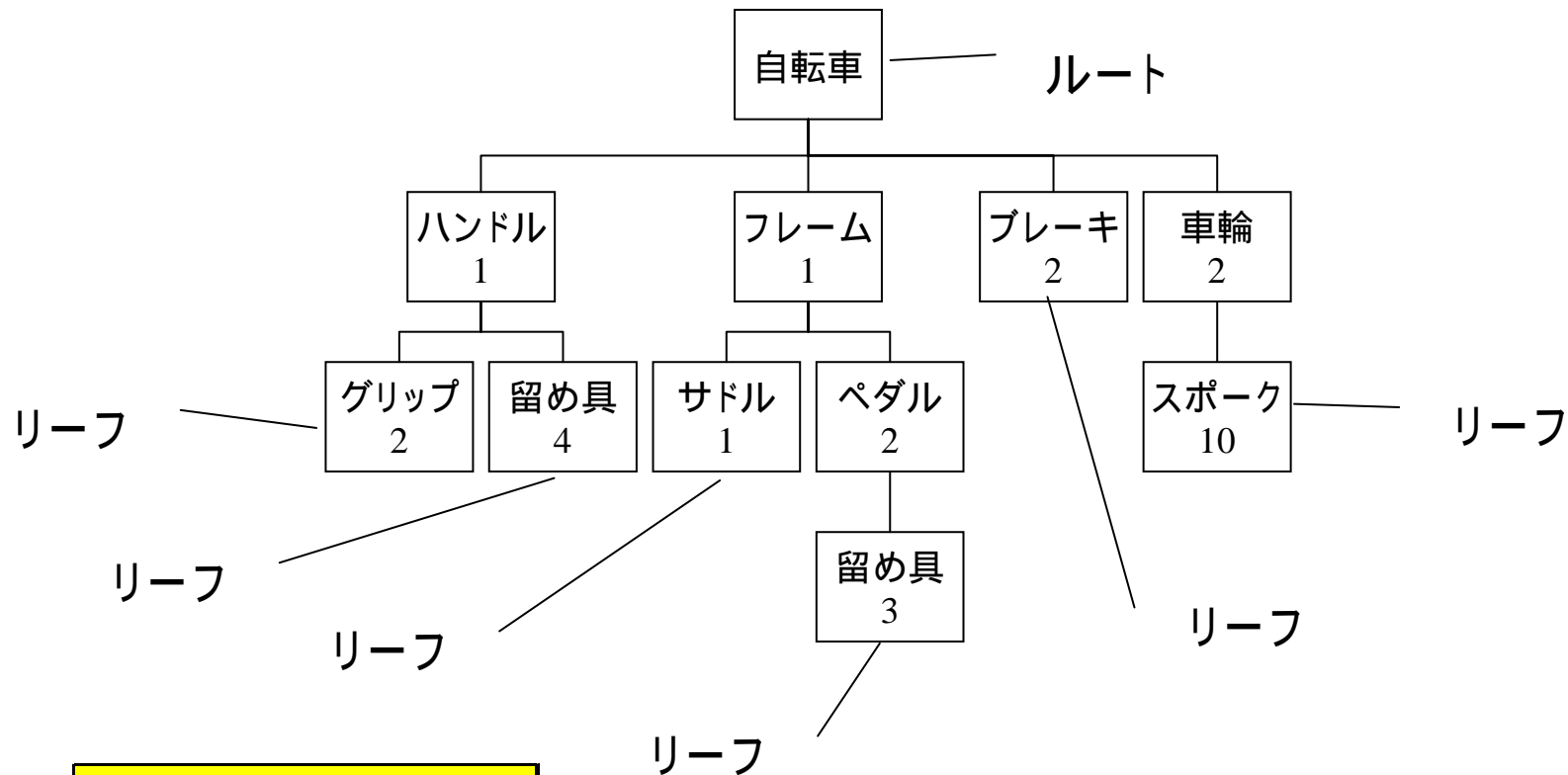


# 木構造を処理するマクロの作成 - SQL プロシジャを利用して -

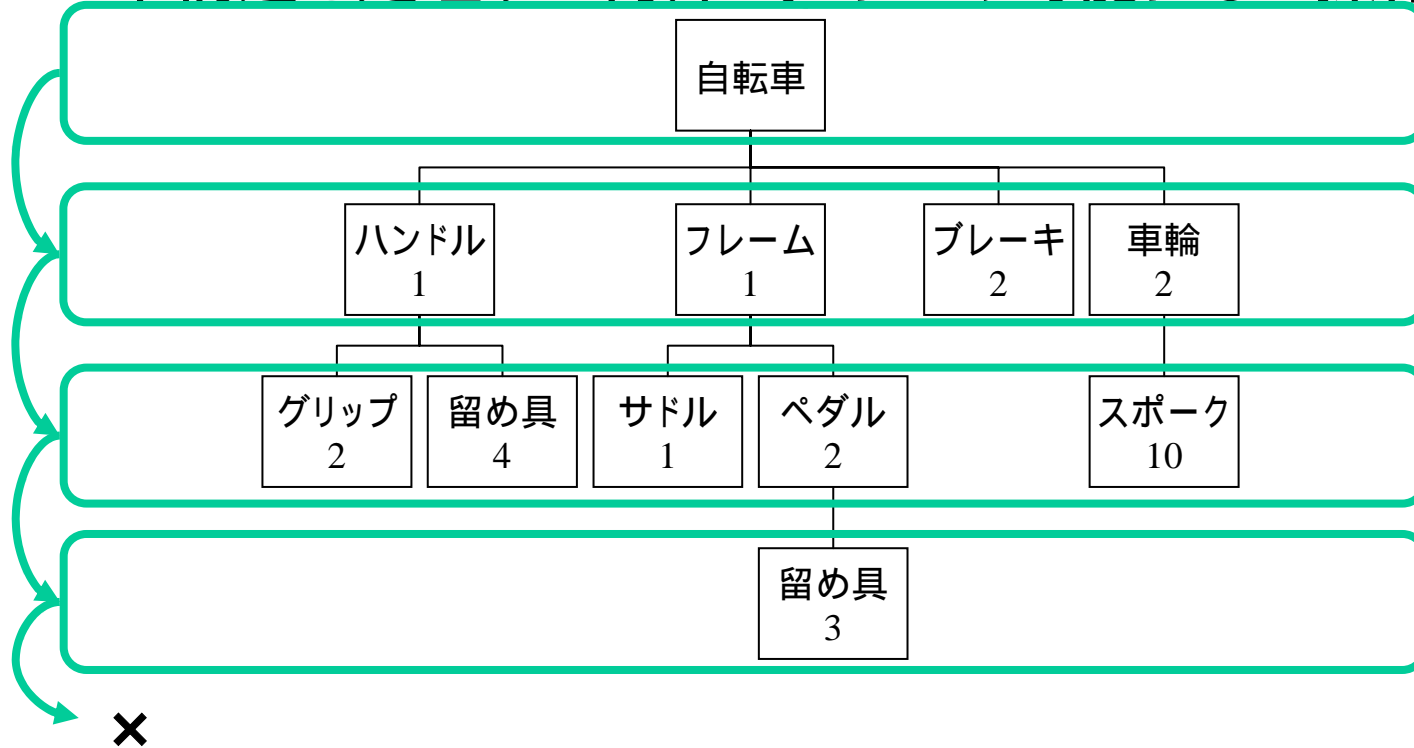
株式会社シーエーシー  
医薬・食品システムビジネスユニット  
堀田真一

# 木構造を処理する再帰SQLにより可能になる操作



parts	subparts	n
自転車	ハンドル	1
自転車	フレーム	1
自転車	車輪	2
自転車	ブレーキ	2
ハンドル	グリップ	2
ハンドル	留め具	4
フレーム	サドル	1
フレーム	ペダル	2
車輪	スポーク	10
ペダル	留め具	3

# 木構造を処理する再帰SQLにより可能になる操作



parts	subparts	n
自転車	ハンドル	1
自転車	フレーム	1
自転車	車輪	2
自転車	ブレーキ	2
ハンドル	グリップ	2
ハンドル	留め具	4
フレーム	サドル	1
フレーム	ペダル	2
車輪	スポーク	10
ペダル	留め具	3

