A Complete Decision Procedure for Linearly Compositional Separation Logic with Data Constraints

Xincai Gu,
State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences
Taolue Chen,
Department of Computer Science,
Middlesex University London
Zhilin Wu,
State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences

Outline

- Background
- Linearly compositional Separation logic with inductive definitions ($SLID_{LC}$)
- Satisfiability
- Entailment

Background

- **Dynamic data structures**
  - Plenty of them
    - lists,
    - nested lists,
    - trees (binary search trees, red-black trees,…)
  - Widely used in system software
    - operating system kernel
    - drivers
    - …
  - Notoriously difficult to specify, analyze, and verify
    - unbounded size
    - data values over infinite domain
    - pointer arithmetic

Background

- **Separation logic (SL)**
  - Proposed around 2000, by P.O.Hearn, J.C. Reynolds, H. Yang, …
  - Local reasoning about heaps: Separation + Frame rule
  - Syntax

    First order logic + separating operators (separating conjunction and implication)
  - SL with inductive definitions (SLID)

    $$X \neq Z \wedge X \mapsto (next, Y) * lseg(Y, Z)$$

    Inductive definition of *lseg*:

    $lseg(E, F) := E = F \wedge emp,$

    $lseg(E, F) := \exists X.\ E \mapsto (next, X) * lseg(X, F)$

Background

- **SLID solvers & verification tools based on SLID**
  - Smallfoot (UCL, UK),
  - Infer (Facebook, Queen Mary, Univ. of London, UK),
  - VeriFast (KU Leuven, Belgium),
  - SLEEK/HIP (NUS, Sigapore),
  - DRYAD (UIUC, USA),
  - CELIA (LIAFA, France),
  - SPEN (LIAFA, France),
  - SLIDE (Verimag, France)
  - GRASSHopper (NYU, USA)

Background

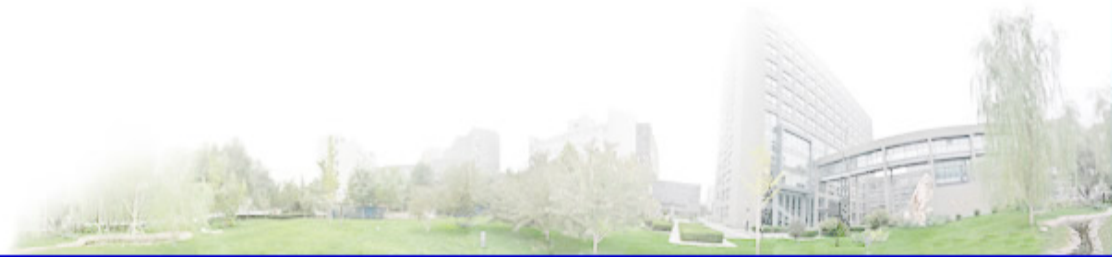
- **Usually incomplete decision procedures, especially when involving data and size constraints, e.g.**
  - VeriFast
  - SLEEK/HIP
  - DRYAD
  - SPEN
- **Why complete decision procedures ?**
  - Theoretically appealing
  - Practical importance, e.g.
    - Specification consistency,
    - Counterexample generation

Background

- **Our contribution**
  - **Linearly compositional** SLID ($SLID_{LC}$)

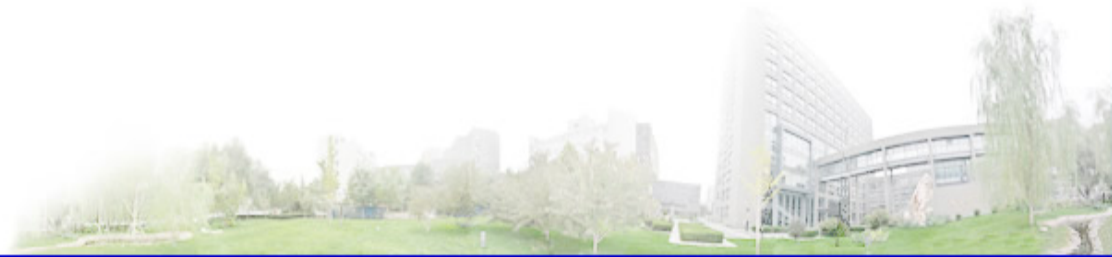
    lists, doubly linked lists, sorted lists,

    lists with length constraints, lists with tail pointers, …
  - **Complete** Decision procedures for $SLID_{LC}$
    - satisfiability
    - entailment



Outline

- Background
- Linearly compositional Separation logic with inductive definitions ($SLID_{LC}$)
- Satisfiability
- Entailment



Linearly compositional SLID ($SLID_{LC}$)

- **Inductive predicates $P(E, \alpha; F, \beta; \xi)$**
  - E, α: source parameters,
  - F, β: destination parameters,
  - ξ: static parameters
- **For instance,**
  - *lseg*(E; F): list segments from E to F
  - *slseg*(E, x; F, x'): sorted list segments from E to F, x is the data value in E, x' is the data value in F,
  - *dllseg*(E, P; F, L): doubly linked list segments from E to F
  - *tlseg*(E; F; B): list segments from E to F with tail pointers to B.

