LingTree User Documentation

H. Andrew Black
SIL International

Contents

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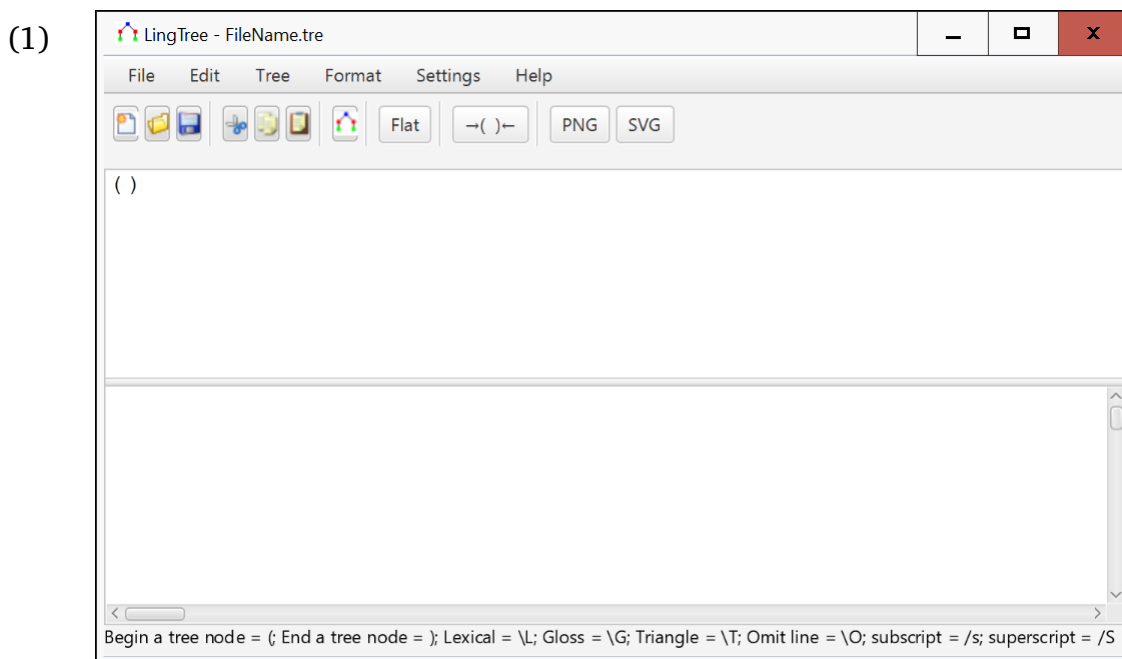
The file name where the tree information is stored is shown in the upper left part of the window. You key the description in the top pane, process the description, and then the tree appears in the bottom pane. See section 2 for information on keying a tree description and processing a tree.

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.[1] (See section 4.2.)

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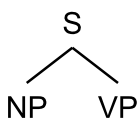
2.1 Basic tree description

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.

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)

```
          S
         / \
       NP    VP
```
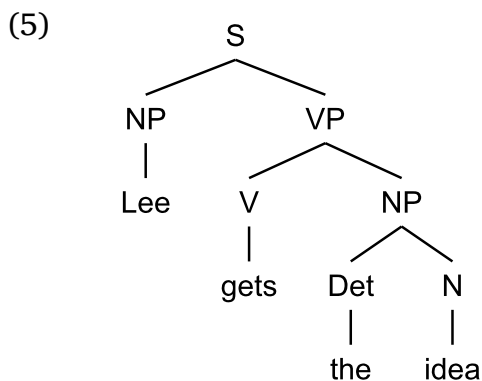
And if you key what is in (4)

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

it will produce what is in (5)

---

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

(5)

S
NP   VP
Lee   V   NP
gets Det   N
the   idea

You may set font and color information for the nodes. See section 3.3. 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 3.3.

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 3.3). Then keying what is in (6)

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

will produce what is in (7).

(7)

```
           S
         /   \
       NP     VP
       |      |
      John    V
              |
            sleeps
```
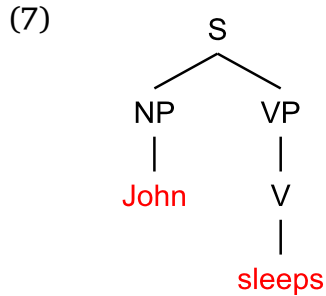
If you do not key the **\L**, then it will look like what is in (8).

