分割可能なアクセス権限値をもつ並行分離論理における帰納的述語の拡張

佐藤 拓海 [1], 中澤 巧爾 [2], 木村 大輔 [3]

[1] 名古屋大学情報学部
sato.takumi@sqlab.jp,

[2] 名古屋大学大学院情報学研究科
knak@i.nagoya-u.ac.jp,

[3] 東邦大学理学部
kmr@is.sci.toho-u.ac.jp

概 要 Brotherston らが提案した SL_{LP} は，並行分離論理をメモリアクセスを制御するための権限値によって拡張した体系である．Lee らは，SL_{LP} の論理式をシンボリック・ヒープに制限した SL_{LP}^{SH} を提案し，そのエンテイルメント判定問題が決定可能であることを示したが，この体系では帰納的述語はリスト断片を表す特定のものに制限されている．本研究では，SL_{LP}^{SH} の帰納的述語を有界木幅条件と呼ばれる条件を満たす述語に一般化し，エンテイルメント判定問題を決定するアルゴリズムを与える．これにより，双方向リストや木構造を表す述語を含む SL_{LP}^{SH} のエンテイルメント判定が可能になる．

1 はじめに

並行分離論理 (concurrent separation logic) [3, 11] は，ホーア論理 [6] をヒープ・メモリを操作する並行プログラムを扱えるように拡張したものである．ホーア論理は,「論理式 A を満たす状態で，プログラム P を実行して停止すれば，論理式 B を満たす状態になっている」ことを表すホーア・トリプル

$$\{A\}\ P\ \{B\}$$

を証明する体系である．分離論理 [12] では互いに共通部分を持たないヒープ A, B の結合を表す**分離連言** $A * B$ により局所的な推論を行うことができる．さらに，並行分離論理では，(Parallel) 規則

$$\frac{\{P_1\}\ C_1\ \{Q_1\} \quad \{P_2\}\ C_2\ \{Q_2\}}{\{P_1 * P_2\}\ C_1 \parallel C_2\ \{Q_1 * Q_2\}}$$

によって，2 つのスレッド C_1, C_2 を並行に動作させた際の仕様を証明することができる．しかし，例えば C_1, C_2 が同一のメモリ領域にアクセスする場合，$P_1 * P_2$ は分離連言の意味から偽を表すことになるため，結論の仕様は自明なものになる．C_1, C_2 がメモリ上のデータを読み込むだけで変更しないようなプログラムである時，これらのプログラムを並行に実行しても共有メモリへのアクセス競合などの問題は発生しないはずであるが，従来の並行分離論理ではこのような状況を表現することができない．

この問題を解決するのが，Brotherston の体系 SL_{LP}[4] である．SL_{LP} では，**分割可能な権限値** (fractional permission) と**弱分離連言** $\circledast$ によって，あるスレッドが共有メモリの特定の領域に対して持つアクセス権限をより詳細に表現することができる．権限値 π は半開区間 $(0, 1]$ の有理数で与えられ，権限値 $0 < \pi < 1$ はデータの読み込みのみが出来ること，権限値 $\pi = 1$ はデータの読み書

きが出来ることを表す．論理式 A^{π} はヒープ領域が論理式 A を満たし，さらにそのヒープ領域へのアクセス権限値が π であることを表す．例えば，論理式 $(x \mapsto a)^{0.5}$ は，ヒープの形はアドレス x に値が a 格納されているような単元ヒープ（単一のメモリセルからなるヒープ領域）で，その権限値は 0.5，すなわち読み込みのみが許可されていることを表す．弱分離連言 $A \circledast B$ は，重複するヒープ領域の権限値の和をとることを表す．例えば，通常の分離連言において，$(x \mapsto a) * (x \mapsto a)$ は 2 つのヒープ領域が重複しているため偽であるが，$(x \mapsto a)^{0.5} \circledast (x \mapsto a)^{0.5}$ は，権限値 $0.5 + 0.5 = 1.0$ をもつ単元ヒープ $(x \mapsto a)^{1.0}$ を表す．以下では権限値が省略されている場合，その権限値は 1.0 であるとする．

SL_{LP} ではこれらの機構に加えて**ラベル**を用いて，権限値の分割と合併を表現する．例えば，アドレス x から y に至る線形リスト断片の構造を持つヒープ領域を表す帰納的述語 $\mathtt{ls}(x, y)$ に対して，権限値の分割 $\mathtt{ls}(x, y) \models \mathtt{ls}(x, y)^{0.5} \circledast \mathtt{ls}(x, y)^{0.5}$ が常に成り立つ．すなわち，論理式 $\mathtt{ls}(x, y) (= \mathtt{ls}(x, y)^{1.0})$ を満たすヒープ領域は，必ず $\mathtt{ls}(x, y)^{0.5} \circledast \mathtt{ls}(x, y)^{0.5}$ を満たす．一方で，$\mathtt{ls}(x, y)^{0.5} \circledast \mathtt{ls}(x, y)^{0.5} \models \mathtt{ls}(x, y)^{1.0}$ は必ずしも成立しない．一般に帰納的述語 $\mathtt{ls}(x, y)$ を満たすリスト断片の形は一意的ではないので，2 つの $\mathtt{ls}(x, y)$ が異なるヒープ領域に対応している可能性があるからである．例えば，アドレス x を始点として $x \mapsto y$，$y \mapsto y$ と終点 y に至るリストからなるヒープ領域 h と，$x \mapsto y$ のみからなるヒープ領域 h' はともに $\mathtt{ls}(x, y)$ を満たす．したがって，h, h' それぞれに権限値 0.5 を付与した $0.5 \bullet h$ と $0.5 \bullet h'$ を合併し，共通部分 $x \mapsto y$ の部分の権限値を加算したヒープ領域は $\mathtt{ls}(x, y)^{0.5} \circledast \mathtt{ls}(x, y)^{0.5}$ を満たすが，$\mathtt{ls}(x, y)^{1.0}$ を満たさない．

そこで，特定のヒープ領域を指示するラベルを導入し，権限値の分割時にラベルを付けることによって，分割されたヒープ領域が合併時に同一のヒープ領域である事を保証する．上の例では，分割時にラベル α を用いることによって，$(\alpha \wedge \mathtt{ls}(x, y))^{1.0} \models (\alpha \wedge \mathtt{ls}(x, y))^{0.5} \circledast (\alpha \wedge \mathtt{ls}(x, y))^{0.5}$ としておけば，$(\alpha \wedge \mathtt{ls}(x, y))^{0.5} \circledast (\alpha \wedge \mathtt{ls}(x, y))^{0.5}$ における二つの $\mathtt{ls}(x, y)^{0.5}$ が同一のヒープ領域 α を表していることが保証されるため，$(\alpha \wedge \mathtt{ls}(x, y))^{0.5} \circledast (\alpha \wedge \mathtt{ls}(x, y))^{0.5} \models (\alpha \wedge \mathtt{ls}(x, y))^{1.0}$ が成立する．

分離論理を用いたソフトウェア検証ではエンテイルメントと呼ばれる論理式間の含意関係の判定が重要である．一般に，帰納的述語を含む分離論理のエンテイルメント判定は決定不能であることが知られている [1] が，論理式を**シンボリック・ヒープ**と呼ばれる形に制限し，帰納的述語に一定の制限をかけることで決定可能なエンテイルメント判定が可能になることが知られている [2, 7, 8, 13].

Lee と中澤は SL_{LP} の論理式をシンボリック・ヒープに制限した体系 SL_{LP}^{SH} を提案し，そのエンテイルメント判定問題を，権限値やラベルを含まない既存のシンボリックヒープのエンテイルメント判定問題に帰着させ，これが決定可能であることを示している [9]．この帰着は，主に正規化とラベル消去の二つのステップで実現されている．正規化では弱分離連言による権限値の加算を行なうことで，弱分離連言のないエンテイルメントを得る．さらに，ラベル消去では，両辺のラベルの権限値を比較しつつ，ラベルによって指示されているメモリ領域を表すエンテイルメントを分離することによって，権限値やラベルを含まない通常の分離論理のエンテイルメントを得ている．しかし，SL_{LP}^{SH} は帰納的述語としてリスト断片 $\mathtt{ls}$ のみを扱っており，その証明は $\mathtt{ls}$ に特化したものであるため，より一般的な帰納的述語をもつ SL_{LP}^{SH} のエンテイルメント判定問題の決定可能性は未解決だった．

本研究では $\mathtt{ls}$ のみに制限されていた SL_{LP}^{SH} の述語を，有界木幅条件 [7] と呼ばれる条件を満たす帰納的述語で拡張しても，エンテイルメント判定問題が決定可能であること示す．先述の通り，シンボリック・ヒープのエンテイルメント判定を決定可能にするためには帰納的述語に対して何らかの制限が必要となる．有界木幅条件はそのような条件のうち最も表現力豊かな条件の一つである．双方向リストや木など，一般に利用されている多くの再帰的データ構造は，有界木幅条件を満たす述語で表現できる．

Lee らの SL_{LP}^{SH} のエンテイルメント判定 [9] においては，特にラベル消去のプロセスで Root 関数

と呼ばれる「その論理式を満たすヒープにおいて，必ず割り当てられなければならないアドレスを表す変数の集合」の計算が必要になるが，本研究では一般の帰納的述語を含む論理式に対してこのRoot関数を計算する方法を与え，ラベル消去のアルゴリズムを拡張した．Root関数の計算においては，Brothersonらが充足可能性判定のために用いた *base pair* 集合 [5] のアイディアを利用している．

