Proving Infinite Satisfiability

Peter Baumgartner
Joshua Bax

NICTA
Australian National University

Goal

Theorem Proving in Hierarchic Combinations of Specifications

Dis

Background theory
linear integer arithmetic

Data structures
list axioms/arrays axioms

Definitions
length/append/isSorted

$\overset{?}{\not\models}$

This work

Conjecture
specific query

"No refutation" does not mean "not entailed"

Approaches

First-order proving modulo theories **- incomplete**
SMT **- incomplete**
(Many specialised procedures in particular for arrays)

Example

Linear integer arithmetic (LIA)

Lists over integers

$(l \approx \mathrm{nil}) \lor (l \approx \mathrm{cons}(\mathrm{head}(l), \mathrm{tail}(l)))$

$\neg(\mathrm{cons}(k, l) \approx \mathrm{nil})$

$\mathrm{head}(\mathrm{cons}(k, l)) \approx k$

$\mathrm{tail}(\mathrm{cons}(k, l)) \approx l$

The inRange predicate

$\mathrm{inRange}(l, n) \leftrightarrow (l \approx \mathrm{nil} \lor (0 \leq \mathrm{head}(l) < n \land \mathrm{inRange}(\mathrm{tail}(l), n)))$

$\models \mathrm{inRange}([1,0,5], 6)$

$\not\models \mathrm{inRange}([1,0,5], 5)$

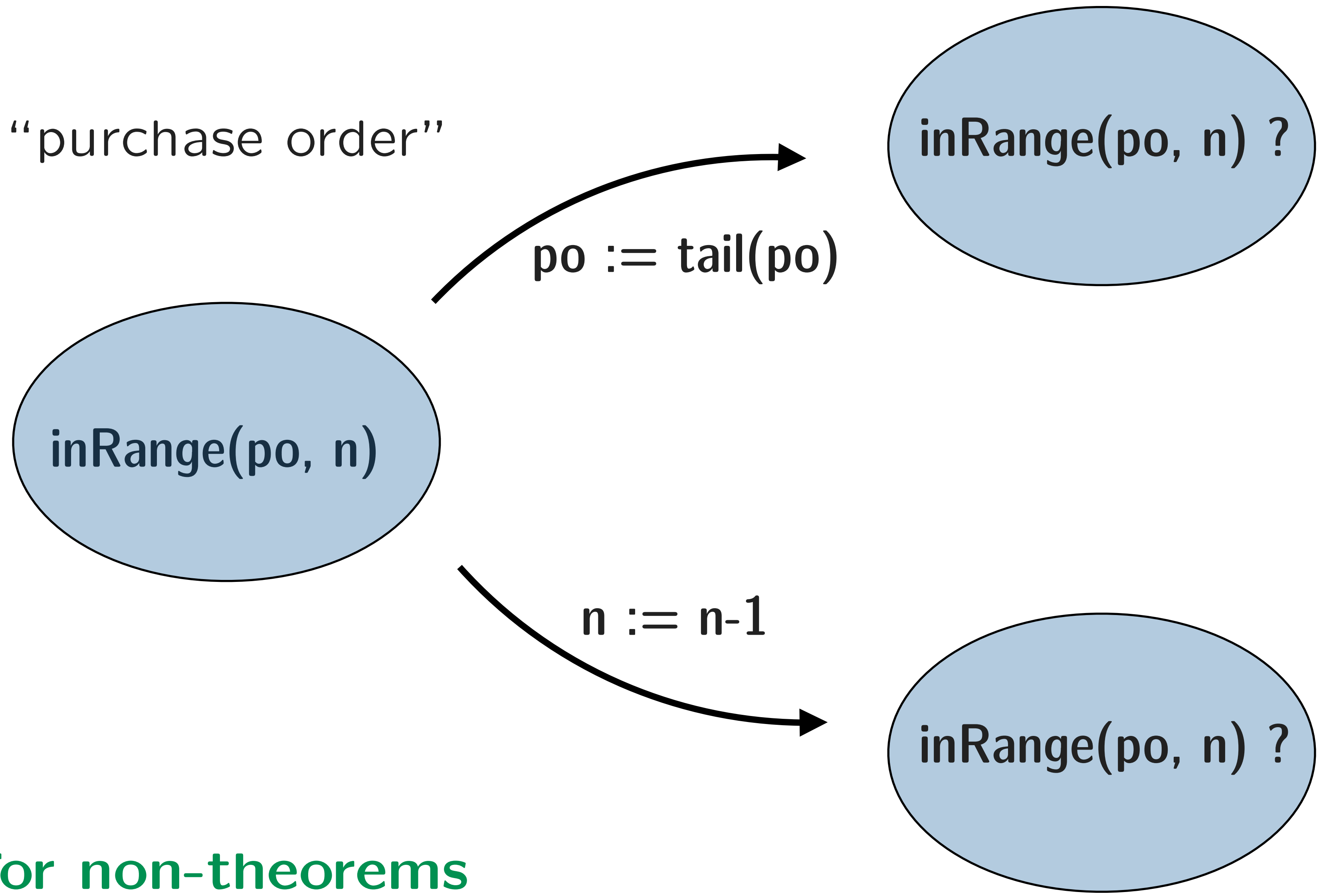
$\not\models \mathrm{inRange}(l, n) \rightarrow \mathrm{inRange}(l, n\text{-}1)$

