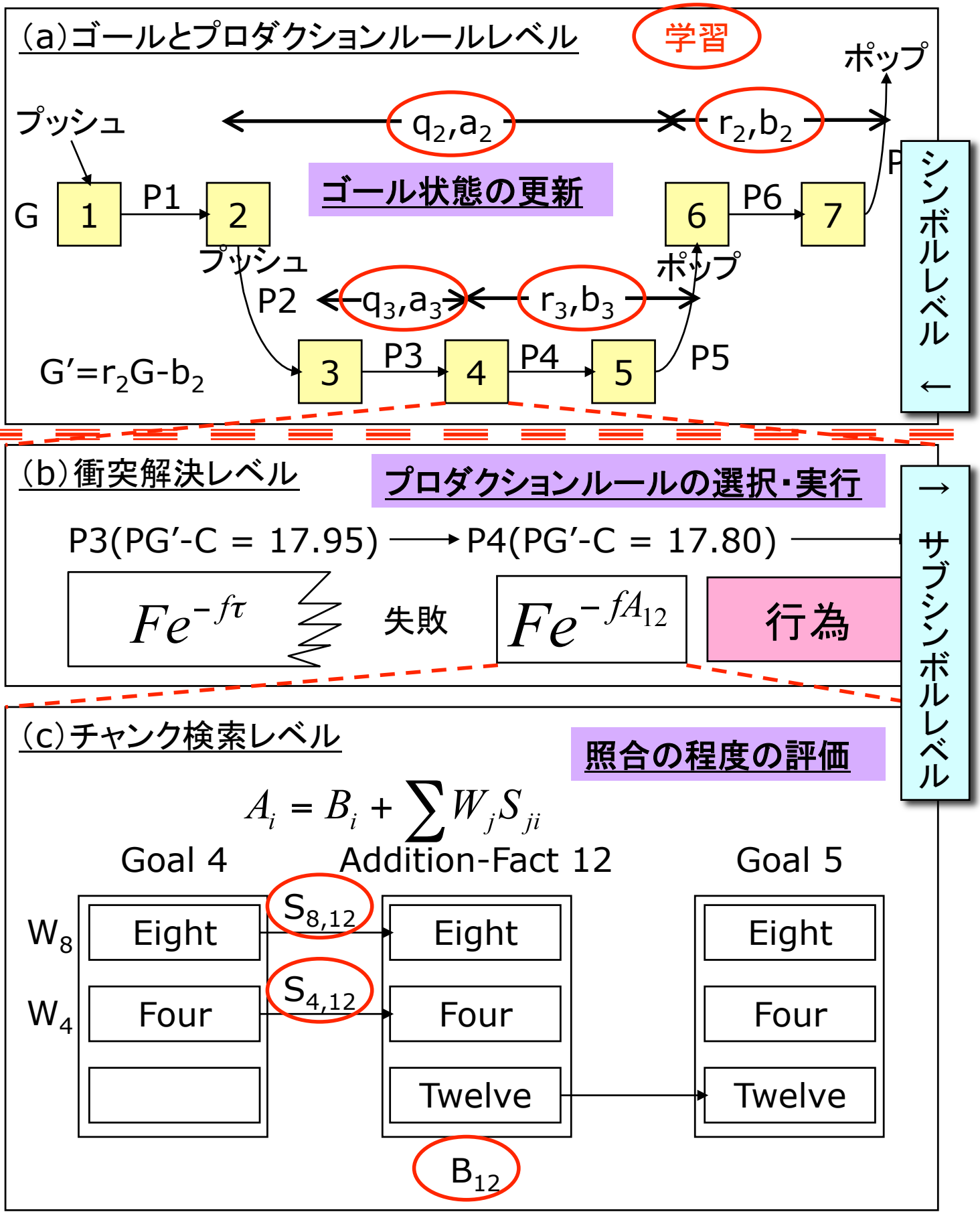
プロダクションシステム（Ⅲ）

The Atomic Components of Thought, John R. Anderson (著), Christian Lebiere (著) (1998/06) Lawrence Erlbaum Assoc Inc

1. 概要、知識表現　pp.1~55
2. 実行（パフォーマンス）　pp.57~100
3. 学習　pp.101~142

ACT-R における学習

- 知識の獲得
  - ▶ シンボルレベル
  1. チャンクの新規生成
     - 宣言的知識に新たなチャンクがどのようにして生成されるのか
  2. プロダクションルールの新規生成
     - 手続き的知識に新たなプロダクションルールがどのようにして生成されるのか
- 知識の強化（時間発展）
  - ▶ サブシンボルレベル
  3. チャンクの強度の強化・減衰
     - ひとたび獲得されたチャンクがどのように強化あるいは忘却されるのか
  4. プロダクションルールの強度の強化・減衰
     - ひとたび獲得されたプロダクションルールがどのように強化あるいは忘却されるのか


