

## BEM-II: Arithmetic Boolean Expression Manipulator (Sapporo 版)の使用法

### 1. 対象機種と起動法

本プログラムは Linux OS で動作する。(gcc, g++, bison, flex を使用)

64 ビット計算機を想定しているが、コンパイルオプション設定により

32 ビット機にも対応可能である。

プログラムの起動方法は、

bemII [-最大 BDD 節点数] [ファイル名]

である。

起動時にファイル名を指定するとファイルに書かれた命令を実行する。ファイル名を省くと標準入力から読み込む。計算結果は標準出力に出力される。端末入出力によるインタプリタ形式の利用法と、ファイル入出力によるフィルタ形式の利用法の両方が可能である。

本プログラムでは、内部で生成する BDD データ (約 30 byte/node) が主記憶をあふれると処理効率が急激に低下するため、マシンの主記憶サイズに応じて最大 BDD 節点数を指定し、その範囲内で記憶管理を行う。(省略値は 400000。) 記憶領域は、起動時に最小限確保され、必要に応じて徐々に限界まで拡張される。計算中に指定した最大ノード数に達した場合は計算を打ちきり、警告メッセージを出力する (計算結果は 0 となる)。

(例)

bemII                    インタプリタモードで起動

bemII script          ファイル名 script に書かれた命令を実行

bemII -100000        最大 BDD 節点数 100000 で起動

### 2. 変数名・数値

本プログラムでは、論理関数の入力変数を表す「論理変数」と、計算結果を一時的に保持する記憶場所を表す「プログラム変数」を使用する。論理変数は英小文字で始まる英数字列、プログラム変数は英大文字で始まる英数字列で表す。英数字列は最大 255 文字までで、2 文字目以降にアンダーバーを含んでもよい。論理変数は最大 65510 個まで使用できる。プログラム変数の個数に制約はない。

数値は、整数だけを扱う。通常の 10 進数の他に、0x で始まる 16 進数、0b で始まる 2 進数を入力することができる。数値の範囲に制限はないが、100 bit を越える大きな数を扱うと計算時間が顕著に増大する。

コマンドの中でファイル名を指定する場合、および入力した文字列をエコーバックさせるときには、両端を引用符「"」で囲む。

(例)

論理変数	a	b	x1	x2	bdd	a5A	f_t6
プログラム変数	A	B	X1	X2	BDD	A5a	F_t6
数値 (10 進数)	0	29	32767	-213			
数値 (16 進数)	0x1A	0xFFFF	0x0	-0x412			
数値 (2 進数)	0b10110	0b110	0b0	-0b11101			
ファイル名	"a"	"b"	"script1.bem"				

### 3. 命令の構成

本プログラムは、基本的に行単位 (1 行 1 命令) で動作する。1 行に複数個の命令を書く場合は、セミコロン ; で区切る。複数行にわたって 1 命令を書く場合は、改行の直前にバックスラッシュ「¥」を置く。文中にシャープ「#」を書くと、次の改行までコメントとして読み飛ばされる。

プログラムの制御に関する命令としては、次の 3 つがある。

source	ファイル名	ファイルに書かれた命令を呼び出して実行する。
help	または ?	使用法を表示する。
quit	または exit	プログラムを終了する。ファイルの終り (EOF) でも終了。

BEM-II の計算を実行する命令は、次の 3 種からなる。

- ・ 宣言文 --- 使用する論理変数の名前とそのコスト、および変数順序を宣言する。
- ・ 代入文 --- 計算結果をプログラム変数に代入する。
- ・ 出力文 --- 計算結果を種々の形式で表示する。

### 4. 宣言文

宣言文は、使用する論理変数の名前と順序をあらかじめ宣言するものである。

symbol [ 論理変数名 [ , 論理変数名, ... ] ]

区切りのコンマは空白でもよい。変数の順序は、左から順に上位（先に展開される）に配置される。論理変数名を1つも書かなかった場合は、現在使用中の論理変数が一覧表示される。

また、論理変数にコストを指定することができる。

symbol [ 論理変数名(コスト) [ , 論理変数名(コスト), ... ] ]