Linearly compositional SLID ($SLID_{LC}$)

- **Syntax**

$\Pi ::= E = F \mid E \neq F \mid \Pi \wedge \Pi$ (Pure formula)

$\Delta ::= true \mid x \circ c \mid x \circ y + c \mid \Delta \wedge \Delta$ (Data formula)

$\Sigma ::= emp \mid E \mapsto \rho \mid P(E, \alpha; F, \beta; \xi) \mid \Sigma * \Sigma$ (Spatial formula)

$\rho ::= (f, X) \mid (d, x) \mid \rho, \rho$

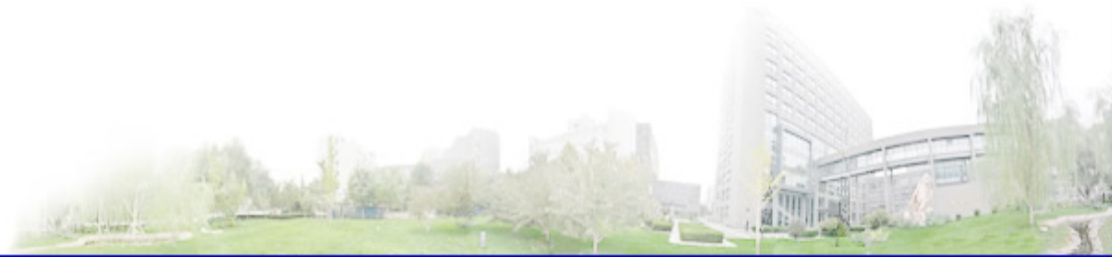
where $\circ \in \{=, \leq, \geq\}$, $c \in \mathbb{Z}$, $f \in \mathcal{F}$, $d \in \mathcal{D}$ and $P(E, \alpha; F, \beta; \xi)$ is a linearly compositional inductive predicate.

Linearly compositional SLID ($SLID_{LC}$)

- **An inductive predicate $P(E, \alpha; F, \beta; \xi)$ is linearly compositional if it is defined by the following two rules**
  - base rule: $P(E, \alpha; F, \beta; \xi) ::= E = F \wedge \alpha = \beta \wedge emp$,
  - inductive rule:
    $P(E, \alpha; F, \beta; \xi) ::= \exists X \exists x.\ \Delta \wedge E \mid\!\rightarrow \rho * P(Y, \gamma; F, \beta; \xi)$ s.t.
    - none of the variables from F, β occurs in $\Delta \wedge E \mid\!\rightarrow \rho$ (which guarantees the composition lemma $P(E_1,\alpha_1; E_2,\alpha_2; \xi) * P(E_2,\alpha_2; E_3,\alpha_3; \xi) \Rightarrow P(E_1,\alpha_1; E_3,\alpha_3; \xi)$)
    - each conjunct of Δ is of the form $\alpha_i \circ \xi_j + c$ or $\alpha_i \circ \gamma_i + c$
    - *for each* α_i, if α_i is a data variable, then either α_i occurs in ρ, or $\alpha_i = \gamma_i + c$ is a conjunct of Δ
    - ...

Linearly compositional SLID ($SLID_{LC}$)

- **Semantics**
  - A state (s, h),
    - s a store: assignment of values to variables
    - h a heap:
      a partial function of finite domain from
      $L \times (F \cup D)$ to $L \cup Z$
  - The semantics is defined by the relation $(s, h) \models \varphi$
    - $(s, h) \models \varphi_1 * \varphi_2$ if h is partitioned into two domain-disjoint heaps h_1 and h_2 s.t. $(s, h_1) \models \varphi_1$ and $(s, h_2) \models \varphi_2$
    - The semantics of $P(E, \alpha; F, \beta; \xi)$ is defined as the least fixed point.



Linearly compositional SLID ($SLID_{LC}$)

- **Examples**
  - Sorted list segments,

    $slseg(E, x; F, x') ::= E = F \wedge x{=}x' \wedge emp,$

    $slseg(E, x; F, x') ::= \exists X, x''.\ x \leq x'' \wedge E|{\rightarrow}((next, X),(data, x))^* slseg(X, x''; F, x')$
  - Doubly linked list segments,

    $dllseg(E, P; F, L) ::= E = F \wedge P = L \wedge emp,$

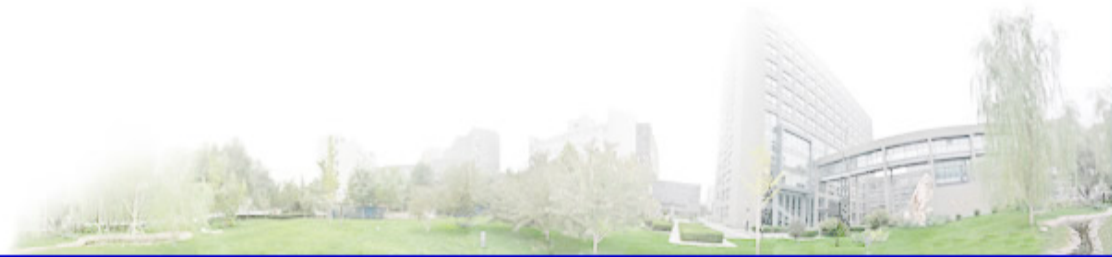
    $dllseg(E, P; F, L) ::= \exists X.\ E\ |{\rightarrow}((next, X),(prev, P))^* dllseg(X, E; F, L)$
  - Doubly linked list segments with length function,

    $ldllseg(E, P, x; F, L, x') ::= E = F \wedge P = L \wedge x{=}x' \wedge emp,$

    $ldllseg(E, P, x; F, \mathrm{L}, x') ::= \exists X, x''.\ x{=}x''{+}1 \wedge E|{\rightarrow}((next, X),(prev,P)) * ldllseg(X, E, x''; F, L, x')$
  - ...

