

Title	強化スケジュール表記システムについての覚書
Sub Title	Notes on the reinforcement schedule notation system
Author	坂上, 貴之(Sakagami, Takayuki)
Publisher	三田哲學會
Publication year	2013
Jtitle	哲學 No.130 (2013. 3) ,p.1- 40
JaLC DOI	
Abstract	<p>Several notation systems for reinforcement schedules have been developed just after the start of this research field. Some systems provided good tools for creating complex schedules and for educational purposes, and sometimes they were interpreted to some common computer languages via specific invented languages. The most successful system among them was the State Notation System(SNS) developed by Snapper and his colleagues and their interpretable language SKED. In this paper, I propose another notation system for reinforcement schedule based on these previous efforts, particularly SNS and SKED. This new system, named Reinforcement schedule notation system (RNS), is composed with two subsystems, the Structural schedule and the Constituent schedule, which are mutually complimented their limited scopes. The Structural schedule consists of the Scenes and the Conditions which sketch the outline of reinforcement schedules an organism faces to. On the other hand, the Constituent schedule consists of connected chains of the Schedule Elements which contains three slots of (1) the evaluation of either response or time counters, (2) the operation of environmental changes and system signals, and (3) the jumped addresses depended on the result of the evaluation slot. Although RNS provides compact descriptions for the complex structures of reinforcement contingencies and many ideas for concrete operations of reinforcement schedules, it has several weak points, such as no elegant descriptive devices for dealing with serial responses (e.g., double alternate learning) and complex configurations of discriminative stimuli (e.g., higher conditional discrimination). These limitations will be solved by development of frameworks for analysis of discriminative stimuli in future.</p>
Notes	特集 : 渡辺茂君・増田直衛君退職記念

	投稿論文
Genre	Journal Article
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN00150430-00000130-0001

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the KeiO Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

投稿論文

強化スケジュール表記システム についての覚書

坂 上 貴 之*

Notes on the Reinforcement Schedule Notation System

Takayuki Sakagami

Several notation systems for reinforcement schedules have been developed just after the start of this research field. Some systems provided good tools for creating complex schedules and for educational purposes, and sometimes they were interpreted to some common computer languages via specific invented languages. The most successful system among them was the State Notation System (SNS) developed by Snapper and his colleagues and their interpretable language SKED. In this paper, I propose another notation system for reinforcement schedule based on these previous efforts, particularly SNS and SKED. This new system, named Reinforcement schedule notation system (RNS), is composed with two subsystems, the Structural schedule and the Constituent schedule, which are mutually complimented their limited scopes. The Structural schedule consists of the Scenes and the Conditions which sketch the outline of reinforcement schedules an organism faces to. On the other hand, the Constituent schedule consists of connected chains of the Schedule Elements which contains three slots of (1) the evaluation of either response or time counters, (2) the operation of environmental changes and system signals, and (3) the jumped addresses depended on the result of the evaluation slot. Although RNS provides compact descriptions for the complex structures of reinforcement contingencies.

* 慶應義塾大学文学部

cies and many ideas for concrete operations of reinforcement schedules, it has several weak points, such as no elegant descriptive devices for dealing with serial responses (e.g., double alternate learning) and complex configurations of discriminative stimuli (e.g., higher conditional discrimination). These limitations will be solved by development of frameworks for analysis of discriminative stimuli in future.

1. はじめに

自発性の行動であるオペラントとそれに後続して随伴された環境事象である強化子や弱化石との時間的確率的関係は強化随伴性と呼ばれ、これまで強化スケジュールの研究を通じて、多くの行動分析学の研究者の重要な研究課題の1つであった。さらにこの強化随伴性に先行する環境事象に対して研究者たちは弁別刺激という用語を当て、それら弁別刺激、オペラント、強化（弱化石）子の3項よりなる関係を、3項強化随伴性と呼んで、オペラント行動の基本的な分析枠とした。そして伝統的には、弁別刺激とオペラントとの関係を分析する研究領域は刺激性制御、オペラントと強化（弱化石）子との関係を分析する研究領域は強化スケジュールと呼ばれてきた。

しかしながら半世紀を越える強化スケジュールの研究の蓄積は、強化スケジュールを3項強化随伴性の後半に留めるというよりも、3項強化随伴性における行動と環境間関係を表した「規則」及びそれに関わる「研究」を指すものへと、その意味内容を拡張させてきたように思われる。たとえばその「規則」について見るならば、特定化された行動のどの側面をオペラントと定義し、そのオペラントの出現に先行あるいは後続する特定化された刺激をどのように配置するのかを記述する。いわば3項強化随伴性全体を実験者側の操作として表現したものが、21世紀初頭における、強化スケジュールの、「規則」としての新しい用法といえる。本稿で考えたいのは、その強化スケジュールの「規則」としての表現の様式、すなわ

ち表記（記法）についてである。

すでにこれまでいくつもの強化スケジュールの表記システムが開発されてきた (Findley, 1962; Mechner, 1959; Skinner, 1958; Snapper, Kadden, and Inglis, 1982). それと同時に, オペラント行動実験の手続きを記述するための数多くの言語もまた開発されてきた (e.g., 浅野, (OPEAB); Elsner, 1982 (PASTOR); Hori, Tanaka, & Nomura, 1993 (SN-FORTH); 久保田ら, 1981 (XANOP, PSI); Snapper, Kadden, and Inglis, 1982 (SKED, SUPER SKED, SKED11)). こうした言語は, 特定の表記システムとは比較的独立して考えられたものもあるが, 特定の表記システムに大きな影響を受けて作られたものもあった. Snapper らの SKED 言語はそうした言語の代表ともいえる. 本稿では, 基本的には表記システムについて考え, そのソフトウェア的な実現としての実験手続き言語については立ち入らない.

表記システムには2つの重要な役割がある. 1つは研究者に対して, 強化スケジュールひいてはその基盤にある強化随伴性を詳細に検討することができるようにし, 新しい強化スケジュールや随伴性の発見に使えるよう形式的な表現を与える. もう1つは学生の教育のためであり, オペラント行動研究における, 複雑なスケジュールの性質を彼らがよりよく理解するために用いられる (Michael & Shafer, 1995). 本稿では, 教育的な利用としての表記システムについては基本的に論じない.

現在において, 表記システムは Snapper らの状態表記システム (State Notation System, SNS) を最後に, 斬新なものは開発されていない. その理由の1つとして, SNS は, 強化スケジュールをプログラミング言語 (SKED) としてそのまま書き下すことができるまでに洗練されたことを挙げることができる. たとえば SKED はコンパイラを使って別のプログラミング言語 (たとえば PDP-8 のマクロアセンブラ言語) に書き直され, さらにそこから機械語に変換されて, 最終的にはコンピュータとその入出

強化スケジュール表記システムについての覚書

力装置に強化スケジュールを実装することができる。一般に、強化スケジュールの種類は無限に考えることができるので、その詳細は、最後には融通性の高い一般的なプログラム言語の力を借りて書き下すことが必要になる。したがってスケジュール自体を表す表記は、大雑把で融通の利くもので事足りてしまうので、たとえば状態表記システム程度のもので十分と考えられてきたと思われる。

しかし、融通の利く SNS であっても、いくつかの不满はある。この SNS の表記法は、図で表現するからわかりやすいのであって、テキストとして表現できないし、またそのような表現にはむいていない。仮にテキストとして表現したとしても、SNS をプログラミング言語として書き起こすのに用いられる SKED 言語は、ほとんど通常のプログラム言語のサブルーチンを縦方向に並べているにすぎず、そこから強化スケジュールの構造を想像したり吟味したりすることは難しい。

さらに SNS では、複数のオペランドにおいて異なる事態が同時に起こっているような実験状況（たとえば並立連鎖スケジュール、条件性マッチング課題、ヨークト（連結）制御など）を表すのが難しかったり、複雑であったりする。また移行条件を確率的に決めたり、他の状態での特定の値に応じて現在の状態でのパラメータを決定したりするようなケースも、SNS のオリジナルな表現では表すことが難しく、これまでに、比較を行ったうえでの分岐（IF…THEN）やシステムの状態を知らせる「パルス」（Z-PULSE）を組み込んだ修正が考案されてきた。

一方、行動の制御を細部まで行うような実験的研究では、これまで一般的に用いられてきた標準的強化スケジュールに代わって、複雑な随伴性を持つ強化スケジュールが使用されるようになってきた。たとえば左のレバーへの反応の回数に応じて右レバーへの反応の強化確率が変化するようなスケジュールは、FCN（Fixed Consecutive Number, 固定長反応連）スケジュールを変形したもので表現できるが、こうしたスケジュールに見られ

