Volume 9 Number 1

ACTA CYBERNETICA

Szeged, 1989

Information for authors: Acta Cybernetica publishes only original papers in English in the field of computer sciences. Review papers are accepted only exceptionally. Manuscripts should be sent in triplicate to one of the Editors. The manuscript must be typed double-spaced on one side of the paper only. For the form of references, see one of the articles previously published in the journal. A list of special symbols should be supplied by the authors.

О строении клона бурле на трехэлементном множестве*

Я. Деметрович, И. А. Мальцев

MTA SZTAKI
BUDAPEST
VICTOR HUGO U. 18—22.

АКАДЕМИЯ НАУК СССР
СИБИРСКОЕ ОТДЕЛЕНИЕ
ИНСТИТУТ МАТЕМАТИКИ
НОВОСИБИРСК

Введение

Пусть A — конечное множество и f—n-арная функция, определенная на A, значения которой также принадлежат A. Функция f удовлетворяет термальному условию, если для любого i, $1 \leqq i \leqq n$, и для любых $x, y, a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n$ из A из

$$f(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_n) = f(b_1, \ldots, b_{i-1}, x, b_{i+1}, \ldots, b_n)$$

следует

$$f(a_1, \ldots, a_{i-1}, y, a_{i+1}, \ldots, a_n) = f(b_1, \ldots, b_{i-1}, y, b_{i+1}, \ldots, b_n).$$

Функции, удовлетворяющие термальному условию, будем называть ТУ-функциями.

Термальное условие впервые было введено независимо Маккензи и Лэмпе [8]. Оно является прямым обобщением некоторых очевидных свойств одноместных функций и свойств линейных операций в векторных пространствах. Различные примеры использования термального условия при построении решеток с заданными свойствами содержит статья Тэйлора [12].

Обозначим через O_A множество всех функций f: $A^n \to A$, зависящих от конечного числа переменных. Предитеративной алгеброй Поста на множестве A называется алгебра $\mathscr{F}_A^* = \langle O_A; \zeta, \tau, \Delta, * \rangle$ типа (1, 1, 1, 2) со следующим об-

* Работа поддерживалась Национальными Научноисследовательским Фондом № 1066.

разом определенными операциями:

$$(\zeta f)(x_1, x_2, \dots, x_n) = f(x_2, x_3, \dots, x_n, x_1),$$

$$(\tau f)(x_1, x_2, \dots, x_n) = f(x_2, x_1, x_3, \dots, x_n),$$

$$(\Delta f)(x_1, x_2, \dots, x_{n-1}) = f(x_1, x_1, x_2, \dots, x_{n-1}),$$

$$(f*g)(x_1, x_2, \dots, x_{n+m-1}) = f\big(g(x_1, \dots, x_m), x_{m+1}, \dots, x_{n+m-1}\big).$$

Если f — унарная функция, то $\zeta f = \tau f = \Delta f = f$. (А. И. Мальцев [9]). Клоном называется подалгебра алгебры $\mathscr{F}_A^*$, содержащая все проекции $e_i^n(x_1, \dots, x_n) = x_i$.

Клон, состоящий из ТУ-функций, называется ТУ-клоном. Каждый ТУ-клон на множестве A содержится в некотором максимальном ТУ-клоне [3]. При $2 < |A| < \omega$ по крайней мере один ТУ-клон (клон Бурле) имеет счетное число подклонов [11]. На двухэлементном множестве существует только один максимальный ТУ-клон, содержащий 11 подклонов. На трехэлементном множестве иммется два максимальных ТУ-клона: клон Бурле и клон линейных функций (их точные определения приведены в следующем параграфе). Клон линейных функций содержит 22 подклона (см. рисунок 1), клон Бурле имеет счетное число подклонов. Строение нижней части решетки его подклонов изучено Я. Деметровичем и И. А. Мальцевым [5, 6]. Найдены все подклоны клона Бурле, состоящие из функций, принимающих не более двух фуксированных значений. Решетка этих подклонов изображена на рис. 2. Этот рисунок показывает, что решетка подклонов клона Бурле значительно сложнее известной решетки Поста всех подалгебр предитеративной алгебры $\mathscr{F}_A^*$, $|A| = 2$.

Цель данной статьи — продолжить описание строения решетки подклонов клона Бурле на трехэлементном множестве. По-видимому, эта решетка является наиболее сложным (по строению, а не по мощности) из известных фрагментов решетки подалгебр алгебры $\mathscr{F}_A^*$, $|A| = 3$.

На четырехэлементном множестве имеется уже 25 максимальных ТУ-клонов [3]. Один из них — клон Бурле, имеет счетное число подклонов. Есть ли среди остальных ТУ-клоны с бесконечным числом подклонов, авторам не известно.

1. Подклоны клона Z_0

Говорят, что определенная на A функция $f(x_1, \dots, x_n)$ существенно зависит от i-той переменной, если существуют такие элементы $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n$, $b, c \in A$, что

$$f(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n) \neq f(a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_n).$$

В этом случае i-тая переменная называется существенной. Переменные, не являющиеся существенными, называются фиктивными. Функция f называется существенно одноместной, если у нее имеется только одна существенная переменная, в противном случае f называется существенно многоместной.

В дальнейшем мы везде полагаем $A=\{0, 1, 2\}$. Как уже сказано, на A существуют два максимальных ТУ-клона: L и B. Клон L состоит из линейных функций, то есть функций, представимых в виде

$$a_0+a_1x_1+a_2x_2+\ldots+a_nx_n,$$

где сложение и умножение производятся по модулю 3. Решетка подклонов клона L (см. рисунок 1) конечная и была описана в работах Бадьинского и Деметровича [1, 2]. Подробное описание ее элементов можно найти также в [5].

Другой максимальный ТУ-клон, B, состоит из существенно одноместных функций и функций, представимых в виде

$$f_0(f_1(x_1)+\ldots+f_n(x_n)), \tag{1.1}$$

где f_0: $\{0, 1\}\to A$, $f_1, \ldots, f_n$: $A\to\{0, 1\}$ и сложение производится по модулю 2. Этот клон впервые упомянут в работе Бурле [4]. В [6] *Я.* Деметровичем и *И. А.* Мальцевым описаны все подклоны клона B, состоящие из функций вида (1.1), у которых f_0 может принимать только значения 0 и 1. Образуемая этими подклонами решетка изображена на рисунке 2. Поскольку эта решетка играет важную роль в наших дальнейших построениях, приводим также рисунки 3 и 4, позволяющие лучше уяснить взаимное расположение подклонов и их обозначения.

