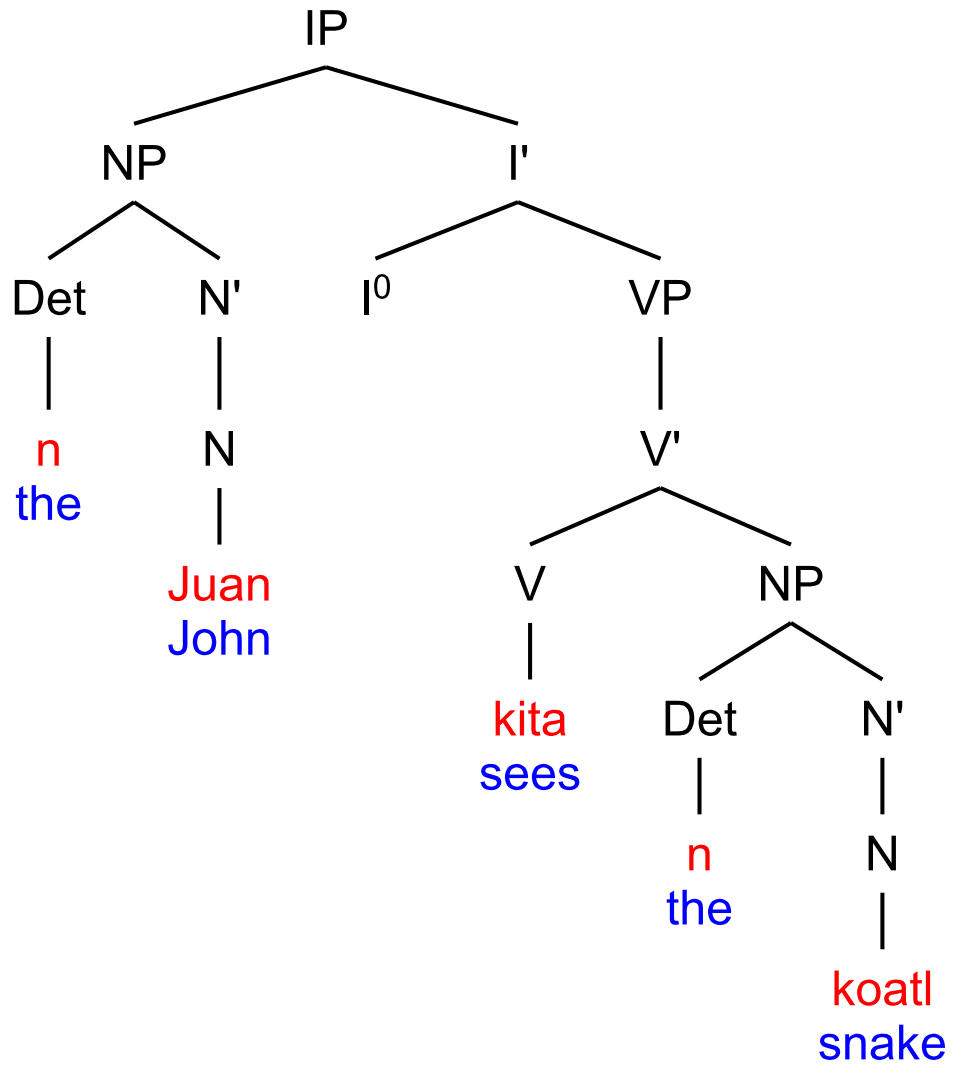


For example, without using the flat mode, the description in (42)

(42) (IP (NP (Det (\L n (\G the))) (N' (N (\L Juan (\G John))))) (I' (I/S0) (VP (V' (V (\L kita (\G sees))) (NP (Det (\L n (\G the))) (N' (N (\L koatl (\G snake)))))))))

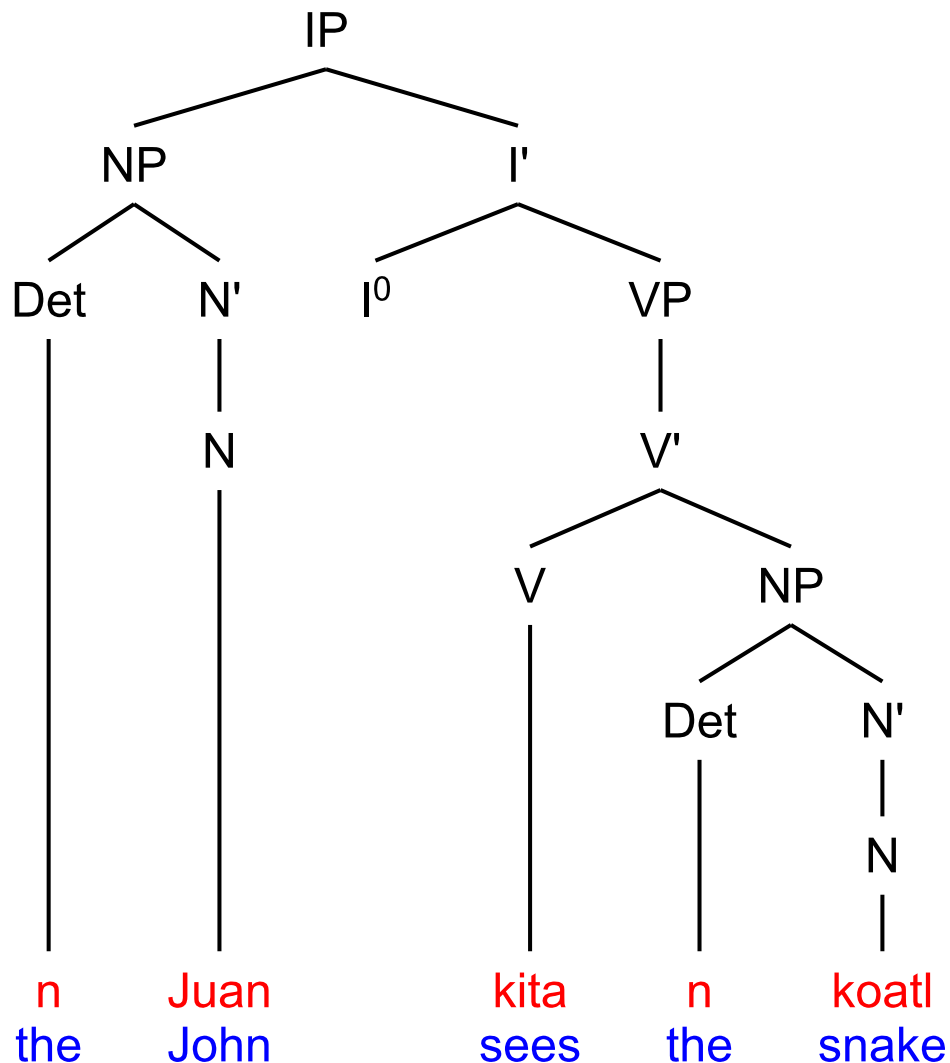
will appear as in (43) (assuming that lexical nodes are set to be in red and gloss nodes are in blue).