る同時平行に起こっている事象間でのデータのやり取りを表現することは、もともとの SNS で表すことが難しい。

以上の問題から、本稿では強化スケジュールの新しい表記システム（今後、この新しい提案システムを Reinforcement schedule Notation System, RNS と呼ぶ）について、1つの提案を試みると同時に、その限界についても述べたいと思う。当初から提案するものの限界について述べるというのは、いささか著者の努力に疑いの目を向けられてしまうが、RNS には、上でもふれたように致命的な問題が存在する。それは強化スケジュールの完全記述を行うには、一般的なプログラム言語のほうが能率がよいということである。しかし、研究者にとって、そのようなプログラム言語での記述は、強化スケジュールが表現することとなる操作や随伴性への直観的な理解をもたらさない。したがって、厳密な記述と直感的な理解とのバランスを図った表記システムが、本論文の目標となる。そして残念ながら、このような目標を持った表記システムには、たいていの場合、何らかの瑕疵が伴ってしまうのである。

2. RNS による強化スケジュールの表記法

1. 実験セッション

RNS の表記の対象は、研究対象となる個体が経験する 1 つの実験セッションでの、研究者が用意した強化スケジュールである。この実験セッションは、たいていの場合、複数の強化スケジュールの組み合わせからできている。

強化スケジュールには、個体が経験すると想定された個別の随伴性の操作そのものを記述する「要素スケジュール」と、実際に個体が直面する個別の随伴性の組み合わせを記述する「構成スケジュール」との 2 つを区別することができる。そこで、RNS は要素スケジュールと構成スケジュールの 2 つのサブシステムを持つと考える。

実験セッションは、まず、構成スケジュールによって個体が直面する随伴性の全体像が述べられる。引き続きその構成スケジュールを成り立たせている個別随伴性の細部が要素スケジュールによって述べられ、具体的にどんな随伴性を作ろうとしているのかを明確に表記する。なおこれ以降で表記に用いられる文字や記号については、表1にまとめておく。

2. 構成スケジュールの表記法

本稿では、RNSを考えるうえで、研究者がこれまでその努力を傾注してきた個別の強化スケジュールの細部の記述よりも、それら個別の強化スケジュールをどのように組み合わせ、実験セッションを成立させるかを記述する構成スケジュールに重点を置く。構成スケジュールと要素スケジュールとの仲立ちをするものとして、ここでは「シーン」という概念を導入する。すなわち構成スケジュールを、個体が置かれている状態を表すシーンの継時的な接続と考え、そのシーンが成立する要件を定めた「継続条件」及びそのシーンの成立に重要な役割を果たす「オペランダム」や「ディスクリミナナム」とともに記述することによって、随伴性の大まかな表現を与える。一方、各シーンは要素スケジュールによってその随伴性の細部が定義され、ある程度実行可能な手続きの詳細がそれらによって表現される。

構成スケジュールの表記は以下のものによってなされる。

2.1 シーン S, E

シーンとは、1つ以上の要素スケジュールからなる、個体が直面している状態を表す。各シーンは次にそのシーンが書き換えられるまでシーンに関わるいくつかのデータを保持すると仮定する^{注1)}。これらのデータは継

^{注1)} 4.1で述べるリセットによる書き換えは変数のリセット以外に影響を与えないと想定しているので、このシーンの書き換えには含まれず、データは保持される。

表1 RNSによる表記で用いられた記号と各スケジュールでの意味

記号	意味	第2節内での場所
[]	オペランド	2.2
{ }	ディスクリミナンド	2.3
< >	継続条件	2.9
()	一般的なグルーピング	2.8
()	スケジュール要素のスロットとパラメータ	4.1
()	関数パラメータ	2.8
:	継時的接続	2.5
,	同時的存在	2.6
,	スロット内のパラメータの区切り	4.1.3
\	排他的存在	2.7
\$ \$	繰り返し	2.11
;	スケジュール要素内のスロットの区切り	4.1
.	スケジュール結合子	4.2
!	生成環境変化における提示終了	4.3
'	分	2.11, 4.1.2
”	秒	4.1.2
#	コメント、構成スケジュール	6.3
@	定義	6.3
?	評価	第3節3)
%	分岐点と出力の組み合わせの区切り	第3節注3
A	アスペクト	6.1
B	ブラケット	6.2
c	比較子	4.1.7
D	ディスクリミナンド提示	4.3
E	強化事象を表すシーン	2.1
F	強化子提示	4.3
f	固定	4.1.1
g	Z信号評価子	4.1.6
h	時隔内反応子	4.1.4
I	シーン開始から中断までの経過時間	2.10
j	関数依存	4.1.1
K	オペランド提示	4.3
L	シーンが中断した際の復帰場所	2.10
M	イベント回数	2.10
n	分岐点子	4.1.8
o	時隔内無反応子	4.1.3
p	確率評価子	4.1.5

強化スケジュール表記システムについての覚書

表1 つづき

記号	意味	第2節内での場所
Q	確率発生	2.4
R	反応回数	2.10
r	反応子	4.1.1
S	シーン（添え字なしは実験セッション全体）	2.1
T	経過時間	2.10
t	時隔子	4.1.2
U	スケジュールパラメータ、実験セッション定数	2.10
V	値	2.10
v	変動	4.1.1
W	関数計算	4.3
x	変数	2.14
y	変数	2.14
Z	Z信号発生	2.4
&	論理積	2.12
	論理和	2.12
~	否定	2.12
==	等しい	2.12
~=	等しくない	2.12
>=	以上	2.12
<=	以下	2.12
>	越えて	2.12
<	未満	2.12
=	代入	2.12
+	和	2.12
-	差	2.12
*	積	2.12
/	商	2.12
^	べき乗	2.12
τ	時隔定数	2.13
ρ	反応回数定数	2.13
π	確率定数	2.13
α	その他の定数	2.13

続条件の判定の際に、データ呼び出し関数によっていつでも呼び出すことができる。どのようなデータかについては、2.10 のデータ呼び出し関数で詳しく述べる。なお、この関数が取り扱う以上の規模の保持データ（たとえば反応間時間の分布、長い反応連や時系列データ）を必要とする RNS については、本稿では取り扱わない。

ここで取り上げる主要なシーンには以下の S と E がある。

S 添え字のついたシーン S とは、個体がさらされている特定可能な状態であり、その間の環境条件は一定である。添え字のない S は全部のシーン、すなわち実験セッション全体を表す。

E E は一般的な強化子（例えば餌）のあるシーンを表す。後述するスケジュール要素での表記法では、（第 2 スロット *envc* 内での）あるスケジュール要素での生成環境変化 F からその後のスケジュール要素での生成環境変化 !F までの間が、E に対応する。

2.2 オペランドム []

個体の行動が環境に対して効果を与える環境との接点は、オペランドムと呼ばれてきた。たとえばハトのキイツつき行動では、キイへの反応が入力装置を介してコンピュータに取り込まれるので、キイがオペランドムとなる。キイへの反応型は問われず、キイに取り付けられたマイクロスイッチを閉じさえすればよい。もしも特にオペランドムを区別する必要があるれば []₁ のように添え字をつける。

2.3 ディスクリミナダム { }

個体のオペラント行動の出現機会を制御する先行刺激は弁別刺激と呼ばれ、その弁別刺激の物理化学的実体をディスクリミナダムと呼ぶ。異なるオペランドムは、異なるディスクリミナダムとなるので、冗長を避け

るために、それらが一対一対応をしている場合には、[]のみで表すが、たとえば同一オペランドの中でキイの色が変わる場合には $\{ \}_{1} : \{ \}_{2}$ というように区別して表記する。添え字は誤解が生じないのであれば、なくてよい。

2.4 システム信号

システム信号とは、個体が接近できない、RNSが生成する信号で、この信号を他のシーンや継続条件が受け取ることで、そのシーンを決定するパラメータの一部となったり、継続条件の判定対象となったりするものである。ここではZとQの2つのシステム信号を考える。

Z 主に先行するSやEのシーンの終了を知らせる役目を持つ、システム信号を表す。

Q 確率を生成し送出するシステム信号を表す。

2.5 継時的接続 :

シーンやシステム信号が継時的に出現したり生成されたりすることをロン:で表す。:の前のシーンは:の後のシーンに対して、時間的に先行することを示す。

2.6 同時的存在 ,

カンマ,の前後のシーンが、同時に存在していることを表す。なお要素スケジュールでの表記では、スケジュール要素のパラメータが複数ある場合の区切りとしても用いられる。

2.7 排他的存在 \

バックスラッシュ\の前後にあるシーンのどれか一つのシーンが存在

していることを表す.

2.8 グルーピングもしくは関数パラメータ ()

シーンなどをひとまとめにするのに用いられたり, () の前にかかれた関数のパラメータを () 内で示したりするのに用いられる. なお要素スケジュールの表記では, スケジュール要素の複数のスロットやパラメータを () の内側に書き込む.

2.9 継続条件 < >

< > の前後にある, シーン及び () でまとめられたシーンの複合体は, < > 内の条件のチェック結果に従って開始, 継続, 終了が決定される. 構成スケジュールにおいて < > はシーンではないので継続的接続: によって直接に囲まれることはない.

