



Visualizing the Davis-Putnam Procedure

An Introduction to the DPvis Tool (v0.9)

Edda-Maria Dieringer and **Carsten Sinz**

Symbolic Computation Group
WSI for Computer Science
University of Tübingen
Sand 13, 72076 Tübingen
Germany

{dieringe, sinz}@informatik.uni-tuebingen.de

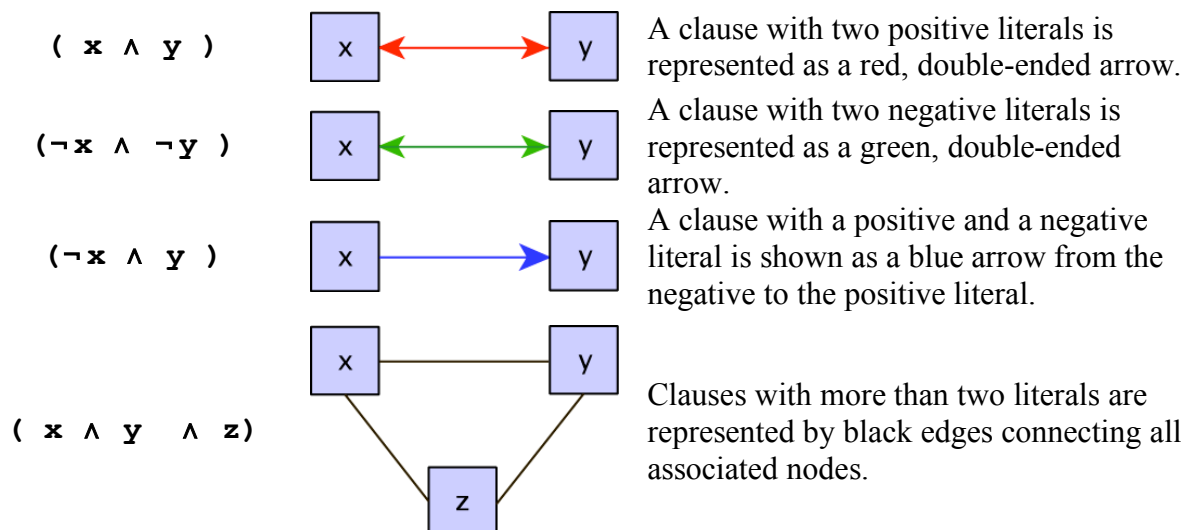
1 Basics	2
1.1 Interaction Graph	2
1.2 Decision Tree	3
2 User Interface	3
2.1 Menu bar	4
2.2 Toolbar	5
2.3 Interaction graph	6
2.4 Decision Tree	6
2.5 Input field and Statistics	6
Literature	8

1 Basics

DPvis is a tool to visualize SAT instances and runs of the DPLL (Davis-Putnam-Logemann-Loveland) procedure. We assume that the reader is familiar with basic terms of SAT-solving. If you are not, we suggest reading some papers that contain an introductory section like [Silva, Sakallah]. In the following we will introduce two additional important terms that we will use in our visualization: *interaction graphs* and *decision trees*.

1.1 Interaction Graph

A definition for interaction graphs can be found in [Dechter, Rish]: An interaction graph “is an undirected graph (...) that contains one node for each propositional variable and an arc connecting any two nodes whose associated variables appear in the same clause”. This definition has been extended by [Sinz] to include a special highlighting for clauses with only two literals:

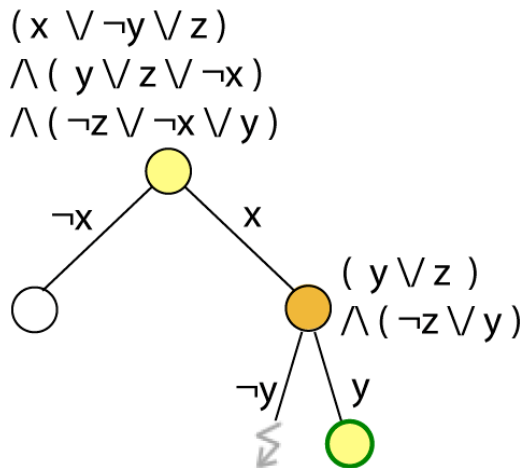


The special highlighting of clauses with only two literals is justified by the fact that the 2-SAT problem is solvable in polynomial time and by the experiences that have been made with the (2+p)-SAT model (see [Monasson *et al*]). The fraction of clauses of length two may therefore have a direct impact on the “hardness” of the problem.

