

5 CAL システムを使った演習

CAL システムは、2000 年頃から京都大学の佐藤雅彦教授が中心となり、亀山らが協力して構築してきた教育用ソフトウェアである。CAL の名称は Computation and Logic から来ており、計算や論理の体系に関する形式的な操作を行ったりチェックするためのシステムである。このシステムを使うことにより、自分の好きな時に、好きなペースで、演習を行うことができるようになる。

CAL システムは、過去数年間使ってきたものであり、ある程度安定して使えるようにはなっているが、商用のソフトウェアではないので、検証が十分でなく、不具合がまったくないとは限らない。また、ドキュメント類についても不備がある可能性がある。不具合などを発見したら連絡してほしい。

また、筑波大学で提供しているソフトウェアは 2002 年頃の安定版である。京都大学の授業で利用しているシステムは、実験的に次々と機能を追加し、また、体系自体も変化しているので、現在ではかなり異なるシステムとなっている。

5.1 CAL システムに必要な環境

現在、情報学類 計算機室の計算機で使用可能である。

これらの計算機では、普通に emacs とたたけばそのまま CAL システムが使用可能である。emacs にはいくつかの異なる版や実装があるが、emacs 20.x 以降以外での動作は保証しない。(たとえば、emacs 19.x では動かない。)

自宅から使う場合は、ネットワーク経由でリモートログインして使ってほしい。

5.2 準備

これまでに CAL システムを使ったことのない人にとって、準備は`~/emacs`に以下の 2 行のいずれかを加えるだけである。(2006 年度のシステムから、各自のホームディレクトリの所在場所によって、加えるべき行が異なるようになった。)

```
;; ホームディレクトリが /home1/.../ にある人
(setq cal-dir "~/kam/cal-system/")
(autoload 'cal "~/kam/cal-system/cal-mode/cal.elc" nil t)
```

または、

```
;; ホームディレクトリが /home2/.../ にある人向け
(setq cal-dir "~/kam2/cal-system/")
(autoload 'cal "~/kam/cal-system/cal-mode/cal.elc" nil t)
```

これらは、CAL システムが `~/kam/cal-system` などの下にあることを意味している。この 2 行を加えたら、一度 emacs を終了してから emacs を再起動してほしい。

以前に CAL システムを起動したことのある人は、自分の home directory の直下にある cal という名前の directory を削除する必要がある。C-SHELL の場合は、次のような操作となる。

```
/bin/rm ~/cal
```

これを削除しないと、システム側のログファイルと矛盾が生じるため意味不明のエラーメッセージ(「CAL の記録ファイルがありません。」)が出て CAL システムを起動できないことがある。上のように削除をしたら、一度 emacs を抜けて再起動してほしい。

5.3 CAL システムの起動と終了

emacs を起動後に,

```
Meta-x cal (通常の設定の場合は Esc x cal)
```

とすれば CAL システムが起動する. また, CAL システムが動いているときに,

```
CAL <123> bye
```

とすれば CAL システムが終了する. (ここで CAL <123> はシステムがだすプロンプトであり, ユーザが入力するのは bye だけである. return キーをたたく必要もない.)

なお, CAL がおかしくなってどうしても終了できない場合には, emacs を終了してしまえばよいが, これはできる限り避けてほしい. bye を入力することによって, そのときまでに解答した演習問題のデータが記録される. (したがって, 強制終了したときには, そのときに解答した記録が残らず, もう一度解答することになる.)

また, CAL システムを同時に複数立ち上げることもやらないでほしい. 問題文を見ながら解答をしたい, といった場合に, そうしたい気持ちはわかるが, CAL システムは一人の人が複数動かすことを想定していないので, せっかく解答したものが消えてしまうことがある.

5.4 有益な情報

システムを起動したら, なにはともあれ, news[1] とやって, ニュースの 1 番目を読んでほしい. また, help[] とやると, 多少のヘルプがでてくる.

また, 問題文を読むのは Q[n] であり (n は問題番号), 問題 n の解答をするのは, A[n][解答] である.

5.5 導出ゲーム

CAL は, 型付きラムダ計算などに特化したシステムではなく, いろいろな体系のチェックをするための一般的なシステムである. そこで, 個々の体系において導出をチェックをすることを「導出ゲーム」と呼んでいる. ゲームというのは, システムが出す課題を人間 (学生) が解くという対話的な点に着目してのことである.

本講義の演習においてプレイするゲームには, 命題論理の導出ゲームや, 型付きラムダ計算における項の導出ゲームや簡約のゲームなどがある.

CAL システムにおいては, 常に「現在有効なゲーム」が決まっていて, そのゲームの課題しか解くことができない. ゲームを切りかえるには play コマンドを使う.

```
CAL <123> play[Prop]
```

とすれば (現在のゲームを終了して) Prop ゲームが開始される. 具体的なゲームの名称と, そのゲームの規則は, それぞれのゲームの項で説明する.