2.10 データ呼び出し関数

継続条件の中で用いられるデータ呼び出し関数としては以下のものを考える.

- I () あるシーンの開始からそのシーンが中断するまでの経過時間
- L () シーンが中断した際に再び戻る復帰場所
- M () イベント回数
- R () 反応回数
- T () 経過時間
- U () スケジュールパラメータ, 実験セッションの定数
- V () 値

これらの関数を用いて, 実際のところ呼び出す可能性があるものは, 下

強化スケジュール表記システムについての覚書

記のものにすぎない。したがって、これらの関数の表記に多大な文字を消費するくらいならば、シーンで保存する必要があるデータをベクトルで表し、そのベクトルの位置で各データを示した方が能率的であろう。しかし、それは今後の課題として残しておく。これらは要素スケジュールでも共通に用いられる。なお $L(S_1)$ 、 $I(S_1)$ は割り込み処理に用いられる。

シーン S_1 で用いられたスケジュール要素のパラメータ $U(S_1)$

シーン S_1 の開始から初発反応までの経過時間 $T(S_1)$

シーン S_1 の開始から一番近い強化提示 E_1 までの経過時間 $T(E_1)$

シーン S_1 でなされた反応回数 $R(S_1)$

他のシーンからのシステム信号によりシーン S_1 が中断した際に再び戻る復帰場所 $L(S_1)$

シーン S_1 が中断するまでのシーンの S_1 開始から経過時間 $I(S_1)$

シーンについての保存データのほかに、システム信号に関するものと、セッション全体に関する保存データを考える必要がある。現実的な対処としては、以下のものを考えておくことで、当面の間は対応可能であろう。

システム信号 Z を送出したときの直前のシーン $V(S)$

システム信号 Z の値（どこから出た Z かを教える値） $V(Z)$

発生させられた確率 Q の値 $V(Q)$

実験セッションの E の回数 $M(E)$

実験セッションの定数 $U(S)$

実験開始からの経過時間 $T(S)$

実験開始からの反応回数 $R(S)$

2.11 繰り返し記号 §

同じ添え字を与えられた § の間は、2 番目の § の直後にある継続条件に従って繰り返される。§₁(S₁:E:) §₁<M(E)==10> は、(S₁:E:) が E10 回分、すなわち、10 回の強化で終了することを示す。もしも <T(S)==60'> であればセッション時間を 60 分に設定している。しかし、スケジュールの骨格を示す場合には、煩雑さを避けるために、通常、省略する。なお、括弧を閉じる直前にコロンが来た場合には (:] :} :)), その括弧の開始からその位置までが繰り返されていると考える。

2.12 論理・数学記号

継続条件や関数内で用いられる論理・数学記号には、以下のものがある。これらは要素スケジュールでも共通に用いられる。

&	論理積
	論理和
~	否定
==	等しい
~=	等しくない
>=	以上
<=	以下
>	越えて
<	未満
=	代入
+	和
-	差
*	積
/	商

^ べき乗

2.13 定数

以下のギリシャ文字を定数を表記するのに用いる。これらは要素スケジュールでも共通に用いられる。

τ 時隔定数

ρ 反応回数定数

π 確率定数

α その他の定数

2.14 変数

変数をあらわすときには x と y を用いる。

3. 構成スケジュールによる表記の実際

ここでは主要な構成スケジュールの例を示すとともに、必要に応じて表記法上の注意を述べる。

3.1 レスポンデント条件付け

以下に3種類のレスポンデント条件付けの手続きを挙げる。いずれも S_1 は試行間間隔 ITI を表している。延滞条件付けでは通常条件刺激 S_2 の提示終了が無条件刺激 E の提示終了と重なっている。これをもし明示的に書くとすれば、 $(S_1 : \{S_2\}_1 : \{S_2, E\}_1 :)$ と書くことになるが、残念ながらこの表記法だと、 S_2 が連続していることを直感的には理解しにくい。要素スケジュールで対応すべきであろう。

延滞条件付け delayed conditioning $(S_1 : \{S_2 : E\} :)$

痕跡条件付け trace conditioning $(S_1:\{S_2\}:S_3:E:)$

逆行条件付け backward conditioning $(S_1:E:\{S_2\}:)$

3.2 複合スケジュール

最近では複合、複雑の区別をつけず、両者とも複合スケジュールとして論じることが多くなったが、本節と次節では、これまでの伝統（たとえば Catania, 1968）に従って2つに分けて取り上げることとする。複合（compound）スケジュールには、混成（多元）、混合、連鎖、連接（連結と呼ばれることが多いが、本稿では yoked を連結、interlocking を連動と呼ぶことにする）といった継時的な結合と、並立、共立といった同時的な結合とを区別することができる。しかし行動分析学の最近の進展によって、こうした伝統的なスケジュールには様々な変形が現れるようになった。構成スケジュールの表記法を重視した理由の1つは、こうした複合スケジュールの変形を効率よく表したいと考えたからである。

まず、以下に代表的な継時的複合スケジュールを挙げる。なお複合スケジュールには高次スケジュールが含まれることが多いが、本稿では複雑スケジュールに分類しておく。

連鎖 chained $S_1 S_2$ $(\{S_1\}_1:\{S_2\}_2:E:)$

連接 tandem $S_1 S_2$ $(S_1:S_2:E:)$

混合 mixed $S_1 S_2$ $(S_1:E:S_2:E:)$

混成 multiple $S_1 S_2$ $(\{S_1\}_1:E:\{S_2\}_2:E:)$

上で挙げた混合、混成スケジュールの例は、2つのスケジュールが交互にあらわれるような、提示順が固定した型のスケジュールである。しかし両者とも、各成分スケジュールで強化子を伴うのが一般的なもので、各成分は完全に独立していると考えてよい。以下に、提示順がランダムとなる混

強化スケジュール表記システムについての覚書

成スケジュールの表記例を示す。Qで発生された確率に従い、 S_1 と S_2 とのどちらかが現れることを示した。

提示乱動型混成 $S_1 S_2$ $((\{ \langle V(Q) \rangle S_1 : E \}_1) \setminus (\{ \langle V(Q) \rangle S_2 : E \}_2))$
:Q:

また、強化子の提示をもう一方のスケジュールへの交替の機会とするのではなく、一定期間 τ の間、ずっとその強化スケジュールが提示され、それが終わるとまたある一定期間、別の強化スケジュールが提示されて、これが繰り返される場合も考えられる。繰り返しを表現する場合には $\{S_1 : E : \}$ のようにEのあとに:を続けて、同じディスクリミナダムの中で餌提示が繰り返されていることを明示する必要がある。

期間型混成 $S_1 S_2$ $(\{S_1 : E : \}_1 \langle U(\tau) \rangle : \{S_2 : E : \}_2 \langle U(\tau) \rangle :)$

最後に時折見かけるケースは、先行するスケジュールのシーン S_1 で生み出された何らかの結果に合わせて、後続するスケジュールのシーンのパラメータが変わる場合である。たとえば先行シーンのスケジュールをVRとし、後続するシーンでは、先行シーンでの強化獲得までの経過時間($T(E_1)$)に合わせるような連結 yoking を行うケースがこれにあたる。このような複雑な連結スケジュールをこれまでに述べてきたような記法で表現することは難しいが、構成だけをプロットするならば以下のようなだろう。しかしこれと期間型混成とを組み合わせ、VR期間でのすべての強化時刻を記録して、もう一方の期間に反映させるとなると、このレベルでの表記法をこえた表現（例えば生起時刻とイベントの対よりなる、時系列データの読み出しや操作、第3節2）を参照）が必要となる。

連結型混成 yoked mult $S_1 S_2 \quad (\{S_1:E_1\}:\{S_2:T(E_1):E_2\}:)$

代表的な同時型の複合スケジュールとして、オペランドが1つである共立スケジュールと、2つである並立スケジュールを挙げる。ちなみに共立スケジュールと類似するスケジュールには論理和スケジュールがある。しかし、構成は共立スケジュールとはまるで異なっており、シーン間での関係に OR (|) という論理記号が必要となる。詳しくは複雑スケジュールにおいて述べる。

共立 conjoint $S_1 S_2 \quad [(S_1:E:), (S_2:E:)]$

並立 concurrent $S_1 S_2 \quad ([S_1:E:]_1, [S_2:E:]_2)$

さらに以下に、より発展させた手続きでの並立スケジュールの表記例を挙げる。ここで $\langle V(Z_1) \rangle$ は $\langle V(Z) == Z_1 \rangle$ の省略を表し、意味の混乱がおこらないかぎりこうした省略形を用いることとする。

並立連鎖 conc chain $S_3 S_4$

$(([S_1:Z_1]_1, [S_1:Z_2]_2) : (\langle V(Z_1) \rangle [S_3:E]_1 \setminus \langle V(Z_2) \rangle [S_4:E]_2) :)$