(a)ゴールとプロダクションルールレベル
学習
ポップ
プッシュ
q_2,a_2
r_2,b_2
ゴール状態の更新
G
1
P1
2
6
P6
7
プッシュ
ポップ
P2
q_3,a_3
r_3,b_3
$G'=r_2G-b_2$
3
P3
4
P4
5
P5
シンボルレベル ←
(b)衝突解決レベル
プロダクションルールの選択・実行
P3(PG'-C = 17.95) → P4(PG'-C = 17.80)
$Fe^{-f\tau}$
失敗
$Fe^{-fA_{12}}$
行為
→ サブシンボルレベル
(c)チャンク検索レベル
照合の程度の評価
$A_i = B_i + \sum W_j S_{ji}$
Goal 4
Addition-Fact 12
Goal 5
W_8
Eight
$S_{8,12}$
Eight
Eight
W_4
Four
$S_{4,12}$
Four
Four
Twelve
Twelve
B_{12}

1. チャンクの新規生成

- チャンクの源は2つ

1. 環境を符号化したもの
   - 具体的な事実に関する知識
2. プロダクションルールの行為部のゴールチャンク
   - 問題を解いた結果に関する知識
   - 同じ問題に遭遇したときに、答えを思い出して回答することが可能になる
   - 抽象的な知識

```
    2 3 9
  + 1 2 5
  -------
```

↓

数字1＞
isa 視覚対象物
value 9
row 上段
column 1の位

6. 変化ありポップ

6+3=?

数えるのを止める
=goal>
isa 合計演算
arg1 =数値1
arg2 =数値2
count =数値2
current =答え
==>
=goal>
sum =答え
!pop!

	0	1	2	3
arg1	6	6	6	6
arg2	3	3	3	3
count	0	1	2	3
current	6	7	8	9
sum	-	-	-	9

ポップされたゴールは宣言的知識として記憶される。将来、6+3=? に遭遇したとき、検索され、問題を解くのに利用される

6+3>
isa 合計演算
arg1 6
arg2 3
count 3
current 9
sum 9

認知科学 by 北島宗雄

ゴールチャンクの学習の例 ～和の表の学習～

- 「６＋３＝？」に答えるとき、宣言的記憶から検索して答える場合と、カウントアップして答える場合がある
  - プロダクションルールは、次ページを参照

1. 答えが記憶にあれば、プロダクションルール「P1:記憶を検索する」を適用
2. 記憶検索に失敗したら、プロダクションルール「P2:カウントする」を適用
   - これには時間がかかるので、衝突解決プロセスでは、「記憶を検索する」よりも期待利得が小さく評価される

「和を答える」に対して適用されるプロダクションルール

P1:和を検索する
IF ゴールは「n1とn2の和を答える」、かつ、n1とn2の和が記憶に存在する
THEN ゴール「和を答える」をプッシュする

P2:カウントする
IF ゴールは「n1とn2の和を答える」
THEN ゴールを「和を答える」をプッシュし、さらに、ゴール「n1とn2の和を求める」をプッシュする

P3:カウントを開始する
IF ゴールは「n1とn2の和を求める」
THEN ゴール「n1を、0からn2までカウントアップする」をプッシュする

P4:和を求めるゴールをポップする
IF ゴールは「n1とn2の和を求める」、かつ、和が得られている
THEN ゴールをポップする

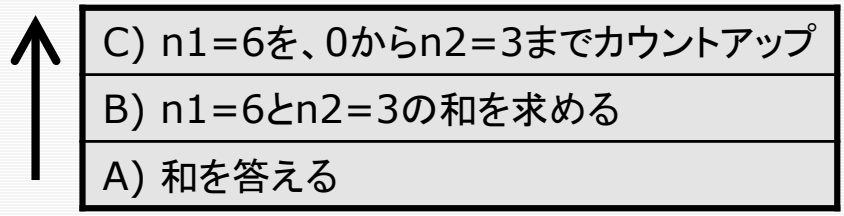
P5:カウントする
IF ゴールは「n1を、n3からn2までカウントアップする」、かつ、n4=n1+1、かつ、n5=n3+1
THEN ゴールを「n4を、n5からn2までカウントアップする」に変更する