上記のコストとは、print 文の /mincover, /mincost スイッチのときに使用するもので、入力変数を1にするためのコストを表す。本プログラムでは、論理関数の充足解を探索する際に、各入力変数のコストの総和が最小となるような解を求めることができる。なお実装の都合上、コストは32ビット整数として計算され、その範囲を超えると桁あふれ発生する。コストの指定を省略した場合は、1が設定される。

宣言文は複数回に分けて実行してもよいが、その場合、あとに宣言された変数が下位に配置される。同じ変数を2度宣言した場合は、変数順序は変わらないが、コストの指定は後の宣言の方が有効となる。

（例）

```
symbol a, b, c
```

```
symbol b, d, e
```

とすると、a b c d eの順になる。

宣言していない論理変数が算術論理式の中で使われた場合は、その場で新たに宣言したものとして、最下位に追加される（警告メッセージが出る）。

（例）

```
symbol a, b, c
```

```
print a & b & c & d
```

とすると、a b c dの順になる。

## 5. 代入文

代入文は、右辺の算術論理式を計算し、得られた整数値論理関数を左辺のプログラム変数に代入するものである。

プログラム変数名 = 算術論理式

プログラム変数名は、あらかじめ宣言する必要はない。代入文の左辺に初めて現れた時点で、記憶領域が確保される。プログラム変数の使用個数に制限はない。同じプログラム変数に重ねて代入すると、以前の内容が消去された後に、新しい値が代入される。論理変数を左辺に置くことはできない。また、代入されたことのないプログラム変数を右辺で参照することはできない。

右辺の算術論理式の文法は、C 言語に準拠している。使用できる演算子を、実行優先順に挙げる。同じ行の演算子は左優先である（？：だけは右優先）。

```
( )  
[ ]  
! ~ + - (単項)  
* / %  
+ - (二項)  
<< >>  
< <= > >=  
== !=  
&  
^  
|  
? :  
UpperBound( ) LowerBound( )
```

[ ]は、F[G] としたとき、G = 0 となる入力組合せを don't care とみなしたときに、F を単純化した関数を返す演算子である。例えば、F[x]とすると、x に 1 を代入したときの F を返す。

「!」は論理否定演算子である。!F としたとき、F=0 となる入力組合せに対して 1 を返し、それ以外は 0 を返す。

「~」はビット反転演算子である。「!」とは異なり、~0 は-1 となる。

? : は選択演算子である。F? G: H としたとき、F = 1 となる入力組合せに対しては G の値を返し、F = 0 となる入力に対しては H を返す。

「/」は整数除算の商（少数点以下切捨て）、「%」は剰余を返す。ある入力組合せに対して、除数が0になるような場合が存在するときには、警告メッセージを出力し、その組合せに対しては被除数をそのまま返す。商が負になる場合には少数点以下は、絶対値が小さい方に揃える。

UpperBound(F) は、入力変数があらゆる組合せをとるときの、F の最大値を返す。  
LowerBound( ) は最小値を返す。

その他の演算子は、通常の C 言語と同様である。

## 6. 出力文

出力文の形式は次の通りである。

```
print [ /スイッチ ] 算術論理式  
print “文字列”
```

print は「?」で代用できる。スイッチは出力形式を指定するもので、省略した場合は、関数に応じて適当に見やすい形式が選択される。  
引用符「”」で囲んだ文字列はそのままエコーバックされる。

以下に、現在使用できるスイッチとその出力形式を説明する。

/bit	内部の2進ベクトルの各ビットの論理関数を非冗長積和形で表示。
/hex	結果が定数の場合16進数で表示。定数でなければ/bitと同じ。
/map	カルノー図で表示。入力変数6個まで表示できる。
/rmap	カルノー図で表示。冗長な入力変数は省いて表示。
/case	場合分け表現で表示。
/asop	算術積和形で表示。
/size	計算結果のBDD節点数（および処理系全体の節点数）を表示。
/count	充足解の個数を表示。
/density	真理値表濃度（充足解の存在確率）を表示。
/mincover	コスト最小の充足解を表示。
/mincost	コスト最小の充足解のコスト値を表示。
/plot	BDDの形を図示する。
/plot0	BDDの形を図示する（否定枝不使用）

