# Bejeweled, Candy Crush and other Match-Three Games are (NP-)Hard

Luciano Gualà, Stefano Leucci, **Emanuele Natale**

University of Rome Tor Vergata, University of L'Aquila, Sapienza University of Rome

IEEE Conference on Computational Intelligence and Games
Dortmund, 26-29 August 2014

# The Computational Complexity of Games

Given a game $A$ that is engaging to play, it is often the case that each problem $B$ in the complexity class $NP$ (or in $PSPACE$) can be trasfomed (i.e. reduced) in polinomial time to an instance $m$ of $A$ such that you can solve $B$ by playing $m$. (Kendall '08, Hearn '09, Forišek '10, Viglietta '12).

# Casual Games

New public of casual players
looking for "soft" gaming:

- easy to play
  $\downarrow$
  simple rules;
- engaging
  $\downarrow$
  complex structure.

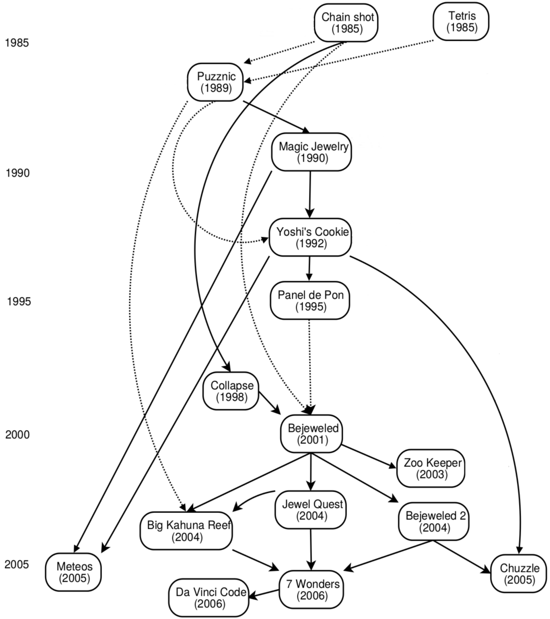
# Casual Games

New public of casual players looking for ”soft” gaming:

- easy to play
  ↓
  simple rules;
- engaging
  ↓
  complex structure.

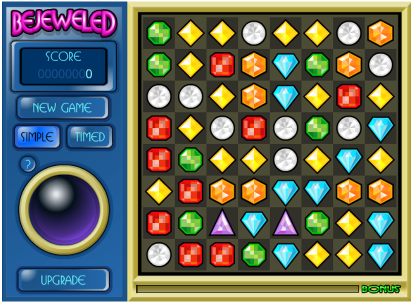
A big class of Casual Games:

Match Three Games.



# The Match-Three Game: Bejeweled

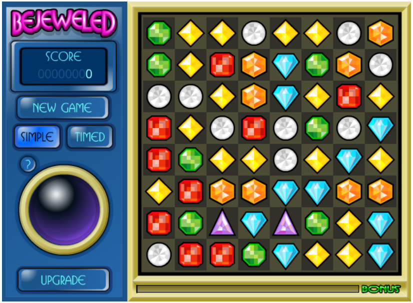
- Played on a 8 × 8 grid filled with gems of 6 types.
- Three or more vertically or horizontally aligned and contiguous gems are said to form a match.



# **The** Match-Three Game: Bejeweled

## The game mechanic:

- **Moving phase:** The player swap two (vertically or horizontally) adjacent gems provided that doing so will create a match, then the popping phase take place. If it is not possible to make such a move, the game is over.
- **Popping phase:** As there is any match, the matched gems pop simultaneously and the remaining gems fall filling the empty space; when there are no more matches, the moving phase take place.



# The Reduction - Preliminaries

## General Bejeweled

Bejeweled played on a $n \times n$ grid (still 6 gems only!).

## Main decision problem

Can we pop a specific gem?
Implies:

- Can we get a score of at least $s$?
- Can we get a score of at least $s$ in less than $k$ moves?
- Can we cause at least $p$ gems to pop?
- Can we play for at least $t$ turns?