P6:カウントを終了する
IF ゴールは「n1を、n2からn2までカウントアップする」
THEN ゴールをポップし、答えにn1を入れる

P7:答える
IF ゴールは「和を答える」
THEN 答えを書き出し、ゴールをポップする

６＋３＝？の実行

ゴールスタック



1. P2:
   IF　ゴールは「n1=6(number1)とn2=3(number2)の和を答える」
   THEN プッシュ「和(sum)を答える」(A)
   　　プッシュ「n1=6(arg1)とn2=3(arg2)の和を求める」(B)
2. P3:
   IF ゴールは「n1=6(arg1)とn2=3(arg2)の和を求める」B
   THEN プッシュ「n1=6(current)を、
   　　0(count)からn2=3(arg2=number2)までカウントアップ」C
3. P5:
   IF　ゴールは「n1=6を、n3=0からn2=3までカウントアップ」(C)、
   　　かつ、「n4(=7)=n1(=6)+1」「n5(=1)=n3(=0)+1」
   THEN ゴール変更「n4=7(current)を、
   　　n5=1(count)からn2=3(arg2)までカウントアップ」
4. P5:
   IF　ゴールは「n1=7を、n3=1からn2=3までカウントアップ」(C)、
   　　かつ、「n4(=8)=n1(=7)+1」「n5(=2)=n3(=1)+1」
   THEN ゴール変更「n4=8(current)を、
   　　n5=2(count)からn2=3(arg2)までカウントアップ」
5. P5:
   IF　ゴールは「n1=8を、n3=2からn2=3までカウントアップ」(C)、
   　　かつ、「n4(=9)=n1(=8)+1」「n5(=3)=n3(=2)+1」
   THEN ゴール変更「n4=9(current)を、
   　　n5=3(count)からn2=3(arg2)までカウントアップ」
6. P6:
   IF　ゴールは「n1=9(current)を、
   　　n2=3(arg2)からn2=3(count)までカウントアップ」(C)
   THEN ゴール(C)をポップ、答えにn1(sum)を入れる
7. P4:
   IF　ゴールは「n1=6とn2=3の和を求める」(B)、
   　　かつ、和が得られている
   THEN ゴール(B)をポップ
8. P7:IF ゴールは「和を答える」(A)
   THEN 答えを書き出し、ゴールをポップ(A)

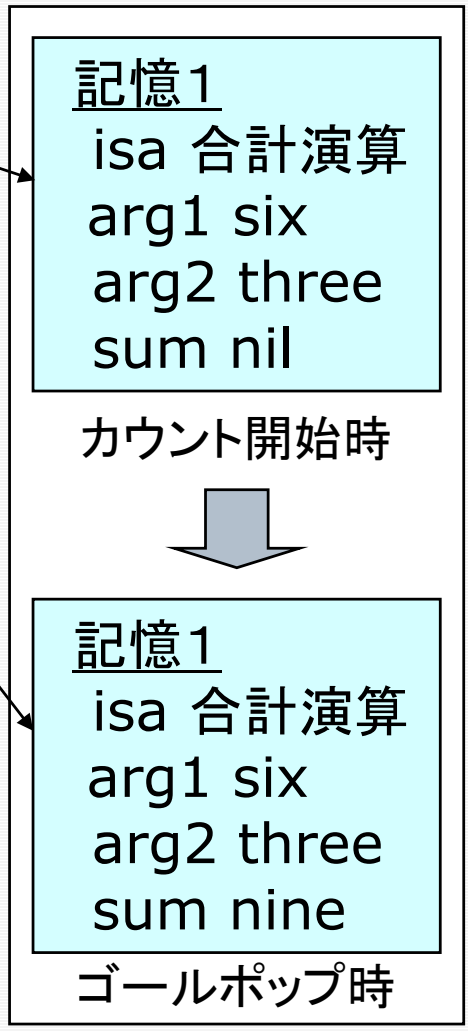
６＋３＝９　の学習

最初に、６＋３＝？に答える場合

1. P2:カウントする
2. P3:カウントを開始する
3. P5:カウントする
4. P5:カウントする
5. P5:カウントする
6. P6:カウントを終了する
7. P4:和を求めるゴールをポップする
8. P7:答える
   - ▶ NINE