## 7. 実行例

### 【インタプリタモード】

```
% bemII
***** BEM-II Sapporo edition *****
> symbol a(3) b(2) c(4) d(1)
> Sum = a*3 + b*4 + c*5 - d*2
> print Sum
+-: !a & !b & !c & d
  3: a & b & c | a & c & !d | !a & !b & !c & d | b & c & !d
  2: b ^ ( a & c & d | !a & c & !d | !a & !c & d )
  1: d ^ ( a & !c )
  0: a ^ c
> print /map Sum
a b : c d
    |  00  01  11  10
00 |   0  -2   3   5
01 |   4   2   7   9
11 |   7   5  10  12
10 |   3   1   6   8
> print /case Sum
12: a & b & c & !d
10: a & b & c & d
  9: !a & b & c & !d
  8: a & !b & c & !d
  7: a & b & !c & !d | !a & b & c & d
  6: a & !b & c & d
  5: a & b & !c & d | !a & !b & c & !d
  4: !a & b & !c & !d
  2: !a & b & !c & d
  1: a & !b & !c & d
  0: !a & !b & !c & !d
 -2: !a & !b & !c & d
> print /asop Sum
3 * a + 4 * b + 5 * c - 2 * d
```

```

> print UpperBound(Sum)
    12
> print Sum > 6
    a & b & !d | a & c & !d | b & c
> F = Sum > 6
> print /mincover F
<Positive>: a b
> print /mincost F
    5
> exit
%
```

### 【8-Queen 問題】

(入力記述)

```

##### 8-Queen Problem #####
symbol a00 a10 a20 a30 a40 a50 a60 a70
symbol a01 a11 a21 a31 a41 a51 a61 a71
symbol a02 a12 a22 a32 a42 a52 a62 a72
symbol a03 a13 a23 a33 a43 a53 a63 a73
symbol a04 a14 a24 a34 a44 a54 a64 a74
symbol a05 a15 a25 a35 a45 a55 a65 a75
symbol a06 a16 a26 a36 a46 a56 a66 a76
symbol a07 a17 a27 a37 a47 a57 a67 a77
```