Outline

- Background
- Linearly compositional Separation logic with inductive definitions ($SLID_{LC}$)
- Satisfiability
- Entailment



Satisfiability

- **Thm. Satisfiability of $SLID_{LC}$ is in NP.**
  - Reducuction to satisfiability of quantifier free linear arithmetic constraint (discharged by some SMT solver e.g. Z3)
  - E.g. $\varphi = E_1 = E_4 \wedge x_1 > x_2 + 1 \wedge ldllseg(E_1, E_3, x_1; E_2, E_4, x_2)$
    - The formula φ is unsatisfiable
    - $Abs(\varphi) = E_1 = E_4 \wedge x_1 > x_2 + 1 \wedge$
      $(\neg[E_1, 1] \wedge \neg[E_4, 1] \wedge E_1 = E_2 \wedge E_3 = E_4 \wedge x_1 = x_2 \wedge k_1 = 0) \vee$
      $([E_1, 1] \wedge [E_4, 1] \wedge E_1 = E_4 \wedge k_1 = 1 \wedge x_1 = x_2 + k_1) \vee$
      $([E_1, 1] \wedge [E_4, 1] \wedge E_1 \neq E_4 \wedge k_1 \geq 2 \wedge x_1 = x_2 + k_1)$
      where $[E_1, 1]$, $[E_4, 1]$ are the newly introduced Boolean variables
      The formula $Abs(\varphi)$ is unsatisfiable.

Outline

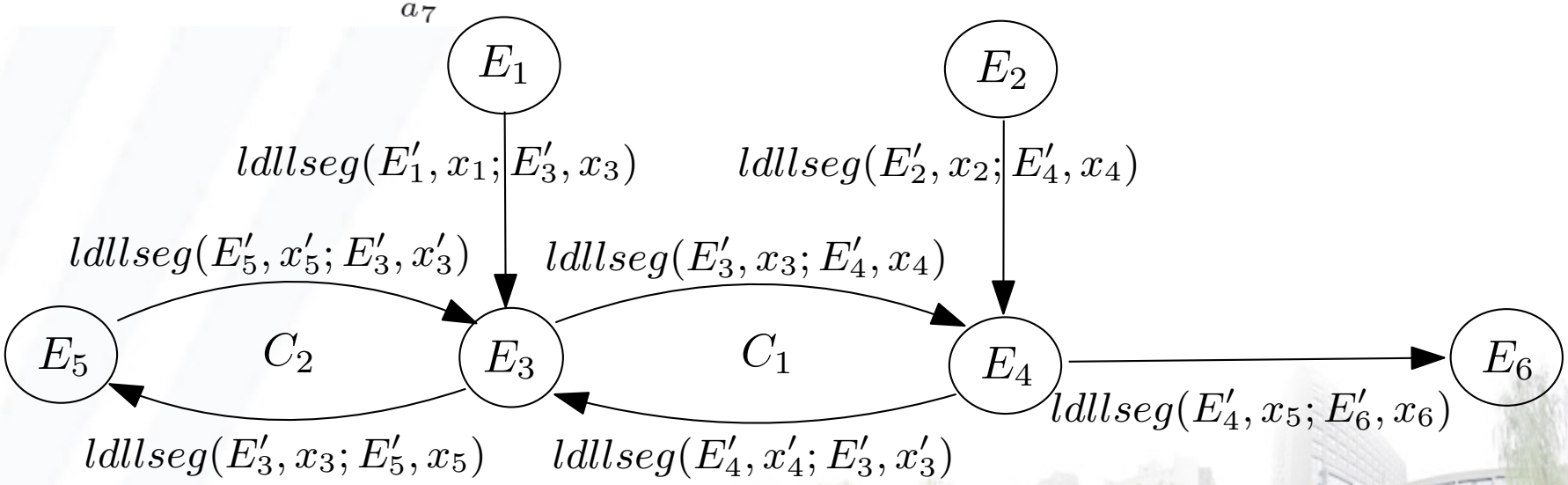
- Background
- Linearly compositional Separation logic with inductive definitions ($SLID_{LC}$)
- Satisfiability
- Entailment

