

Interactive Visualization of the Tic Tac Toe Game Graph

Graph Drawing Contest 2015

Evmorfia Argyriou, Michael Baur, Anne Eberle, and Martin Siebenhaller (yWorks GmbH)
{evmorfia.argyriou, michael.baur, anne.eberle, martin.siebenhaller}@yworks.com

General Idea

Since the graph of all possible Tic Tac Toe board states and transitions forms a directed acyclic graph, we choose a classical hierarchical (Sugiyama) layout style for our visualization. The n -th layer contains all nodes that represent board states after the n -th turn.

Our initial experiments with the hierarchical layout algorithm in yFiles quickly showed that a drawing of the complete graph in this way does not produce satisfying results with respect to readability. Therefore, we decided to implement an interactive visualization tool that combines a playable Tic Tac Toe board with an interactive and filtered view of the state graph (see Fig. 1). Our tool is a web application based on yFiles for HTML and can be found here: <http://live.yworks.com/gd2015-contest/TicTacToe/>.

Board and state graph view are synchronized, that means that one updates itself automatically whenever the other is changed by the user. Clicking on the leftmost button of the Tic Tac Toe board view starts a new game. The user can alternatively set X's (red colored) and O's (blue colored). Adding X's or O's will automatically update the state graph view and its layout. The node representing the current board state is highlighted by a rectangular, orange border. Furthermore, the user can directly navigate through the state graph. Clicking on its nodes automatically updates the view including the layout as well as the Tic Tac Toe board view.

Available Views

Neighborhood (recommended). Shows all predecessors that lead to the current state as well as all possible continuations up to three levels.

Optimal. Only shows nodes that represent states that may appear with best play of both players (see Fig. 2). Note that for board states that are not contained in the optimal subgraph, the state graph view only shows a single node.

Complete. Always shows the complete graph.

Visual Features

Node Visualization. The interior of each node clearly displays the associated board state. Since the symbols X and O are not distinguishable at small sizes, we use solid cells of different colors (red and blue).

Auto-rotated Board State. To avoid confusion due to the class of symmetric board positions that each node represents, each node shows the (rotated/mirrored) board state that corresponds to the state of the current board. In other words, the state displayed in two adjacent nodes differs in exactly one cell (unless one of the nodes has several matching states).

Winning State. Nodes are sometimes decorated with a small, circular icon on their upper right corner. The color of the icon indicates which party (red/blue) wins the game from this state assuming best play of both players. If there is no such icon, the game ends in a draw (again assuming best play of both players), a violet colored icon indicates a safe draw state which means that nobody can win independent of the continuation (without assuming best play). The winning states where determined by a graph analysis starting at the nodes representing the final states and walking up to the start node. Furthermore, a red/blue colored border around a node indicates a final state where one player has already won the game.

Edge Color. The color of the outgoing edges of a node (incident on the nodes' bottom) indicates which player is to move.

