Model-Based Reasoning about Quantified Formulas in CVC4

Andrew Reynolds

May 23, 2013

University of Iowa

Outline

- Introduction to SMT and applications
- Model-Based approach for handling quantifiers
- How can we construct good models?
- Experimental Results

Satisfiability Modulo Theories (SMT)

- SMT solvers are powerful tools
  - Used in many formal methods applications
  - Support many background *theories*
    - Arithmetic, bitvectors, arrays, datatypes, …
  - May generate:
    - Proofs
      - Theorem proving, software/hardware verification
    - Models
      - Failing instances of aforementioned applications
      - Invariant synthesis, scheduling, test case generation

Satisfiability Modulo Theories

$$(f(a) = b \vee f(a) = c) \wedge c+1 = b \wedge \forall x.\ f(x) = g(x)$$

Satisfiability Modulo Theories

$$(\, f(a) = b \lor f(a) = c \,) \land c+1 = b \land \forall x.\ f(x) = g(x)$$

$\Downarrow$ Abstract to propositional logic

$$(\, A \lor B \,) \land C \land D$$

Satisfiability Modulo Theories

$$(f(a) = b \lor f(a) = c) \land c+1 = b \land \forall x.\ f(x) = g(x)$$

$$(A \lor B) \land C \land D$$

$\underbrace{A}_{\texttt{true}}$, $\underbrace{C}_{\texttt{true}}$, $\underbrace{D}_{\texttt{true}}$

Find satisfying assignment: A , C , D

Satisfiability Modulo Theories

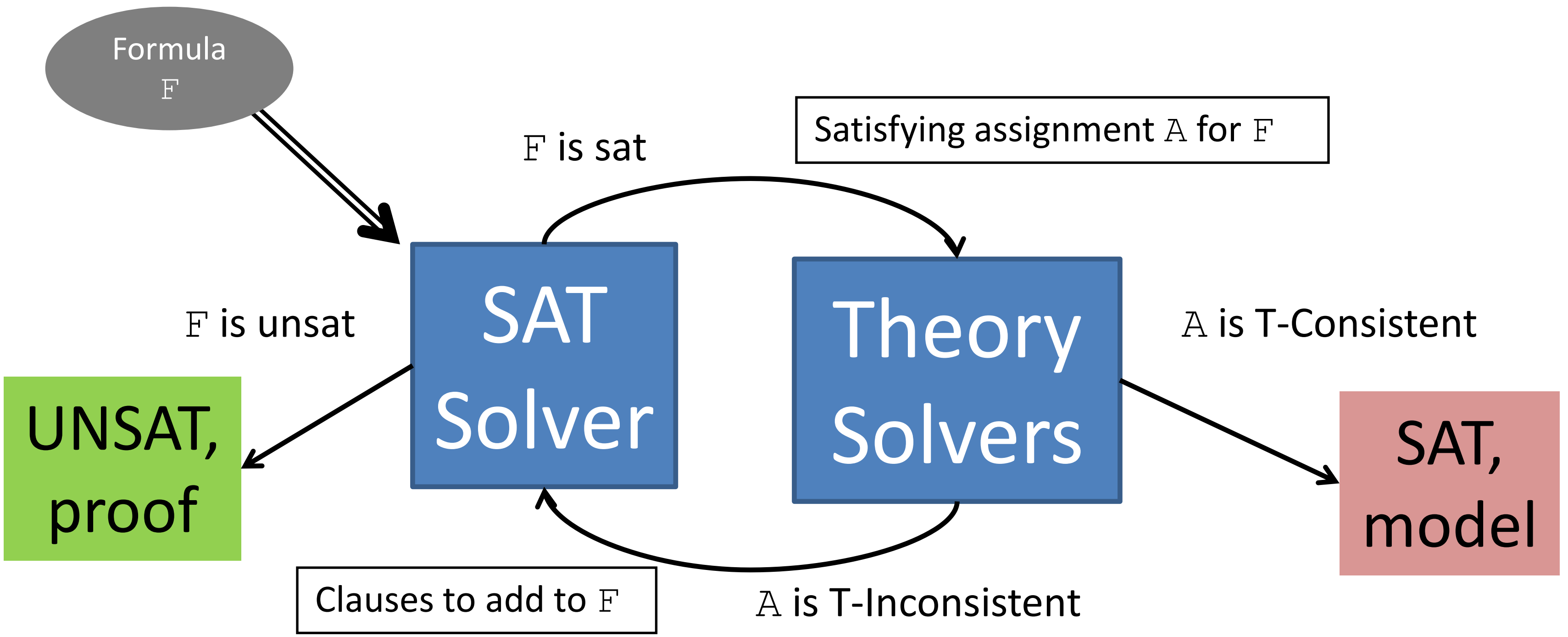
$$(f(a) = b \vee f(a) = c) \wedge c+1 = b \wedge \forall x.\ f(x) = g(x)$$

$$(\underbrace{A}_{\texttt{true}} \vee B) \wedge \underbrace{C}_{\texttt{true}} \wedge \underbrace{D}_{\texttt{true}}$$