本論文の構成は以下の通りである．まず，2章では論理式をシンボリックヒープに制限した並行分離論理の体系 SL_{LP}^{SH} とエンテイルメント判定の概要 [9] を紹介する．3章では有界木幅条件を満たす述語による SL_{LP}^{SH} の拡張とそれに伴うエンテイルメント判定のための証明規則などの拡張を行う．4章では拡張したRoot関数の計算の手順を与える．最後に5章では前章までのまとめと，今後の展望について述べる．

2 並行分離論理 SL_{LP}^{SH}

本章では，SL_{LP}^{SH}[9] を紹介する．SL_{LP}^{SH} は，SL_{LP}[4] の論理式をシンボリック・ヒープに制限したものである．

権限値 π は $0 < \pi \leqslant 1$ を満たす有理数とする．直観的には，あるヒープの権限値 $\pi = 1$ の時，書き込みと読み込みを許す．$0 < \pi < 1$ の時，読み込みのみを許す．$\mathtt{Perm}$ は権限値の集合を表す．

2.1 SL_{LP}^{SH} の構文と意味論

SL_{LP}^{SH} の構文

定義 2.1 (変数と項)．$\mathtt{Var} = \{x, y, \ldots\}$ は変数の集合，$\mathtt{Label} = \{\alpha, \beta, \gamma, \ldots\}$ はラベルの集合を表す．また，**項** t を以下で定義する．

$$t ::= \mathtt{nil} \mid x$$

定義 2.2 (ヒープ論理式)．**ヒープ論理式**は以下のように定義される．正整数 N は固定して考える．

$$S ::= \mathtt{emp} \mid t \mapsto (t_1, \ldots, t_N) \mid \mathtt{ls}(t, t') \mid \alpha^{\pi} \qquad \text{(原子ヒープ論理式)}$$

$$\Sigma ::= S \mid \Sigma * \Sigma \mid \Sigma \circledast \Sigma \qquad \text{(ヒープ論理式)}$$

ラベルも $\circledast$ も含まない式を，ラベル・フリー論理式とよび，Σ^{-} で表す．$FV(\Sigma)$ と $\mathrm{label}(\Sigma)$ によって，それぞれ Σ 中に現れる変数とラベルの集合を表す．$\mathtt{Spat}$ をヒープ論理式の集合とする．

$\mathtt{emp}$ は空であるヒープを表す．項の組 $(t_1, \ldots, t_N)$ を，$\vec{t}$ などとも書く．$t \mapsto \vec{t'}$ はアドレスが t であって，値の組 $\vec{t'}$ を保持する1つのメモリセルで構成される単元ヒープを表す．$\mathtt{ls}(t, t')$ は t から t' に至るリスト断片を表す（ただし，循環を許すものとする）．$t \mapsto \vec{t'}$，$\mathtt{ls}(t, t')$ の権限値の既定値は1である．α^{π} はラベル α が表すヒープであり，かつ，権限値が π であることを表す．以降，$\alpha^{1.0}$ を単に α と書く．また，SL_{LP}^{SH} において，1未満の権限値を付することはラベルにおいてのみ許される．

$\Sigma * \Sigma'$ は分離連言とよばれ，互いに共通部分を持たない，それぞれ Σ と Σ' を満たすヒープの和集合のヒープを意味する．$\circledast$ は弱分離連言とよばれ，分離連言と同様であるが，ヒープの重複部分について権限値を足し合わせることを許している点が異なる．

定義 2.3 (スタック論理式)．**スタック論理式**は以下のように定義される．

$$\Pi ::= \top \mid t = t' \mid t \neq t' \mid \Pi \wedge \Pi \qquad \text{(スタック論理式)}$$

スタック論理式 Π が，$t \neq t$ と，異なる項 t, t' に対する $t = t'$ を含まないとき，Π は単射的であるという．$\mathtt{Pure}$ をスタック論理式の集合とする．

定義 2.4 (@論理式)．**@論理式**は以下のように定義される．

$$\Xi ::= \top \mid @_\alpha \Sigma^- \mid \Xi \wedge \Xi \qquad \text{(@論理式)}$$

$\mathtt{At}$ を@論理式の集合とする．

$@_\alpha\Sigma^-$ は jump modality とよばれ，ラベル α で表されるヒープで Σ^- が成り立つことを意味する．Lee ら [9] は，@論理式はスタック論理式の一部としていたが，本論文では@論理式として区別する．

定義 2.5 (シンボリック・ヒープ)．シンボリック・ヒープ ϕ は以下のような形をとる論理式である．

$$\phi := \Pi \mid \Xi \mid \Sigma$$

ここで ϕ において，@論理式 $@_\alpha\Sigma^-$ は，Σ の各ラベル α につきちょうど 1 つだけ現れるものとする．また，Σ に含まれないラベル β に関する@論理式 $@_\beta\Sigma^-$ は現れないものとする．Π が単射的であるとき，シンボリック・ヒープ $\Pi \mid \Xi \mid \Sigma$ は単射的であるという．

ラベルに関する条件について，例えば，$\top \mid @_\alpha\Sigma_1 \wedge @_\alpha\Sigma_2 \mid \alpha$ はラベル α に関する式が 2 つ現れているので許されない．また，$\top \mid @_\alpha\Sigma_1 \mid \mathtt{emp}$ はそのヒープ論理式の部分にラベル α が含まれないので許されない．

$\Pi \mid \Xi \mid \Sigma$ の意味は $\Pi \wedge \Xi \wedge \Sigma$ と同じである．また，論理式 A において A 中に現れる変数 x すべてに項 t を代入したものを $A[x := t]$ と書く．

意味論

$\mathtt{Val} = \mathbb{N}$ を値の集合とする．また，0 でない自然数をメモリアドレスとしても用いることにし，$\mathtt{Loc} \subseteq \mathbb{N}\backslash\{0\}$ をヒープとして使用可能なロケーションの集合とする．

定義 2.6 (権限値付きヒープ・モデル)．**権限値付きヒープ・モデル**は 3 項組 (s, h, ρ) で表される．s は**スタック**，h は**権限値付きヒープ**，ρ は**ラベルの付値**を表す．

スタック s は関数 $s : \mathtt{Var} \to \mathtt{Val}$ である．スタック s は $s(\mathtt{nil}) = 0$ により項上の関数に拡張される．また，s が項の集合からの関数として単射であるとき，権限値付きヒープ・モデル (s, h, ρ) を単射的であるという．項の集合 X に対して，$s(X)$ は集合 $\{s(t) \mid t \in X\}$ を表すとする．また，項の組 $(t_1, \ldots, t_N)$ に対して，$s(\vec{t})$ は値の組 $(s(t_1), \ldots, s(t_N))$ を表すとする．

権限値付きヒープ h は有限部分関数 $h : \mathtt{Loc} \to_{fin} \mathtt{Val}^N \times \mathtt{Perm}$ である．ここで，すべての権限値付きヒープの集合を $\mathtt{PHeap}$ と表す．また，$\mathrm{dom}(h)$ で h の定義域，つまり h が定義されているロケーションの集合を表す．

ラベルの付値 ρ は関数 $\rho : \mathtt{Label} \to \mathtt{PHeap}$ である．$\rho[\alpha \to h]$ は以下のような付値を表す．

$$\rho[\alpha \to h](\beta) = \begin{cases} h & (\beta = \alpha \text{ のとき}) \\ \rho(\beta) & (\text{それ以外のとき}) \end{cases}$$

定義 2.7. 権限値付きヒープ h の権限値 π による乗算 $\pi \bullet h$ について以下のように定義する．

$$(\pi \bullet h)(l) = \begin{cases} (v, \pi \times \pi') & h(l) = (v, \pi') \text{ のとき} \\ \text{undefined} & l \notin \mathrm{dom}(h) \text{ のとき} \end{cases}$$

定義 2.8 (権限値付きヒープの合成)．2 つの権限値付きヒープ h_1, h_2 について，$\mathrm{dom}(h_1) \cap \mathrm{dom}(h_2) = \varnothing$ であるとき，それらは**互いに素**であるという．互いに素なヒープ h_1, h_2 間の演算**強いヒープ合成** $h_1 \circ h_2$ は以下のように定義される．

$$(h_1 \circ h_2)(l) = \begin{cases} h_1(l) & l \in \mathrm{dom}(h_1) \text{ のとき} \\ h_2(l) & l \in \mathrm{dom}(h_2) \text{ のとき} \\ \text{undefined} & l \notin \mathrm{dom}(h_1) \cup \mathrm{dom}(h_2) \text{ のとき} \end{cases}$$