(43)



When using the flat mode, however, it will look like the tree in (44).

(44)

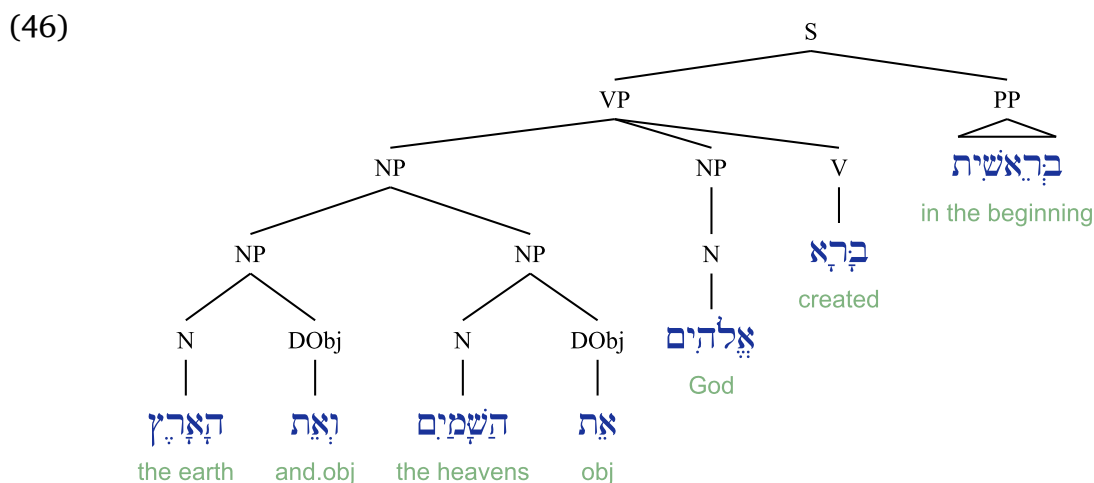


Note that if you use the “Flat” option while using triangles (see section 2.3.2), you will need to also use either the `\L` or the `\G` at the same time as using the `\T`. Otherwise, the triangle node may not be at the same level as other “flattened” nodes.

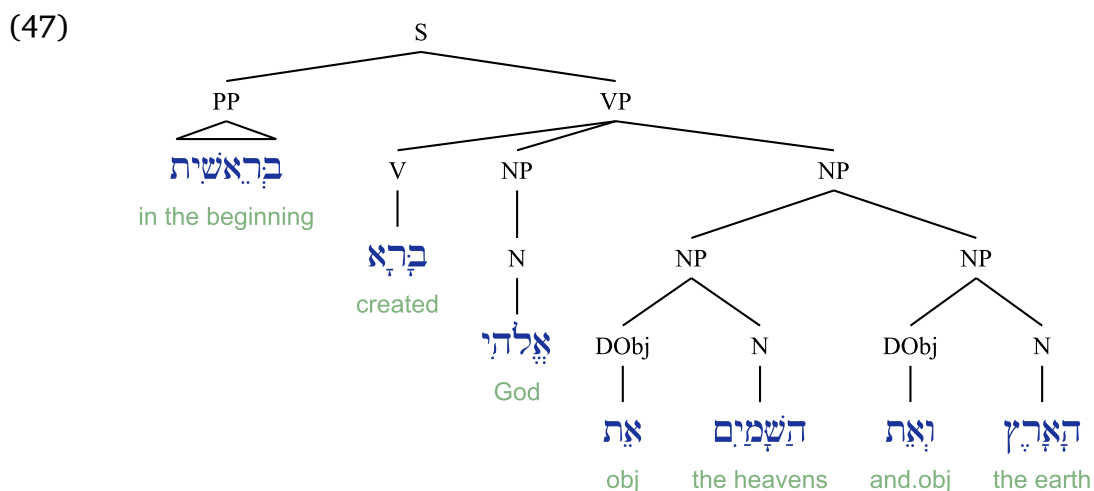
3.2 Draw a tree right-to-left

The second item under the main **Format** menu item is **Draw tree right-to-left**. When this has a check mark before it, the tree will be drawn from right-to-left. For example, if the non-terminal, lexical, and gloss fonts and colors have been set as shown in example (46) below, then the description given in (45) looks like example (46) when the **Draw tree right-to-left** is checked.

(45) (S (PP (\T\L בְּרֵאשִׁית (\G in the beginning))) (VP (V (\L בָּרָא (\G created))) (NP (N (\L אֱלֹהִים (\G God)))) (NP (NP (DObj (\L אֶת (\G obj))) (N (\L הַשָּׁמַיִם (\G the heavens)))) (NP (DObj (\L וְאֵת (\G and.obj))) (N (\L הָאָרֶץ (\G the earth)))))))



The output looks like example (47) when the **Draw tree right-to-left** is left unchecked.