# The Reduction - Preliminaries

## General Bejeweled

Bejeweled played on a $n \times n$ grid (still 6 gems only!).

## Main decision problem

Can we pop a specific gem?

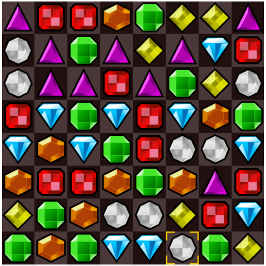
## Main difficulty

Pops affects everything above.

# The Reduction - Preliminaries

### General Bejeweled

Bejeweled played on a $n \times n$ grid (still 6 gems only!).



### Main decision problem

Can we pop a specific gem?

### Main difficulty

Pops affects everything above.

# The Reduction - Preliminaries

### General Bejeweled

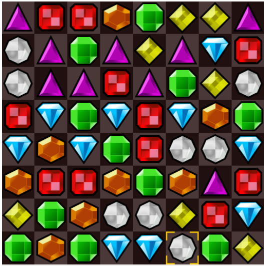
Bejeweled played on a $n \times n$ grid (still 6 gems only!).

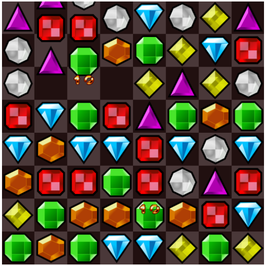
### Main decision problem

Can we pop a specific gem?

### Main difficulty

Pops affects everything above.





# The Reduction - Preliminaries

### General Bejeweled

Bejeweled played on a $n \times n$ grid (still 6 gems only!).

### Main decision problem

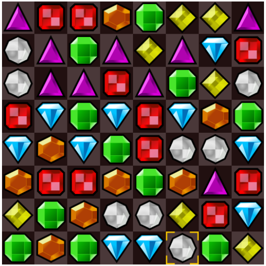
Can we pop a specific gem?

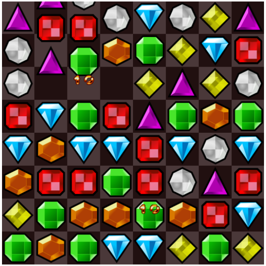
### Main difficulty

Pops affects everything above.

### Strategy

- Preserve structure by modularity
- Make swaps irreversible





# The Reduction from 1-in-3 Positive SAT

## Instance

- $n$ variables $x_1, \ldots, x_n$;
- $m$ clauses with at most 3 variables each.

## Goal

An assignment that satisfies all clauses by setting exactly one variable to true for each of them.

## Example

Instance:

$$(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5)$$

Bad assignment:

$$\begin{cases} x_1, x_2 & \leftarrow \textit{true} \\ x_3, x_4, x_5 & \leftarrow \textit{false} \end{cases}$$

Good assignment:

$$\begin{cases} x_2 & \leftarrow \textit{true} \\ x_1, x_3, x_4, x_5 & \leftarrow \textit{false} \end{cases}$$

# The Reduction from 1-in-3 Positive SAT

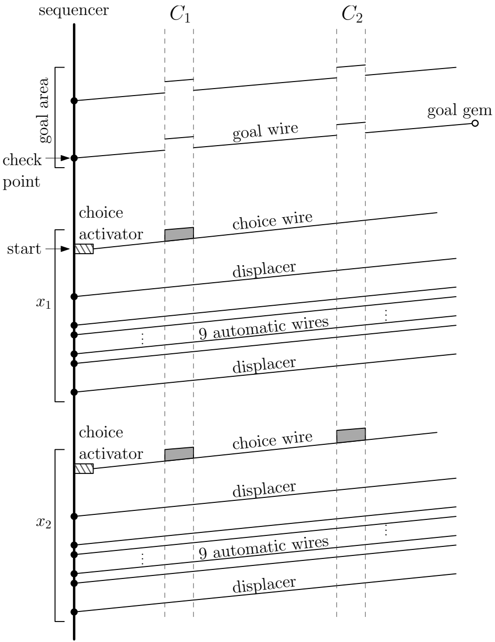
## Instance

- $n$ variables $x_1, \ldots, x_n$;
- $m$ clauses with at most 3 variables each.