$$
\begin{array}{rcl}
s,h,\rho \models t = t' & \Leftrightarrow & s(t) = s(t') \\
s,h,\rho \models t \neq t' & \Leftrightarrow & s(t) \neq s(t') \\
s,h,\rho \models A \wedge B & \Leftrightarrow & s,h,\rho \models A \text{ かつ } s,h,\rho \models B \\
s,h,\rho \models \top & \Leftrightarrow & \text{常に成り立つ} \\
s,h,\rho \models \mathtt{emp} & \Leftrightarrow & \mathrm{dom}(h) = \varnothing \\
s,h,\rho \models t \mapsto \vec{t'} & \Leftrightarrow & \mathrm{dom}(h) = \{s(t)\} \text{ かつ } h(s(t)) = (s(\vec{t'}), 1) \\
s,h,\rho \models \mathtt{ls}(t,t') & \Leftrightarrow & \text{ある } n \text{ で } s,h,\rho \models \mathtt{ls}^n(t,t') \\
s,h,\rho \models \mathtt{ls}^0(t,t') & \Leftrightarrow & \text{成立しない} \\
s,h,\rho \models \mathtt{ls}^{n+1}(t,t') & \Leftrightarrow & s(t) = s(t') \wedge \mathrm{dom}(h) = \varnothing, \text{ または} \\
 & & \exists a \in \mathtt{Val}. s[z := a], h, \rho \models t \mapsto (z, \mathtt{nil}, \ldots, \mathtt{nil}) * \mathtt{ls}^n(z, t') \\
s,h,\rho \models \alpha^\pi & \Leftrightarrow & h = \pi \bullet \rho(\alpha) \\
s,h,\rho \models @_\alpha A & \Leftrightarrow & s, \rho(\alpha), \rho \models A \\
s,h,\rho \models A * B & \Leftrightarrow & \exists h_1, h_2. h = h_1 \circ h_2 \text{ かつ } s,h_1,\rho \models A \text{ かつ } s,h_2,\rho \models B \\
s,h,\rho \models A \circledast B & \Leftrightarrow & \exists h_1, h_2. h = h_1 \bar{\circ} h_2 \text{ かつ } s,h_1,\rho \models A \text{ かつ } s,h_2,\rho \models B \\
s,h,\rho \models \Pi \mid \Xi \mid \Sigma & \Leftrightarrow & s,h,\rho \models \Pi \text{ かつ } s,h,\rho \models \Xi \text{ かつ } s,h,\rho \models \Sigma
\end{array}
$$

図 1. SL^{SH}_{LP} における充足関係

h_1, h_2 が互いに素でなければ $h_1 \circ h_2$ は定義されない.

2 つの権限値付きヒープ h_1, h_2 について，すべての $l \in \mathrm{dom}(h_1) \cap \mathrm{dom}(h_2)$ に関して，「$h_1(l) = (v_1, \pi_1)$ かつ $h_2(l) = (v_2, \pi_2)$ であるならば $v_1 = v_2$ かつ $\pi_1 + \pi_2 \leqslant 1$ である」とき，h_1, h_2 は**整合的である**であるという．整合的なヒープ h_1, h_2 間の演算**弱いヒープ合成** $h_1 \bar{\circ} h_2$ は以下のように定義される.

$$
(h_1 \bar{\circ} h_2)(l) = \begin{cases}
(v, \pi_1 + \pi_2) & h_1(l) = (v, \pi_1) \text{ かつ } h_2(l) = (v, \pi_2) \text{ のとき} \\
h_1(l) & l \in \mathrm{dom}(h_1) - \mathrm{dom}(h_2) \text{ のとき} \\
h_2(l) & l \in \mathrm{dom}(h_2) - \mathrm{dom}(h_1) \text{ のとき} \\
\text{undefined} & l \notin \mathrm{dom}(h_1) \cup \mathrm{dom}(h_2) \text{ のとき}
\end{cases}
$$

h_1, h_2 が整合的でなければ $h_1 \bar{\circ} h_2$ は定義されない.

以降は，以下の label-disjointness condition を満たすような権限値付きヒープ・モデルに制限して SL^{SH}_{LP} のエンテイルメント判定問題を考える.

定義 2.9. すべての $\alpha \neq \beta$ について $\rho(\alpha)$ と $\rho(\beta)$ が互いに素であるとき，権限値付きヒープ・モデル (s,h,ρ) は label-disjoint であるという.

この label-disjointness condition は@論理式について，$@_\alpha \Sigma^-$ の Σ^- がラベル・フリー論理式であるという制限を反映した条件であるといえる．つまり，$\top \mid @_\alpha \beta \wedge @_\beta \mathtt{ls}(x,y) \mid \alpha$ で表されるような，重複したヒープ領域を異なるラベルで表示するような状況を認めないということである.

定義 2.10. SL^{SH}_{LP} における充足関係 $s,h,\rho \models A$ は図 1 のように定義される．スタック論理式 Π について，$s,h,\rho \models \Pi$ は h,ρ に依らないので，単に $s \models \Pi$ と書くこともある．同様に，ラベルを含まない論理式 A' について，$s,h,\rho \models A'$ は ρ に依らないので，単に $s,h \models A'$ と書くこともある．シンボリック・ヒープ ϕ, ψ について，$\phi \models \psi$ は，任意の label-disjoint な (s,h,ρ) について，$s,h,\rho \models \phi$ ならば $s,h,\rho \models \psi$ であることを意味する．また，$A \equiv B$ で $A \models B$ かつ $B \models A$ を意味する．また，$\phi \equiv \psi$ で $\phi \models \psi$ かつ $\psi \models \phi$ を意味する.

$\{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha\}$

$\{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5} \circledast \alpha^{0.5}\}$ [エンテイルメント ($\circledast$)]

(Parallel)

$$\begin{array}{l} \{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5}\} \\ \mathtt{foo(x,y)} \\ \{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5}\} \end{array} \Bigg\| \begin{array}{l} \{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5}\} \\ \mathtt{foo(x,y)} \\ \{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5}\} \end{array}$$

$\{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5} \circledast \alpha^{0.5}\}$ (Parallel)

$\{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha\}$ [エンテイルメント ($\circledast$)]

図 2. プログラム C の証明

ここで，$\mathtt{ls}^n (n \in \mathbb{N})$ は直観的には長さが高々n であるようなリスト断片を表している．また，ls は循環を含みうることに注意する．例えば，$x \mapsto y * y \mapsto x \models ls(x,x)$ が成り立つ．また，単射であるスタック s に対して，$s \models \Pi$ であることと，Π が単射的であることは同値である．

SL_{LP}^{SH} のエンテイルメント判定問題 $\phi \vdash \psi$ は $\phi \models \psi$ か否かを判定する問題とする．

SL_{LP}^{SH} において，以下が成立する．

補題 2.1 ([4, 9]). 1. SL_{LP}^{SH} において，式 A, B, C について以下が成り立つ．

$$A * B \equiv B * A \qquad A * \mathtt{emp} \equiv A \qquad A * (B * C) \equiv (A * B) * C$$
$$A \circledast B \equiv B \circledast A \qquad A \circledast \mathtt{emp} \equiv A \qquad A \circledast (B \circledast C) \equiv (A \circledast B) \circledast C$$
$$A * B \models A \circledast B$$

2. （権限値の分割と合併）ラベル α，$\pi + \sigma \leqslant 1$ なる権限値 π, σ について以下が成り立つ．

$$\alpha^\pi \circledast \alpha^\sigma \equiv \alpha^{\pi+\sigma} \qquad (\circledast)$$

3. （ラベルの消去）すべてのスタック論理式 Π，@論理式 Ξ，ラベル・フリー論理式 Σ^-，ヒープ論理式 Σ とフレッシュなラベル α について以下が成り立つ．

$$\Pi \mid @_\alpha \Sigma^- \wedge \Xi \mid \alpha * \Sigma \models \Pi \mid \Xi \mid \Sigma^- * \Sigma$$

2.2 SL_{LP}^{SH} による並行プログラムの証明例

本節では前節までで紹介した SL_{LP}^{SH} による並行プログラムの証明例を示す．証明中では Lee らの推論規則 [9] を用いている．

以下の並行プログラム C がリスト断片 $\mathtt{ls}(x,y)$ を読み込むことを考える．

$$\mathtt{foo(x,y)} \parallel \mathtt{foo(x,y)}$$

ここで，foo は以下のように再帰的に定義される．

```
foo(x,y) {if (x == y) return;else {z = *x; foo(z,y)}}
```

foo は $\mathtt{ls}(x,y)$ を読み込むだけのプログラムであり，任意の権限値 π について

$$\{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^\pi\}\ \mathtt{foo(x,y)}\ \{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^\pi\}$$

が証明できる [9].

このとき，プログラム C に対して，

$$\{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha\}\ C\ \{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha\}$$

の証明アウトラインは図 2 のようになる．ここで，推論規則 (Parallel) のインスタンスは，

$$\frac{\{\phi\}\ \mathtt{foo}(x,y)\ \{\phi\} \quad \{\phi\}\ \mathtt{foo}(x,y)\ \{\phi\}}{\{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5} \circledast \alpha^{0.5}\}\ C\ \{\top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5} \circledast \alpha^{0.5}\}}\ (\text{Parallel})$$

($\phi = \top \mid @_\alpha \mathtt{ls}(x,y) \mid \alpha^{0.5}$) の形である．

2.3 エンテイルメント判定の概要

本節では，Lee らによる SL_{LP}^{SH} のエンテイルメント判定の概要 [9] を示す．手順としては，まず正規化と呼ばれる各ラベルの権限値を計算して式を正規形に変形する処理を行う．ここで正規形とは，$\circledast$ がなく，各ラベルが高々1 回しか現れない式の形のことをいう．その後，正規形の式からラベルの消去を行い，既存の判定器 [13] に還元する．

@論理式をスタック論理式と分離したことにより，Lee らによる表記 [9] と異なるが，本質的には同じものである．

2.3.1 正規化

まず，正規化によって各ラベルの権限値を計算する．

定義 2.11 (正規形)**.** 以下の形のヒープ論理式で，各ラベルが高々1 回しか出現しないものを正規形という．

$$\Sigma_{nf} ::= \alpha^{\pi} \mid \Sigma^{-} \mid \Sigma_{nf} * \Sigma_{nf}$$

定義 2.12. ラベル・フリー論理式上の関数 E を以下で定義する．

$$E(S) = S \qquad (S \text{ は原子ヒープ論理式})$$

$$E(\Sigma_1^- * \Sigma_2^-) = \begin{cases} E(\Sigma_2^-) & (E(\Sigma_1^-) = \mathtt{emp}) \\ E(\Sigma_1^-) & (E(\Sigma_1^-) \neq \mathtt{emp} \text{ かつ } E(\Sigma_2^-) = \mathtt{emp}) \\ E(\Sigma_1^-) * E(\Sigma_2^-) & (\text{それ以外}) \end{cases}$$