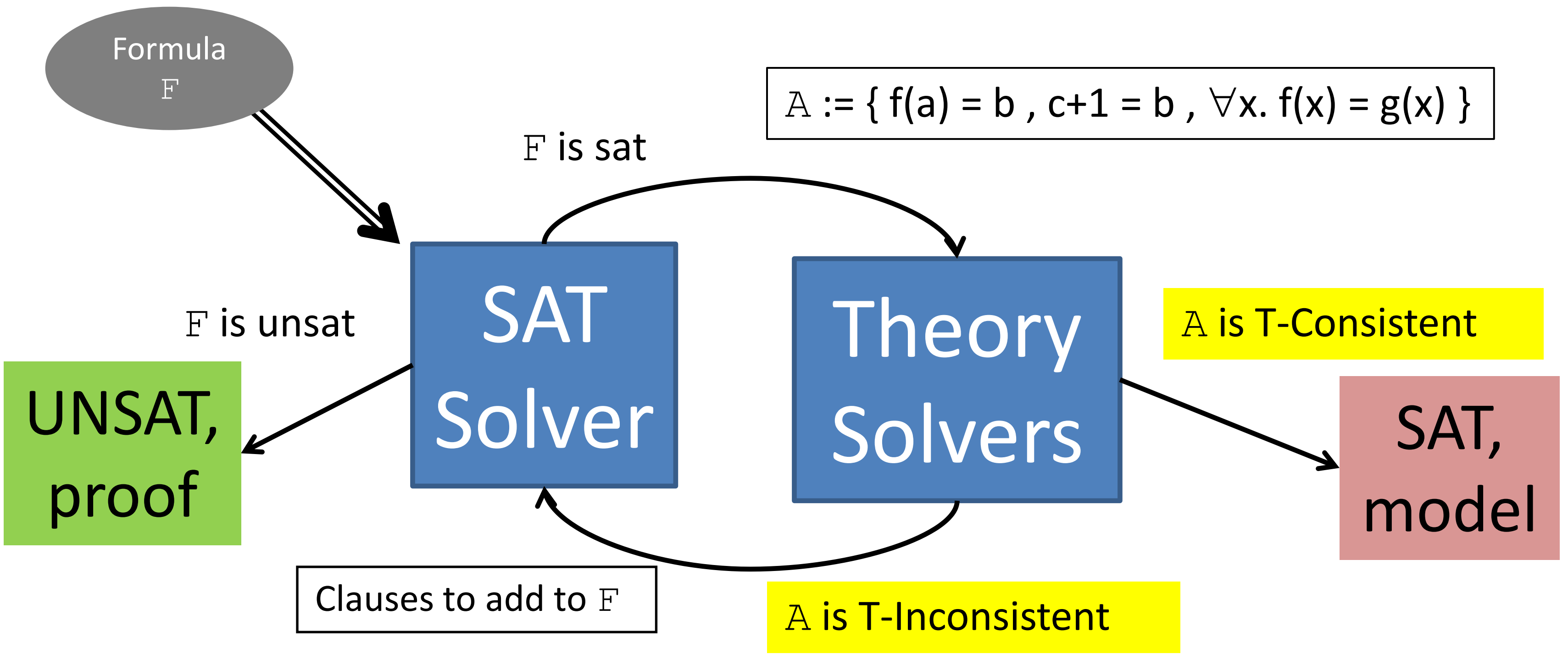
Find satisfying assignment: A , C , D

Check T-consistency: $f(a) = b$, $c+1 = b$, $\forall x.\ f(x) = g(x)$

DPLL(T) Architecture



DPLL(T) Architecture : Challenge



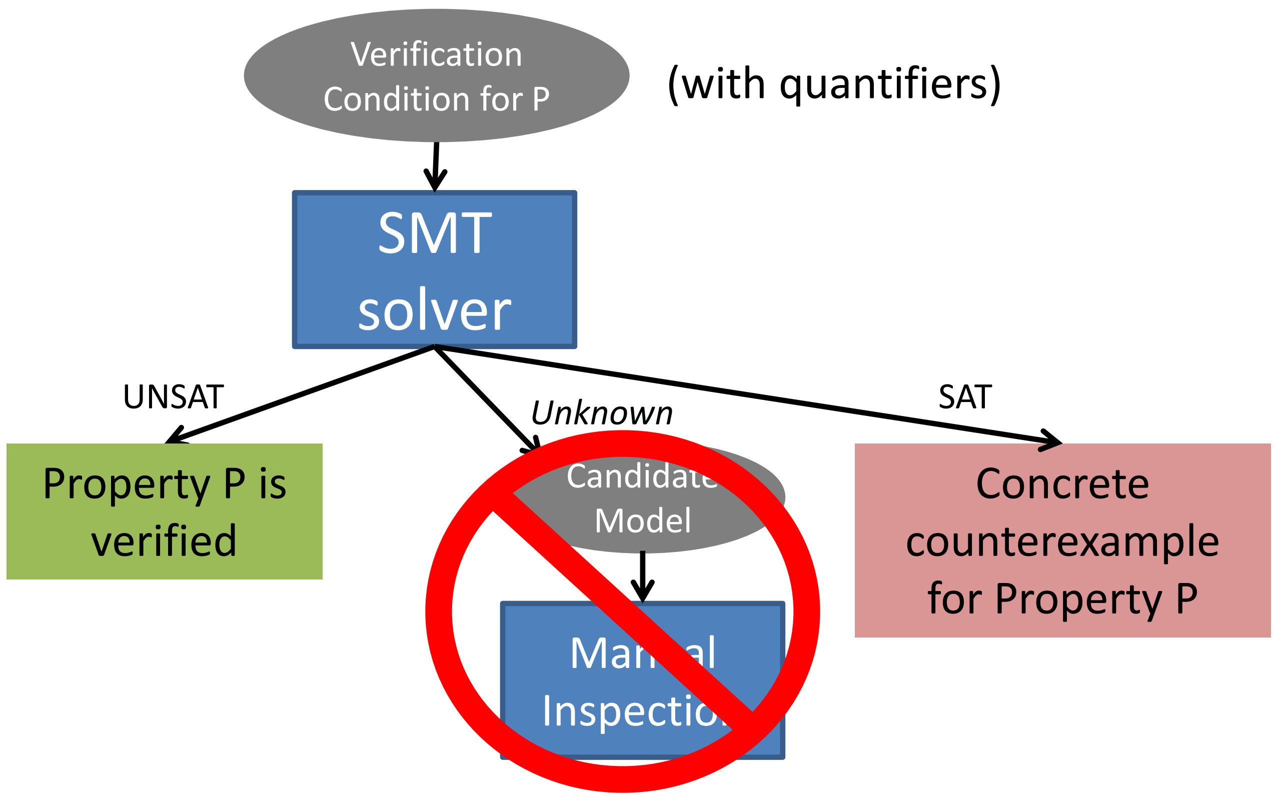
- Challenge: What if determining the consistency of A is difficult?
- For quantified formulas, determining consistency is *undecidable*