Entailment

- **Thm. Entailment of $\mathbf{SLID}_{\mathbf{LC}}$ is in Π_3^P**
  - The problem $\varphi \models \psi$ s.t. $\mathrm{Vars}(\psi) \subseteq \mathrm{Vars}(\varphi)$, and $\mathit{Flds}(\varphi) = \mathit{Flds}(\psi)$.
    E.g. $\mathit{ldllseg}(E, P, x; F, L, x') \models \mathit{dllseg}(E, P; F, L)$
  - Idea: Extend the idea of graph homomorphisms (Cook et al. CONCUR 2011)
  - Proof sketch
    - $\varphi \Rightarrow G_\varphi$ (resp. $\psi \Rightarrow G_\psi$):
      $E \sim F$ iff $\mathit{Abs}(\varphi) \models E = F$ (resp. $\mathit{Abs}(\psi) \models E = F$)
    - Allocating plans $\mathcal{AP}$ of G_φ :
      $G_{\mathcal{AP}}$: DAG-like graphs obtained by simplifying G_φ
    - For each allocating plan $\mathcal{AP}$,
      check the existence of a homomorphism from G_ψ to $G_{\mathcal{AP}}$

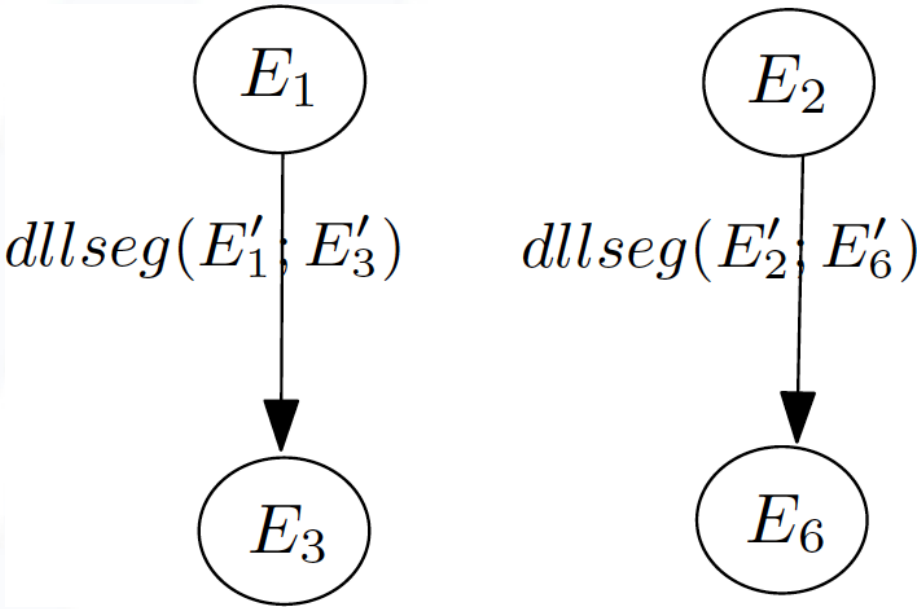
Graph representation G_φ

$$\varphi = \underbrace{ldllseg(E_1, E_1', x_1; E_3, E_3', x_3)}_{a_1} * \underbrace{ldllseg(E_2, E_2', x_2; E_4, E_4', x_4)}_{a_2} *$$
$$\underbrace{ldllseg(E_3, E_3', x_3; E_4, E_4', x_4)}_{a_3} * \underbrace{ldllseg(E_4, E_4', x_4'; E_3, E_3', x_3')}_{a_4} *$$
$$\underbrace{ldllseg(E_3, E_3', x_3; E_5, E_5', x_5)}_{a_5} * \underbrace{ldllseg(E_5, E_5', x_5'; E_3, E_3', x_3')}_{a_6} *$$
$$\underbrace{ldllseg(E_4, E_4', x_5; E_6, E_6', x_6)}_{a_7}.$$

