演習 III 第 1 期 最終発表

Quantitative Type Theory

稲吉悠羽

May 26, 2025

自分の選んだテーマ

Robert Atkey による 依存型と線形型のある体系を形式化した論文 Syntax and Semantics of Quantitative Type Theory を読む.

依存型 項に依存する型.

線形型 プログラムの中でちょうど n 回使う項であることを保証する型.

動機

依存型のある体系ではしばしば型検査にしか使われず, 実行時の計算には必要のない項が生じる. **実行時の計算に使われる回数**を線形型の使用回数とすれば, 型を見ると実行時に「使う (present)」それとも「使わない (erased)」かを判別できる.

使用回数は, 全く使わない 0, ちょうど 1 回つかう 1, 制限なしの ω で区別するということをよくやる. 文脈の結合や分配に対応するためにはこれらの要素に対する半環を考える.

Section 2 : 構文論

内容

- McBride(2016) によるアイディアに必要な数量的な注釈と半環について
- BNF 記法による項や型の文法
- 型付けの推論規則 (双方向型付け)
- 構文論的性質に関する補題 (弱化や代入など)

ポイント

- 文脈に対する注釈
- Dependent Tensor Product Types
- McBride(2016) による推論規則の問題点の指摘&修正

semiring elements $\rho, \pi \in R$

precontexts $\Gamma ::= \diamond \mid \Gamma, x \overset{\rho}{:} S$

pretypes

$$S, T, U ::= (x \overset{\pi}{:} S) \to T \mid (x \overset{\pi}{:} S) \otimes T \mid I \mid \mathsf{Bool} \mid \mathsf{El}(M) \mid \mathsf{Set}$$

preterms

$$\begin{aligned} M, N, O ::= {} & x \mid \lambda x \overset{\pi}{:} S.M^T \mid \mathsf{App}_{(x \overset{\pi}{:} S)T}(M, N) \mid (M, N)_{x \overset{\pi}{:} S.T} \mid * \mid \mathsf{fst}_{x \overset{\pi}{:} S.T}(M) \\ & \mid \mathsf{snd}_{x \overset{\pi}{:} S.T}(M) \mid \mathsf{let}_{x \overset{\pi}{:} S.T}(x, y) = M \text{ in } N \mid \mathsf{let}_I * = M \text{ in } N \mid \mathsf{true} \mid \mathsf{false} \\ & \mid \mathsf{ElimBool}_{(z)T}(M_t, M_f, N) \end{aligned}$$

$(+)$: 複数の文脈における使用を累積させる

ElimBool は依存 if 文のような構文. 条件節の文脈 Γ_2 と then/else 節 の文脈 Γ_1 における使用を合わせたものが全体の使用にあたる. M_t, M_f のどちらの項も同じ文脈から導かれているので, 片方しか評価されないこともわかる.

$$\frac{0\Gamma_1, z \overset{0}{:} \mathsf{Bool} \vdash T \quad \Gamma_1 \vdash M_t \overset{\sigma}{:} T[\mathsf{true}/z] \quad \Gamma_1 \vdash M_f \overset{\sigma}{:} T[\mathsf{false}/z] \quad \Gamma_2 \vdash N \overset{\sigma}{:} \mathsf{Bool} \quad 0\Gamma_1 = 0\Gamma_2}{\Gamma_1 + \Gamma_2 \vdash \mathsf{ElimBool}_{(z)T}(M_t, M_f, N) \overset{\sigma}{:} T[N/z]} \tag{B-ELIM}$$

$(\cdot)$: ネストした使用のカウント

関数内で各 π 回使われる変数があり, 関数自体を σ 回使うとすると, 文脈の中で変数は計 $\sigma\pi$ 回使われる.

$$\frac{\Gamma, x \overset{\sigma\pi}{:} S \vdash M \overset{\sigma}{:} T}{\Gamma \vdash \lambda x \overset{\pi}{:} S.M^T \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T} \tag{LAM}$$

Section 3 : 圏論的意味論

- 依存型の圏論的意味論に用いる Category with Families (CwF) の導入
- 上記を resource annotation への対応のために拡張した QCwF の導入
- Type Formers のための拡張の方針
- QTT の意味論と健全性 (証明の概要)

Section 4 : Realisability

- BCI-代数の拡張 Linear Combinatory Algebra(LCA) と 線形含意
- R-Linear Comibinatory Algebra (R-LCA) の導入
- Category of Assemblies の導入
- R-LCA から QCwF の構成

表示的意味論

プログラムを部分式毎に数学的な対象に対応させて意味を考える.
全体の意味は部分から合成的に構成される. → 構成的な演繹体系との対応関係が自然

圏論的意味論

表示的意味論の一種. その圏に固有な構造/性質によって, プログラムの構造の本質的な部分を抜き出す.
すると, 基本的には構造を保存する関手を意味を与える map とみなせる.
具体的なモデルの意味を抽象化された数学的構造に変換することで, それを元に新たな解釈に基づく別のモデルを構成することもできる.

→ Atkey(2024) ではリソースを計算資源の使用量から時間量へと再解釈することで, プログラムが多項式時間で動くことを保証するモデルを考えている

補足: 圏論の知識

圏 $\mathbb{C}$ は次のものからなる 4 つ組のこと.

- 対象の集まり $Ob(\mathbb{C})$
- 各 $A, B \in Ob(\mathbb{C})$ について A から B への射の集まり $\mathbb{C}(A, B)$
- 各 $A \in Ob(\mathbb{C})$ に対し, 射 $id_A \in \mathbb{C}(A, A)$
- 各 $A, B, C \in Ob(\mathbb{C})$ に対し, 写像 $\circ : \mathbb{C}(B, C) \times \mathbb{C}(A, B) \to \mathbb{C}(A, B)$ で結合律と単位律を満たすもの

圏 $\mathbb{C}, \mathbb{D}$ について, $\mathbb{C}$ から $\mathbb{D}$ への関手とは,

- 写像 $A \mapsto F(A) : Ob(\mathbb{C}) \to Ob(\mathbb{D})$
- 各 $A, B \in Ob(\mathbb{C})$ に対し, 写像 $f \mapsto F(f) : \mathbb{C}(A, B) \to \mathbb{D}(F(A), F(B))$

の組で, $F(id_A) = id_{F(A)}$ および $F(g \circ f) = F(g) \circ F(f)$ をみたすもの.

