Grammar Editor and Path Parser

August 16, 2007

This document describes the Java application "Grammar Editor and Path Parser." The program is a simple way to describe and generate artificial grammars based on nodes, their relations with other nodes, and labels describing their relationship.

Requirements

This program was developed and tested on a PowerMac machine running Mac OS X 10.4, Tiger. It was also tested on a Pentium 4 machine running Windows XP. Furthermore, thisprogram should run on any machine that has the Java JRE 1.5 or grater installed.

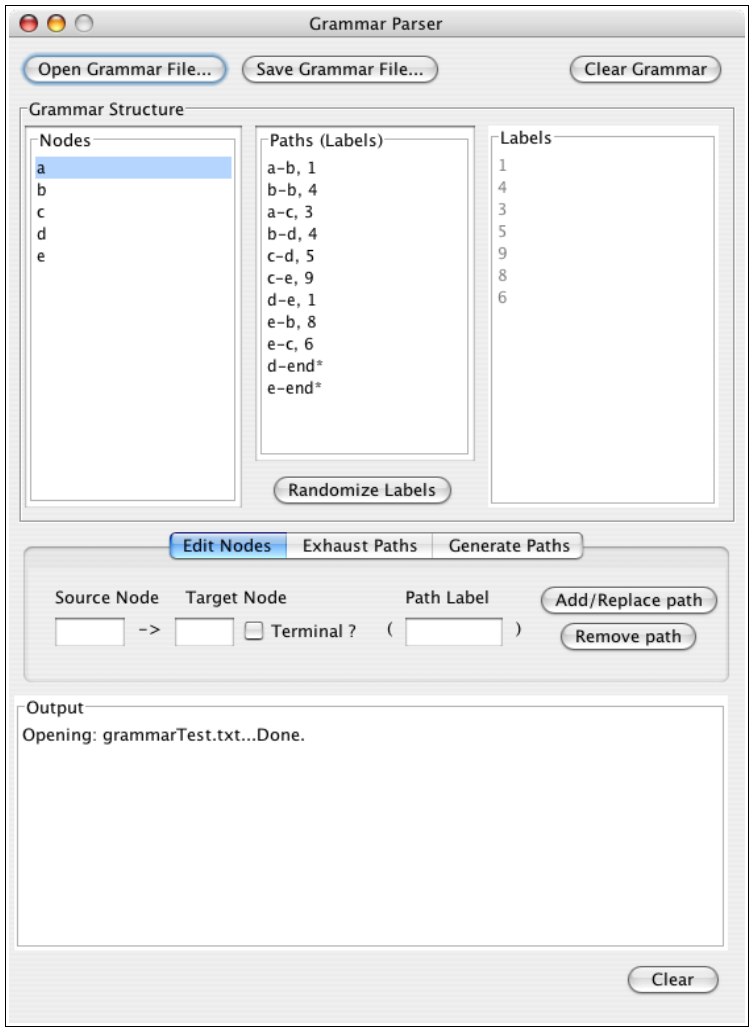

Figure 1: Screen capture of the application

Main Windows

Figure 1 shows the main window. The three columns on the "Grammar Structure" panel keep track of the grammar's components and structure. The column labeled "Grammar Nodes"shows all the nodes present in the grammar **at any time**. The "Paths" column shows the relationships between the nodes, as well as the label of that relationship. The "Labels" column shows all the labels describing the paths.

Modifying the Grammar Structure

The middle tabbed panel contains three independent tabs. The "Edit Nodes" tab is an interface for entering new paths and grammar nodes, as well as removing existing paths/nodes. To add a new path to a grammar, type in the start node in the "Source Node" input field. Then enter the target node in the "Target Node" field and finally enter the label for the path in the "Path Label" field. If you want to specify the target node as a terminal node, select the "Terminal Node" check box. If the target node is a terminal node, you don't need to specify a label, otherwise, all three fields must be filled for the path to be valid. Once you click the "Add/Replace Path" button, the program will do the following depending on the different cases:

1. If either the start and target nodes have not been created yet, they will be added to the Grammar Nodes automatically. **Note:** Node names are case-sensitive.
2. If the path you give already exists but the label is different, the label of the old path will be updated with the new one. If the old label is no longer used in any path, it will be automatically removed from the list of possible labels.