次に、６＋３＝？に答える場合

9. P1:和を検索する
10. P7:答える
    - ▶ NINE



「n1とn2の和を求める」を表現するゴールチャンク

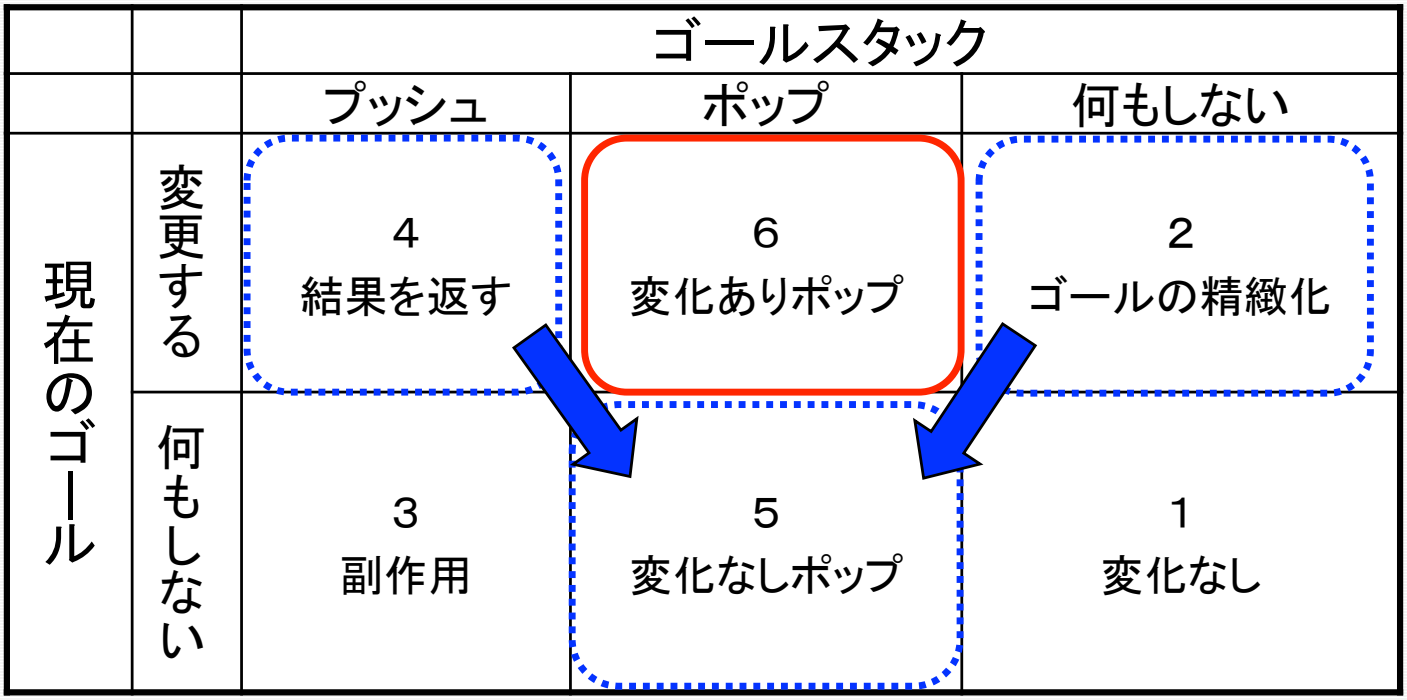
2. プロダクションルールの新規生成

ここまでは、プロダクションルールが既に存在するものとして説明してきたが、ここでは、それが、どのようにして新たに生成されるのかを説明する

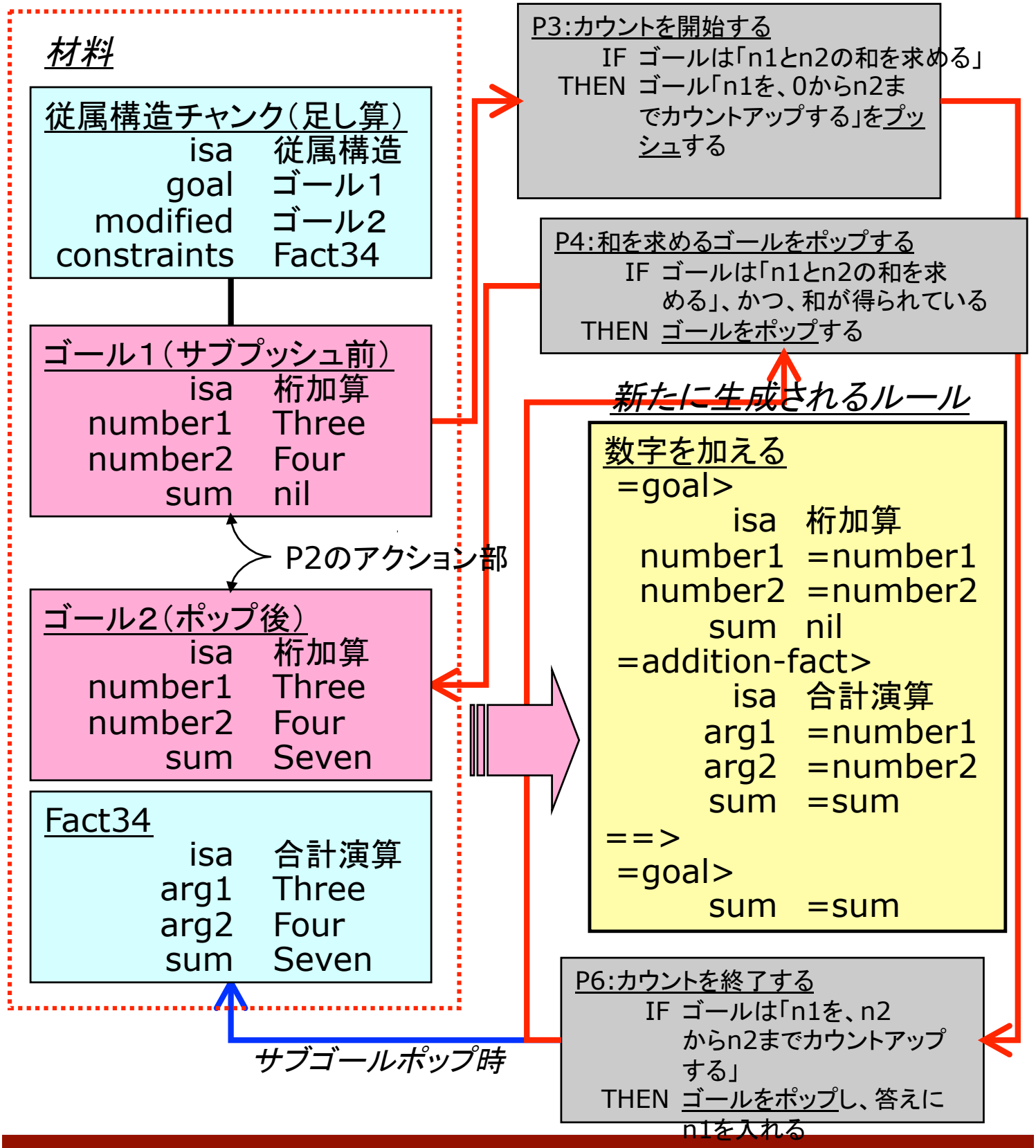
従属構造を利用したプロダクションルールの生成

- プロダクションルールはゴール構造を変換する
- 現在のゴールがポップされたとき、ゴール構造がどのように変更されたかを分析して、プロダクションルールを生成する
- 利用するチャンク

1. 問題を解く前のゴールの状態を記憶するチャンク
2. 問題を解いた後のゴールの状態を記憶するチャンク
3. 前後の関係（従属構造）を記憶するチャンク



従属構造からプロダクションルールを生成する ～足し算問題の場合～



従属構造からプロダクションルールを生成する（説明）

- 問題解決行動エピソードのなかの特定のステップを理解しようと考えたときに、チャンクタイプ「従属構造」を持つゴールチャンクが生成され、ゴールスタックにプッシュされる
  ・・・　従属構造チャンク（足し算）のプッシュ

- 従属構造ゴールがポップされるときに、プロダクションルールが生成される
  ・・・　プロダクションルール「数字を加える」が生成される