強化スケジュール切り替えキー型並立 changeover-key (Findley) conc $S_3 S_4$

$([S_1:Z_1:S_1:Z_2]_1, [\langle V(Z_1) \rangle \{S_3:E\}_1 \setminus \langle V(Z_2) \rangle \{S_4:E\}_2]_2)$

強制型並立 dependent scheduling (Stubbs & Pliskoff) conc $S_2 S_3$

$(S_1:Q : (\langle V(Q) \rangle S_2:E]_1, [\langle V(Q) \rangle S_3:E]_2) :)$

調整型並立連鎖 adjusting conc chain (FR1 FR1) ($S_2 S_3$)

$$(([S_1:Z_1]_1, [S_1:Z_2]_2) : ((\langle V(Z_1) \rangle [S_2:E_1]_1) \setminus (\langle V(Z_2) \rangle [S_3 \langle V(Z_1|Z_2) \rangle : E_2]_2))) :)$$

3.3 複雑スケジュール

一般に複雑 (complex) スケジュールといった時には、論理和、論理積、連動、累進スケジュールが取り上げられるケースが多いが、ここでは高次スケジュールもその仲間に入れる。他の複雑スケジュールとよく似た構成であるためである。また、複合スケジュールの例として挙げた調整型並立連鎖スケジュールの「調整型」を意味する調整スケジュール (adjusting schedule) も、複雑スケジュールの1つである。

複雑スケジュールは構成スケジュールよりも要素スケジュールとして表記したほうがよい可能性も残っている。当初は、これら複雑スケジュールを要素スケジュールのレベルで表記しようと試みていたのであるが、そのために必要なスケジュール要素の数や仕組みが大きく増えることを嫌って、この構成スケジュールの枠組みに入れることとした。

一般に、複合スケジュールは、継時的、同時的な時間構造がはっきりしており、それぞれの成分となるスケジュールがこれらの構造の中に組み込まれる形式となっている。一方複雑スケジュールでは、たしかに2つ以上の成分スケジュールが存在してはいるが、それらの間の論理的関係や数量的関係を基盤として随伴性を形成している点が形式的に大きく異なる。

論理和、論理積両スケジュールについては、2種類の Z を利用して、その論理和や論理積が真になるかをチェックすることで強化子を提示することとした。連動スケジュールについては、 S_1 に固定時隔 (FI)、 S_2 に固定比率 (FR) スケジュールを仮定して表記してみたが、前者についてはFRでの反応回数をパラメータに、後者についてはFIでの経過時間をパラメータに、それぞれ基づきながら、もともとのスケジュール設定値を変えていくというスケジュールを想定している。

累進スケジュールには累進比率スケジュールと累進時隔スケジュールが

ある。累進比率スケジュールでは、食餌に至る比率が算術級数もしくは幾何級数的に増加するが、ここでは単に設定されるシーン S_1 での反応数を次の S_1 での反応数を定める条件として新たに設定するように書き表してみた。実際には 4.3 に取りあげる関数計算 w で、何らかの演算がなされる。

高次スケジュールは、たとえば FR ρ (FI τ) と書いた場合に、FI τ を 1 つの行動単位として、それを ρ 回行うと強化に至るといったケースを言う。下の例では S_1 に FI を当て、回数を調べる関数を使って表現した。しかしすべてこのように高次スケジュールが表わせるとは限らない。現実の実験場面では、単なる接続 tandem スケジュールである場合が多いように思われるので、要素スケジュールでもある程度表記できるであろう。

論理和 alternative $S_1 S_2 ((S_1:Z_1), (S_2:Z_2)) <V(Z_1) | V(Z_2)> :E:$

論理積 conjunctive $S_1 S_2 ((S_1:Z_1), (S_2:Z_2)) <V(Z_1) \& V(Z_2)> :E:$

連動 interlocking $S_1 S_2 ((S_1) <R(S_2)>, (S_2) <T(S_1)>) :E:$

累進 progressive $S_1 ((S_1) <R(S_1)>) :E:$

高次 higher order $S_1 ((S_1) <M(S_1)>) :E:$

3.4 その他のスケジュール

強化スケジュールは無数にあるので、そのすべてについてここで表記例を挙げることはできないが、「1. はじめに」で挙げた固定長反応連スケジュールについてここで取り上げてみよう。下に示すようにオペラダム 1 でのシーンの通過を Z で表し、それを条件として、オペラダム 2 でのシーン（具体的には 1 回反応をする）を通過して餌に至ると考えた。ただ、構成スケジュールには、 S_1 での通過条件をクリアしない場合、それまでの反応数がリセットされるなどといったことは書くことができない。そうした意味からも、構成スケジュールは強化随伴性の全体的なスケッチであるといえる。なお、別の表現の仕方については、第 6 節 1) を参照さ

りたい。

固定長反応連 fixed consecutive number

([S₁:Z₁]₁, (<V(Z₁)>[S₂]₂:E:))

続いて単純見本合わせ課題を構成スケジュールで表してみた。中央のキーには2種類の異なる弁別刺激のどちらかが提示され、その後、両脇の2か所に異なる弁別刺激（ただし、ここでは見本に出された刺激のどちらを提示するかは書いていない）の提示があり、見本に対応する反応をすることによって強化子に結び付くという流れが表記されている。

単純見本合わせ課題 simple matching-to-sample task

([{S₁:Z₁ }₁ \ {S₂:Z₂ }₂]₂ : ([<V(Z₁)> {S₃ }]₁, [<V(Z₂)> {S₄ }]₃) : E :)

4. 要素スケジュールの表記法

要素スケジュールはRNSのサブシステムの1つであり、構成スケジュールにおける各シーンを、主に反応と経過時間の評価に基づいて環境変化の生成操作を行なうスケジュール要素の結合として、別の側面から再現する。要素スケジュールと構成スケジュールの関係については節を改めて論じることとするが、大まかに言えば、シーンでは個体が出会う強化随伴性が強調されるが、スケジュール要素では実験者がどのように環境を制御するのが強調される。上述したように、たとえばEは強化子が提示されているシーンであるが、要素スケジュールではこのシーンをFという強化子提示装置の提示開始と終了とで表現する。

構成スケジュールのシーンは1つ以上のスケジュール要素によって形成され、スケジュール要素はシーンをまたぐことがないだけでなく、シーン内で完結している必要がある。

4.1 スケジュール要素

要素スケジュールの本体であるスケジュール要素を考えるには、分岐を形成することができる条件を枚挙する必要がある。これまで研究に使われてきた様々な強化スケジュールから、現時点で考えられるそうした条件には、(1) 時間経過、(2) 反応回数、(3) 確率、(4) システム信号、にかかわる評価の4つがある。したがってスケジュール要素は、これらから構成できる条件の組み合わせに基づいて用意することになる。

スケジュール要素と共に与えられなくてはならないものに、3種のスロットがある。これらスロットは、

(1) スケジュールのパラメータを満たしたか、あるいは設定条件を満たしているかの判断や評価 (*eval*)

(2) 条件を満たした場合に生成する環境変化 (*envc*)

(3) スケジュール要素の必要に応じて与えることになる、条件を満足しない場合のリターン分岐先 *return*、条件を満足しないが、そのスケジュール要素のカウンタのみをリセットする場合のリセット分岐先 *reset*、条件を満足したときのジャンプ分岐先 *jump*、の3つを、デフォルトで与えられた分岐先以外で指定する場合の分岐先 (*addr*)

の3つでありスケジュール要素に引き続く () 内で表現される。 *eval*, *envc*, *addr* の各スロットは、セミコロン ; によって区分される。スロットの中での複数のパラメータは、カンマ、によって区分されるが、混乱が生じる可能性がある場合には *resp=*, *time=*, *prob=*, *reset=*, *return=*, *jump=*, といった具合に明記する。ここでイコール = は代入を表す。これらについてのより詳しい説明は、スケジュール要素の中、あるいは要素を説明したのちに行うこととする。

スケジュール要素では、特別な事情がない限り、スケジュールパラメー

タが充足されてはじめて先に進むことができる。その判断はスケジュール要素間にあるスケジュール結合子の直前で行われると想定している。特別な事情とは、スケジュール要素自体の特性、セッションの終了などのシステム信号、もしくはのちに述べるアスペクトとブラケットの発生や消滅についての信号が他からやってくる場合などである。

スケジュール要素には現在のところ、以下に述べる r , t , o , h , p , g , c , n を考える。

4.1.1 反応子 $r(\text{resp}=\rho; \text{envc}; \text{addr})$

r は反応回数 of スケジュール要素であり、設定された回数を満足すれば、次の要素に移行する。満たさなければ、この要素の先頭に戻り比較を続ける。 $r(f10; F)$ はこれのみで固定比率 10 強化スケジュールの強化子の提示までを表すことができる。() 内の *eval* の場所で用いられる英小文字 f は固定値、 v は変動値、 j は関数によって計算された値を表す。固定値とはそのあとに書かれた数字が固定して用いられることを、変動値とはそのあとの数字が平均値となる疑似ランダムに並べられた数表に従って変わっていくことを意味している。関数依存値 j は $j(x)$ のように () 内に変数を入れる。このスケジュール要素では、ある反応カウンタが想定されており、そのカウンタの内容が *eval* 内のパラメータと比較される。*eval* に続くセミコロン ; の後には、条件が満たされた後に生成される環境変化 *envc* が記入され、今の例では強化子が提示されている (F. なお F の終了 !F は別のスケジュール要素で行われる)。また、*addr* が省略されているのは、この後に引き続きスケジュール要素に移行するためである。

