A Decidable Fragment in Separation Logic with Inductive Predicates and Arithmetic

Quang Loc Le (SUTD-NUS) Makoto Tatsuta (NII)
Jun Sun (SUTD) Wei-Ngan Chin (NUS)

First NII Programming and Logic Workshop - Tokyo, Japan

March. 02, 2017

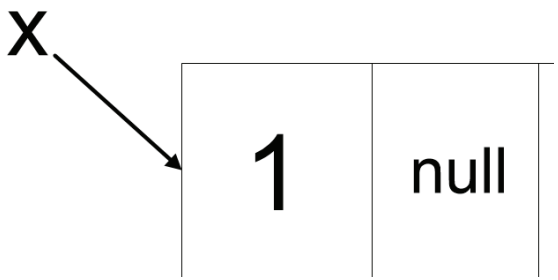
Syntax

empty heap predicate

$$\texttt{emp} \wedge x{=}\texttt{null} \wedge n{=}0$$

Syntax

Points-to predicate

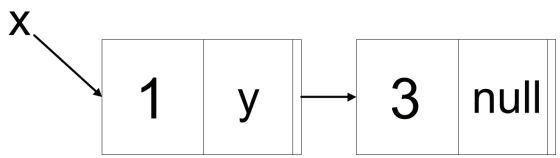


`struct node{int val; struct node * next;}`

$x \mapsto node(1, \texttt{null})$

Syntax

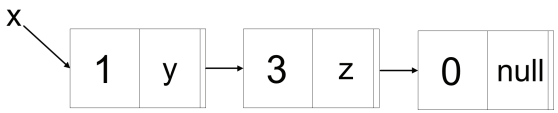
separating predicates



$$x \mapsto node(1,y) * y \mapsto node(3,\texttt{null})$$

Syntax

inductive predicate

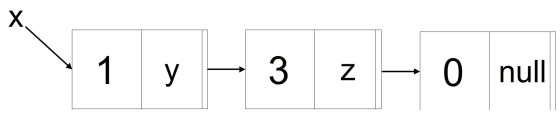


Singly-linked list

$$\texttt{pred}\ \mathit{ll}(\texttt{root}) \equiv \texttt{emp} \wedge \texttt{root} = \texttt{null}$$
$$\vee\ \exists\, r \cdot\ \texttt{root} \mapsto \mathit{node}(\_, r) * \mathit{ll}(r)$$

Syntax

inductive predicate



Singly-linked list with size property

$$\texttt{pred}\ \mathit{ll\_size}(\texttt{root},n) \equiv \texttt{emp}\wedge\texttt{root}=\texttt{null}\wedge n=0$$
$$\vee\ \exists\, r\cdot\ \texttt{root}\mapsto \mathit{node}(\_,r) * \mathit{ll\_size}(r,n-1)$$

$$\mathit{ll\_size}(x,i) \wedge i=3$$

A fragment of Separation Logic

$$
\begin{array}{lll}
\text{Formula} & \Phi & ::= \Delta \mid \Phi_1 \vee \Phi_2 \qquad\qquad \Delta ::= \exists \bar{v} \cdot (\kappa \wedge \pi) \\
\text{Spatial formula} & \kappa & ::= \mathtt{emp} \mid x \mapsto c(v_i) \mid \mathtt{P}(\bar{v}) \mid \kappa_1 * \kappa_2 \\
\text{Pure formula} & \pi & ::= \pi_1 \wedge \pi_2 \mid b \mid \alpha \mid \phi
\end{array}
$$

- b: Boolean formula
- α: Pointer (Dis)Equalities
- ϕ: Presburger arithmetic

Satisfiability Problem

- Input: A formula Δ in the fragment
- Question: Is Δ satisfiable?

Challenging

- Unbounded heaps
- Infinite numerical domain

Satisfiability Problem

Tatsuta *et. al.* (APLAS:2016): the satisfiability problem is undecidable.

