線形論理 CLL_e のモデルを用いた意味論

5AH－1

南澤 吉昭　米崎 直樹

東京工業大学情報理工学研究科計算工学専攻

1 はじめに

古典論理と異なり、線形論理 [6] では命題の個数に言及することが可能である。このことから、計算機科学の分野ではリソースを扱うことのできる論理として注目され、研究されている。

線形論理は古典論理に比べ多くの結合子を持っているため、表現力は高いが、論理式の意味を直感的に理解することは難しい。しかし、計算機科学の分野への応用を考えた場合、わかりやすい意味を与えることが重要である。

線形論理に対する代数的意味論は [6, 3, 8] など幾つか提案されているが、いずれも複雑な代数系を使用しているのでわかりやすいとは言えない。一方、線形論理の直感的な意味論として、ゲーム意味論 [1, 7, 5, 4] や、可能世界意味論 [2] が研究されているが、それぞれ問題を抱えており、決定的なものとはいえない。

本研究では、線形論理に対し、モデルをもとにした、直感的に理解しやすい意味論を提案する。さらにシーケント計算のこの意味論に対する完全性の証明の概略を述べる。

2 線形論理

[8] では線形論理をいくつかの体系に分類している。

CLL_0 は、乗法性結合子と加法性結合子のみを持つ命題線形論理の体系である。この体系に様相結合子 $!, ?$ を加えたものを CLL_e と呼んでいる。さらに、これらを一階の述語論理に拡張したものをそれぞれ CLL_q、CLL と呼んでいる。

本稿では様相結合子を含む命題線形論理 CLL_e を扱う。

定義 2.1 [線形論理 CLL_e] $\mathbb{P}_a$ を原子命題の集合、$\mathbb{P}_c$ を定数の集合 (i.e. $\{\mathbf{0}, \top, \bot\}$) とする。この時、線形論理の式は以下で定義されるものに限る。

1. 原子命題 ($p \in \mathbb{P}_a$) および、定数 ($\mathbf{0}, \top, \bot$) は式。
2. A, B が式の時、$A * B, A \multimap B, A \sqcap B, A \sqcup B, !A$ は式。

ここでは $\sim A, A + B, \mathbf{1}, ?A$ は以下の省略形とする。

$\sim A \equiv A \multimap \mathbf{0}, A + B \equiv \sim A \multimap B, \mathbf{1} \equiv \sim \mathbf{0}, ?A \equiv \sim !\sim A$

3 意味論

定義 3.1 [付値] 付値 v は写像

$$v : \mathbb{P}_a \cup \{\mathbf{0}\} \longrightarrow \mathbf{N} \times \mathbb{Z}$$

(Nは非負の整数の集合、$\mathbb{Z}$ は整数の集合) のうち、

$$v(p) = (r, b) \Rightarrow -r \leq b$$

を満たすものである。

特に、写像 v で $(0,0)$ 以外に写像される原始命題 (および定数 $\mathbf{0}$) からなる集合を $\mathbb{P}(v)$ で表す。

直感的には $v(p) = (r, b)$ の時、r が "$\multimap$" の左側にある命題 p の個数、b が "$\multimap$" の右側にある p の数から左側にある p の数 (r) を引いた値を表している。例えば $a \multimap b$ を「a を消費して b を得る」のように見た場合、r が要求されている p の個数を表し、b が全体の収支を表している。

定義 3.2 [モデル] モデル $\mathcal{M}$ は付値の multiset s の集合である。

定義 3.3 [モデルの演算 $+$]

$$\mathcal{M} + \mathcal{M}' \stackrel{df}{=} \begin{cases} \{s \uplus s' | s \in \mathcal{M}, s' \in \mathcal{M}'\} \cup \{\emptyset\} \\ \qquad [\emptyset \in (\mathcal{M} \cup \mathcal{M}') \text{ の時}] \\ \{s \uplus s' | s \in \mathcal{M}, s' \in \mathcal{M}'\} \\ \qquad [\text{otherwise}] \end{cases}$$

ただし、$s \uplus s'$ は multiset の意味での和集合を表す。

定義 3.4 [付値の演算 $\cdot$] v_1 を $\mathbb{P}(v_1) = \{\mathbf{0}\}, v_1(\mathbf{0}) = (1, 0)$ となる付値とする。$v_i(p) = (r_i, b_i), v_j(p) = (r_j, b_j)$ とする。

1. $p \in \mathbb{P}_a$ の時、
$$(v_i \cdot v_j)(p) \stackrel{df}{=} (\max\{r_i, r_j - b_i\}, b_i + b_j)$$
2. $p = \mathbf{0}$ の時、どちらかが v_1 ならば
$$(v \cdot v_1)(\mathbf{0}) = (v_1 \cdot v)(\mathbf{0}) \stackrel{df}{=} v(\mathbf{0})$$
そうでなければ 1. と同様。

Model based semantics for linear logic CLL_e

Yoshiaki Minamisawa, Naoki Yonezaki

Department of Computer Science, Tokyo Institute of Technology

2-12-1, Ookayama, Meguro-ku Tokyo 152, Japan

この演算 · では結合法則は成立するが、交換法則は成立しない。

直感的には、例えば $a \multimap b$ を意味する付値と $b \multimap c$ を意味する付値がある時に、これらを合成し $a \multimap c$ を意味する付値にするのがこの演算 · である。