4.1.2 時隔子 $t(\text{time}=\tau; \text{envc}; \text{addr})$

t は経過時間のスケジュール要素であり、設定された時間を満足すれば、次の要素に移行する。満たさなければ、この要素の先頭に戻り比較を続ける

る。 $t(v10''; F)$ は、これのみで変動時間 10'' 強化スケジュールの強化子提示までを表すことができる。このスケジュール要素では時間カウンタが走らされ、*eval* 内に設定された値と比較される。 $t(f10'').r(f1; F).t(f3''; !F)$ と組み合わせることで、固定時隔 10'' スケジュールの強化子提示 3 秒の終了までを書き表すことができる。 $!F$ の $!$ は操作の終了を意味する。なお 10' とした場合は 10 分を表す。この例で示されている 3 つのスケジュール要素は、後で述べるスケジュール結合子（ピリオド $.$ ）によって結合されており、その直前でその前にある要素の条件が充足されたか否かの判断がなされていると考える。

4.1.3 時隔内無反応子 $o(\text{time}=\tau, \text{resp}=\rho; \text{envc}; \text{addr})$

eval スロットの第 2 パラメータで定められた回数の反応が起こらないで第 1 パラメータで定められた時間が経過したときに次へと移行するスケジュールで、 $o(f10'', f1; F)$ は $\text{DRO}10''$ に相当する。ここでは、時間と反応の 2 つのカウンタが走らされていることに注意すべきである。もしもこのスケジュールで 10 秒以内に反応がなされたら、デフォルトでは経過時間と反応回数のリセットされ、このスケジュール要素の先頭に戻る。この位置をデフォルトでのリセット分岐点 *reset* とよぶ。10 秒以内で、かつ反応もなされていない場合は、どのカウンタもリセットされず、計測を継続するために、この要素の先頭に戻る。これをデフォルトでのリターン分岐点 *return* とよぶ。さらに、反応がなされずに 10 秒が経過すれば、2 つのカウンタはリセットされ次に書かれた要素の先頭に行く。これをデフォルトでのジャンプ分岐点 *jump* と呼ぶ。これらデフォルト以外の分岐点を設定する場合には、第 3 スロットの *addr* で指定する。

4.1.4 時隔内反応子 $h(\text{time}=\tau, \text{resp}=\rho; \text{envc}; \text{addr})$

第 1 スロットの第 1 パラメータで定められた時間の経過中に、第 2 パラ

強化スケジュール表記システムについての覚書

メータで定められた回数の反応が起これば、その反応回数を完遂した時点で次へと移行するスケジュールで、 $h(f1'', f1; F)$ は DRH1'' に相当する。もしもこのスケジュールで 1'' 以内に 1 反応がなされなかったら、デフォルトでは経過時間と反応回数がリセットされ、このスケジュール要素の先頭（デフォルトでのリセット分岐点）に戻る。

4.1.5 確率評価子 $p(\text{prob}=\pi; \text{envc}; \text{addr})$

$p(0.7; F)$ は、デフォルトでは 70% の確率で F を開始し次の要素に移行する一方、30% の確率でこの要素 p の 1 つ前の要素に戻る。リセットはない。70% の確率で生成される環境変化は envc に書かれるが、30% の確率で生成される環境変化は飛び先の分岐で定義することとする。たとえば

$$n_0.r(f1).p(0.1;F;return=n_1).t(3'';!F;jump=n_0).n_1(;D).t(10'';!D;jump=n_0).$$

は乱動比率 10 スケジュールで餌の提示を見るが、残りの確率で要素 n_1 に行き、 D を開始し、10'' 秒後 D を終了し元にもどる。

4.1.6 Z 信号評価子 $g(V(Z)==\alpha; \text{envc}; \text{addr})$

$g(z_1;F;return=n)$ と表記した場合は、 $V(Z)$ が z_1 ならば F を開始して次の要素に、そうでなければ n にジャンプする。 addr のスロットに何も書かなければ z_1 が来るまで単に待機することになる。 g が判定するシステム信号は現在のところ Z のみである。

4.1.7 比較子 $c(\text{eval}; \text{envc}; \text{addr})$

c は比較と評価を行う要素である。比較の一方には、データ呼び出し関数が入ることがほとんどである。 p や g と異なり、比較が成立した場合には 2 つ先の、比較が成立しなかった場合には次のスケジュール要素に飛

ぶ、そうすることで、簡単な場合分け処理に対処する。gと同様、cについても生成環境変化と分岐点、ならびに reset 時のリセット対象となるカウンタの指定などをうまくまとめて書き下してしまいたいところであるが、 $\text{reset}(r_1)=n_1$ で反応子 r_1 のカウンタをリセットし n_1 に分岐するという程度の記法で対処することとする（第3節1）の固定長反応連タイプ2を参照）。

4.1.8 分岐点子 n (*envc; addr*)

ジャンプ先の分岐点を表す要素、いわゆる強化スケジュールとしての働きはない。デフォルトでは、次の要素にそのまま移行する。設けられるスロットとしては基本的には、必要があれば第2スロットで生成環境変化の指定、第3スロットで飛び先の分岐点の指定となる。ただし、随伴性の操作に関わらないで単に観察だけを行っているような場合には、この n を積極的にその用途に使うときがある。添え字をつけて、異なる分岐点を表す。

4.2 スケジュール結合子

上述したスケジュール要素を結合するものにスケジュール結合子 \cdot がある。スケジュール要素の次に来る結合子の直前で、それぞれのスケジュール要素での評価、分岐の決定、環境の操作、システム信号の送出行がられていると考える。結合子の直後から、概念上は、スケジュール要素の支配が及ぶ領域に入る。デフォルトでのリターンとリセットの2つの分岐点は、この結合子の後に想定されており、設定されているパラメータ値やそのスケジュール要素での過去の記録状態が維持されたままで繰り返しが始まる（リセットの場合は、反応と時間のカウンタはクリアされる）ので、スケジュール要素の新しい開始とは区別される。その一方で、直前のスケジュール要素からの移行やデフォルトでのジャンプ分岐点は、この結

合子を通過して、そのあとに書かれたスケジュール要素を「新たに」開始すると考えている。新しい開始では、反応カウンタや時間カウンタなどすべての関係する変数はクリアされ、それと同時に、全く新しいスケジュール要素としての操作とその記録が始まる。ただし、これらはデフォルトの場合であり、第3スロットの分岐先にそれぞれのケースが指定された場合には、その指定に従った働きがなされる。

4.3 生成環境変化

スケジュール要素の第2スロットである生成環境変化 *envc* の対象はさまざまなので、ここでまとめて述べておく。生成環境変化には、まず、個体にとって明示的な物理的・化学的実体をもつ以下の3つがある。提示の開始は以下に示す英大文字を、終了は各文字の前に！をつけることで表す。必要に応じて添え字をつける。

- D ディスクリミナダム（の提示）
- K オペランドム（の提示）
- F 強化子（の提示）

第2に、個体がアクセスできない環境としてすでに述べたシステム信号 Q と Z があり、これらもこのスロットに書き込まれるが、時間幅を持たない環境操作と見なされ、開始と終了の区別はない。それぞれはスケジュール要素 p と g によって評価される。

第3に、ここでは詳述しないが、1つ以上のスケジュール要素の結合をアスペクト A (6.1 参照) と呼んだ時、このアスペクトを生成環境変化の対象とする。すなわち他の環境変化と同様、開始と終了をスケジュール要素から求めることができる。それに加え、システム信号の評価を行うスケジュール要素を中心とした、1つ以上のスケジュール要素の結合をブラ

ケット B (6.2 参照) と名付け, アスペクトと同じように, その開始と終了をスケジュール要素からできることとする.

第 4 に変数の操作を行うことも, この *envc* スロット内の生成環境変化の 1 つとする. この操作も第 2 のものと同様, 時間幅を持っていないと考える. この操作を主に行うものが関数計算 $W()$ で $W(x=x+1)$ のように書く.

5. 要素スケジュールによる表記の実際

5.1 基本的な要素スケジュール

以下に示すのは, オペラント条件づけで用いられる基本的な強化スケジュールを要素スケジュールで表記したものである.

