SyMT: finding symmetries in SMT formulas
(Work in progress)

Carlos Areces, David Déharbe,
Pascal Fontaine, and Ezequiel Orbe

UFRN (Natal, Brazil)
Loria, INRIA, Université de Nancy (France)
FaMaF (Córdoba, Argentina)

SMT, July 8-9, 2013

Outline

Introduction

Satisfiability solving:

- problem encoding of primal importance
- doing many times the same thing is a waste of time

Previously (CADE 2011):

- breaking symmetries on QF_UF gives impressive results

In this talk:

- beyond tailored heuristics
- generalize symmetry finding

Outline

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

- Let's satisfy every clause:
- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

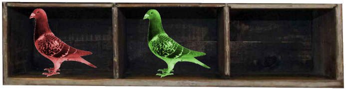
- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

- Let's satisfy every clause:
  $p_1 = B_1$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

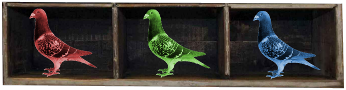
- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$



- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2$
- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$



- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2, p_3 = B_3$
- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

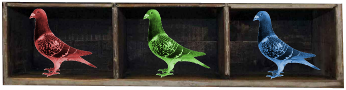
- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2, p_3 = B_3, p_4 = ?$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$



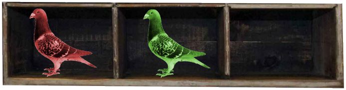
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2, p_3 = B_3$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

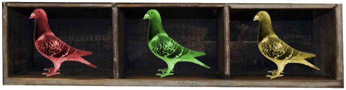


- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$



- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2, p_4 = B_3$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

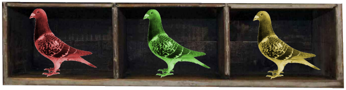
- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2, p_4 = B_3, p_3 = ?$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

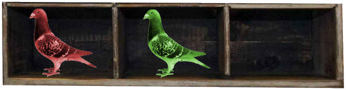


- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2, p_4 = B_3$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$



- Let's satisfy every clause:
  $p_1 = B_1, p_2 = B_2$
- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

- Let's satisfy every clause:
  $p_1 = B_1$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$

- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

- Let's satisfy every clause:
  $p_1 = B_1, p_3 = B_2$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Symmetry breaking: break a factorial

- 4 distinct pigeons:
  $p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge$
  $p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4$
- every pigeon in a hole:
  $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$
  $p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3$
  $p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3$
  $p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3$

- Let's satisfy every clause:
  $p_1 = B_1$

- Whatever the colors, there will always be one pigeon out
- symmetries imply many similar reasoning paths
- detecting symmetries *a priori*: search one path out of many
- large decrease of search space, large decrease in solving times

Previous technique

The formula:

$$
\begin{array}{l}
p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4 \\
p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3 \\
p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3 \\
p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3 \\
p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3 \\
p_1 = B_1 \\
p_2 = B_2
\end{array}
$$

- B_1, B_2, B_3 appear the same number of times, in the same number of clauses,...
- formula preserved by permutation of B_1, B_2, B_3 ($\vee$, $\wedge$ commutative)
- $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$ holds, so let's say $p_1 = B_1$
- resulting formula is still symmetric w.r.t. B_2, B_3. Repeat

Previous technique

The formula:

$$p_1 \neq p_2 \land p_1 \neq p_3 \land p_1 \neq p_4 \land p_2 \neq p_3 \land p_2 \neq p_4 \land p_3 \neq p_4$$
$$p_1 = B_1 \lor p_1 = B_2 \lor p_1 = B_3$$
$$p_2 = B_1 \lor p_2 = B_2 \lor p_2 = B_3$$
$$p_3 = B_1 \lor p_3 = B_2 \lor p_3 = B_3$$
$$p_4 = B_1 \lor p_4 = B_2 \lor p_4 = B_3$$
$$p_1 = B_1$$
$$p_2 = B_2$$

- B_1, B_2, B_3 appear the same number of times, in the same number of clauses,...
- formula preserved by permutation of B_1, B_2, B_3 ($\lor$, $\land$ commutative)
- $p_1 = B_1 \lor p_1 = B_2 \lor p_1 = B_3$ holds, so let's say $p_1 = B_1$
- resulting formula is still symmetric w.r.t. B_2, B_3. Repeat

Previous technique

The formula:

$$
\begin{array}{l}
p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4 \\
p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3 \\
p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3 \\
p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3 \\
p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3 \\
p_1 = B_1 \\
p_2 = B_2
\end{array}
$$

- B_1, B_2, B_3 appear the same number of times, in the same number of clauses,...
- formula preserved by permutation of B_1, B_2, B_3 ($\vee$, $\wedge$ commutative)
- $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$ holds, so let's say $p_1 = B_1$
- resulting formula is still symmetric w.r.t. B_2, B_3. Repeat

Previous technique

The formula:

$$
\begin{array}{l}
p_1 \neq p_2 \wedge p_1 \neq p_3 \wedge p_1 \neq p_4 \wedge p_2 \neq p_3 \wedge p_2 \neq p_4 \wedge p_3 \neq p_4 \\
p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3 \\
p_2 = B_1 \vee p_2 = B_2 \vee p_2 = B_3 \\
p_3 = B_1 \vee p_3 = B_2 \vee p_3 = B_3 \\
p_4 = B_1 \vee p_4 = B_2 \vee p_4 = B_3 \\
p_1 = B_1 \\
p_2 = B_2
\end{array}
$$