定義 2.13 (正規化)**.** シンボリック・ヒープの正規化関数 *normalization* を以下で定義する．

1. $n_s(\Sigma) \in \mathtt{Spat}$：

$$\begin{aligned} n_s(S) =& S \qquad (S \text{ が原子ヒープ論理式のとき}) \\ n_s(\Sigma_1 \circledast \Sigma_2) =& \ast_{i=1}^{n} {\alpha_i}^{min(\pi_i+\sigma_i,1)} * {\Sigma_1}' * {\Sigma_2}' \\ & (n_s(\Sigma_1) = \ast_{i=1}^{n} {\alpha_i}^{\pi_i} * {\Sigma_1}' \text{ かつ} \\ & n_s(\Sigma_2) = \ast_{i=1}^{n} {\alpha_i}^{\sigma_i} * {\Sigma_2}' \text{ かつ,} \\ & \{\alpha_i\}_{1 \leqslant i \leqslant n} = \mathrm{label}(\Sigma_1) \cap \mathrm{label}(\Sigma_2)) \\ n_s(\Sigma_1 * \Sigma_2) =& n_s(\Sigma_1 \circledast \Sigma_2) \end{aligned}$$

2. $n_E(\Sigma) \subseteq \mathtt{Label}$：

$$\begin{aligned} n_E(S) &= \varnothing \qquad (S \text{ が原子ヒープ論理式のとき}) \\ n_E(\Sigma_1 * \Sigma_2) &= (\mathrm{label}(\Sigma_1) \cap \mathrm{label}(\Sigma_2)) \cup n_E(\Sigma_1) \cup n_E(\Sigma_2) \\ &\quad \cup \{\alpha \mid n_s(\Sigma_1) = \alpha^{\pi} * {\Sigma_1}', n_s(\Sigma_2) = \alpha^{\sigma} * {\Sigma_2}', \pi + \sigma > 1\} \\ n_E(\Sigma_1 \circledast \Sigma_2) &= n_E(\Sigma_1) \cup n_E(\Sigma_2) \end{aligned}$$

3. $n_p(\Xi, L) \in \mathtt{Pure} \times \mathtt{At}$ for $L \subseteq \mathtt{Label}$：

$$\begin{aligned} n_p(\Xi, L) &= (\top, \Xi) \qquad (\mathrm{label}(\Xi) \cap L \neq \varnothing \text{ のとき}) \\ n_p(@_\alpha \Sigma^-, L) &= (\bigwedge_{i=1}^{n} s_i = t_i, @_\alpha \mathtt{emp}) \qquad (\alpha \in L, E(\Sigma^-) = \ast_{i=1}^{n} \mathtt{ls}(s_i, t_i) \text{ のとき}) \\ n_p(@_\alpha \Sigma^-, L) &= (\mathtt{nil} \neq \mathtt{nil}, @_\alpha \mathtt{emp}) \qquad (\alpha \in L, E(\Sigma^-) \text{ が} \mapsto \text{ を含むとき}) \\ n_p(\Xi_1 \wedge \Xi_2, L) &= ({\Pi_1}' \wedge {\Pi_2}', {\Xi_1}' \wedge {\Xi_2}') \quad (n_p(\Xi_i, L) = ({\Pi_i}', {\Xi_i}') \ (i = 1, 2) \text{ のとき}) \end{aligned}$$

4. シンボリック・ヒープの正規化:

$$normalization(\Pi \mid \Xi \mid \Sigma) = \Pi \wedge \Pi' \mid \Xi' \mid n_s(\Sigma) \qquad (n_p(\Xi, n_E(\Sigma)) = (\Pi', \Xi') \text{ のとき})$$

関数 n_s は各ラベルの権限値を計算し，$\circledast$ を消去する．論理式 A に対して $A_1 * A_2 * \cdots * A_n$ の意味で $*_{i=1}^{n} A_i$ を用いている．分離連言は共通部分を持たないことを意味するため，$\Sigma_1 * \Sigma_2$ において Σ_1, Σ_2 の両方にラベル α が現れる場合，α は空ヒープを表していなければならない．このようなラベルを関数 n_E で計算し，関数 n_p によってそれらのラベルが空ヒープを表していることを表す@論理式を追加している．もし，$\alpha \in n_E(\Sigma)$ であるのに，$@_\alpha \Sigma^-$ の Σ^- が空ヒープによって満たされない場合，その論理式は矛盾しているため，$\mathtt{nil} \neq \mathtt{nil}$ を追加している．

この正規化によって，もとのシンボリック・ヒープと論理同値な正規形が得られる [9].

エンテイルメントの両辺を正規化した後，出現する変数および nil の間の $=$ および $\neq$ に関する場合分けを行なう．具体的には，エンテイルメント $\Pi \mid \Xi \mid \Sigma \vdash \phi$ が異なる変数 x, y を含むとき，

$$x \neq y \wedge \Pi \mid \Xi \mid \Sigma \vdash \phi \qquad\qquad (\Pi \mid \Xi \mid \Sigma)[x := y] \vdash \phi[x := y]$$

の二つの場合に分け，左辺が充足可能なエンテイルメントに対してエンテイルメント判定手続を行なう.（左辺が充足不可能なエンテイルメントは自明に妥当である.）シンボリック・ヒープの充足可能性判定は，補題 2.1 の 3 の両辺は充足可能性同値なので，この素朴なラベル除去を行った後，既存の判定手法 [5] を用いることで決定することができる．

2.3.2 ラベル消去の規則と *Root* 関数

前節の正規化と場合分けの結果，左辺が単射的かつ正規形であるようなエンテイルメントが得られる．以下では，このようなエンテイルメントのラベルと権限値を消去する手続きを説明する．

素朴に補題 2.1 の 3 に基づいたラベルの消去を行った場合，エンテイルメントの妥当性が変わってしまうことがある．例えば，

$$\top \mid @_\alpha \mathtt{ls}(x, y) \mid \alpha * y \mapsto y \vdash \top \mid @_\alpha \mathtt{ls}(x, y) \mid \alpha$$

について，二つの単元ヒープ $h_1 = (s(x) \mapsto s(y))^{1.0}$，$h_2 = (s(y) \mapsto s(y))^{1.0}$ と $\rho(\alpha) = h_1$ となる ρ を考えると，$(s, h_1 \circ h_2, \rho)$ は左辺を満たすが右辺を満たさないため，このエンテイルメントは妥当ではない．しかしながら，補題 2.1 の 3 にしたがってラベルの消去を行うとエンテイルメントは

$$\top \mid \top \mid \mathtt{ls}(x, y) * y \mapsto y \vdash \top \mid \top \mid \mathtt{ls}(x, y)$$

となり，妥当となってしまう．

そこで，エンテイルメントの妥当性を変えずにラベルを消去するために，

$$\frac{\Pi \mid \top \mid \Sigma_1 \vdash \Pi' \mid \top \mid \Sigma_1' \qquad \Pi \mid \Xi \mid \Sigma_2 \vdash \Pi' \mid \Xi' \mid \Sigma_2'}{\Pi \mid @_\alpha \Sigma_1 \wedge \Xi \mid \alpha^\pi * \Sigma_2 \vdash \Pi' \mid @_\alpha \Sigma_1' \wedge \Xi' \mid \alpha^\pi * \Sigma_2'}$$

という推論規則を考える．この規則が妥当性を変えないこと，すなわち，上式の全てのエンテイルメントが妥当ならば下式のエンテイルメントが妥当であること（局所健全性）と，その逆（局所完全性）を示す必要があるが，局所完全性を示そうとすると，以下のような問題が生じる．まず，下式のエンテイルメントが妥当であることを仮定し，$s, h, \rho \models \Pi \mid \Xi \mid \Sigma_2$ となる任意のモデルをとる．この時，ヒープ h を $\Pi \mid @_\alpha \Sigma_1 \wedge \Xi \mid \alpha^\pi * \Sigma_2$ を満たすようなヒープ h' に拡張できれば，下式のエンテイルメントよりモデル h' が $\Pi' \mid @_\alpha \Sigma_1' \wedge \Xi' \mid \alpha^\pi * \Sigma_2'$ を満たすこと用いて s, h, ρ が $\Pi' \mid \Xi' \mid \Sigma_2'$ を満たすことがわかる．しかし，この h から h' への拡張は必ずできる訳ではない．例えば，Σ_1 が部分式として $x \mapsto t$ を含んでいるとき，$s(x) \in \mathrm{dom}(h)$ であるような h に対して，分離連言の性質より，上記のような h' への拡張はできない．

このため，Lee らは上記の推論規則を以下のように修正している [9].

定義 2.14. 変数の有限集合 X に対し，論理式 $X\uparrow$ の意味を，

$$s,h,\rho \models X\uparrow \Leftrightarrow \ \forall x \in X.s(x) \notin \mathrm{dom}(h)$$

で与える．$X\uparrow \mid \Pi \mid \Xi \mid \Sigma \vdash \phi$ の形の式を拡張エンテイルメントと呼ぶ.

