JSSST2021 3 日目 PPL(4-1)[45-L]

ハイパーグラフ書き換え系への構文駆動で compositional な構文・意味論の提案

2021/09/03

日本ソフトウェア科学会第 38 回大会

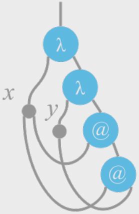
早稲田大学 基幹理工学研究科 情報理工・情報通信専攻

佐野仁　上田和紀

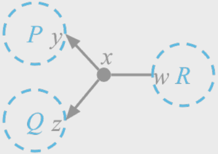
背景｜ハイパーグラフ

λ 計算の項，並行プロセスの状態，ポインタなど，
コンピュータサイエンスに現れる多くのものは
ハイパーグラフ[1] でモデル化できる

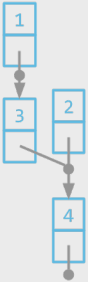
- 辺がそれぞれ 1 つ以上の頂点を結ぶことができるようにグラフを拡張したもの



$\lambda x.\lambda y.x(xy)$



$x(y).P \mid x(z).Q \mid \overline{x}\langle w\rangle.R$



従って，ハイパーグラフを第一級に扱う簡明な計算モデルがあるのが望ましい

[1] Berge Claude. Hypergraphs: Combinatorics of Finite Sets. North-Holland, 1989.

背景｜グラフ書き換え系

（ハイパー）グラフを第一級に扱うパラダイムとして
（ハイパー）グラフ書き換え系[2][3] がある

構文駆動でない意味論（次スライドから説明）を持つプログラミング言語・ツール：

- GP 2[4]， Dactl[5], GrGen[6], . . .

構文駆動な意味論を持つ計算モデルかつプログラミング言語：

- LMNtal (elemental)[7]

[4] Christopher Bak. GP 2: Efficient Implementation of a Graph Programming Language. PhD thesis. University of York, Sept. 2015.
[5] J. R. W. Glauert et al. Dactl - an Experimental Graph Rewriting Language. In: *Proc. Graph Grammars 1990*. Vol. 532. LNCS. Springer, 1991, pp. 378–395.
[6] Rubino Geiß et al. GrGen: A Fast SPO-Based Graph Rewriting Tool. In: *Proc. ICGT 2006*. Springer, 2006, pp. 383–397.
[7] Kazunori Ueda. LMNtal as a hierarchical logic programming language. In: **Theoretical Computer Science** 410.46 (2009), pp. 4784–4800.

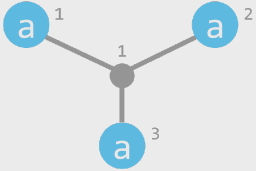
[2] H. Ehrig et al. Fundamentals of Algebraic Graph Transformation. Springer, 2006.
[3] Grzegorz Rozenberg. Handbook of Graph Grammars and Computing by Graph Transformation. World Scientific, 1997.

既存のハイパーグラフ書き換え系の意味論の問題点 $\langle 1/2 \rangle$

一般に，ハイパーグラフ書換え系の意味論では，**ハイパーグラフ** を
頂点集合，辺集合，頂点から辺への対応，ラベリング関数 などの組で定義する[2][3]

- e.g.

$$\left\langle \begin{array}{ll} \text{頂点集合：} & \{1,2,3\}, \\ \text{辺集合：} & \{1\}, \\ \text{頂点から辺への対応：} & \{1 \mapsto 1, 1 \mapsto 2, 1 \mapsto 3\}, \\ \text{ラベリング関数：} & \{1 \mapsto a, 2 \mapsto a, 3 \mapsto a\} \end{array} \right\rangle$$



これは compositional でない

→ サブグラフに分割 ⇄ 複数のサブグラフを合成

のような行き来ができない（とてもしづらい）

[2] H. Ehrig et al. Fundamentals of Algebraic Graph Transformation. Springer, 2006.
[3] Grzegorz Rozenberg. Handbook of Graph Grammars and Computing by Graph Transformation. World Scientific, 1997.