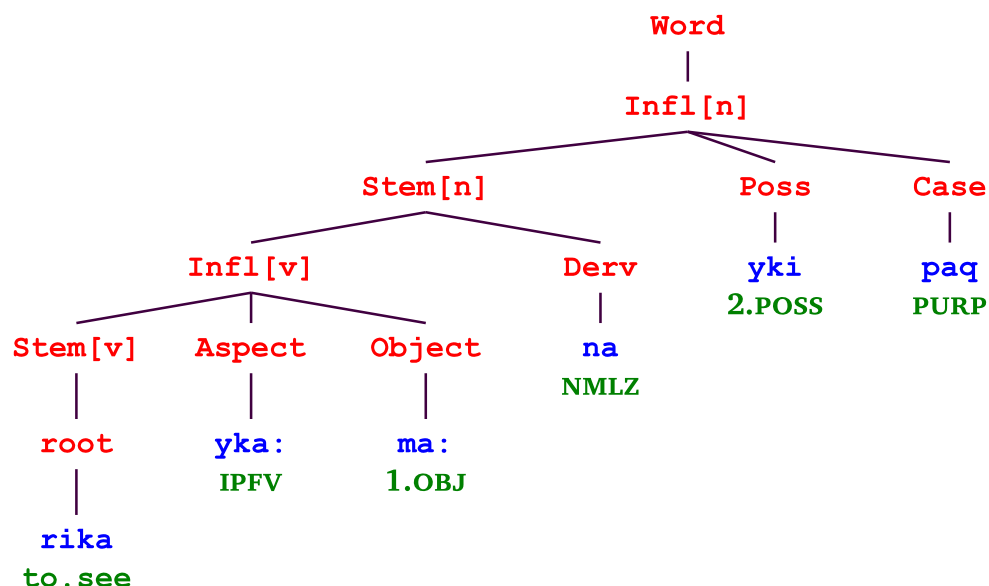


3.3 Use column-oriented algorithm

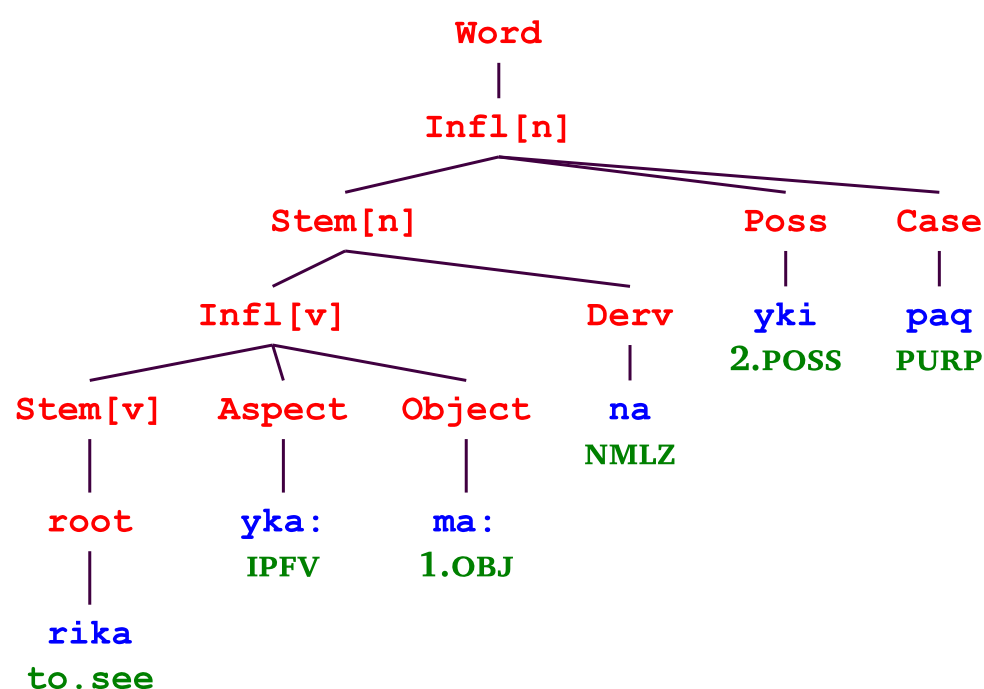
The third item under the main **Format** menu item is **Use column-oriented algorithm**. When this has a check mark before it, the tree will be drawn using an algorithm that figures out the column widths of each node and centers every node within its column. As a result, it places the nodes in a tighter position than the default algorithm (which is more of an edge-out algorithm). In fact, the default algorithm assumes that the text in higher nodes is shorter than the text in lower nodes. The column-oriented algorithm does not make this assumption.

Examples (48–49) illustrate the difference.

(48)



(49)

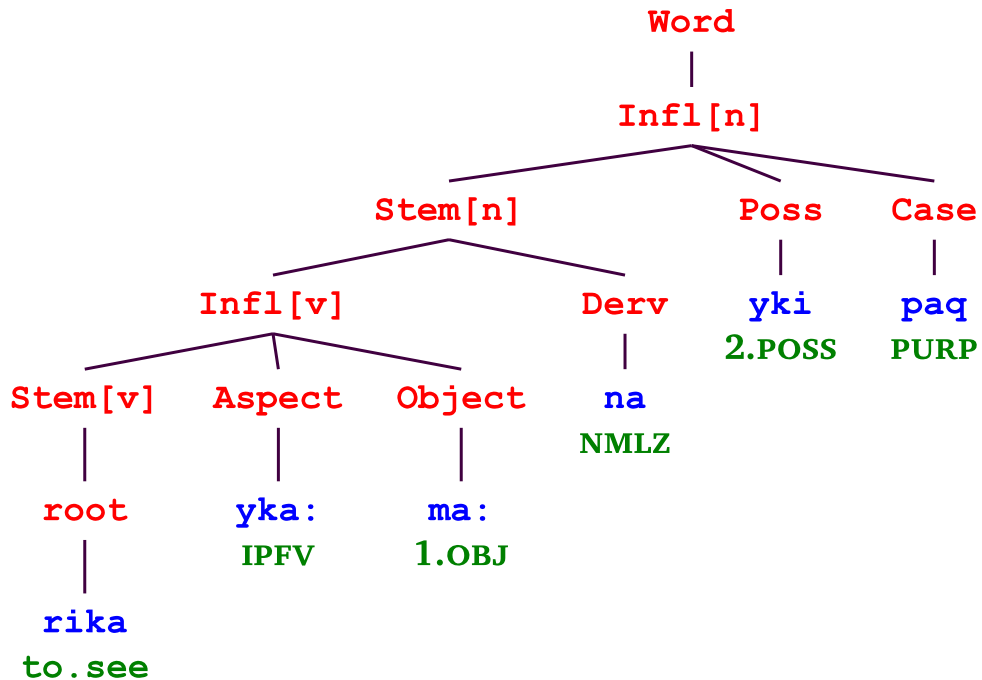


Notice how the higher nodes are centered over the entire tree in example (49) whereas they are not in example (48).

3.4 Center each node over its daughters

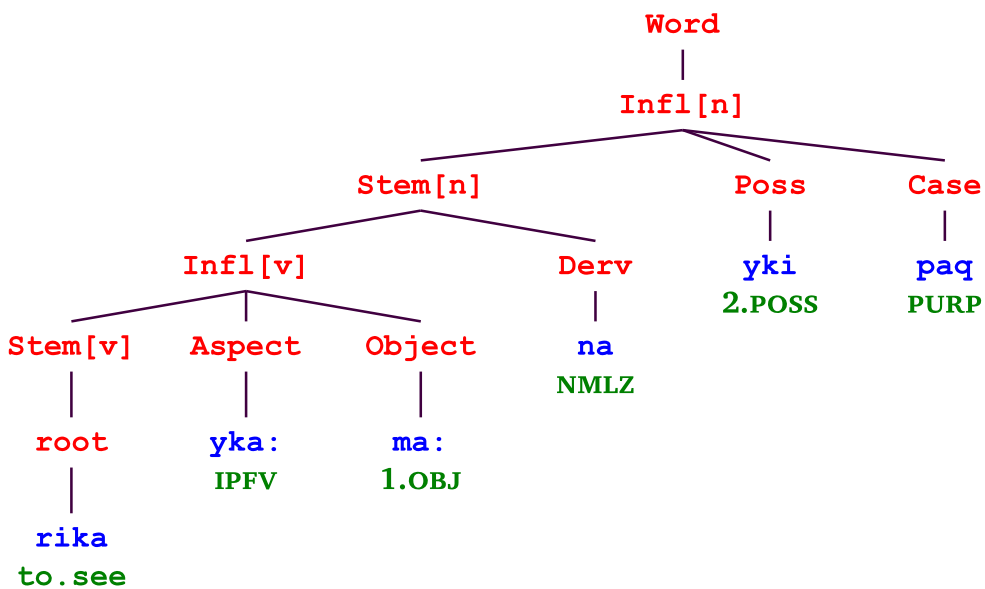
The next item under the main **Format** menu item is **Center each node over its daughters**. This option is only available when using the column-oriented algorithm (see section 3.3). When this has a check mark before it, the tree will be drawn using the column-oriented algorithm but will center each node over the space used by its daughter nodes. Example (50) shows how the sample tree in example (49) appears using this option.