Not directly refutable by Z3, Beagle
Easy with our method

Example in Context [TABLEAUX 2014]

Analysis of dynamical systems

po means "purchase order"



Source for non-theorems

- Bugs
- Task is reachability (planning)
- Partial-order reduction analysis: many simple ones

Our Approach

“Disproving by proving”

The goal is to establish $Ax \cup Def \not\models Con$

(1) **Suppose Ax is satisfiable (wrt hierarchic interpretations)**
This needs to be shown once and for all

(2) **Make sure $Ax \cup Def$ is satisfiable**
We provide a template language for Def’s for that

(3) **Prove $Ax \cup Def \models \neg Con$ by a theorem prover/SMT solver**
It follows $Ax \cup Def \not\models Con$ as desired

Proof:
By (2) there is an interpretation I such that $I \models Ax \cup Def$
With (3) conclude $I \models \neg Con$, hence $I \not\models Con$
Together $Ax \cup Def \not\models Con$ $\square$

Rest of this talk: (1) - (3) for lists and for arrays

(1) Suppose Ax is satisfiable (Lists)

Satisfiability of list axioms can be shown automatically

$(l \approx \mathrm{nil}) \vee (l \approx \mathrm{cons}(\mathrm{head}(l), \mathrm{tail}(l)))$

$\neg(\mathrm{cons}(k, l) \approx \mathrm{nil})$

$\mathrm{head}(\mathrm{cons}(k, l)) \approx k$

$\mathrm{tail}(\mathrm{cons}(k, l)) \approx l$

$\exists d\,.\, \mathrm{head}(\mathrm{nil}) \approx d$ // required for sufficient completeness

$\mathrm{tail}(\mathrm{nil}) \approx \mathrm{nil}$ // required for sufficient completeness

Hierarchic superposition terminates with a finite saturation

Together with sufficient completeness this entails satisfiability

(1) Suppose Ax is satisfiable (Arrays)

Satisfiability of array axioms can be shown automatically

$\mathrm{read}(\mathrm{write}(a, i, x), i) \approx x$
$\mathrm{read}(\mathrm{write}(a, i, x), j) \approx \mathrm{read}(a, j) \vee i \approx j$

$\mathrm{read}(a, i) \not\approx \mathrm{read}(b, i) \vee a \approx b$ // Extensional equality