既存のハイパーグラフ書き換え系の意味論の問題点 ⟨2/2⟩

一般に，ハイパーグラフ書換え系の意味論では，

サブグラフへのマッチングや生成 は

頂点集合などの組で定義されたハイパーグラフ への **射** を用いて定義される

- double/single-pushout （これらは圏論の用語）

これは構文駆動でない

→ 構文要素から直接的に操作的意味論が定義できておらず，
高度な数学的素養を要求・実際のプログラムとの対応が取りづらい

↔ λ 計算，π 計算[8]，IMP などの 広く普及している意味論はどれも構文駆動

[8] Davide Sangiorgi et al. The Pi-Calculus: A Theory of Mobile Processes. Cambridge University Press, 2001.

研究の背景と目的

背景 ハイパーグラフ書き換え系の意味論は
一般に構文駆動でない
→（圏論などを深く理解している人でないと）難しい

目的 λ 計算，π 計算などのような
構文駆動な意味論 を持つ計算モデルを提案する

ハイパーグラフ書き換え系が備えるべき機能｜辺の融合

頂点の多重集合へのマッチングと書き換えができれば，それだけで良い？

↔ Dactl[5] は，Matching，Building（書き換え後のサブグラフ生成）の後に

Redirection

他の頂点が持つ参照をあるアドレスの方へ向け直す

→ 無向グラフでは，ある辺を他の辺と融合させる ことに対応する

を行う

- Redirection はコンピュータサイエンスにおいて普遍的に見られる概念

→ ハイパーリンクの融合も精密に形式化したい

[5] J. R. W. Glauert et al. Dactl - an Experimental Graph Rewriting Language. In: *Proc. Graph Grammars 1990*. Vol. 532. LNCS. Springer, 1991, pp. 378–395.

関連研究 | プロセス代数

並行プロセス を扱う
構文駆動な計算モデル

CSP, CCS, π 計算[8], ...,
Fusion calculus[9] →

[8] Davide Sangiorgi et al. The Pi-Calculus: A Theory of Mobile Processes. Cambridge University Press, 2001.
[9] J. Parrow et al. The fusion calculus: expressiveness and symmetry in mobile processes. In: *Proc. LICS 1998*. 1998, pp. 176–185.

本研究ではプロセス代数における「名前の隠蔽（スコープ）」の構文・意味論をグラフ書き換え系に新たに導入した

Fusion calculus

π 計算の派生
チャネルを受信する際に

- 受信したチャネルを局所的に束縛するのではなく，
- 大域的にそれらのチャネルを同じものとする，

チャネルの融合（fusion）を行う
fusion はハイパーリンクの融合と直に対応する概念

関連研究｜階層グラフ書き換え言語 LMNtal (elemental)[7]

階層グラフ書き換えに基づく
構文駆動な計算モデルかつプログラミング言語

本研究では階層化の機能を省いた Flat LMNtal をベースに，
拡張する形でハイパーグラフ書き換え系を形式化した

ただし，LMNtal におけるグラフの辺（リンク）はハイパーリンクではなく，
それを前提とした意味論になっている

→ ハイパーリンクへの拡張のためには，根本から手を入れる必要がある

[7] Kazunori Ueda. LMNtal as a hierarchical logic programming language. In: **Theoretical Computer Science** 410.46 (2009), pp. 4784–4800.

関連研究 | HyperLMNtal[14]

ハイパーリンクの機能を提供する，LMNtal の 実装上の拡張 （2012 〜）

アプリケーション

λ 計算の Encoding[10]，第一級書き換え規則・メタインタプリタ[11]，
Capability Type Checking[12]，G-Machine の実装[13]，…

[10] A. Yasen et al. Hypergraph Representation of Lambda-Terms. In: *TASE 2016*. IEEE Computer Society, July 2016, pp. 113–116.
[11] Yutaro Tsunekawa et al. Implementation of LMNtal Model Checkers: A Metaprogramming Approach. In: **Journal of Object Technology** 17.1 (Nov. 2018).
[12] Stefan Walter. Capability Typing for HyperLMNtal. Master's Thesis. Waseda University, 2021.
[13] Jin Sano. Implementing G-Machine in HyperLMNtal. https://arxiv.org/abs/2103.14698. Bachelor's Thesis. Waseda University, 2021.

