

NumberPlace.lisp をコンパイルして高速に実行する

CLISP の場合、ソースコードをコンパイルしてから実行するとコンパイル前に比べて 4 倍～5 倍実行速度が速くなります。

NumberPlace.lisp が開発時にサンプルとして使用している 10 個の数独（ナンプレ）の問題を実行した例だと

- ・ インタープリタでの合計実行時間 = 約 451 秒
- ・ コンパイル後の合計実行時間 = 約 106 秒

と約 4.3 倍高速になっています。これは「理詰め」での解法を最優先で解く最も時間が掛る「machine-level」での実行時間ですが、基本手筋と試行錯誤法だけで解く速度最優先の「speed-first」では

- ・ インタープリタでの合計実行時間 = 約 22.49 秒
- ・ コンパイル後の合計実行時間 = 約 4.57 秒

と、それぞれ同じモードで約 20 倍高速です。インタープリタ上で machine-level で実行した場合と、コンパイル・コードを speed-first で実行した場合ではほぼ 100 倍の速度差があります。

NumberPlace.lisp をコンパイルしてから実行することは難しくないのですが、コンパイルしておいてから実行することをお勧めします。コンパイルは一回行っておけば、コンパイル結果として生成される「NumberPlace.fas」ファイルを削除しない限り 2 度目以降にコンパイル作業を再度行う必要はありません。

CLISP のアイコンをダブル・クリックしてインストール済みの CLISP を実行します。起動後にファイルをコンパイルする関数「compile-file」の引数として、NumberPlace.lisp をパス名を含めて指定します。NumberPlace.lisp がカレント・ディレクトリに置かれているのであればパス名を省略しても大丈夫です(青字が入力部分)。

```
[1]> (compile-file "numberplace.lisp")
;; Compiling file C:\Users\daigo\Documents\lisp\NumberPlace\NumberPlace.lisp ...
;; Wrote file C:\Users\daigo\Documents\lisp\NumberPlace\NumberPlace.fas
0 errors, 0 warnings
#P"C:\Users\daigo\Documents\lisp\NumberPlace\NumberPlace.fas" ;
NIL ;
NIL
```

エラーと警告メッセージが共に「0」であることを確認し、作成されたコンパイル済みの「fas」ファイルをロードします。CLISP は開発速度を重視した処理系なので、このファイ

ルは機械語ではなく、コンパイル時間が短くて済む中間コード形式です。ロードするには以下のようにします。

```
[2]> (load "numberplace.fas")  
;; Loading file numberplace.fas ...  
;; Loaded file numberplace.fas  
T
```

以降はソースコードを直接ロードしてインタプリタで実行する場合とまったく同じです。「machine-level」で「sample-board-6」の解法過程を表示させるのであれば

```
[3]> (machine-level)
```

```
[4]> (teach sample-board-6)
```

とすれば OK です。 ■