## Goal

An assignment that satisfies all clauses by setting exactly one variable to true for each of them.

## 1-in-3 positive SAT embedding in Bejeweled:

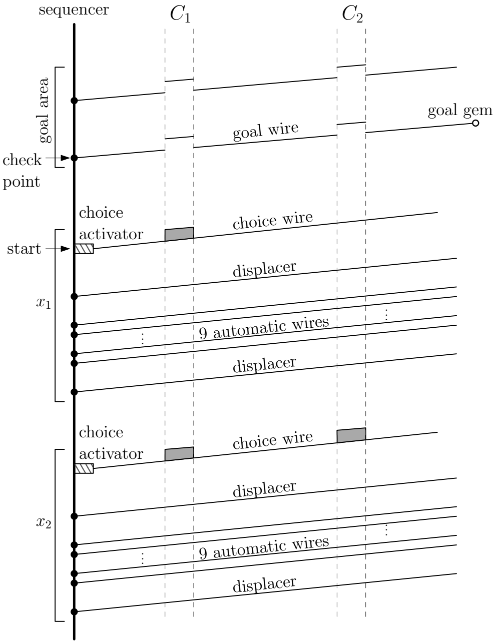


# The Gadgets - (Partial) Overview

Choice wire

Sequencer

Goal wire



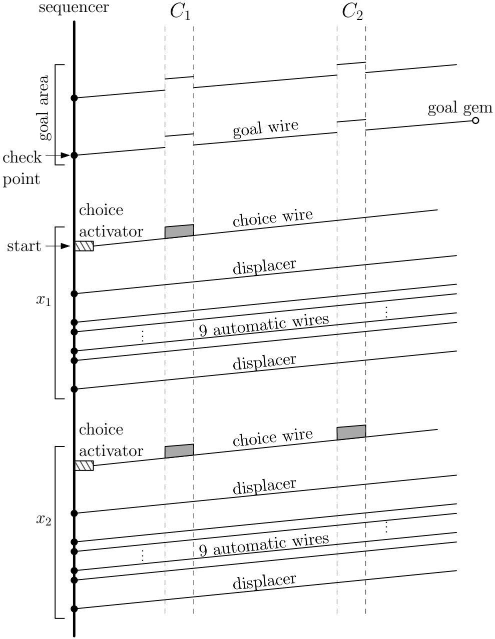
# The Gadgets - (Partial) Overview

## Choice wire

The choice wire is either activated or skipped by swapping a gem in the choice activator. The activation shift some *clause column* by two rows, constructing a truth assignment.

## Sequencer

## Goal wire



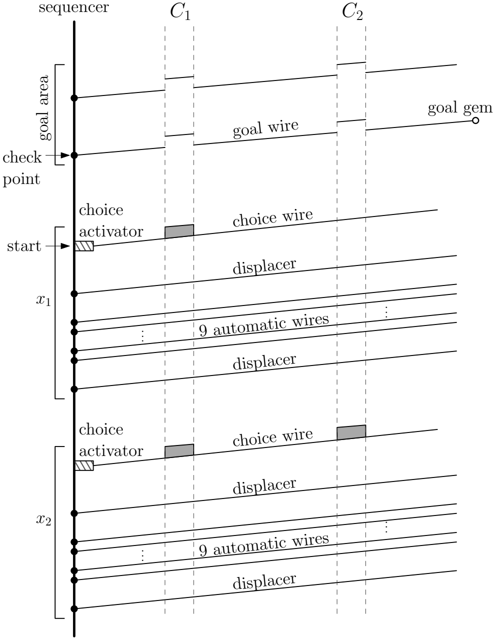
# The Gadgets - (Partial) Overview

## Choice wire

## Sequencer

The sequencer make possible only to swap gems placed in the choice activator (from the topmost to the bottommost).

