MELL のカット除去規則に基づく階層グラフ書換え言語の拡張

JSSST2024 2 日目 [4b-3-R]

2024/9/11

早稲田大学 基幹理工学研究科 情報理工・情報通信専攻

田久健人　上田和紀

概要

背景 階層グラフ書換え系は有用なモデリング・計算体系である．
しかしながら，論理体系との対応関係が自明ではない．

先行研究 階層グラフ書換え言語 LMNtal 上で，
MELL Proof Nets のカット除去規則のエンコードを行った [TU23]．

課題 既存のエンコードでは，Contraction, Weakening の操作を
直接的に表現できない．

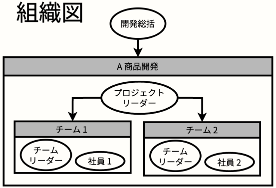
貢献 これらに直接的に対応する操作を定義，実装した上で，

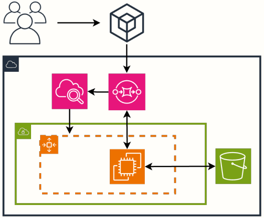
- これらの操作の有用性を示した．
- LMNtal の Proof Nets のワークベンチとしての有用性を示した．

背景：階層グラフ

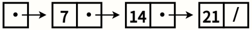
接続関係，階層構造は普遍的．

組織図

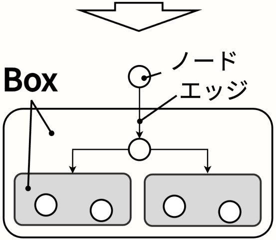


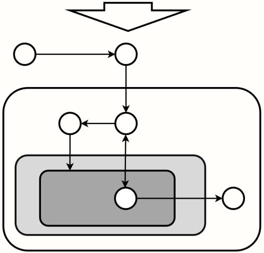


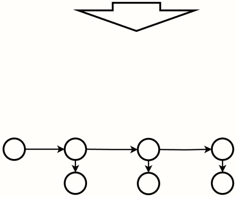
サービスのアーキテクチャ図



データ構造（リンクリスト）







階層グラフ(ノード + エッジ + **Box**) でこれらを表現できる．

背景：階層グラフ書換え系の例

階層グラフ書き換え系：

階層グラフの書換えにより計算・モデリングを行う体系.

- CHAM (Chemical Abstract Machine) [BB92]
- BRS (Bigraphical Reactive System) [Mil01]
- Hierarchical Graph Transformation [DHP02]
- LMNtal (elemental と発音) [Ued09]
- Attributed Hierarchical Port Graph [EFP18]

これらは主にモデリング言語として提案されている.

多くのものは処理系，ツールの実装もされている.

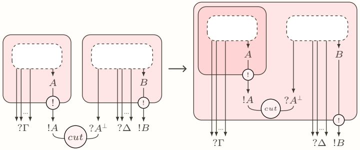
背景：階層グラフ書換え系と論理体系

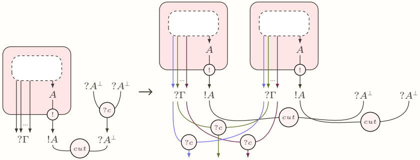
位置付けの明瞭化のため，他体系との対応関係の明示が重要．
特に，**論理体系との対応があるとより嬉しい．**

- 例：Curry-Howard 同型

Multiplicative Exponential Linear Logic (**MELL**) [Gir87]：

- Exponential を持つ体系の **Proof Nets** は，**Promotion Box** [Gir87] と呼ばれる Box 構造を持つ
- MELL Proof Nets のカット除去規則は，階層グラフの書換えを行う





→ **MELL Proof Nets とそのカット除去は，**
階層グラフ書換え系と何らかの対応関係が考えられる？

背景：MELL の定義，性質

定義. MELL の論理式

MELL の論理式は以下のように定義される．

$$\text{formula } F ::= X(atomic) \mid F^{\perp} \mid F \otimes F \mid F \parr F \mid\, !F \mid\, ?F$$

(結合の強さ：$^{\perp} \quad > \quad !, ? \quad > \quad \otimes, \parr$)

$$A^{\perp\perp} := A \qquad (A \otimes B)^{\perp} := A^{\perp} \parr B^{\perp} \qquad (A \parr B)^{\perp} := A^{\perp} \otimes B^{\perp}$$

$$(!A)^{\perp} := \ ?(A^{\perp}) \quad (?A)^{\perp} := !(A^{\perp})$$

$$A \multimap B := A^{\perp} \parr B$$

古典論理の含意 $\to$ は線形含意 $\multimap$ により以下のように埋め込まれる．

$$A \to B \ \mapsto \ !A \multimap B$$

背景：MELL の推論規則 (one-sided)

A, B: formula, Γ, Δ: lists of formulae

$$\frac{}{\vdash A, A^\perp}\,ax \qquad \frac{\vdash \Gamma, A \qquad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta}\,cut \qquad \frac{}{\vdash 1}\,1$$

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\,\parr \qquad \frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\,\otimes \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}\,\perp$$

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, \,!A}\,! \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\,?d \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\,?c \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}\,?w$$

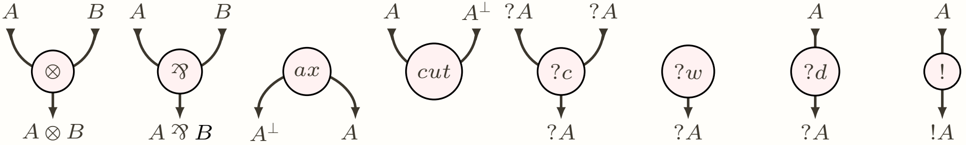
カット除去定理が成り立つ [Gir87].