以上の拡張を行った上で，ラベル消去の推論規則を以下のように修正する

$$\frac{\begin{array}{l}(X \cup \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_2))\uparrow \mid \Pi \mid \top \mid \Sigma_1 \vdash \Pi' \mid \top \mid {\Sigma_1}' \\ (X \cup \mathrm{Root}(\Pi \mid \top \mid \Sigma_1))\uparrow \mid \Pi \mid \Xi \mid \Sigma_2 \vdash \Pi' \mid \Xi' \mid \Sigma_2'\end{array}}{X\uparrow \mid \Pi \mid @_\alpha\Sigma_1 \wedge \Xi \mid \alpha^\pi * \Sigma_2 \vdash \Pi' \mid @_\alpha{\Sigma_1}' \wedge \Xi' \mid \alpha^\pi * {\Sigma_2}'}\ \text{(Label Equality Check)}$$

ここで，関数 Root は以下のように定義される.

定義 2.15 (Root[9]). 正規形のシンボリック・ヒープ ϕ に対して $\mathrm{Root}(\phi)$ を以下のように定義する.

$$\begin{aligned}\mathrm{Root}(\Pi \mid \Xi \mid \mathtt{emp}) &= \varnothing \\ \mathrm{Root}(\Pi \mid \Xi \mid x \mapsto y) &= \{x\} \\ \mathrm{Root}(\Pi \mid \Xi \mid \mathtt{ls}(x,y)) &= \{x\} \quad (x \neq y) \\ \mathrm{Root}(\Pi \mid \Xi \mid \mathtt{ls}(x,x)) &= \varnothing \\ \mathrm{Root}(\Pi \mid @_\alpha\Sigma^- \wedge \Xi \mid \alpha^\pi) &= \mathrm{Root}(\Pi \mid \top \mid \Sigma^-) \\ \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_1 * \Sigma_2) &= \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_1) \cup \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_2)\end{aligned}$$

$\mathrm{Root}(\phi)$ により表される変数集合は，直観的には「ϕ を満たすモデルにおいて必ず含まれていなければいけないアドレスを表す変数の集合」を表す．$\mathtt{ls}(x,x)$ は，空ヒープもモデルとして持つため，$\mathrm{Root}(\Pi \mid \Xi \mid \mathtt{ls}(x,x))$ は空集合となる． Root 関数については以下の性質が重要である.

補題 2.2 ([9],Lemma D.1.). $X\uparrow \mid \Pi \mid \Xi \mid \Sigma$ は単射的であるとする．このとき，単射であるような任意の s に対して，$s,h,\rho \models X\uparrow \mid \Pi \mid \Xi \mid \Sigma$ かつ $\mathrm{dom}(h) = \{s(x) \mid x \in \mathrm{Root}(\top \mid \top \mid \Sigma)\}$ を満たすような h,ρ が存在する.

Π が単射的であるとき，$X\uparrow \mid \Pi \mid \Xi \mid \Sigma$ は単射的であるという．ここで，$\mathrm{dom}(h) = \{s(x) \mid x \in \mathrm{Root}(\Sigma)\}$ となるような h が必ず存在することは，帰納的述語 $\mathtt{ls}$ 特有の性質であることに注意する.

推論規則 (Label Equality Check) について，以下が証明されている.

命題 2.1 ([9]). 推論規則 (Label Equality Check) は，左辺が単射的な拡張エンテイルメントに対して，局所健全かつ局所完全である.

正規形のエンテイルメントに対して (Label Equality Check) などの推論規則を下から上へ繰り返し適用すれば，ラベルも権限値もないエンテイルメントが得られ，これらのエンテイルメントを既存のアルゴリズム [13] によって判定することができる.

3 述語の拡張

前章の SL_{LP}^{SH}[9] では帰納的述語を $\mathtt{ls}$ のみに制限していた．本章では SL_{LP}^{SH} の帰納的述語を有界木幅条件 [7] を満たす述語に拡張する．通常の分離論理に対しては，有界木幅条件を満たす述語を含むシンボリック・ヒープのエンテイルメント判定は決定可能であることが知られている [7].

3.1 有界木幅条件を満たす帰納的述語

定義 3.1. $\mathtt{Pred} = \{P, Q, \dots\}$ を述語記号の集合とし，各 $P \in \mathtt{Pred}$ はアリティをもつとする．原子ヒープ論理式の定義を以下のように変更する．

$$S ::= \mathtt{emp} \mid t \mapsto (t_1, \dots, t_N) \mid P(x_1, \dots, x_n) \mid \alpha^{\pi} \qquad \text{(原子ヒープ論理式)}$$

$$\Sigma ::= S \mid \Sigma * \Sigma \mid \Sigma \circledast \Sigma \qquad \text{(ヒープ論理式)}$$

アリティ n の述語記号 P に対して，$P(x_1, \dots, x_n) ::= \exists z_1 \dots z_m(\Pi \mid \Sigma)$ の形式で，$\Pi \mid \Sigma$ は権限値もラベルも含まず，$FV(\Pi \mid \Sigma) - \{z_1, \dots, z_m\} \subseteq \{x_1, \dots, x_n\}$ を満たすものを P の定義節という．帰納的定義集合 Φ とは，定義節の有限集合であり，以下を満たすものとする．

> 任意の定義節 $P(x_1, \dots, x_n) ::= \exists z_1 \dots z_n(\Pi \mid \Sigma) \in \Phi$ と，Σ に現れる任意の Q について Φ は Q の定義節を含む．

帰納的定義集合 Φ と述語記号 P について，$\Phi|_P$ を Φ 中の P の定義節全体からなる集合とする．帰納的定義集合 Φ が，定義節として $P(\vec{x}) ::= \exists \vec{z_1} A_1 | \dots | \exists \vec{z_n} A_n$ を含むとは，$\Phi|_P = \{\exists \vec{z_i} A_i\}_{1 \leqslant i \leqslant n}$ であることをいう．また，帰納的定義集合 Φ により定義される述語を帰納的述語と呼ぶ．

Φ における帰納的述語の意味は，通常どおり最小不動点によって与える．

Iosif ら [7] はエンテイルメント判定が決定可能になるための十分条件として，有界木幅条件を提案した．帰納的定義集合 Φ に対する有界木幅条件は以下の 3 つからなる．

1. 進行性 (progress)：任意の定義節 $P(x, \vec{y}) ::= \exists \vec{z}(\Pi \mid \Sigma) \in \Phi$ について，Σ は $x \mapsto (t_1, \dots, t_N) * \Sigma'(\Sigma'$ は $\mapsto$ の出現を含まない) の形である．
2. 結合性 (connectivity)：任意の定義節 $P(x, \vec{y}) ::= \exists \vec{z}(\Pi \mid x_1 \mapsto (t_1 \dots, t_N) * \Sigma) \in \Phi$ と，Σ に出現する各述語 $Q(v, \vec{w})$ について，$v \in \{t_1, \dots, t_N\}$ である．
3. 確立性 (establishment)：任意の定義節 $P(x, \vec{y}) ::= \exists \vec{z}(\Pi \mid \Sigma) \in \Phi$ について，$\Pi \mid \Sigma$ を Φ の定義節に従って述語なしの状態になるまでどのように展開しても，すべての $z_i \in \vec{z}$ について $z_i \mapsto \vec{t'}$ が含まれている．

有界木幅条件を満たす帰納的述語によって以下に示すような，非空なリスト断片，双方向リスト，スキップリストや木などのより多くのデータ構造の記述が可能になる [13].

$$\text{非空リスト断片}: \mathtt{lsn}(x, y) ::= x \mapsto y \mid \exists z(x \mapsto (z) * \mathtt{lsn}(z, y))$$

$$\text{双方向リスト}: \mathtt{dll}(h, p, n, t) ::= h = t \wedge h \mapsto (p, n) \mid \exists z(h \mapsto (p, z) * \mathtt{dll}(z, h, n, t))$$

$$\text{スキップリスト}: \mathtt{skl1}(x, y) ::= x \mapsto (\mathtt{nil}, y) \mid \exists z(x \mapsto (\mathtt{nil}, z) * \mathtt{skl1}(z, y)),$$

$$\mathtt{skl2}(x, y) ::= \exists z(x \mapsto (y, z) * \mathtt{skl1}(z, y)$$

$$\mid \exists z_1 z_2(x \mapsto (z_1, z_2) * \mathtt{skl1}(z_2, z_1) * \mathtt{skl2}(z_1, y))$$

$$\text{木}: \mathtt{tree}(x) ::= (x \mapsto (\mathtt{nil}, \mathtt{nil})) \mid \exists lr.(x \mapsto (l, r) * \mathtt{tree}(l) * \mathtt{tree}(r))$$

オリジナルの SL_{LP}^{SH} における述語 $\mathtt{ls}$ は，空リストの場合も含むため，有界木幅条件を満たさないことに注意．しかし，$\mathtt{ls}(x, y) \equiv \mathtt{lsn}(x, y) \vee (x = y \mid \mathtt{emp})$ であるため，$\mathtt{lsn}$ のエンテイルメント判定に帰着できる．

以下では有界木幅条件を満たす帰納的述語に拡張した SL_{LP}^{SH} のエンテイルメント判定を行うことを考える．帰納的述語の拡張に応じて，2 章で述べた正規化とラベル消去に関する証明規則を拡張する．

正規化関数については n_p の変更のみが必要となる．有界木幅条件を満たす述語を満たすヒープ・モデルは必ず非空であるため，n_p の定義を以下のように変更すればよい．