A new approach in the research on the nature of SAT problems suggests visualizing the problems as interaction graphs to study the (hidden) structure of SAT instances empirically (see [Sinz]). Interaction graphs have widely proven to be a useful concept to visualize SAT instances not only in the attempt to reveal their hidden structure. Although the representation as interaction graph is not lossless (you cannot reconstruct the SAT problem from the interaction graph), the visualization is a good instrument to get an idea about the size and the structure of a particular SAT problem.

1.2 Decision Tree

The decision tree depicts the traversal of the DPLL procedure through the search space. The root of the decision tree represents the original problem. From the root the algorithm branches on a particular decision variable chosen by a decision heuristic. By convention DPvis shows an assignment with the boolean value `false` as a branch to the left child and the opposite case as a branch to the right child. In any case the child node represents a simplified problem. The algorithm descends in the decision tree as long as it finds a solution or a conflict. A leaf in the decision tree represents both events.

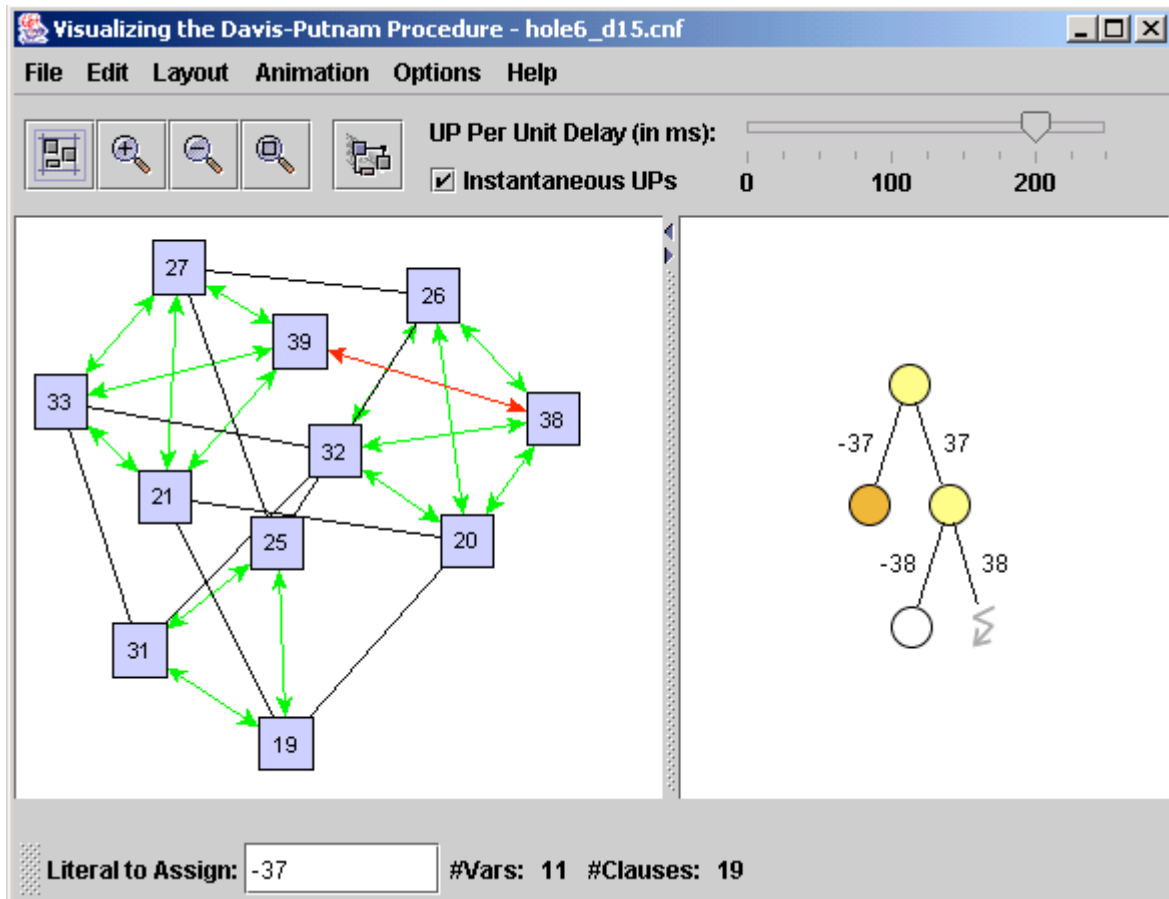


In order to further clarify the concept of the decision tree, we look at the proceeding of the DPLL procedure on a little example and how this is represented in the decision tree. The root node represents the original problem. First of all the algorithm chooses to branch on the decision variable x . First `true` is assigned to x and the position node (painted in orange instead of yellow like the other already visited nodes or white for unvisited nodes) is moved from the root to the right child. Through the assignment the problem simplifies. We assume that the DPLL procedure chooses to assign `false` to y next. This decision

leads to a conflict, which is represented by a flash. As the algorithm has reached a leaf, it must go back and check the alternative assumption (y). Again the algorithm has reached a leaf, but this time the leaf represents a solution. As you can see in the figure the green border of the node indicates the solution.