再帰SQL

parts	subparts	sum
自転車	ハンドル	1
自転車	フレーム	1
自転車	車輪	2
自転車	ブレーキ	2
ハンドル	グリップ	2
ハンドル	留め具	4
フレーム	サドル	1
フレーム	ペダル	2
車輪	スポーク	20
ペダル	留め具	6

# SQLプロシジャとマクロの組み合わせで再帰SQLのような機能を実現

```
%macro マクロ名( ~ , ... , ~ ) ;  
proc sql ;
```

```
create table 出力データセット名  
( ~ , ~ , ... , ~ ) ;
```

```
create table バッファデータセット名 as  
select ~ , ~ , ... , ~  
from 入力データセット名  
where ルートの条件 ;
```

```
%do %while(&SQLOBS>0) ;
```

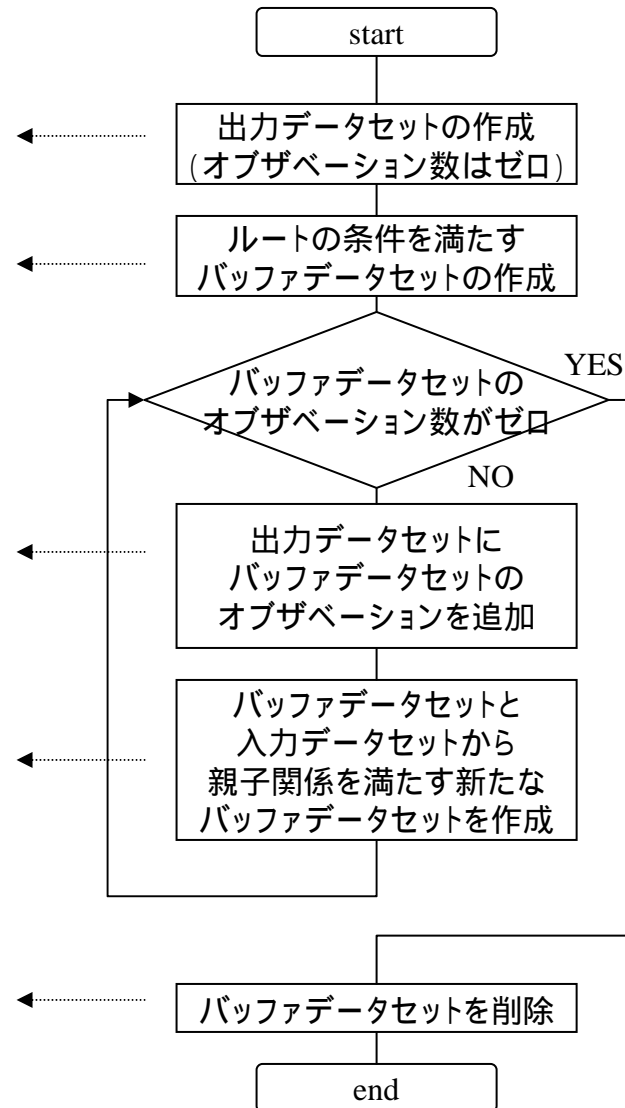
```
create table 出力データセット名 as  
select * from 出力データセット名  
union all  
select * from バッファデータセット名 ;
```

```
create table バッファデータセット名 as  
select ~ , ~ , ... , ~  
from 入力データセット名 ,  
      バッファデータセット名  
where 親子関係の記述 ;
```

```
%end ;
```

```
drop table バッファデータセット名 ;
```

```
quit ;  
%mend ;
```