What is decidable?

Related Work

- Berdine *et. al.* (FSTTCS:2004): Decidability of fragment with list predicates
  - idea: a list is equi-satisfiable with the lists of lengths zero and two.
  - ex: $ll(x)$ equi-satisfiable with $\texttt{emp} \wedge x = \texttt{null}$ and $\exists q \cdot x \mapsto node(\_, q) * q \mapsto node(\_, \texttt{null})$
- Brotherston *et. al.* (LICS:2014): Decidability of fragment with heap-only inductive predicates.
  - idea: an arbitrary heap-only inductive predicate is equi-satisfiable with a finite (disjunctive) set of base pairs
  - ex: $ll(x)$ equi-satisfiable with $\{(\emptyset, \{x = \texttt{null}\}), (\{x\}, \{x \neq \texttt{null}\})\}$

there exists a finite representation that is equi-satisfiable to every shape-based predicate

Related Work

- Tatsuta *et. al.* (APLAS:2016): Decidability of fragment where spatial part of an inductive predicate can be computed as eventually periodic sets.

The proposal

- idea:
  - Compute for each inductive predicate a finite representation that precisely characterises its satisfiability.
  - The deciability of the problem is reduced into the decidability of the spatial part, e.g. eventually periodic sets
- ex: $ll\_sizeN(x, n)$ is equi-satisfiable with:

$$ll\_sizeN(n) \equiv n{=}0 \vee \exists\, n_1 \cdot ll\_sizeN(n_1) \wedge n{=}n_1{+}1$$

Decidable Fragment

- base fragment with empty heap ($\mathtt{emp}$), points-to ($\mapsto$), spatial conjunction ($*$) and Presburger Arithmetic

$$\begin{array}{ll} \mathtt{SAT} & \Delta_1 \equiv \mathtt{emp} \wedge x = \mathtt{null} \wedge n = 0 \\ \mathtt{UNSAT} & \Delta_2 \equiv x \mapsto \mathit{node}(n, y) * y \mapsto \mathit{node}(n-1, \mathtt{null}) \wedge x = y \end{array}$$

This base fragment is decidable (Cristiano *et. al.* 2001)

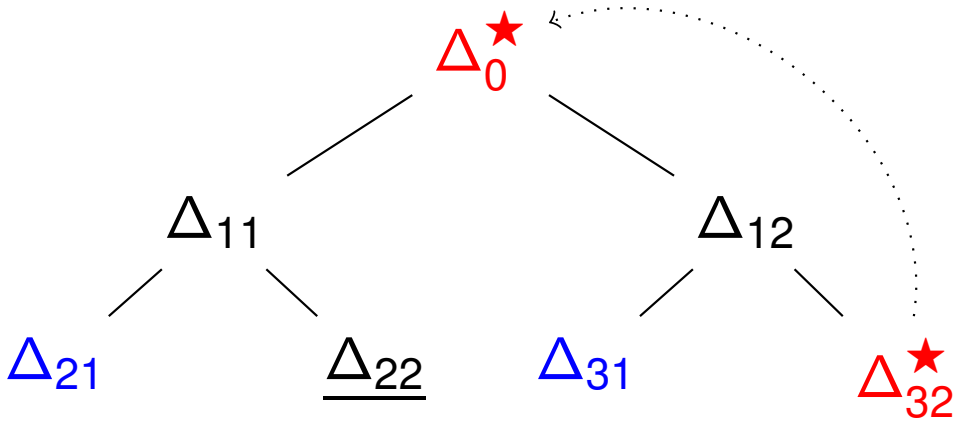
- we show that existentially quantified variables are not externally visible wrt. the satisfiability problem
  $\Delta_1' \equiv \exists r \cdot \mathit{ll\_size}(r, n) \wedge x = \mathtt{null} \wedge n = 0$ is equi-satisfiable with
  $\Delta_1'' \equiv \exists r \cdot \mathit{ll\_sizeN}(n) \wedge x = \mathtt{null} \wedge n = 0$