背景：MELL と Proof Nets

MELL のシークエントに対応するグラフ構造を定義する．

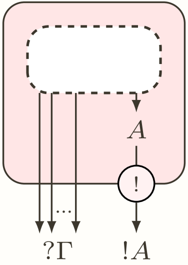
1. **証明構造**と呼ばれるグラフ構造を導入する
2. 証明構造の中で MELL シークエントと対応の取れるものを **Proof Nets** と呼ぶ
    - 整合性条件 ([DR89] 等) を満たすものとして定義される

背景：証明構造 ① セル・ワイヤー



- 有向非巡回多重グラフ
- セル (ノード) は MELL の推論規則でラベル付けされる
- ワイヤー (エッジ) は MELL 論理式でラベル付けされる
- **セル $\otimes$，$\parr$ のみ入力が順序づけられる**
  (cut と $?c$ は入力に区別がない)

背景：証明構造 ② Promotion Box



$$\frac{\vdash\ ?\Gamma,\ A}{\vdash\ ?\Gamma,\ !A}\,!$$

- !-規則に対応する
- 点線内には任意の証明構造が入る
- 必ず互いに分離しているか入れ子になる必要がある
- 内部を変換から保護し，簡約を順序づける

背景：証明構造，Proof Nets

定義. MELL の証明構造

MELL の証明構造は，

セル，ワイヤーと Promotion Box からなる有向非巡回多重グラフである．

定義. MELL Proof Nets

MELL Proof Nets は，

MELL の証明構造のうち，整合性条件 [DR89] を満たすものである．

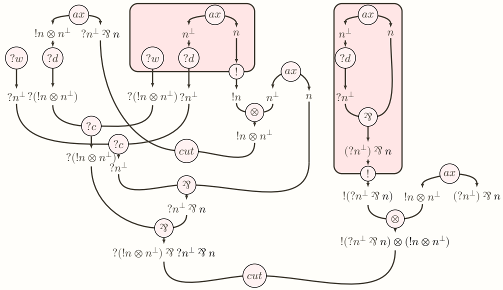
(詳細は省略，本文に記載．)

Proof Nets は，対応する MELL シークエントがある．

背景：MELL Proof Nets の例

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$!n \otimes n^\perp, ?n^\perp \parr n$}
\RightLabel{$?d$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp \parr n$}
\RightLabel{$?w$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp, ?n^\perp \parr n$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$n^\perp, n$}
\RightLabel{$?d$}
\UnaryInfC{$?n^\perp, n$}
\RightLabel{$?w$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp, n$}
\RightLabel{$!$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp, !n$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$n, n^\perp$}
\RightLabel{$\otimes$}
\BinaryInfC{$?(!n \otimes n^\perp), ?n^\perp, n, !n \otimes n^\perp$}
\RightLabel{cut}
\BinaryInfC{$?(!n \otimes n^\perp), ?(!n \otimes n^\perp), ?n^\perp, ?n^\perp, n$}
\RightLabel{$?c$}
\UnaryInfC{$?(!n \otimes n^\perp), ?(!n \otimes n^\perp), ?n^\perp, n$}
\RightLabel{$?c$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp, n$}
\RightLabel{$\parr$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp \parr n$}
\RightLabel{$\parr$}
\UnaryInfC{$?(!n \otimes n^\perp) \parr ?n^\perp \parr n$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$n^\perp, n$}
\RightLabel{$?d$}
\UnaryInfC{$?n^\perp, n$}
\RightLabel{$\parr$}
\UnaryInfC{$?n^\perp \parr n$}
\RightLabel{$!$}
\UnaryInfC{$!(?n^\perp \parr n)$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$!n \otimes n^\perp, ?n^\perp \parr n$}
\RightLabel{$\otimes$}
\BinaryInfC{$!(?n^\perp \parr n) \otimes (!n \otimes n^\perp), ?n^\perp \parr n$}
\RightLabel{cut}
\BinaryInfC{$?n^\perp \parr n$}
\end{prooftree}
$$

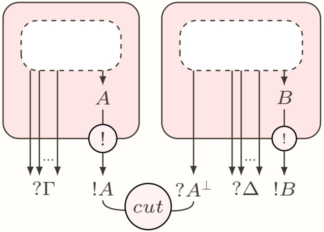
$$\mapsto$$



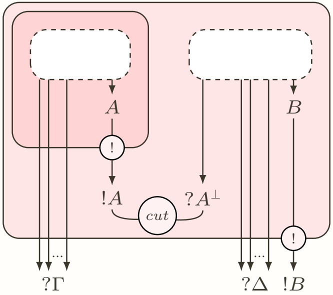
カット除去規則の例 ① box-nested

$$\dfrac{\dfrac{?\Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad \dfrac{?\Delta, B, ?A^{\perp}}{?\Delta, !B, ?A^{\perp}}\,!}{\vdash ?\Gamma, ?\Delta, !B}\,cut \quad \rightarrow \quad \dfrac{\dfrac{\dfrac{?\Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad ?\Delta, B, ?A^{\perp}}{\vdash ?\Gamma, ?\Delta, B}\,cut}{\vdash ?\Gamma, ?\Delta, !B}\,!$$

$$\downarrow \qquad\qquad\qquad \downarrow$$



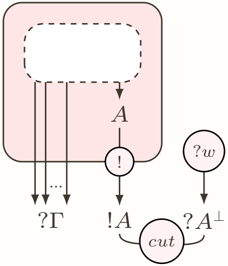
$$\rightarrow$$



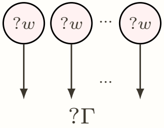
Box の移動をしている.

カット除去規則の例 ② box-weakening

$$\dfrac{\dfrac{?\Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad \dfrac{\Delta}{\Delta, ?A^{\perp}}\,?w}{\vdash ?\Gamma, \Delta}\,cut \quad \longrightarrow \quad \dfrac{\Delta}{\vdash ?\Gamma, \Delta}\,?w*$$



$\longrightarrow$

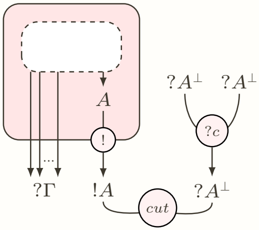


Box の削除をしている.