(8)

```
           S
         /   \
       NP     VP
       |      |
      John    V
              |
            sleeps
```

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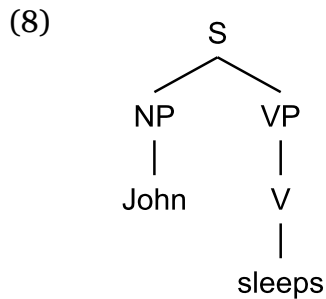
2.2.3 Indicating a gloss node

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 one is going to mark glosses, one 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 3.3). 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)

```
           S
         /   \
       NP     VP
       |      |
      Juan    V
      John    |
            duerme
            sleeps
```
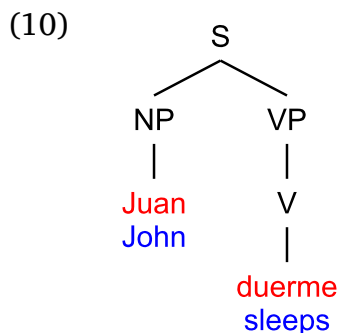
Notice that there are no lines between the lexical nodes and their respective glosses and that the glosses are relatively close to the lexical node.

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

(11)

```
               S
            /     \
         NP        VP
          |         |
        Juan        V
          |         |
        John      duerme
                    |
                  sleeps
```
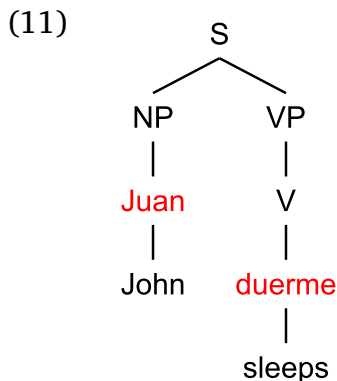
Notice that there is a line between the lexical node and their daughter node. The blue color also does not show for the daughter nodes.

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 one has 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 3.3). Then keying what is in (12)
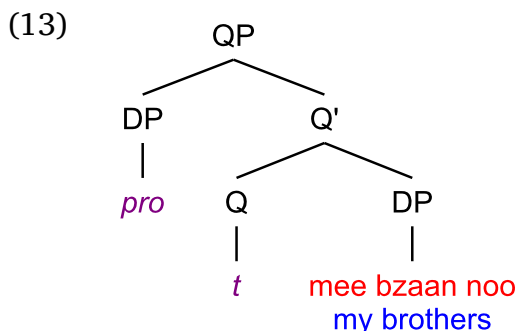
```
(12)  (QP (DP (\E pro)) (Q' (Q (\E t)) (DP (\L mee bzaan noo
      (\G my brothers)))))
```

produces what is in (13).

(13)

```
               QP
            /      \
         DP          Q'
          |        /    \
        pro      Q        DP
                  |         |
                  t    mee bzaan noo
                         my brothers
```

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

(14)

```
              QP
           ╱      ╲
        DP          Q'
        |        ╱      ╲
       pro      Q          DP
                |          |
                t      mee bzaan noo
                        my brothers
```
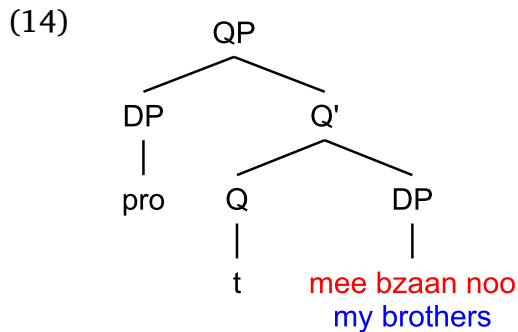
Notice that the empty pronoun and trace nodes are treated as non-terminal nodes.

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)

```
              NP
            ╱────╲
           ╱      ╲
      all the King's men
```
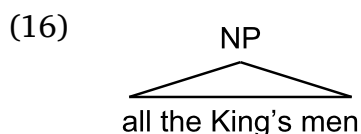
If you do not key the **\T**, then it will look like what is in (17).