5.6 導出

形式的体系では、 $\Gamma \vdash M : A$ であることを確認するため、それを導く際に使われたルールを木の形に積みあげた図形を使った。これが、導出である。CAL システムは各種の導出のチェックをするシステムであると言える。

ただし、導出を木の形に書くとチェックが大変であるし、何より人間が導出を書くのは通常下からであるので、そのまま表示するのは具合が悪い。そこで、CAL システムでは

$$\frac{\text{部分導出 1} \quad \text{部分導出 2} \quad \text{部分導出 3}}{\text{結論}} \text{ルール名}$$

という形の導出を以下の形で表現することにした。

```
結論 by ルール名 {  
  部分導出 1 ;  
  部分導出 2 ;  
  部分導出 3  
}
```

ここで、by や{ (開き中括弧), } (閉じ中括弧), ; (セミコロン) などの記号の使い方に注意してほしい。とくに、セミコロンは、最後の部分導出の後には書かない(C 言語とは違う)ことに注意してほしい。

5.7 判断 (judgment)

CAL システムのもう 1 つの特徴として、宣言を毎回書かなくてよいようにした点がある。たとえば、

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

という導出では、 Γ が 3 回も出てくるので、これが長い場合は書くのが大変である。

CAL では、これをすっかり省略するかわりに、導出を書く前に前置きとして、宣言や結論などを書くことにした。つまり、以下の形である。

宣言 $\vdash$ 結論 in ゲーム名 since 導出

ここで「宣言 $\vdash$ 結論」という形を判断という。

宣言は最初だけ書くのであって、導出の中では書かない。なお、CAL 内部では、宣言のことを仮定列と呼ぶこともある。

ゲーム名は、当然ながら、現在有効なゲーム名と一致しなければならない。

5.8 宣言 (仮定列)

Γ には、宣言だけでなく、その式で使われているすべての変数 (型をあらわす変数も含む) を書く。また、通常の変数の型宣言には、名前を付ける。

たとえば、 $A \rightarrow B, A \vdash B$ という判断は、CAL システムでは、

$$A, B, X1 :: A \rightarrow B, X2 :: A \vdash B$$

という判断で表現される．ここで， A, B は仮定ではなく，単に「この判断の中で A, B という基本命題を使っている」という情報に過ぎない． $X1 :: A \rightarrow B$ や $X2 :: A$ が仮定であり，これらはそれぞれ $X1$ と $X2$ という名前が付けられている．仮定に名前を付ける理由は，後でその仮定を使うときに，どの仮定であるかを指定するためである．仮定につける名前は，互いに異なるものを付けるべきである．また，仮定につける名前は基本命題とは異なるものとすべきである．

5.9 キー入力

論理記号や「 $\rightarrow$ 」「 $\times$ 」「 $+$ 」「 λ 」などの記号は，キーボードから 2 文字程度で簡単に打ちこめるようになっている．

当面使う主な記号入力方法を以下にあげる．たとえば，“/” (スラッシュ) と “l” (英小文字のエル) をつづけて打てば “ λ ” と入力される．

入力	表示	入力	表示
/f	$\perp$	/l	λ
/p	$\vdash$	/*	$\times$
/n	$\neg$	/+	$+$
/i	$\supset$	/-	$\rightarrow$
/a	$\wedge$	/l	$\downarrow$
/o	$\vee$		

これにより，日本語の文字を入力をする必要は全くない．

以上の一覧は `help[key]` としても表示される．

5.10 詳しい資料

京都大学の佐藤教授も「計算と論理」という名称で，CAL システムを使った講義を行っている．その講義資料は，以下の場所から入手可能である．

<http://www.sato.kuis.kyoto-u.ac.jp/>

ただし，この講義資料は，独特の論理体系を用いているため，筑波大学での講義には直接役立たない．CAL システムにある程度慣れた後に，その設計思想や狙い (哲学) を知る目的であれば，非常に有益な資料となるであろう．

6 PROP ゲーム

命題論理における導出は Prop という名前の導出ゲームである。ただし、このゲームは古典論理ではなく、直観主義論理に対応するものであるので、 $\neg\neg E$ 規則を使うことはできない。もっとも、演習問題を解くにあたって、この事を意識する必要はあまりない。

Prop 導出ゲームにおける解答の形式は以下の通りである。

```
A[問題番号] [
  仮定列 ⊢ 命題 in Prop since
  導出
]
```

仮定列 (あるいは宣言列) は、基本命題および「名前::命題」の形の仮定の列である。ただし、空列でもよい。

Prop ゲームのルールを CAL システムにおいて表現したものは以下の通りである。最初からこれらの形を暗記することは必要なく、演習問題を解いていくうちに理解を深まるようになっていく。

```
A by assume {
  A という仮定に付けた名前 (X1 など)
}
```

なお, assume は長いので as としてもよい。

```
A ∧ B by ∧ I {
  A の導出 ;
  B の導出
}
```

```
A by ∧ EL {
  A ∧ B の導出
}
```

```
B by ∧ ER {
  A ∧ B の導出
}
```

```
A ∨ B by ∨ IL {
  A の導出
}
```

```
A ∨ B by ∨ IR {
  B の導出
}
```

```
C by ∨ E {
  A ∨ B の導出 ;
  (名前 1 :: A)[ C の導出 ] ;
}
```

```

    (名前 2 :: B)[ C の導出 ]
  }

A ⊃ B by ⊃ I {
  (名前 :: A)[ B の導出 ]
}

B by ⊃ E {
  A ⊃ B の導出 ;
  A の導出
}

¬ A by ¬ I {
  (名前 :: A)[ ⊥ の導出 ]
}

B by ¬ E {
  ¬ A の導出 ;
  A の導出
}

A by ⊥ E {
  ⊥ の導出
}

```