## Goal wire



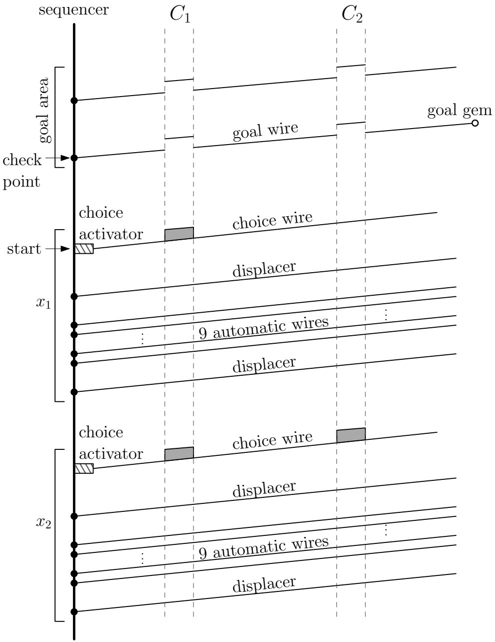
# The Gadgets - (Partial) Overview

## Choice wire

## Sequencer

## Goal wire

Toward the end of the game, an activating gem ends up in the check point. A sequence of swaps/pops reaching the goal gem along the goal wire will take place if and only if all clause are satisfied.



# The Gadgets - Filler and Sequencer

In the filling pattern no match can be formed, even if the column fall by different amounts.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| C | E | C | E | C | E | C |
| D | F | D | F | D | F | D |
| C | E | C | E | C | E | C |
| D | F | D | F | D | F | D |
| C | E | C | E | C | E | C |
| D | F | D | F | D | F | D |
| C | E | C | E | C | E | C |

# The Gadgets - Filler and Sequencer

In the filling pattern no match can be formed, even if the column fall by different amounts.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| C | E | C | E | C | E | C |
| D | F | D | F | D | F | D |
| C | E | C | E | C | E | C |
| D | F | D | F | D | F | D |
| C | E | C | E | C | E | C |
| D | F | D | F | D | F | D |
| C | E | C | E | C | E | C |

The sequencer controls the order in which the other gadgets are activated.

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| A | · | · | · | · |
| A | · | · | · | · |
| C | · | · | · | · |
| D | · | · | · | · |
| A | · | · | · | · |
| A | · | · | · | · |
| B | · | · | · | · |
| C | · | · | · | · |
| D | · | · | · | · |
| C | · | · | · | · |
| A | A | A | · | · |
| B | B | B | · | · |
| C | · | · | · | · |
| C | · | · | · | · |
| D | · | · | · | · |
| D | A | A | · | · |
| C | A | A | · | · |
| C | B | B | · | · |
| D | · | · | · | · |
| D | · | B | · | · |
| C | A | A | · | · |
| C | A | A | B | B |

# The Gadgets - The Choice Wire

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B | · | · | B | B | · | · | · | B | · | · | B | B | · |
| A | · | · | A | A | · | · | · | A | · | · | A | A | · |
| C | · | · | · | · | · | B | · | · | B | B | · | · | · |
| B | · | B | · | · | · | A | · | · | A | A | · | · | · |
| A | · | A | · | B | · | · | B | B | · | · | · | · | · |
| D | · | · | · | A | · | · | A | A | · | · | · | · | · |
| C | · | · | · | · | B | B | · | · | · | · | · | · | · |
| B | · | · | · | · | A | A | · | · | · | · | · | · | · |
| A | · | · | B | B | · | · | · | · | · | · | · | · | · |
| A | · | · | A | A | · | · | · | · | · | · | · | · | · |
| C | · | · | · | · | · | · | · | · | · | · | · | · | · |
| C | · | B | · | · | · | · | · | · | · | · | · | · | · |
| B | B | B | · | · | · | · | · | · | · | · | · | · | · |
| A | A | A | · | · | · | · | · | · | · | · | · | · | · |
| A | · | A | · | · | · | · | · | · | · | · | · | · | · |
| C | · | B | · | · | · | · | · | · | · | · | · | · | · |
| B | · | B | · | · | · | · | · | · | · | · | · | · | · |
| B | · | A | · | · | · | · | · | · | · | · | · | · | · |
| C | · | A | · | · | · | · | · | · | · | · | · | · | · |
| C | · | · | · | · | · | · | · | · | · | · | · | · | · |
| D | B | B | · | · | · | · | · | · | · | · | · | · | · |
| D | A | A | · | · | · | · | · | · | · | · | · | · | · |
| C | B | B | · | · | · | · | · | · | · | · | · | · | · |
| C | A | A | · | · | · | · | · | · | · | · | · | · | · |