(50)



Compare this to using the default algorithm as shown in example (51) (repeated from example (48)).

(51)



3.5 Setting the font and color information

The next portion of the main **Format** menu item is for setting font and color information for the four node types (and for abbreviations):

1. Non-terminal (2.2.1)
2. Lexical (2.2.2)
3. Gloss (2.2.3)

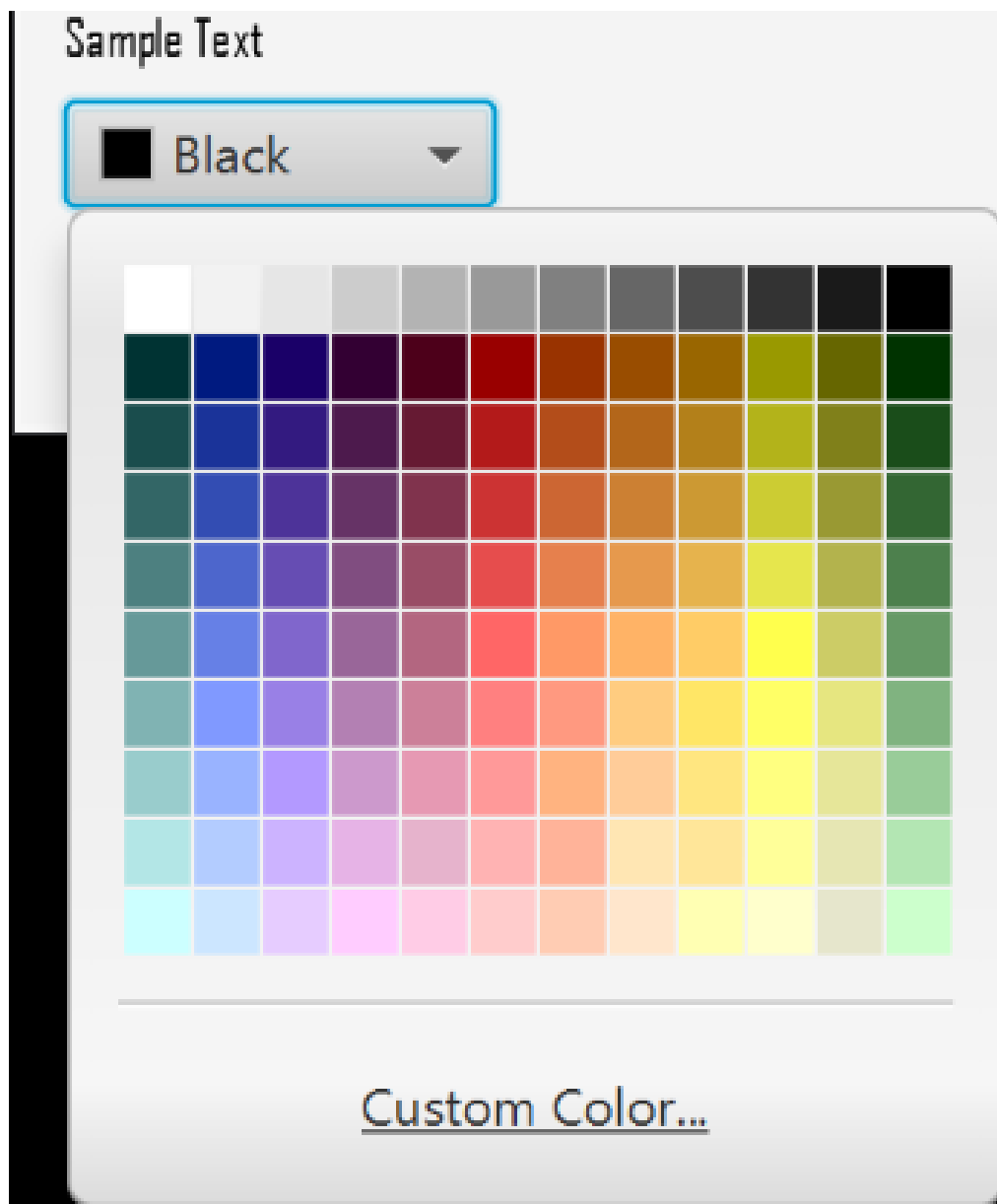
4. Abbreviation (2.4.2)⁵
5. Empty element (2.2.4)

When you select one of these, it brings up a dialog box showing font family, style, size and color options. The set of style options shown depends on what styles the selected font has. So if the font only has “Regular,” then only Regular will appear in the list of styles.

If you click in the Font list, then you can key a letter and it will go to the first font name that begins with that letter.

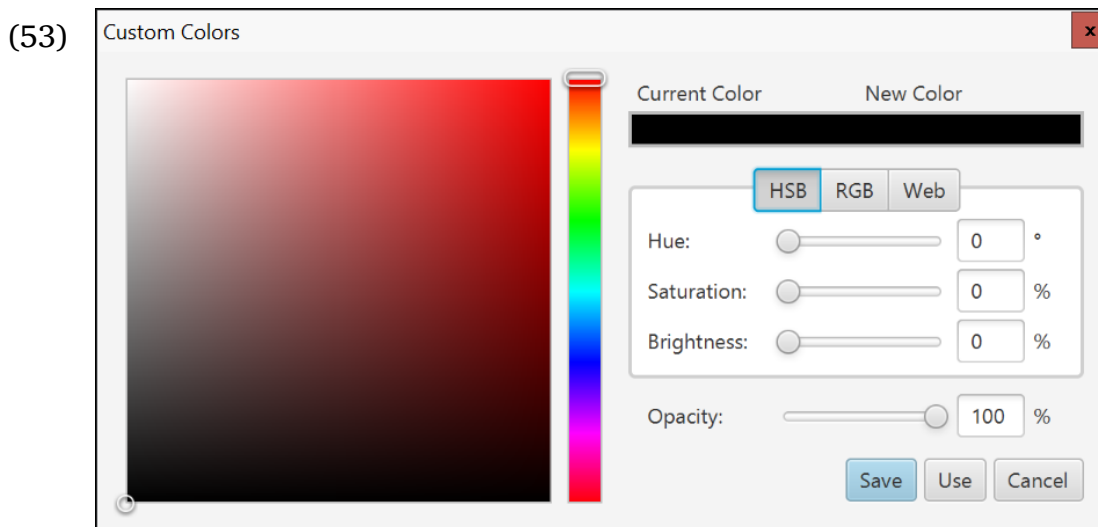
To get the color chooser, click on the drop-down box in the lower left portion of the dialog. It will look something like what is in (52).

(52)



⁵Whenever you open a [LingTree](#) document created before version 1.3.0, [LingTree](#) will automatically set the font information for abbreviation nodes to be the same as what was used for gloss nodes.

You can click on the color you want or you can click on the “Custom Color” link. Doing so gives you what is in (53).



If you click on the Web tab, you can key in a code for the color. A list of color names, their color, and the code can be found at https://www.w3schools.com/cssref/css_colors.asp.