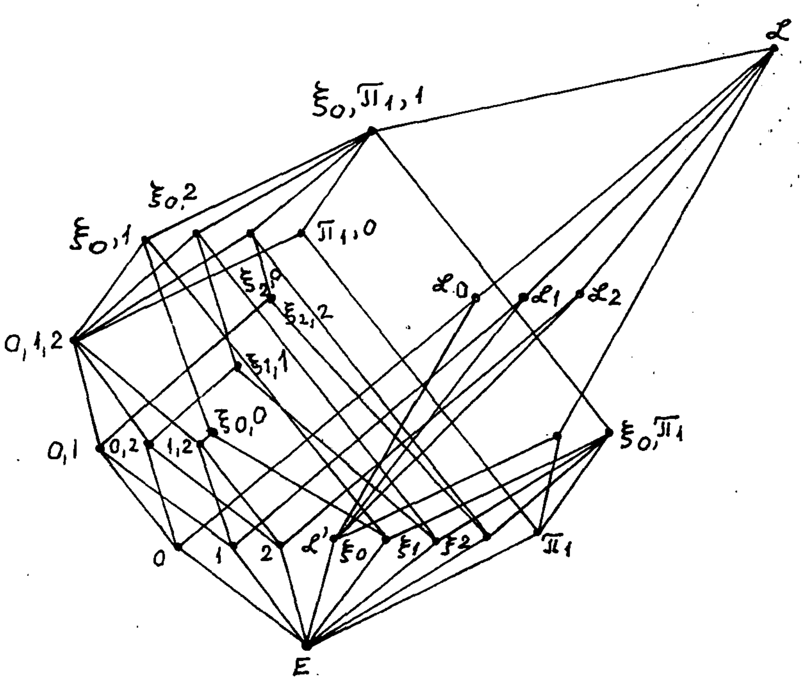
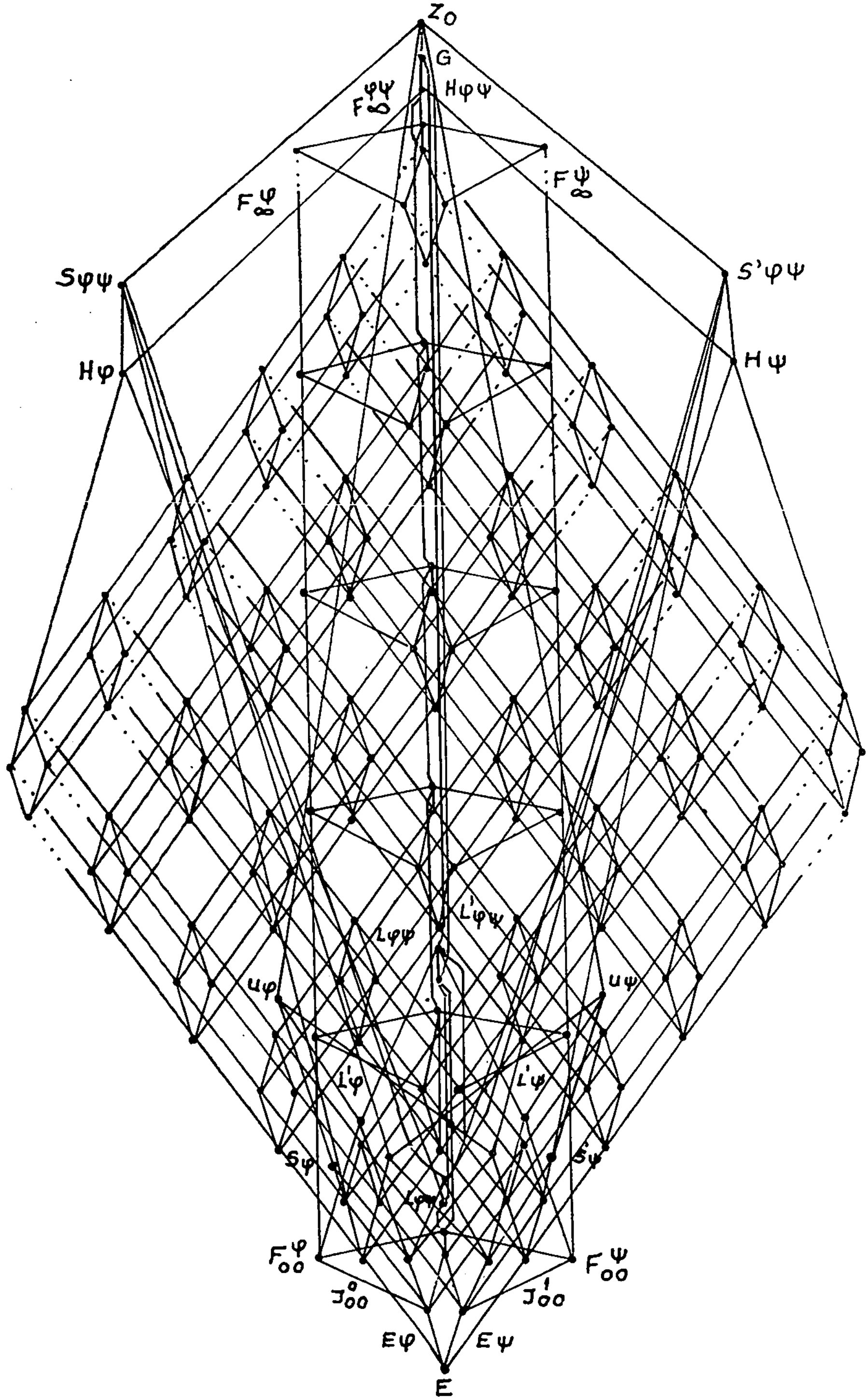


Рис. 1



Zo
G
$F_{\infty}^{\varphi\psi}$
Hφψ
F_{∞}^{φ}
F_{∞}^{ψ}
Sφψ
S'φψ
Hφ
Hψ
L'φψ
Lφψ
uφ
uψ
L'φ
L'ψ
Sφ
Sψ
F_{00}^{φ}
F_{00}^{ψ}
J_{00}^{0}
J_{00}^{1}
Eφ
Eψ
E

Рис. 2

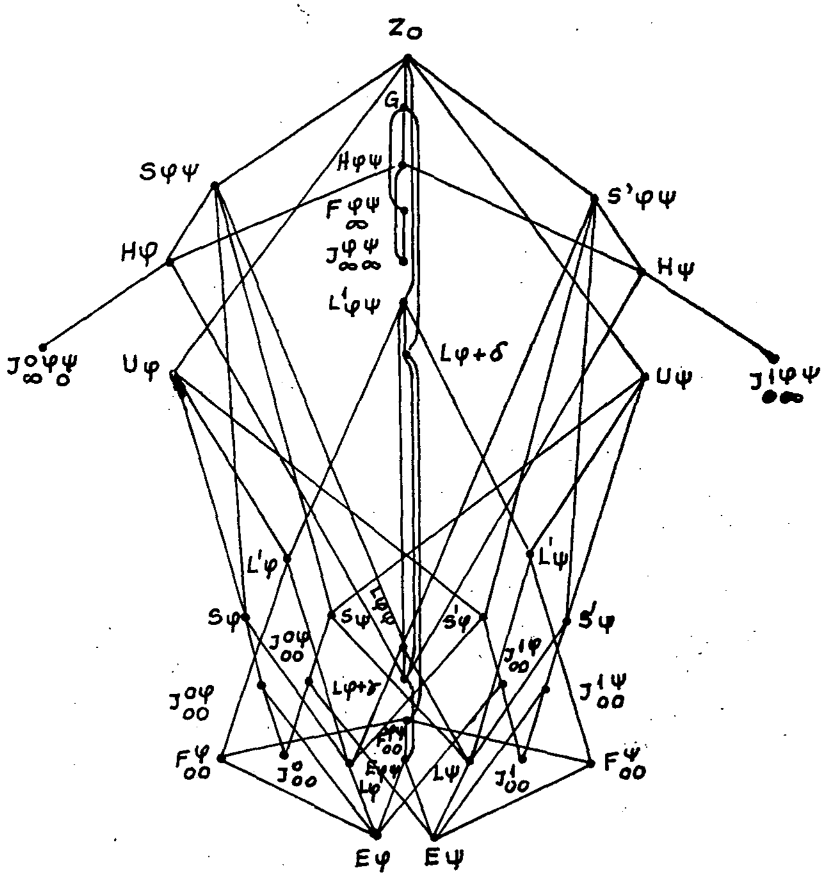


Рис. 3

В дальнейшем везде знак + означает сложение по модулю 2. Первые восемь функций, указанных в таблице 1, позволяют используя разложение (1.1) задать любую функцию из клона Z_0, образованного теми функциями клона B, значения которых попадают в множество $\{0, 1\}$.