(17)

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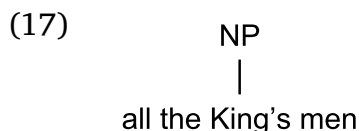
You may key both a triangle and either a lexical item or an empty element for a given node. The order in which you key the **\T** and either the **\L** or the **\E** is not crucial. Either order will work. If you key what is in (18),

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

it will look like what is in (19).

(19)

NP

all the King's men

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

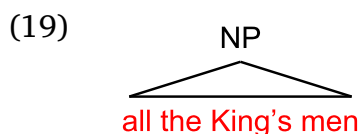To not have a line show above a node, key **\o** before it (i.e., right after the opening parenthesis). While it is not required to include a space between the **\o** and whatever follows it, it can make the description easier to read. For example, keying what is in (21)[2]

(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).

(22)   σ   σ

         O   N   O   N

              R   R   C

         t   e   p   i   k

---

[2]In this particular tree diagram, "O" is an abbreviation for "Onset."

If you do not key the **\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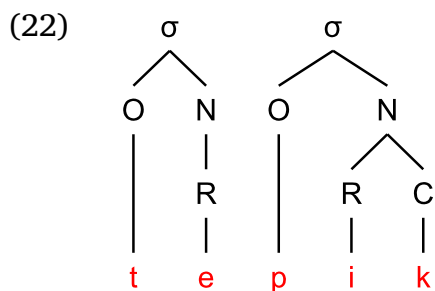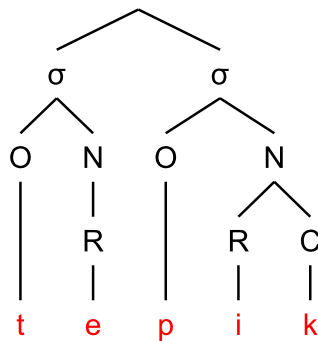** or the **\E** is not crucial. Either order will work. You may find that it is easier to read and think about, though, if you key the **\O** first since it has to do with what occurs over the node, while the **\L** or **\E** has to do with what is in the node.

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 and for keying parentheses that will occur as content.

2.4.1 Indicating subscripts or superscripts

One 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). One does 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)

```
                    S
           ┌────────┴────────┐
          NP₁               VP
           │         ┌───────┴───────┐
           N         V              NP₂
           │         │               │
          dogs     chase             N
                                     │
                                    cats
```
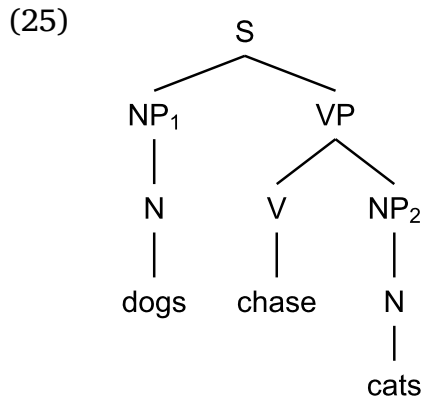
Note that the subscript will use the same font family and color that the node it is in uses. So if the node is a non-terminal, then it uses the non-terminal font family and color. Similarly for lexical, gloss, and empty node types. The subscript text also is always in regular style (i.e., neither bold 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)

```
                    S
           ┌────────┴────────┐
          NP1               VP
           │         ┌───────┴───────┐
           N         V              NP2
           │         │               │
          dogs     chase             N
                                     │
                                    cats
```

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)

```
                    S
              ╱           ╲
          NP₁              VP
           │            ╱      ╲
           N           V        NP₂
           │           │         │
         dogs        chase       N
                                 │
                                cats
```
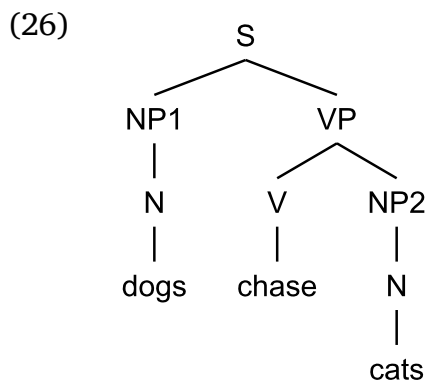
2.4.1.3 Superscript

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). One does 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)

```
          IP
        ╱    ╲
      DP      I¹
```

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.