When you click on the main dialog's OK button, the tree description and tree diagram will reflect the new font and color information.

3.5.1 Setting custom font information for a node

Occasionally, you may want to highlight a particular node by, say, color or font type. You can set custom font information for a given node in two ways. We highly recommend using the first way.

3.5.1.1 Click on the node in the tree diagram

The first way is to (left) click on the node you wish to change the font information for. This will bring up the same font dialog as is used for setting font information discussed above. When you change the font information for a node, the tree description will also change. What shows is described in the next section 3.5.1.2.

When you click on the main text of a node with a subscript and/or a superscript and change the font information, the font information of the subscript or superscript will also change. If you click on the subscript or superscript instead and change the font information, only the subscript or the superscript portion will change. In addition, the size for the subscript or superscript will be whatever the dialog is set to. **LingTree** normally sets the size of a subscript or superscript to 70% of the default font size of the node type. Since the dialog box does not allow choices that include fractions (e.g., $12 * .70 = 8.4$), you will most likely need to edit the size value by hand as described in section 3.5.1.2

3.5.1.2 Key custom font information

The second way to create custom font information for a node is to type special commands **after** the text of the node. To indicate custom font information, key `/f` before the font information and key `/F` after it. Example (54) lists the possible commands. Notice that they all start with a vertical bar character (`|`). You can key them in any order. When you have more than one command, no spaces should appear between them.

(54)	Command	Meaning	Example
	<code> b</code>	Use bold weight.	<code> b</code>
	<code> c</code> <i>color code</i>	Use the color indicated by <i>color code</i> .	<code> c</code> blue, <code> c</code> #4d1a4d
	<code> f</code> <i>font family</i>	Use the <i>font family</i> .	<code> f</code> Arial Black
	<code> i</code>	Use italic style.	<code> i</code>
	<code> r</code>	Use regular weight and style.	<code> r</code>
	<code> s</code> <i>size</i>	Use a font size of <i>size</i> .	<code> s</code> 12.75

For color codes, see <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/paint/Color.html> and scroll down to the Field Summary section. You will find a list of color names, their codes and an example of the color. When using the numerical code, you must start it with the hash mark (#) character.

For font family names, LingTree will check to see if the font family is one that it knows about. If not, there will be an error message displayed in the bottom pane.

You can use both `|b` and `|i` together. If you use `|r` after any `|b` or `|i`, then the `|r` will overrule the others. Similarly, if you key either `|b` or `|i` after a `|r`, the `|r` will be ignored.

3.5.1.3 Removing custom font information on a node

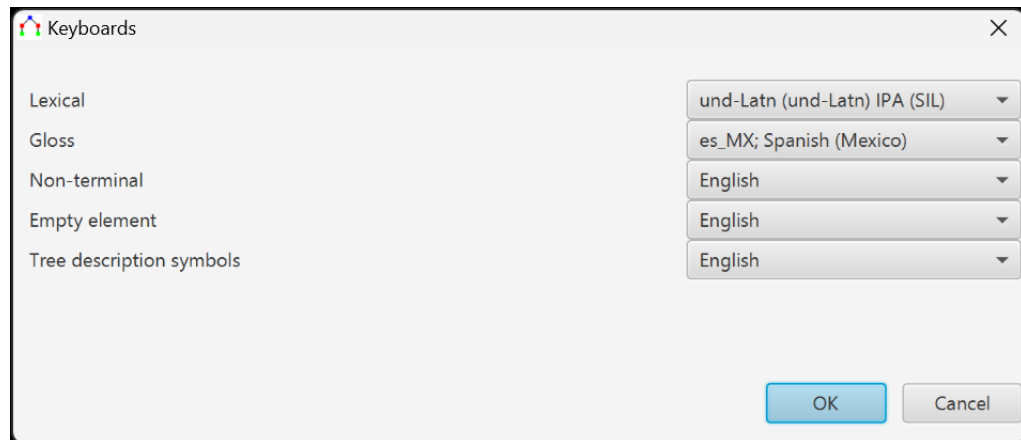
Whenever you decide to remove any custom font information on a node, you can either erase the font information in the tree description, or you can right-click on the node and choose the “Restore to default font” option.

3.6 Set keyboards

The next item under the main **Format** menu item is **Set keyboards**. Currently, this is fully available on Windows and partly available on macOS. On macOS, regular keyboards work but **Keyman** keyboards do not.⁶ When you select this menu item, it brings up a dialog that looks like what is in (55).

⁶This is a bug in current versions of Java on macOS. It is a known issue and hopefully will be fixed soon. In the meantime, you may be able to create a custom keyboard for your needs by using the **Ukelele** program (see <https://software.sil.org/ukelele/>).

(55)



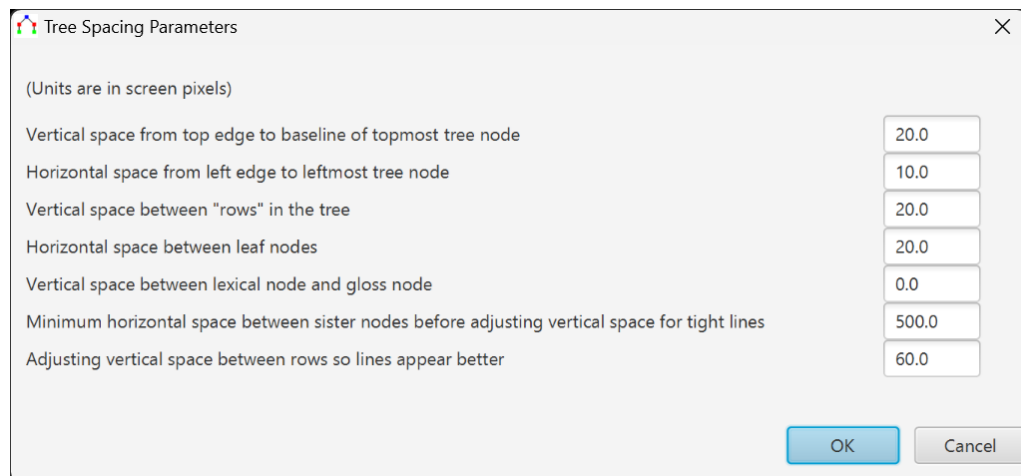
You can control which keyboard is to be used with the various kinds of items in the tree description. Click on the drop-down arrow to select one of the available keyboards on your computer.

Note that whenever you move or copy a [LingTree](#) description file from one computer to another, you will need to manually set each keyboard before they will be in effect.

3.7 Tree spacing parameters

The next item in the main **Format** menu item is **Tree spacing parameters**. When you select this menu item, it brings up a dialog that looks like what is in (56).

