

# Bdotty User's Guide

BiG team

Last Update: Jun. 14 2012

## 1 Overview

This document describes how to use *bdotty*, our small extension of *dotty* graph editing system. It consists of lefty scripting language source files and one main shell script. It is mainly intended to interactively edit graphs that is fed to and produced by our bidirectional graph transformation system. Since system encodes input and output markers into node labels using specific syntactic convention, *bdotty* supports editing those markers respecting that syntax which is hidden to users.

## 2 Installation

**Requirements** You should have DOT system<sup>1</sup>[2] installed. It is available from <http://hoagland.org/Dot.html>.

### 2.1 Files

*bdotty* consists of the following lefty[3] scripts and a shell script:

**big.lefty** Main program.

**kmp.lefty** Implementation of Knuth-Morris-Pratt string matching algorithm.

**parsenode.lefty** Parsers for node labels.

**string.lefty** String manipulation functions.

**bdotty** Main shell script.

**Installation** Copy these files to either the lefty directory under Graphviz<sup>2</sup> package such as “graphviz-version/share/graphviz/lefty”, or a separate directory. It may be convenient to invoke *bdotty* instead of *dotty*, since *bdotty* inherits all the functionalities of *dotty*. You can do this by adding symbolic link

```
% ln -s path_to_bdotty dotty
```

in any directory that is included in your PATH environment variable.

## 3 Usage

Since *bdotty* is an extension of *dotty*, basic usage is identical to that of *dotty* [1]. If you load one of the files the bidirectional transformation module has produced, nodes of the graphs are rendered either in an *elliptic* shape (Figure 1) or *record* shape (Figure 2).

Extensions we made are:

---

<sup>1</sup>graph layout products developed by AT&T

<sup>2</sup><http://www.graphviz.org/>

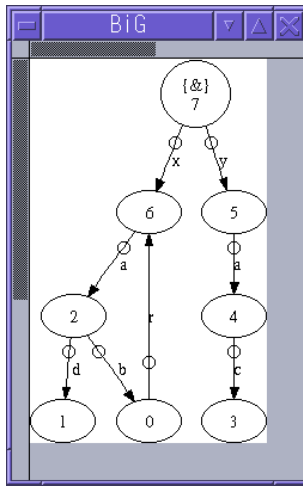


Figure 1: Graph rendered in elliptic mode

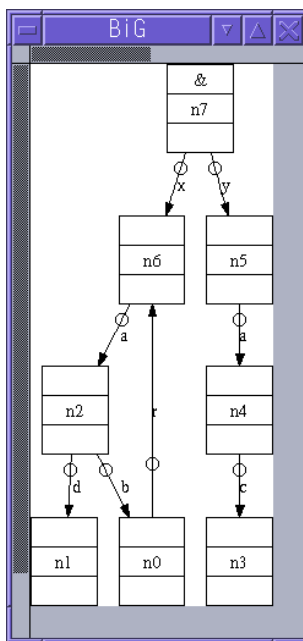


Figure 2: Graph rendered in record mode

### Menu for editing edge labels

Although original *dotty* can edit any attribute through “set attr” menu, user must always retype values that is already stored, in addition to attribute names. “set edge label” (Figure 3) among the menus that appears when the circular handle of an edge is right-clicked<sup>3</sup> enables easier editing of the label of an edge. If the label is already set, it appears in the text box (Figure 4). Syntactic convention on labels are the same as that on labels written in UnCAL source. For example, double-quoted strings are recognized as string, while unquoted strings are treated as labels. Unquoted strings `true` and `false` are treated as boolean values.

### Menu for editing markers

Context menu that appears by right-clicking nodes is extended with “set imarks”, “set omarks” for editing input markers and output markers, respectively. User should specify markers separated by commas (“,”). In an elliptic mode, input and output markers are rendered bracketed using curly brace “{}”, above and below the node ID, respectively. Figure 5 shows a root node that is also marked with two output markers  $&x$  and  $&y$ . Uncustomized *dotty* can still edit markers by manually respecting marker encoding syntax

<sup>3</sup>Control+click for Macintosh. Be sure to enable “three button emulation” in your X11 system such as XQuartz.

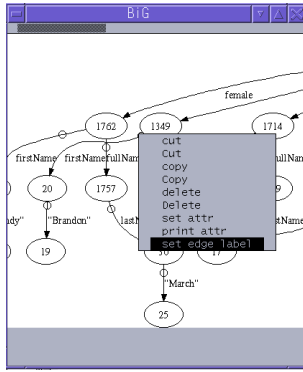


Figure 3: Context menu for edge editing with “set edge label”



Figure 4: Dialog to set edge label

via “set attr” menu. For example, when you are in the elliptic mode, you can place I/O markers shown in Figure 5 by setting `label={&}\n\n{n}{&x,&y}`. When you are in the record mode, same thing can be achieved by setting `label={&|\N|&x,&y}`. This slightly simpler syntax is the reason why record node has been used in spite of redundancy (most of the nodes have no markers at all).

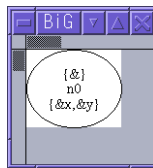


Figure 5: A node with input and output markers (elliptic mode)

## Node insertion

Newly inserted nodes also respects the current mode (elliptic or record). If you are in record mode (i.e., global node shape attribute is set to `record`), node is inserted with three fields, with node ID in the middle. Input and output markers associated with a record node are placed in top and bottom field, respectively, without curly braces.

## Keyboard shortcut

Typing ‘q’ causes *bdotty* to quit (without saving). Together with ‘s’ (save graph), it facilitates quick bidirectional editing.

## 4 Limitations

*bdotty*, as well as *dotty*, can not edit or specify ID of existing nor newly inserted nodes. To avoid node ID conflict with other graphs, this function is essential.

*bdotty* do not validate I/O markers. For example, duplicates are not eliminated.

## References

- [1] S.C. North E. Koutsofios. Editing graphs with *dotty*. Technical report, AT&T, Murray Hill, NJ, July 1994. <http://www.graphviz.org/pdf/dottyguide.pdf>.

- [2] Emden Gansner, Eleftherios Koutsofios, and Stephen North. Drawing graphs with dot. Technical report, 2006. <http://www.graphviz.org/Documentation/dotguide.pdf>.
- [3] E. Koutsofios. *Editing Pictures with lefty*. Murray Hill, NJ, July 1994. <http://www.graphviz.org/pdf/leftyguide.pdf>.