SMT with Quantified Formulas

- When quantified formulas are asserted
  - Most SMT solvers will:
    - Answer unsat, if they happen to find a proof
    - Run indefinitely
    - Give up, reporting "unknown"

Why Models are Important

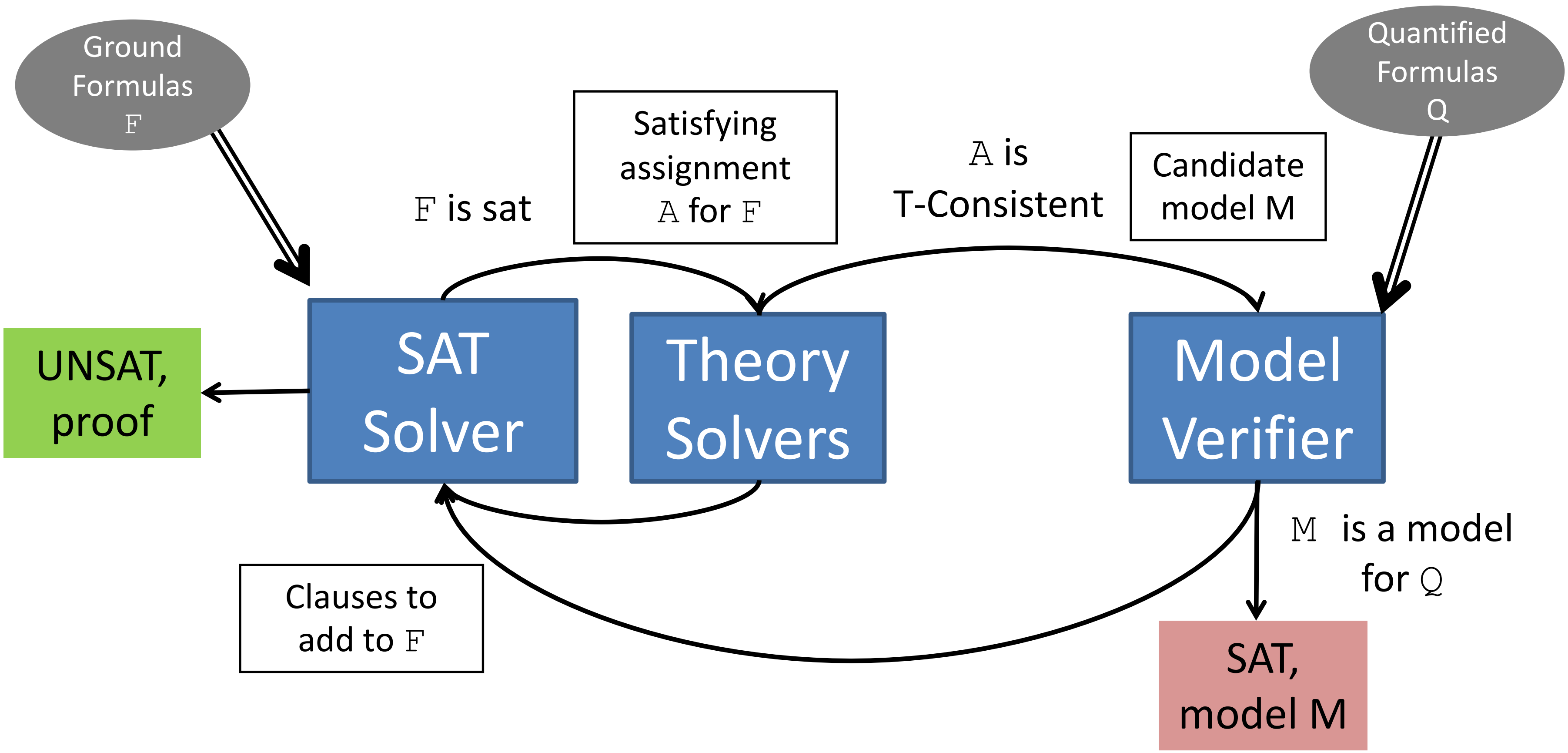
⇒ *Solver needs way of answering satisfiable when quantified formulas are asserted*



Model-Based Approach for Quantifiers

- Given:
  - Set of ground formulas $\mathtt{F}$
  - Set of quantified axioms $\mathtt{Q}$
- Determine the satisfiability of $\mathtt{F} \wedge \mathtt{Q}$
- Idea:
  - Construct candidate models for $\mathtt{Q}$ based on satisfying assignments for $\mathtt{F}$
    - Ge and deMoura [2009]

Model-Based Approach for Quantifiers



Running Example

$person_1$, $person_2$, $person_3$: Person

blue, brown, blonde : Color

eye, hair : Person -> Color

distinct(blue, brown, blonde)