Таблица 1

x	φ_0	ψ_0	γ_0	δ_0	$\bar{\varphi}_0$	$\bar{\psi}_0$	c_0	c_1	λ_2
0	0	0	0	1	1	1	0	1	1
1	1	1	0	1	0	0	0	1	0
2	0	1	1	0	1	0	0	1	2

Более того, так как $\delta_0(x)=1+\gamma_0(x)$, $\bar{\varphi}_0(x)=1+\varphi_0(x)$ и $\bar{\psi}_0(x)=1+\psi_0(x)$, то любая отличная от проекции функция из Z_0 может быть единственным образом

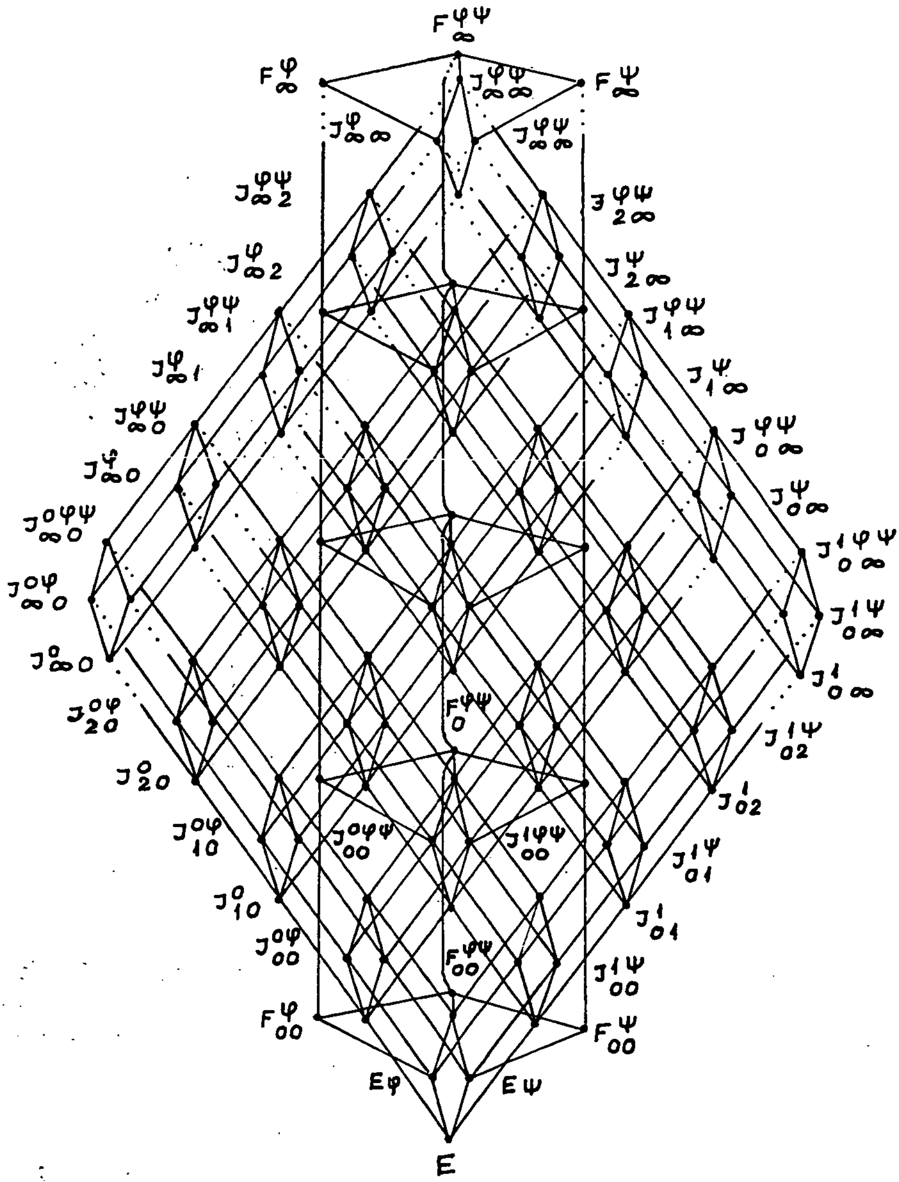

$F^{\varphi\psi}_{\infty}$
F^{φ}_{∞}
$J^{\varphi\psi}_{\infty\infty}$
F^{ψ}_{∞}
$J^{\varphi}_{\infty\infty}$
$J^{\varphi\psi}_{\infty\infty}$
$J^{\varphi\psi}_{\infty 2}$
$J^{\varphi}_{\infty 2}$
$J^{\varphi\psi}_{\infty 1}$
$J^{\varphi}_{\infty 1}$
$J^{\varphi\psi}_{\infty 0}$
$J^{\varphi}_{\infty 0}$
$J^{0\varphi\psi}_{\infty 0}$
$J^{0\varphi}_{\infty 0}$
$J^{0}_{\infty 0}$
$J^{0\varphi}_{20}$
J^{0}_{20}
$J^{0\varphi}_{10}$
J^{0}_{10}
$J^{0\varphi}_{00}$
F^{φ}_{00}
$J^{\varphi\psi}_{2\infty}$
$J^{\psi}_{2\infty}$
$J^{\varphi\psi}_{1\infty}$
$J^{\psi}_{1\infty}$
$J^{\varphi\psi}_{0\infty}$
$J^{\psi}_{0\infty}$
$J^{1\varphi\psi}_{0\infty}$
$J^{1\psi}_{0\infty}$
$J^{1}_{0\infty}$
$J^{1\psi}_{02}$
J^{1}_{02}
$J^{1\psi}_{01}$
J^{1}_{01}
$J^{1\psi}_{00}$
F^{ψ}_{00}
$F^{\varphi\psi}_{0}$
$J^{0\varphi\psi}_{00}$
$J^{1\varphi\psi}_{00}$
$F^{\varphi\psi}_{00}$
$E\varphi$
$E\psi$
E

Рис. 4

записана в виде

$$h(x_1, \ldots, x_n) = c + \sum_{j=1}^{g} \gamma_0(x_{i_j}) + \sum_{j=g+1}^{g+f} \varphi_0(x_{i_j}) + \sum_{j=g+f+1}^{g+f+p} \psi_0(x_{i_j}),$$

где $c \in \{0, 1\}$, $g, f, p \geqq 0$, а индексы $1 \leqq i_1, \ldots, i_{g+f+p} \leqq n$ попарно различны. Как правило нас не интересует порядок слагаемых и функция h будет кратко записываться как $c+g\gamma_0+f\varphi_0+p\psi_0$. Если число слагаемых невелико, то, например, вместо $\varphi_0(x_1)+\gamma_0(x_2)$, $1+\psi_0(x_1)+\psi_0(x_2)$ будем писать $\varphi_0+\gamma_0$, $1+\psi_0+\psi_0$ и т. п. В подобных сокращенных записях будут использоваться и другие одноместные функции, которые будут введены позднее. Введем также специальные обозначения для некоторых многоместных функций:

$$f_i^n(x_1, \ldots, x_n) = \gamma_i(x_1) + \ldots + \gamma_i(x_n),$$

$$u_i^n(x_1, \ldots, x_n) = \varphi_i(x_1) + \ldots + \varphi_i(x_n),$$

$$P_i^n(x_1, \ldots, x_n) = \psi_i(x_1) + \ldots + \psi_i(x_n).$$

Выражения вида $f_i^n + u_i^m + \ldots$ будет сокращением для

$$f_i^n(x_1, \ldots, x_n) + u_i^m(x_{n+1}, \ldots, x_{n+m}) + \ldots$$

По определению полагаем $\sum_{i=1}^{0} a_i = 0$.

Очевидно, запись $c+g\gamma_0+f\varphi_0+p\psi_0$ относится не к одной функции, а к целому семейству функций со следующим общим свойством: для любых двух функций этого семейства можно перейти от одной функции к другой путем введения или опускания фиктивных переменных и изменения порядка слагаемых. Кажный подклон клона B представляет собой объединение подобных множеств, пополненное проекциями.

Пусть $\mu \in O_A$ — одноместная функция, принимающая три значения, $f \in B$. Если

$$f^\alpha(x_1, \ldots, x_n) = \mu(f(\mu^{-1}(x_1), \ldots, \mu^{-1}(x_n))),$$

то α: $f \to f^\alpha$ —автоморфизм клона B. Функцию f^α мы будем называть μ-двойственной к f. Если K — подклон клона B и $\mu K = \{f^\alpha | f \in K\}$, то клон μK будем называть μ-двойственным клону K. При $\mu K = K$ клон K назовем μ-самодвойственным.

В таблице 2 (из работы [6]) для каждого подклона клона Z_0 указан пример базиса, λ_2-двойственный клон, а также указаны все функции, в нем содержащиеся (проекции опущены).