他に出てくる概念としては自然変換, 終対象, pullback だけ.

Martin Hofmann の Syntax and Semantics of Dependent Types(1997) における定義/記法に基づいているのでそちらの内容を参考に説明する.

- 項モデルにおける意味の概念として context morphism と generalised substitution (§ 2.4)
- CwF の定義 (§ 3.1)
- type former のための拡張 (§ 3.3)
- 文法の解釈, 健全性 (§ 3.5)

後ろ三つの項目に関しては, それぞれ依存型のための内容を Hofmann の論文で見て, QTT のための拡張を Atkey の論文で見るという流れを繰り返す.

CwF でやろうとしていること

まずは CwF によって依存型体系の構文論に対する意味の核の部分を与える. このときには項として変数の場合のみを考える. そこに各 type former の解釈のための要素を適宜追加していく.

定義 2.11 context morphism

$\Gamma \vdash f \Rightarrow \Delta$ とは, valid context Γ と $\Delta \stackrel{\text{def}}{=} x_1 : \sigma_1, \ldots, x_n : \sigma_n$, $f \stackrel{\text{def}}{=} (M_1, \ldots, M_n)$ について, $\Gamma \vdash M_i : \sigma_i[M_1/x_1] \ldots [M_{i-1}/x_{i-1}]$ $(i = 1, 2, \cdots, n)$ が成り立つことを表す. f は Γ から Δ への context morphism であるという.

- 任意の文脈 Γ から空の文脈への一意な射 $() :: \Gamma \to \diamond$ がある.
- $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$ として, $\Gamma \vdash \sigma\ type$ かつ x が fresh ならば射 $(x_1, \ldots, x_n)$ が $p(\Gamma, \sigma) : \Gamma, x : \sigma \to \Gamma$ をなす.
- 任意の項 $\Gamma \vdash M : \sigma$ について, $\Gamma \vdash \overline{M} \Rightarrow \Gamma, x : \sigma$ となる $\overline{M} \equiv (x_1, \ldots, x_n, M)$ が構成できる. ただし, $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$ とする.

---

valid context のルール

$$\frac{}{\vdash \diamond\ ctxt}\text{C-Emp} \qquad \frac{\Gamma \vdash \sigma\ type}{\vdash \Gamma, x : \sigma\ ctxt}\text{C-Ext} \qquad \frac{\vdash \Gamma = \Delta\ ctxt \quad \Gamma \vdash \sigma = \tau\ type}{\vdash \Gamma, x : \sigma = \Delta, y : \tau\ ctxt}\text{C-Ext-Eq}$$

変数 x, y は fresh であるとする.

変数の付け替えによる等価な文法を $\equiv$ で表す.

$\Gamma \vdash f \Rightarrow \Delta$ と pre-type τ に関して, $\Delta \equiv x_1 : \sigma_1, \ldots, x_n : \sigma_n$, $f \equiv (M_1, \ldots, M_n)$ ならば τ に対する同時置換を次のように定める.

$$\tau[f/\Delta] \equiv \tau[M_1/x_1][M_2/x_2] \ldots [M_n/x_n]$$

他の pre-terms,pre-contexts, 判断 $\mathcal{J}$ で, $M : \sigma, \sigma\ type, M = N : \sigma = \tau\ type$ という形のものについても同様に $-[f/\Delta]$ を定義する. Δ が明らかな場合は省略して $\tau[f]$ などとかく.

命題 2.12

$\Gamma \vdash f \Rightarrow \Delta$ かつ $\Delta, \Theta \vdash \mathcal{J}$ ならば $\Gamma, \Theta[f/\Delta] \vdash \mathcal{J}[f/\Delta]$

identity

$\Gamma \vdash id_\Gamma \Rightarrow \Gamma$ が $id_\Gamma \equiv (x_1, \ldots, x_n)$ として存在する.

composition

$\Gamma \vdash f \Rightarrow \Delta, \Delta \vdash g \Rightarrow \Theta$, $g \equiv (N_1, \ldots, N_k)$ について $g \circ f \equiv (N_1[f], \ldots, N_k[f])$ として合成できる. $\sigma[g \circ f] \equiv \sigma[g][f], M[g \circ f] \equiv M[g][f]$ が成り立ち, 結合律と単位律を満たす.

---

context morphism の定義

$\Gamma \vdash f \Rightarrow \Delta$ は $\Delta \overset{\text{def}}{=} x_1 : \sigma_1, \ldots, x_n : \sigma_n$, $f \overset{\text{def}}{=} (M_1, \ldots, M_n)$ について,
$\Gamma \vdash M_i : \sigma_i[M_1/x_1] \ldots [M_{i-1}/x_{i-1}]$ $(i = 1, 2, \cdots, n)$ が成り立つこと.

CwF は 6 つ組 $(C, Ty, Tm, \top, -.-, \langle -, - \rangle)$ で定義される.

記号 $\Gamma \in C$, $f : \Gamma \to \Delta$,$\sigma \in Ty(\Delta)$

Table: CwF と項モデルの対応 1. 文法の要素

CwF 側の概念	項モデルにおける対応物
対象	文脈
射	context morphism
$Ty(\Gamma)$	$\Gamma \vdash \sigma$ なる pre-type σ の集合
$Tm(\Gamma, \sigma)$	$\Gamma \vdash M : \sigma$ なる pre-term M の集合

正確には, 上記の対応物の definitional equality (型の規則の中では $=$ 記号であらわされているもの) に関する同値類に対応. definitional equality は変数記号に x を使うか y を使うかだけ違うなどの, メタレベルでわかる等しさのこと. これらは同じ意味を持ってほしいので, 同値類をまとめて一つの対象に対応させる.