Each choice wire corresponds to a variable $x_i$; if activated, it sets $x_i$ to true by making all clauses containing $x_i$ fall by some number $l \equiv 2 \pmod 6$ (while others fall by multiplies of 6).

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | · | · | · | · | · | · | B | B | · | · | · | · | · | · | · | · | · | · | · | · |
| C | · | · | · | · | · | · | A | A | · | · | · | · | · | · | · | · | · | · | · | · |
| D | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | B | · | · |
| D | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | A | · | · |
| C | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | B | · | · | · | · |
| C | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | A | · | · | · | · |
| D | · | · | · | · | · | · | · | · | B | · | · | · | · | B | · | · | · | · | · | · |
| D | · | · | · | · | · | · | · | · | A | · | · | · | · | A | · | · | · | · | · | · |
| C | · | · | · | · | · | · | · | · | · | · | · | B | · | · | · | · | · | · | B | B |
| C | · | · | · | · | · | · | · | · | · | · | · | A | · | · | · | · | · | · | A | A |
| D | · | · | · | · | · | · | · | · | · | B | · | · | · | · | · | · | B | B | · | · |
| D | · | · | · | · | · | · | · | · | · | A | · | · | · | · | · | · | A | A | · | · |
| C | · | · | · | · | · | · | · | B | · | · | · | · | · | · | B | B | · | · | · | · |
| C | · | · | · | · | · | · | · | A | · | · | · | · | · | · | A | A | · | · | · | · |
| D | · | · | · | · | · | B | · | · | · | · | · | · | B | B | · | · | · | B | · | · |
| D | · | · | · | · | · | A | · | · | · | · | · | · | A | A | · | · | · | A | · | · |
| C | · | · | · | B | · | · | · | · | · | · | B | B | · | · | · | B | · | · | · | · |
| C | · | · | · | A | · | · | · | · | · | · | A | A | · | · | · | A | · | · | · | · |
| D | · | · | · | · | · | · | · | · | B | B | · | · | · | B | · | · | · | · | B | B |
| D | · | B | · | · | · | · | · | · | A | A | · | · | · | A | · | · | · | · | A | A |
| C | · | A | · | · | · | · | B | B | · | · | · | B | · | · | · | · | B | B | · | · |
| C | · | · | · | · | · | · | A | A | · | · | · | A | · | · | · | · | A | A | · | · |
| D | · | · | · | · | B | B | · | · | · | B | · | · | · | · | B | B | · | · | · | · |
| D | · | · | · | · | A | A | · | · | · | A | · | · | · | · | A | A | · | · | · | · |
| C | · | · | B | B | · | · | · | B | · | · | · | · | B | B | · | · | · | B | · | · |
| C | · | · | A | A | · | · | · | A | · | · | · | · | A | A | · | · | · | A | · | · |
| D | · | · | · | · | · | B | · | · | · | · | B | B | · | · | · | B | · | · | B | B |
| D | · | · | · | · | · | A | · | · | · | · | A | A | · | · | · | A | · | · | A | A |
| C | · | · | · | B | · | · | · | · | B | B | · | · | · | B | · | · | B | B | · | · |
| C | · | B | · | A | · | · | · | · | A | A | · | · | · | A | · | · | A | A | · | · |
| D | · | A | · | · | · | · | B | B | · | · | · | B | · | · | B | B | · | · | · | · |
| D | · | · | · | · | · | · | A | A | · | · | · | A | · | · | A | A | · | · | · | · |
| C | · | · | · | · | B | B | · | · | · | B | · | · | B | B | · | · | · | · | · | · |
| C | · | · | · | · | A | A | · | · | · | A | · | · | A | A | · | · | · | · | · | · |
| D | · | · | B | B | · | · | · | B | · | · | B | B | · | · | · | · | · | · | · | · |
| D | · | · | A | A | · | · | · | A | · | · | A | A | · | · | · | · | · | · | · | · |