文献 [14] に，非形式的に仕様が述べられているだけに留まる

→ 構文駆動な計算モデルとはなっていない

[14] Kazunori Ueda et al. HyperLMNtal: An Extension of a Hierarchical Graph Rewriting Model. In: **Künstliche Intelligenz** 26.1 (2012), pp. 27–36.

本研究の成果

構文駆動なハイパーグラフ書き換え系の
ミニマムな形式的定義として，
Flat HyperLMNtal を提案する

(Process)

$P ::= \mathbf{0}$		Null
$\mid p(X_1, \ldots, X_m)$		$m \geq 0$, Atom
$\mid (P, P)$		Molecule
$\mid \nu X.P$		Hyperlink creation
$\mid (P \mathrel{:\!-} P)$		Rule

$$fn(\mathbf{0}) = \emptyset$$
$$fn(p(X_1, \ldots, X_m)) = \bigcup_{i=1}^{m} \{X_i\}$$
$$fn((P, Q)) = fn(P) \cup fn(Q)$$
$$fn(\nu X.P) = fn(P) \setminus \{X\}$$
$$fn((P \mathrel{:\!-} Q)) = \emptyset$$

(E1) $(\mathbf{0}, P) \equiv P$
(E2) $(P, Q) \equiv (Q, P)$
(E3) $(P, (Q, R)) \equiv ((P, Q), R)$
(E4) $P \equiv P' \Rightarrow (P, Q) \equiv (P', Q)$
(E5) $P \equiv Q \Rightarrow \nu X.P \equiv \nu X.Q$
(E6) $\nu X.(X \bowtie Y, P) \equiv \nu X.P\langle Y/X \rangle$
（ただし $X \in fn(P) \vee Y \in fn(P)$）
(E7) $\nu X.\nu Y.X \bowtie Y \equiv \mathbf{0}$
(E8) $\nu X.\mathbf{0} \equiv \mathbf{0}$
(E9) $\nu X.\nu Y.P \equiv \nu Y.\nu X.P$
(E10) $\nu X.(P, Q) \equiv (\nu X.P, Q)$
（ただし $X \notin fn(Q)$）

(R1) $\dfrac{P \longrightarrow P'}{(P, Q) \longrightarrow (P', Q)}$

(R2) $\dfrac{P \longrightarrow Q}{\nu X.P \longrightarrow \nu X.Q}$

(R3) $\dfrac{Q \equiv P \quad P \longrightarrow P' \quad P' \equiv Q'}{Q \longrightarrow Q'}$

(R4) $(P, (P \mathrel{:\!-} Q)) \longrightarrow (Q, (P \mathrel{:\!-} Q))$

1. 研究の背景
2. **Flat HyperLMNtal の抽象構文**
3. Flat HyperLMNtal の操作的意味論
4. Flat HyperLMNtal に関して行なった証明
5. まとめ

構文要素｜識別子と予約名

- X は（ハイパー）リンク名を表す
- p はアトム名（頂点のラベル）を表す
  - $\bowtie$ のみ予約名
  - アトム $X \bowtie Y$ は **fusion** と呼び，直感的には引数の ハイパーリンク同士を繋げる 機能を持つ

Flat HyperLMNtal の抽象構文

(Process)

