Normalization of symbolic heaps for entailment checking in concurrent separation logic with fractional permissions

LEE Yeonseok (Nagoya University), NAKAZAWA Koji (Nagoya University)

研究目標

背景：separation logicの証明体系でdecidable entailment checkが必要

目標：symbolic Heapとlist segment述語に制限したfractional permissionがあるentailment判定問題のdecidabilityを示す

Separation Logic [Reynolds2002]

- ポインタを使うプログラムを検証するためHoare論理の拡張

  {pre-condition} C {post-condition}
- $x \mapsto y$ (ヒープ中のポインタを表現)
- $P * Q$ (separating conjunction) : P and Q hold for **disjoint** portions (heaps) → 分離が大事 → **modular reasoning**
- inductively defined predicates (e.g. list segments (ls) or tree) → 再帰的データ構造を表現するため

Concurrent SL [Brotherston2020 & Brookes2007]

$$\frac{\{(\mathsf{ls}\ xy)^{0.5}\}\ foo(x,y)\ \{(\mathsf{ls}\ xy)^{0.5}\} \qquad \{(\mathsf{ls}\ xy)^{0.5}\}\ foo(x,y)\ \{(\mathsf{ls}\ xy)^{0.5}\}}{\{(\mathsf{ls}\ xy)^{0.5} \circledast (\mathsf{ls}\ xy)^{0.5}\}\ foo(x,y) \parallel foo(x,y)\ \{(\mathsf{ls}\ xy)^{0.5} \circledast (\mathsf{ls}\ xy)^{0.5}\}}$$

$foo(x,y)$ は list segments ls(x,y)をreadするだけ、書き込みはしない

- read-only : $0 <$ permission value < 1, e.g. $x \overset{0.5}{\mapsto} 3$
- writable : permission value $= 1$, e.g. $y \overset{1.0}{\mapsto} 2$

  Data Race を避けるため！
- $P \circledast Q$ (weak separating conjunction) : $P * Q$ あるいは **overlap**される部分に対して、permission valueの足し算を行う
- permission heap model

  s : var → val、変数間の等価関係などを表現

  h : loc $\to_{fin}$ val × perm、Heapの状態を表現

  ρ : label → val × perm、assigning a single p-heap $\rho(\alpha)$ to each label α.
- $*$ と $\circledast$ のSemantics

  $s, h, \rho \vDash \Sigma_1 * \Sigma_2 \Leftrightarrow$
  $\exists h_1, h_2 . h = h_1 \circ h_2$ and $s, h_1, \rho \vDash \Sigma_1$ and $s, h_2, \rho \vDash \Sigma_2$

  $s, h, \rho \vDash \Sigma_1 \circledast \Sigma_2 \Leftrightarrow$
  $\exists h_1, h_2 . h = h_1 \bar{\circ} h_2$ and $s, h_1, \rho \vDash \Sigma_1$ and $s, h_2, \rho \vDash \Sigma_2$

Labelの役割

- 最終的には

  $\{(\mathsf{ls}\ xy)^{1.0}\}\ foo(x,y) \parallel foo(x,y)\ \{(\mathsf{ls}\ xy)^{1.0}\}$を導きたい、しかし、$(\mathsf{ls}\ xy)^{0.5} \circledast (\mathsf{ls}\ xy)^{0.5} \nvDash (\mathsf{ls}\ xy)^{1.0}$

→ lost the information that two $(\mathsf{ls}\ xy)^{0.5}$ were actually from the **SAME** heap. 例えば、

$(x \overset{0.5}{\mapsto} y \circledast y \overset{0.5}{\mapsto} y) \circledast x \overset{0.5}{\mapsto} y \equiv x \mapsto y \circledast y \overset{0.5}{\mapsto} y \nvDash (\mathsf{ls}\ xy)^{1.0}$