固定時間 $FT\tau''$	$t(f\tau'';)$.
変動時間 $VT\tau''$	$t(v\tau'';)$.
乱動時間 $RT(1/\pi)''$	$t(f1'') \cdot p(\pi;)$.
消去 EXT	n.
固定時隔 $FI\tau''$	$t(f\tau'') \cdot r(f1;)$.
変動時隔 $VI\tau''$	$t(v\tau'') \cdot r(f1;)$.
乱動時隔 $RI(1/\pi)''$	$t(f1'') \cdot p(\pi) \cdot r(f1;)$.
固定比率 $FR\rho$	$r(f\rho;)$.
変動比率 $VR\rho$	$r(v\rho;)$.
乱動比率 $RR(1/\pi)$	$r(f1) \cdot p(\pi;)$.
低反応率分化 $DRL\tau''$	$o(f\tau'', f1) \cdot r(f1;)$.
高反応率分化 $DRH\tau''$	$h(f\tau'', f1;)$.
他行動分化 $DRO\tau''$	$o(f\tau'', f1;)$.

5.2 その他の要素スケジュール

上のほかにもいくつかの複合・複雑スケジュールを要素スケジュールで表すことができるが、それでは構成スケジュールを使って全体の構成を大まかにとらえるという意味がなくなるので、ここではそれらについてはとりあげない。ここでは、それらがどう表記されるのかを別の例で示そう。

たとえば DRL 10" LH5" というスケジュールは低比率分化強化スケジュールに制限時間 limited hold が付けられたもので、この場合は、反応間時間が 10" 以上の反応で、かつ、そこから 5" 以内に反応すれば強化子が提示される。これは以下のように表現されるだろう。

```
n.o(f10", f1).h(f5", f1; F; reset=n).t(f3"; !F).
```

通常 DRO は第 1 スロットの時間カウンタには固定された値が設定される。これを変動とするスケジュールを仮に DRV と呼ぶとすれば、下のよう連鎖させることで、すなわち chain DRV DRH とすることで、反応時間を測定するのに有効なスケジュールとなろう。

```
n(; !D).o(vτ, f1; D).h(f5", f1; !D, F; return=n).
```

6. 構成スケジュールと要素スケジュールの合体

ここまでで、構成、要素両スケジュールのあらましを、具体例を使って示してきた。本節で考えるのは、両者の合体である。別の表現をすれば、両者の関係を論じることになる。

構成スケジュールは、強化スケジュールが示す強化随伴性の全体像を表現するのに適していると考えられる、様々な表記法を用いている。そこではオペランドやディスクリミナンド、それからシーンとその継続条件が中心となっている。一方、要素スケジュールは、条件比較とその評価、

生成環境変化，分岐といったスロットを持つ，いくつものスケジュール要素の結合として表現されており，分岐だけを示す要素 (n) や比較だけを役目とする要素 (c) をさらに加えることで，強化スケジュールの持つ細部の随伴性までも，柔軟に表現できるよう考案されている．この2つのスケジュールを組み合わせて，随伴性の全体像と細部とを表すには，まだいくつかの表記法上の工夫が必要である．

6.1 アスペクト A

アスペクトは，1つ以上のスケジュール要素からなる，要素の継続を表現する．それと同時に，そうしたスケジュール要素が，構成スケジュールのどの部分と対応するかということもあらわす．たとえば

$$A_2 @ (S_2 : E) @ \tau (f1 ; F) . t (f3'' ; !F, A_1, !A_2) .$$

という1行は，(1) 先頭にアスペクトとその番号，(2) 初めの@と2番目の@の間にはそのアスペクトに対応する構成スケジュールのシーン，(3) この例では2番目の@の後，空白を開けてFR1に相当する要素スケジュールが書き込まれている．そこでは，反応子 τ で反応が1回あれば餌の提示が始まり，時隔子 t で3秒後に餌の提示が終わり，それと同時に別の1つのアスペクト A_1 が生成され，(ここでは明示的に書かれているが，) 自分のアスペクトである A_2 を終了させている．なお，ここでもしも A_2 を生成すれば，再び A_2 が開始される．

アスペクトの1つの大きな特徴は，その最終のスケジュール要素が実行されると，そのアスペクト自身が自動的に消滅するという点である．したがって，上の例で $!A_2$ と明示的に書かなくても，この A_2 というアスペクトは次に呼び出されるまで活動することはない．

別の見方をすれば各アスペクトは，一連の構成スケジュール，要素スケ

ジュールのまとまりの開始地点を示しているとみることができる。したがってアスペクトの生成とは、同時進行している複数のアスペクトを監視している実行管理表に、生成対象となったアスペクトの最初のアドレスが書き込まれ、次のタイミングでそのアスペクトが実行されることであると考えることもできる。一方アスペクトの終了とは、この実行管理表から、現在処理されているアスペクト内のアドレスを取り除き、そのアスペクトに関する処理を中断することに相当する。

このアイデアは、Z PULSE というシステム信号と共に、SKED, OPEAB, XANOP など使われてきた、古くからあるタイムイベントによる割り込み処理を用いて制御された、時分割（タイムシェアリング）による同時並行処理システムでよく利用された考え方によっている。

6.2 ブラケット B

ブラケットは、アスペクトと同じ構成スケジュールや要素スケジュールから成り立っているが、基本的には、条件の評価に対応する $\langle \rangle$ （アングルブラケット）部分を取り出して、その評価とそれに基づく生成環境変化だけを実行するために用いる場合が多い。したがってこのブラケットと前述したアスペクトとをうまく使い分けることは、構成・要素両スケジュールの合体を成功させるうえで、重要な鍵となる。

このことと関連しているブラケットのもう一つの重要な特徴は、明示的に終了を表記しないかぎり、アスペクトと異なり、自動消滅しないでブラケットの先頭に自動的にもどるという点である。したがって、このブラケットにおいては条件評価をした後に、条件評価と関係のないスケジュール要素を付け加えている場合に注意が必要となる。

g や p では、*eval* スロットの評価が成立すれば *envc* スロットに書かれた変化を生成し、*addr* スロットに *jump=* で飛び先が書かれていない限り、次のスケジュール要素を実行するが、何もなければ最後の結合子まで進ん

でブラケットの先頭に戻る。一方、*eval* スロットの評価が成立しない場合は、*envc* スロットに書かれた変化は生成されず、*addr* スロットに *return=* で飛び先が書かれていない限り、*g* では自分自身に、*p* では 1 つ前の要素に、*c* では 1 つ先の要素に進む。そして最後の結合子まで進むと自動的にブラケットの先頭に戻る。

6.3 コメントと定義 # と @

以下改行までは、構成スケジュール、あるいはコメントを表す。

@ は 6.1 で述べたように、左に表記されたものを右の表記のように名づけることを意味する。ただし、 $a@b$ のように、@ が 1 つであれば a と b は完全に同等であるが、 $a@b@c$ のように 2 つの @ に挟まれた b は、 a や c に対応するが交換可能ではない。一方、この時、 a と c は同等である。別の言い方をすれば、@ と @ の間にあるのは、ある種のコメントに相当し、アスペクトで出てきた例でいえば、単に、要素スケジュールと構成スケジュールとの対応を表しているに過ぎない。

6.4 構成スケジュールと要素スケジュールによる強化スケジュールの表記法

以下に構成スケジュールと要素スケジュールの合体例を示す。いずれも、表記が大変難しいスケジュールであるが、どのように構成スケジュールが分解され、それがアスペクトとブラケットという 2 つのスケジュール要素の結合にまとめられ、それらによってそれら自身が生成、消滅するのかという仕組みを見ていただきたい。

なお調整型並立連鎖スケジュールの例では、調整対象となっている時間パラメータの計算を、スケジュール要素 *c* の中で行っている。

強化スケジュール表記システムについての覚書

1) 並立連鎖

$\# ([S_1:Z_1]_1, [S_1:Z_2]_2) : (\langle V(Z_1) \rangle [S_3:E]_1 \setminus \langle V(Z_2) \rangle [S_4:E]_2) :$
 $A_0 @ n (; A_1, A_2) .$
 $A_1 @ [S_1:Z_1] @ t(v\tau_1) . r(f1; Z_1, !A_2, B_1) .$
 $A_2 @ [S_1:Z_2] @ t(v\tau_1) . r(f1; Z_2, !A_1, B_2) .$
 $B_1 @ \langle V(Z_1) \rangle @ g(Z_1; A_3, K_1, D_3, !B_1) .$
 $B_2 @ \langle V(Z_2) \rangle @ g(Z_2; A_4, K_2, D_4, !B_2) .$
 $A_3 @ [S_3:E:] @ r(v\rho; !D_3, !K_1, F) . t(f3''; !F, A_1, A_2) .$
 $A_4 @ [S_4:E:] @ r(f\rho; !D_4, !K_2, F) . t(f3''; !F, A_1, A_2) .$