$\mathrm{read}(\mathrm{init}(x), i) \approx x$ // Constant arrays

Hierarchic superposition terminates with a finite saturation
Together with sufficient completeness this entails satisfiability

(2) Make sure *Ax* ∪ *Def* is satisfiable - general

Let Σ be a signature (e.g. Σ_{LIST})

Def [admissible definition]

Given:

- *op*, a new operator not in Σ (e.g. **length**)
- *Def*(*op*), a set of $\Sigma \cup \{op\}$ -sentences (e.g. **length** def)

Def(*op*) is *admissible* iff every Σ-interpretation I with domain D can be extended to a $\Sigma \cup \{op\}$-interpretation J with domain D such that $J \models Def(op)$

Justifies stepwise extensions of Ax in a stratified way

- Assume *Ax* is satisfiable, by (1)
- Build stepwise extension $Ax \cup \{Def(op_1), \ldots, Def(op_n)\}$ with admissible definitions
- It follows $Ax \cup \{Def(op_1), \ldots, Def(op_n)\}$ is satisfiable

Example: Extend lists by **length**, **count**, **inRange**, **append**, ...

(2) Make sure *Ax* ∪ *Def* is satisfiable - list relations

Given $\Sigma^+ \supseteq \Sigma_{\mathrm{LIST}}$, domain D = LIST, new pred symbol $P \notin \Sigma^+$

Template for admissible definition *Def*(*P*)

$$\forall\, k_{\mathbb{Z}}\ l_{\mathrm{LIST}}\ .\ P(k,l) \leftrightarrow$$

$$l \approx \mathrm{nil} \wedge B[k] \qquad \text{(Base case nil)}$$

$$\vee\ \exists\, \mathrm{h}_{\mathbb{Z}}\ \mathrm{t}_{\mathrm{LIST}}\ .\ \mathrm{l} \approx \mathrm{cons}(h,\ t) \wedge C[k,h,t] \qquad \text{(Base case cons)}$$

$$\vee\ \exists\, \mathrm{h}_{\mathbb{Z}}\ \mathrm{t}_{\mathrm{LIST}}\ .\ \mathrm{l} \approx \mathrm{cons}(h,\ t) \wedge D[k,h,t] \wedge P(k,t) \qquad \text{(Recursion case)}$$

where B, C and D are Σ^+-formulas of the proper arities

Example: *Def*(**inRange**)

Proposition: templates *Def*(*P*) provide admissible definitions

Proof sketch: by induction on LIST define least model J of *Def*(*P*) in the ← direction bottom-up

Because J is the least model it also satisfies the → direction □

(3) Prove $Ax \cup Def \models \neg Con$

List examples

$$\mathsf{inRange}(n,l) \Leftrightarrow l \approx \mathsf{nil} \vee \exists\, h_{\mathbb{Z}}\ t_{\mathsf{LIST}} \,.\, (l \approx \mathsf{cons}(h,t) \wedge 0 \leq h \wedge h < n \wedge \mathsf{inRange}(n,t))$$

Problem	Beagle	Spass+T	Z3
$\mathsf{inRange}(4, \mathsf{cons}(1, \mathsf{cons}(5, \mathsf{cons}(2, \mathsf{nil}))))$	6.2	0.3	0.2
$n > 4 \Rightarrow \mathsf{inRange}(n, \mathsf{cons}(1, \mathsf{cons}(5, \mathsf{cons}(2, \mathsf{nil}))))$	7.2	0.3	0.2
$\mathsf{inRange}(n, \mathsf{tail}(l)) \Rightarrow \mathsf{inRange}(n,l)$	3.9	0.3	0.2
$\exists\, n_{\mathbb{Z}}\ l_{\mathsf{LIST}} \,.\, l \not\approx \mathsf{nil} \wedge \mathsf{inRange}(n,l) \wedge n - \mathsf{head}(l) < 1$	2.7	0.3	0.2
$\mathsf{inRange}(n,l) \Rightarrow \mathsf{inRange}(n-1,l)$	8.2	0.3	>60
$l \not\approx \mathsf{nil} \wedge \mathsf{inRange}(n,l) \Rightarrow n - \mathsf{head}(l) > 2$	2.8	0.3	0.2
$n > 0 \wedge \mathsf{inRange}(n,l) \wedge l' = \mathsf{cons}(n-2,l) \Rightarrow \mathsf{inRange}(n,l')$	4.5	5.2	0.2

