Quantified Boolean Formulae: Call the Plumber!

Josef Lindsberger, Alexander Maringele, Georg Moser

Department of Computer Science, Universität Innsbruck, Austria
josef.lindsberger@web.de,
{alexander.maringele, georg.moser}@uibk.ac.at

LPAR-21
May 9th, 2017
Motivation

Public relations and teaching

- https://www.uibk.ac.at/lndf/
- https://www.uibk.ac.at/jungeuni/
- Logic 101
- Is the game beatable?
- Is the game tough?
- How can we generate game instances?
Super Formula World

*SFW* is a simple platform game with two-dimensional game maps. It uses the game mechanics of Nintendo’s Super Mario World, which introduced more complex challenges than its predecessors.

*SFW* encodes a reduction from the satisfiability of (quantified) Boolean formulae to graphs of gadgets.

Our hero Mario has to cope with maps drawn from such graphs to rescue princess Peach, while in fact he is being exploited as a (Q)SAT solver.
Outline

- Satisfiability of QBF
- NP framework
- PSPACE framework
- Summary
- Demo
Satisfiability of QBF

\[ \text{QSAT}(\phi) \]

\[ \forall y \exists z (x \land (y \lor \bar{z}) \land (\bar{y} \lor z)) \]

\[ \forall y \exists z ((y \lor \bar{z}) \land (\bar{y} \lor z)) \]

\[ \exists z (z) \]

\[ \exists z (\bar{z}) \]

\[ x \ y \ z \ \bar{y} \ \bar{z} \]
NP Framework

φ = (x ∨ y ∨ z) ∧ (¬x ∨ ¬y) ∧ (x ∨ y ∨ ¬z), R_{NP}(φ):

SAT(φ) ⇔ R_{NP}(φ) beatable
Variables and Literal Paths
Bricked and Traversable Clause

\((r \lor \bar{s} \lor t)\)
PSPACE Framework

\[ \phi = \forall x \exists y (x \lor \neg y) \land (\neg x \lor y), \quad R_{P\lor}(\phi) : \]

\[
\begin{align*}
\text{S} & \quad \forall x \quad \text{open } x \quad \exists y \quad \text{open } y \\
& \quad \forall x \quad \text{close } \neg x \quad \exists y \quad \text{close } \neg y
\end{align*}
\]

\[
\begin{align*}
\text{QSAT}(\phi) & \iff R_{P\lor}(\phi) \text{ beatable}
\end{align*}
\]
Closed and Traversable Door
Clause with Doors

- open the door
- close the door
- traverse the clause
- traverse the door

\[ x \lor \bar{y} \]
Existential quantifier

enter $\ell^+$
open: $\rightarrow$ traverse: $\rightarrow$ close: $\rightarrow$
next ... return
previous
...
enter $\ell^-$
open: $\rightarrow$ traverse: $\rightarrow$ close: $\leftarrow$
next ... return
previous
Universal quantifier

\[
\ell^+ \quad \text{open:} \leftarrow \\
\text{open:} \rightarrow \text{traverse:} \rightarrow \text{close:} \rightarrow \\
\text{next} \ldots \text{return} \\
\text{traverse:} \leftarrow \text{close:} \leftarrow \\
\ell^- \quad \text{open:} \leftarrow \\
\text{open:} \rightarrow \text{traverse:} \rightarrow \text{close:} \rightarrow \\
\text{next} \ldots \text{return} \\
\text{traverse:} \leftarrow \text{close:} \leftarrow \\
\text{previous}
\]
Summary
Super Formula World

We have

- Refinements of the NP and PSPACE frameworks established by Aloupis et al.
- Instantiations of the frameworks for Super Mario World with precise definitions and fixed topologies
- Running implementations of the frameworks
- Exposition of computational complexity of video games
- Improvable game experience
  - Game play is too repetitive
  - Maps are limited to prenex CNF
  - User cannot be smart
Super Formula World

Life Demo

- NP framework

or

- PSPACE framework

\[ \bar{x} \lor y \]

\[ \forall x (\bar{x} \lor y) \]
Future Work

- Less repetition
- Smaller maps
- Construct from NNF
References