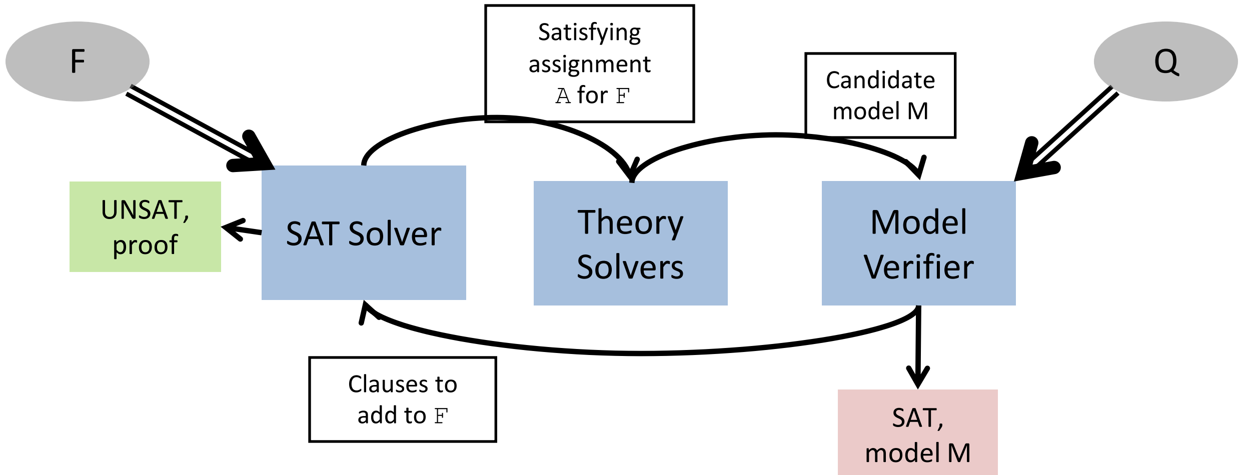
hair($person_1$) = brown

eye($person_2$) = blue

hair($person_3$) = blonde

$\forall$ x : Person. eye(x)=blue $\Rightarrow$ hair(x)=blonde

Running Example



F:

distinct(brown, blue, blonde)

$hair(person_1) = brown$

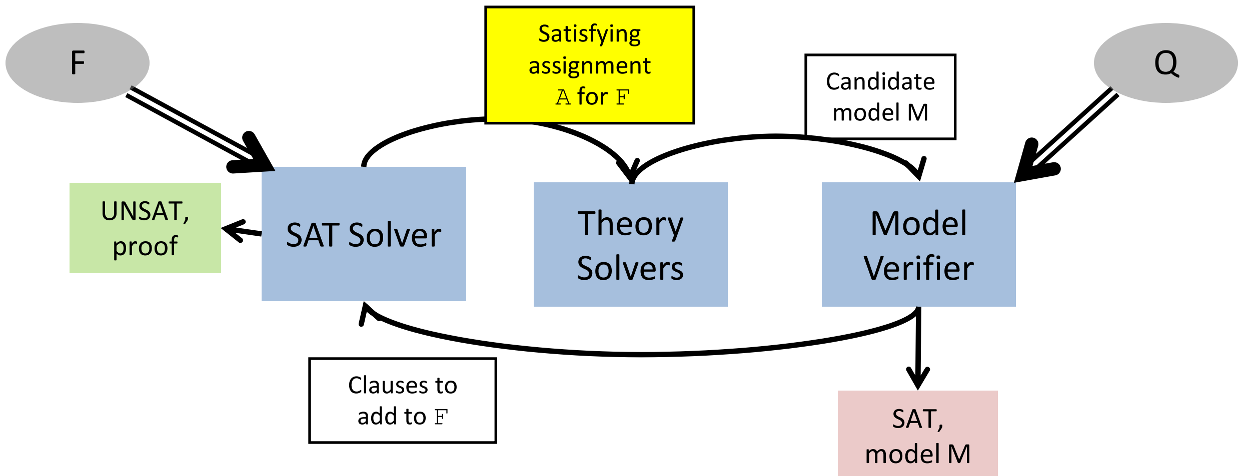
$eye(person_2) = blue$

$hair(person_3) = blonde$

Q:

$\forall$ x : Person. eye(x)=blue $\Rightarrow$ hair(x)=blonde

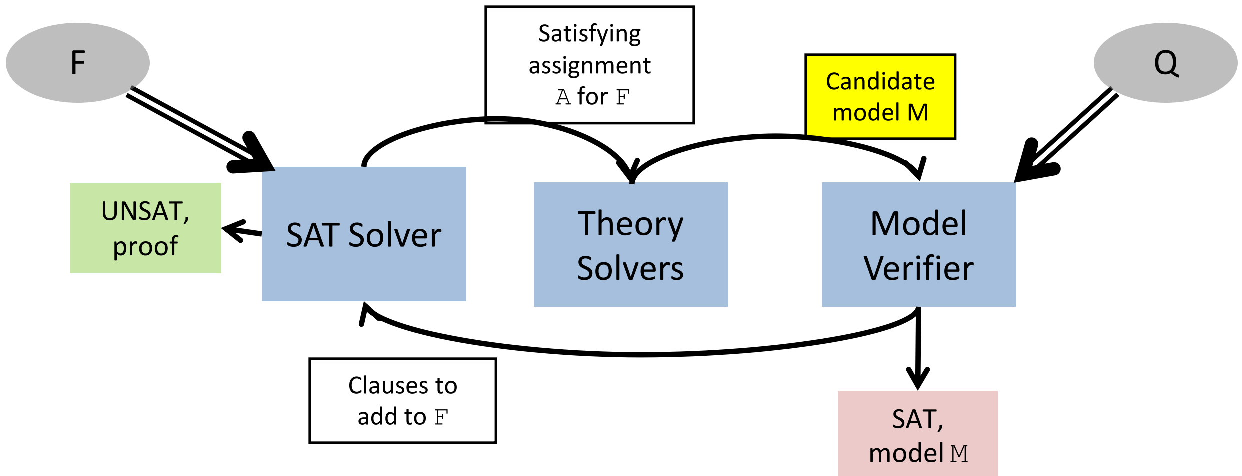
Find Satisfying Assignment A for F



true	distinct(brown, blue, blonde)
true	hair($person_1$) = brown
true	eye($person_2$) = blue
true	hair($person_3$) = blonde
	$\forall$ x : Person. eye(x)=blue $\Rightarrow$ hair(x)=blonde

- A *is Theory-Consistent according to the theory of equality*

From A, construct candidate model M



A :=

{ distinct(brown, blue, blonde),

hair($person_1$) = brown,

eye($person_2$) = blue,

hair($person_3$) = blonde }

M :=

hair : $person_1$ -> brown

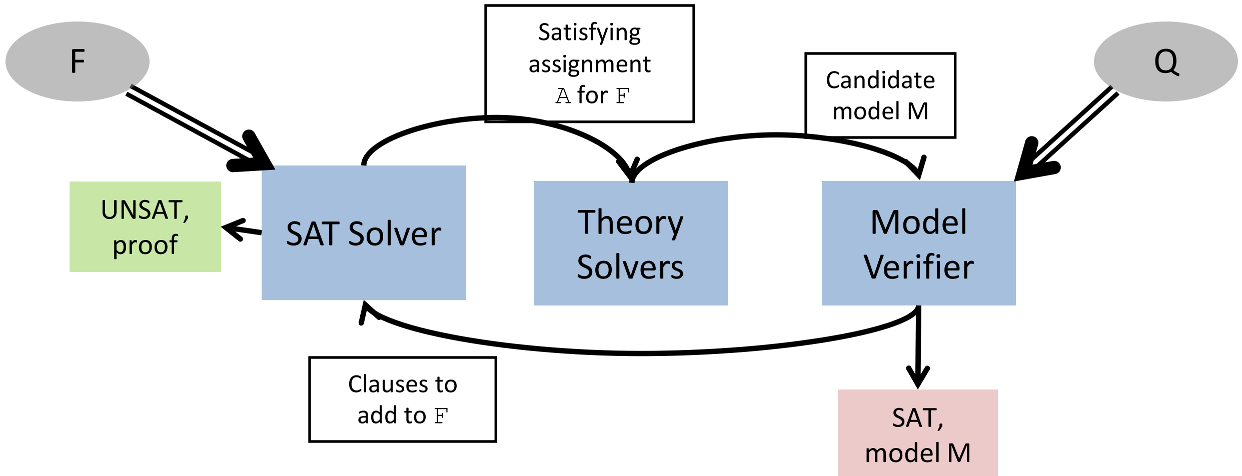
$person_3$ -> blonde

else -> brown

eye : $person_2$ -> blue

else -> blue

Check whether M is a model of Q



M :=

hair : $person_1$ -> brown

$person_3$ -> blonde

else -> brown

eye : $person_2$ -> blue

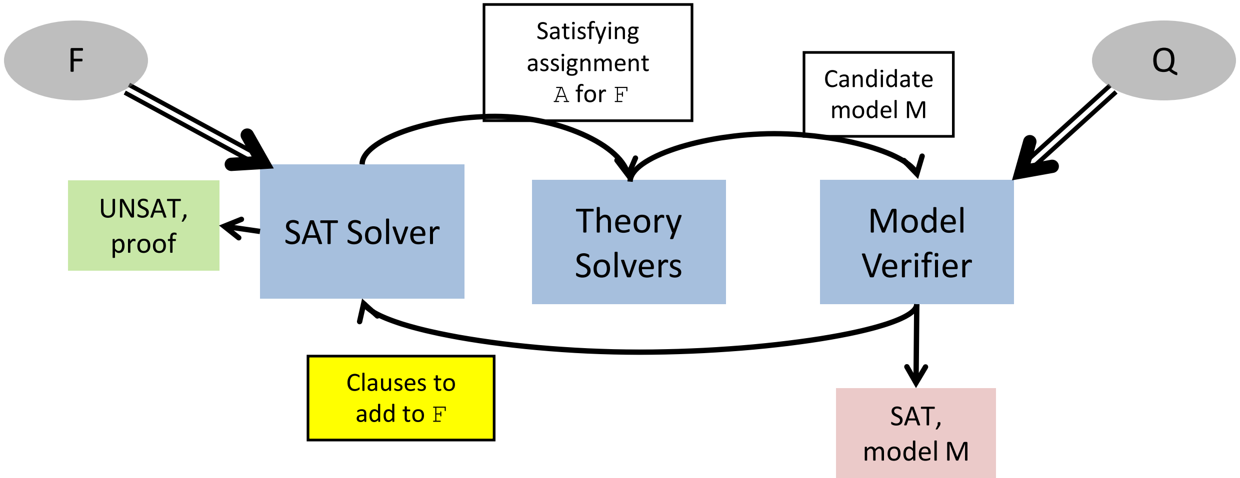
else -> blue

Q :=

$\forall$ x : Person. eye(x)=blue $\Rightarrow$ hair(x)=blonde

- *Q is false for $person_2$*

Check whether $\mathbb{M}$ is a model of $\mathbb{Q}$



$\mathbb{M}$:=

hair : $person_1$ -> brown
$person_3$ -> blonde
else -> brown
eye : $person_2$ -> blue
else -> blue

$\mathbb{Q}$:=

$\forall$ x : Person. eye(x)=blue $\Rightarrow$ hair(x)=blonde

- *$\mathbb{Q}$ is false for $person_2$*

- Add (eye($person_2$)=blue $\Rightarrow$ hair($person_2$)=blonde) to $\mathbb{F}$
- Will rule out $\mathbb{M}$ on next iteration
  - Can be thought of as model “refinement”

What are good candidate models?