Таблица 2

Клон	λ_2-двойственный клон	Пример базиса или порождающего множества	Вид функций, принадлежащих клону
E	E	e_2^2	e_i^n
E_φ	E_ψ	φ_0	φ_0
$E_{\varphi\psi}$	$E_{\varphi\psi}$	φ_0, ψ_0	φ_0, ψ_0
I_{l0}^{0}	I_{0l}^{1}	f_0^l	f_0^n $(n \leqq l)$
$I_{l0}^{0\varphi}$	$I_{0l}^{1\psi}$	φ_0, f_0^l	φ_0, f_0^n $(n \leqq l)$
$I_{l0}^{0\psi}$	$I_{0l}^{1\varphi}$	ψ_0, f_0^l	ψ_0, f_0^n $(n \leqq l)$
$I_{l0}^{0\varphi\psi}$	$I_{0l}^{1\varphi\psi}$	φ_0, ψ_0, f_0^l	φ_0, ψ_0, f_0^n $(n \leqq l)$
I_{lm}	I_{ml}	f_0^l, $1+f_0^m$	$f_0^{n_1}$ $(n_1 \leqq l)$, $1+f_0^{n_2}$ $(n_2 \leqq m)$
I_{lm}^{φ}	I_{ml}^{ψ}	φ_0, f_0^l, $1+f_0^m$	$\varphi_0, f_0^{n_1}$ $(n_1 \leqq l)$, $1+f_0^{n_2}$ $(n_2 \leqq m)$
I_{lm}^{ψ}	I_{ml}^{φ}	ψ_0, f_0^l, $1+f_0^m$	$\psi_0, f_0^{n_1}$ $(n_1 \leqq l)$, $1+f_0^{n_2}$ $(n_2 \leqq m)$
$I_{lm}^{\varphi\psi}$	$I_{ml}^{\varphi\psi}$	φ_0, ψ_0, f_0^l, $1+f_0^m$	φ_0, ψ_0, $f_0^{n_1}$ $(n_1 \leqq l)$, $1+f_0^{n_2}$ $(n_2 \leqq m)$
$I_{\infty 0}^{0}$	$I_{0\infty}^{1}$	базнса нет f_0^l $(l=1, 2, \ldots)$	f_0^n $(n \geqq 0)$
$I_{\infty 0}^{0\varphi}$	$I_{0\infty}^{1\psi}$	базиса нет φ_0, f_0^l $(l=1, 2, \ldots)$	φ_0, f_0^n $(n \geqq 0)$
$I_{\infty}^{0\psi}$	$I_{0\infty}^{1\varphi}$	базиса нет ψ_0, f_0^l $(l=1, 2, \ldots)$	ψ_0, f_0^n $(n \geqq 0)$
$I_{\infty 0}^{0\varphi\psi}$	$I_{0\infty}^{1\varphi\psi}$	базиса нет φ_0, ψ_0, f_0^l $(l=1, 2, \ldots)$	φ_0, ψ_0, f_0^n $(n \geqq 0)$
$I_{\infty m}$	$I_{m\infty}$	базиса нет $1+f_0^m, f_0^l$ $(l=1, 2, \ldots)$	$f_0^{n_1}$ $(n_1 \geqq 0)$ $1+f_0^{n_2}$ $(n_2 \leqq m)$
$I_{\infty m}^{\varphi}$	$I_{m\infty}^{\psi}$	базиса нет φ_0, f_0^l $(l=1, 2, \ldots)$, $1+f_0^m$	$\varphi_0, f_0^{n_1}$ $(n_1 \geqq 0)$, $1+f_0^{n_2}$ $(n_2 \leqq m)$
$I_{\infty m}^{\psi}$	$I_{m\infty}^{\varphi}$	базиса нет ψ_0, f_0^l $(l=1, 2, \ldots)$, $1+f_0^m$	$\psi_0, f_0^{n_1}$ $(n_1 \geqq 0)$, $1+f_0^{n_2}$ $(n_2 \leqq m)$
$I_{\infty m}^{\varphi\psi}$	$I_{m\infty}^{\varphi\psi}$	базиса нет φ_0, ψ_0, f_0^l $(l=1, 2, \ldots)$, $1+f_0^m$	$\varphi_0, \psi_0, f_0^{n_1}$ $(n_1 \geqq 0)$, $1+f_0^{n_2}$ $(n_2 \leqq m)$
$I_{\infty\infty}$	$I_{\infty\infty}$	базиса нет f_0^m, $1+f_0^m$ $(m=1, 2, \ldots)$	f_0^n, $1+f_0^n$ $(n \geqq 0)$
$I_{\infty\infty}^{\varphi}$	$I_{\infty\infty}^{\psi}$	базиса нет φ_0, f_0^m, $1+f_0^m$ $(m=1, 2, \ldots)$	φ_0, f_0^n, $1+f_0^n$ $(n \geqq 0)$
$I_{\infty\infty}^{\varphi\psi}$	$I_{\infty\infty}^{\varphi\psi}$	базиса нет φ_0, ψ_0, f_0^m, $1+f_0^m$ $(m=1, 2, \ldots)$	φ_0, ψ_0, f_0^n, $1+f_0^n$ $(n \geqq 0)$
F_{00}^{φ}	F_{00}^{ψ}	$\bar{\varphi}_0$	φ_0, $\bar{\varphi}_0$
$F_{00}^{\varphi\psi}$	$F_{00}^{\varphi\psi}$	$\bar{\varphi}_0, \bar{\psi}_0$	φ_0, $\bar{\varphi}_0$, ψ_0, $\bar{\psi}_0$
F_{m}^{φ}	F_{m}^{ψ}	$\bar{\varphi}_0, f_0$	φ_0, $\bar{\varphi}_0$, f_0^n, $1+f_0^n$ $(n \leqq m)$

Таблица 2 (продолжение)

Клон	λ_2-двойственный клон	Пример базиса или порождающего множества	Вид функцуй, принадлежащих клону
$F_m^{\varphi\psi}$	$F_m^{\varphi\psi}$	$\bar{\varphi}_0, \bar{\psi}_0, f_0^m$	$\varphi_0, \bar{\varphi}_0, \psi_0, \bar{\psi}_0, f_0^n,$ $1+f_0^n$ $(n\leqq m)$
F_∞^φ	F_∞^ψ	$\bar{\varphi}_0, f_0^m$ $(m=1, 2, \ldots)$	$\varphi_0, \bar{\varphi}_0, f_0^n,$ $1+f_0^n$ $(n\geqq 0)$
$F_\infty^{\varphi\psi}$	$F_\infty^{\varphi\psi}$	$\bar{\varphi}_0, \bar{\psi}_0, f_0^m$ $(m=1, 2, \ldots)$	$\varphi_0, \bar{\varphi}_0, \psi_0, \bar{\psi}_0, f_0^n,$ $1+f_0^n$ $(n\geqq 0)$
L_φ	L_ψ	u_0^3	u_0^{2k+1} $(k>0)$
S_φ	S'_ψ	u_0^2	u_0^n $(n\geqq 0)$
L'_φ	L'_ψ	$1+u_0^3$	$u_0^{2k+1}, 1+u_0^{2k+1}$ $(k\geqq 0)$
S'_φ	S_ψ	$1+u_0^2$	$u_0^{2k+1}, 1+u_0^{2k}$ $(k\geqq 0)$
U_φ	U_ψ	$u_0^2, 1+u_0^2$	$u_0^n, 1+u_0^n$ $(n\geqq 0)$
$L_{\varphi+\gamma}$	$L_{\varphi+\gamma}$	$\varphi_0+\gamma_0$	$\varphi_0+f_0^n, \psi_0+f_0^n$
$L_{\varphi\psi}$	$L_{\varphi\psi}$	u_0^3, p_0^2	$g\gamma_0+f\varphi_0+p\psi_0,$ $f+p=2k+1$
$L_{\varphi+\delta}$	$L_{\varphi+\psi}$	$1+\gamma_0+\varphi_0$	$c+\varphi_0+f_0^n, c+\psi_0+f_0^n,$ $(c\in\{0, 1\}, n\geqq 0)$
$L_{\varphi\psi}^1$	$L_{\varphi\psi}^1$	$1+\varphi_0+\psi_0+\psi_0$	$c+g\gamma_0+f\varphi_0+p\psi_0,$ $f+p=2k+1$
H_φ	H_ψ	$\varphi_0+\gamma_0, c_0$	$f_0^n, \varphi_0+f_0^n, \psi_0+f_0^n$ $(n\geqq 0)$
$S_{\varphi\psi}$	$S'_{\varphi\psi}$	$\varphi_0+\psi_0$	$g\gamma_0+f\varphi_0+p\psi_0$
$H_{\varphi\psi}$	$H_{\varphi\psi}$	$\varphi_0+\gamma_0, c_0, c_1$	$f_0^n, 1+f_0^n, \varphi_0+f_0^n,$ $\psi_0+f_0^n$ $(n\geqq 0)$
G	G	$\varphi_0+\gamma_0, \bar{\varphi}_0, c_0$	$c+f_0^n, c+\varphi_0+f_0^n,$ $c+\psi_0+f_0^n$ $(n\geqq 0)$
Z_0	Z_0	$\varphi_0+\psi_0, \bar{\varphi}_0$	$c+g\gamma_0+f\varphi_0+p\psi_0.$