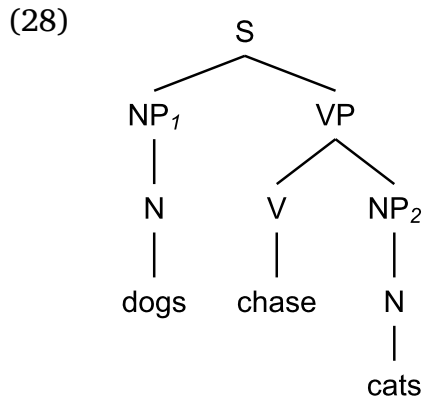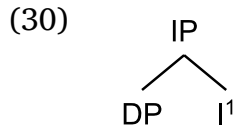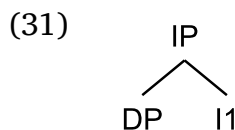If you do not key the **/S**, then it will look like what is in (31).

(31)

```
          IP
        ╱    ╲
      DP      I1
```

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 **/ˆ**, instead. The only difference between **/S** and **/ˆ** 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)

```
        IP
       /\
      /  \
    DP    I¹
```

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)

```
      NP
      |
      N'
      |
      N⁰₍₊wh₎
```

2.4.2 Handling parentheses

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 (36)

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

it will come out as in (37).

(37)

```
         NP
         |
    Paul (the bear)
```

If you do not quote the parentheses with a backslash, then it will look like what is in (38) because the parentheses are interpreted as a new tree subnode.

(38)

```
        NP
        |
       Paul
        |
      the bear
```

3 Formating a tree

You can adjust the positioning of the tree and its nodes as well as choose the fonts, sizes, and colors for the non-terminal, lexical, gloss and empty nodes. 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 (39) indicates what this button looks like in its two states.

(39) **Show as flat   Show regularly**
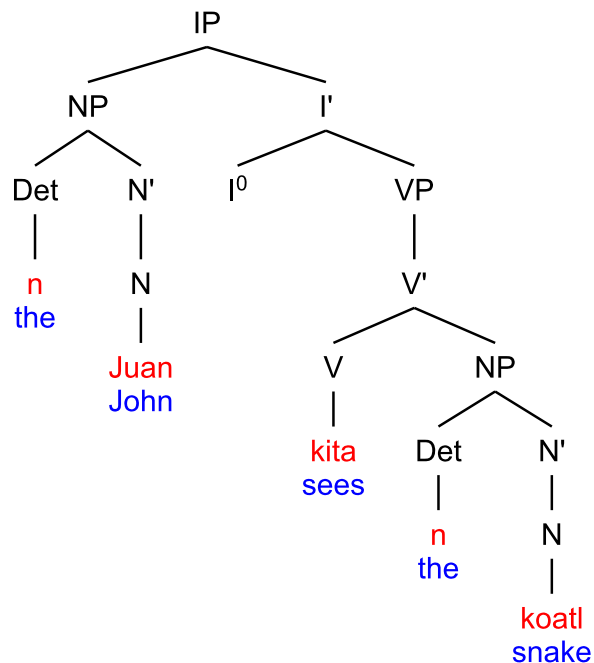
        Flat   Flat