2) 強化スケジュール切り替えキー型並立

$\# ([S_1:Z_1:S_1:Z_2:]_1, [\langle V(Z_1) \rangle \{S_3:E\}_1 \setminus \langle V(Z_2) \rangle \{S_4:E\}_2]) :$
 $A_0 @ n (; A_1, Z_1, K_2, D_2, B_1, B_2) .$
 $A_1 @ [S_1:Z_1:S_1:Z_2:] @ r(f1; Z_1) . r(f1; Z_2; A_1) .$
 $B_1 @ \langle V(Z_1) \rangle @ g(Z_1; A_3, !A_4, K_1, D_3, !B_1) .$
 $B_2 @ \langle V(Z_2) \rangle @ g(Z_2; A_4, !A_3, K_1, D_4, !B_2) .$
 $A_3 @ (S_3: E:) @ r(v\rho; F) . t(f3''; !F, A_3) .$
 $A_4 @ (S_4: E:) @ r(f\rho; F) . t(f3''; !F, A_4) .$

3) 強制型並立

$\# (S_1:Q: (\langle V(Q) \rangle S_2:E]_1, [\langle V(Q) \rangle S_3:E]_2) :$
 $A_0 @ n_1 (; A_1, K_2, K_3) .$
 $A_1 @ (S_1:Q:) @ t(v\tau; Q, B_1) .$
 $B_1 @ \langle V(Q) \rangle @ p(\pi; A_2; !B_1 \text{ return}=n_2) .$
 $A_2 @ (S_2:E) @ r(f1; F) . t(f3''; !F, A_1) .$
 $A_3 @ (S_3:E) @ n_2 . r(f1; F) . t(f3''; !F, A_1) .$

4) 調整型並立連鎖

$$\#([S_1:Z_1]_1, [S_1:Z_2]_2) : ((\langle V(Z_1) \rangle [S_2:E_1]_1) \setminus (\langle V(Z_2) \rangle [S_3 \langle V(Z_1|Z_2) \rangle : E_2]_2)) :)$$

$$A_0 @ n_1 (; A_1, A_2, K_1, K_2, B_1, B_2) .$$

$$A_1 @ [S_1:Z_1] @ r(f1; Z_1, !A_2) .$$

$$A_2 @ [S_1:Z_2] @ r(f1; Z_2, !A_1) .$$

$$B_1 @ \langle V(Z_1) \rangle @ g(Z_1; A_3) .$$

$$B_2 @ \langle V(Z_2) \rangle @ g(Z_2; A_4) .$$

$$A_3 @ [S_2:E_1] @ t(f\tau; F_1) . t(f3''; !F_1, A_1) .$$

$$A_4 @ [S_3 \langle V(Z_1|Z_2) \rangle E_2] @ c(V(Z) == Z_1; W(x=x-1)) . c(V(Z) == Z_2; W(x=x+1)) . n_2 . t(j(x''); F_2) . t(f3''; !F_2, A_2) .$$

3. おわりに

この小論を締めくくるにあたって、現在までに分かっているいくつかの問題点を挙げておきたい。

1) 1つの表記に収斂するか

ある強化スケジュールを構成スケジュールや要素スケジュールで表現し、それを合体して強化随伴性についての理解を深めようとするときに、そのスケジュールがたった1つの表記に収斂するかという問題は、RNSの無駄のなさ(冗長性の少なさ)を評価するうえで重要である。しかし、それと同時に、現実的な問題、つまりRNSが求める表記システムがどの程度の冗長性を持つと強化スケジュールの表記に便利で、研究を進めるうえでも役に立つものとなるのか、という問題ともかかわっている。

たとえば固定長反応連スケジュールは、少なくとも以下のように3種類書くことができる。タイプ1, 2ではオペランド1での反応回数 ρ を満足すると別のアスペクト A_2 が動き出す。ただ、タイプ1では ρ を満足したかをブラケットによって判定し、別のアスペクト A_2 はその判定で動き

出す。一方、タイプ2ではブラケットを用意せず、 ρ が満足されると別のアスペクト A_2 が走り出し、それが判定に回されている。

タイプ3はタイプ2とよく似ているが、その構想はかなり異なっている。つまり、タイプ2の A_1 の代わりに B_1 をもうけ、反応は観察されるままで評価対象となる Z を産出しない。 B_1 はそのまま維持される一方、 B_2 も同時に走らせられ、 B_1 での反応回数を見て判定しているという構成である。このような構成スケジュールにしておくと、 B_1 での反応回数によって B_2 で与える強化確率を変えるような要素スケジュールも比較的簡単に書くことができる。

(1) 固定長反応連タイプ1

```
#([S1: Z:]1, (<V(Z)>[S2]2: E:))
A0@ n0(; A1, K1, K2).
A1@[S1: Z:]1@ r1(fρ; Z1, B1).
B1@<V(Z)>@ g(Z1; A2, !B1).
A2@([S2]2: E:)@ r2(f1; F).t(fτ; !F, A1).
```

(2) 固定長反応連タイプ2

```
#([S1: Z:]1, ([S2]2<V(Z)>: E:))
A0@ n0(; A1, K1, K2).
A1@[S1: Z:]@ n1.r1(fρ; Z1, A2).
A2@[S2]2<V(Z)>: E:)@ r2(f1;).c(V(Z)==Z1; F;
reset(r1)=n1).n2.t(fτ; !F, A1).
```

(3) 固定長反応連タイプ3

```
#([S1:]1, ([S2]2<R(S1)>: E:))
A0@ n0(; B1, B2, K1, K2).
```

$B_1 @ [S_1 :] @ n_1 .$
 $B_2 @ ([S_2]_2 < R(S_1) > E :) @ r_2 (f1) . c (R(B_1) > = \rho ; F) . n_2 (; B_1 , B_2) . t$
 $(f \tau ; ! F , B_1 , B_2) .$

これらから観察できるのは、表記法内部の固有な問題として表現の冗長性があるだけでなく、強化スケジュール、あるいは強化随伴性についての考え方の違いも冗長性を生みだしているということである。後者のタイプの冗長性は、新しいスケジュールや随伴性についての発想をもたらす可能性があり、表記法の限界から、これを抑制することは望ましくない。これまでの人工言語の歴史が示すように、暫定的な表記法に基づくある程度の実践を通して、最終的な表記法が決定されるのかもしれない。

2) システムは何を記憶していなければならないか

「はじめに」でも述べたように、無限の種類強化スケジュールを精密に表すには、それに対応できる一般的なプログラム言語が望ましい。しかし、それを見たところで、随伴性についての直観を得るのが難しいがゆえに、「適度」に自由度の高い表記法によってもたらされるスケジュールの構成なり要素なりの表現が必要であった。

同様に「適度」さが求められるものには、表記法で必要となる、システムが提供するデータ内容とその範囲がある。極端な例を挙げれば、その日の実験セッションで用いられる、スケジュール要素のパラメータの変動値のリストを、前日の実験セッションで得られたすべての強化子提示までの経過時間で構成するような場合である。この場合、データ呼び出し関数である $T(E_1)$ (シーン S_1 の開始から一番近い強化提示 E_1 までの経過時間) を何らかの形で保存をしておかなくてはならないが、すでに述べたように、こうした広い範囲のデータを保存したりカバーしたりする道具を、ここで考えてきた RNS は持っていない。しかし現在の強化スケジュールで

は、同時的もしくは継時的に走る、別の構成スケジュールの中のデータ呼び出し関数で取り扱える行動データを本体の構成スケジュールを規定しているパラメータに反映させるようなことは、頻繁に行われている。

この問題は、どのようなデータを研究者が科学共同体に対して残しておくのかという問題とも関係している。最近、動物を用いた行動実験のデータが得られにくくなっていることから、行動データに関わるデータ・コンソーシアムを作ってデータを共有しようとする試みがある。こうした場面で共有されるデータは、基本的には、時間とイベントが一組となった時系列データとなり、イベントには、スケジュール要素で生成した環境変化と、常に観察されている（随伴性と直接関係を持たない行動も含めた）研究対象の行動が記載されることになろう。現在のハードウェアの進歩とともに、こうしたデータの記録量にはほとんど制約はなく、あるとすれば記録されたデータを利用した何らかの指標（例えば、現在から100個前までの反応間時間の平均を次のVI値に設定するなど）を作成するうえでの処理時間の大きさだけが問題となってくる。

これらのデータの中のどのような側面が、現在曝されている強化随伴性と意味ある関係を結んでいるのかについては、現在も研究が進んでいるので簡単に決めることはできないから、データの利用の側面を制限していくことはよい方法とは思われない。こうして、おそらく、時系列データの利用についての、効率の良い表記法もまた必要となると思われる^{注2)}。

注2) もしも時間とイベントが一組となった時系列データを、常にRNSが参照できる程度にまで計算速度が十分速くなったとしたならば、スケジュール要素 r は反応回数ではなく、実験者の指定したイベント一般の回数を取り扱う要素となるかもしれない。同様にスケジュール要素 t も任意の2つのイベント間の時隔（時間間隔）を取り扱う要素となるだろう。要素 o や h もこれに倣うことになる。そして、この考え方を拡張すれば、イベント生起とイベント生起時隔を評価対象とし、それらの論理的組み合わせによって得られた「式」をある設定値と比較するというただ1つのスケジュール要素を考えればよいということになるかもしれない。しかし、おそらくそれは、表記法の効率化には役に立っても、随伴性の創造と理解のためにはあまり役に立たないだろう。