If $\mathit{ll\_sizeN}(n)$ can be represented by a Presburger formula, then Δ_1'' is decidable

Decidable Fragment

Given an inductive predicate $P(\bar{x}) \equiv \Phi$,

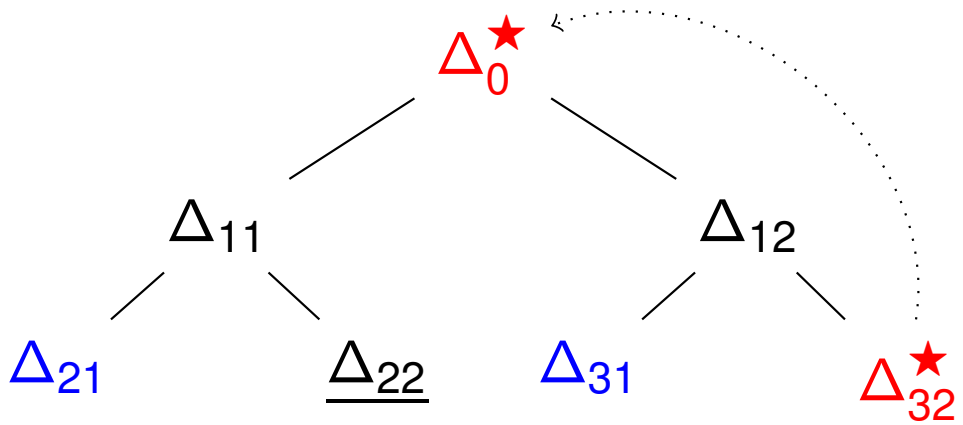
1. Construct a cyclic unfolding trees for $\Delta_0 \equiv P(\bar{x})$ through iterations of four steps:
   1. Detecting base leaves.
   2. Over-approximation: Closing unsat leaves.
   3. Back-Link Detection: Closing circular paths.
   4. Unfolding.

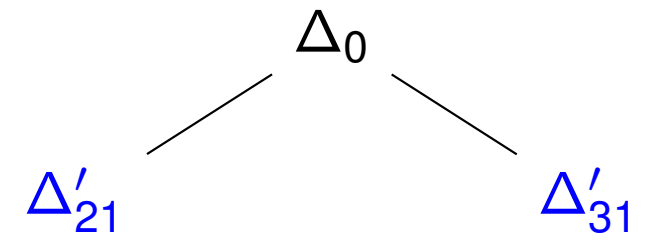


Decidable Fragment

Given an inductive predicate $P(\bar{x}) \equiv \Phi$,

1. Construct a cyclic unfolding trees
2. Flatten the cyclic unfolding trees into a disjunctive set of base formulas





$\mathtt{base}^{\mathcal{P}}(\Delta_0) \equiv \{\Delta'_{21}, \Delta'_{31}\}$

Constructing Cyclic Unfolding Tree

$$\texttt{pred}\ \mathtt{Q}(x,y,n) \equiv \exists\, y_1 \cdot x \mapsto node(\texttt{null}, y_1) \wedge y = \texttt{null} \wedge x \neq \texttt{null} \wedge n = 1$$
$$\vee\ \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) * \mathtt{Q}(x, y_1, n_1) \_ \wedge y \neq \texttt{null} \wedge n = n_1 + 2;$$

$$\Delta_0 \equiv \mathtt{Q}(x,y,n)$$

1. Base Detection. None
2. Over-Approximation. $\pi_0 \equiv \texttt{true}$. Not UNSAT
3. Cyclic Detection. None

Δ_0

Figure : Unfolding Tree $\mathcal{T}_0$.