3. $n_p(\Xi, L) \in \texttt{Pure} \times \texttt{At}$ for $L \subseteq \texttt{Label}$:

$$
\begin{aligned}
n_p(\Xi, L) &= (\top, \Xi) && (label_p(\Xi) \cap L \neq \varnothing \text{ のとき}) \\
n_p(@_\alpha \Sigma^-, L) &= (\texttt{nil} \neq \texttt{nil}, @_\alpha \texttt{emp}) && (\alpha \in L \text{ かつ } \Sigma^- \text{ が} \mapsto \text{または述語を含むとき}) \\
n_p(@_\alpha \Sigma^-, L) &= (\top, @_\alpha \texttt{emp}) && (\text{それ以外}) \\
n_p(\Xi_1 \wedge \Xi_2, L) &= ({\Pi_1}' \wedge {\Pi_2}', {\Xi_1}' \wedge {\Xi_2}') && (n_p(\Xi_i, L) = ({\Pi_i}', {\Xi_i}') \ (i = 1, 2))
\end{aligned}
$$

3.2 ラベル消去手続きの拡張

2 章で見たように，ラベル消去における推論規則の局所健全性，局所完全性の証明のために，Root 関数を導入した．帰納的述語 ls に対しては，補題 2.2 で示したように「ϕ のモデル (s, h, ρ) として，$s(\mathrm{Root}(\phi)) = \mathrm{dom}(h)$ となるようなものが取れる」という条件を満たす Root 関数が簡単に定義できたが，一般の帰納的述語に関してこのような Root 関数を定義することはできない．例えば，以下のような有界木幅条件を満たす帰納的述語 P を考える．

$$P(x, y, z) ::= x \mapsto y * y \mapsto \texttt{nil} \mid x \mapsto z * z \mapsto \texttt{nil}$$

$P(x, y, z)$ を満たすヒープでは変数 x, y または，x, z のいずれか一方が必ず割り当てられるため，上記のような条件を満たす $\mathrm{Root}(P(x, y, z))$ を一意に定めることができない．すなわち，補題 2.2 を満たすような Root 関数は存在しない．

$P(x, y, z)$ のように，そのヒープ・モデルが「必ず含まなければならないアドレス」として複数の場合が考えられるような状況に対応するために，$\mathrm{Root}(\phi)$ を変数集合の集合となるように拡張し，この場合分けを実現する．$\mathrm{Root}(\phi)$ は以下の条件を満たすことを仮定する．

> （条件 R）任意の単射的な正規形のシンボリック・ヒープ $\phi = \Pi \mid \Xi \mid \Sigma$ に対して，$Root(\phi) \in \mathcal{P}(\mathcal{P}(\texttt{Var}))$ は，以下の条件を満たす．
>
> 1. 任意の $X \in \mathrm{Root}(\phi)$，単射 s，および有限の $W \subseteq \texttt{Loc} - s(X)$ に対して，$s, h, \rho \models \phi$，かつ，$\mathrm{dom}(h) \cap W = \varnothing$ であるような h, ρ が存在する．
> 2. $s, h, \rho \models \phi$ ならば，$s(X) \subseteq \mathrm{dom}(h)$ であるような $X \in \mathrm{Root}(\phi)$ が存在する．

このような Root 関数が，一般の帰納的述語に対して定義できることは，次章で示す．

拡張した Root 関数を用いて，新たなラベル消去の規則 (Label equality check) を以下のように定義する．

$$\frac{\begin{array}{l}\{(X \cup S) \uparrow \mid \Pi \mid \top \mid \Sigma_1 \vdash \Pi' \mid \top \mid \Sigma_1'\}_{\substack{X \cap S = \varnothing \\ S \in \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_2)}} \\ \{(X \cup T) \uparrow \mid \Pi \mid \Xi \mid \Sigma_2 \vdash \Pi' \mid \Xi' \mid \Sigma_2'\}_{\substack{X \cap T = \varnothing \\ T \in \mathrm{Root}(\Pi \mid \top \mid \Sigma_1)}}\end{array}}{X \uparrow \mid \Pi \mid @_\alpha \Sigma_1 \wedge \Xi \mid \alpha^\pi * \Sigma_2 \vdash \Pi' \mid @_\alpha \Sigma_1' \wedge \Xi' \mid \alpha^\pi * \Sigma_2'}$$

拡張した (Label equality check) は，局所健全性と局所完全性を満たす．

命題 3.1. 左辺が単射的な拡張エンテイルメントに対して，(Label equality check) は局所健全かつ局所完全である．

Proof. はじめに局所健全性について示す．以下を仮定する．

1. $(X \cup S) \uparrow \mid \Pi \mid \top \mid \Sigma_1 \models \Pi' \mid \top \mid \Sigma_1'$ for each $S \in \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_2)$ s.t. $X \cap S = \varnothing$
2. $(X \cup T) \uparrow \mid \Pi \mid \Xi \mid \Sigma_2 \models \Pi' \mid \Xi' \mid \Sigma_2'$ for each $T \in \mathrm{Root}(\Pi \mid \top \mid \Sigma_1)$ s.t. $X \cap T = \varnothing$
3. $s, h, \rho \models X \uparrow \mid \Pi \mid @_\alpha \Sigma_1 \wedge \Xi \mid \alpha^\pi * \Sigma_2$ for injective s

ここでは,

$$s, h, \rho \models \Pi' \mid @_\alpha \Sigma'_1 \wedge \Xi' \mid \alpha^\pi * \Sigma'_2$$

を示せばよい．仮定 3 より以下を満たすような h_2 が存在する．

$$h = (\pi \bullet \rho(\alpha)) \circ h_2$$
$$s, \rho(\alpha), \rho \models X \uparrow \mid \Pi \mid \Xi \mid \Sigma_1 \quad \cdots (\star)$$
$$s, h_2, \rho \models X \uparrow \mid \Pi \mid \Xi \mid \Sigma_2$$

とくに，$(\star)$ より

$$s, \rho(\alpha), \rho \models X \uparrow \mid \Pi \mid \top \mid \Sigma_1$$

を得る．Root 関数に仮定する条件 R の 2. より，以下を満たす $T \in \mathrm{Root}(\Pi \mid \top \mid \Sigma_1)$ が存在する．

$$s(T) \subseteq \mathrm{dom}(\rho(\alpha))$$

$\rho(\alpha)$ と h_2 は互いに素であるから

$$s, h_2, \rho \models T \uparrow$$

を得る．また，$s, \rho(\alpha), \rho \models X \uparrow$ であるから

$$X \cap T = \varnothing$$

を得る．したがって,

ある $T \in \mathrm{Root}(\Pi \mid \top \mid \Sigma_1)$ に対して $s, h_2, \rho \models (X \cup T) \uparrow \mid \Pi \mid \Xi \mid \Sigma_2$ かつ $X \cap S = \varnothing$

を得る．同様に,

ある $S \in \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_2)$ に対して $s, \rho(\alpha), \rho \models (X \cup S) \uparrow \mid \Pi \mid \top \mid \Sigma_1$ かつ $X \cap S = \varnothing$

である．仮定 1，2 より

$$s, \rho(\alpha), \rho \models \Pi' \mid \top \mid \Sigma'_1$$
$$s, h_2, \rho \models \Pi' \mid \Xi' \mid \Sigma'_2$$

を得る．ここで，s, ρ は Ξ' を満たすため

$$s, \rho(\alpha), \rho \models \Pi' \mid \Xi' \mid \Sigma'_1$$

を得る．したがって,

$$s, (\pi \bullet \rho(\alpha)) \circ h_2, \rho \models \Pi' \mid @_\alpha \Sigma'_1 \wedge \Xi' \mid \alpha^\pi * \Sigma'_2$$

を得る．ここで，$(\pi \bullet \rho(\alpha)) \circ h_2 = h$ である．

つづいて局所完全性について示す．以下を仮定する．

4. $X \uparrow \mid \Pi \mid @_\alpha \Sigma_1 \wedge \Xi \mid \alpha^\pi * \Sigma_2 \models \Pi' \mid @_\alpha \Sigma'_1 \wedge \Xi' \mid \alpha^\pi * \Sigma'_2$

ただし，この左辺は充足可能であるとする．

はじめに,

5. 単射である s ，$X \cap S = \varnothing$ であるような $S \in \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_2)$ について，$s, h, \rho \models (X \cup S) \uparrow \mid \Pi \mid \top \mid \Sigma_1$

を仮定して,

$$s, h, \rho \models \Pi' \mid \top \mid \Sigma'_1$$

を示す．$W = \mathrm{dom}(h) \cup X$ とおくと，$s, h \models S \uparrow$ より，$W \subseteq \mathtt{Loc} - s(S)$ である．ここで，仮定 5 と Root 関数に仮定する条件 R の 1. より，以下を満たす h', ρ' が存在する．

$$s, h', \rho' \models X \uparrow \mid \Pi \mid \Xi \mid \Sigma_2$$
$$\mathrm{dom}(h') \cap W = \varnothing$$

とくに，$s, h' \models X\uparrow$，h と h' は互いに素である．
仮定 5 と Σ_1 がラベルを含まないことから，

$$s, h, \rho' \models X\uparrow \mid \Pi \mid \Xi \mid \Sigma_1$$

を得る．したがって，Σ_2 がラベル α を含まないことから，$\rho_1 = \rho'[\alpha \to h]$ とおくと，

$$s, (\pi \bullet \rho_1(\alpha)) \circ h', \rho_1 \models X\uparrow \mid \Pi \mid @_\alpha\Sigma_1 \wedge \Xi \mid \alpha^\pi * \Sigma_2$$

を得る．仮定 4 より

$$s, (\pi \bullet \rho_1(\alpha)) \circ h', \rho_1 \models \Pi' \mid @_\alpha\Sigma_1' \wedge \Xi' \mid \alpha^\pi * \Sigma_2'$$