Table: CwF と項モデルの対応 2. 型理論のルールを解釈するための定数と操作

CwF 側の概念	項モデルにおける対応物
$-\{f\} : Ty(\Delta) \to Ty(\Gamma)$	型に対する同時置換
$-\{f\} : Tm(\Delta, \sigma) \to Tm(\Gamma, \sigma\{f\})$	項に対する同時置換
終対象 $\top$	空の文脈

- 項モデルにおいて代入は同時置換: $\sigma\{f\} := \sigma[f]$ かつ $M\{f\} := M[f]$.
  - 命題 2.12 より $\Gamma \vdash f \Rightarrow \Delta$, $\Delta \vdash x : \sigma$ ならば, $\Gamma \vdash x[f] : \sigma[f]$.
  - $\Delta \vdash \sigma\ type, \Gamma \vdash \sigma[f]\ type$ も満たすはず.
  - semantic substitution の適用によって, 恒等射と合成が保存されるようにする. 型の方では $\sigma\{\mathrm{id}\} = \sigma$ かつ $\sigma\{g \circ f\} = \sigma\{g\}\{f\}$
- 任意の Γ に対して, context morphism $() : \Gamma \to \diamond$ はちょうど一つ存在するので $\diamond$ を終対象に対応付ける.

Table: CwF と項モデルの対応 3. 型理論のルールを解釈するための定数と操作 (続)

CwF 側の概念	項モデルにおける対応物
対象 $\Gamma.\sigma$ (σ の comprehension)	context extension $\Gamma, x : \sigma$
射 $\langle f, M \rangle : \Delta' \to \Delta.S$	context morhism (f, M)

- 1 つ目は変数による文脈拡張に対応
- 2 つ目は項による文脈拡張 (の一般化) に対応

素朴な項による文脈拡張

任意の項 $\Gamma \vdash M : \sigma$ について, $\Gamma \vdash \overline{M} \Rightarrow \Gamma, x : \sigma$ となる $\overline{M} \equiv (x_1, \ldots, x_n, M)$ が構成できる. ただし, $\Gamma = x_1 : \sigma_1, \ldots, x_n : \sigma_n$ とする.

より一般には次のようにとらえられる.

$$\frac{\Delta' \vdash f \Rightarrow \Delta \quad \vdash \Delta, x : S\ ctxt \quad \Delta' \vdash M : S[f]}{\Delta' \vdash (f, M) \Rightarrow \Delta, x : S} \text{ (Mor-Cons)}$$

$$\frac{\Delta' \vdash f \Rightarrow \Delta \quad \vdash \Delta, x : S\ ctxt \quad \Delta' \vdash M : S[f]}{\Delta' \vdash (f, M) \Rightarrow \Delta, x : S} \text{ (Mor-Cons)}$$

CwF においては, context morhism (f, M) に対応する概念として
$\langle f, M \rangle : \Delta' \to \Delta.S$ を考えた.
項モデルでは $\overline{M} \equiv (x_1, \ldots, x_n, M)$ を一般化して (f, M) を考えたが, CwF では (f, M) に対応する方を定義に組み込み, そこから $\overline{M}$ に相当するものを作る.

$$\overline{M} \stackrel{\text{def}}{=} \langle \mathsf{id}_\Gamma, M \rangle_\sigma : \Gamma \to \Gamma.\sigma$$

項による文脈拡張についても, 項モデルで考えた射
$\Gamma, x : \sigma \vdash p(\Gamma, \sigma) \equiv (x_1, \ldots, x_n) \Rightarrow \Gamma$ に対応するものを CwF の方で定義できる.

$$\langle p(\sigma), v_\sigma \rangle \stackrel{\text{def}}{=} \mathsf{id}_{\Gamma.S}$$

CwF 側の概念	項モデルにおける対応物
$p(\sigma) : \Gamma.\sigma \to \Gamma$	射 $p(\Gamma, \sigma)$
$v_\sigma \in Tm(\Gamma.\sigma, \sigma\,\{p(\sigma)\})$	項 x

Hofmann(1997) の定義では以下の性質 (1)-(4) を満たすことを要求している. こちらでは p, v も CwF の定義の組に入れて考える. 一方, Atkey の定義では (4) を満たすようにして p, v を定義し, 自然同型を用いて定義することによって残りの性質も保証しているようだ.
記号 $f : \Delta' \to \Delta$, $\langle f, M \rangle_\sigma : \Delta' \to \Delta.\sigma$, $g : B \to \Delta'$.

$$p(\sigma) \circ \langle f, M \rangle_\sigma = f : \Delta' \to \Delta \tag{1}$$
$$v_\sigma \{\langle f, M \rangle_\sigma\} = M \in Tm(\Gamma, \sigma\{f\}) \tag{2}$$
$$\langle f, M \rangle_\sigma \circ g = \langle f \circ g, M\{g\} \rangle_\sigma : B \to \Delta.\sigma \tag{3}$$
$$\langle p(\sigma), v_\sigma \rangle_\sigma = \mathrm{id}_{\Delta.\sigma} : \Delta.\sigma \to \Delta.\sigma \tag{4}$$

$$\frac{\Delta' \vdash f \Rightarrow \Delta \quad \vdash \Delta, x : S\ ctxt \quad \Delta' \vdash M : S[f]}{\Delta' \vdash (f, M) \Rightarrow \Delta, x : S} \qquad \frac{\Delta.S \vdash p(\sigma) \Rightarrow \Delta \quad \vdash \Delta.S\ ctxt \quad \Delta.S \vdash v_\sigma\{p(\sigma)\}}{\Delta.S \vdash \mathrm{id}_{\Delta.S} \Rightarrow \Delta.S}$$

射 $f : X \to Z, g : Y \to Z$ がある. g の f 沿いの pullback とは射の組 $p_1 : P \to X, p_2 : P \to Y$ であって, $f \circ p_1 = g \circ p_2$ となり, 同じように可換図を成り立たせる $q_1 : Q \to X, q_2 : Q \to Y$ について一意な射 $u : Q \to P$ があって, $q_i \circ u = p_i \ (i = 1, 2)$ が成り立つ.