Constructing Cyclic Unfolding Tree

$$\texttt{pred}\ \mathtt{Q}(x,y,n) \equiv \exists\, y_1 \cdot x \mapsto node(\texttt{null}, y_1) \wedge y = \texttt{null} \wedge x \neq \texttt{null} \wedge n = 1$$
$$\vee\ \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) * \mathtt{Q}(x, y_1, n_1) \_ \wedge y \neq \texttt{null} \wedge n = n_1 + 2;$$

$\Delta_0 \equiv \mathtt{Q}(x,y,n)$

$\Delta_{11} \equiv \exists\, y_1 \cdot x \mapsto node(\texttt{null}, y_1) \wedge y = \texttt{null} \wedge x \neq \texttt{null} \wedge n = 1$

$\Delta_{12} \equiv \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) * \mathtt{Q}(x, y_1, n_1) \wedge y \neq \texttt{null} \wedge n = n_1 + 2$

1. Base Detection. Δ_{11}
2. Over-Approximation.
   $\pi_{12} \equiv \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) \wedge \texttt{true}$
   $\wedge y \neq \texttt{null} \wedge n = n_1 + 2$. Not UNSAT
3. Cyclic Detection. None

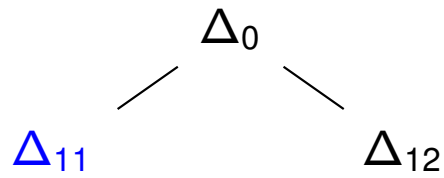


Figure : Unfolding Tree $\mathcal{T}_1$.

Constructing Cyclic Unfolding Tree

$\texttt{pred}\ Q(x,y,n) \equiv \exists\, y_1 \cdot x \mapsto node(\texttt{null}, y_1) \wedge y = \texttt{null} \wedge x \neq \texttt{null} \wedge n = 1$
$\quad \vee\ \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) * Q(x, y_1, n_1) \_ \wedge y \neq \texttt{null} \wedge n = n_1 + 2;$

$\Delta_0 \equiv Q(x,y,n)$
$\Delta_{11} \equiv \exists\, y_1 \cdot x \mapsto node(\texttt{null}, y_1) \wedge y = \texttt{null} \wedge x \neq \texttt{null} \wedge n = 1$
$\Delta_{12} \equiv \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) * Q(x, y_1, n_1) \wedge y \neq \texttt{null} \wedge n = n_1 + 2$
$\Delta_{21} \equiv \exists\, x_1, y_1, n_1, y_2 \cdot y \mapsto node(x_1, y_1) * x \mapsto node(\texttt{null}, y_2) \wedge$
$\quad y_1 = \texttt{null} \wedge x \neq \texttt{null} \wedge n_1 = 1 \wedge y \neq \texttt{null} \wedge n = n_1 + 2$
$\Delta_{22} \equiv \exists\, x_1, y_1, n_1, x_2, y_2, n_2 \cdot y \mapsto node(x_1, y_1) * y_1 \mapsto node(x_2, y_2) *$
$\quad Q(x, y_2, n_2) \wedge y_1 \neq \texttt{null} \wedge n_1 = n_2 + 2 \wedge y \neq \texttt{null} \wedge n = n_1 + 2$

1. Base Detection. Δ_{21}
2. Over-Approximation. $\pi_{22} \equiv .....$
   Not UNSAT
3. Cyclic Detection.

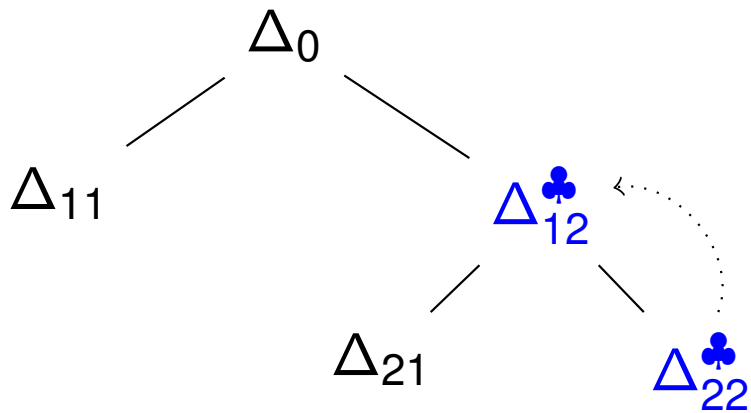


Figure : $\mathcal{T}_2^Q$.