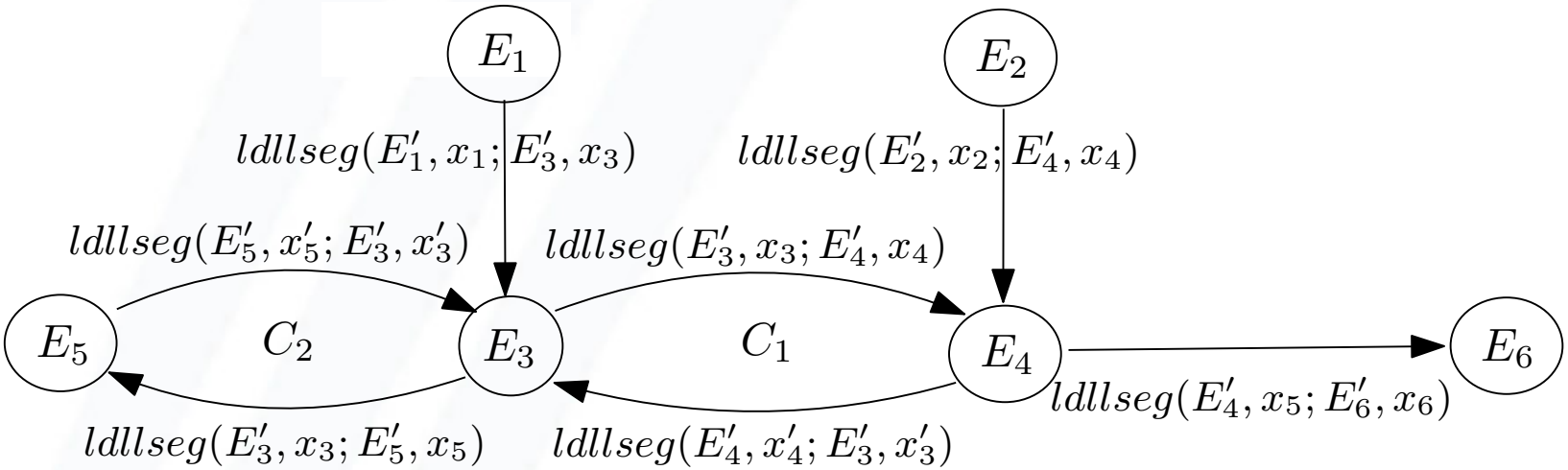


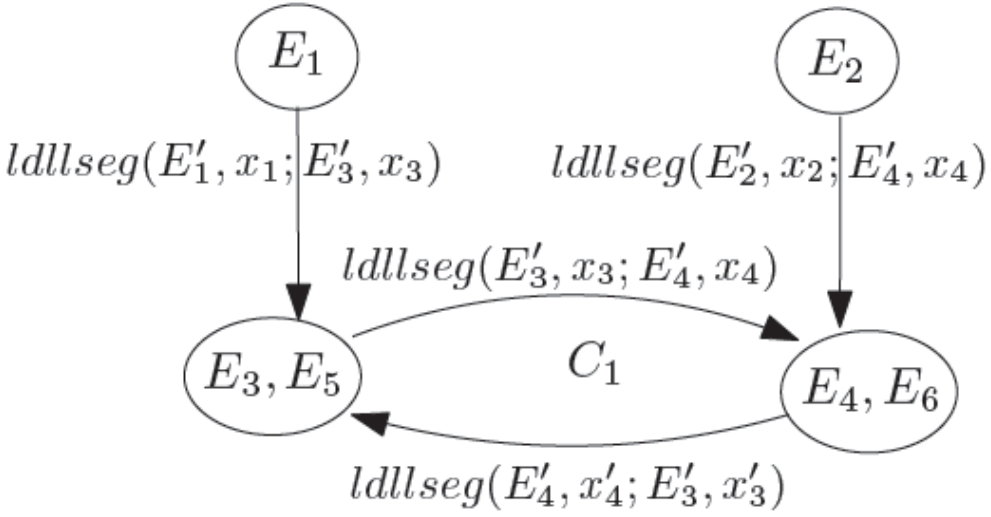
Graph representation G_ψ

- $\psi = dllseg(E_1, E_1'; E_3, E_3') * dllseg(E_2, E_2'; E_6, E_6')$

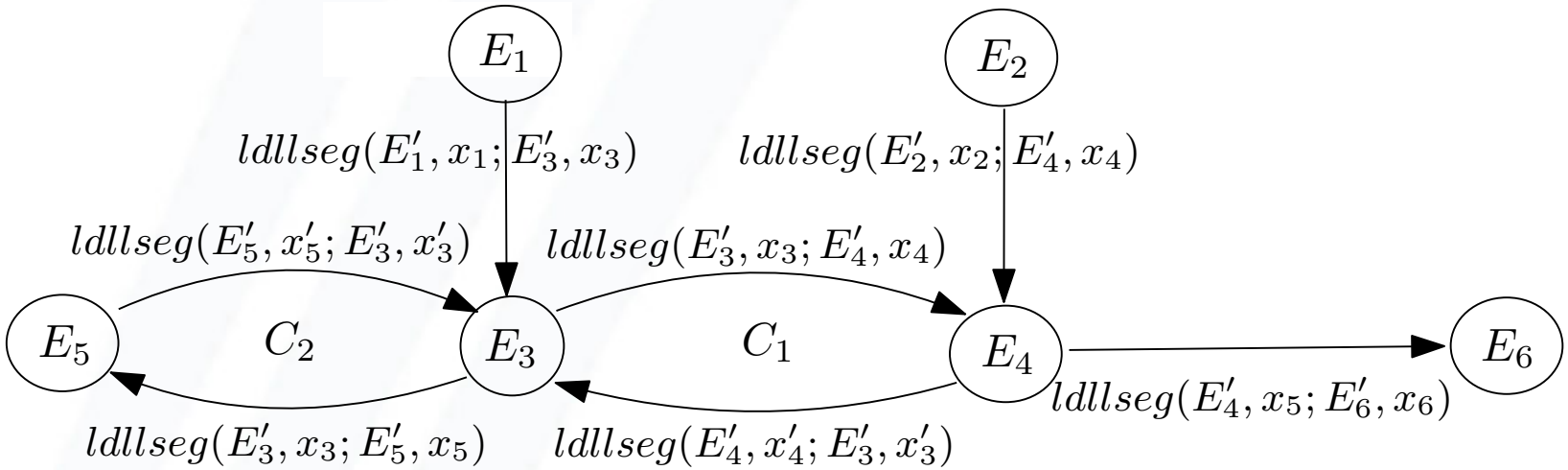


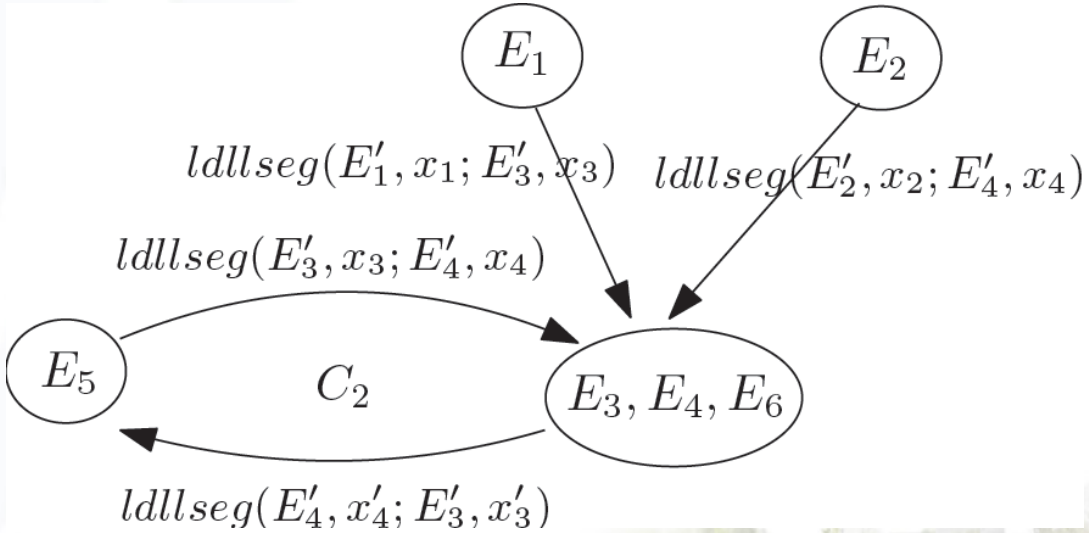
Allocating plans of G_φ