- B_1, B_2, B_3 appear the same number of times, in the same number of clauses,...
- formula preserved by permutation of B_1, B_2, B_3 ($\vee$, $\wedge$ commutative)
- $p_1 = B_1 \vee p_1 = B_2 \vee p_1 = B_3$ holds, so let's say $p_1 = B_1$
- resulting formula is still symmetric w.r.t. B_2, B_3. Repeat

Previous technique: weaknesses

Besides being sensitive to obfuscation:

- finding symmetries highly heuristic: guess, and then check symmetry
- symmetry breaking tailored to special case: not easily generalizable

But impressive improvements on QF_UF: worth trying to extend

Goal: being more general in finding and breaking symmetries

Outline

Finding symmetries: graph isomorphism

Graph isomorphism problem

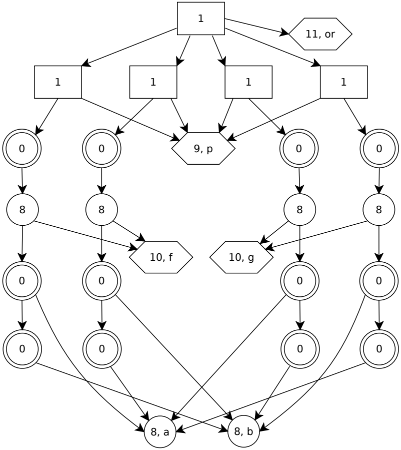
Finding non trivial renaming of nodes resulting in isomorphic graph

- graph isomorphism finding is in NP. P? NPC?
- efficient algorithms exist (time never an issue in our experiments)
- formulas: not exactly graphs, but can easily be translated
- btw, isomorphism finding for DAGs is not simpler
- good tools: saucy, bliss

Finding symmetries: from formulas to (acyclic) graphs

$$p(f(a,b)) \vee p(f(b,a)) \vee p(g(a,b)) \vee p(g(b,a))$$

- node colored by (HOL) sort
- node for every symbol
- node for each subterm
- special nodes "0" for lists of terms
- node for term linked to top symbol and list of subterms
- commutative symbols: directly link to subterms
- maximal sharing



Graph isomorphic iff formula symmetric

Finding symmetries: the tool

- parse formula
- simplify (several options)
- graph isomorphism tool: saucy, bliss
- output group generators

```
./SyMT -enable-simp smt-lib2/QF_UF/NEQ/NEQ004_size4.smt2
    (p7 p9)(c12 c13)
    (c_3 c_1)
    (c_2 c_1)
    (c_0 c_1)
```

SMT-LIB and symmetries

Category	#Inst	#Sym[P]	Avg[GS]	Time
AUFLIA	6480	6258	134.00	378.79
AUFLIRA	19917	16476	1.08	9.13
AUFNIRA	989	985	1.00	0.41
QF_AUFLIA	1140	78	1.00	0.72
QF_AX	551	22	1.00	0.37
QF_IDL	1749	756	12745.43	327.95
QF_LIA	5938	1200	104.55	486.19
QF_LRA	634	210	110.49	29.06
QF_NIA	530	169	5.92	3.92
QF_NRA	166	43	1.00	0.23
QF_RDL	204	24	0.00	10.13
QF_UF	6639	3638	44.00	34.58
QF_UFIDL	431	189	1.00	2.70
QF_UFLIA	564	198	0.00	0.45
UFNIA	1796	1070	47.08	543.26

Outline

Teaser: symmetry simplification

Issues with symmetries:

- graph isomorphism tools provide (small number of) generators
- maybe redundant even if…
- …some kind of non redundancy property holds
- figuring out what generators mean can be difficult

Work in progress:

- simplify generators
- identify subgroups that are full permutation groups on some symbols
- computational group theory: Schreier-Sims (polynomial)

E.g.

```
./SyMT -enable-simp smt-lib2/QF_UF/NEQ/NEQ004_size4.smt2
    (p7 p9)(c12 c13)
    (c_3 c_1)
    (c_2 c_1)
    (c_0 c_1)
```

Teaser: symmetry simplification

Issues with symmetries:

- graph isomorphism tools provide (small number of) generators
- maybe redundant even if…
- …some kind of non redundancy property holds
- figuring out what generators mean can be difficult

Work in progress:

- simplify generators
- identify subgroups that are full permutation groups on some symbols
- computational group theory: Schreier-Sims (polynomial)

E.g.

```
./SyMT -enable-simp smt-lib2/QF_UF/NEQ/NEQ004_size4.smt2
    (p7 p9)(c12 c13)
    [c_0 c_1 c_2 c_3]
```

Teaser: symmetry breaking

Symmetry breaking for propositional logic? Set of formulas:

$$\psi_{i,\sigma} =_{\text{def}} \Big(\bigwedge_{1 \le j < i} p_j \equiv p_j\sigma \Big) \Rightarrow (p_i \Rightarrow p_i\sigma).$$

assuming an order on propositional variables

- SBP on SAT: large, need advanced techniques
- working on recasting to SMT
- SMT symmetries are more “structural”
- hopefully easier to break efficiently (?)
- Symmetry breaking for SMT unifies several heuristics, e.g. diamonds

Outline

Conclusion

- symmetry-based techniques sensitive to obfuscation
- **users should break their symmetries themselves**, i.e. generate symmetry-free formulas
- SyMT, a tool to find out symmetries
- in the near future, SyMT will provide hints for symmetry breaking predicates
- open-source (w.i.p.). `http://www.veriT-solver.org`