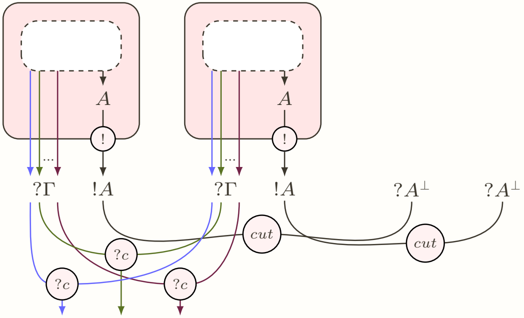
カット除去規則の例 ③ box-contraction

$$\dfrac{\dfrac{?\Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad \dfrac{\Delta, ?A^{\perp}, ?A^{\perp}}{\Delta, ?A^{\perp}}\,?c}{\vdash ?\Gamma, \Delta}\,cut \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{?\Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad \dfrac{\dfrac{?\Gamma, A}{?\Gamma, !A}\,! \qquad \Delta, ?A^{\perp}, ?A^{\perp}}{?\Gamma, \Delta, ?A^{\perp}}\,cut}{\vdash ?\Gamma, ?\Gamma, \Delta}\,cut}{\vdash ?\Gamma, \Delta}\,?c*$$

$$\downarrow \qquad\qquad\qquad\qquad \downarrow$$



$$\longrightarrow$$

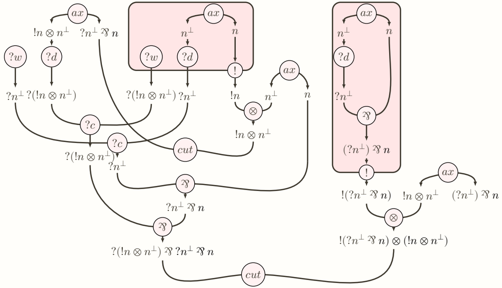


Box の複製をしている．

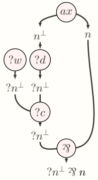
背景：カット除去の適用例

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$!n \otimes n^\perp, ?n^\perp \parr n$}
\RightLabel{$?d$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp \parr n$}
\RightLabel{$?w$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp, ?n^\perp \parr n$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$n^\perp, n$}
\RightLabel{$?d$}
\UnaryInfC{$?n^\perp, n$}
\RightLabel{$?w$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp, n$}
\RightLabel{$!$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp, !n$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$n, n^\perp$}
\RightLabel{$\otimes$}
\BinaryInfC{$?(!n \otimes n^\perp), ?n^\perp, n, !n \otimes n^\perp$}
\RightLabel{cut}
\BinaryInfC{$?(!n \otimes n^\perp), ?(!n \otimes n^\perp), ?n^\perp, ?n^\perp, n$}
\RightLabel{$?c$}
\UnaryInfC{$?(!n \otimes n^\perp), ?(!n \otimes n^\perp), ?n^\perp, n$}
\RightLabel{$?c$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp, n$}
\RightLabel{$\parr$}
\UnaryInfC{$?(!n \otimes n^\perp), ?n^\perp \parr n$}
\RightLabel{$\parr$}
\UnaryInfC{$?(!n \otimes n^\perp) \parr ?n^\perp \parr n$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$n^\perp, n$}
\RightLabel{$?d$}
\UnaryInfC{$?n^\perp, n$}
\RightLabel{$\parr$}
\UnaryInfC{$?n^\perp \parr n$}
\RightLabel{$!$}
\UnaryInfC{$!(?n^\perp \parr n)$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$!n \otimes n^\perp, ?n^\perp \parr n$}
\RightLabel{$\otimes$}
\BinaryInfC{$!(?n^\perp \parr n) \otimes (!n \otimes n^\perp), ?n^\perp \parr n$}
\RightLabel{cut}
\BinaryInfC{$?n^\perp \parr n$}
\end{prooftree}
\quad \twoheadrightarrow \quad
\begin{prooftree}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$n^\perp, n$}
\RightLabel{$?d$}
\UnaryInfC{$?n^\perp, n$}
\RightLabel{$?w$}
\UnaryInfC{$?n^\perp, ?n^\perp, n$}
\RightLabel{$?c$}
\UnaryInfC{$?n^\perp, n$}
\RightLabel{$\parr$}
\UnaryInfC{$?n^\perp \parr n$}
\end{prooftree}
$$

$$\downarrow \qquad\qquad\qquad \downarrow$$



$$\twoheadrightarrow$$



背景：カット除去に関する諸性質

定理. カット除去定理 [Gir87]

MELL Proof Nets は任意のカットを除去できる.

定理. 安定性 [Gir87]

任意の MELL Proof Net は，カット除去後も MELL Proof Net である.

定理. 合流性 [Gir87; PF10]

MELL Proof Nets のカット除去は合流性を持つ.

定理. 強正規化性 [Gir87; PF10]

MELL Proof Nets のカット除去は強正規化性を持つ.

背景：Proof Nets と階層グラフ書換え系

Proof Nets から出発した (階層) グラフ書換え系の例：

- Interaction Nets [Laf89]
- Interaction Nets with Boxes [AFM11]
- Hypernet [Mur20]

✓ 主に関数型言語の計算モデルとして有用

▲ 表現可能なグラフ構造が限定されていて，
モデリング言語としては十分とは言えない

背景：階層グラフ書換え言語 LMNtal

本研究では，モデリング目的で設計された階層グラフ書換え系から出発し，論理体系との対応を考察する．

今回は，**階層グラフ書換え言語 LMNtal** を採用する．

- ✓ 構造的操作的意味論 (SOS) を持つ
  - グラフ変換は，Double Pushout (圏論) による意味論が主流
- ✓ 様々な他体系との対応が調べられている
  - 並行計算モデルやラムダ計算等
- ✓ オープンソースの処理系 [石川 +08] がある
  - https://github.com/lmntal，メンテナンスが続いている
- ✓ 処理系がモデル検査器機能を提供している [GHU11]

背景：LMNtal グラフ

The syntax of LMNtal

プロセス

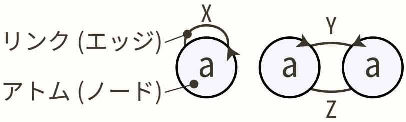
$P ::= 0$ (空)
| $p(X_1, \dots, X_m)$ (アトム)
| P, P (分子)
| $\{P\}$ (膜)
| [*RuleName* @@] T :- T (ルール)

プロセステンプレート

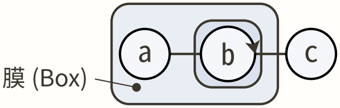
$T ::= 0$ (空)
| $p(X_1, \dots, X_m)$ (アトム)
| T, T (分子)
| $\{T\}$ (膜)
| [*RuleName* @@] T :- T (ルール)
| @p (ルール文脈)
| \$p[$X_1, \dots, X_m$ | A] (プロセス文脈)
| $p({}^*X_1, \dots, {}^*X_m)$ (アトム集団)

剰余引数

$A ::= []$
| ${}^*X$ (リンク束)




```
a(X,X), a(Y,Z), a(Y,Z).
```




```
{a(X),{b(Y,X)}},c(Y).
```

- アトム (小文字)，リンク (大文字)
- 無向多重ポートグラフ [PMD12]
- 矢印は引数順序に対応する
- リンク名は α 変換可能 (重要でない)