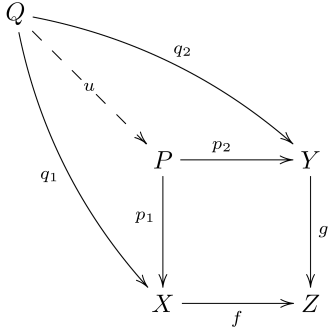

Figure: pullback の可換図[1]

[1] https://en.wikipedia.org/wiki/Pullback_(category_theory)

QTT のための CwF の拡張, QCwF(Atkey(2018) Definition 3.2)

1. CwF $(C, Ty, Tm, \top, -.-, \langle -, - \rangle)$: C の対象が対応するのは, リソース注釈なしの文脈
2. 圏 $\mathcal{L}$ と忠実関手 $U : \mathcal{L} \to C$: 圏 $\mathcal{L}$ があり, その対象がリソース注釈ありの文脈に対応. U は注釈を消す操作に対応している. なぜ忠実である必要があるのかはあまりわかってない.
3. (Addition structure)
   - $\mathcal{L} \times_C \mathcal{L}$ を $\mathcal{L} \xrightarrow{U} C \xleftarrow{U} \mathcal{L}$ の pullback であるとする.
   - 関手 $(+) : \mathcal{L} \times_C \mathcal{L} \to \mathcal{L}$ であって, $U(\Gamma_1 + \Gamma_2) = U\Gamma_1 = U\Gamma_2$ を満たすものをリソース付き文脈の和を解釈するために使う.
   - pullback からの関手として注釈付き文脈の加算を考えることで, 加算できる文脈は注釈を消すと同じものであるという $0\Gamma_1 = 0\Gamma_2$ の制約を反映している.
   - ある対象 $\diamond \in \mathcal{L}$ があって $U\diamond = \top$.

4. (Scaling structure) 各 $\rho \in R$ について関手 $\rho(-) : \mathcal{L} \to \mathcal{L}$ があり, $U(\rho(-)) = U(-)$ を満たす. 注釈を何倍しても, 消してしまえば同じ文脈になってほしい.
5. (Resourced Terms) $\Gamma \in \mathcal{L}$ と $S \in Ty(U\Gamma)$ について, semantic resourced terms $RTm(\Gamma, S)$ が, 単射の関数 $U_{\Gamma.S} : RTm(\Gamma, S) \to Tm(U\Gamma, S)$ とともにある. 各射 $\Gamma' \to \Gamma$ と $S \in Ty(U\Gamma)$ について, $-\{f\} : RTm(\Gamma, S) \to RTm(\Gamma', S\{f\})$ が存在し, $U(M\{f\}) = (UM)\{Uf\}$ を満たす.
   Tm の場合と同様の定義, 型にリソースはつかないので RTy はない.
   項に代入してからリソースを消しても, 項と context morphism からリソースを消したうえで代入しても結果は同じ.

6 (Resourced context extension)
$\Gamma \in \mathcal{L}$, $\rho \in R, S \in Ty(U\Gamma)$ について, 対象 $\Gamma.\rho S \in \mathcal{L}$ があって,
$U(\Gamma.\rho S) = U\Gamma.S$ となり, 以下のような自然変換が存在する.

$$
\begin{aligned}
emp_\pi &: \diamond \to \pi\diamond \\
emp_+ &: \diamond \to \diamond + \diamond \\
ext_\pi &: \pi\Gamma.(\pi\rho)S \to \pi(\Gamma.\rho S) \\
ext_+ &: (\Gamma_1 + \Gamma_2).(\rho_1 + \rho_2)S \to \Gamma_1.\rho_1 S + \Gamma_2.\rho_2 S
\end{aligned}
$$

これらは $U(emp_\pi) = U(emp_+) = \mathrm{id} : \top \to \top, U(ext_\pi) = \mathrm{id}, U(ext_+) = \mathrm{id}$
を満たす.

文脈の積 (i) $\pi(\diamond) = \diamond$ (ii) $\pi(\Gamma, x \overset{\rho}{:} S) = \pi\Gamma, x \overset{\pi\rho}{:} S$

文脈の和 (i) $\diamond + \diamond = \diamond$ (ii) $(\Gamma_1, x \overset{\rho_1}{:} S) + (\Gamma_2, x \overset{\rho_2}{:} S) = (\Gamma_1 + \Gamma_2), x \overset{\rho_1 + \rho_2}{:} S$

任意の $\sigma \in Ty(\Gamma), \tau \in Ty(\Gamma.\sigma)$ について型 $\prod(\sigma,\tau) \in Ty(\Gamma)$ があり, 各 $M \in Tm(\Gamma.\sigma,\tau)$ について項 $\lambda_{\sigma,\tau}(M) \in Tm(\Gamma, \prod(\sigma,\tau))$ がある. また, 任意の $M \in Tm(\Gamma, \Pi(\sigma,\tau)), N \in Tm(\Gamma,\sigma)$ について項 $App_{\sigma,\tau}(M,N) \in Tm(\Gamma, \tau\left\{\bar{N}\right\})$ が存在し, 以下の性質を満たす.

$$
\begin{aligned}
&App_{\sigma,\tau}(\lambda_{\sigma,\tau}(M),N) = M\left\{\bar{N}\right\}\\
&\Pi(\sigma,\tau)\left\{f\right\} = \Pi(\sigma\left\{f\right\}, \tau\left\{q(f,\sigma)\right\})\\
&\lambda_{\sigma,\tau}(M)\left\{f\right\} = \lambda_{\sigma\{f\},\tau\{q(f,\sigma)\}}(M\left\{q(f,\sigma)\right\}\\
&App_{\sigma,\tau}(M,N)\left\{f\right\} = App_{\sigma\{f\},\tau\{q(f,\sigma)\}}(M\left\{f\right\}, N\left\{f\right\})
\end{aligned}
$$

関数適用による引数への項の代入を表す射として App コンビネータを考える別の定義もできる. こちらについてはあとで少し触れる.

Dependent function space

$$\frac{\Gamma \vdash \sigma\ type \quad \Gamma, x:\sigma \vdash \tau\ type}{\Gamma \vdash \Pi x:\sigma.\tau\ type}\ \Pi\text{-F} \qquad \frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash \lambda x:\sigma.M^{\tau} : \Pi x:\sigma.\tau}\ \Pi\text{-I} \qquad \frac{\Gamma \vdash M : \Pi x:\sigma.\tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash App_{[x:\sigma]\tau}(M,N) : \tau[N/x]}\ \Pi\text{-E}$$

Def. 3.5.

