Constraint Solvers for the Working PL Researcher

Nadia Polikarpova
The SAT/SMT Revolution

- hardware verification
- software verification
- software synthesis & repair

- network configuration synthesis
- biological modeling
- architecture
Boolean SATisfiability

\[(\text{gin} \lor \text{tonic}) \land (\text{minor} \Rightarrow \neg \text{gin}) \land \text{minor}\]
Boolean SATisfiability

\[(gin \lor \text{tonic}) \land (\text{minor} \Rightarrow \neg gin) \land \text{minor}\]

Solution:

\[
\begin{align*}
\text{minor} & \mapsto T \\
\text{gin} & \mapsto F \\
\text{tonic} & \mapsto T
\end{align*}
\]
Satisfiability Modulo Theories

\[(\text{gin} \lor \text{tonic}) \land (\text{age} < 21 \implies \text{abv} = 0) \land (\text{age} = 20)\]
Satisfiability Modulo Theories

\[(\text{gin } \lor \text{ tonic}) \land (\text{age} < 21 \Rightarrow \text{abv} = 0) \land (\text{age} = 20)\]

In the United States, "gin" is defined as an alcoholic beverage of no less than 40% ABV... Wikipedia
Satisfiability Modulo Theories

\[(\text{gin} \lor \text{tonic}) \land (\text{age} < 21 \Rightarrow \text{abv} = 0) \land (\text{age} = 20) \land (\text{gin} \Rightarrow \text{abv} \geq 40)\]

In the United States, "gin" is defined as an alcoholic beverage of no less than 40% ABV... Wikipedia
Satisfiability Modulo Theories

\[(\text{gin} \lor \text{tonic}) \land (\text{age} < 21 \Rightarrow \text{abv} = 0) \land (\text{age} = 20) \land (\text{gin} \Rightarrow \text{abv} \geq 40)\]

age $\mapsto$ 20
abv $\mapsto$ 0
gin $\mapsto$ F
tonic $\mapsto$ T
Satisfiability Modulo Theories

\[(\text{gin} \lor \text{tonic}) \land (\text{age} < 21 \Rightarrow \text{abv} = 0) \land (\text{age} = 20) \land (\text{gin} \Rightarrow \text{abv} \geq 40)\]

\[
\begin{align*}
\text{age} & \mapsto 20 \\
\text{abv} & \mapsto 0 \\
\text{gin} & \mapsto F \\
\text{tonic} & \mapsto T
\end{align*}
\]

theory of Linear Integer Arithmetic
Popular Solvers

Z3

Microsoft

SMT competition: http://smtcomp.sourceforge.net

CVC4

Stanford

SRI

JKU Linz, Austria

Yices2

Boollector

.smt2 // SMTLib format

(declare-fun (Int) age)
(declare-fun (Int) abv)
Plan for Today

How to use Z3 for:
1. Constraint programming
2. Program verification
3. Program synthesis
Problem: Array Partitioning

Partition an array of size $N$ evenly into $P$ sub-ranges
Problem: Array Partitioning

Partition an array of size $N$ evenly into $P$ sub-ranges

$N = 8$

$P = 4$
Problem: Array Partitioning

Partition an array of size $N$ evenly into $P$ sub-ranges

$N = 8$

$P = 4$

$sz_1$, $sz_2$, $sz_3$, $sz_4$
Problem: Array Partitioning

Partition an array of size $N$ evenly into $P$ sub-ranges

$N = 10$

$P = 4$
Problem: Array Partitioning

Partition an array of size $N$ evenly into $P$ sub-ranges

$N = 10$

$P = 4$

$sZ_1$  $sZ_2$  $sZ_3$
Problem: Array Partitioning

Partition an array of size $N$ evenly into $P$ sub-ranges

$N = 10$

$P = 4$

$sz_1$ $sz_2$ $sz_3$ $sz_4$
Problem: Array Partitioning

Partition an array of size $N$ evenly into $P$ sub-ranges

$N = 10$

$P = 4$

$SZ_1$  $SZ_2$  $SZ_3$  $SZ_4$
Problem: Array Partitioning

Partition an array of size $N$ evenly into $P$ sub-ranges

$N = 10$

$P = 4$

Can we always make them differ by at most 1?
to the rescue!
Plan for Today

How to use Z3 for:
1. Constraint programming
2. Program verification
3. Program synthesis
Plan for Today

How to use Z3 for:
1. Constraint programming
2. Program verification
3. Program synthesis
CEGIS

\[ \{N_0, N_1, \ldots, N_k\} \]

verified for all \( N! \)

wrong for \( N = N_k \)
What we have seen:

How to use Z3 for:
1. Constraint programming
2. Program verification
3. Program synthesis

You can find all the code from this talk here:

https://github.com/nadia-polikarpova/smt-talk