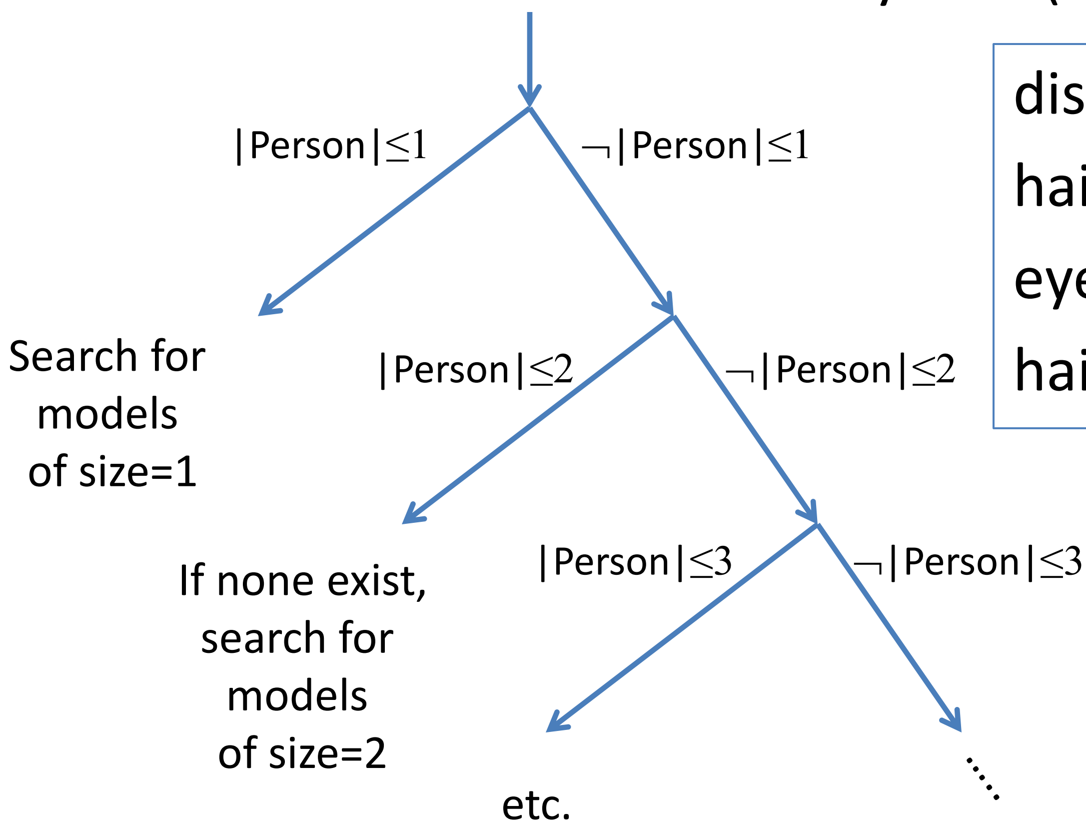
- Good candidate models
  - Have small domain sizes
  - Most instances of axioms Q are likely to be true

- For small domain sizes,
  - Use specialized theory solver within DPLL(T)
- For making most instances true,
  - Use ground solver to guide model construction

- These features are implemented in SMT solver CVC4

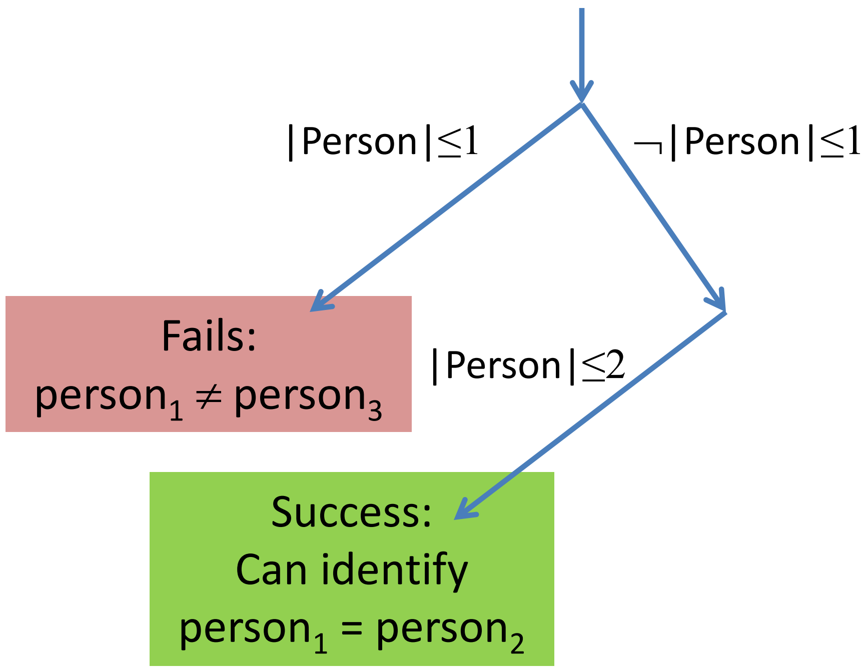
Finding Minimal Models in DPLL(T) Search