任意の $\Delta \in C, S \in Ty(\Delta), T \in Ty(\Delta.S)$ と $\pi \in R$ について semantic type $\Pi\pi ST \in Ty(\Delta)$ があり, Δ に関する自然同型 $\Lambda : Tm(\Delta.S, T) \cong Tm(\Delta, \Pi\pi ST)$ がある.
0 fragment における解釈のためには $M \in Tm(\Delta, \Pi\pi ST)$ と $N \in Tm(\Delta, S)$ について $App^{0,0}_{\Delta,S,T} = (\Lambda^{-1}(M))\left\{\bar{N}\right\} \in Tm(\Delta, T\left\{\bar{N}\right\})$ として定める.

Def. 3.6.

対象 Γ に関する自然同型 $\Lambda_{\mathcal{L}} : RTm(\Gamma.\pi S, T) \cong RTm(\Gamma, \Pi\pi ST)$ があり,
$U \circ \Lambda_{\mathcal{L}} = \Lambda \circ U$ かつ $U \circ \Lambda^{-1}_{\mathcal{L}} = \Lambda^{-1} \circ U$ を満たす.

$$\frac{0\Gamma \vdash S \quad 0\Gamma, x \overset{0}{:} S \vdash T}{0\Gamma \vdash (x \overset{\pi}{:} S) \to T}(\text{PI}) \qquad \frac{\Gamma, x \overset{\sigma\pi}{:} S \vdash M \overset{\sigma}{:} T}{\Gamma \vdash \lambda x \overset{\pi}{:} S.M^T \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T}(\text{LAM})$$

$$\frac{\Gamma_1 \vdash M \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T \quad \Gamma_2 \vdash N \overset{\sigma'}{:} S \quad 0\Gamma_1 = 0\Gamma_2, \quad \sigma' = 0 \Leftrightarrow (\pi = 0 \vee \sigma = 0)}{\Gamma_1 + \pi\Gamma_2 \vdash \mathsf{App}_{(x \overset{\pi}{:} S)T}(M, N) \overset{\sigma}{:} T[N/x]}(\text{APP})$$

$\pi = 1$ の場合

$M \in RTm(\Gamma_1, \Pi\pi ST), N \in RTm(\Gamma_2, S), U\Gamma_1 = U\Gamma_2$

$App^{11}_{\Gamma_1,\Gamma_2,\pi,S,T}(M,N) = \Lambda^{-1}_{\Gamma_1,\pi,S,T}(M)\left\{\overline{\pi N}\right\} \in RTm(\Gamma_1 + \pi\Gamma_2, T\left\{\overline{UN}\right\})$

$\pi = 0$ の場合

$M \in RTm(\Gamma, \Pi 0 ST), N \in Tm(U\Gamma, S)$

$App^{10}_{\Gamma,S,T}(M,N) = \Lambda^{-1}_{\Gamma,0,S,T}(M)\left\{\overline{N}\right\} \in RTm(\Gamma, T\left\{U\overline{N}\right\})$

$U(App^{1x}(M,N)) = App^{00}(UM, UN)$ for $x \in \{0,1\}$ が必要.

$$\frac{0\Gamma \vdash S \quad 0\Gamma, x \overset{0}{:} S \vdash T}{0\Gamma \vdash (x \overset{\pi}{:} S) \to T}(\mathsf{PI}) \qquad \frac{\Gamma, x \overset{\sigma\pi}{:} S \vdash M \overset{\sigma}{:} T}{\Gamma \vdash \lambda x \overset{\pi}{:} S.M^T \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T}(\mathsf{LAM})$$

$$\frac{\Gamma_1 \vdash M \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T \quad \Gamma_2 \vdash N \overset{\sigma'}{:} S \qquad 0\Gamma_1 = 0\Gamma_2, \quad \sigma' = 0 \Leftrightarrow (\pi = 0 \vee \sigma = 0)}{\Gamma_1 + \pi\Gamma_2 \vdash \mathsf{App}_{(x \overset{\pi}{:} S)T}(M,N) \overset{\sigma}{:} T[N/x]}(\mathsf{APP})$$

直感的な CwF と型理論の概念の対応を形式化し, 健全性を示す.
partial interpretation function $[\![-]\!]$ であって, 次のようなマッピングを持つもの

- pre-context から C の対象
- pre-context Γ と pre-type σ の組 $\Gamma;\sigma$ から $Ty([\![\Gamma]\!])$ のなかの族
- pre-context Γ と pre-term M の組 $\Gamma;M$ から, ある $\sigma \in Ty([\![\Gamma]\!])$ について $Tm(\sigma)$ の項.

$$[\![\diamond]\!] = \top$$
$$[\![\Gamma, x:\sigma]\!] = [\![\Gamma]\!].[\![\Gamma;\sigma]\!] \quad \text{(if x not in } \Gamma\text{, o.w. undefined)}$$
$$[\![\Gamma;\Pi x:\sigma.\tau]\!] = \Pi([\![\Gamma;\sigma]\!],[\![\Gamma,x:\sigma;\tau]\!])$$
$$[\![\Gamma,x:\sigma;x]\!] = v_{[\![\Gamma;\sigma]\!]}$$
$$[\![\Gamma,x:\sigma,\Delta,y:\tau;x]\!] = [\![\Gamma,x:\sigma,\Delta;x]\!]\,\{p([\![\Gamma,x:\sigma,\Delta;\tau]\!])\}$$
$$[\![\Gamma;App_{[x:\sigma]\tau}(M,N)]\!] = App_{[\![\Gamma;\sigma]\!],[\![\Gamma,x:\sigma;\tau]\!]} \circ \left\langle \overline{[\![\Gamma;M]\!]}, [\![\Gamma;N]\!]^{+} \right\rangle_{[\![\Gamma;\Pi x:\sigma.\tau]\!]^{+}}$$
$$[\![\Gamma;\lambda x:\sigma.M^{\tau}]\!] = \lambda_{[\![\Gamma;\sigma]\!],[\![\Gamma,x:\sigma;\tau]\!]}([\![\Gamma,x:\sigma;M]\!])$$