背景：ルール，プロセス文脈，リンク束

The syntax of LMNtal

プロセス

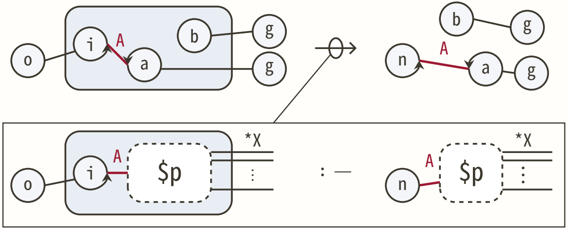
$P ::= 0$		(空)
|	$p(X_1, \dots, X_m)$	(アトム)
|	P, P	(分子)
|	$\{P\}$	(膜)
|	$[\mathit{RuleName}$ @@$]\ T$:- T	(ルール)

プロセステンプレート

$T ::= 0$		(空)
|	$p(X_1, \dots, X_m)$	(アトム)
|	T, T	(分子)
|	$\{T\}$	(膜)
|	$[\mathit{RuleName}$ @@$]\ T$:- T	(ルール)
|	@p	(ルール文脈)
|	\$$p[X_1, \dots, X_m \mid A]$	(プロセス文脈)
|	$p({}^*X_1, \dots, {}^*X_m)$	(アトム集団)

剰余引数

$A ::= []$		
|	${}^*X$	(リンク束)




```
o(B),{i(A,B),a(A,G1),b(G2)},g(G1),g(G2). //init
o(B),{i(A,B),$p[A|*X]} :- n(A),$p[A|*X]. //rule
```

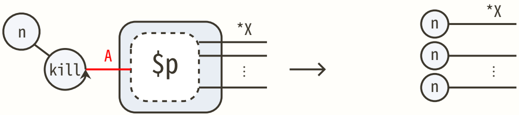
- **自由リンク**：片側が接続先未定のリンク
- `$p` ：ワイルドカード
- `*X` ：不特定多数本の自由リンクの列

ルールの適用は SOS[Ued09] に従う．

背景：nlmem ライブラリ

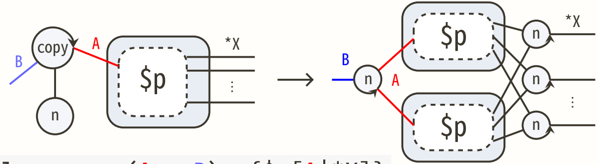
ライブラリ nlmem (non-linear membrain) [乾 +08]：
不特定多数の自由リンクを持つ 膜の消去，複製を安全に行う

膜の消去 nlmem.kill：




```
nlmem.kill(A,n), {$p[A|*X]}
```

膜の複製 nlmem.copy：

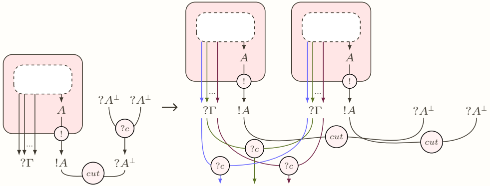



```
nlmem.copy(A,n,B), {$p[A|*X]}
```

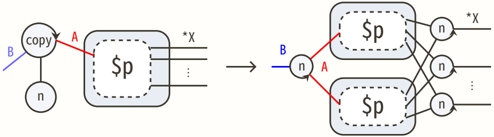
※ ルール前後の自由リンクの本数が等しい = ポインタ安全

課題：box-contraction のエンコード

エンコード [TU23] において，Contraction に関して，



nlmem.copy を用いたが，



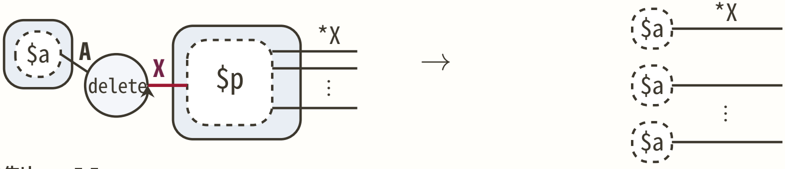
✖ アトム以外の構造を不特定多数個生成できない

✖ cut セル接続部分が直接的でない

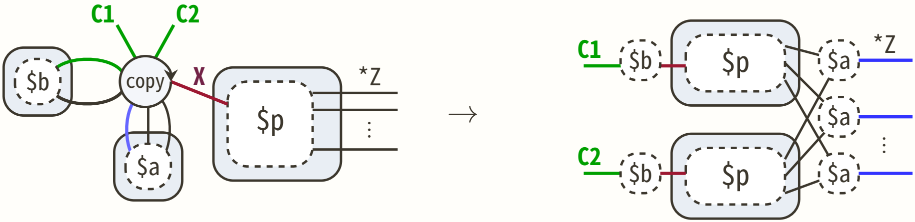
→ 直接的に記述することはできなかった

提案する拡張：mell ライブラリ

膜の消去 mell.delete：



膜の複製 mell.copy：



※ 詳しい仕様は本発表では省略

mell ライブラリの実装

`mell.delete`, `mell.copy` を実装した.

- LMNtal 処理系上のライブラリとして実装
- feature/mell ブランチで公開：
  https://github.com/lmntal/slim/tree/feature/mell
- 本ブランチをビルドで使用可能

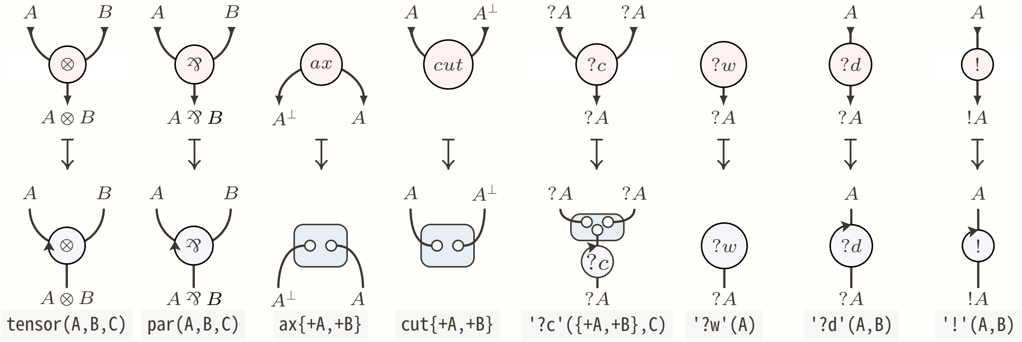
※ 以降の例題は全て処理系上で動作確認済み

MELL Proof Nets のエンコード

MELL Proof Nets を拡張 LMNtal でエンコードする．

1. 証明構造の構成要素のエンコード
2. カット除去規則のエンコード
3. 実行例：単純型付きラムダ計算の β 簡約の状態空間描画
4. エンコードの正しさに関する議論
5. 応用例：ルールの追加，検証