Constructing Cyclic Unfolding Tree

Cyclic Detection

$$\Delta_{12} \equiv \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) * \mathcal{Q}(x, y_1, n_1) \wedge y \neq \texttt{null} \wedge n = n_1 + 2$$
$$\Delta_{22} \equiv \exists\, x_1, y_1, n_1, x_2, y_2, n_2 \cdot y \mapsto node(x_1, y_1) * y_1 \mapsto node(x_2, y_2) * \mathcal{Q}(x, y_2, n_2) \wedge y_1 \neq \texttt{null} \wedge n_1 = n_2 + 2 \wedge y \neq \texttt{null} \wedge n = n_1 + 2$$

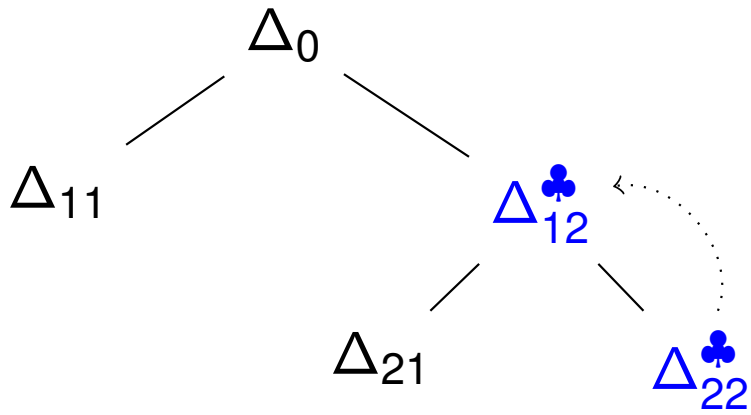
Steps

1. matching externally visible points-to predicate: $y \mapsto node(\_, \_)$
2. matching externally visible inductive predicates: $\mathcal{Q}(x, \_, \_)$
3. matching externally visible (dis)equalities over pointers: $y \neq \texttt{null}$

Flattening Cyclic Unfolding Tree

$$\Delta_{12} \equiv \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) * \mathcal{Q}(x, y_1, n_1) \wedge y \neq \texttt{null} \wedge n = n_1 + 2$$

$$\Delta_{22} \equiv \exists\, x_1, y_1, n_1, x_2, y_2, n_2 \cdot y \mapsto node(x_1, y_1) * y_1 \mapsto node(x_2, y_2) * \mathcal{Q}(x, y_2, n_2) \wedge y_1 \neq \texttt{null} \wedge n_1 = n_2 + 2 \wedge y \neq \texttt{null} \wedge n = n_1 + 2$$



$$\Delta_{21}^{flat} \equiv \exists\, x_1, y_1, n_1, y_2 \cdot (y \mapsto node(x_1, y_1) * x \mapsto node(\texttt{null}, y_2) \wedge x \neq \texttt{null} \wedge y \neq \texttt{null} \wedge n = n_1 + 1) \wedge (y_1 = \texttt{null} \wedge n_1 = 1)$$