(2) Make sure $Ax \cup Def$ is satisfiable - list functions

Given $\Sigma^+ \supseteq \Sigma_{\mathrm{LIST}}$, domain D = LIST, new fun symbol $f \notin \Sigma^+$

Template for admissible definition $Def(f)$

$f(k, \mathrm{nil}) \approx b[k] \leftarrow B[k]$ (Base case)

$f(k, \mathrm{cons}(h, t)) \approx c_1[k, h, t, f(k, t)] \leftarrow C_1[k, h, t, f(k, t)]$ (Recursion case 1)

...

$f(k, \mathrm{cons}(h, t)) \approx c_n[k, h, t, f(k, t)] \leftarrow C_n[k, h, t, f(k, t)]$ (Recursion case n)

where B, C_i are Σ^+-formulas and c_i is a Σ^+-term of the proper arities

Proposition: templates $Def(f)$ provide admissible definitions
if all recursion cases are consistent (which is a theorem proving task)

(3) Prove $Ax \cup Def \models \neg Con$

List examples

$$\mathsf{length}(\mathsf{nil}) \approx 0$$
$$\mathsf{length}(\mathsf{cons}(h,t)) \approx 1 + \mathsf{length}(t)$$
$$\mathsf{count}(k,\mathsf{nil}) \approx 0$$
$$\mathsf{count}(k,\mathsf{cons}(h,t)) \approx \mathsf{count}(k,t) \Leftarrow k \not\approx h$$
$$\mathsf{count}(k,\mathsf{cons}(h,t)) \approx \mathsf{count}(k,t) + 1 \Leftarrow k \approx h$$

$$\mathsf{append}(\mathsf{nil},l) \approx l$$
$$\mathsf{append}(\mathsf{cons}(h,t),l) \approx \mathsf{cons}(h,\mathsf{append}(t,l))$$
$$\mathsf{in}(k,l) \Leftrightarrow \mathsf{count}(k,l) > 0$$

Problem	Beagle	Spass+T	Z3
$\mathsf{length}(l_1) \approx \mathsf{length}(l_2) \Rightarrow l_1 \approx l_2$	4.3	9.0	0.2
$n \geq 3 \wedge \mathsf{length}(l) \geq 4 \Rightarrow \mathsf{inRange}(n,l)$	5.4	1.1	0.2
$\mathsf{count}(n,l) \approx \mathsf{count}(n,\mathsf{cons}(1,l))$	2.5	0.3	>60
$\mathsf{count}(n,l) \geq \mathsf{length}(l)$	2.7	0.3	>60
$l_1 \not\approx l_2 \Rightarrow \mathsf{count}(n,l_1) \not\approx \mathsf{count}(n,l_2)$	2.4	0.8	>60
$\mathsf{length}(\mathsf{append}(l_1,l_2)) \approx \mathsf{length}(l_1)$	2.1	0.3	0.2
$\mathsf{length}(l_1) > 1 \wedge \mathsf{length}(l_2) > 1 \Rightarrow \mathsf{length}(\mathsf{append}(l_1,l_2)) > 4$	37	>60	>60
$\mathsf{in}(n_1,l_1) \wedge \neg\mathsf{in}(n_2,l_2) \wedge l_3 \approx \mathsf{append}(l_1,\mathsf{cons}(n_2,l_2)) \Rightarrow \mathsf{count}(n,l_3) \approx \mathsf{count}(n,l_1)$	>60 (6.2)	9.1	>60