# 木構造の利用 どれも空で無い部分集合への分割

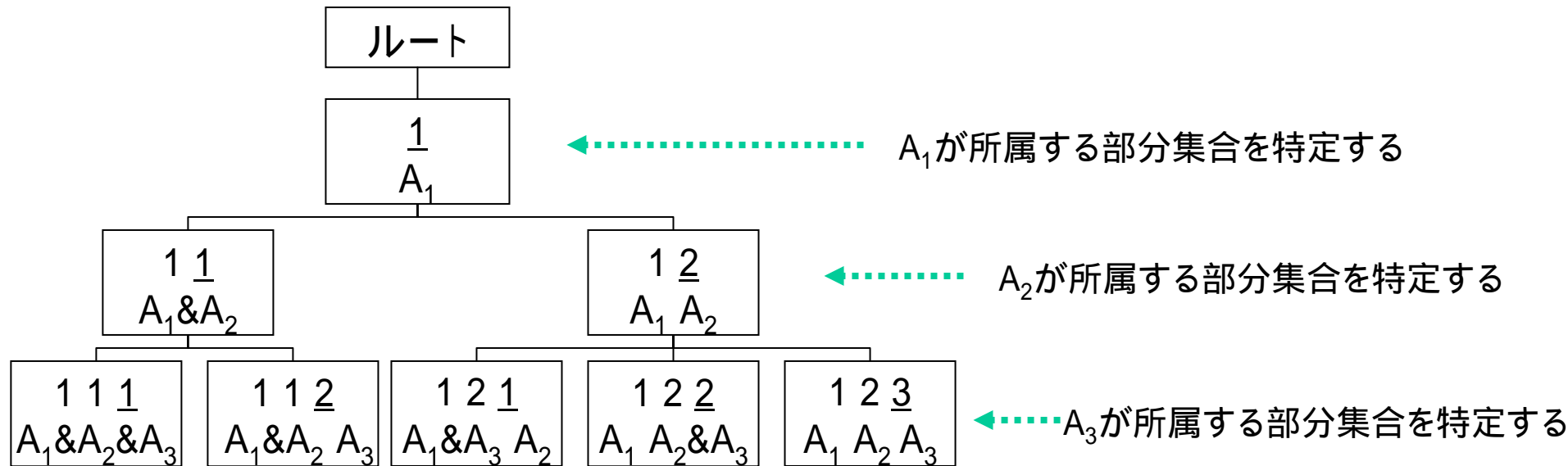
集合 $\{A_1, A_2, \dots, A_n\}$ をどれも空でない $r$ 個の部分集合に分割する。

分割された部分集合に1から始まる番号を割り当て、要素 $A_i$ が部分集合 $j$ に含まれるとき $C_i=j$ とする。分割の方法を $C_1C_2\cdots C_n$ で表す。

$n=3$ のとき

分割	$C_1C_2C_3$
$\{A_1, A_2, A_3\}$	1 1 1
$\{A_1, A_2\}\{A_3\}$	1 1 2
$\{A_1, A_3\}\{A_2\}$	1 2 1
$\{A_1\}\{A_2, A_3\}$	1 2 2
$\{A_1\}\{A_2\}\{A_3\}$	1 2 3

