Visualization of Problem Solving with Constraint Processing: Case Study of the Minesweeper

By Chase Resio and Berthe Y. Choueiry
Minesweeper and NP-Completeness

- Minesweeper is a popular computer game from the 1990’s
- Click on cells to reveal a mine or a number indicating how many adjacent cells contain mines
- Deciding whether an instance of Minesweeper consistent is NP-Complete
• Use Constraint Processing to model and solve Minesweeper
• Study Constraint Processing
• Learn and use React framework
• Solve an NP-Complete problem
Approach

- Minesweeper is a logic-based game
- Model it as a Constraint Satisfaction Problem
- Every cell is either safe or a mine
- Safe cells contain a number that constrains neighboring cells
- Use constraint satisfaction approaches to decide if a cell is safe

- The 1 places a constraint on cells A,B,C,G,H,K,L,M that one of them must contain a mine and the rest are safe.
- The 3 places a constraint on cells D,E,F,I,J,N,O,P that three of them must contain a mine and the other five are safe.
Six levels of consistency implemented:

- **Unary** ensures the consistency of each single constraint.
- **GAC** ensures the consistency of each constraint and propagates to other constraints via the shared variables.
- **2wC** ensures the consistency of every combination of two constraints with shared cells.
- **3WC** ensures the consistency of every combination of three constraints that share cells.

- **4WC** ensures the consistency of every combination of four constraints that share cells.

- **Backbone** ensures the consistency by finding the cell that has the same value in all solutions.
• There is currently no way to efficiently solve all instances
• Finding one would have profound impact on computing

Future improvements
• Better mobile view
• Use number of mines to solve cells once at steady state
Acknowledgments

 Developers: Kenneth Bayer, Tomo Bessho, Taylor DeMint, Joshua Snyder, and Robert Woodward

UNL UCARE

NSF REU supplements grant RI-1619344