For example, without using the flat mode, the description in (40)

```
(40) (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 (41) (assuming that lexical nodes are set to be in red and gloss nodes are in blue).

(41)

When using flat mode, however, it will look like the tree in (42).

(42)
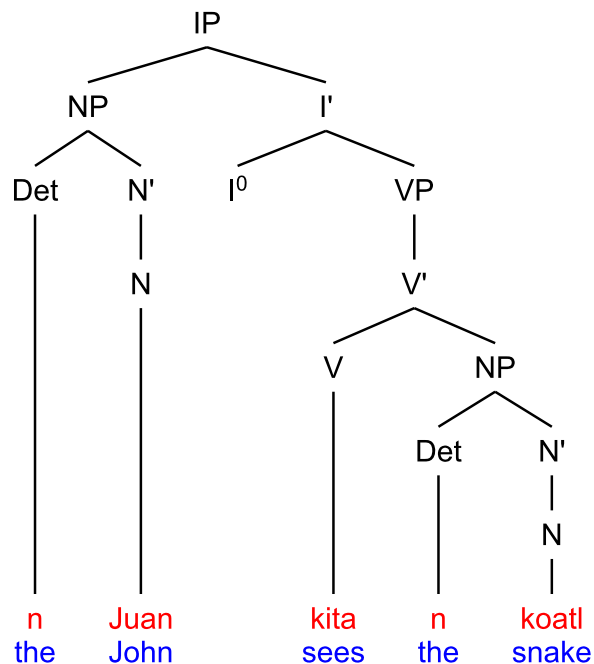
Note that if you use the "Flat" option while using triangles (see section 2.3.2), you will need to also use either the `\L` or the `\G` at the same time as using the `\T`. Otherwise, the triangle node may not be at the same level as other "flattened" nodes.

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 (44) below, then the description given in (43) looks like example (44) when the Draw tree right-to-left is checked.

(43) (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 אֵת (\G obj))) (N (\L הַשָּׁמַיִם (\G
      the heavens)))) (NP (DObj (\L וְאֵת (\G and.obj)))
      (N (\L הָאָרֶץ (\G the earth)))))))

(44)

S
├── VP
│   ├── NP
│   │   ├── NP
│   │   │   ├── N — הָאָרֶץ (the earth)
│   │   │   └── DObj — וְאֵת (and.obj)
│   │   └── NP
│   │   ├── N — הַשָּׁמַיִם (the heavens)
│   │   └── DObj — אֵת (obj)
│   ├── NP
│   │   └── N — אֱלֹהִים (God)
│   └── V — בָּרָא (created)
└── PP — בְּרֵאשִׁית (in the beginning)
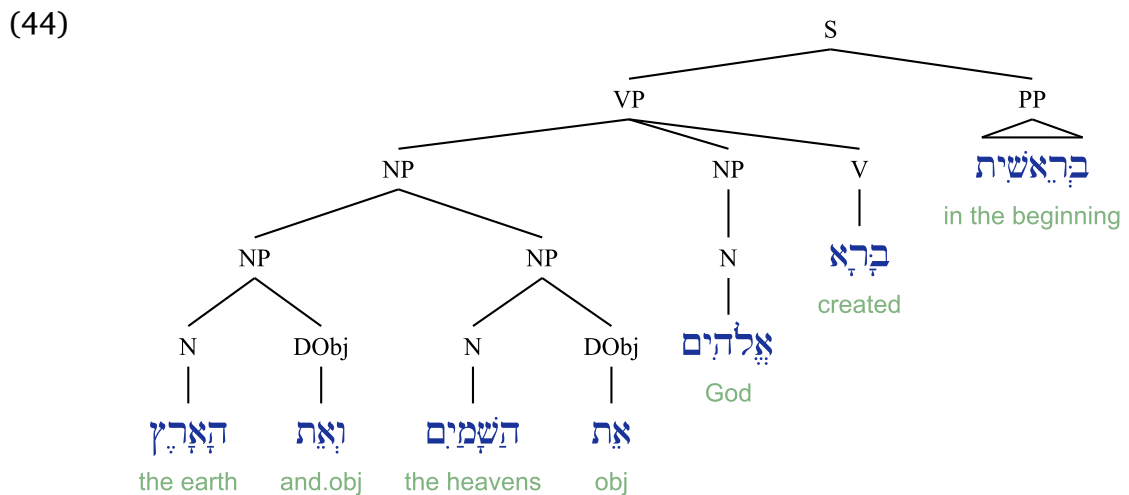
The output looks like example (45) when the Draw tree right-to-left is left unchecked.

(45)

S
├── PP — בְּרֵאשִׁית (in the beginning)
└── VP
    ├── V — בָּרָא (created)
    ├── NP
    │   └── N — אֱלֹהִי (God)
    └── NP
        ├── NP
        │   ├── DObj — אֵת (obj)