# 木構造の利用 どれも空で無い部分集合への分割

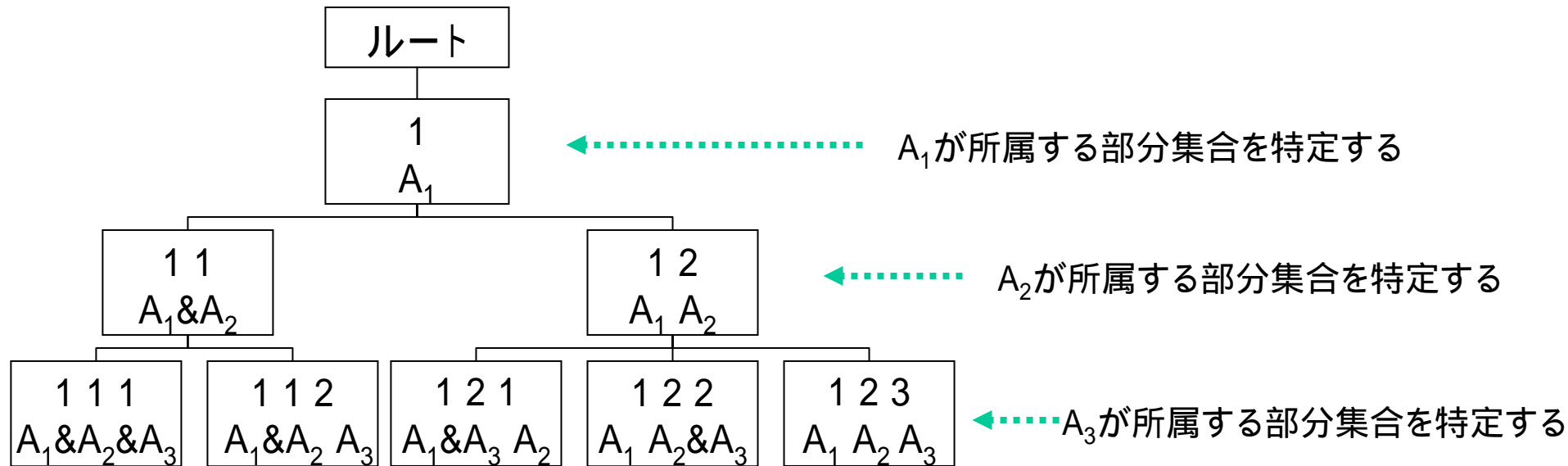


集合 $\{A_1, A_2, \dots, A_n\}$ の $k$ 個の部分集合からなる分割 $C_1 C_2 \dots C_k$ から、集合 $\{A_1, A_2, \dots, A_n, A_{n+1}\}$ の分割が $k+1$ 個できる。

$$\begin{aligned}
 &\{A_1, \dots\} \{A_j, \dots\} \dots \{\dots, A_n\} \longrightarrow \begin{aligned} &\{A_1, \dots, A_{n+1}\} \{A_j, \dots\} \dots \{\dots, A_n\} \\ &\{A_1, \dots\} \{A_j, \dots, A_{n+1}\} \dots \{\dots, A_n\} \\ &\vdots \\ &\{A_1, \dots\} \{A_j, \dots\} \dots \{\dots, A_n, A_{n+1}\} \\ &\{A_1, \dots\} \{A_j, \dots\} \dots \{\dots, A_n\} \{A_{n+1}\} \end{aligned} \\
 &\quad + \\
 &\quad A_{n+1}
 \end{aligned}$$

このような操作を $\{A_1\}$ の分割1から始めて、要素 $A_n$ まで繰り返すと集合 $\{A_1, A_2, \dots, A_n\}$ の全ての部分集合への分割が求められる。

# 木構造の利用 どれも空で無い部分集合への分割



indata1

pr	ch
.	1
1	1
1	2
2	1
2	2
2	3

indata2

id	name
1	A
2	B
3	C

マクロ処理

model	dest
A B C	1 1 1
A & B C	1 1 2
A & C B	1 2 1
A B & C	1 2 2
A & B & C	1 2 3

## 部分集団への分割を行うマクロ

```
%macro divide(indata1=, indata2=, outdata=, level=level, id=id, tmp=tmp);
proc sql noprint ;
create table &outdata (depth num, divide num, dest char(200), model char(200), pr num, ch num) ;
create table &tmp as
select 1 as depth,
       1 as divide,
       compress(put(struc.ch,best.)) as dest,
       compbl(ord.&level) as model,
       struc.pr,
       struc.ch
from &indata1 struc, &indata2 ord
where pr=. and calculated depth=ord.id ;
%do %while(&SQL OBS) ;
create table &outdata as select * from &outdata union all select * from &tmp :
create table &tmp as
select pr.depth+1 as depth,
       max(ch.pr,ch.ch) as divide,
       compbl(pr.dest||put(ch.ch,best.)) as dest,
       case when ch.pr<ch.ch then
         compbl(pr.model||" "||compress(ord.&level))
       else
         tranwrd(pr.model,scan(pr.model,ch.ch),compress(scan(pr.model,ch.ch)||"="||compress(ord.&level)))
       end
       as model,
       ch.pr,
       ch.ch
from &tmp pr, &indata1 ch, &indata2 ord
where pr.divide=ch.pr and pr.depth+1=ord.id ;
%end ;
drop table &tmp ;
quit ;
%mend ;
```



# モデル選択によって複数の水準間の平均値を考察

level	y
level1	129
:	:
level1	127
level2	198
:	:
level2	123
level3	188
:	:
level3	123

モデル:level1,level2,level3

$$\begin{pmatrix} y \\ 129 \\ : \\ 127 \\ 198 \\ : \\ 123 \\ 188 \\ : \\ 123 \end{pmatrix} = \begin{pmatrix} \text{切片 1} & 2 & 3 \\ 1 & 1 & 0 & 0 \\ : & : & : & : \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ : & : & : & : \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ : & : & : & : \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} + \epsilon$$