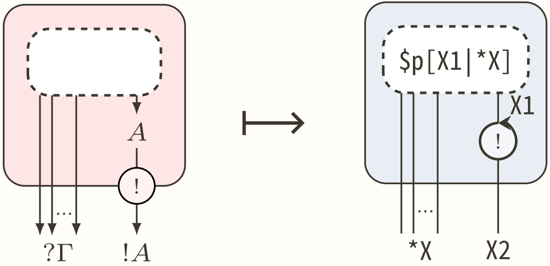
エンコード 1：セル・ワイヤー



- $\otimes$，$\parr$ の入力の順序づけは，アトムで表現
- ax，cut は入力に順序がないので，膜で表現
- $?c$ は，入出力の区別はアトム，順序のない入力は膜

※ 論理式を表す自由リンクは，アトム formula を繋げる

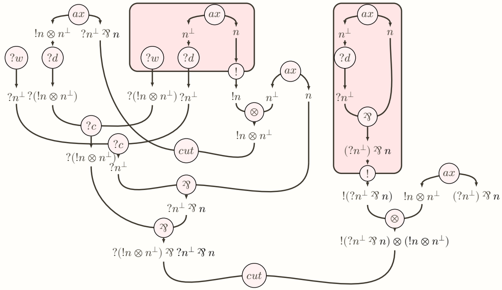
エンコード 1：Promotion Box



`{'!'(X1,X2),$p[X1|*X]}.`

- Box の外枠は膜で表す．
- $?\Gamma$ は，リンク束 `*X` で表す．
- 空白部分は，プロセス文脈 `$p[X1|*X]` で表す．

エンコード 1：例




```
ax{+A1,+A2},'?d'(A1,A3),'?w'(A4).
{ax{+B1,+B2},'?d'(B1,B3),'?w'(B4),'!'(B2,B5)}.
'?c'({+A3,+B4},C1).'?c'({+A4,+B3},C2).tensor(B5,T1,D2),ax{+T1,+T2}.
cut{+A2,+D2}.par(C2,T2,P1).par(C1,P1,F).
cut{+F,+D4}.
{ax{+E1,+E2},'?d'(E1,E3),par(E3,E2,E4),'!'(E4,E5).}.
tensor(E5,T3,D4),ax{+T3,+T4}.formula(T4).
```

エンコード 2：カット除去規則

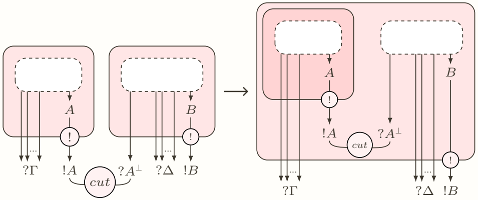
愚直にルールとして落とし込む.

実装：

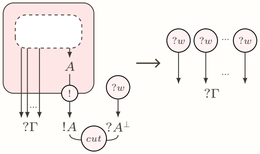
https://github.com/kyawaway/MELL-Nets-in-LMNtal/tree/mell-module

(補足) カット除去規則の例

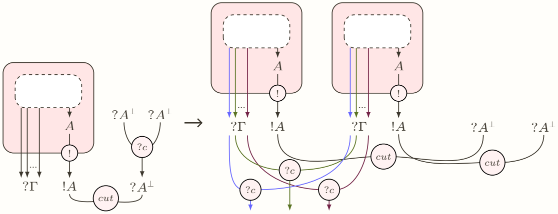
box-nested:



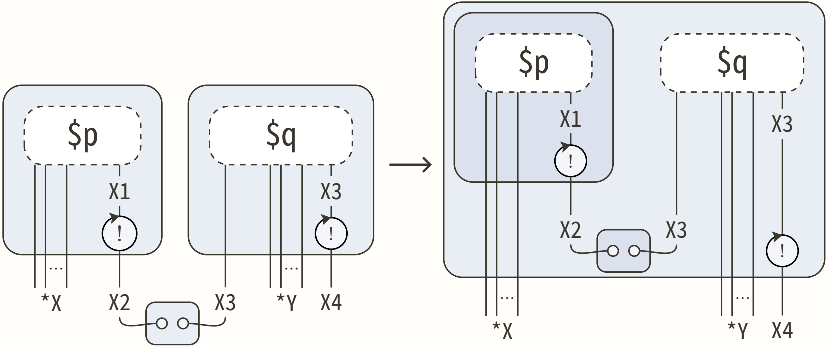
box-weakening:



box-contraction:

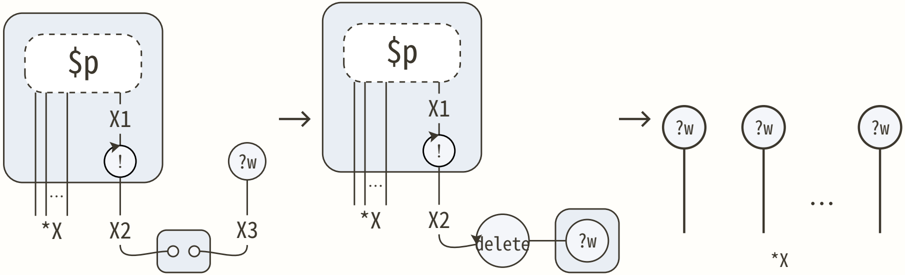


エンコード 2：box-nested



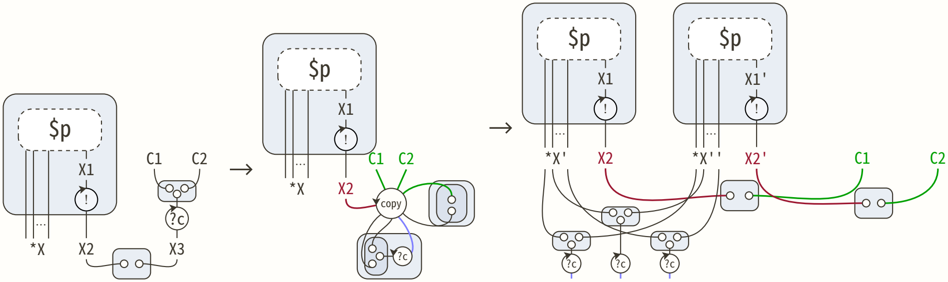

```
box_nested@@
{'!'(X1,X2),$p[X1|*X]},{$q[X3|*Y]},cut{+X2,+X3}
  :- {{'!'(X1,X2),$p[X1|*X]},$q[X3|*Y],cut{+X2,+X3}}.
```