        │   └── N — הַשָּׁמַיִם (the heavens)
        └── NP
            ├── DObj — וְאֵת (and.obj)
            └── N — הָאָרֶץ (the earth)
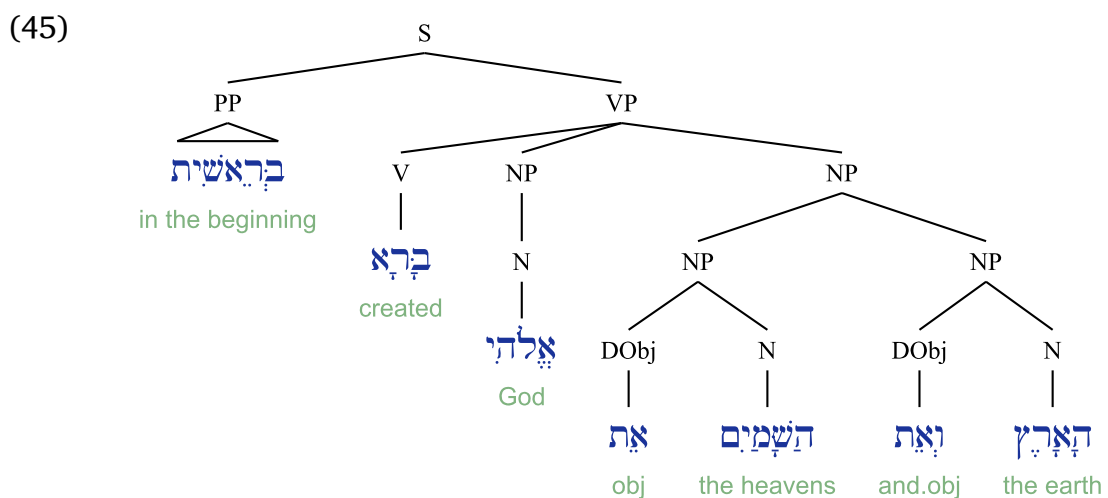
3.3 Setting the font and color information

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

1. Non-terminal (2.2.1)
2. Lexical (2.2.2)
3. Gloss (2.2.3)
4. 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 (46).

(46)
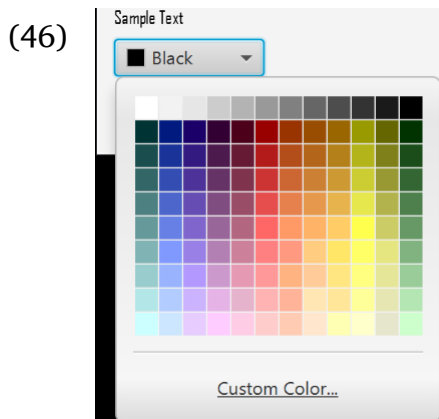
You can click on the color you want or you can click on the "Custom Color" link. Doing so gives you what is in (47).

(47)
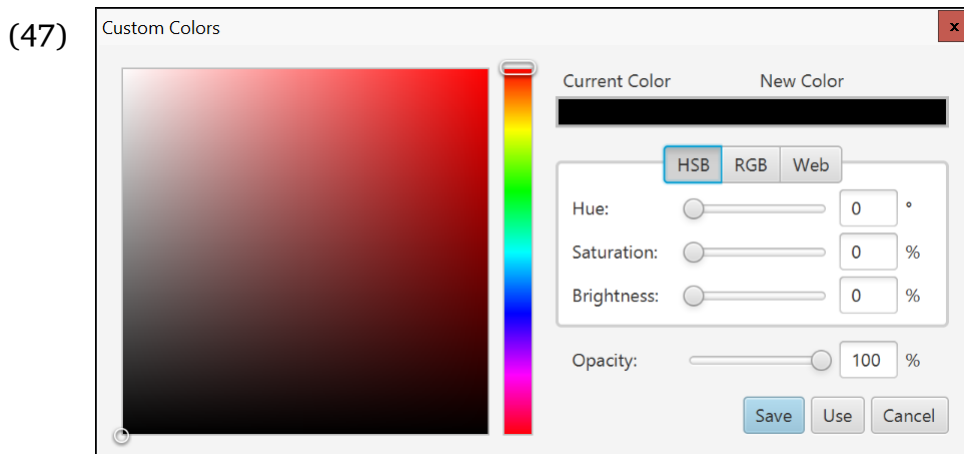
If you click on the Web tab, you can key in a code for the color. A list of color names, their color, and the code can be found at https://www.w3schools.com/cssref/css_colors.asp.

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

3.4 Tree spacing 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 (48).

(48)
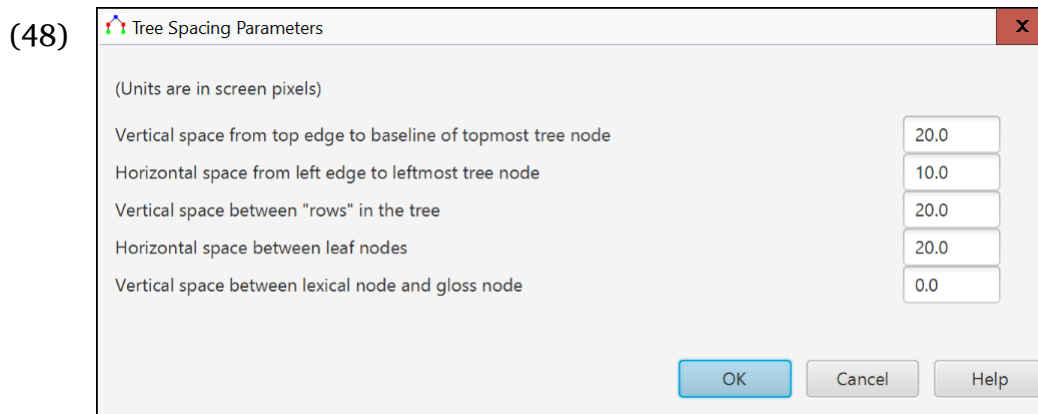
Each row has a description of the parameter and a box to key a numerical value. You can use negative numbers, if necessary. For example, if the difference between the lexical and gloss font is such that there is too much space between the bottom of the lexical items and the top of the gloss items, you can set the "Vertical space between lexical node and gloss node" to a negative value.

3.5 Background and line parameters

The second 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 (49).

(49)
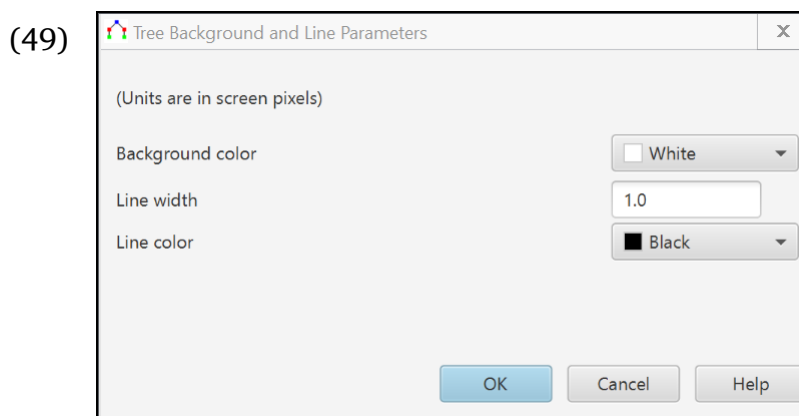
You can set the color of the tree background and the width and color of the lines.

3.6 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 (50).

(50)

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 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 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 a rough, most likely often inaccurate version of Spanish. Any corrections to the Spanish are welcome.

4.6 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 one usually wants 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 (51) and (52).

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

| PNG | PNG |

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

| SVG | 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: 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.

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 (53) lists some possibilities.

(53)

Operating System	Tool	Potential formats
Windows	Paint	.bmp, .jpg, .gif, .tif
Mac OS X	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 (54) lists the error types LingTree reports along with a brief description of what the error type might mean.

(54)

Error Type	Meaning
Missing closing parenthesis	At least one closing parenthesis is missing.
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 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.
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.[3] 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 3.4 and 3.5).
- 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.

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/.

---

[3]The 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.