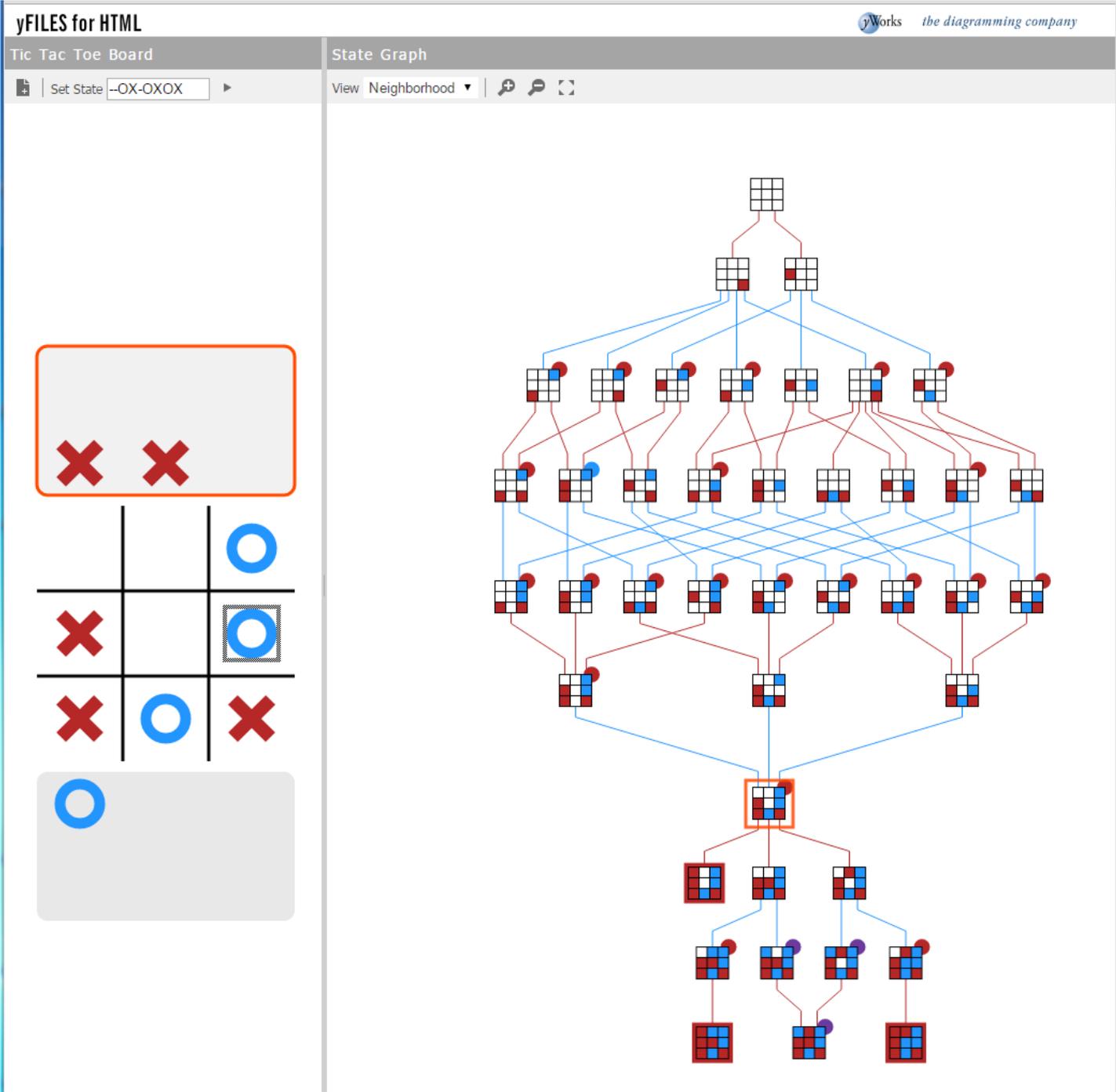


Fig. 1. Screenshot of our web application.

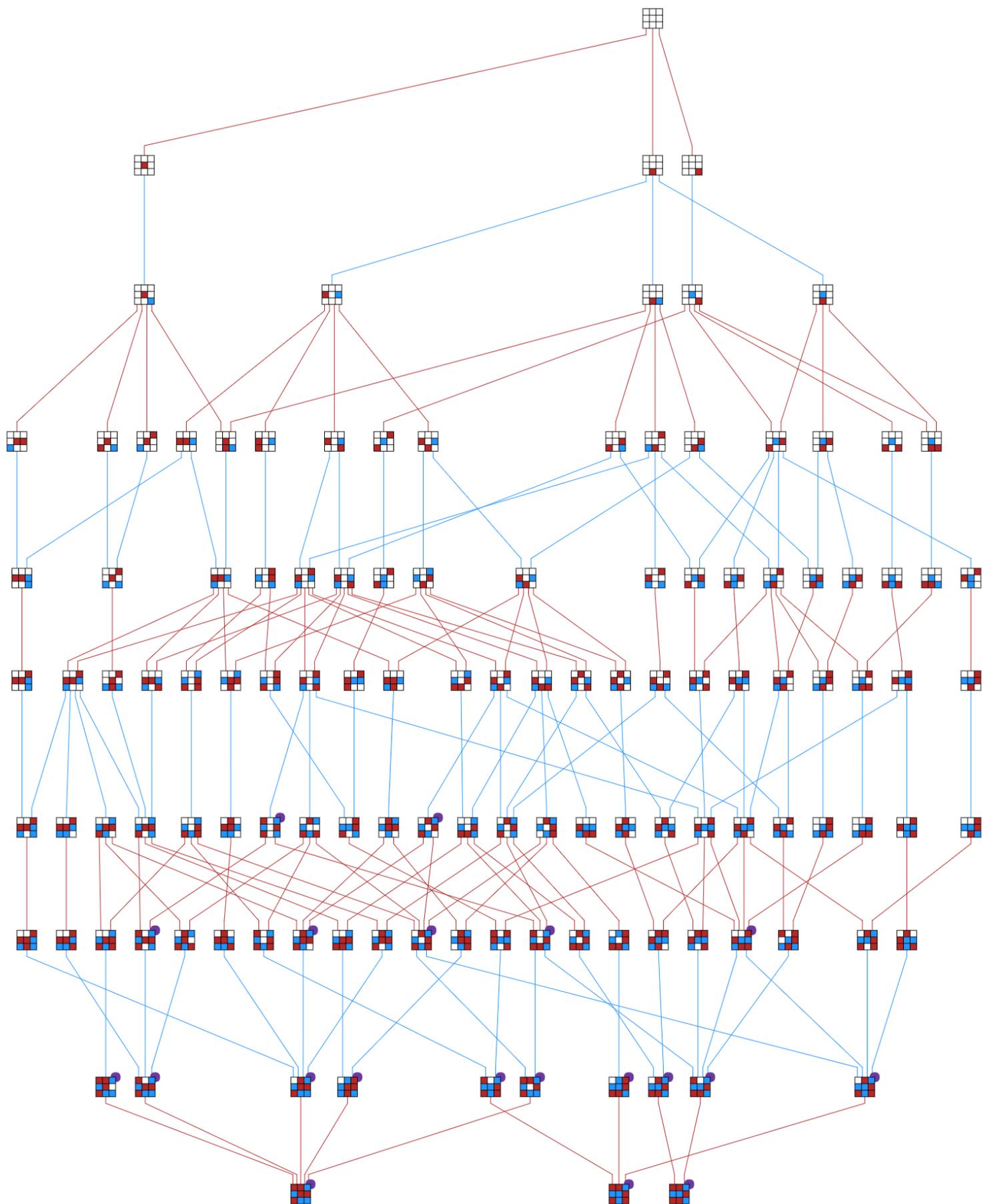


Fig. 2. The optimal subgraph that shows all nodes representing states that may appear when we assume best play of both parties. It is well-known that in this case the game always ends in a draw. The diagram also shows that after the 6-th turn there are already states (represented by nodes with circular icon on the upper right corner) in which no party can win independently of the continuation.