(56)



Each row has a description of the parameter and a box to key a numerical value. Each value is in terms of screen pixels. The items are:

1. Vertical space from the top edge of the tree diagram to the baseline of the topmost tree node
2. Horizontal space from the left edge of the tree diagram to the leftmost tree node
3. Vertical space between "rows" in the tree
4. Horizontal space between leaf nodes

5. Vertical space between lexical node and gloss node
6. Minimum horizontal space between sister nodes before adjusting vertical space for tight lines
7. Adjusting vertical space between rows so lines appear better

The last two are for cases with a very wide tree where some rows have several nodes and the lines between these nodes are not clearly visible. The idea is that when there is at least the minimum horizontal space between two nodes in the same row (item 6), then add the vertical space adjustment given by the value in item 7. By using these values, you can seek to improve how the lines appear.

You can use negative numbers, if necessary. For example, if the difference between the lexical and gloss font is such that there is too much space between the bottom of the lexical items and the top of the gloss items, you can set the “Vertical space between lexical node and gloss node” to a negative value.

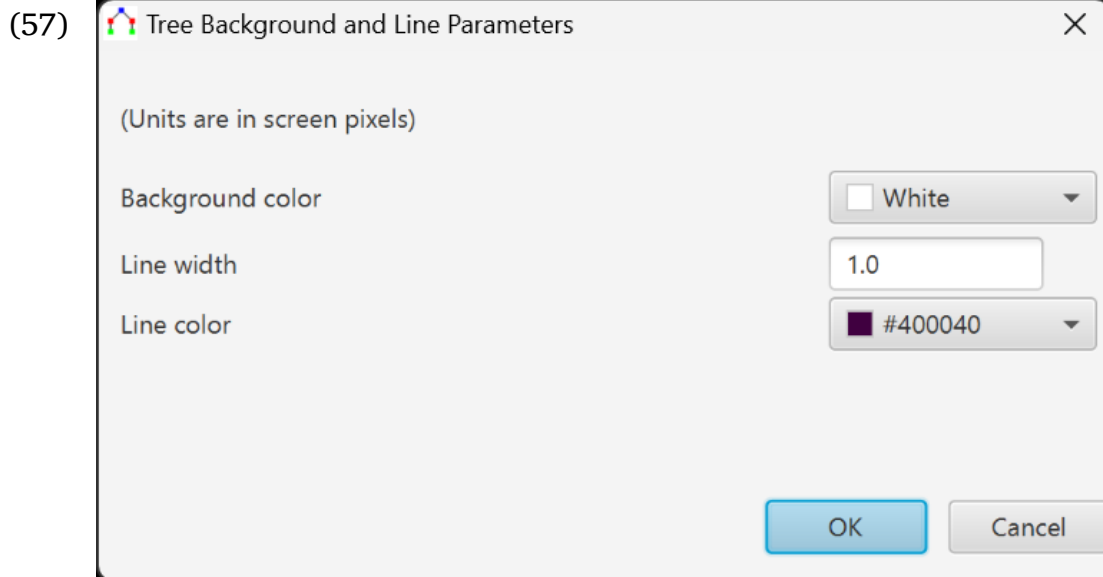
Another thing to note about the “Vertical space between lexical node and gloss node” parameter is the following:

- If the value is equal to zero, then the position of the gloss node is the same as any other node in that “row” of the tree.
- If the value is less than zero, then the position of the gloss node is the same as any other node in that “row” of the tree minus the value.
- If the value is greater than zero, then the space between the lexical and gloss nodes is that value, no matter what “row” of the tree the gloss node appears in.

Depending on a particular tree diagram, you may find it helpful to set this value to a positive one.

3.8 Background and line parameters

The third from last item in the main **Format** menu item is **Tree spacing parameters**. When you select this menu item, it brings up a dialog that looks like what is in (57).



You can set the color of the tree background and the width and color of the lines. The width is in terms of screen pixels.

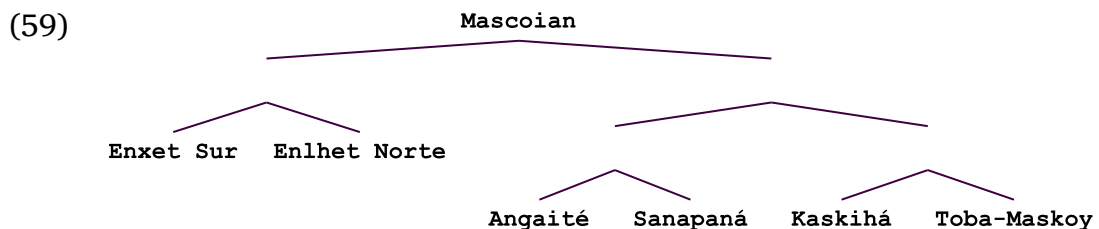
3.9 Draw vertical line with empty text

The second from last item in the main **Format** menu item is **Draw vertical line with empty text**. When this menu item has a check mark before it, any place where a text node is empty will have a vertical line connecting what comes above it with what comes below it.

For example, suppose you have a language family tree diagram like what is in (58).

(58) (Mascoian ((Enxet Sur) (Enlhet Norte)) (((Angaité)
(Sanapaná)) ((Kaskihá) (Toba-Maskoy))))

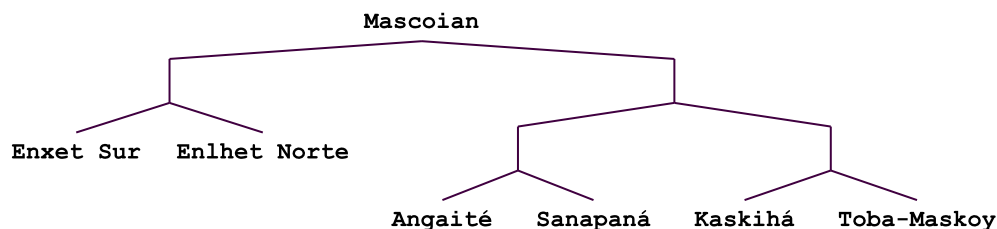
When **Draw vertical line with empty text** is not checked, this will look like what is in



This, of course, is not what is desired. One could try and add vertical line characters in the empty nodes, but that is not ideal, either.

When **Draw vertical line with empty text** is checked, however, the tree will be as in (60).

(60)



This is much better for this particular case. The default is for **Draw vertical line with empty text** to not be checked. If you are making a number of these kinds of trees, you can use what is in section 3.10 to make this menu item be checked for new tree diagrams.

3.10 Saving format information for use with new tree diagrams

The last item in the main **Format** menu item is **Save the current tree parameters to use for new tree diagrams**. When you select this menu item, the set of tree formatting values used for the current tree are remembered for any new tree diagrams in the future.

4 User convenience options

The **Settings** menu item has several options you can use to (hopefully) improve your experience with using **LingTree**. These are described below along with one useful item under the **Help** menu.

Please note that these settings are associated with the **LingTree** program on your computer. Once you set them, they will be used each time you use **LingTree** (until you change them, of course).

4.1 Font size for tree description symbols

The first item under the main **Settings** menu item is **Font size for tree description symbols**. This lets you set the font size for the symbols used in a tree description, that is (,), \L, \G, etc. The motivation for this is that some very high resolution screens may make it a bit difficult to see these or, alternatively, they may appear quite large in comparison to the other material in a tree description. So you can change their size.