$$X0 = (a00 + a10 + a20 + a30 + a40 + a50 + a60 + a70 == 1)$$

$$X1 = (a01 + a11 + a21 + a31 + a41 + a51 + a61 + a71 == 1)$$

$$X2 = (a02 + a12 + a22 + a32 + a42 + a52 + a62 + a72 == 1)$$

$$X3 = (a03 + a13 + a23 + a33 + a43 + a53 + a63 + a73 == 1)$$

$$X4 = (a04 + a14 + a24 + a34 + a44 + a54 + a64 + a74 == 1)$$

$$X5 = (a05 + a15 + a25 + a35 + a45 + a55 + a65 + a75 == 1)$$

$$X6 = (a06 + a16 + a26 + a36 + a46 + a56 + a66 + a76 == 1)$$

$$X7 = (a07 + a17 + a27 + a37 + a47 + a57 + a67 + a77 == 1)$$

$$Y0 = (a00 + a01 + a02 + a03 + a04 + a05 + a06 + a07 == 1)$$

$$Y1 = (a10 + a11 + a12 + a13 + a14 + a15 + a16 + a17 == 1)$$

$$Y2 = (a20 + a21 + a22 + a23 + a24 + a25 + a26 + a27 == 1)$$

$$Y3 = (a30 + a31 + a32 + a33 + a34 + a35 + a36 + a37 == 1)$$

$$Y4 = (a40 + a41 + a42 + a43 + a44 + a45 + a46 + a47 == 1)$$

$$Y5 = (a50 + a51 + a52 + a53 + a54 + a55 + a56 + a57 == 1)$$

$$Y6 = (a60 + a61 + a62 + a63 + a64 + a65 + a66 + a67 == 1)$$

$$Y7 = (a70 + a71 + a72 + a73 + a74 + a75 + a76 + a77 == 1)$$

$$Z1 = (a10 + a01 < 2)$$

$$Z2 = (a20 + a11 + a02 < 2)$$

$$Z3 = (a30 + a21 + a12 + a03 < 2)$$

$$Z4 = (a40 + a31 + a22 + a13 + a04 < 2)$$

$$Z5 = (a50 + a41 + a32 + a23 + a14 + a05 < 2)$$

$$Z6 = (a60 + a51 + a42 + a33 + a24 + a15 + a06 < 2)$$

$$Z7 = (a70 + a61 + a52 + a43 + a34 + a25 + a16 + a07 < 2)$$

$$Z8 = (a71 + a62 + a53 + a44 + a35 + a26 + a17 < 2)$$

$$Z9 = (a72 + a63 + a54 + a45 + a36 + a27 < 2)$$

$$Za = (a73 + a64 + a55 + a46 + a37 < 2)$$

$$Zb = (a74 + a65 + a56 + a47 < 2)$$

$$Zc = (a75 + a66 + a57 < 2)$$

$$Zd = (a76 + a67 < 2)$$

$$W1 = (a06 + a17 < 2)$$

$$W2 = (a05 + a16 + a27 < 2)$$

$$W3 = (a04 + a15 + a26 + a37 < 2)$$

$$W4 = (a03 + a14 + a25 + a36 + a47 < 2)$$

$$W5 = (a02 + a13 + a24 + a35 + a46 + a57 < 2)$$

$$W6 = (a01 + a12 + a23 + a34 + a45 + a56 + a67 < 2)$$

$$W7 = (a00 + a11 + a22 + a33 + a44 + a55 + a66 + a77 < 2)$$

$$W8 = (a10 + a21 + a32 + a43 + a54 + a65 + a76 < 2)$$

$$W9 = (a20 + a31 + a42 + a53 + a64 + a75 < 2)$$

$$Wa = (a30 + a41 + a52 + a63 + a74 < 2)$$

$$Wb = (a40 + a51 + a62 + a73 < 2)$$

$$Wc = (a50 + a61 + a72 < 2)$$

$$Wd = (a60 + a71 < 2)$$

$$C = 1$$



```

C = C & X0 & X1 & X2 & X3 & X4 & X5 & X6 & X7
C = C & Y0 & Y1 & Y2 & Y3 & Y4 & Y5 & Y6 & Y7
C = C & Z1 & Z2 & Z3 & Z4 & Z5 & Z6 & Z7 & Z8 & Z9 & Za & Zb & Zc & Zd
C = C & W1 & W2 & W3 & W4 & W5 & W6 & W7 & W8 & W9 & Wa & Wb & Wc & Wd

print /size C
print /count C
print /mincover C

```

(実行結果)

```

% bemII queen8.bem
    2450 (3014)
    92
<Positive>: a77 a36 a05 a24 a53 a12 a61 a40
%

```

### 【SUBSET SUM 問題】

(入力記述)

```

##### SUBSET-SUM Problem #####
symbol a b c d e f

```

```

Sum = 2*a + 3*b + 4*c + 3*d + 5*e + 6*f
print /map Sum

```

```

S = Sum * (Sum <10)
print /map S

```

```

C = UpperBound(S)
print C
print (C == S)

```

(実行結果)

```

% bemII subsetsum.bem
a b c : d e f
      | 000 001 011 010 | 110 111 101 100

```

000		0	6	11	5		8	14	9	3
001		4	10	15	9		12	18	13	7
011		7	13	18	12		15	21	16	10
010		3	9	14	8		11	17	12	6
110		5	11	16	10		13	19	14	8
111		9	15	20	14		17	23	18	12
101		6	12	17	11		14	20	15	9
100		2	8	13	7		10	16	11	5

a b c : d e f

		000	001	011	010		110	111	101	100
000		0	6	0	5		8	0	9	3
001		4	0	0	9		0	0	0	7
011		7	0	0	0		0	0	0	0
010		3	9	0	8		0	0	0	6
110		5	0	0	0		0	0	0	8
111		9	0	0	0		0	0	0	0
101		6	0	0	0		0	0	0	9
100		2	8	0	7		0	0	0	5