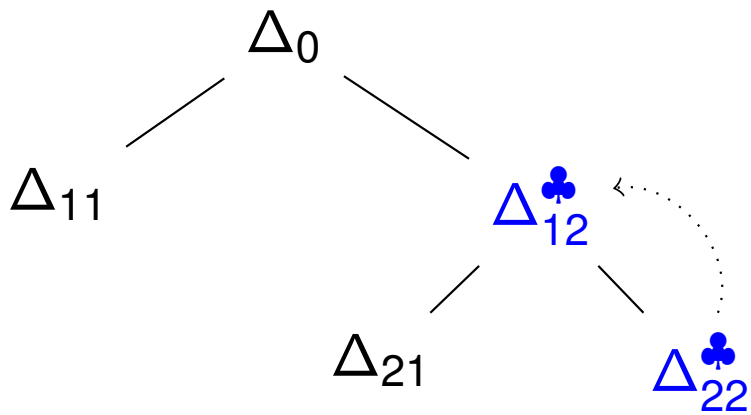
$$\vee \exists\, x_1, y_1, n_1, x_2, y_2, n_2, y_3 \cdot (y \mapsto node(x_1, y_1) * x \mapsto node(\texttt{null}, y_3) \wedge x \neq \texttt{null} \wedge y \neq \texttt{null} \wedge n = n_1 + 1) * (y_1 \mapsto node(x_2, y_2) * \wedge y_2 = \texttt{null} \wedge \underline{n_1 = n_2 + 2} \; n_2 = 1)$$

$$\vee \ldots$$

Flattening Cyclic Unfolding Tree

$$\Delta_{12} \equiv \exists\, x_1, y_1, n_1 \cdot y \mapsto node(x_1, y_1) * \mathcal{Q}(x, y_1, n_1) \wedge y \neq \mathtt{null} \wedge n = n_1 + 2$$

$$\Delta_{22} \equiv \exists\, x_1, y_1, n_1, x_2, y_2, n_2 \cdot y \mapsto node(x_1, y_1) * y_1 \mapsto node(x_2, y_2) * \mathcal{Q}(x, y_2, n_2) \wedge y_1 \neq \mathtt{null} \wedge n_1 = n_2 + 2 \wedge y \neq \mathtt{null} \wedge n = n_1 + 2$$

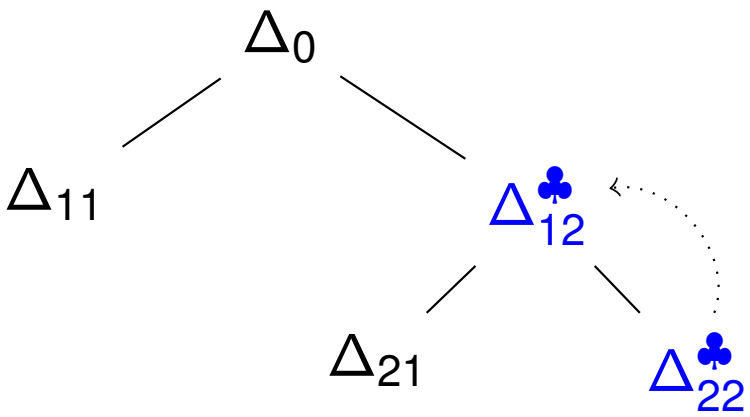


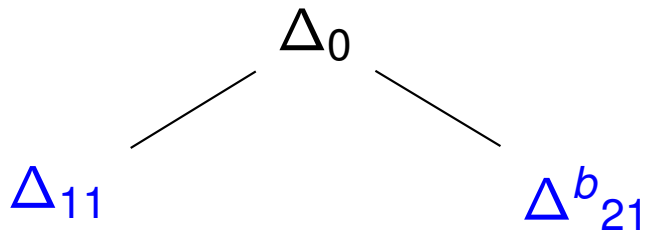
$$\mathtt{P}_{\mathtt{cyc}}(n_1) \equiv n_1 = 1 \vee \exists n_2 \cdot n_1 = n_2 + 2 \wedge \mathtt{P}_{\mathtt{cyc}}(n_2)$$

$$\Delta_{21} \equiv \exists\, x_1, y_1, n_1, y_2 \cdot y \mapsto node(x_1, y_1) * x \mapsto node(\mathtt{null}, y_2) \wedge y_1 = \mathtt{null} \wedge x \neq \mathtt{null} \wedge n_1 = 1 \wedge y \neq \mathtt{null} \wedge n = n_1 + 2$$

$$\Delta^{b}_{21} \equiv \exists\, x_1, y_1, x_2, y_2, n_1 \cdot (y \mapsto node(x_1, y_1) * x \mapsto node(\mathtt{null}, y_2) \wedge x \neq \mathtt{null} \wedge y \neq \mathtt{null} \wedge n = n_1 + 1) \wedge (\exists k \cdot n_1 = 2k + 1 \wedge k \geq 0)$$