- Idea: try to fix domain sizes 1,2,3,….
  - Prioritize decisions made by DPLL(T) search



distinct(brown, blue, blonde)
hair($person_1$) = brown
eye($person_2$) = blue
hair($person_3$) = blonde

Finding Minimal Models in DPLL(T) Search

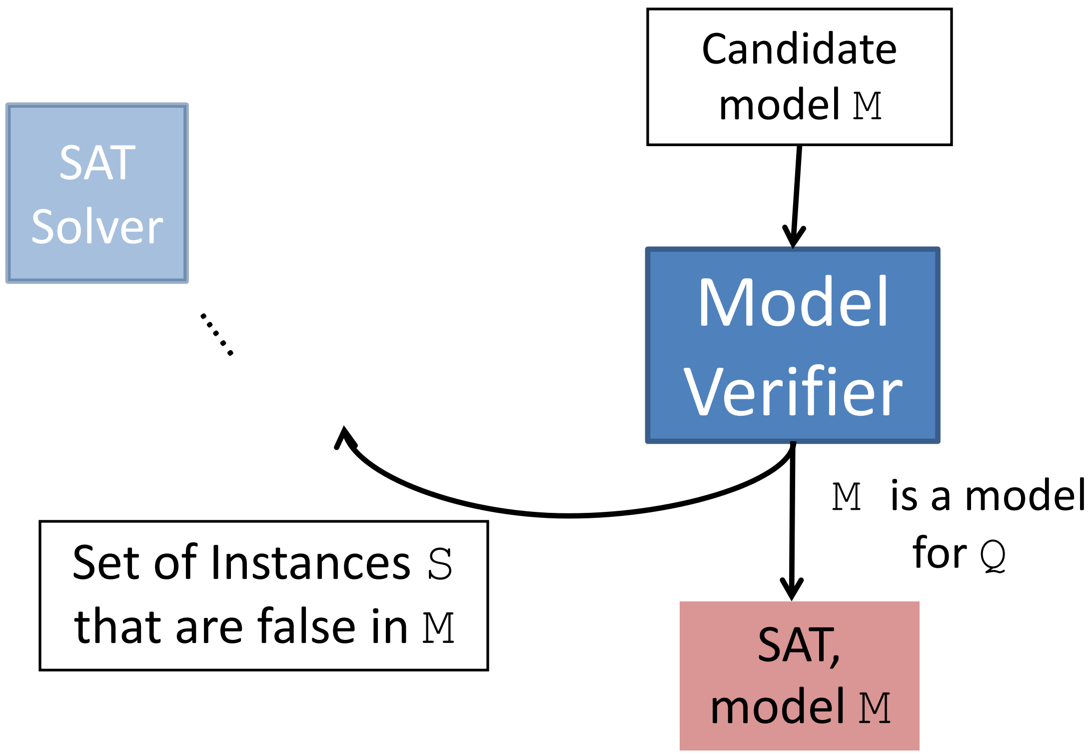


distinct(brown, blue, blonde)
$hair(person_1)$ = brown
$eye(person_2)$ = blue
$hair(person_3)$ = blonde

- Implementation in CVC4 uses:
  - Splitting on demand to shrink model sizes
  - Efficient methods for clique detection

$\Rightarrow$ *Theory of finite cardinality constraints [CAV 2013]*

Constructing Good Candidate Models



- Set S can be very large
  - For Q with n variables with domain size d, |S| can be $O(d^n)$
- Would prefer if most instances of Q are true in M

Constructing Good Candidate Models

- Idea for axiom $\mathtt{Q}$:
  - Chose default values in model $\mathbb{M}$ based on one satisfying ground instance of $\mathtt{Q}$

distinct(brown, blue, blonde)

hair($person_1$) = brown

eye($person_2$) = blue

hair($person_3$) = blonde

$\mathtt{Q}$: ∀ x : Person. eye(x)=blue ⇒ hair(x)=blonde

- See how $\mathtt{Q}$ is satisfied for one instance, then generalize this [CADE 2013]

Constructing Good Candidate Models

distinct(brown, blue, blonde)

$hair(person_1) = brown$

$eye(person_2) = blue$

$hair(person_3) = blonde$

$\mathcal{Q}$: $\forall\ x : Person.\ eye(x)=blue \Rightarrow hair(x)=blonde$

$eye(person_1)=blue \Rightarrow hair(person_1)=blonde$

- Consider $\mathcal{Q}[person_1/x]$

Constructing Good Candidate Models

true — distinct(brown, blue, blonde)

true — $hair(person_1) = brown$

true — $eye(person_2) = blue$

true — $hair(person_3) = blonde$

$\forall$ x : Person. eye(x)=blue $\Rightarrow$ hair(x)=blonde

$eye(person_1)=blue \Rightarrow hair(person_1)=blonde$

false

- Find satisfying assignment

Constructing Good Candidate Models

distinct(brown, blue, blonde)
$hair(person_1) = brown$
$eye(person_2) = blue$
$hair(person_3) = blonde$
$\forall$ x : Person. eye(x)=blue $\Rightarrow$ hair(x)=blonde

$eye(person_1)=blue \Rightarrow hair(person_1)=blonde$

- Construct candidate model

$\mathbb{A}$:=
{ distinct(brown, blue, blonde),
$hair(person_1) = brown$,
$eye(person_2) = blue$,
$hair(person_3) = blonde$,
$eye(person_1) \neq blue$ }