ここでの $+$ 記号は弱化に対応する context morphism の代入の省略で, ここでは $p([\![\Gamma, x:\sigma;\tau]\!])$ の代入を表している. 右辺の $App_{\sigma,\tau} : \Gamma.\sigma.\Pi(\sigma,\tau)^+ \to \Gamma.\sigma.\tau$ は関数適用による引数への項の代入を表す射. App の項に対するマップの定義で, 右辺が射になっているが Π-E 規則との対応を考えると $Tm(\Gamma, \tau\{\bar{N}\})$ に属する項のはず.

$$[\![\Gamma; App_{[x:\sigma]\tau}(M,N)]\!] = App_{[\![\Gamma;\sigma]\!],[\![\Gamma,x:\sigma;\tau]\!]} \circ \left\langle \overline{[\![\Gamma; M]\!]}, [\![\Gamma; N]\!]^+ \right\rangle_{[\![\Gamma;\Pi x:\sigma.\tau]\!]^+}$$

命題 3.16 の修正 (App コンビネータ)

原文では application combinator として, $M \in Tm(\Pi(\sigma,\tau)), N \in Tm(\sigma)$ に対し $g :\equiv App_{\sigma,\tau} \circ \langle N, M^+ \rangle_{\Pi(\sigma,\tau)^+}$ として, $v_\tau\{g\} \in Tm(\tau\{p(\tau) \circ g\}) = Tm(\tau\{\bar{N}\})$ があるとされている. → g のなかの N を $\bar{N}$ に直す.

これを参考に, 次のようにマップを修正する.

$$[\![\Gamma; App_{[x:\sigma]\tau}(M,N)]\!] = v_{[\![\Gamma,x:\sigma;\tau]\!]} \left\{ App_{[\![\Gamma;\sigma]\!],[\![\Gamma,x:\sigma;\tau]\!]} \circ \left\langle \overline{[\![\Gamma; N]\!]}, [\![\Gamma; M]\!]^+ \right\rangle_{[\![\Gamma;\Pi x:\sigma.\tau]\!]^+} \right\}$$

$$\frac{\Gamma \vdash \sigma\ type \quad \Gamma, x:\sigma \vdash \tau\ type}{\Gamma \vdash \Pi x:\sigma.\tau\ type}\ \Pi\text{-F} \qquad \frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash \lambda x:\sigma.M^\tau : \Pi x:\sigma.\tau}\ \Pi\text{-I} \qquad \frac{\Gamma \vdash M:\Pi x:\sigma.\tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash App_{[x:\sigma]\tau}(M,N) : \tau[N/x]}\ \Pi\text{-E}$$

定理 3.35.

1. $\vdash \Gamma\ ctxt$ ならば $[\![\Gamma]\!]$ は C の対象
2. $\Gamma \vdash \sigma$ ならば $[\![\Gamma; \sigma]\!]$ は $Ty([\![\Gamma]\!])$ の要素である.
3. $\Gamma \vdash M : \sigma$ ならば $[\![\Gamma; M]\!]$ は $Tm([\![\Gamma; \sigma]\!])$ の要素
4. $\vdash \Gamma = \Delta\ ctxt$ ならば $[\![\Gamma]\!] = [\![\Delta]\!]$
5. $\Gamma \vdash \sigma = \tau\ type$ ならば $[\![\Gamma; \sigma]\!] = [\![\Gamma; \tau]\!]$
6. $\Gamma \vdash M = N : \sigma$ ならば $[\![\Gamma; M]\!] = [\![\Gamma; N]\!]$

証明の概略

基本的には導出木の深さに関する帰納法で, 文脈/型/項の規則に従った場合分けをしていけばよい. しかし, 構文上の弱化/代入と意味上の代入 $- \{-\}$ を協調させるための代入補題を示す部分は難しい. 詳細を見るところまでいけなかった.

解釈 $[\![\Delta; M]\!]$ は usage annotation なし precontext Δ に対応する usage annotated precontext Γ と resourced semantic term t の組を生む. $[\![\lfloor\Gamma\rfloor; M]\!]$ は注釈付き文脈として Γ を返すような resourced semantic term へのマップを表す.

定理 3.7.

1. $\vdash \Gamma\ ctxt$ ならば $[\![\Gamma]\!]$ は $\mathcal{L}$ の対象
2. $\Gamma \vdash S$ ならば $[\![\Gamma; S]\!]$ は $Ty(U[\![\Gamma]\!])$ の要素である.
3. $\Gamma \vdash S \equiv T$ ならば $[\![\Gamma; S]\!] = [\![\Gamma; T]\!]$
4. $\Gamma \vdash M \overset{0}{:} S$ ならば $[\![\Gamma; M]\!]$ は $Tm(U[\![\Gamma]\!], [\![\Gamma; S]\!])$ の要素
5. $\Gamma \vdash M \equiv N \overset{0}{:} S$ ならば $[\![\Gamma; M]\!] = [\![\Gamma; N]\!]$
6. $\Gamma \vdash M \overset{1}{:} S$ ならば $[\![\lfloor\Gamma\rfloor; M]\!]$ は $RTm([\![\Gamma]\!], [\![\Gamma; S]\!])$ の要素
7. $\Gamma \vdash M \equiv N \overset{1}{:} S$ ならば $[\![\lfloor\Gamma\rfloor; M]\!] = [\![\lfloor\Gamma\rfloor; N]\!]$

導入

- 数学の証明では, ときとして非構成的な証明がなされることがある.
- 構成的な証明とは, 存在量化されている命題 $\exists x.P(x)$ が真であるのを示すのに, 具体的に $P(t)$ が真となるような t を提示するもの.
- 存在量化されている命題 $\exists x.P(x)$ が真であると証明できても, 実際にそれを成り立たせる $P(t)$ なる t を構成できるとは限らない
- 古典論理の排中律や二重否定除去などを使うと非構成的

動機

もっと "構成的" に命題に対応する "計算的手続き" を考えられるようにしたい.

モデルの作り方

構成的なものだけを扱う直観主義論理に関する BHK 解釈が重要な役割を果たす.