$P \quad ::= \quad \mathbf{0}$ Null

グラフも書き換え規則も何もない状態

$\mid \quad p(X_1, \ldots, X_m) \quad m \geq 0$ Atom

グラフのラベル付き頂点 →

$\mid \quad (P, P)$ Molecule

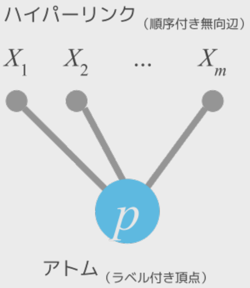
頂点や書き換え規則の多重集合

$\mid \quad \nu X.P$ **Hyperlink creation**

ハイパーリンクのスコープ
プロセス代数から新たに導入

$\mid \quad (P \text{ :- } P)$ Rule

サブグラフの書き換え規則



自由ハイパーリンク

プロセス P における自由ハイパーリンクの集合を $fn(P)$ で表し，以下のように帰納的に定義する

プロセス代数から新たに導入

$$
\begin{aligned}
fn(\mathbf{0}) &= \emptyset \\
fn(p(X_1, \dots, X_m)) &= \bigcup_{i=1}^{m} \{X_i\} \\
fn((P, Q)) &= fn(P) \cup fn(Q) \\
fn(\nu X.P) &= fn(P) \setminus \{X\} \\
fn((P :\text{-}\ Q)) &= \emptyset
\end{aligned}
$$

Flat HyperLMNtal の構文条件

ルール $(P \mathrel{:\!-} Q)$ は以下に示す条件を満たす必要がある.

1. ルールは P に出現してはならない（LMNtal と同様）
2. $fn(P) \supseteq fn(Q)$ （**今回新しく導入**）

直感的には,

「右辺で新しく出現するハイパーリンクは

ν で宣言してやる必要がある」と読める

1. 研究の背景

2. Flat HyperLMNtal の抽象構文

3. Flat HyperLMNtal の操作的意味論

4. Flat HyperLMNtal に関して行なった証明

5. まとめ

操作的意味論｜構造合同規則

グラフを扱うためには，抽象構文木 $+\ \alpha$ 変換だけでは不足

- 構文的には異なっていても，グラフとしては等しいものもある
- e.g. $(a(X), b(X))$ と $(b(X), a(X))$

構造合同規則 は，いくつかの公理でこれらの等価性を定義する

- e.g. $(P, Q) \equiv (Q, P)$
- プロセス代数において，普遍的に用いられている
- LMNtal の意味論においても採用されている

操作的意味論のための準備｜ハイパーリンク代入

$P\langle Y/X\rangle$ はハイパーリンク代入であり，
全ての自由出現する X を Y で置き換える
プロセス代数から新たに導入

事前に α 変換が必要になる可能性がある

- ν（名前の隠蔽）が存在するため

↔ LMNtal では ν などを用いずに
自由リンクを管理する
(後で詳しく解説)
ため，事前の α 変換は不要だった

$$\mathbf{0}\langle Y/X\rangle \stackrel{\text{def}}{=} \mathbf{0}$$

$$p(X_1,\dots,X_m)\langle Z/Y\rangle \stackrel{\text{def}}{=} p(X_1\langle Z/Y\rangle,\dots,X_m\langle Z/Y\rangle)$$

ただし $X_i\langle Z/Y\rangle \stackrel{\text{def}}{=} \begin{cases} Z & \text{if } X_i = Y \\ X_i & \text{if } X_i \neq Y \end{cases}$

$$(P,Q)\langle Y/X\rangle \stackrel{\text{def}}{=} (P\langle Y/X\rangle, Q\langle Y/X\rangle)$$

$$(\nu X.P)\langle Z/Y\rangle \stackrel{\text{def}}{=} \begin{cases} \nu X.P & \text{if } X = Y \\ \nu X.P\langle Z/Y\rangle & \text{if } X \neq Y \wedge X \neq Z \\ \nu W.(P\langle W/X\rangle)\langle Z/Y\rangle & \text{if } X \neq Y \wedge X = Z \\ & \wedge W \notin fn(P) \wedge W \neq Z \end{cases}$$

$$(P \mathbin{:\!-} Q)\langle Y/X\rangle \stackrel{\text{def}}{=} (P \mathbin{:\!-} Q)$$