しかしそのような表記法の改善までは、旧態依然としたデータ呼び出し関数で私たちは満足する必要がある。

3) 表記法の使いやすさはどうか

人工言語との比較をするまでもなく、現在でも教育の中で使われている SNS は、おそらく極めて使いやすい表記法の一つであり、それゆえここまで残ってきたのであろう。しかし、すでに述べたように、いくつかの欠点があり、その改善を目指した本稿のような試みがある。

ちょうど構成スケジュールと要素スケジュールをハイブリッドにしたような表記法も考えることができる。しかしながら、前者が個体が直面する状態を中心に書かれているのに対して、後者は研究者の操作を中心として書いているので、完全に融合するという事は難しい。たとえば、並立連鎖、強化スケジュール切り替えキー型並立、強制型並立、調整型並立連鎖を次のように書き表してみよう。上の行は元の構成スケジュール、下の方は要素スケジュールとのハイブリッドである。

並立連鎖 $VR\rho$ $FR\rho$

$$\#([S_1:Z_1]_1, [S_1:Z_2]_2) : (\langle V(Z_1) \rangle [S_3:E]_1 \setminus \langle V(Z_2) \rangle [S_4:E]_2) : \\ (([VI\tau'':Z_1]_1, [VI\tau'':Z_2]_2) : (?Z_1:[VR\rho:E]_1 \setminus ?Z_2:[FR\rho:E]_2) :)$$

強化スケジュール切り替えキー型並立 $VR\rho$ $FR\rho$

$$\#([S_1:Z_1:S_1:Z_2:]_1, [\{\langle V(Z_1) \rangle [S_3:E]_1 \setminus \langle V(Z_2) \rangle [S_4:E]_2\}]_2) \\ ([FR1:Z_1:FR1:Z_2:]_1, [(?Z_1:\{VR\rho:E\}_1) \setminus (\langle V(Z_2) \rangle (FR\rho:E)_2)]_2)$$

強制型並立 $RI(\tau/\pi)''$ $RI(\tau/(1-\pi))''$

$$\#(S_1:Q : (\langle V(Q) \rangle S_2:E]_1, [\langle V(Q) \rangle S_3:E]_2) :) \\ (VT\tau'':Q : ([?\pi:FR1:E]_1, [?(1-\pi):FR1:E]_2) :)$$

強化スケジュール表記システムについての覚書

調整型並立連鎖 FTτ" FTj(x")

$$\#((([S_1:Z_1]_1, [S_1:Z_2]_2) : ((\langle V(Z_1) \rangle [S_2:E_1]_1) \setminus (\langle V(Z_2) \rangle [S_3 \langle V(Z_1 | Z_2) \rangle : E_2]_2)) :))$$

$$(([FR1:Z_1]_1, [FR1:Z_2]_2) : ((?Z_1:W(x=x-1) : [FT\tau":E_1:]) \setminus$$

$$(?Z_2:W(x=x+1) : [FTj(x") : E_2]_2)) :))$$

ここでは新しい表記として評価?を導入し、<>で行われていた判断を表現した。また強化スケジュールのパラメータ変更の実行も関数計算W()を使うことで、ある種のシーンとして取り扱ってみた。この程度まで書き下せれば、かなり複雑といわれるスケジュールについても、ある程度の見通しがつくのではないかと思われる。今後は、考えつく様々なスケジュールを、構成、要素の各スケジュールとそのハイブリッドで書き表して、RNSの実効性を確認するとともに、これを発展させていくことが課題となる。

すでにこれまで述べたところにも、表記法の使いにくさを生み出している、現段階でのRNSの持つ表記の弱さや力不足がある。1つは、スケジュール要素の3つの分岐点の指定に関わるものである。たとえばスケジュール要素では、条件が成立した場合には、その中で生成環境変化と分

注3 ここで取り上げてきた3つの問題は、スケジュール要素が持つ3つのスロットの考え方を拡張するだけでも、そのいくつかを取り扱うことができる。たとえば、すでに認めている *envc* での変数のセットを利用すれば、リセットを *envc* の一つとして記述できる。また、注2で取り上げた時系列データを *eval* の中で評価できれば、多様な反応系列や刺激布置をある程度まで処理できるだろう。たとえばある要素のスロットを次の4スロットで表現するとする。

(*eval*; *envc*%jump; *envc*%return; *envc*%reset)

eval には対象とするイベント番号と設定値、それ以降の各スロットには、生成環境変化と成立、不成立、リセットの各飛び先(%の後に記述)による3つの組を使い、環境操作と処理の飛び先を考えることができる。

岐先が自由に書ける半面、成立しない場合やりセットの場合には、取り扱いがやや面倒になる。

2つ目の問題は、スケジュール要素内の生成環境変化では、現在のところ4種類（K, D, Eという実体のある環境，システム信号Z, Q, アスペクトとブラケット，変数の操作）をその対象としているが、この自由度が高すぎないかという問題である。6.4では、調整型並立連鎖の例において、一方（調整側）のスケジュールの値が個体の選択肢によって増減することを表記してみた。また別のところでは、アスペクトとブラケットというスケジュール要素のいくつかの結合の生成と消滅も例示してみた。実際、こうした自由な表記は、例えばSKEDではかなり認められている。自由度の高さは、表記の曖昧性に結びつく一方で、表記の容易さとも関連しているので、今後も慎重に考える必要がある。

第3のまだ解決策が考えられていない問題は、弁別刺激となる反応系列や刺激配置にかかわる問題である。この問題は、強化スケジュール研究がこれまでほとんど過去の随伴性の結果や現在の刺激性制御に積極的にかかわってこなかったという経緯から、研究のスコープの外においてきた問題でもある。ある種の系列学習、たとえば二重交替学習課題は、過去にどのような選択をしてきたかに基づいて強化子が提示される。また、ある種の条件性弁別では、複数の弁別刺激の相互関係によって、正刺激となるディスプレイミナナムや反応対象となるオペランダムが指定される。たとえば二重交替学習課題は過去の試行にどのような反応をしてきたかが強化提示の基礎となるので、過去の複数試行（例えば4試行分）で取りうるパターンの各々を異なる1つ1つの反応（例えば16種類の反応）とみなすことで、強化条件を書いていくことができるかもしれないが、少なくとも、現在の表記法がそれに適しているとは考えにくい^{注3)}。

こうして、ここで展開したRNSは、実験的行動分析でこれまでに研究されたさまざまな強化スケジュールを完全に自由に表記するには至ってい

ないといえる。しかし RNS を、サブシステムとしての構成スケジュールと要素スケジュールに分けたように、上で論じた問題も、これとはおそらく異なるタイプのサブシステムを考えることで、表記上の見やすさと正確さとの間でのバランスの取れた解決を見出すかもしれない。ここで述べた暫定的な RNS が、そうした、より完成度の高い RNS に仕上がっていくための踏み台の 1 つになってくれることを期待する。

References

- Catania, A. C. (1968). Glossary. In A. C. Catania (Ed.), *Contemporary research in operant behavior* (pp. 327-349). Glenview, IL: Scott Foresman.
- Elsner, J. (1982). PASTOR: A new schedule programming language. *Behavior Research Methods & Instrumentation*, 14, 254-263.
- Findley, J. D. (1962). An experimental outline for building and exploring multi-operant behavior repertoires. *Journal of the Experimental Analysis of Behavior*, 5, 113-166.
- Hori, K., Tanaka, J., & Nomura, M. (1993). SN-FORTH/MSX: A state notation system for controlling behavioral experiments with a low-priced computer. *Journal of the Physiological Society of Japan*, 55, 47-53. (パーソナルコンピュータを用いた行動実験制御: 状態表記型制御システム SN-FORTH/MSX. 日本生理学雑誌, 55 (2), 47-53 1993 年.)
- 久保田新・坂上貴之・堀耕司・田中毅 (1981). マイクロコンピュータによる行動実験用オンラインシステム—1. リアルタイムマルチジョブモニター. 昭和 56 年 9 月日本心理学会第 45 回大会発表論文集.
- Mechner, F. (1959). A notation system for description of behavioral processes. *Journal of the Experimental Analysis of Behavior*, 2, 133-150.
- Michael, J., & Shafer, E. (1995). State notation for teaching about behavioral procedures. *The Behavior Analyst*, 18, 123-140.
- Skinner, B. F. (1958). Diagramming schedules of reinforcement. *Journal of the Experimental Analysis of Behavior*, 1, 67-68.
- Snapper, A. G., Kadden, R. M., & Inglis, G. B. (1982). State notation of behavioral procedures. *Behavior Research Methods and Instrumentation*, 14, 329-342.