2 User Interface

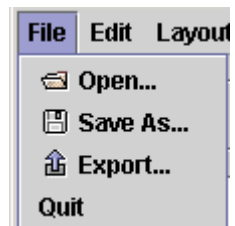
In this section we will describe the user interface of DPvis in order to give you a rough overview of DPvis. Below you see a screenshot of DPvis after opening the problem `hole6_d15.cnf` (you can see the name of the problem in the title of the window) and descending one step in the decision tree. We will now describe the parts of the window in greater detail.



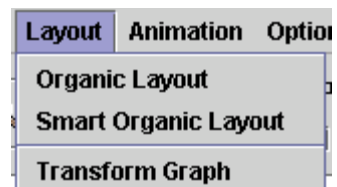
2.1 Menu bar

File Edit Layout Animation Options Help

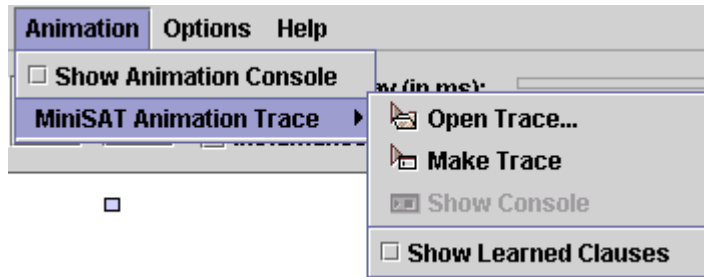
As you can see there are six menus available in the menu bar. First of all there is the **file** menu that allows you to **open** a new problem. With this distribution you have downloaded some example problems of different size in the `cnf\` folder. The “**Save As ...**” menu item allows you to save your (potentially simplified) SAT problem as `cnf` file, whereas you can export your interaction graph in `jpg`, `gif` or `eps` format with the “**Export ...**” menu item.



The **edit** menu has only one menu item that offers a **search** function for variables. Especially if you are working with large problems you can use the **zoom selected** function (see below) to zoom to the selected node that you have searched. The next menu relates to the **layout** of the interaction graph. DPvis offers access to two layout algorithms provided by `yFiles` (see [yFiles]): **Organic Layout** and **Smart Organic Layout**. With both algorithms you can achieve good results in revealing the structure of SAT instances. Unfortunately we haven’t yet found simple parameters that achieve good results with every possible problem. The last menu item allows you to **transform** the interaction graph with actions like rotation or mirroring.

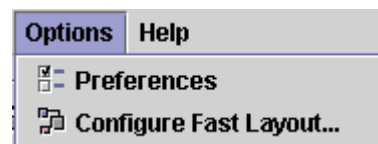


The **animation** menu offers you the possibility to watch the DPLL procedure working on the open SAT instance without user interaction. The first item (**show animation console**) opens an animation console that offers control to the build-in simple SAT solver. The second item

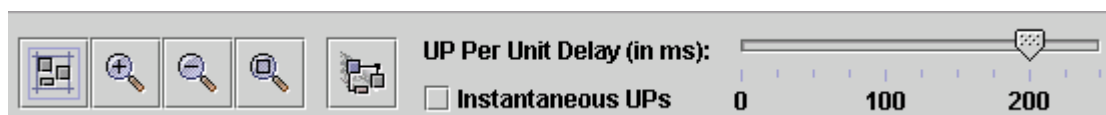


gives access to the interaction with MiniSAT (see [Één, Sörensson]). Along with DPvis you have downloaded a slightly modified version of MiniSAT that produces traces of DPLL runs for the control of DPvis. If you have produced a MiniSAT trace to the currently opened SAT instance outside DPvis you may open it with the “**Open Trace ...**” menu item. If you haven’t got produced a trace you may do it from within DPvis with the “**Make Trace**” menu item. After the execution of MiniSAT the “**Show Console**” menu item will be enabled and an animation console will be shown after clicking on the menu item. If you want to see the learned clauses (as orange edges) during the animation too you may select the checkbox associated with the “**Show Learned Clauses**” menu item.

The **options** menu offers to kind of functionality: First with the **preferences** menu item you may change the (general) layout of the interaction graph such as the shape of the nodes. Second with the “**Configure Fast Layout ...**” menu item you can change the behaviour of the corresponding button (see below).