- 適切に定数を変数化することによって、抽象度を高める
  1. 条件部、行為部の複数のスロットに共通する定数は、変数化する
     - Three ⇒ =number1
     - Four ⇒ =number2
     - Seven ⇒ =sum
  2. 同様に、チャンク名を変数化する
     - Fact34 ⇒ =addition-fact
  3. スロットの値として1度しか現れないものは、定数とする
     - nil

3．チャンクの強化

イベントの生起確率の評価を統計的な推論により行う（統計的学習）

★ どのチャンクを検索するのか

★ どれくらい速く検索できるのか

□ 基礎活性の学習

□ 連合強度の学習

$$A_i = B_i + \sum_j W_j S_{ji}$$

チャンクの活性の表式

- チャンクの活性
  ＝そのチャンクが現在のサイクルでプロダクションにマッチする（必要とされる）ということに関する対数事後オッズ

H_i; チャンク i が必要とされるという仮説
E ; 現在のコンテクストに現れる要素（スロット）

$$\frac{P(H_i \mid E)}{1 - P(H_i \mid E)} = \frac{P(H_i \mid E)}{P(\overline{H}_i \mid E)} = \frac{P(H_i)}{P(\overline{H}_i)} \prod_{j \in E} \frac{P(j \mid H_i)}{P(j \mid \overline{H}_i)}$$