4.2 Draw the tree as you type

The second item under the main **Settings** menu item is **Draw tree as you type**. When this menu item has a check mark before it, each time you change or edit something in the tree description, **LingTree** will redraw the tree diagram.

When using this mode, whenever you key an opening parenthesis, **LingTree** will automatically insert a space and a closing parenthesis. The idea is to help you keep opening and closing parentheses matched.

Sometimes the automatic drawing of the tree diagram will cause an error message to appear where the tree diagram normally shows; for example, if you have deleted a parenthesis and no longer have a matching opening or closing one, then there will be a message about a missing opening or closing parenthesis. As soon as the tree description is well-formed, the tree diagram will appear.

4.3 Show matching parenthesis with arrow keys

As mentioned in section 2.1, when you key a parenthesis, **LingTree** will temporarily select the matching parenthesis to help you see the nested structure of the description. There are times, however, when you need to see matching parentheses and it would be nice if you did not have to delete a parenthesis and immediately re-key it. For this reason, there is the third item under the main **Settings** menu item: **Show matching parenthesis with arrow keys**. When this is checked, whenever you use an arrow key to cross over a parenthesis in the description, **LingTree** will temporarily select the parenthesis that matches the one crossed over.

Besides using this menu option, you can also use the toolbar button shown in (61).

(61) **Show matching with arrows** **Do not show matching with arrows**



4.4 Delay to use when showing matching parentheses

The fourth item under the main **Settings** menu item is **Delay to use when showing matching parentheses**. The default delay is 750 milliseconds (i.e., three quarters of a second). Depending on your typing speed, you may want to change this value. Use this menu item to change the delay value to be shorter or longer. The shortest allowed value is 125 milliseconds (i.e., one eighth of a second) and the longest value allowed is 4000 milliseconds (i.e., 4 seconds).

4.5 Show full file path

The fifth item under the main **Settings** menu item is **Show full file path**. When this menu item has a check mark before it, the file name at the top of the **LingTree** window will include the full file location information. When it does not have a check mark before it, just the file name will appear at the top.

4.6 Change the interface language

You can set the user interface language by using the **Settings / Change the interface language** menu item. This brings up a dialog box showing the current interface language in a drop down chooser. Click on the chooser's drop down button to see other interface language choices. The choices given use the name of the language in the current interface language (so if the current interface language is English, then it will show "Spanish" as an option; if the current interface language is Spanish, it will show English as "inglés").

The current version has English, French and Spanish. Any corrections to the French or Spanish are welcome.

4.7 Quick Reference Guide

The main **Help** menu item has a **Quick Reference Guide** item. When you select this item, it brings up a dialog box containing a list of the special symbols you can key to get a particular result. You can position and/or resize this dialog so that you can see it while also keying a tree description. That is, unlike most dialog boxes, you do not have to close this dialog box before you can continue working. It will stay open until you close it.

5 Saving the tree

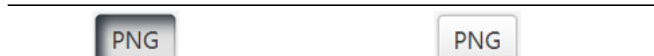
While the resulting tree looks nice within the **LingTree** program, what you usually want to do is to embed the tree diagram in some other document. **LingTree** allows you to save the tree in up to two formats:

- A Portable Network Graphics (*.png) file.
- Scalable Vector Graphics format (*.svg) file.

You select which of these formats you wish to save by clicking on the **File** menu item and then clicking on the **Save tree as .png** and/or **Save tree as .svg** items.

Besides using these menu options, you can also use the toolbar buttons shown in (62) and (63).

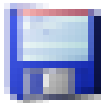
(62) **Save tree as .png Do not save tree as .png**

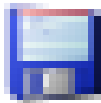


(63) **Save tree as .svg Do not save tree as .svg**



You merely check the formats you wish to use. **LingTree** will produce a file with the given extension for each checked format when you do a save operation: that



is, click on the  toolbar button, use **File / Save tree**, or type **Ctrl-S** (holding the **Ctrl** key down while pressing the **S** key). The file name (i.e., the part before the extension) is the same for all of them.

In addition, you will find a file with an extension of ".tre" which contains information for **LingTree** to draw the tree. Do not edit this file. This has all the information about the tree description that **LingTree** uses.

Note that once you set these Save Formats, they stay in effect for all ensuing trees you build.

Of these two formats, we highly recommend that you use the Scalable Vector Graphics format (*.svg) whenever possible. This is because SVG format will always look nice in a web browser or a PDF file, no matter how large the user makes the page appear.

Another reason for using the SVG format is that if you find that **LingTree** cannot do everything you need for a particular tree diagram, you can produce the SVG form and then use a tool like **InkScape** to do the rest. See <https://inkscape.org/en/> for more on **InkScape**.

LingTree also has a third option under the **File** menu item: **Save tree as a collapsible .svg file**. When this option has a check mark by it, **LingTree** will create an SVG file whose name is the same as the file name with “Collapsible” appended to it. When this file is loaded into a web browser, you can click on a non-root (and non-terminal) node in the tree and it will collapse what is under that node into an ellipse text with a triangle over it. This can be useful for someone reading a very wide tree diagram with many nodes.

If you do find yourself in a situation where you absolutely must have a pure graphics image file and the Portable Network Graphics (*.png) form is not acceptable, you can use various tools to convert your PNG file to some other graphics image format. Example (64) lists some possibilities.

(64)	Operating System	Tool	Potential formats
	Windows	Paint	.bmp, .jpg, .gif, .tif
	macOS	Preview	.jpeg, .tiff
	Linux	Image Viewer	.bmp, .jpeg, .gif, .tiff

In addition, there is the **XnConvert** program available at <https://www.xnview.com/en/xnconvert/>. It runs on all three operating systems and can convert to many different image formats.

6 Error messages

Whenever you process a tree description, if **LingTree** notices any error in the description, it will show the error message in the same place where the tree diagram usually appears. The message tries to indicate what the problem is and where it was found (in terms of line number and character position). It also shows where in the tree description the error was found. While these messages are intended to be helpful, sometimes it may be the case that what needs to be done to fix the error is not necessarily what the message implies will fix it.

For example, when the message is about a missing closing parenthesis, the message actually indicates the innermost place where the outer parentheses match. So it may not be the best place to insert a closing parenthesis; but, on the other hand, it may work just fine. You may want to use the arrow keys to find the best place to fix the problem (see section 4.3).

Another thing to note is that for longer, more complicated tree descriptions, it may take a 2-3 times longer for the tree drawing process to happen if it finds an error.

Example (65) lists the error types [LingTree](#) reports (in alphabetical order) along with a brief description of what the error type might mean.