9

a & b & c & !d & !e & !f | a & !b & c & d & !e & !f | !a & b & !c & !d & !e & f | !a & !b & c & !d & e & !f | !a & !b & !c & d & !e & f

# 【最小木問題】

(入力記述)

```
# Minimum-Tree Problem
# Node: A1, A2, A3, A4, A5, A6, A7
# Edge: a: (A1, A2)
#       b: (A1, A3)
#       c: (A2, A3)
#       d: (A2, A4)
#       e: (A3, A4)
#       f: (A2, A5)
#       g: (A3, A6)
```

# h: (A4, A5)  
# i: (A4, A6)  
# j: (A5, A6)  
# k: (A5, A7)

symbol a(3) b(4) c(5) d(2) e(3) f(3) g(2) h(1) i(2) j(3) k(4)

A1 = 1  
A2 = 0  
A3 = 0  
A4 = 0  
A5 = 0  
A6 = 0  
A7 = 0

A1 = A1 | (a & A2) | (b & A3)  
A2 = A2 | (a & A1) | (c & A3) | (d & A4) | (f & A5)  
A3 = A3 | (b & A1) | (c & A2) | (e & A4) | (g & A6)  
A4 = A4 | (d & A2) | (e & A3) | (h & A5) | (i & A6)  
A5 = A5 | (f & A2) | (h & A4) | (j & A6) | (k & A7)  
A6 = A6 | (g & A3) | (i & A4) | (j & A5)  
A7 = A7 | (k & A5)

A1 = A1 | (a & A2) | (b & A3)  
A2 = A2 | (a & A1) | (c & A3) | (d & A4) | (f & A5)  
A3 = A3 | (b & A1) | (c & A2) | (e & A4) | (g & A6)  
A4 = A4 | (d & A2) | (e & A3) | (h & A5) | (i & A6)  
A5 = A5 | (f & A2) | (h & A4) | (j & A6) | (k & A7)  
A6 = A6 | (g & A3) | (i & A4) | (j & A5)  
A7 = A7 | (k & A5)

A1 = A1 | (a & A2) | (b & A3)  
A2 = A2 | (a & A1) | (c & A3) | (d & A4) | (f & A5)  
A3 = A3 | (b & A1) | (c & A2) | (e & A4) | (g & A6)  
A4 = A4 | (d & A2) | (e & A3) | (h & A5) | (i & A6)  
A5 = A5 | (f & A2) | (h & A4) | (j & A6) | (k & A7)

A6 = A6 | (g & A3) | (i & A4) | (j & A5)  
A7 = A7 | (k & A5)

A1 = A1 | (a & A2) | (b & A3)  
A2 = A2 | (a & A1) | (c & A3) | (d & A4) | (f & A5)  
A3 = A3 | (b & A1) | (c & A2) | (e & A4) | (g & A6)  
A4 = A4 | (d & A2) | (e & A3) | (h & A5) | (i & A6)  
A5 = A5 | (f & A2) | (h & A4) | (j & A6) | (k & A7)  
A6 = A6 | (g & A3) | (i & A4) | (j & A5)  
A7 = A7 | (k & A5)

A1 = A1 | (a & A2) | (b & A3)  
A2 = A2 | (a & A1) | (c & A3) | (d & A4) | (f & A5)  
A3 = A3 | (b & A1) | (c & A2) | (e & A4) | (g & A6)  
A4 = A4 | (d & A2) | (e & A3) | (h & A5) | (i & A6)  
A5 = A5 | (f & A2) | (h & A4) | (j & A6) | (k & A7)  
A6 = A6 | (g & A3) | (i & A4) | (j & A5)  
A7 = A7 | (k & A5)

A1 = A1 | (a & A2) | (b & A3)  
A2 = A2 | (a & A1) | (c & A3) | (d & A4) | (f & A5)  
A3 = A3 | (b & A1) | (c & A2) | (e & A4) | (g & A6)  
A4 = A4 | (d & A2) | (e & A3) | (h & A5) | (i & A6)  
A5 = A5 | (f & A2) | (h & A4) | (j & A6) | (k & A7)  
A6 = A6 | (g & A3) | (i & A4) | (j & A5)  
A7 = A7 | (k & A5)