2. Метод дальнейшего описания подклонов клона B

Для описания всех подклонов клона B достаточно привести список всех таких подклонов. Однако если этим и ограничиться, то останутся без ответа следующие важные вопросы: каково взаимное расположение этих подклонов, какие функции содержит каждый подклон, какие у него имеются максимальные подклоны. Конечно, многое зависит от способа задания подклонов, однако в любом случае трудно ожидать, что конкретный список даст удовлетворительный ответ на каждый из этих вопросов. К тому же весьма трудной представляется проблема доказательства полноты такого списка.

Исключительно удобным и наглядным является задание решетки с помощью ее диаграммы. Достаточно взглянуть на диаграмму решетки Поста $\mathscr{L}(\mathscr{F}_A^*)$ всех подалгебр алгебры $\mathscr{F}_A^*$, $|A|=2$, [13], чтобы получить ответ на многие вопросы: конечная она или бесконечная, какова ее высота, ширина и т. д. Приведенная там же таблица подалгебр вместе с указанной диаграммой дает полную информацию об элементах решетки. Аналогичную задачу нам удалось решить в работе [6]: Построена диаграмма решетки $\mathscr{L}(Z_0)$ подклонов клона Z_0 и приведен список всех подклонов с указанием базисов, содержащихся функций и т. д.

К сожалению, поступить подобным образом при описании подклонов клона B мы не сможем. Решетка $\mathscr{L}(B)$ содержит 3 копии решетки $\mathscr{L}(Z_0)$, технически уже трудно изобразить их на одном рисунке. В дальнейшем будет показано, что средняя часть решетки $\mathscr{L}(B)$ устроена ещё сложнее.

Чтобы обойти указанные трудности и в тоже время получить достаточно наглядное представление о строении решетки $\mathscr{L}(B)$, примем следующий план. Пусть все подклоны клона B составляют внутренность шестиугольника на рисунке 5. Треугольники a, b, c содержат клоны, значения функций которых попадают в множества $\{0, 1\}$, $\{0, 2\}$, и $\{1, 2\}$ соответственно. Таким образом треугольник a содержит все подклоны клона Z_0. Образуемая ими решетка изображена на рисунке 1, а сами клоны, их базисы и функции указаны в таблице 2. Очевидно, треугольники b и c содержат лишь изоморфные копии клонов, содержащихся в треугольнике a. Пунктирная линия, делящая треугольник a пополам, указывает на существование автоморфизма клона Z_0, вследствие чего в описании нуждается лишь часть его подклонов.

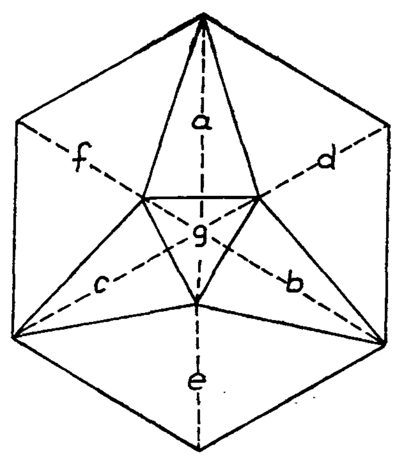


Рис. 5

Данная работа посвящена описанию клонов, попадающих в четырехугольник d — клонов, содержащих как функции, значения которых принадлежат множеству $\{0, 1\}$, так и функции, значения которых принадлежат множеству $\{0, 2\}$. К сожалению, образуемая этими клонами решетка уже настолько сложна, что изобразить ее на одном рисунке невозможно, и мы довольствуемся тем, что рисуем отдельные ее фрагменты, получаемые следующим образом. Берется некоторый фрагмент решетки $\mathscr{L}(Z_0)$ (чаще интервал) и для каждого клона K из этого фрагмента строится новый клон ${}^{\mu}K$ следующим образом: к порождающим клона K добавляется функция $\mu \notin Z_0$, получается система порождающих клона ${}^{\mu}K$. В некоторых случаях получаем семейство клонов, для элементов которого включения очевидны, и мы довольствуемся диаграммой. В более сложных случаях дается доказательство.

Пунктирная линия в четырехугольнике d также указывает на наличие автоморфма, позволяющего описывать лишь клоны, принадлежащие одному из двух треугольников. Область e содержит клоны функций типов c и b, область f-клоны типов a и c. Очевидно, обе они в отдельном описании не нуждаются.

Ограниченный объем статьи не позволяет привести описание клонов, содержащих одновременно функции всех трех типов-клонов, попадающих в треугольник g. Этих клонов относительно немного, что упрощает построения. В треугольнике g также находятся клоны, содержащие одноместные функции, принадлежащие три значения.

3. Клоны типов ${}^{\varphi}K$, ${}^{2}K$, ${}^{2\varphi}K$, ${}^{\bar\varphi}K$ при $K \leqq Z_0$

Пусть K-подклон клона Z_0, Z_2-клон, образованный теми функциями из B, значения которых принадлежат множеству $\{0, 2\}$, $\mu \in Z_2$, M — система порождающих функций клона K. Клон, порождаемый множеством $M \cup \{\mu\}$ будем обозначать либо через $K \sqcup \mu$, либо через ${}^{\mu}K$. В данном параграфе последнее обозначение будет употребляться только в том случае, когда K является максимальным подклоном клона $K \sqcup \mu$. Поскольку клонов такого типа много, то их описание, описание их максимальных подклонов и даже описание их обозначений получаются очень громоздкими. С целью упрощения изложения мы приводим рисунки, из которых легко уяснить обозначения вновь появляющихся клонов, их взаимное разположение, выделить все максимальные подклоны.

В дальнейшем через $f^{\triangledown}$ будет обозначаться множество всех тех функций, которые могут быть получены из $f \in O_A$ добавлением и изъятием несущественных переменных и применением операций ζ и τ. Введем специальные обозначения для девяти одноместных функций. Первые восемь из них принадлежат клону Z_2 и λ_0-двойственны одноместным функциям клона Z_0 (Табл. 3).

Сначала мы опишем клоны, порождаемые функциями подклонов клона $F_{\infty}^{\varphi\psi}$ (см. рисунок 3) вместе с функциями φ_2, c_2 или $\bar\varphi_2$. Рисунок 6 дает представление о взаимном расположении описываемых интервалов (заштрихованы) в решетке $\mathscr{L}(B)$. Эти интервалы не изоморфны друг другу, однако по меньшей мере три средних изоморфно вкладываются в решетку $\mathscr{L}(F_{\infty}^{\varphi\psi})$, то есть в нижний интервал.