を得る．したがって，

$$s, h, \rho_1 \models \Pi' \mid \Xi' \mid \Sigma_1'$$

を得る．Σ_1' がラベルを含まないことから，

$$s, h, \rho \models \Pi' \mid \top \mid \Sigma_1'$$

を得る．
次に

6. 単射である s について，$s, h, \rho \models (X \cup T)\uparrow \mid \Pi \mid \Xi \mid \Sigma_2$
7. $T \in \mathrm{Root}(\Pi \mid \top \mid \Sigma_1)$ かつ $X \cap T = \varnothing$

を仮定して，

$$s, h, \rho \models \Pi' \mid \Xi' \mid \Sigma_2'$$

を示す．$W' = \mathrm{dom}(h) \cup X$ とおくと，$s, h \models T\uparrow$ より，$W' \subseteq \mathtt{Loc} - s(T)$ である．ここで，仮定 6, 7 と Root 関数に仮定する条件 R の 1. より，以下を満たす h'' が存在する．

$$s, h'', \rho \models X\uparrow \mid \Pi \mid \top \mid \Sigma_1$$
$$\mathrm{dom}(h'') \cap W' = \varnothing$$

とくに，$s, h'' \models X\uparrow$，h と h'' は互いに素である．
ここで，$X\uparrow \mid \Pi \mid \top \mid \Sigma_1$ はラベルを含まないので，上式は任意の ρ で成立する．
仮定 6 より

$$s, h, \rho \models X\uparrow \mid \Pi \mid \Xi \mid \Sigma_2$$

を得る．Σ_2 がラベル α を含まないことから，$\rho_2 = \rho[\alpha \to h'']$ とおくと，

$$s, (\pi \bullet \rho_2(\alpha)) \circ h, \rho_2 \models X\uparrow \mid \Pi \mid @_\alpha\Sigma_1 \wedge \Xi \mid \alpha^\pi * \Sigma_2$$

を得る．仮定 4 より，

$$s, (\pi \bullet \rho_2(\alpha)) \circ h, \rho_2 \models \Pi' \mid @_\alpha\Sigma_1' \wedge \Xi' \mid \alpha^\pi * \Sigma_2'$$

を得る．したがって，

$$s, h, \rho_2 \models \Pi' \mid \Xi' \mid \Sigma_2'$$

を得る．$\Pi' \mid \Xi' \mid \Sigma_2'$ はラベル α を含まないので

$$s, h, \rho \models \Pi' \mid \Xi' \mid \Sigma_2'$$

を得る． □

また，Lee らの推論規則 (Empty label elimination 1)[9] についても以下のように変更する．

$$\frac{\{(X \cup S)\uparrow \mid \Pi \mid \Xi \mid \mathtt{emp} \vdash \Pi' \mid \Xi' \mid \Sigma_1\}_{\substack{X \cap S = \varnothing \\ S \in \mathrm{Root}(\Pi \mid \Xi \mid \Sigma_2)}} \qquad X\uparrow \mid \Pi \mid \Xi \mid \Sigma_2 \vdash \Pi' \mid \Xi' \mid \Sigma_2'}{X\uparrow \mid \Pi \mid @_\alpha\mathrm{Emp} \wedge \Xi \mid \alpha^\pi * \Sigma_2 \vdash \Pi' \mid @_\alpha\Sigma_1 \wedge \Xi' \mid \alpha^\sigma * \Sigma_2'}$$

ここで，Emp は，$\mathtt{emp}$ と $*$ のみで構成されるヒープ論理式とする．
拡張した (Empty label elimination 1) についても，局所健全性と局所完全性が成り立つ．

命題 3.2. 左辺が単射的な拡張エンテイルメントに対して，(Empty label elimination 1) は局所健全かつ局所完全である．

Proof. まず

$$\Pi \mid @_\alpha \mathrm{Emp} \wedge \Xi \mid \alpha^\pi * \Sigma \equiv \Pi \mid @_\alpha \mathrm{Emp} \wedge \Xi \mid \alpha^\sigma * \Sigma \quad \cdots (\star)$$

が任意の権限値 π, σ で成立することを示す．

$$\begin{aligned}
& s, h, \rho \models \Pi \mid @_\alpha \mathrm{Emp} \wedge \Xi \mid \alpha^\pi * \Sigma \\
\Leftrightarrow & s, \rho(\alpha), \rho \models \mathrm{Emp} \text{ かつ } s, h, \rho \models \Pi \mid \Xi \text{かつ} \\
& h = (\pi \bullet \rho(\alpha)) \circ h_1 \text{であるような } h_1 \text{が存在する．ただし,} s, h_1, \rho \models \Sigma \text{である．} \\
\Leftrightarrow & \rho(\alpha) = \varnothing \text{ かつ } s, h, \rho \models \Pi \mid \Xi \text{ かつ } s, h, \rho \models \Sigma
\end{aligned}$$

これは権限値 π に依らない．ゆえに $(\star)$ を得る．

ここで，$\mathrm{Root}(\Pi \mid \Xi \mid \mathtt{emp}) = \{\varnothing\}$ であることに注意して $(\star)$ を (Label equality check) に適用すると (Empty label elimination 1) を得る．(Label equality check) は局所完全かつ局所健全であるから，(Empty label elimination 1) も局所健全かつ局所完全である． □

以上の拡張により，Lee ら [9] と同様の方法で，有界木幅条件を満たす一般の帰納的述語を含む SL_{LP}^{SH} のエンテイルメントは，ラベル・フリーなエンテイルメントの判定問題に帰着できる．有界木幅条件を満たす帰納的述語を含む通常の分離論理のエンテイルメント判定問題は決定可能である [7] ので，以下が得られる．

定理 3.1 (帰納的述語を拡張した SL_{LP}^{SH} のエンテイルメント判定)**.** 有界木幅条件を満たす帰納的述語を含む SL_{LP}^{SH} のエンテイルメント判定問題は決定可能である．

4 Root 関数の計算

本章では，ラベル消去で用いる Root 関数を，帰納的述語定義から計算する方法を示す．Root 関数は，Brotherston が分離論理における充足可能性判定のために導入した base pair 集合を利用して計算することができる．有界木幅条件に限らず全ての帰納的述語に対して，base pair 集合は計算でき，以下で述べる性質が成立する．

命題 4.1 (base pair 集合 [5])**.** 帰納的定義集合 Φ に対して次のような $base^\Phi$ を計算することができる：Φ 中の各アリティ n の述語記号 P に対して

$$base^\Phi(P)(t_1, \ldots, t_n) \in \mathcal{P}(\mathcal{P}(\mathtt{Var}) \times \mathtt{Pure})$$

であり，次を満たす．

1. (Soundness) 任意の $(X, \Pi) \in base^\Phi(P)(t_1, \ldots, t_n)$，$s \models \Pi$，および有限集合 $W \subseteq \mathtt{Loc} - s(X)$ に対して，$s, h \models P(t_1, \ldots, t_n)$ かつ $\mathrm{dom}(h) \cap W = \varnothing$ を満たすような h が存在する．
2. (Completeness) $s, h \models P(t_1, \ldots, t_n)$ であるとき，$s(X) \subseteq \mathrm{dom}(h)$ かつ $s \models \Pi$ を満たすような $(X, \Pi) \in base^\Phi(P)(t_1, \ldots, t_n)$ が存在する．

これを用いて，帰納的述語集合 Φ 上のシンボリック・ヒープに対する Root 関数を定義する．まず，準備として素朴なラベル除去関数を定義する．

定義 4.1 (NLE(ϕ))**.** 正規形のシンボリック・ヒープ間の関係 $\to_{\mathrm{NLE}}$ を

$$\frac{}{\Pi \mid @_\alpha \Sigma^- \mid \alpha^\pi * \Sigma \ \to_{\mathrm{NLE}}\ \Pi \mid \top \mid \Sigma^- * \Sigma}$$

$$\frac{\Pi \mid \Xi \mid \Sigma \to_{\mathrm{NLE}} \Pi \mid \Xi' \mid \Sigma'}{\Pi \mid \Xi \wedge \Xi'' \mid \Sigma \to_{\mathrm{NLE}} \Pi \mid \Xi' \wedge \Xi'' \mid \Sigma'} \qquad \frac{\Pi \mid \Xi \mid \Sigma \to_{\mathrm{NLE}} \Pi \mid \Xi' \mid \Sigma'}{\Pi \mid \Xi'' \wedge \Xi \mid \Sigma \to_{\mathrm{NLE}} \Pi \mid \Xi'' \wedge \Xi' \mid \Sigma'}$$

を満たす最小の関係の反射推移閉包とする．正規形のシンボリック・ヒープ ϕ に対して，NLE(ϕ) を，「$\phi \to_{\mathrm{NLE}} \phi'$，かつ，$\phi'$ はラベルを含まない」を満たす ϕ' とする．

正規形のシンボリック・ヒープの定義より，NLE(ϕ) は必ず一意的に存在し，それは $\Pi \mid \top \mid \Sigma$ の形である．NLE に関して以下が成り立つ．

補題 4.1. 任意の $L \subseteq \mathtt{Loc}$ と s に対して，以下は同値である．

1. $\exists h, \exists \rho.\ (s, h, \rho \models \phi$，かつ $\mathrm{dom}(h) = L)$
2. $\exists h.\ (s, h \models \mathrm{NLE}(\phi)$，かつ $\mathrm{dom}(h) = L)$