# The Gadgets - The Goal Wire

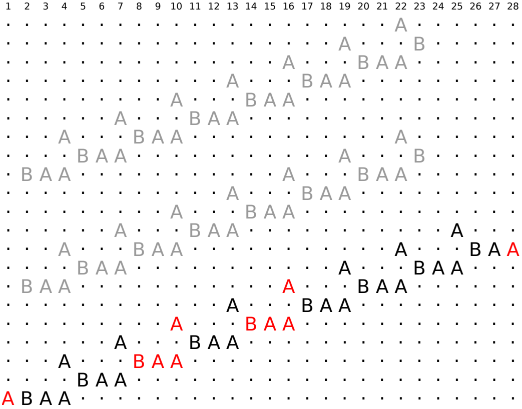
Once we have traversed all the variable gadgets, the sequencer gives the "control" to the goal wire.

```
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  ·  ·  A  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  ·  ·  ·  ·  ·
  ·  B  A  A  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·
  ·  ·  ·  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A
  ·  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·
  ·  B  A  A  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  ·  ·  A  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·
  ·  ·  ·  ·  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·
  A  B  A  A  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·  ·
```

The goal wire ensure that the goal gem can be reached from the check point only if the choice wire activations result in a satisfying assignment.

# The Gadgets - The Goal Wire

Once we have traversed all the variable gadgets, the sequencer gives the "control" to the goal wire.



```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
. . . . . . . . . . . . . . . . . . . . . A . . . . . .
. . . . . . . . . . . . . . . . . . A . . . B . . . . .
. . . . . . . . . . . . . . . A . . . B A A . . . . . .
. . . . . . . . . . . . A . . . B A A . . . . . . . . .
. . . . . . . . . A . . . B A A . . . . . . . . . . . .
. . . . . . A . . . B A A . . . . . . . . . . . . . . .
. . . A . . . B A A . . . . . . . . . . . A . . . . . .
. . . . B A A . . . . . . . . . . . A . . . B . . . . .
. B A A . . . . . . . . . . . A . . . B A A . . . . . .
. . . . . . . . . . . . A . . . B A A . . . . . . . . .
. . . . . . . . . A . . . B A A . . . . . . . . . . . .
. . . . . . A . . . B A A . . . . . . . . . . . A . . .
. . . A . . . B A A . . . . . . . . . . . A . . . B A A
. . . . B A A . . . . . . . . . . . A . . . B A A . . .
. B A A . . . . . . . . . . . A . . . B A A . . . . . .
. . . . . . . . . . . . A . . . B A A . . . . . . . . .
. . . . . . . . . A . . . B A A . . . . . . . . . . . .
. . . . . . A . . . B A A . . . . . . . . . . . . . . .
. . . A . . . B A A . . . . . . . . . . . . . . . . . .
. . . . B A A . . . . . . . . . . . . . . . . . . . . .
A B A A . . . . . . . . . . . . . . . . . . . . . . . .
```

The goal wire ensure that the goal gem can be reached from the check point only if the choice wire activations result in a satisfying assignment.

# Open Questions

What about the complexity of Bejeweled with only 5 kind of gems (or even less)?

# Open Questions

What about the complexity of Bejeweled with only 5 kind of gems (or even less)?

What about other match-three-games (no swaps, no square grid, . . . e.g. Puzzle Bobble)?

# Open Questions

What about the complexity of Bejeweled with only 5 kind of gems (or even less)?

What about other match-three-games (no swaps, no square grid, . . . e.g. Puzzle Bobble)?

Is Bejeweled PSPACE-complete?