Flat HyperLMNtal の操作的意味論｜構造合同規則

LMNtal と同様

(E1)	$(\mathbf{0}, P)$	$\equiv$	P
(E2)	(P, Q)	$\equiv$	(Q, P)
(E3)	$(P, (Q, R))$	$\equiv$	$((P, Q), R)$
(E4)	$P \equiv P'$	$\Rightarrow$	$(P, Q) \equiv (P', Q)$

プロセス代数から新たに導入

(E5)	$P \equiv Q$	$\Rightarrow$	$\nu X.P \equiv \nu X.Q$

fusion のための規則

（本研究の核心部分）

(E6)	$\nu X.(X \bowtie Y, P)$	$\equiv$	$\nu X.P[Y/X]$
	（ただし $X \in \mathit{fn}(P) \vee Y \in \mathit{fn}(P)$）		
(E7)	$\nu X.\nu Y.X \bowtie Y$	$\equiv$	$\mathbf{0}$

プロセス代数から新たに導入

(E8)	$\nu X.\mathbf{0}$	$\equiv$	$\mathbf{0}$
(E9)	$\nu X.\nu Y.P$	$\equiv$	$\nu Y.\nu X.P$
(E10)	$\nu X.(P, Q)$	$\equiv$	$(\nu X.P, Q)$
	（ただし $X \notin \mathit{fn}(Q)$）		

LMNtal では … | リンクの結合

LMNtal では，特殊なアトム（connector） $X = Y$ が以下の構造合同規則によって， X と Y を結合する

$$(X = Y, P) \;\equiv\; P[Y/X]$$

（ただし X は P の **自由リンク**（次スライド））[1]

$P[Y/X]$ は P に出現する全てのリンク X を Y で置き換えるリンク代入

LMNtal では，さらに「P はアトム」という条件があるが，flat LMNtal においては重要でないので割愛した

LMNtal では … | 自由リンク・局所リンク

LMNtal の自由・局所リンク

- P に 1 回出現：P の自由リンク
- P に 2 回出現：P の局所リンク

LMNtal のリンクはハイパーリンクではない

→ 高々 2 回出現する

→ 出現回数を数えるだけで，リンクの局所性がわかる

↔ ハイパーリンクは出現回数に制限がないため，
このように見分けることはできない．

ハイパーリンクの融合の素朴な定義

ハイパーリンクは（ν がなければ）自由・局所リンクか見分けられない

→ 自由リンクに関する付帯条件をなくす？(実はダメ)

ハイパーリンクの融合を

ハイパーリンク融合規則 ver. 1

$(X \bowtie Y, P) \equiv P\langle Y / X\rangle$ （付帯条件なし）

のように定義してしまうと,
グラフ書き換え系として意味をなさなくなる（次スライド）

ハイパーリンクの融合の素朴な定義の問題点

再掲：ハイパーリンク融合規則 ver. 1

$(X \bowtie Y, P) \equiv P\langle Y/X \rangle$ （付帯条件なし）

とすると…

$$\begin{aligned}
&(a(X), (X \bowtie Y, b(X, Y))) &&\equiv_{\text{(E2, E3)}} (X \bowtie Y, (a(X), b(X, Y))) \\
&\equiv_{\text{(E4), 融合規則 ver. 1}} \underbrace{(a(X), b(Y, Y))}_{X\text{ と }Y\text{ が出現}} &&\equiv_{\text{融合規則 ver. 1}} \underbrace{(a(Y), b(Y, Y))}_{\text{全て }Y\text{ に変換された}}
\end{aligned}$$