Пусть $K \leqq F_{\infty}^{\varphi\psi}$. Так как $\varphi_2 * \varrho = c_0$ для любой функции $\varrho \in Z_0$, $\varrho \notin E$, то клон $K \sqcup \varphi_2$ содержит c_0. Из $\psi_2 * \varphi_2 = \gamma_0$ следует, что $\psi_0 \in K \Rightarrow \gamma_0 \in K \sqcup \varphi_2$. Равенства $\gamma_0 * \varphi_2 = \gamma_0$, $\varphi_0 * \varphi_2 = c_0$ показывают, что если клон K содержит γ_0, то $K \sqcup \varphi_2$ отличается от него лишь функциями из $\varphi_2^{\triangledown}$. Очевидно, для любого $K \leqq F_{\infty}^{\varphi\psi}$ клон K является максимальным подклоном клона $K \sqcup \varphi_2$. Получаем семейство клонов, образующее решетку, изображенную на рисунке 7. Из этого рисунка легко определить все отличные от K максимальные подклоны клона ${}^{\varphi}K$.

Подклоны клона Z_{02}, λ_0-двойственные подклонам клона ${}^{\varphi}F_{\infty}^{\varphi\psi}$, образуют решетку, изоморфную решетке $\mathscr{L}({}^{\varphi}F_{\infty}^{\varphi\psi})$. Согласно заключенному выше соглашению, их обозначения получаем добавлением к обозначениям соответст-

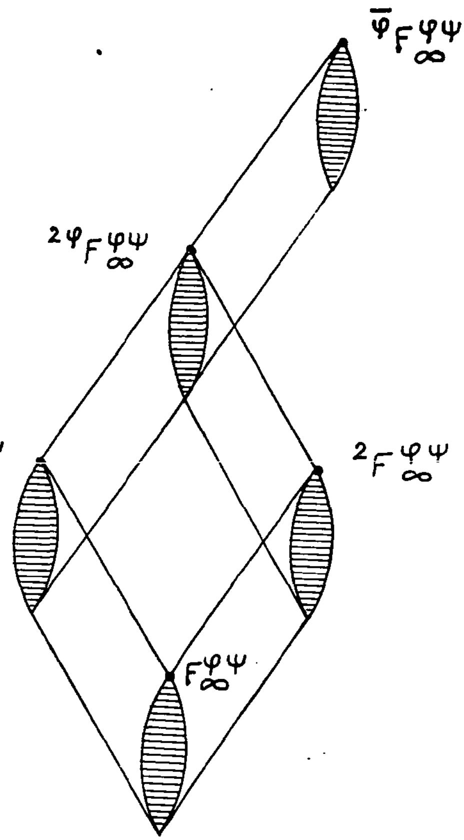


Рис. 6

вующих подклонов клона ${}^{\varphi}F_\infty^{\varphi\psi}$ слева внизу индекса λ_0. Решетки $\mathscr{L}({}^{\varphi}F_\infty^{\varphi\psi})$ и $\mathscr{L}({}_{\lambda_0}^{\varphi}F_\infty^{\varphi\psi})$ имеют несколько общих элементов. Так как клоны первой решетки из одноместных функций могут содержать лишь e_1^1, c_0, c_1, φ_0, ψ_0, γ_0, $\bar{\varphi}_0$, φ_2, а второй — e_1^1, c_0, c_2, φ_2, ψ_2, γ_2, $\bar{\varphi}_2$, φ_0, то общими являются только клоны, содержащие e_1^1, c_0, φ_0, φ_2, то есть E, ${}^{\varphi}E$, I_{00}^0, ${}^{\varphi}I_{00}^0$, E_φ, $I_{00}^{0\varphi}$, ${}^{\varphi}I_{00}^{0\varphi}$.

Так как $\gamma_0 * c_2 = c_1$, то из $f_n \in K$ и $1+f^m \in K$ следует, что $1+f_{n-1} \in K \sqcup c_2$ и $f_{m-1} \in K \sqcup c_2$ для любого $K \leqq F_\infty^{\varphi\psi}$. Если $K \sqcup c_2$ содержит φ_0 или $\bar{\psi}_0$, то он содержит и c_0, а из $\psi \in K \sqcup c_2$ или $\bar{\varphi} \in K \sqcup c_2$ следует $c_1 \in K \sqcup c_2$. Легко убедиться, что в остальном клон $K \sqcup c_2$ отличается от K лишь функциями из c_2^∇. Клоны указанного вида образуют решетку, изображенную на рисунке 8. В пересечение $\mathscr{L}({}^{2}F_\infty^{\varphi\psi})$ с $\mathscr{L}({}_{\lambda_0}^{2}F_\infty^{\varphi\psi})$ попадают клоны E и I_{00}.

Переходя к описанию клонов вида ${}^{2}K \sqcup \varphi_2$, где ${}^{2}K \leqq F_\infty^{\varphi\psi}$, заметим, что все комбинации содержащихся в ${}^{2}K \sqcup \varphi_2$ функций уже рассмотрены, кроме $\varphi_2 * c_2 = c_2 * \varphi_2 = c_2$, которые показывают, что ${}^{2}K \sqcup \varphi_2 = {}^{2}K \cup \varphi_2^\nabla$, то есть ${}^{2}K \sqcup \varphi_2 = {}^{\varphi 2}K = {}^{2\varphi}K$. Клоны вида ${}^{2\varphi}K$ образуют решетку, изображенную на рисунке

9. Кроме клонов, изображенных на этом рисунке, максимальными подклонами клона ${}^{2\varphi}K$ *являются* клоны ${}^{2}K$ и ${}^{\varphi}K$.

Так как $\bar\varphi_2 * \varrho = c_2$ для любого $\varrho \in Z_0$ и $\varphi_2 * \varphi_2 = \varphi_2$, то клон $K \sqcup \bar\varphi$, $K \leqq F_\infty^{\varphi\psi}$, содержит c_2 и φ_2; то есть $K \sqcup \bar\varphi_2 = {}^{2\varphi}K \sqcup \bar\varphi_2$. Из $\gamma_0 * \bar\varphi_2 = \delta_0$, $\delta_0 * \bar\varphi_2 = \gamma_0$ следует важный вывод: если $f_n \in {}^{2\varphi}K \sqcup \varphi_2$, $1 + f_m \in {}^{2\varphi}K \sqcup \varphi_2$, то $1 + f_n \in {}^{2\varphi}K \sqcup \varphi_2$ и $f_m \in {}^{2\varphi}K \sqcup \sqcup \varphi_2$. Имеются еще равенства

$$\varphi_0 * \bar\varphi_2 = c_0, \quad \psi_0 * \varphi_2 = \delta_0, \quad \bar\varphi_0 * \bar\varphi_2 = c_1, \quad \bar\psi_0 * \bar\varphi_2 = \gamma_0, \quad \bar\varphi_2 * c_2 = c_0,$$

однако если ${}^{2\varphi}K \geqq {}^{2\varphi}I_{11}$, то $\{c_0, c_1, c_2, \gamma_0, \delta_0\} \subseteqq {}^{2\varphi}K$. Получаем решетку клонов вида ${}^{\bar\varphi}K$ (рисунок 10).

Клонов вида $K \sqcup \varrho_1$, $K \sqcup \{\varrho_1, \varrho_2\}$, где $\varrho_1, \varrho_2 \in \{\varphi_2, \bar\varphi_2, c_2\}$ и $K \nleqq F_\infty^{\varphi\psi}$, $K \leqq Z_0$ лишь конечное число, и притом небольшое. Все они указаны в таблице 4, которая одновременно является и доказательством отсутствия других клонов указанного вида. Сама таблица получена следующим образом. Берется клон K, удовлетворяющий указанным выше ограничениям. Из таблицы 2 находится вид функций, в нем содержащихся. Учитывая результаты суперпозиций $\mu * \varrho$, где μ — функция из Z_0, а $\varrho \in \{\varphi_2, c_2, \bar\varphi_2\}$, находим вид функций клонов $K \sqcup \varphi_2$, $K \sqcup c_2$, $K \sqcup \varphi_2 \sqcup c_2$, $K \sqcup \bar\varphi_2$. По таблице 2 устанавливаем максимальный подклон этого клона.