参考: Realisability. Thomas Streicher. https://www2.mathematik.tu-darmstadt.de/~streicher/REAL/REAL.pdf

Brouwer-Heyting-Kolmogorov (BHK) 解釈

- $A \wedge B$ の証拠 (witness) は, A の証拠と B の証拠の組
- $A \Rightarrow B$ の証拠は A の証拠をもらって B の証拠を返す関数
- $A \vee B$ は組 $(0, \text{A の証拠})$ か $(1, \text{B の証拠})$
- $\forall x.A(x)$ の証拠は任意の c に対し, $A(c)$ の証拠を返す関数
- $\exists x.A(x)$ はある要素 c と $A(c)$ の証拠の組

BHK 解釈をもとにした, Curry-Howard 同型対応の方が馴染み深いので比較.

	BHK 解釈	実現可能性	Curry-Howard 対応
対象の体系	直観主義論理	任意の計算モデル	型付き λ 計算
証拠	構成的なナニカ	適当な数学的構造 (realiser)	λ 項

当初 realiser として元は万能チューリング機械のコード (に相当するな自然数) を考えた. 以降では λ 計算やコンビネータ代数を用いることが多い. 実現可能性を記述するのには Category of Assemblies をよく使う.

3 章の圏論的意味論と比較して, より実行時の計算に焦点を当てたものであること. QTT においてはリソース注釈が実際の計算にどのように反映されるのかというところを形式化するためにこれを考える.

Section 4：Realisability

- BCI-代数の拡張 Linear Combinatory Algebra(LCA) と 線形含意
- R-Linear Comibinatory Algebra (R-LCA) の導入
- Category of Assemblies の導入
- R-LCA から QCwF の構成

線形論理とは

(古典論理の) シークエント計算から弱化 (weakening) 規則と縮約 (contraction) 規則を除いた論理体系. 「資源としての仮説」という解釈の下で, 全ての仮説は証明において「一回だけ」消費される.

推論規則の一部抜粋 [1]

直感的な解釈として, $\langle A \rangle$ は A という資源がちょうど 1 回使えることを表す.

$$\frac{}{\langle A \rangle \vdash A}\ \langle \mathrm{Id} \rangle$$

線形含意 $A \multimap B$ は直感的には A を消費すると B がえられることを表している.

$$\frac{\Gamma, \langle A \rangle \vdash B}{\Gamma \vdash A \multimap B}\ (\multimap\text{-I}) \qquad \frac{\Gamma \vdash A \multimap B \quad \Delta \vdash A}{\Gamma, \Delta \vdash B}\ (\multimap\text{-E})$$

[1] A taste of linear logic*. Philip Wadler.
https://homepages.inf.ed.ac.uk/wadler/papers/lineartaste/lineartaste-revised.pdf

Geometry of Interaction (GoI)

GoI は線形論理のとくに計算的な側面をモデル化する意味論. カット除去を動的なプロセス間の情報の流れとして解釈する. トレース付きモノイダル圏によって GoI を構成することができる.

- QTT でのリソース使用の意味をこの GoI におけるトークンのやりとりに対応付けて考えることで, 数量の指定が計算にどのように反映されるかを形式化する.
- GoI は次に紹介する LCA というコンビネータ代数でモデル化できる[1].
- QTT に対してはリソース注釈のための拡張を施した R-LCA を考える.

[1] Geometry of Interaction and Linear Combinatory Algebras. S. Abramsky, E. Haghverdi & P.J. Scott. https://www.researchgate.net/publication/220173613_Geometry_of_Interaction_and_Linear_Combinatory_Algebras

Linear Combinatory Algebra, LCA

LCA $(A, \cdot, !)$ とは, 適用構造 $(A, \cdot)$, 1 項演算子 $! : A \to A$ および区別される A の要素 $B, C, I, K, W, D, \delta, F$ から成る.

規則	LCA の演算	線形論理
Composition, Cut	$Bxyz = x(yz)$	$B : (\beta \multimap \gamma) \multimap (\alpha \multimap \beta) \multimap \alpha \multimap \gamma$
Exchange	$Cxyz = (xz)y$	$C : (\alpha \multimap \beta \multimap \gamma) \multimap \beta \multimap \alpha \multimap \gamma$
Identity	$Ix = x$	$I : \alpha \multimap \alpha$
Weakening	$Kx!y = x$	$K : \alpha \multimap !\beta \multimap \alpha$
Contraction	$Wx!y = x!y!y$	$W : (!\alpha \multimap !\alpha \multimap \beta) \multimap !\alpha \multimap \beta$
Dereliction	$D!x = x$	$D : !\alpha \multimap \alpha$
Comultiplication	$\delta!x = !!x$	$\delta : !\alpha \multimap !!\alpha$
Monoidal Functoriality	$F!x!y = !(xy)$	$F : !(\alpha \multimap \beta) \multimap !\alpha \multimap !\beta$

純粋線形 λ 計算に指数的結合子!(様相論理の必然性演算子に相当, 資源的解釈では任意生産を表す) を追加した体系に対してコンビネータ完全.

R-LCA

R-LCA は, BCI 代数 $(A, \cdot, B, C, I)$, (任意の $\rho \in R$ に対する)1 項演算子 $!_\rho : A \to A$ および区別される A の要素 $K, W_{\pi\rho}, D, \delta_{\pi\sigma}, F_\rho$ から成る.

規則	LCA の演算	R-LCA の演算
Weakening	$Kx!y = x$	$K \cdot x \cdot !_0 y = x$
Contraction	$Wx!y = x!y!y$	$W_{\pi\rho} \cdot x \cdot !_{\pi+\rho} y = x \cdot !_\pi y \cdot !_\rho y$
Dereliction	$D!x = x$	$D \cdot !_1 x = x$
Comultiplication	$\delta!x = !!x$	$\delta_{\pi\rho} \cdot !_{\pi\rho} x = !_\pi !_\rho x$
Monoidal Functoriality	$F!x!y = !(xy)$	$F_\rho \cdot !_\rho x \cdot !_\rho y = !_\rho (xy)$

また, 全ての $x, y \in \mathcal{A}$ について $!_0 x = !_0 y$ を要求.