モデル:level1,level2=level3

$$\begin{pmatrix} y \\ 129 \\ : \\ 127 \\ 198 \\ : \\ 123 \\ 188 \\ : \\ 123 \end{pmatrix} = \begin{pmatrix} \text{切片 1} & 2\&3 \\ 1 & 1 & 0 \\ : & : & : \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ : & : & : \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ : & : & : \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} b \\ a_1 \\ a_2 \end{pmatrix} + \epsilon$$

モデル:level1=level3,level2

$$\begin{pmatrix} y \\ 129 \\ : \\ 127 \\ 198 \\ : \\ 123 \\ 188 \\ : \\ 123 \end{pmatrix} = \begin{pmatrix} \text{切片 1\&3} & 2 \\ 1 & 1 & 0 \\ : & : & : \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ : & : & : \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ : & : & : \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} b \\ a_1 \\ a_2 \end{pmatrix} + \epsilon$$

モデル:level1=level2,level3

$$\begin{pmatrix} y \\ 129 \\ : \\ 127 \\ 198 \\ : \\ 123 \\ 188 \\ : \\ 123 \end{pmatrix} = \begin{pmatrix} \text{切片 1\&2} & 3 \\ 1 & 1 & 0 \\ : & : & : \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ : & : & : \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ : & : & : \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} b \\ a_1 \\ a_2 \end{pmatrix} + \epsilon$$

モデル:level1=level2=level3

$$\begin{pmatrix} y \\ 129 \\ : \\ 127 \\ 198 \\ : \\ 123 \\ 188 \\ : \\ 123 \end{pmatrix} = \begin{pmatrix} \text{切片 1\&2\&3} \\ 1 & 1 \\ : & : \\ 1 & 1 \\ 1 & 1 \\ : & : \\ 1 & 1 \\ 1 & 1 \\ : & : \\ 1 & 1 \end{pmatrix} \begin{pmatrix} b \\ a_1 \end{pmatrix} + \epsilon$$

モデルから求められるAIC等の情報量基準の比較により、どの平均値が等しいか考察する

# モデル選択によって複数の水準間の平均値を考察

level	y
level1	5.042413687
:	:
level1	2.840268307
level2	1.088922739
:	:
level2	2.220740628
level3	0.699288068
:	:
level3	3.495788622

id	level
1	level1
2	level2
3	level3

+

model	dest
level1=level2=level3	1 1 1
level1=level2 level3	1 1 2
level1=level3 level2	1 2 1
level1 level2=level3	1 2 2
level1 level2 level3	1 2 3

←

```
proc glmmod ~ ;
class subgrp ;
model y=subgrp/point ;
by model ;
:
```

model	subgrp	subgrp	level	y
level1 level2 level3	level1	1	level1	5.042413687
:	:	:	:	:
level1 level2 level3	level1	1	level1	2.840268307
level1 level2 level3	level2	2	level2	1.088922739
:	:	:	:	:
level1 level2 level3	level2	2	level2	2.220740628
level1 level2 level3	level3	3	level3	0.699288068
:	:	:	:	:
level1 level2 level3	level3	3	level3	3.495788622
level1 level2=level3	level1	1	level1	5.042413687
:	:	:	:	:
level1 level2=level3	level1	1	level1	2.840268307
level1 level2=level3	level2=level3	2	level2	1.088922739
:	:	:	:	:
level1 level2=level3	level2=level3	2	level2	2.220740628
level1 level2=level3	level2=level3	2	level3	0.699288068
:	:	:	:	:
level1 level2=level3	level2=level3	2	level3	3.495788622
:	:	:	:	:
level1=level2=level3	level1=level2=level3	1	level1	5.042413687
:	:	:	:	:
level1=level2=level3	level1=level2=level3	1	level1	2.840268307
level1=level2=level3	level1=level2=level3	1	level2	1.088922739
:	:	:	:	:
level1=level2=level3	level1=level2=level3	1	level2	2.220740628
level1=level2=level3	level1=level2=level3	1	level3	0.699288068
:	:	:	:	:
level1=level2=level3	level1=level2=level3	1	level3	3.495788622