事後オッズ　事前オッズ　尤度比

$$P(H_i \mid E)P(E) = P(H_i)P(E \mid H_i)$$
$$P(\overline{H}_i \mid E)P(E) = P(\overline{H}_i)P(E \mid \overline{H}_i)$$

j が独立

$$\ln\left[\frac{P(H_i \mid E)}{P(\overline{H}_i \mid E)}\right] = \ln\left[\frac{P(H_i)}{P(\overline{H}_i)}\right] + \sum_{j \in E} \ln\left[\frac{P(j \mid H_i)}{P(j \mid \overline{H}_i)}\right]$$

$$A_i = B_i + \sum_j W_j S_{ji}$$

W_j はキューの妥当性を表す

基礎活性 B_i の学習

- 基礎活性は、コンテクストに依存せずに、次のサイクルでチャンクがプロダクションに必要とされる程度を示す指標
- そのチャンクが必要とされた頻度、最後に必要とされてからの経過時間に依存（Anderson & Schooler, 1991）

$$\text{Odds} = \frac{P(H_i)}{P(\overline{H}_i)} = a\sum_{j=1}^{n} t_j^{-d}$$

チャンク i が過去の $t_1, t_2, \ldots, t_n$ に必要とされた

基礎活性学習方程式

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) + \beta$$

$$\beta = \ln(a)$$

基礎活性の学習の例

- 200 秒の間に、20 回、チャンクが必要とされる
- 必要とされたタイミングはランダム
- β= 0, $d = 0.5$

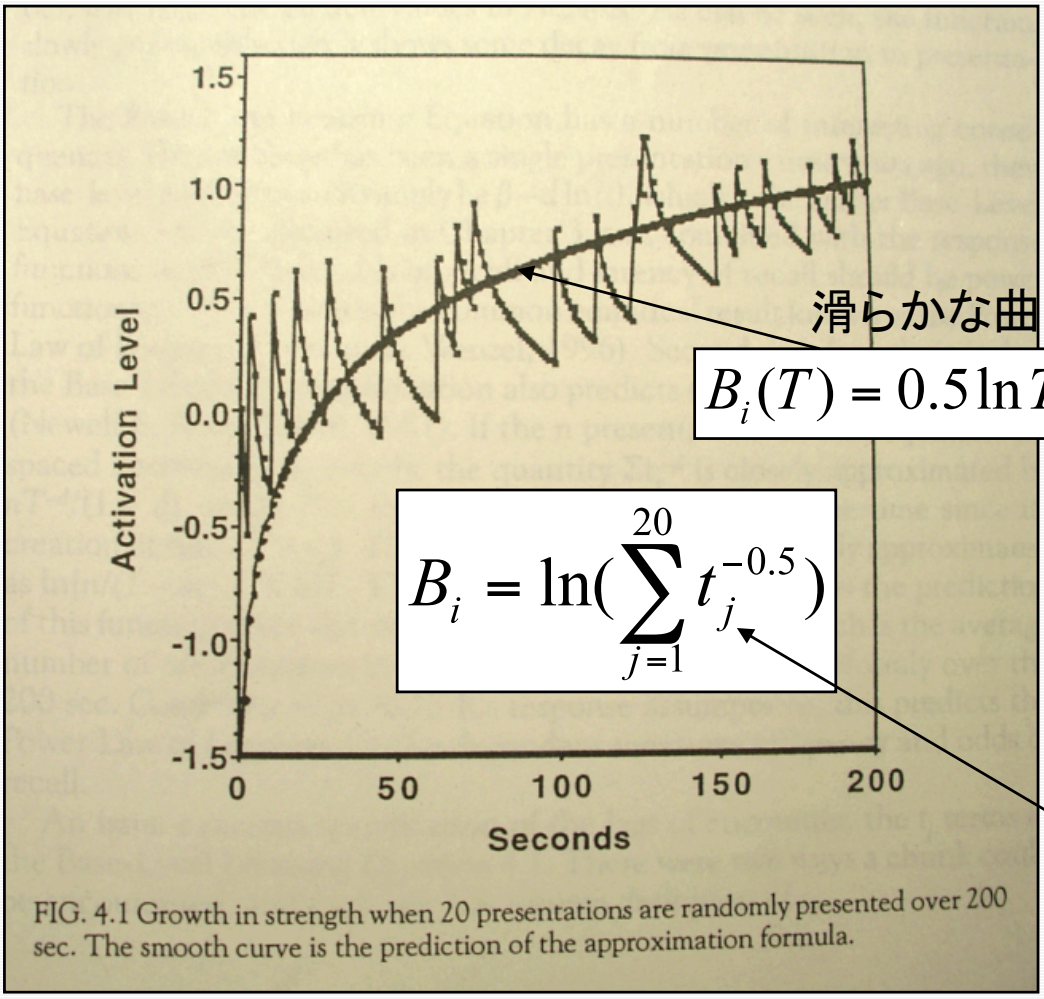


FIG. 4.1 Growth in strength when 20 presentations are randomly presented over 200 sec. The smooth curve is the prediction of the approximation formula.