「プログラムに出現する任意のハイパーリンクが等しい」ことが導出可能

→ グラフ書き換え系としての意味をなさない

LMNtal では … | リンクの結合におけるリンクの局所性

再掲：LMNtal のリンク結合規則

$$(X = Y, P) \ \equiv \ P[Y/X]$$

（ただし X は P の自由リンク）

において…

1. X は $X = Y$ に出現する
2. 付帯条件より X は P にも出現する
3. LMNtal では 2 回出現したリンクは局所リンクであるため X は $(X = Y, P)$ の **局所リンク**

→ 従って，$(X = Y, P)$ の外で X が出現しない

ハイパーリンクへの局所性の素朴な導入

ハイパーリンク融合規則 ver. 2

$(X \bowtie Y, P) \equiv P\langle Y/X \rangle$

（ただし $\underline{(X \bowtie Y, P) \text{ の外で } X \text{ が出現しない}}$）

こうすれば…

$$\underbrace{(a(X), (X \bowtie Y, b(X, Y)))}_{(X \bowtie Y,\, b(X, Y)) \text{ の外で } X \text{ が出現}} \not\equiv_{\text{(E4), 融合規則 ver. 2}} (a(X), b(Y, Y))$$

ただし，このままでは「プログラム全体」について言及してしまっているため，部分だけで意味を定義することができておらず，compositional でない

ν を導入した理由

そこで，本研究では，局所リンクと自由リンクという概念を compositional にハイパーリンクに持ち込むために，明示的なスコープを新たな構文 $\nu X.P$ (hyperlink creation) として導入した

- $\nu X.(X \bowtie Y, P)$ が「$(X \bowtie Y, P)$ の外で X が出現しない」保証の役割を果たす

そして，LMNtal におけるリンクの結合に対応する規則は

ハイパーリンク融合規則 ver. 3

$$\nu X.(X \bowtie Y, P) \equiv P\langle Y/X \rangle$$

になる（...というのはウソでまだ続く）

ハイパーリンク融合規則 (E6) の右辺の ν

再掲：ハイパーリンク融合規則 ver. 3

$\nu X.(X \bowtie Y, P) \equiv P\langle Y/X \rangle$

とすると...，X と Y がどちらも同じ名前だったときに困る．例えば

$$\underbrace{\nu X.(X \bowtie X, a(X))}_{\text{自由出現するハイパーリンクはない}} \equiv_{\text{融合規則 ver. 3}} \underbrace{a(X)}_{X\text{ が自由出現}}$$

となり，$\equiv$ の両辺で自由ハイパーリンクの集合が変化してしまう

そこで，$\equiv$ の右辺にも $\nu X.$ を補う

ハイパーリンク融合規則 ver. 4

$\nu X.(X \bowtie Y, P) \equiv \nu X.P\langle Y/X \rangle$

こうすれば...

$$\underbrace{\nu X.(X \bowtie X, a(X))}_{\text{自由出現しない}} \equiv_{\text{融合規則 ver. 4}} \underbrace{\nu X.a(X)}_{\text{自由出現しない}}$$

ハイパーリンク融合規則 (E6) の付帯条件

再掲：ハイパーリンク融合規則 ver. 4

$\nu X.(X \bowtie Y, P) \equiv \nu X.P\langle Y/X\rangle$

のままだと…，X と Y のどちらも P に出現しなかった時に困る．例えば

$$\underbrace{\nu X.(X \bowtie Y, a())}_{Y\text{ が自由出現する}} \equiv_{\text{融合規則 ver. 4}} \underbrace{a()}_{\text{自由出現するハイパーリンクはない}}$$

となり，$\equiv$ の両辺で自由ハイパーリンクの集合が変化してしまう

そこで，付帯条件「$X \in \mathit{fn}(P) \lor Y \in \mathit{fn}(P)$」を補う

構造合同規則 (E6)

$\nu X.(X \bowtie Y, P) \equiv \nu X.P\langle Y/X\rangle$

(ただし $X \in \mathit{fn}(P) \lor Y \in \mathit{fn}(P)$)

こうすれば…

$$\underbrace{\nu X.(X \bowtie Y, a())}_{Y\text{ が自由出現する}} \not\equiv_{\text{(E6)}} \underbrace{a()}_{\text{自由出現しない}}$$

ハイパーリンク融合規則 (E7) $\langle 1/2 \rangle$

ただし，

再掲：構造合同規則 (E6)