# モデル選択によって複数の水準間の平均値を考察

model			y	col1	col2	col3
level1 level2 level3			5.042413687	1	0	0
:			:	:	:	:
level1 level2 level3			2.840268307	1	0	0
level1 level2 level3			1.088922739	0	1	0
:			:	:	:	:
level1 level2 level3			2.220740628	0	1	0
level1 level2 level3			0.699288068	0	0	1
:			:	:	:	:
level1 level2 level3			3.495788622	0	0	1
level1 level2=level3			5.042413687	1	0	.
:			:	:	:	:
level1 level2=level3			2.840268307	1	0	.
level1 level2=level3			1.088922739	0	1	.
:			:	:	:	:
level1 level2=level3			2.220740628	0	1	.
level1 level2=level3			0.699288068	0	1	.
:			:	:	:	:
level1 level2=level3			3.495788622	0	1	.
:			:	:	:	:
level1=level2=level3			5.042413687	1	.	.
:			:	:	:	:
level1=level2=level3			2.840268307	1	.	.
level1=level2=level3			1.088922739	1	.	.
:			:	:	:	:
level1=level2=level3			2.220740628	1	.	.
level1=level2=level3			0.699288068	1	.	.
:			:	:	:	:
level1=level2=level3			3.495788622	1	.	.



```
proc reg ~ ;
model y=col: / AIC ;
by model ;
      :
      :
```

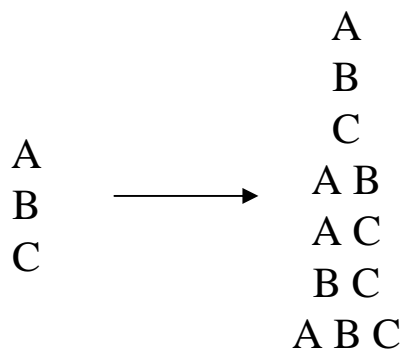


model	_AIC_
level1 level2=level3	112.759
level1 level2 level3	113.161
level1=level3 level2	157.824
level1=level2 level3	173.156
level1=level2=level3	177.316

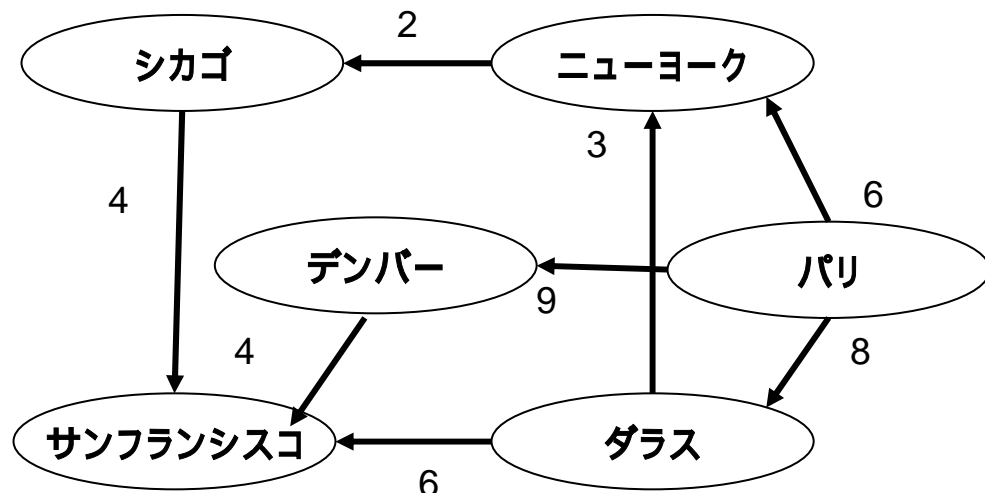
# まとめ

## 長所

・CやPascalにおける再帰と同じような処理をSASデータセットを対象として行える。



m個の要素から任意のn個を選んで記述



経路表から全ての経路を取り出す

## 注意

- ・データの性質によっては無限ループに陥ることがある。
- ・特に組合せ問題では、データ量が多くなると計算量が膨大になるため実行時間が長くなる。