(65) Error Type	Meaning
Bad bar code	There is something between a beginning custom font marker (/f) and its ending custom font marker (/F) but it is not recognized as one of the correct bar codes. See section 3.5.1, especially example (54).
Bad size number	There is a bar s (s) in a custom font definition, but the number after it is not recognized. See section 3.5.1.
Color definition not found	There is a bar c (c) in a custom font definition but the color definition is not recognized as a legitimate color. See section 3.5.1.
Font family not found	There is a bar f (f) in a custom font definition but the font family is not found on your computer. See section 3.5.1.
Missing abbreviation end marker	There is a beginning abbreviation marker (/a) but no abbreviation end marker (/A). See section 2.4.2.
Missing bar character	In a custom font definition, a bar character () is missing. Try inserting one. See section 3.5.1.
Missing closing parenthesis	At least one closing parenthesis is missing.
Missing color definition	In a custom font definition, there is the beginning of a color definition (i.e., c), but nothing more. Add a color definition. See section 3.5.1.
Missing content after abbreviation begin marker;	There is a beginning abbreviation marker (/a) followed immediately by an ending abbreviation marker (/A). Put something between the markers. See section 2.4.2.

Missing content after
custom font begin marker

There is a beginning custom font marker ($/\mathfrak{F}$) followed immediately by an ending custom font marker ($/\mathbf{F}$). Put something between the markers. See section 3.5.1, especially example (54).

Missing content after
subscript

There is a subscript symbol but there is not content after it. See sections 2.4.1.1 and 2.4.1.2.

Missing content after
superscript

There is a superscript symbol but there is not content after it. See sections 2.4.1.3 and 2.4.1.4.

Missing custom font end
marker

There is a beginning custom font marker ($/\mathfrak{F}$) but no custom font end marker ($/\mathbf{F}$). See section 3.5.1.

Missing font family

In a custom font definition, there is the beginning of a font family definition (i.e., $|\mathfrak{F}$), but nothing more. Add a font family name. See section 3.5.1.

Missing opening
parenthesis

An opening parenthesis needs to be inserted. This can happen when there is a node with content followed by a node type or a line type. See sections 2.2 and 2.3.

Missing size number

In a custom font definition, there is the beginning of a size definition (i.e., $|\mathfrak{s}$), but nothing more. Add a number. See section 3.5.1.

Syntax error in description

Perhaps obviously, this indicates some kind of error, but [LingTree](#) failed to identify it. Look at where it was found and see if you can guess what might be wrong.

There is content after a
completed tree

A “completed tree” means that there is a matching closing parenthesis for the first opening parenthesis. This error indicates that some content or nodes occur after the closing parenthesis which matches the first opening parenthesis. It may be that you need another opening parenthesis before the first opening parenthesis or maybe some content was keyed in the wrong place at the end.

Too many closing parentheses	A closing parenthesis was found that does not have a matching opening parenthesis.
Too many line types	Only one line type is allowed in a node. See section 2.3.
Too many node types	Only one node type is allowed in a node. See section 2.2.

7 Known problems

The following items are known to be less than desirable with this version of LingTree:

- When you have a [LingTree](#) description file created by the older version of [LingTree](#) (version 0.7.5 or earlier), this newer version of [LingTree](#) will open and attempt to convert the file to the new format. Sometimes, the conversion process works less than ideally.⁷ For example, the line thickness may be quite large. In such cases, use the items in the **Format** menu to fix the problem (see section 3, especially 3.7 and 3.8).
- When showing matching parentheses while using arrow keys (see section 4.3), if you type several arrow keys quickly, the cursor caret may not be where you expect it. Either
 - turn off the show matching parentheses with the arrow keys option;
 - set the delay to be shorter (see section 4.4); or
 - wait for the matching parenthesis to show before pressing the next arrow key.
- The default algorithm for drawing trees assumes that the text in higher nodes in a tree are shorter than the text in lower nodes. If your tree needs wider higher nodes, try using the column-oriented algorithm described in section 3.3.
- On macOS: if you double click a tree description file in [Finder](#), then [LingTree](#) will open the last opened file, not the one you clicked on. This is due to a communication problem between the macOS operating system and the Java language [LingTree](#) is written in.
- On macOS, regular keyboards work but [Keyman](#) keyboards do not.
- When you move a [LingTree](#) file from one computer to another, you may well need to reset the keyboards manually. This is because the keyboard settings may be different on the two machines.
- Increasing the size of a (part of a) gloss node by using the custom font information on a gloss node (see section 3.5.1) may result in the gloss text material overlapping the text material of the lexical node above it. We recommend not trying to change the font size. Rather, use color or bold or italic to get the distinction you desire.

⁷The main issue here is that the size units in the older version were in .01 mm while the new version uses pixels. With some high resolution screens (at least on Windows operating systems), the conversion process is not always correct.

8 Support

If you have any questions with [LingTree](#) or find bugs in it, please send an email to lingtree_support@sil.org or go to the [LingTree](#) web site at <https://software.sil.org/lingtree/>.

Index

In this index, when more than one page number is given, the one in bold indicates the main place to look.

- Abbreviations 25
- Center nodes over daughters 32
- Color 5, **33**
 - Background 38
 - Lines 38
 - Text in nodes 36
- Column-oriented algorithm 31, 46
- Draw tree as you type 40
- Draw vertical line with empty text 39
- Empty element node 11
- Empty gap in tree 39
- Error messages 43
- File name 2, **41**
- Flat tree 28
- Font 5, **33**
 - Abbreviations 25, **34**
 - Custom font for a node 35
 - Empty element node 34
 - Gloss node 33
 - Lexical node 33
 - Non-terminal node 33
 - Tree symbols 40
- Formatting a tree
 - Flat tree, *see* Flat tree.
- Gloss node 8
 - Increasing size of a gloss node 46
- Graphic formats 43
- [Inkscape](#) 43
- Keyboards 36, 46
- Language of UI 41
- Lexical node 6
- Lines 12
 - Omit a line 14
 - Overlapping or tight lines 38
 - Regular 12
 - Triangle, *see* Triangle line type.
 - Width 38
- macOS 36, 46
- New tree description 40
- Non-terminal node 6
- Parentheses
 - Insert closing parenthesis automatically 4, **40**
 - Insert parenthesis in node content 25
 - Show matching parenthesis 4, **41**, 46
 - Delay 4, **41**
- Phrase, *see* Triangle line type.
- PNG format 42
- pro (empty pronoun) 11
- Process a tree description 3
 - Draw as you type, *see* Draw tree as you type.
- Quick reference guide 42
- Right-to-left 30
- Save 42
- Spacing
 - Between leaf nodes 37
 - Between left edge of diagram and left-most node 37
 - Between lexical item and gloss 38
 - Between top of diagram and the top-most node 37
 - Between “rows” in the tree 37
- Subscript 17, 35
 - Italic 19
 - With superscript 23
- Superscript 17, **20**, 35
 - Italic 22
 - With subscript 23
- SVG format 42
 - Collapsible 43
- Trace 11
- Tree description 3, 4
 - Custom font information 35
 - Node types 5
 - Empty element, *see* Empty element node.
 - Gloss, *see* Gloss node.

- Lexical, *see* Lexical node.
- Non-terminal, *see* Non-terminal node.
- Process a description, *see* Process a tree description.
- Spaces 4
- Tabs 4
- Tree spacing parameters 37. *See also* Spacing.
- Triangle line type 12
- Windows operating system 36
- [XnConvert](#) 43