$\mathbb{M}$:= hair : $person_1$ -> brown
$person_3$ -> blonde
else -> …
eye : $person_1$ -> brown
$person_2$ -> blue
else -> …

Constructing Good Candidate Models

distinct(brown, blue, blonde)

$hair(person_1) = brown$

$eye(person_2) = blue$

$hair(person_3) = blonde$

$\forall\, x : Person.\ eye(x)=blue \Rightarrow hair(x)=blonde$

$eye(person_1)=blue \Rightarrow hair(person_1)=blonde$

$\mathbb{A}$:=

{ distinct(brown, blue, blonde),

$hair(person_1) = brown$,

$eye(person_2) = blue$,

$hair(person_3) = blonde$,

$eye(person_1) \neq blue$ }

$\mathbb{M}$:= hair : $person_1$ -> brown

$person_3$ -> blonde

else -> brown

eye : $person_1$ -> brown

$person_2$ -> blue

else -> brown

Model-Based Approach in CVC4

- CVC4 is state of the art SMT solver with
  - Support for many theories
- Features implemented in CVC4:
  - Theory solver for handling cardinality constraints
  - Techniques for constructing candidate models
  - Efficient methods for verifying candidate models
    - Not mentioned in this talk

Experiments

- DVF Benchmarks
  - Taken from verification tool DVF used by Intel
  - Both SAT/UNSAT benchmarks
    - SAT benchmarks generated by removing necessary pf assumptions
  - Many theories: UF, arithmetic, arrays, datatypes
  - Quantifiers only over free sorts
    - Memory addresses, Values, Sets, …
- TPTP Benchmarks
  - Unsorted, equality, function symbols
  - Heavy use of quantifiers

Experiments: DVF

SAT	german	refcount	agree	apg	bmk	Total
#	45	6	42	19	37	149
cvc3	0	0	0	0	0	0
yices	2	0	0	0	0	2
z3	**45**	1	0	0	0	46
cvc4	2	0	0	0	0	2
cvc4+f	**45**	**6**	**42**	**19**	**37**	**149**

UNSAT	german	refcount	agree	apg	bmk	Total
#	145	40	488	304	244	1221
cvc3	**145**	**40**	457	267	229	1138
yices	**145**	**40**	**488**	**304**	**244**	**1221**
z3	**145**	**40**	**488**	**304**	**244**	**1221**
cvc4	**145**	**40**	484	**304**	**244**	1217
cvc4+f	**145**	**40**	471	300	242	1198

- Configurations :
  - cvc4 : heuristic inst.
  - cvc4+f : model-based

- cvc4+f effective for sat
- cvc4+f solves 4 unsat that cvc4 cannot

Experiments: TPTP

- For 1995 satisfiable benchmarks:
  - Paradox solves 1305
  - iProver solves 1231
  - z3 solves 887
  - cvc4+f solves 1186
    - Includes 3 problems with rating 1.0
- For 12568 unsatisfiable benchmarks:
  - z3 solves 5934
  - iProver solves 5556
  - cvc4 solves 5415
  - cvc4+f solves 3028
    - Orthogonal to other approaches
    - 282 cannot be solved by z3

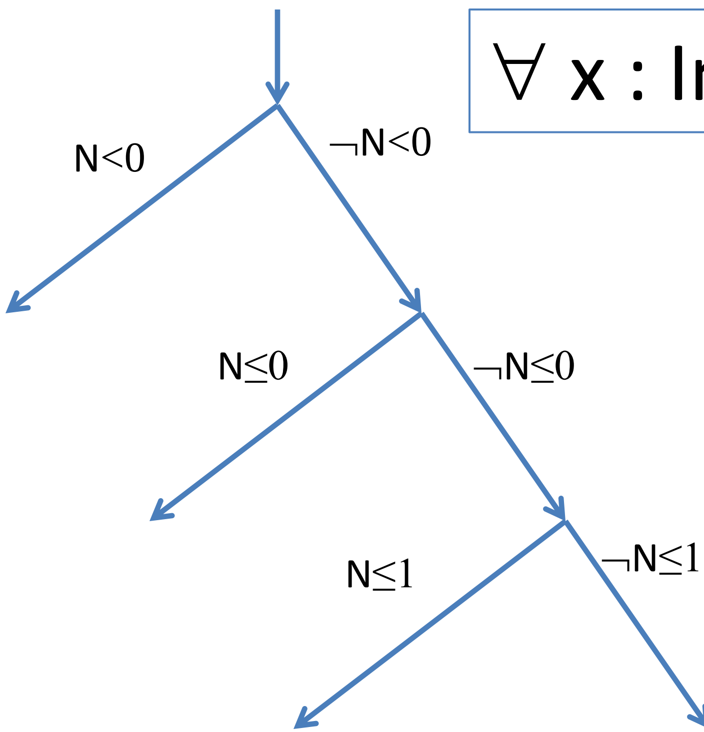
Summary

- Completed work in CVC4:
  - Ground solver for finding small models
  - Methods for constructing and verifying candidate models
- Publicly available : http://cvc4.cs.nyu.edu/
- Current work:
  - Fair strategies for minimizing models for multiple sorts
  - Improve existing approaches for answering UNSAT
  - Other applications
    - Theory of Strings : bounded length
    - Integer quantification within bounded ranges

Current Work

- Extension to bounded integer quantification
  - Can use similar approach

$$\forall\, x : \text{Int.}\ 0 \le x \le N \Rightarrow P(x)$$



Thanks

- Collaborators:
  - Cesare Tinelli, Amit Goel, Sava Krstic, Clark Barrett, Morgan Deters, Leonardo de Moura
- Questions?