Note also that you can easily edit existing path labels or remove paths by selecting them in the "Paths" column. Once you select a path, the input boxes in the "Edit Nodes" tab will automatically be filled with the correct information. Then you may update the label of that path or click the "Remove Path" button to remove that particular path.

Generating Paths

There are two options to generate paths depending on the type of grammar you are dealing with. The first method is to exhaust all the paths up to a given size in a grammar containing terminals. The second method is to generate paths based on transition probabilities between nodes. You specify either method by selecting the respective tab in the middle tabbed panel, "Exhaust Paths" or "Generate Paths." For both of these, you may save the results in two ways. You may click the button "Go" to show the results in the text area labeled "Output" at the bottom of the window. You may also click the button "Save Paths" to save the results in a text file.

To exhaust all paths in an artificial grammar **with terminals**, first highlight the root node in the "Nodes" column. Then, type in the wanted maximum size in the "Path Size" field.

To generate paths based on a probabilistic grammar, you must represent the grammar using the node names as the different states and the labels as the transition probabilities between nodes. Note that all labels must be real numbers less than or equal to 1. To generate this kind of paths, specify the path size wanted and the number of paths you need. Next you have the option of either specifying a starting node, or allowing the program to choose it randomly at each trial. Check the "Use random root nodes" check box to activate the latter option.

Randomizing Labels

One experimental way to modify a grammar with terminals is to randomize the labels of all the paths. The "Grammar Structure" section contains a button labeled "Randomize Labels," located below the "Paths" column. This function will randomize the existent labels of all paths. Note that some of the labels may repeat and some may not show up at all; however, none are not deleted from the "Labels" column until manual changes are done to the grammar.

Grammar Files

A user can open or save a text file containing a grammar structure. One may first create the grammar within this program and then save it to assure the formatting is correct. Click the "Save Grammar File..." button on the top of the window to save the grammar structure in a text file.

A user can also create the grammar structure in an external text editor. Figure 2 and figure 3 show a graph of an artificial grammar and the equivalent representation in a grammar file, respectively.

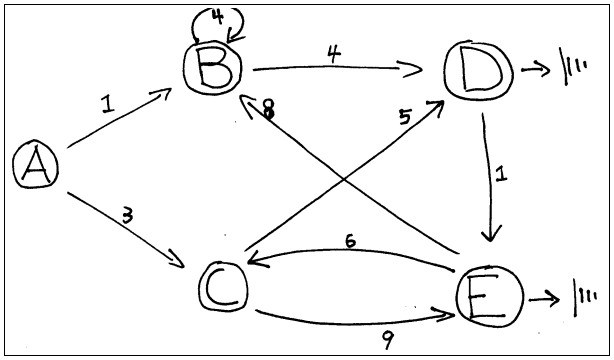

Figure 2: A graph of an artificial grammar

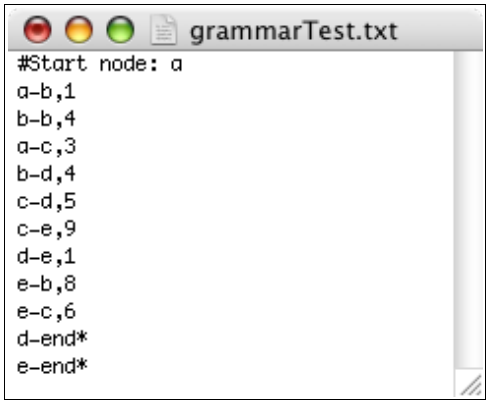

Figure 3: A grammar file describing the grammar in figure 2.

As you can see, each line should contain a node-to-node path along with its label separated by a comma. Again, remember that **all nodes are case-sensitive**. Also, note that if you want to specify a terminal node, you must put an asterisk (*) on either side of that node. In Figure 3, for example, the node "end" is a terminal node because it has an asterisk next to it.

A grammar structure may also indicate the start node of a grammar at the beginning of the text file by writing "#Start node: " and then the start node. In Figure 3, the node "a" will be the start node.

To open a grammar structure file, click the "Open a Grammar File..." button on the top of the window. This will read in all the path definitions and automatically keep track of all the nodes and labels. If a start node is specified, that node will be automatically highlighted in the "Nodes" column.

Output

All messages and notifications will be shown in the "Output" text area at the bottom of the window.