Proof. 次を示せばよい．任意の $L \subseteq \mathtt{Loc}$ と s に対して，以下は同値である．

1. $\exists h, \exists \rho.\ (s, h, \rho \models \Pi \mid @_\alpha \Sigma^- \wedge \Xi \mid \alpha^\pi * \Sigma$，かつ $\mathrm{dom}(h) = L)$
2. $\exists h, \exists \rho.\ (s, h, \rho \models \Pi \mid \Xi \mid \Sigma^- * \Sigma$，かつ $\mathrm{dom}(h) = L)$

(1⇒2) $s, h, \rho \models \Pi \mid @_\alpha \Sigma_1^- \wedge \Xi \mid \alpha^\pi * \Sigma_2$ とする．定義より，次を満たす h_2 が存在する．

- $h = (\pi \bullet \rho(\alpha)) \circ h_2$
- $s, \rho(\alpha), \rho \models \Pi \mid \Xi \mid \Sigma_1^-$
- $s, h_2, \rho \models \Pi \mid \Xi \mid \Sigma_2$

よって，$s, \rho(\alpha) \circ h_2, \rho \models \Pi \mid \Xi \mid \Sigma_1^- * \Sigma_2$．ここで，$\mathrm{dom}(\rho(\alpha) \circ h_2) = \mathrm{dom}((\pi \bullet \rho(\alpha)) \circ h_2) = \mathrm{dom}(h)$ である．

(2⇒1) $s, h, \rho \models \Pi \mid \Xi \mid \Sigma_1 * \Sigma_2$ とする．定義より，次を満たす h_1, h_2 が存在する．

- $h = h_1 \circ h_2$
- $s, h_i, \rho \models \Pi \mid \Xi \mid \Sigma_i\ (i = 1, 2)$

よって，$\rho' = \rho[\alpha \to h_1]$ とすると，α は Σ_1, Σ_2 に現れないことより，$s, h_i, \rho' \models \Pi \mid \Xi \mid \Sigma_i$ であり，よって $s, (\pi \bullet \rho'(\alpha)) \circ h_2, \rho' \models \Pi \mid @_\alpha \Sigma_1^- \wedge \Xi \mid \alpha^\pi * \Sigma_2$ である．ここで，$\mathrm{dom}((\pi \bullet \rho'(\alpha)) \circ h_2) = \mathrm{dom}(h_1 \circ h_2) = \mathrm{dom}(h)$ である． □

定義 4.2 (Root)**.** Φ を帰納的定義集合，$\phi = \Pi \mid \Xi \mid \Sigma$ を Φ 上の単射的かつ正規形のシンボリック・ヒープとする．P_ϕ をフレッシュな述語記号とし，Φ_ϕ を Φ に帰納的定義

$$P_\phi(x_1, \ldots, x_n) ::= \Pi \mid \Sigma' \qquad (\mathrm{NLE}(\phi) = \Pi \mid \top \mid \Sigma'，\text{かつ } FV(\Pi \mid \Sigma') = \{x_1, \ldots, x_n\})$$

を追加した帰納的定義集合とする．このとき，

$$\mathrm{Root}(\phi) = \{X \mid (X, \Pi') \in \mathit{base}^{\Phi_\phi}(P_\phi)(x_1, \ldots, x_n)，\text{かつ } \Pi' \text{は単射的}\}$$

と定義する．

この Root 関数は，前章で挙げた条件 R を満たす．

命題 4.2. $\phi = \Pi \mid \Xi \mid \Sigma$ は単射的かつ正規形のシンボリック・ヒープとする．

1. 任意の $X \in \mathrm{Root}(\phi)$ と，単射である s，および $W \subseteq \mathtt{Loc} - s(X)$ に対して，$s, h, \rho \models \phi$，かつ $\mathrm{dom}(h) \cap W = \varnothing$ であるような h, ρ が存在する．
2. $s, h, \rho \models \phi$ ならば，$s(X) \subseteq \mathrm{dom}(h)$ であるような $X \in \mathrm{Root}(\phi)$ が存在する．

Proof. 1. $X \in \mathrm{Root}(\phi)$，$s$ は単射，$W \subseteq \mathtt{Loc} - s(X)$ とする．Root の定義より，ある単射的な Π' があって，$(X, \Pi') \in \mathit{base}^{\Phi_\phi}(P_\phi)(x_1, \ldots, x_n)$．$s$ は単射であるから $s \models \Pi'$．base^Φ の健全性 (命題 4.1 の 1) より，$s, h \models P_\phi(x_1, \ldots, x_n)$，かつ $\mathrm{dom}(h) \cap W = \varnothing$ を満たす h が存在する．P_ϕ の帰納的定義と補題 4.1 より，

$$\begin{aligned} s, h \models P_\phi(x_1, \ldots, x_n) &\Leftrightarrow s, h \models \mathrm{NLE}(\phi) \\ &\Leftrightarrow \exists h' \rho.(s, h', \rho \models \phi \text{ かつ } \mathrm{dom}(h') = \mathrm{dom}(h)) \end{aligned}$$

2. $s, h, \rho \models \phi$ とすると，補題 4.1 より，ある h' が存在して，$s, h' \models \mathrm{NLE}(\phi)$，かつ $\mathrm{dom}(h') = \mathrm{dom}(h)$．$P_\phi$ の帰納的定義より，$s, h' \models P_\phi(x_1, \ldots, x_n)$．$\mathit{base}^\Phi$ の完全性 (命題 4.1 の 2) より，$s(X) \subseteq \mathrm{dom}(h')$ かつ $s \models \Pi'$ を満たすような $(X, \Pi') \in \mathit{base}^{\Phi_\phi}(P_\phi)(x_1, \ldots, x_n)$ が存在する．とくに，s は単射より，Π' は単射的であり，$X \in \mathrm{Root}(\phi)$ である． □

5 おわりに

本論文では，分割可能なアクセス権限値とラベルを含む分離論理 SL_{LP}^{SH}[9] の帰納的述語を有界木幅条件 [7] を満たす一般の帰納的述語に拡張しても，エンテイルメント判定問題が決定可能であることを示した．とくに，既存のアルゴリズムにおけるラベル消去のステップで必要である Root 関数を，Brotherston ら [5] の base pair 集合のアイディアを用いることで計算できることを示し，一般化された Root 関数を用いてラベル消去の推論規則を拡張し，この局所健全性，および局所完全性を証明した．

オリジナルの SL_{LP}[4] では，スレッド間の権限値の受け渡しを分離含意を用いて表現しているが，権限値などを含まない通常の分離論理においても，分離含意を含むエンテイルメント判定は難しいことが知られている．権限値をもつ並行分離論理において分離含意を含むエンテイルメント判定問題の決定可能性を明らかにすることは今後の課題である．通常の分離論理に対しては，論理式をガード付き論理式と呼ばれる形に制限することによってエンテイルメント判定が決定可能になることが知られており [10]，SL_{LP} においても同様な制限によって決定可能なエンテイルメント判定が得られることが期待できる．

参考文献

[1] Antonopoulos, T., Gorogiannis, N., Haase, C., Kanovich, M., and Ouaknine, J.: Foundations for decision problems in separation logic with general inductive predicates, *International Conference on Foundations of Software Science and Computation Structures*, Springer, 2014, pp. 411–425.

[2] Berdine, J., Calcagno, C., and O'Hearn, P.: Symbolic execution with separation logic, *Programming Languages and Systems (APLAS 2005)*, Lecture Notes in Computer Science, Vol. 3780, Springer, 2005, pp. 52–68.

[3] Brookes, S.: A semantics for concurrent separation logic, *Theoretical Computer Science*, Vol. 375, No. 1-3(2007), pp. 227–270.

[4] Brotherston, J., Costa, D., Hobor, A., and Wickerson, J.: Reasoning over Permissions Regions in Concurrent Separation Logic, *International Conference on Computer Aided Verification*, Springer, 2020, pp. 203–224.

[5] Brotherston, J., Fuhs, C., Pérez, J. A. N., and Gorogiannis, N.: A decision procedure for satisfiability in separation logic with inductive predicates, *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2014, pp. 1–10.

[6] Hoare, C. A. R.: An axiomatic basis for computer programming, *Communications of the ACM*, Vol. 12, No. 10(1969), pp. 576–580.

[7] Iosif, R., Rogalewicz, A., and Simacek, J.: The tree width of separation logic with recursive definitions, *International Conference on Automated Deduction*, Springer, 2013, pp. 21–38.

[8] Katelaan, J., Matheja, C., and Zuleger, F.: Effective entailment checking for separation logic with inductive definitions, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2019)*, Lecture Notes in Computer Science, Vol. 11428, Springer, 2019, pp. 319–336.

[9] Lee, Y. and Nakazawa, K.: Decidable entailment checking for concurrent separation logic with fractional permissions, コンピュータソフトウェア, Vol. 40, No. 4(2023), pp. 4_67–4_86.

[10] Matheja, C., Pagel, J., and Zuleger, F.: A Decision Procedure for Guarded Separation Logic, *ACM Transactions Computational Logic*, Vol. 24, No. 1(2023), pp. 1–76.

[11] O'Hearn, P. W.: Resources, concurrency, and local reasoning, *Theoretical computer science*, Vol. 375, No. 1(2007), pp. 271–307.

[12] Reynolds, J. C.: Separation logic: A logic for shared mutable data structures, *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*, IEEE, 2002, pp. 55–74.

[13] Tatsuta, M., Nakazawa, K., and Kimura, D.: Completeness of cyclic proofs for symbolic heaps with inductive definitions, *Asian Symposium on Programming Languages and Systems*, Springer, 2019, pp. 367–387.