エンコード 2：box-weakening




```
box_weakening@@
{'!'(X1,X2),$p[X1|*X]},cut{+X2,+X3},'?w'(X3)
  :- mell.delete(X,A),{$p[X|*X]},{'?w'(A)}.
```

エンコード 2：box-contraction




```
box_contraction@@
{'!'(X1,X2),$p[X1|*X]},'?c'({+C1,+C2},X3),cut{+X2,+X3},
  :- mell.copy(X2,A1,A2,A3,B1,B2,C1,C2),
     {'!'(X1,X2),$p[X1|*X]},
     {'?c'({+A1,+A2},A3)},{cut{+B1,+B2}}.
```

nlmem による box-$?c$ の記述 (再掲)

旧バージョン (nlmem.copy を使用)：

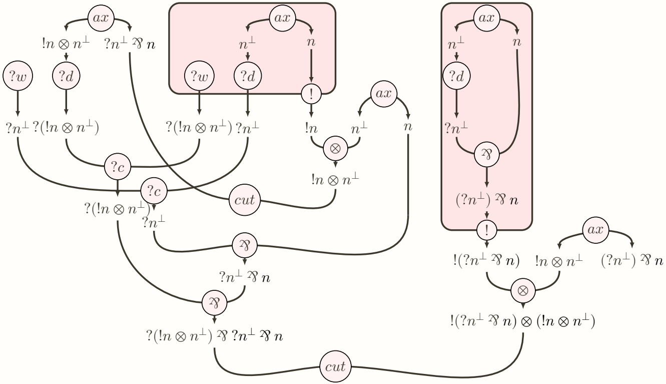
```
box_contraction_step1@@
{'!'(X1,X2),$p[X1|*X]},cut{+X2,+X3},'?c'({+C1,+C2},X3)
  :- nlmem.copy(X2,'?c_sub',C),{'!'(X1,X2),$p[X1|*X]},cut_sub(C,C1,C2).

box_contraction_step2@@
'?c_sub'(A,B,C) :- '?c'({+A,+B},C).

box_contraction_step3@@
'?c'({+X21,+X22},XC),cut_sub(XC,C1,C2) :- cut{+X21,+C1},cut{+X22,+C2}.
```

✓ 3 ステップに分けていたが，**1 ステップで記述できた**

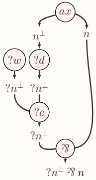
実行例：単純型付きラムダ計算

$$(\lambda f : n \to n \,.\, \lambda x : n \,.\, f\ x)\ (\lambda x : n \,.\, x) \quad \twoheadrightarrow \quad (\lambda x : n \,.\, x)$$

$$\downarrow \qquad\qquad \downarrow$$



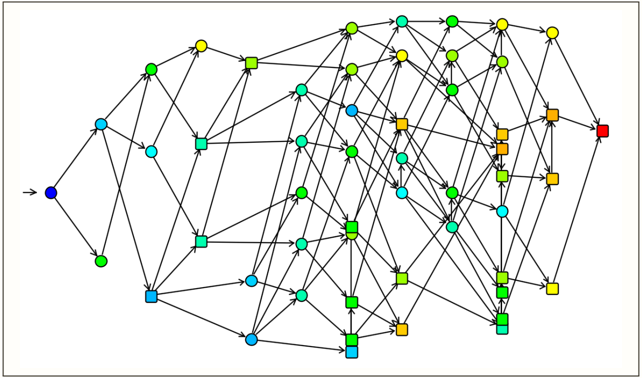
$$\twoheadrightarrow$$



この書換えの状態空間を出力する．

実行例：実行結果

状態空間 (StateViewer [綾野 +10] により出力)：



任意の状態から有限ステップで唯一の終了状態 (赤) に到達．

→ 合流性，強正規性が現れている

エンコードの正しさに関する議論

一般的には，，，

1. 仕様が与えられ，
2. それを満たすアルゴリズムを記述し，
3. アルゴリズムの健全性，完全性，決定性などを証明する

今回の場合は，**アルゴリズムが限りなく仕様に近い**.

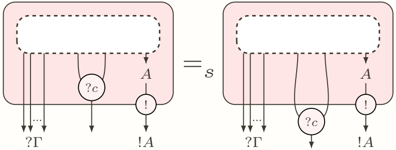
→ どちらも図表現があり，非形式的には図より明らか

※ 形式的には，それぞれの数学的定義間の対応を取る

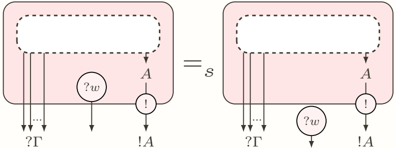
応用例：ルールの追加

以下の同値関係 $=_s$ [Dan90; DG99] の導入を考える.