2.2 Toolbar



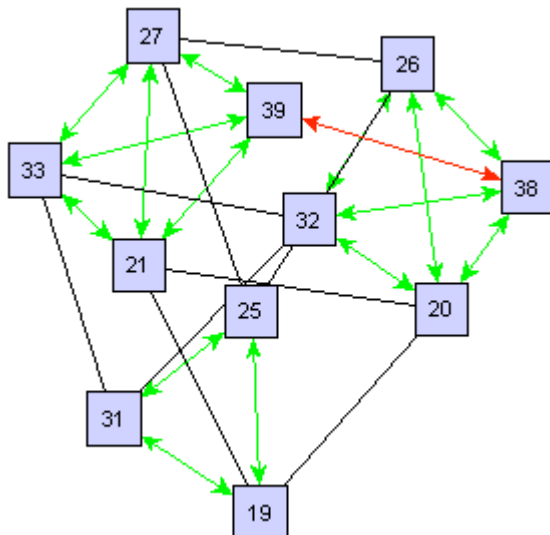
The toolbar below the menu bar consists of two areas. The area on the left contains a group of buttons for controlling the view on the interaction graph. More precisely (from left to right): The interaction graph may be fitted to the view, you can zoom in and out on the interaction graph and the selected nodes of the graph may be zoomed in. The latter may be useful if you have searched (Edit → Search Variable) a variable that is now selected or if you have dragged a selection rectangle with the mouse on an area you want to zoom in.

The rightmost button is the **fast layout** button. If you often want to layout graphs with the same parameters and the same algorithm you may save your settings to this button (Options → Configure Fast Layout) and start the layout algorithm with your settings quickly via this button instead of enter your parameters over and over again in the configuration dialog of the layout algorithm.

Whereas the left side controls the view on the interaction graph the right side of the toolbar influences the visualization of unit propagations. After the user assigned a value to a variable DPvis starts displaying the effects of this assignment graphically. Initially the node representing the chosen variable starts blinking. The frequency and length of this blinking

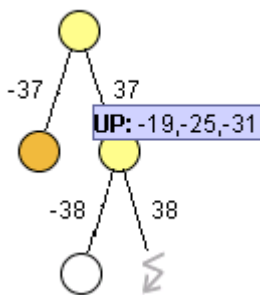
may be set with the slider. After the chosen node has stopped blinking it disappears. If the assignment of the chosen variable causes another variable to become unit this is visualized in the same way as the assignment of the chosen variable by blinking and subsequently disappearing. If you don't want to see all of the nodes blinking and disappearing sequentially, you may choose **"Instantaneous UPs"**. If this checkbox is selected all effects of an assignment are shown at once without any blinking.

2.3 Interaction graph



The user can assign a boolean value to a variable via the interaction graph. If you click the left mouse button on a node the associated variable is assigned the value `true` and if you click with the right mouse button the variable is assigned the value `false`. Automatically an unit propagation is performed after the assignment. Furthermore if you move the mouse over a node or an edge in the interaction graph a tooltip is shown with some information on the variable or clause that induced this node or edge.

2.4 Decision Tree



The decision tree like the interaction graph offers several ways of interaction for the user. Like in the interaction graph tooltips are shown if you move the mouse over an edge (no tooltips are shown over nodes). In the screenshot on the left a tooltip for the edge labeled with 37 is shown. The content of the tooltip tells you which variables have become unit as a consequence of the assignment of the variable 37 with `true`.

Besides this information you may navigate in the search space of the DPLL procedure by clicking on the nodes of the tree. DPvis will compute the appropriate interaction graph for the node you have selected. For the current position node the context menu (click the right mouse button) offers you the possibility to delete the subtree of the node.

2.5 Input field and Statistics



If you don't like starting an unit propagation with mouse clicks you may use the “**Literal to Assign**” input field. The value to assign to the variable is indicated by the sign of the number you enter in the field. If you press ENTER the unit propagation is started.

On the right side of the input field DPvis tells you some statistical information about the problem namely the current number of variables and clauses of the shown SAT instance.

Literature

Dechter, R. und Rish, I. (1994): “Directional resolution: The Davis-Putnam procedure revisited” *Proc. of the 4th Int. Conf. on Principles of KR&R*, 134–145.

Niklar Éen, Niklas Sörensson: “An extensible SAT-solver”

João P. Marques Silva and Karem A. Sakallah: “GRASP – A New Search Algorithm for Satisfiability”

Sinz, Carsten (2004): “Visualizing the internal structure of SAT instances (preliminary report)” *Proc. of the 7th Intl. Conf. on Theory and Applications of Satisfiability Testing (SAT 2004)*.

yFiles: http://www.yworks.com/en/products_yfiles_about.htm