$\nu X.(X \bowtie Y, P) \equiv \nu X.P\langle Y/X \rangle$

（ただし $X \in fn(P) \vee Y \in fn(P)$）

は付帯条件「ただし $X \in fn(P) \vee Y \in fn(P)$」のために，

X，Y が $X \bowtie Y$ にしか出現しない場合に，

fusion を吸収できない．

そこで，このケースに対応するために，さらに

構造合同規則 (E7)

$\nu X.\nu Y.X \bowtie Y \equiv \mathbf{0}$

を用意した

ハイパーリンク融合規則 (E7) $\langle 2/2 \rangle$

再掲：構造合同規則 (E7)

$\nu X.\nu Y.X \bowtie Y \equiv \mathbf{0}$

の左辺において $\nu Y.$ も必要な理由は，

$\nu Y.$ を省いて，

$$\underbrace{\nu X.X \bowtie Y}_{Y\text{ が自由出現する}} \quad \equiv \quad \underbrace{\mathbf{0}}_{\text{自由出現するハイパーリンクはない}}$$

とすると，$\equiv$ の両辺で自由ハイパーリンクの集合が変化してしまうため

Admissible な規則

提案した Flat HyperLMNtal の構造合同規則では
Flat LMNtal における構造合同規則と比較して

局所リンクの α 変換 $\quad P \equiv P[Y/X]$
（ただし X は P の局所リンク）

connector の対称性 $\quad X = Y \equiv Y = X$

に対応するものが欠けている

これはこれらの規則が他の規則から導出できること，
すなわち admissible であることがわかったため

→ それぞれ定理 1, 2 として証明を行なった

Flat HyperLMNtal の構造合同規則｜まとめ

$$
\begin{array}{llcl}
\text{(E1)} & (\mathbf{0}, P) & \equiv & P \\
\text{(E2)} & (P, Q) & \equiv & (Q, P) \\
\text{(E3)} & (P, (Q, R)) & \equiv & ((P, Q), R) \\
\text{(E4)} & P \equiv P' & \Rightarrow & (P, Q) \equiv (P', Q) \\
\text{(E5)} & P \equiv Q & \Rightarrow & \nu X.P \equiv \nu X.Q \\
\text{(E6)} & \nu X.(X \bowtie Y, P) & \equiv & \nu X.P\langle Y/X \rangle \\
 & \multicolumn{3}{l}{(\text{ただし } X \in \mathit{fn}(P) \vee Y \in \mathit{fn}(P))} \\
\text{(E7)} & \nu X.\nu Y.X \bowtie Y & \equiv & \mathbf{0} \\
\text{(E8)} & \nu X.\mathbf{0} & \equiv & \mathbf{0} \\
\text{(E9)} & \nu X.\nu Y.P & \equiv & \nu Y.\nu X.P \\
\text{(E10)} & \nu X.(P, Q) & \equiv & (\nu X.P, Q) \\
 & \multicolumn{3}{l}{(\text{ただし } X \notin \mathit{fn}(Q))}
\end{array}
$$

定義にあたっては，

自明でない試行錯誤 を要した

- 何度も「コーナーケース」を発見し，そのたびに修正を続けた

→ 重要な性質は定理として証明した

ただし成果物は **非常に簡明**

Flat HyperLMNtal の操作的意味論｜遷移規則

LMNtal と同様

(R1) $$\frac{P \longrightarrow P'}{(P, Q) \longrightarrow (P', Q)}$$

プロセス代数から新たに導入

(R2) $$\frac{P \longrightarrow P'}{\nu X.P \longrightarrow \nu X.P'}$$

LMNtal と同様

(R3) $$\frac{Q \equiv P \quad P \longrightarrow P' \quad P' \equiv Q'}{Q \longrightarrow Q'}$$

(R4) $$(P, (P \text{ :- } Q)) \longrightarrow (Q, (P \text{ :- } Q))$$

1. 研究の背景

2. Flat HyperLMNtal の抽象構文

3. Flat HyperLMNtal の操作的意味論

4. Flat HyperLMNtal に関して行なった証明

5. まとめ

λ 計算や π 計算は，

- α 変換が（構造）合同性を保つことや，
- 自由変数の集合が構造合同なプロセス間で変化しないこと [2]，
- 自由変数の集合が遷移に伴い広義単調減少すること