Для удобства читателя в таблице 5 указан состав клонов $L_\psi - S'_{\varphi\psi}$. Функции новых клонов из таблицы 4 легко находятся по следующим правилам:

$${}^{\varphi}K = K \cup \varphi_2^{\nabla}, \quad {}^{2}K = K \cup c_2^{\nabla}, \quad {}^{2\varphi}K = K \cup \{\varphi_2^{\nabla}, c_2^{\nabla}\}, \quad {}^{\bar\varphi}K = K \cup \{\varphi_2^{\nabla}, \bar\varphi_2^{\nabla}, c_2^{\nabla}\}$$

Таблица 4

K	$K \sqcup \varphi_2$	$K \sqcup c_2$	$K \sqcup \varphi_2 \sqcup c_2$	$K \sqcup \bar\varphi_2$
L_φ	${}^{\varphi}S_\varphi$	${}^{2}S_\varphi$	${}^{2\varphi}S_\varphi$	${}^{\bar\varphi}S_\varphi$
S_φ	${}^{\varphi}S_\varphi$	${}^{2}S_\varphi$	${}^{\varphi}S_\varphi$	${}^{\bar\varphi}U_\varphi$
L'_φ	${}^{\varphi}U_\varphi$	${}^{2}U_\varphi$	${}^{2\varphi}U_\varphi$	${}^{\bar\varphi}U_\varphi$
S'_φ	${}^{\varphi}U_\varphi$	${}^{2}U_\varphi$	${}^{2\varphi}U_\varphi$	${}^{\bar\varphi}U_\varphi$
U_φ	${}^{\varphi}U_\varphi$	${}^{2}U_\varphi$	${}^{\varphi}U_\varphi$	${}^{\bar\varphi}U_\varphi$
$L_{\varphi+\gamma}$	${}^{\varphi}H_\varphi$	${}^{2}H_{\varphi\psi}$	${}^{2\varphi}H_{\varphi\psi}$	${}^{\bar\varphi}G$
$L_{\varphi\psi}$	${}^{\varphi}S_{\varphi\psi}$	${}^{2}Z_0$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
$L_{\varphi+\delta}$	${}^{\varphi}H_{\varphi\psi}$	${}^{2}H_{\varphi\psi}$	${}^{2\varphi}H_{\varphi\psi}$	${}^{\bar\varphi}G$
$L^1_{\varphi\psi}$	${}^{\varphi}Z_0$	${}^{2}Z_0$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
H_φ	${}^{\varphi}H_\varphi$	${}^{2}H_{\varphi\psi}$	${}^{2\varphi}H_{\varphi\psi}$	${}^{\bar\varphi}G$
$S_{\varphi\psi}$	${}^{\varphi}S_{\varphi\psi}$	${}^{2}Z_0$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
$H_{\varphi\psi}$	${}^{\varphi}H_{\varphi\psi}$	${}^{2}H_{\varphi\psi}$	${}^{2\varphi}H_{\varphi\psi}$	${}^{\bar\varphi}G$
G	${}^{\varphi}G$	${}^{2}G$	${}^{2\varphi}G$	${}^{\bar\varphi}G$
Z_0	${}^{\varphi}Z_0$	${}^{2}Z_0$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
L_ψ	${}^{\varphi}S_{\varphi\psi}$	${}^{2}S'_\psi$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
S'_ψ	${}^{\varphi}Z_0$	${}^{2}S'_\psi$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
L'_ψ	${}^{\varphi}Z_0$	${}^{2}U_\psi$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
S_ψ	${}^{\varphi}S_{\varphi\psi}$	${}^{2}U_\psi$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
U_ψ	${}^{\varphi}Z_0$	${}^{2}U_\psi$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$
H_ψ	${}^{\varphi}H_\psi$	${}^{2}H_{\varphi\psi}$	${}^{2\varphi}H_{\varphi\psi}$	${}^{\bar\varphi}G$
$S'_{\varphi\psi}$	${}^{\varphi}Z_0$	${}^{\varphi}Z_0$	${}^{2\varphi}Z_0$	${}^{\bar\varphi}Z_0$

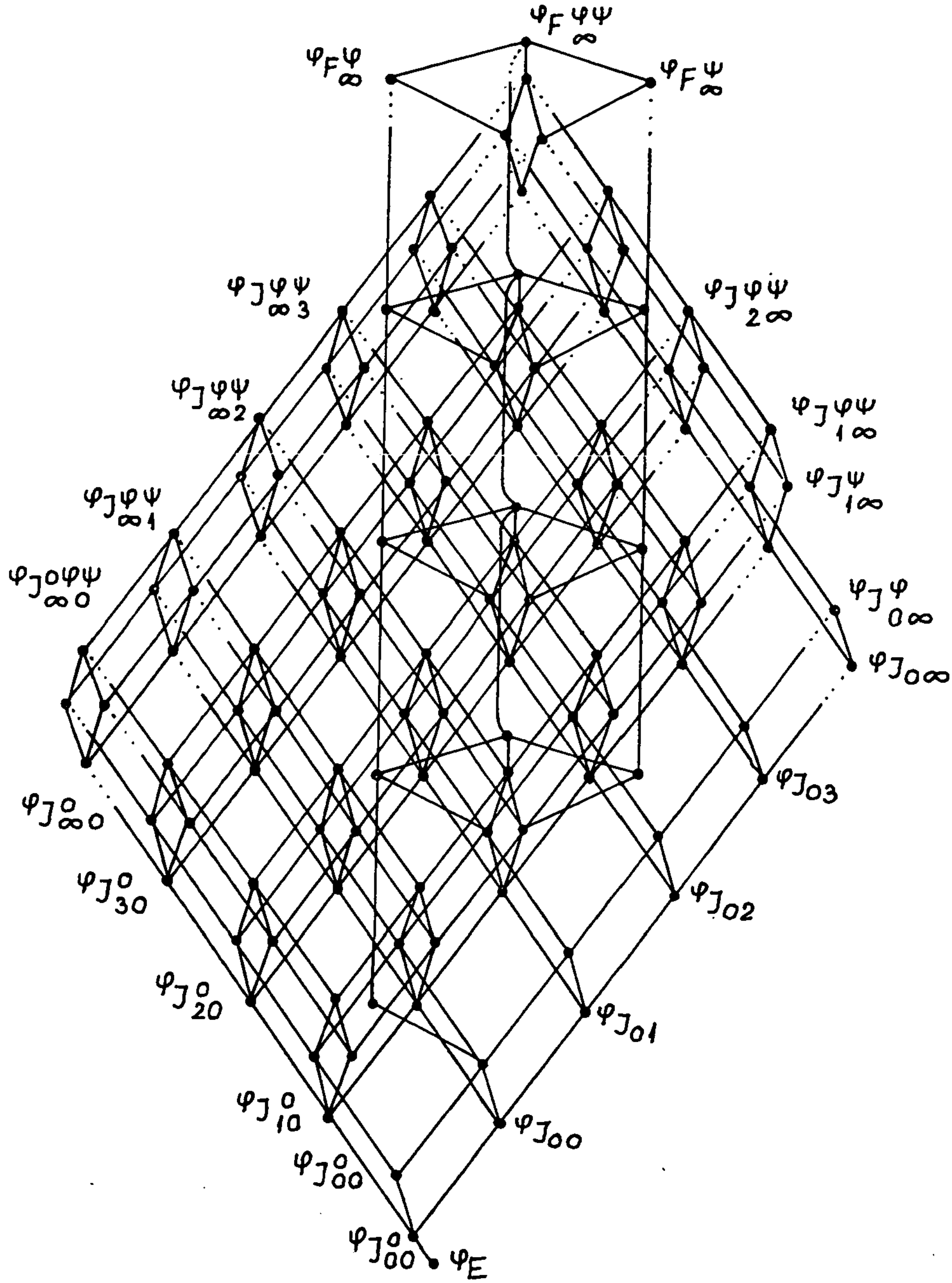

$\varphi_{F_\infty^{\varphi\psi}}$
$\varphi_{F_\infty^{\varphi}}$
$\varphi_{F_\infty^{\psi}}$
$\varphi_{J^{\varphi\psi}_{\infty 3}}$
$\varphi_{J^{\varphi\psi}_{2\infty}}$
$\varphi_{J^{\varphi\psi}_{\infty 2}}$
$\varphi_{J^{\varphi\psi}_{1\infty}}$
$\varphi_{J^{\varphi\psi}_{\infty 1}}$
$\varphi_{J^{\psi}_{1\infty}}$
$\varphi_{J^{0\varphi\psi}_{\infty 0}}$
$\varphi_{J^{\varphi}_{0\infty}}$
$\varphi_{J_{0\infty}}$
$\varphi_{J^{0}_{\infty 0}}$
$\varphi_{J_{03}}$
$\varphi_{J^{0}_{30}}$
$\varphi_{J_{02}}$
$\varphi_{J^{0}_{20}}$
$\varphi_{J_{01}}$
$\varphi_{J^{0}_{10}}$
$\varphi_{J_{00}}$
$\varphi_{J^{0}_{00}}$
$\varphi_{J^{0}_{00}}$
φ_E