Flattening Cyclic Unfolding Tree





$\texttt{base}^{\mathcal{P}}(\Delta_0) \equiv \{\Delta_{11}, \Delta^{b}_{21}\}$

Decidable Fragment

Relies on the deciability of arithmetic inductive predicates.
Some systems can be used:

- DPI (Tatsuta *et. al.* - APLAS:2016)
- periodic sets (Bozga *et. al.* - CAV 2010)

Implementation and Evaluation

Implemented the decision procedure based on HIP/SLEEK (Chin *et. al SCP12*).

1. SLL: $\{\texttt{emp}\wedge\texttt{root}{=}\texttt{null}\wedge n{=}0; \texttt{root}\mapsto c_1(\_,\_)\wedge n{>}0\}$
2. Even llist: $\{\texttt{emp}\wedge\texttt{root}{=}\texttt{null}\wedge n{=}0; \exists i\cdot\ \texttt{root}\mapsto c_1(\_,\_)\wedge i{>}0 \wedge n{=}2{*}i\}$
3. Doubly llist: $\{\texttt{emp}\wedge\texttt{root}{=}\texttt{null}\wedge n{=}0; \texttt{root}\mapsto c_1(\_,\_)\wedge n{>}0\}$
4. TLL: $\{\texttt{root}\mapsto c_4(\_,\_,\_,\_)\wedge\texttt{root}{=}\texttt{ll}; \texttt{root}\mapsto c_4(\_,\_,\_,\_){*}ll\mapsto c_4(\_,\_,\_,\_)\}$

Implementation and Evaluation

Over-approximation

1. CompleteT (size, minheight):

$$\{\texttt{emp}\wedge\texttt{root}=\texttt{null}\wedge n=0\wedge minh=0;\ \texttt{root}\mapsto c_2(\_,\_)\wedge n\leq 2*minh-1\wedge nmin\leq n;\}$$

2. Heap trees (size, maxelem):

$$\{\texttt{emp}\wedge\texttt{root}=\texttt{null}\wedge n=0\wedge mx=0;\ \texttt{root}\mapsto c_2(\_,\_)\wedge n>0\wedge mx\geq 0\}$$

3. AVL (height, size):

$$\{\texttt{emp}\wedge\texttt{root}=\texttt{null}\wedge n=0\wedge bal=1;\ \texttt{root}\mapsto c_2(\_,\_)\wedge h>0\wedge n\geq h\wedge n\geq 2*h-2\}$$

4. RBT(size, color, blackheight) ....

Implementation and Evaluation

Table : Experimental Results on Satisfiability Problems

Data Structure (pure props)	#q	#unsat	#sat	sec.
Singly llist (size)	666	75	591	0.85
Even llist (size)	139	125	14	1.04
Sorted llist (size, sorted)	217	21	196	0.46
Doubly llist (size)	452	50	402	1.08
CompleteT (size, minheight)	387	33	354	55.41
Heap trees (size, maxelem)	487	67	400	7.22
AVL (height, size)	881	64	817	52.15
RBT (size, blackheight, color)	1741	217	1524	40.85
TLL	128	13	115	0.39

Conclusion

A decision procedure for an extensible decidable fragment in separation logic including general inductive predicates and arithmetic

More to do

1. using the decision procedure in verifying memory safety for heap programs, array programs
2. applying the algorithm into other logics (i.e., string logic)