- labelの役割：**denoting the same heap**
- labelによる証明例：

$$\{(\mathsf{ls}\ xy)^{1.0}\}$$
$$\{(\alpha \wedge \mathsf{ls}\ xy)^{1.0}\}$$
$$\{(\alpha \wedge \mathsf{ls}\ xy)^{0.5} \circledast (\alpha \wedge \mathsf{ls}\ xy)^{0.5}\}$$
$$\{(\alpha \wedge \mathsf{ls}\ xy)^{0.5}\}\ foo(x,y)\ \{(\alpha \wedge \mathsf{ls}\ xy)^{0.5}\} \quad \parallel \quad \{(\alpha \wedge \mathsf{ls}\ xy)^{0.5}\}\ foo(x,y)\ \{(\alpha \wedge \mathsf{ls}\ xy)^{0.5}\}$$
$$\{(\alpha \wedge \mathsf{ls}\ xy)^{0.5} \circledast (\alpha \wedge \mathsf{ls}\ xy)^{0.5}\}$$
$$\{(\alpha \wedge \mathsf{ls}\ xy)^{1.0}\}$$
$$\{(\mathsf{ls}\ xy)^{1.0}\}$$

labelのおかげでpermission valueの足し算ができる

Symbolic Heapへの制限

symbolic heap : $\Pi \wedge \Sigma$

$\Pi ::= x = y \mid x \neq y \mid \Pi \wedge \Pi \mid @_\alpha \Sigma$

$\Sigma ::= \mathsf{emp} \mid x \mapsto y \mid \mathsf{ls}(x,y) \mid \Sigma * \Sigma \mid \Sigma \circledast \Sigma \mid \Sigma^\pi \mid \alpha$

Π : 変数の性質　Σ : heapの状態を表現

$s, h, \rho \vDash @_\alpha \Sigma \Leftrightarrow s, \rho(\alpha), \rho \vDash \Sigma$

$s, h, \rho \vDash \alpha \Leftrightarrow h = \rho(\alpha)$

$\Pi \wedge \Sigma \Longrightarrow \Pi \wedge (\Sigma \wedge \alpha) \equiv (\Pi \wedge @_\alpha \Sigma) \wedge \alpha$

label_intro by [Brotherston2020]　transformation via the equivalence (≡)

正規形

$\Pi_{nf} ::= x = y \mid x \neq y \mid \Pi_{nf} \wedge \Pi_{nf}$

$\Sigma_{nf} ::= \mathsf{emp} \mid x \overset{\pi}{\mapsto} y \mid \mathsf{ls}(x,y)^\pi \mid \Sigma_{nf} * \Sigma_{nf} \mid \Sigma_{nf} \circledast \Sigma_{nf}$

Π_{nf}: @-free

Σ_{nf}: label-free & permission values are only atomic form

- 正規化は以下のstepからなる
  1. simplify (permission valueの計算)
     - distribution : $(\Sigma * \Sigma)^\pi \equiv \Sigma^\pi * \Sigma^\pi$
     - addition : $\Sigma^\pi \circledast \Sigma^\sigma \equiv \Sigma^{\pi \oplus \sigma}$

       // Σがlabelならば、いつでも足し算できる
  2. label_elimination

     $(\Pi \wedge @_\alpha \Sigma) \wedge (\Sigma' \circledast \alpha^{0.5}) \vDash \Pi \wedge (\Sigma' \circledast \Sigma^{0.5})$

正規化の例

$@_\alpha \mathsf{tree}(x)^\pi \mid \alpha^{0.5} \circledast \alpha^{0.5} \Rightarrow$ // simplify

$@_\alpha \mathsf{tree}(x)^\pi \mid \alpha^{1.0} \Rightarrow$ // label elimination

$\top \mid \mathsf{tree}(x)^\pi$

Conjecture

1. Validity of entailments is unchanged by the normalization

$$\Pi \wedge \Sigma \vDash \Pi' \wedge \Sigma'$$
$$\Updownarrow \quad nf(\Pi \wedge \Sigma) : \Pi \wedge \Sigma\text{の正規形}$$
$$nf(\Pi \wedge \Sigma) \vDash nf(\Pi' \wedge \Sigma')$$

2. 正規形のentailment checkはdecidable
→ [Berdine2005]の体系に帰着できると予想

References

[Reynolds2002] Reynolds, John C. "Separation logic: A logic for shared mutable data structures." *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*. IEEE, 2002.

[Brookes2007] Brookes, Stephen. "A semantics for concurrent separation logic." *Theoretical Computer Science* 375.1-3 (2007): 227-270.

[Berdine2005]Berdine, Josh, Cristiano Calcagno, and Peter W. O'hearn. "Symbolic execution with separation logic." *Asian Symposium on Programming Languages and Systems*. Springer, Berlin, Heidelberg, 2005.

[Brotherston2020] Brotherston, James, et al. "Reasoning over Permissions Regions in Concurrent Separation Logic." *International Conference on Computer Aided Verification*. Springer, Cham, 2020.