定義 3.5 [*eval*] *eval* は付値の multiset から付値の集合への関数。付値の multiset $s = \{v_1, v_2, \cdots, v_n\}$ の時、

$$eval(s) \stackrel{df}{=} \{(v_{k_1} \cdot v_{k_2} \cdots v_{k_n}) | \; k_1 k_2 \cdots k_n \text{は1から} n \text{までの順列}\}$$

モデル中の付値の multiset には式の構造に関する情報が含まれている。この multiset を利用して命題の真偽を判定する時、multiset から付値の情報を取り出す必要がある。演算 · を利用すれば付値を合成し一つの付値にすることができるが、上述のように演算 · では交換法則が成立しない。つまり、multiset 中の要素の並べ方により取り出せる付値が変わる。

一つの multiset から、その要素の合成により得られる付値をすべて考慮するために、*eval* では要素の順序を入れ換えて · 演算を行なっている。

以上で定義したモデルと演算を用いて、モデル $\mathcal{M}$ と式 A との関係 $\mathcal{M} \models A$ を以下のように定義する。

定義 3.6 [式の解釈] 以下では $\alpha \in (\mathbb{P}_a \cup \{\mathbf{0}\})$ である。

$$\begin{aligned}
\mathcal{M} \models \alpha &\Leftrightarrow \exists s \in \mathcal{M}, \exists v \in eval(s), \\
&\quad (v(\alpha) = (0,1) \text{ and } \mathbb{P}(v) = \{\alpha\}) \\
\mathcal{M} \models A * B &\Leftrightarrow \mathcal{M}_1 \models A \text{ and } \mathcal{M}_2 \models B \\
&\quad \text{and } \mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2 \\
\mathcal{M} \models A \multimap B &\Leftrightarrow \forall \mathcal{M}'(\mathcal{M}' \models A \Rightarrow \mathcal{M} + \mathcal{M}' \models B) \\
\mathcal{M} \models A \sqcap B &\Leftrightarrow \mathcal{M} \models A \text{ and } \mathcal{M} \models B \\
\mathcal{M} \models A \sqcup B &\Leftrightarrow \mathcal{M} \models A \text{ or } \mathcal{M} \models B \\
\mathcal{M} \models !A &\Leftrightarrow \mathcal{M} \models A \text{ and } \mathcal{M} = \mathcal{M} + \mathcal{M}
\end{aligned}$$

定義 3.7 [定数の解釈]

$$\forall \mathcal{M}\ (\mathcal{M} \models \top) \quad , \quad \forall \mathcal{M}\ (\mathcal{M} \not\models \bot)$$

定義 3.8 [真、恒真] 線形論理の論理式 A について、

- "A がモデル $\mathcal{M}$ で真" **iff** $\mathcal{M} \models A$。
- "A が真" **iff** モデル $\{\emptyset\}$ で A が真。
- "A が恒真" **iff** 任意のモデルで A が真。

4 完全性

定理 4.1 "シーケント計算で $\vdash A$ が証明可能" と "式 A が真" は同値である。

証明の概略

($\Longrightarrow$) 証明図の構造に関する帰納法で証明する。

- Base case として、シーケント計算の axiom 規則 ($a \vdash a$) と定数に関する規則 ($\Gamma \vdash \top, \Delta$ など) が真であることを示す。
- Induction step として、シーケント計算の各規則が真であることを保存することを示す。つまり、規則の前提部が真であると仮定したときに結論部も真になることを示す。

($\Longleftarrow$) 式 A は真なのでモデル $\{\emptyset\}$ で真となる。この時、真偽の判定を行なうために原子命題または定数になるまで、A は分解される。この分解の過程と、原子命題が真となる時の *eval* での付値の並び方から、シーケント計算の証明図を得ることができる。

5 まとめ

本稿では命題線形論理 CLL_e に対し、モデルをもとに直感的にわかりやすい意味論を提案した。この意味論では命題の収支に着目することで式の真偽を判定している。この意味論により論理式の直感的な理解を助けることで、線形論理による仕様記述や検証の助けになると考える。

参考文献

[1] ABRAMSKY, S. and JAGADEESAN, R. Games and Full Completeness for Multiplicative Linear Logic, *The Journal of Symbolic Logic*, **59**, 2 (June 1994), 543–574.

[2] ALLWEIN, G. and DUNN, J. Kripke Models for Linear Logic, *The Journal of Symbolic Logic*, **58**, 2 (June 1993), 514–545.

[3] AVRON, A. The semantics and proof theory of linear logic, *Theoretical Computer Science*, **57** (1988), 161–184.

[4] BLASS, A. A game semantics for linear logic, *Annals of Pure and Applied Logic*, **56** (1992), 183–220.

[5] BLASS, A. Is Game Semantics Necessary?, *Lecture Notes in Computer Science*, **832** (1993), 66–77.

[6] GIRARD, J. Y. Linear logic, *Theoretical Computer Science*, **50**, 1 (1987), 1–102.

[7] LAFONT, Y. and STREICHER, T. Games Semantics for Linear Logic, *Proc. IEEE 6th annual Symposium on Logic in Computer Science* (1991), 43–50.

[8] TROELSTRA, A. S. *Lectures on Linear Logic*, No. 29 in CSLI Lecture Notes, Center for the Study of Language and Information, Stanford University (1992).