A1 = A1 | (a & A2) | (b & A3)  
A2 = A2 | (a & A1) | (c & A3) | (d & A4) | (f & A5)  
A3 = A3 | (b & A1) | (c & A2) | (e & A4) | (g & A6)  
A4 = A4 | (d & A2) | (e & A3) | (h & A5) | (i & A6)  
A5 = A5 | (f & A2) | (h & A4) | (j & A6) | (k & A7)  
A6 = A6 | (g & A3) | (i & A4) | (j & A5)  
A7 = A7 | (k & A5)

C = A1 & A2 & A3 & A4 & A5 & A6 & A7



```

b & c & e & i & j & k | b & c & f & g & h & k | b & c & f & g & i & k |
b & c & f & h & i & k | b & c & f & h & j & k | b & c & f & i & j & k |
b & c & g & h & i & k | b & c & g & h & j & k | b & c & g & i & j & k |
b & d & e & f & g & k | b & d & e & f & i & k | b & d & e & f & j & k |
b & d & e & g & h & k | b & d & e & g & j & k | b & d & e & h & i & k |
b & d & e & h & j & k | b & d & e & i & j & k | b & d & f & g & i & k |
b & d & f & g & j & k | b & d & g & h & i & k | b & d & g & h & j & k |
b & d & g & i & j & k | b & e & f & g & h & k | b & e & f & g & j & k |
b & e & f & h & i & k | b & e & f & h & j & k | b & e & f & i & j & k |
b & f & g & h & i & k | b & f & g & h & j & k | b & f & g & i & j & k

```

27 (159)

508

<Positive>: a d g h i k

%

## 8. 本プログラムの命令文法

本プログラムが受理する命令の文法を、yacc に準ずる記法で示す。

bem

```

: lines
: lines exit
;

```

exit

```

: "exit¥n"
: "quit¥n"
;

```

lines

```

: lines commands "¥n"
| lines "source" STRINGS
;

```

commands

```

: commands ";" command
| command
;

```

command

```

        : /* empty */
        | help
        | define
        | assign
        | output
        ;

help
    : "help"
    | "?"
    ;

define
    : "symbol" symbols
    | "symbol"
    ;

symbols
    : SYMNAME comma symbols
    | SYMNAME
    | SYMNAME "(" NUMBER ")" comma symbols
    | SYMNAME "(" NUMBER ")"
    ;

comma
    : ",",
    | /* empty */
    ;

assign
    : VARNAME "=" expression
    ;

expression
    : "(" expression ")"
    | SYMNAME
    | VARNAME
    | NUMBER
    | expression "[" expression "]"
    | expression "?" expression ":" expression
    | expression "+" expression
    | expression "-" expression

```

```

| expression "*" expression
| expression "/" expression
| expression "%" expression
| expression "&" expression
| expression "|" expression
| expression "^" expression
| expression "==" expression
| expression "!=" expression
| expression ">" expression
| expression ">=" expression
| expression "<" expression
| expression "<=" expression
| expression "<<" expression
| expression ">>" expression
| "~" expression
| "!" expression
| "-" expression
| "+" expression
| "UpperBound" "(" expression ")"
| "LowerBound" "(" expression ")"
;

```

output

```

: print expression
| print switch expression
| print STRINGS
;

```

print

```

: "print"
| "?"
;

```

switch

```

| "/hex"
| "/bit"
| "/map"
| "/case"
| "/asop"

```



```

| "/rmap"
| "/size"
| "/density"
| "/count"
| "/mincover"
| "/mincost"
| "/plot"
;

SYMNAME
: [a-z][a-zA-Z0-9_]*
;

VARNAME
: [A-Z][a-zA-Z0-9_]*
;

NUMBER
: [0-9]+
| 0[xX][0-9a-fA-F]+
| 0[bB][0-1]+
;

STRINGS
: ¥" [^¥" ^¥n] +¥"
;

```

以上。