基礎活性学習方程式の性質

- t 秒前に1度だけ必要とされた場合の基礎活性　・・・　基礎活性方程式となる

PS（Ⅱ）28ページ

$$B_i = \beta - d\ln(t)$$

- 検索確率方程式と組み合わせると、検索確率のオッズ比、検索潜時は、時間のべき関数になる

検索確率のオッズ比

$$Odds = \frac{P}{1-P} = e^{(A-\tau)/s} = e^{(\beta - d\ln(t)-\tau)/s} = kt^{-c}$$

$$k = e^{(\beta-\tau)/s}, \quad c = d/s$$

忘却のべき法則

$$\mathrm{Time}_i = Fe^{-B_i} = Fe^{-\beta+d\ln(t)} = k't^d$$

基礎活性学習方程式の性質

- 練習のべき法則が導かれる

$$T_n = T_1 \times n^{-\alpha}$$

- n 回の機会が均等に生じていたと仮定する

$$\sum_{j=1}^{n} t_j^{-d} \cong \frac{nT^{-d}}{1-d}\,;\, T = t_n - t_1$$

$$B = \ln\left[\frac{nT^{-d}}{1-d}\right] = \ln\left[\frac{n}{1-d}\right] - d\ln T$$

$$\text{Time} = Fe^{-B} = F \times \frac{1-d}{\mu^d} \times n^{-(1-d)}\,;\, n = \mu \times T$$

連合強度 S_{ji} の学習

- 連合強度 S_{ji} は、スロット j に現在の値が現れているということによって、チャンク i が必要とされる確率がどれくらい増加するかを表す
- 例えば、チャンク「4+8=12」は、「4」がゴールスロットにあることによって、必要とされる確率が高くなる

C_j; jがゴールスロットに存在するというイベント

N_i; チャンクi必要とされるというイベント

$$S_{ji} = \ln\left[\frac{P(C_j \mid N_i)}{P(C_j \mid \overline{N}_i)}\right] \cong \ln(R_{ji}) \longleftarrow$$ *14ページの式参照*

ここで、$$R_{ji} = \frac{P(C_j \mid N_i)}{P(C_j)} = \frac{P(N_i \mid C_j)}{P(N_i)}$$

なぜならば、ほとんどのiにおいて、

$$P(C_j \mid \overline{N}_i) = \frac{P(\overline{N}_i \mid C_j)}{P(\overline{N}_i)} \times P(C_j) \cong P(C_j)$$

初期値 R^*_{ji} の計算法

- チャンクｉが生成されたとき、初期値 R^*_{ji} が与えられる

j が *i* に現れていない場合

$$R^*_{ji} = 1 \quad \cdots \quad S_{ji} = 0$$

j が *i* に現れている場合

$P(N_i \mid C_j) = \frac{1}{n}$; nは、jに結合しているチャンク数

$P(N_i) = \frac{1}{m}$; mは、チャンク数

$$R^*_{ji} = \frac{1/n}{1/m} = \frac{m}{n}$$

$$S_{ji} = \ln(m) - \ln(n)$$

R_{ji}の計算法

□ 経験に基づいて初期値を更新

$$R_{ji} = \frac{\alpha \times R^{*}_{ji} + F(C_j) \times \dfrac{P_e(N_i \mid C_j)}{P_e(N_i)}}{\alpha + F(C_j)}$$

$$P_e(N_i \mid C_j) = \frac{F(N_i \,\&\, C_j)}{F(C_j)}$$

$F(N_i \,\&\, C_j)$; jが存在するときにiが必要とされた回数

$F(C_j)$; jがゴールスロットに現れた回数

$$P_e(N_i) = \frac{F(N_i)}{F(i)}$$

$F(N_i)$; iが必要とされた回数

$F(i)$; iが生成されてからのプロダクション発火回数

4．プロダクションルールの強化

行為のコストの評価を統計的な推論により行う（統計的学習）

★　どのプロダクションを発火させるのか

★　どれくらいの速度で発火できるのか

<u>関連するパラメタ</u>

S_p：プロダクション強度

q：そのプロダクションが成功する確率

r：以降のプロダクションが成功する確率

a：そのプロダクションの発火コスト

b：以降のプロダクションの発火コスト

プロダクション強度の学習

- プロダクションが次のサイクルで発火するということに関する対数オッズの推定値になるので、基礎活性学習方程式と同一の表式になる

プロダクション強度方程式

$$S_P = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) + \beta$$

このプロダクションルールが過去の $t_1, t_2, \ldots, t_n$ に発火した
$d = 0.5$（デフォルト）

プロダクション強度の特徴

- プロダクション強度は、プロダクションルールが衝突解決プロセスで選択されたときに、それが、実行されるときの速度を決定する　$\propto e^{-Sp}$
- プロダクション強度が大きいからといって、衝突解決プロセスで有利になるわけではない