(a) contraction-pull-equivalence



(b) weakening-pull-equivalence



- 代入表現に関する細かい区別において必要となる

これらを追加した時のカット除去の性質はどうなるか？

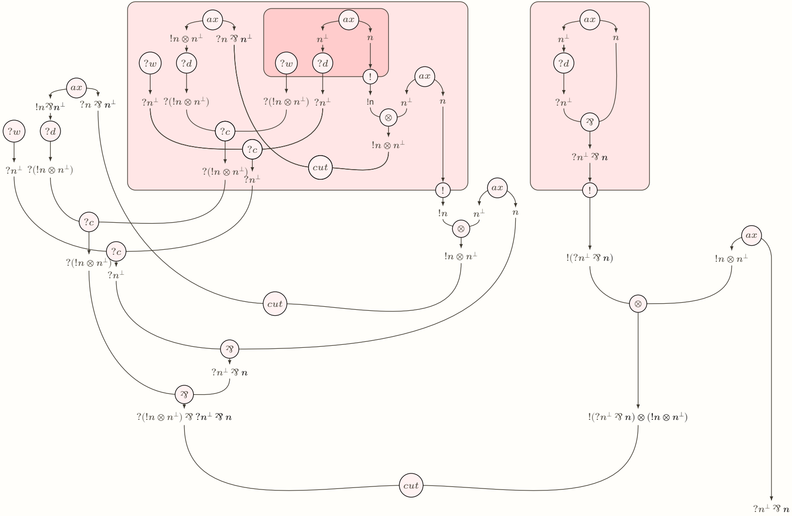
応用例：$=_s$ ルールの記述


```
box_contraction_pull@@
{'!'(X1,X2),'?c'({+X3,+X4},X5),$p[X1,X3,X4|*X]}
  :- {'!'(X1,X2),$p[X1,X3,X4|*X]},'?c'({+X3,+X4},X5).

box_contraction_push@@
{'!'(X1,X2),$p[X1,X3,X4|*X]},'?c'({+X3,+X4},X5)
  :- { '!'(X1,X2),$p[X1,X3,X4|*X],'?c'({+X3,+X4},X5)}.

box_weakening_pull@@
{'!'(X1,X2),'?w'(X3),$p[X1|*X]} :- {'!'(X1,X2),$p[X1|*X]},'?w'(X3).

box_weakening_push@@
{'!'(X1,X2),$p[X1|*X]},'?w'(X3) :- {'!'(X1,X2),'?w'(X3),$p[X1|*X]}.
```

$\rightarrow_s$：pull(Box から取り出す)，$\leftarrow_s$：push(Box に入れる)

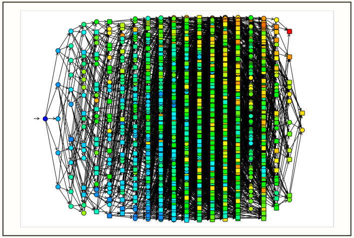
応用例：適用対象の Proof Nets

$(\lambda f : n \to n \,.\, \lambda x : n \,.\, f\ (f\ x))\ (\lambda x : n \,.\, x)$ に対応する：



応用例：Contraction の $=_s$

pull:

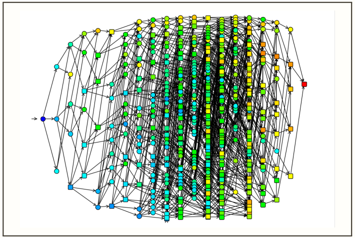


状態数：1808

終了状態数：1

(CR，SN)

push:

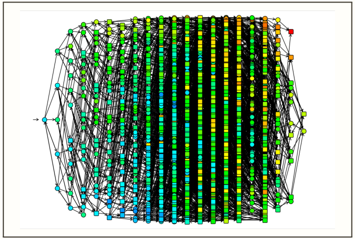


状態数：476

終了状態数：1

(CR，SN)

equivalence:



状態数：1808

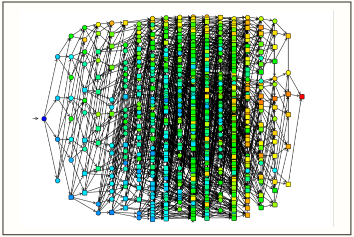
終了状態数：1

(CR)

※ equivalence ($=_s$) = pull ($\to_s$) + push ($\leftarrow_s$)

応用例：Weakening の $=_s$

pull:

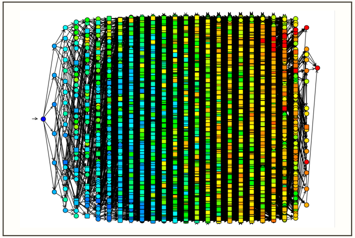


状態数：756

終了状態数：1

(CR，SN)

push:



状態数：41216

終了状態数：16

(SN)

equivalence:

状態数：? (爆発)

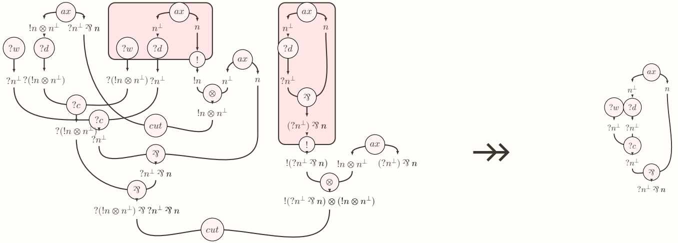
- ✔ Promotion Box の書換え規則を容易に追加できた
- ✔ 状態空間描画機能を用いて，性質の変化を視覚化できた

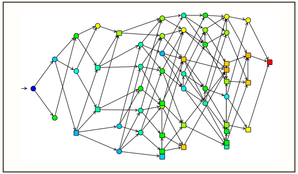
エンコード：まとめ

拡張 LMNtal 上で MELL のカット除去をエンコードした．

- **自然に**エンコードできた
- 適用パターンの状態空間が構築できた
- ルールの追加が容易に行えた

→ **Proof Nets のワークベンチとして有用である**





その他の使用例

`mell` ライブラリの階層グラフ書換えにおける有用性を，
例題の記述によって検討する．

? Promotion Box は，階層グラフ書換えによる
計算やモデリングにおいて有用か？

→ ① π 計算や ② Ambient 計算等，
プロセス計算のエンコードにおいて使用可能
(Ambient の表現や，プロセスの複製，強制終了等)

使用例：プロセス計算 ① 同期 π 計算

プロセスの複製，強制終了のエンコードで使用可能：

1. 同期 π 計算：プロセスの強制終了

$$(x(y).P + M \mid \bar{x}\langle z\rangle.Q + N) \rightarrow P[z/y] \mid Q$$

`nlmem.kill` [Ued09]：

```
comm@@ {$x,+C1,+C2},{get(C1,Y,{$p[Y|*V1]}),$m},{snd(C2,Z,{$q}),$n}
      :- {$x},$p[Z|*V1],$q,nlmem.kill({$m,$n}).
```

`mell.delete`：

```
comm@@ {$x,+C1,+C2},{get(C1,Y,{$p[Y|*V1]}),$m},{snd(C2,Z,{$q}),$n}
  :- {$x},$p[Z|*V1],$q,mell.delete({$m,$n},d).
```