といった性質を満たす

これらは構文駆動な計算モデルにおいて普遍的な性質だと言える

そこで，これらの性質が Flat HyperLMNtal においても成り立つことを証明した

1. ただし，π 計算では，構造合同規則 Sc-Mat においてのみ，構造合同なプロセス間で自由変数の集合が変化する可能性がある

証明した定理

定理 1 α 変換

$$\nu X.P \equiv \nu Y.P\langle Y/X\rangle \text{ where } Y \notin \mathit{fn}(P)$$

定理 2 $\bowtie$ の対称性

$$X \bowtie Y \equiv Y \bowtie X$$

定理 3 構造合同なプロセス間での自由ハイパーリンクの集合の等価性

$$\mathit{fn}(P) = \mathit{fn}(Q) \text{ if } P \equiv Q$$

定理 4 遷移に伴う自由ハイパーリンクの広義単調減少性

$$\mathit{fn}(P) \supseteq \mathit{fn}(Q) \text{ if } P \longrightarrow Q$$

1. 研究の背景

2. Flat HyperLMNtal の抽象構文

3. Flat HyperLMNtal の操作的意味論

4. Flat HyperLMNtal に関して行なった証明

5. まとめ

まとめ

本研究では...

1. ハイパーグラフ書き換え系の構文駆動な構文・意味論を提案した
2. 提案した意味論が，構文駆動な計算モデルにおいて
   重要な性質を満たすことを証明した
3. (本発表では割愛)
   提案した意味論を応用して，
   HyperLMNtal（「階層グラフ」書き換え言語 LMNtal へハイパーリンクを導入した拡張）
   の形式化も行った

参考文献

[1] Berge Claude. Hypergraphs: Combinatorics of Finite Sets. North-Holland, 1989.

[2] H. Ehrig et al. Fundamentals of Algebraic Graph Transformation. Springer, 2006.

[3] Grzegorz Rozenberg. Handbook of Graph Grammars and Computing by Graph Transformation. World Scientific, 1997.

[4] Christopher Bak. GP 2: Efficient Implementation of a Graph Programming Language. PhD thesis. University of York, Sept. 2015.

[5] J. R. W. Glauert et al. Dactl - an Experimental Graph Rewriting Language. In: *Proc. Graph Grammars 1990*. Vol. 532. LNCS. Springer, 1991, pp. 378–395.

[6] Rubino Geiß et al. GrGen: A Fast SPO-Based Graph Rewriting Tool. In: *Proc. ICGT 2006*. Springer, 2006, pp. 383–397.

[7] Kazunori Ueda. LMNtal as a hierarchical logic programming language. In: **Theoretical Computer Science** 410.46 (2009), pp. 4784–4800.

参考文献

[8] Davide Sangiorgi et al. The Pi-Calculus: A Theory of Mobile Processes. Cambridge University Press, 2001.

[9] J. Parrow et al. The fusion calculus: expressiveness and symmetry in mobile processes. In: *Proc. LICS 1998*. 1998, pp. 176–185.

[10] A. Yasen et al. Hypergraph Representation of Lambda-Terms. In: *TASE 2016*. IEEE Computer Society, July 2016, pp. 113–116.

[11] Yutaro Tsunekawa et al. Implementation of LMNtal Model Checkers: A Metaprogramming Approach. In: **Journal of Object Technology** 17.1 (Nov. 2018).

[12] Stefan Walter. Capability Typing for HyperLMNtal. Master's Thesis. Waseda University, 2021.

[13] Jin Sano. Implementing G-Machine in HyperLMNtal. https://arxiv.org/abs/2103.14698. Bachelor's Thesis. Waseda University, 2021.

[14] Kazunori Ueda et al. HyperLMNtal: An Extension of a Hierarchical Graph Rewriting Model. In: **Künstliche Intelligenz** 26.1 (2012), pp. 27–36.