- 対象はアセンブリ $\Gamma = (|\Gamma|, \vDash_\Gamma)$.
  - $|\Gamma|$ は集合.
  - 二項関係 $\vDash_\Gamma \subseteq \mathcal{A} \times |\Gamma|$ について $a \vDash_\Gamma \gamma$ は, a が γ を実現すると読める.
  - 各 γ に対して少なくとも一つ $a \vDash_\Gamma \gamma$ となる a が存在する
    (というのが通常の定義だが, QTT の場合はこれが不適当であるため, 修正案が Atkey(2023) の中で提案されている)

- $(|\Gamma|, \vDash_\Gamma)$ から $(|\Gamma'|, \vDash_{\Gamma'})$ への射は, 実現可能な関数 $f : |\Gamma| \to |\Gamma'|$.
  - 実現可能な関数とは, ある $a_f \in \mathcal{A}$ が存在して, すべての $\gamma \in |\Gamma|, a_\gamma \in \mathcal{A}$ に対し $a_\gamma \vDash_\Gamma \gamma$ ならば $a_f \cdot a_\gamma \vDash_{\Gamma'} f(\gamma)$ を満たすようなもののこと.

$C = \mathrm{Set}, L = \mathrm{Ass}(\mathcal{A})$ とする.

文脈の解釈

文脈 Γ はアセンブリ $(|\Gamma|, \vDash_\Gamma)$ に対応付けられる.

関手 $U : \mathcal{L} \to C$ は 台集合と関数へのマップ. (forgetful functor なので忠実)

文脈の構成に関して

- scaling : $\pi\Gamma = (|\Gamma|, \vDash_{\pi\Gamma})$, $x \vDash_{\pi\Gamma} \gamma \Leftrightarrow \exists y. x = !_\pi y \wedge y \vDash_\Gamma \gamma$.
- addition : $\Gamma_1 = (|\Gamma|, \vDash_{\Gamma_1}), \Gamma_2 = (|\Gamma|, \vDash_{\Gamma_2})$ について, $\Gamma_1 + \Gamma_2 = (|\Gamma|, \vDash_{\Gamma_1+\Gamma_2})$ とする. ただし, $x \vDash_{\Gamma_1+\Gamma_2} \gamma \Leftrightarrow \exists y, z. x = [y, z] \wedge y \vDash_{\Gamma_1} \gamma \wedge z \vDash_{\Gamma_2} \gamma$
- empty context : $\diamond = (\{*\}, \vdash_\diamond)$, $x \vDash_\diamond * \Leftrightarrow x = I$

一般の文脈に対する解釈は空文脈に対する解釈と, 文脈拡張 (文脈に変数を追加する操作) に対する解釈から帰納的に定められる.

型の解釈

semantic type の集まり $Ty(\Delta)$ の要素は Δ をインデックスとしたアセンブリの族 $\{(|S(\delta)|, \vDash_{S(\delta)})\}_{\delta \in \Delta}$ (S はある型)

- 集合 Δ の要素は文脈の各変数に対応する項の組としてとらえるのが一般的. ここではより抽象的に文脈の状態を指定するナニカ? これがインデックスになっているのは, 型が文脈の値に依存しうる依存型の体系だから
- 型の解釈で考えるアセンブリはあくまで型の要素 (項) に対する realiser を考えるための枠組みを提供するもので, 型自体に対する realiser を考えているわけではない

Unresourced Semantic Terms

集合 Δ とアセンブリ族 $S \in Ty(\Delta)$ について, semantic term $Tm(\Delta, S)$ は型 $\forall \delta \in \Delta. |S(\delta)|$ の関数から成る.

Resourced Context Extension

$$|\Gamma.\rho S| = \{(\gamma, s) \mid \gamma \in |\Gamma|, s \in |S(\gamma)|\}$$
$$x \vDash_{\Gamma.\rho S} (\gamma, s) \Leftrightarrow \exists y, z.\ x = [y, !_\rho z] \wedge y \vDash_\Gamma \gamma \wedge z \vDash_{S(\gamma)} s$$

Resourced Semantic Terms

$$RTm(\Gamma, S) = \left\{ f : \forall \gamma \in |\Gamma|.|S(\gamma)| \mid \exists x.\forall \gamma \in |\Gamma|, y \in \mathcal{A}.y \vDash_\Gamma \gamma \Rightarrow x \cdot y \vDash_{S(\gamma)} f(\gamma) \right\}$$

例えば $v_{\Gamma.S} \in RTm(0\Gamma.1S, S\,\{p_{U\Gamma.S}\})$ について. 以下で $y =!_0 y'$ である.

$[y, !_1 z] \vDash_{0\Gamma.1S} (\gamma, s) \Leftrightarrow y \vDash_{0\Gamma} \gamma$かつ $z \vDash_{S(\gamma)} s$
$x = \lambda^*[y, !_1 z].\, K \cdot (D \cdot !_1 z) \cdot y$ として $x \cdot [y, !_1 z] \equiv z \vDash_{S(\gamma)} s$ より
$f : |0\Gamma.1S| \ni (\gamma, s) \mapsto s \in |S(\gamma)|$

規則	LCA の演算	R-LCA の演算
Weakening	$Kx!y = x$	$K \cdot x \cdot !_0 y = x$
Dereliction	$D!x = x$	$D \cdot !_1 x = x$

$\lambda^* x.M$ は項 $\lambda x.M$ をコンビネータにより変数 x が含まれない形に変換したもの

Idris

- Idris は first class type を特徴とする純粋関数型言語.
- Haskell 風の文法で, 依存型を持つ.

Idris2

Idris1 ではどの項が実行時に必要か区別できないという問題があった.
→ これを解決するために QTT をコアに据えた
QTT の動機 (再掲): 依存型のある体系ではしばしば型検査にしか使われず, 実行時の計算には必要のない項が生じる. 実行時の計算に使われる回数を線形型の使用回数とすれば, 型を見ると実行時に「使う (present)」それとも「使わない (erased)」かを判別できる.

- 数量は $\{0, 1, \omega\}$ で区別される. デフォルトは制限なしの ω.
- 数量制限 ω の項は $0, 1, \omega$ のどの数量の引数にも渡せる.
- 型にしか現れない変数の数量は 0.