使用例：プロセス計算 ② ambient 計算

2. Ambient 計算：プロセスの複製

$$!(\mathrm{open}\ m.P) \mid m[Q] \rightarrow P \mid Q \mid !(\mathrm{open}\ m.P)$$

nlmem.copy [Ued08]：

```
open_repl@@ open_repl(M,{$p}),{amb(M1),{id,+M1,-M2,$mm},$q,@q},{id,+M,+M2,$m}
    :- nlmem.copy({$p},cp,Copies),copies(Copies,P),
       $q,{id,+M3,$m,$mm},open_repl(M3,P).
open_repl_aux@@ copies(cp(C1,C2),P),{+C1,$p1} :- $p1,P=C2.
```

mell.copy：

```
open_repl@@ open_repl(M,{$p}),{amb(M1),{id,+M1,-M2,$mm},$q,@q},{id,+M,+M2,$m}
  :- mell.copy({$p},A1,A2,A3,B1,B2,remove,P),{cp(A1,A2,A3)},{B1=B2},
     $q,{id,+M3,$m,$mm},open_repl(M3,P).
open_repl_aux@@ remove({$p}) :- $p.
```

使用例：まとめ

LMNtal 上で，MELL の Promotion Box の動作に基づく操作を定義し，処理系に実装した．

- 膜の copy，delete 操作を定義した

プロセス計算におけるプロセスの削除，複製が表現できた．

- Promotion Box は排除されることが多い
  - Optimal Reduction 等，局所的書換えに基づく書換え系

✓ 一方で，**並行性が求められる動作の記述においては有用**

まとめ

LMNtal 上で，MELL Proof Nets の Promotion Box の動作に基づく操作 `mell` を実装したことで，

- ✓ **有用な Proof Nets のワークベンチが提供できた**
- ✓ **Promotion Box を，階層グラフ書換えにおいて並行性のある動作の記述に応用できた**

→ 階層グラフ書換え側，Proof Nets 側の双方に利点あり

Reference I

[TU23] Kento Takyu and Kazunori Ueda. "Encoding MELL Cut Elimination into a Hierarchical Graph Rewriting Language". In: *The 21st Asian Symposium on Programming Languages and Systems SRC&Posters*. 2023.

[BB92] Gérard Berry and Gérard Boudol. "The chemical abstract machine". In: *Theoretical Computer Science* 96.1 (1992), pp. 217–248. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(92)90185-I.

[Mil01] Robin Milner. "Bigraphical Reactive Systems". In: *Proceedings of the 12th International Conference on Concurrency Theory*. CONCUR '01. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 16–35. ISBN: 3-540-42497-0.

[DHP02] Frank Drewes, Berthold Hoffmann, and Detlef Plump. "Hierarchical Graph Transformation". In: *Journal of Computer and System Sciences* 64.2 (2002), pp. 249–283. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.2001.1790.

Reference II

[Ued09] Kazunori Ueda. “LMNtal as a hierarchical logic programming language”. In: *Theoretical Computer Science* 410.46 (2009), pp. 4784–4800. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2009.07.043.

[EFP18] Nneka Chinelo Ene, Maribel Fernández, and Bruno Pinaud. “Attributed hierarchical port graphs and applications”. English. In: *Electronic Proceedings in Theoretical Computer Science, EPTCS* 265 (Feb. 2018). Publisher: Open Publishing Association, pp. 2–19. ISSN: 2075-2180.

[Gir87] Jean-Yves Girard. “Linear logic”. In: *Theoretical Computer Science* 50.1 (1987), pp. 1–101. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(87)90045-4.

[DR89] Vincent Danos and Laurent Regnier. “The Structure of Multiplicatives”. In: *Archive for Mathematical Logic* 28.3 (1989), pp. 181–203. DOI: 10.1007/bf01622878.

Reference III

[PF10] Michele Pagani and Lorenzo Tortora de Falco. "Strong normalization property for second order linear logic". In: *Theoretical Computer Science* 411.2 (2010), pp. 410–444. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2009.07.053.

[Laf89] Yves Lafont. "Interaction Nets". In: *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '90. New York, NY, USA: Association for Computing Machinery, 1989, pp. 95–108. ISBN: 0-89791-343-4. DOI: 10.1145/96709.96718.

[AFM11] Sandra Alves, Maribel Fernández, and Ian Mackie. "A new graphical calculus of proofs". In: *Electronic Proceedings in Theoretical Computer Science* 48 (Feb. 2011). DOI: 10.4204/EPTCS.48.8.

[Mur20] Koko Muroya. "Hypernet semantics of programming languages". Ph.D. thesis. University of Birmingham, 2020.

Reference IV

[石川 +08] 石川 力 et al. “軽量な LMNtal 実行時処理系 SLIM の設計と実装”. In: 全国大会講演論文集. Vol. 第 70 回. Mar. 2008, pp. 153–154.

[GHU11] Masato Gocho, Taisuke Hori, and Kazunori Ueda. “Evolution of the LMNtal runtime to a parallel model checker”. English. In: *Computer Software* 28.4 (2011), pp. 137–157. ISSN: 0289-6540.

[PMD12] Bruno Pinaud, Guy Melançon, and Jonathan Dubois. “PORGY: A Visual Graph Rewriting Environment for Complex Systems”. In: *Computer Graphics Forum*. Eurographics Conference on Visualization (EuroVis 2012) 31.3 (2012). Publisher: Wiley, pp. 1265–1274. DOI: `10.1111/j.1467-8659.2012.03119.x`.

[乾 +08] 乾 敦行 et al. “階層グラフ書換えモデルに基づく統合プログラミング言語 LMNtal”. In: コンピュータソフトウェア 25.1 (2008), 1_124–1_150. DOI: `10.11309/jssst.25.1_124`.

Reference V

[綾野 +10] 綾野 貴之 et al. "統合開発環境による LMNtal モデル検査". In: コンピュータソフトウェア 27.4 (2010), 4_197–4_214. DOI: 10.11309/jssst.27.4_197.

[Dan90] Vincent Danos. "La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du Lambda-calcul)". Ph.D. thesis. University of Paris VII, 1990.

[DG99] Roberto Di Cosmo and Stefano Guerrini. "Strong Normalization of Proof Nets Modulo Structural Congruences". In: *Rewriting Techniques and Applications*. Ed. by Paliath Narendran and Michael Rusinowitch. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 75–89. ISBN: 978-3-540-48685-5.

[Ued08] Kazunori Ueda. "Encoding Distributed Process Calculi into LMNtal". In: *Electronic Notes in Theoretical Computer Science* 209 (2008), pp. 187–200. ISSN: 1571-0661. DOI: https://doi.org/10.1016/j.entcs.2008.04.012.