- 使えば使うほど速くなる　⇒ Thorndike’s Law of Exercise
- より多く使ったからといって、選ばれやすくなるわけではない。成功率と、コストが選択の基準　⇒　Thorndike’s Law of Effect (1913)

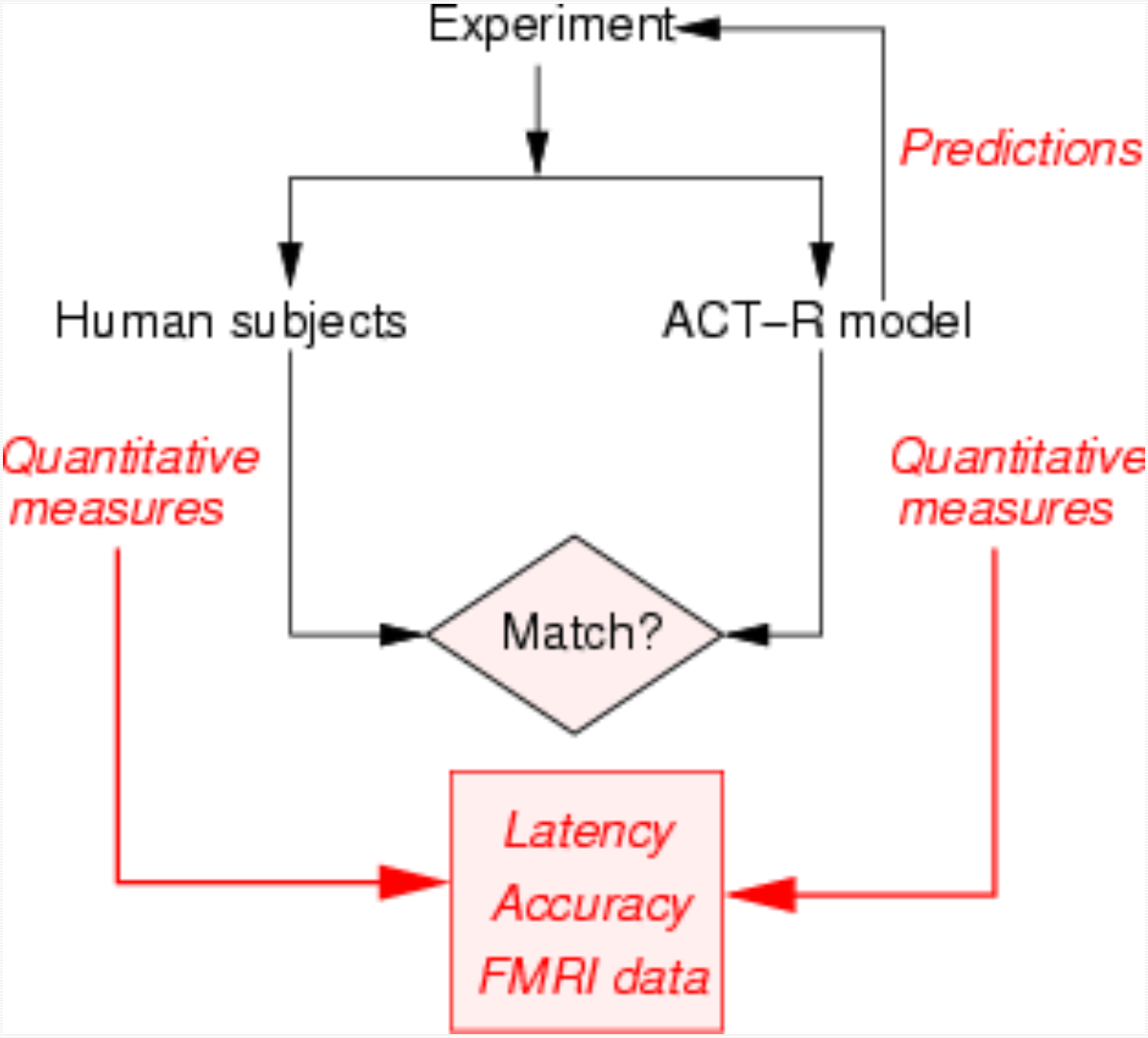
プロダクションルールの成功率とコスト

$$E = PG - C = q \times r \times G - (a + b)$$

$$q = \frac{\text{Successes}}{\text{Successes} + \text{Failures}}$$

$$a = \frac{\text{Efforts}}{\text{Successes} + \text{Failures}}$$

Efforts は、そのプロダクションルールを発火させたときに、要した時間。成功した場合、失敗した場合を含む

ACT-Rを利用した認知の解明

http://act-r.psy.cmu.edu/