(2) Make sure *Ax* ∪ *Def* is satisfiable - array relations

Given $\Sigma^+ \supseteq \Sigma_{\text{ARRAY}}$, domain D = ARRAY, new operators $f, P \notin \Sigma^+$

Template for admissible definition *Def*(*P*)

$\forall\, k_{\mathbb{Z}}\ a_{ARRAY}\ .\ P(a,k) \Leftrightarrow C[a,k]$

where C is a Σ^+-formula of the proper arity

Template for admissible definition *Def*(*f*)

$f(a, k) \approx y \leftarrow C_1[a, k, y]$ (Case 1)

...

$f(a, k) \approx y \leftarrow C_n[a, k, y]$ (Case n)

where C_i is a Σ^+-formula of the proper arities

As with lists one has to establish that the cases are consistent

(3) Prove *Ax* ∪ *Def* ⊨ ¬*Con*

Array examples

$$\mathsf{rev}(a,n) \approx b \Leftarrow \forall\, i_{\mathbb{Z}}\ .\, 0 \leq i \wedge i < n \wedge \mathsf{read}(b,i) \approx \mathsf{read}(a, n-(i+1)) \\ \vee\ ((0 > i \vee i \geq n) \wedge \mathsf{read}(b,i) \approx \mathsf{read}(a,i))$$

$$\mathsf{inRange}(a,r,n) \Leftrightarrow \\ \forall\, i\,.\,(n \geq i \wedge i \geq 0) \\ \Rightarrow (r \geq \mathsf{read}(a,i) \wedge \mathsf{read}(a,i) \geq 0)$$

$$\mathsf{distinct}(a,n) \Leftrightarrow \\ \forall\, i,j\,.\,(n > i \wedge n > j \wedge j \geq 0 \wedge i \geq 0) \\ \Rightarrow \mathsf{read}(a,i) \approx \mathsf{read}(a,j) \Rightarrow i \approx j)$$

$$\mathsf{max}(a,n) \approx w \Leftarrow \forall\, i\,.\,(n > i \wedge i \geq 0) \Rightarrow w \geq \mathsf{read}(a,i)) \wedge (\exists\, i\,.\,n > i \wedge i \geq 0 \wedge \mathsf{read}(a,i) \approx w)$$

Problem	Beagle	Spass+T	Z3
$n \geq 0 \Rightarrow \mathsf{inRange}(a, \mathsf{max}(a,n), n)$	1.40	0.16	u
$\mathsf{distinct}(\mathsf{init}(n), i)$	0.98	0.15	u
$\mathsf{read}(\mathsf{rev}(a, n+1), 0) = \mathsf{read}(a,n))$	>60	>60(0.27)	>60
$\mathsf{sorted}(a,n) \Rightarrow \neg\mathsf{sorted}(\mathsf{rev}(a,n), n)$	>60	0.11	0.36
$\exists\, n_{\mathbb{Z}}\,.\,\neg\mathsf{sorted}(\mathsf{rev}(\mathsf{init}(n), m), m)$	>60	0.16	u
$\mathsf{sorted}(a,n) \wedge n > 0 \Rightarrow \mathsf{distinct}(a,n)$	2.40	0.17	0.01

Conclusions

Experiments

Run with same prover settings

Include all definitions, even not needed ones

Works well *on the examples shown*

Cannot disprove $\exists\, \mathbf{n}_{\mathbb{Z}}\ \forall\, \mathbf{l}_{\mathsf{LIST}}\ \mathbf{length(cons(n, l))} \approx \mathbf{0}$

Finite model finders

Cannot use finite model finders, LIST has only infinite models

(Injective functions that are not surjective do not admit finite domains)

Satisfiability task

Same thing: to show that $Ax \cup Def \cup \{ \mathrm{F} \}$ is satisfiable

it suffices to prove $Ax \cup Def \models \mathrm{F}$

Future work

Implement method in full, integrate into model checker