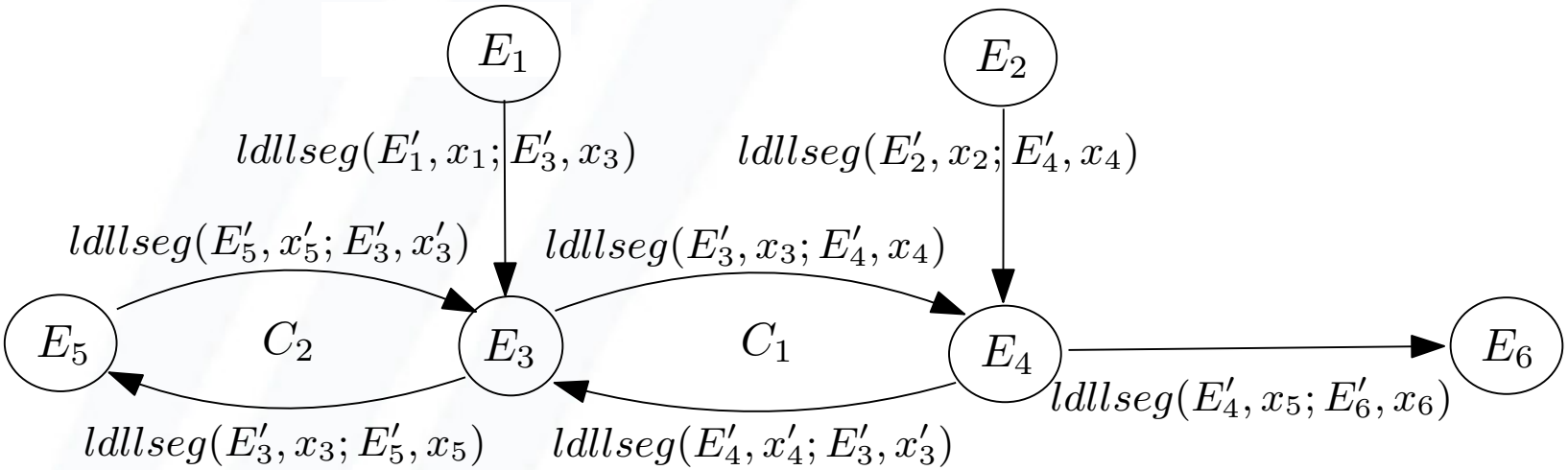


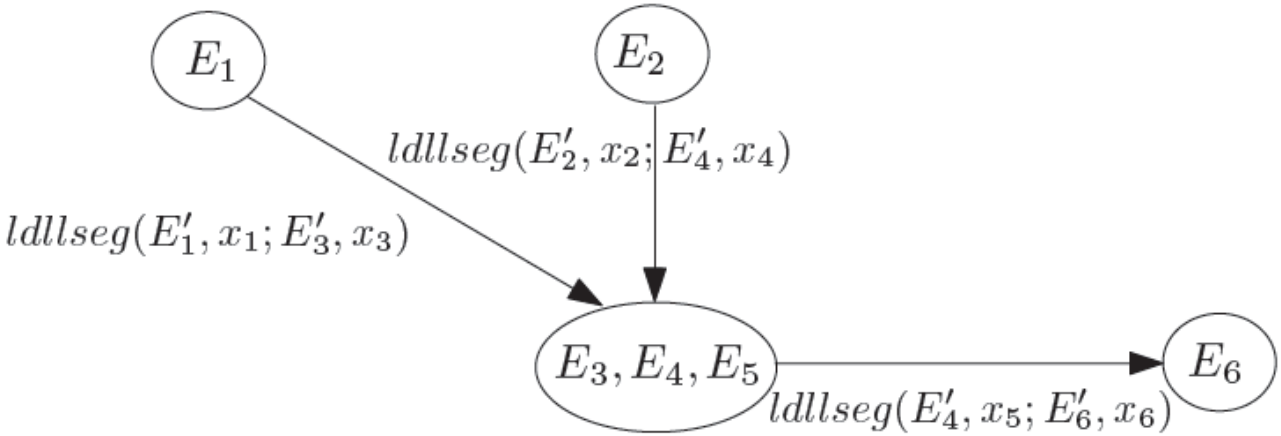
Allocating plans of G_φ



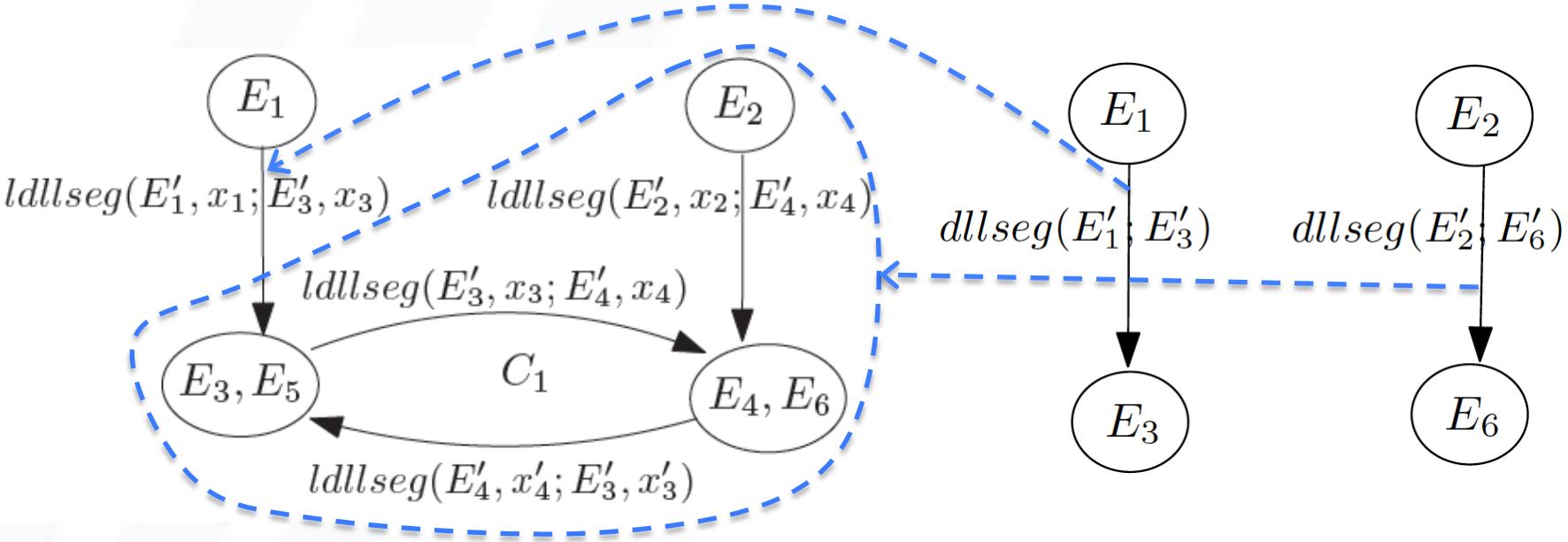


Allocating plans of G_φ

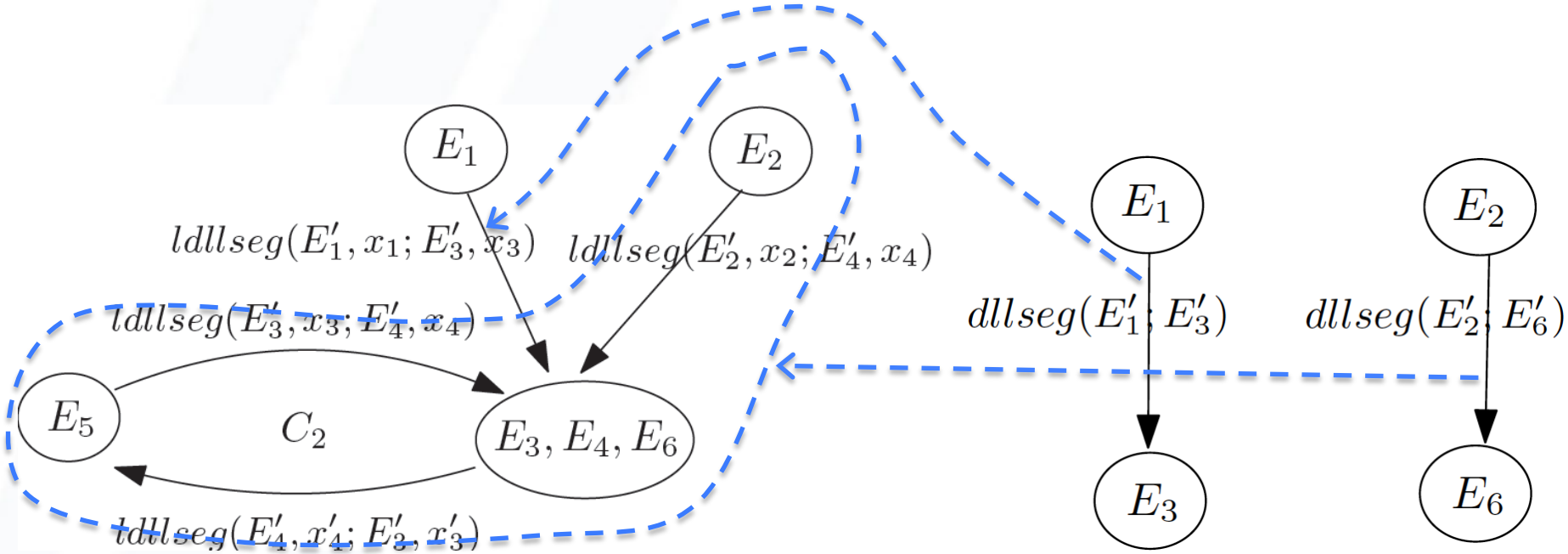




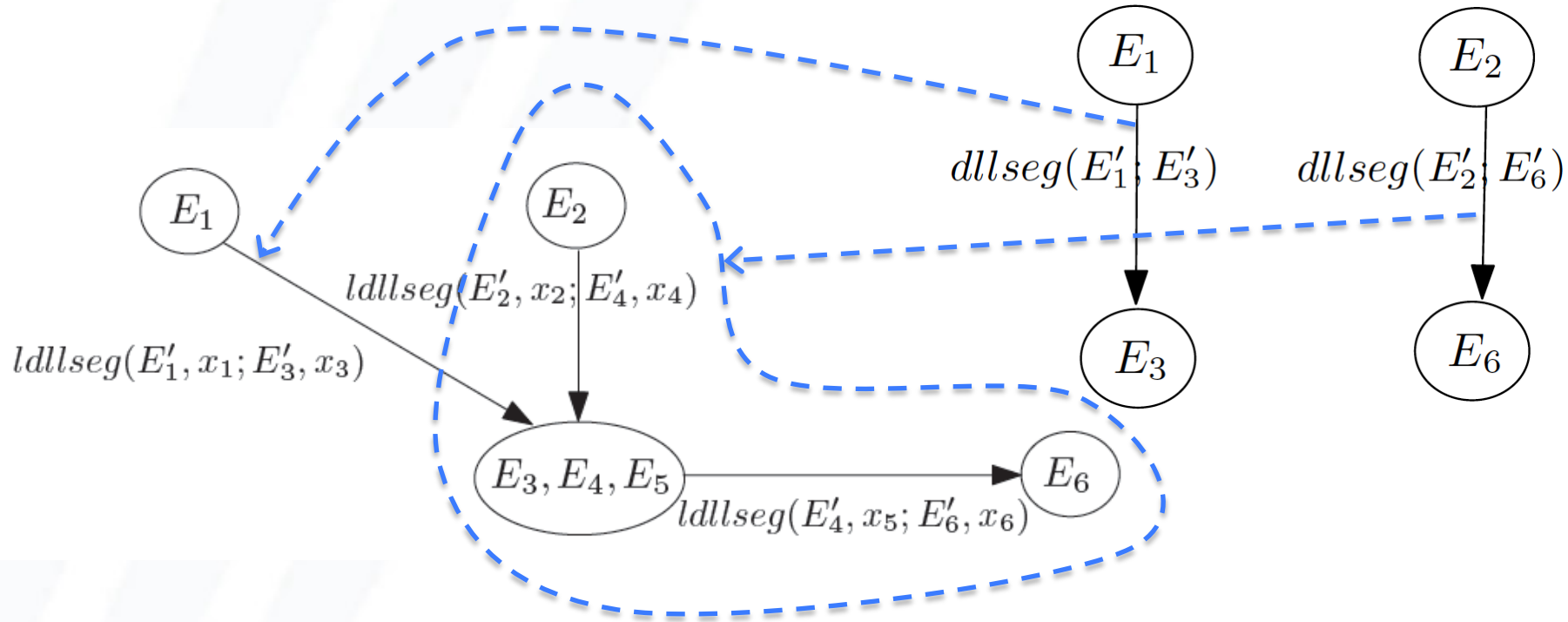
Homomorphism from G_{ψ} to $G_{\mathcal{AP}}$



Homomorphism from G_{ψ} to $G_{\mathcal{AP}}$



Homomorphism from G_{ψ} to $G_{\mathcal{AP}}$



Conclusion and future work

- **To summarize**
  - Linearly compositional separation logic with inductive definitions ($SLID_{LC}$)
  - Complete decision procedures for the satisfiability and entailment problem of $SLID_{LC}$
- **Ongoing and future work**
  - We are implementing the decision procedures (Based on SPEN)
  - Extend to set, multiset constraints
  - Extend to tree structures (e.g. binary search trees, red-black trees, …)