Рис. 7

Таблица 5

Клон	Базис	Содержит функции
L_ψ	p_0^3	p_0^{2k+1}
S'_ψ	$1+p_0^2$	p_0^{2k+1}, $1+p_0^{2k}$
L'_ψ	$1+p_0^3$	p_0^{2k+1}, $1+p_0^{2k+1}$
S_ψ	p_0^2	p_0^n
U_ψ	p_0^2, $1+p_0^2$	p_0^n, $1+p_0^n$
H_ψ	$\psi_0+\gamma_0$, c_1	$1+f_0^n$, $\psi_0+f_0^n$, $\varphi_0+f_0^n$
S'_φ	$1+\varphi_0+\psi_0$	$1+g\gamma_0+f\varphi_0+p\psi_0$, $f+p=2k$, $g\gamma_0+f\varphi_0+p\psi_0$, $f+p=2k+1$.

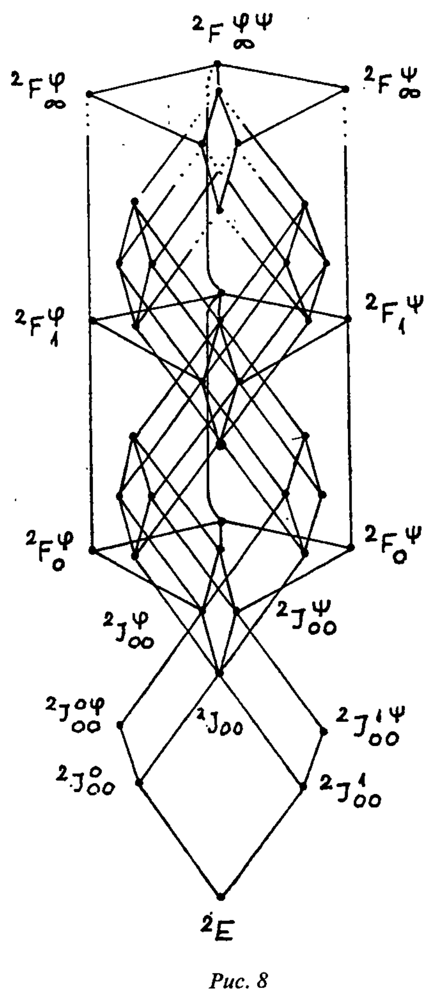


Рис. 8

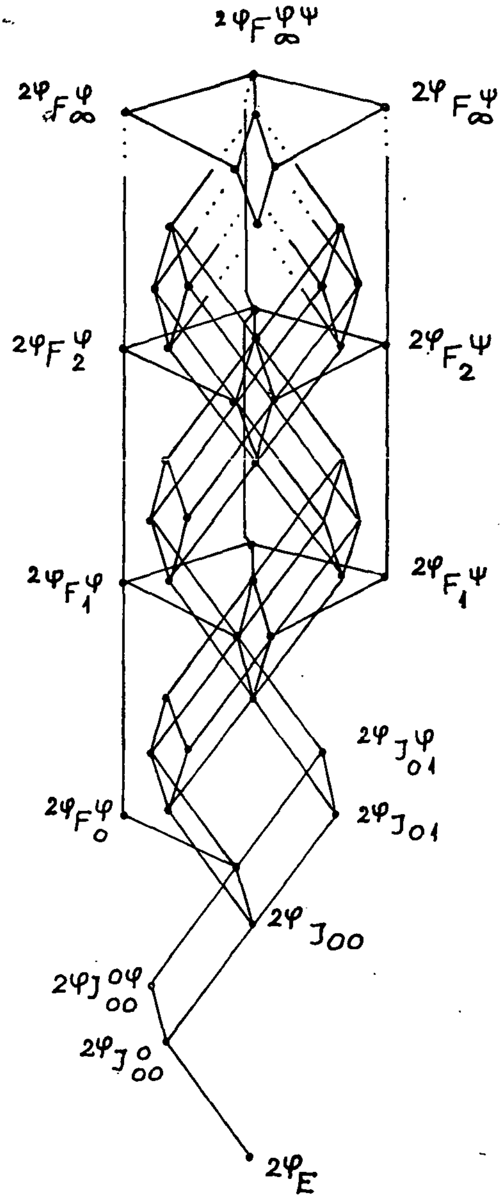

2φF∞φψ
2φF∞φ
2φF∞ψ
2φF2φ
2φF2ψ
2φF1φ
2φF1ψ
2φJ01φ
2φF0φ
2φJ01
2φJ00
2φJ00 0φ
2φJ00 0
2φE

Рис. 9

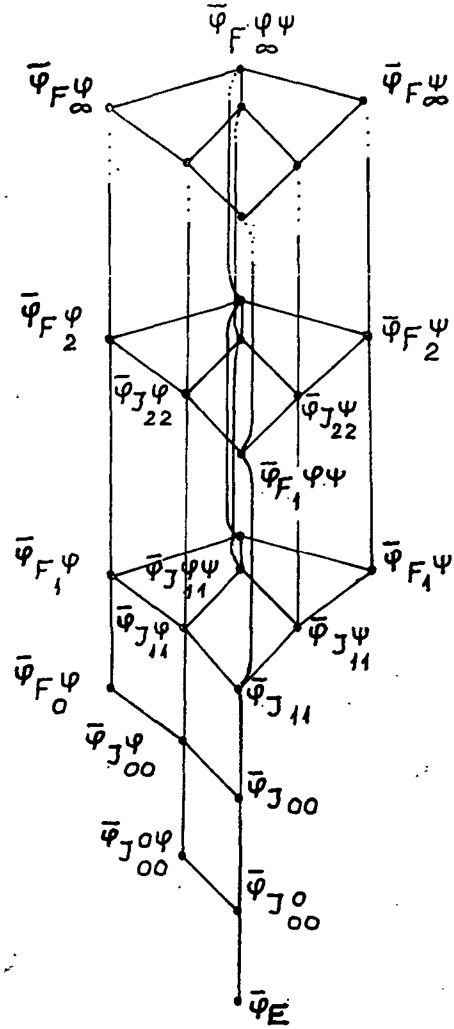

φ̄F∞φψ
φ̄F∞φ
φ̄F∞ψ
φ̄F2φ
φ̄F2ψ
φ̄J22φ
φ̄J22ψ
φ̄F1φψ
φ̄F1φ
φ̄J11φψ
φ̄F1ψ
φ̄J11φ
φ̄J11ψ
φ̄F0φ
φ̄J11
φ̄J00φ
φ̄J00
φ̄J00 0φ
φ̄J00 0
φ̄E

Рис. 10

Рисунок 11 дает наглядное представление о взаимном расположении новых клонов. Здесь легко заметить «слоистость» в строении, отмеченную на рисунке 6. Читатель может выделить эти слои и присоединить их к соответствующим слоям на рисунках 7—10.

4. Другие подклоны клона $Z_0 \cup Z_2$, порождаемые с помощью унарных функций из Z_2

Большое количество клонов, описанных в предыдущем параграфе, объясняется определенной «нейтральностью» функций φ_2, $\bar{\varphi}_2$, c_2 по отношению к функциям из Z_0: функции φ_2, $\bar{\varphi}_2$, c_2 не меняют значения при изменении переменной на множестве $0, 1$. Оставшиеся одноместные функции из Z_2 этим

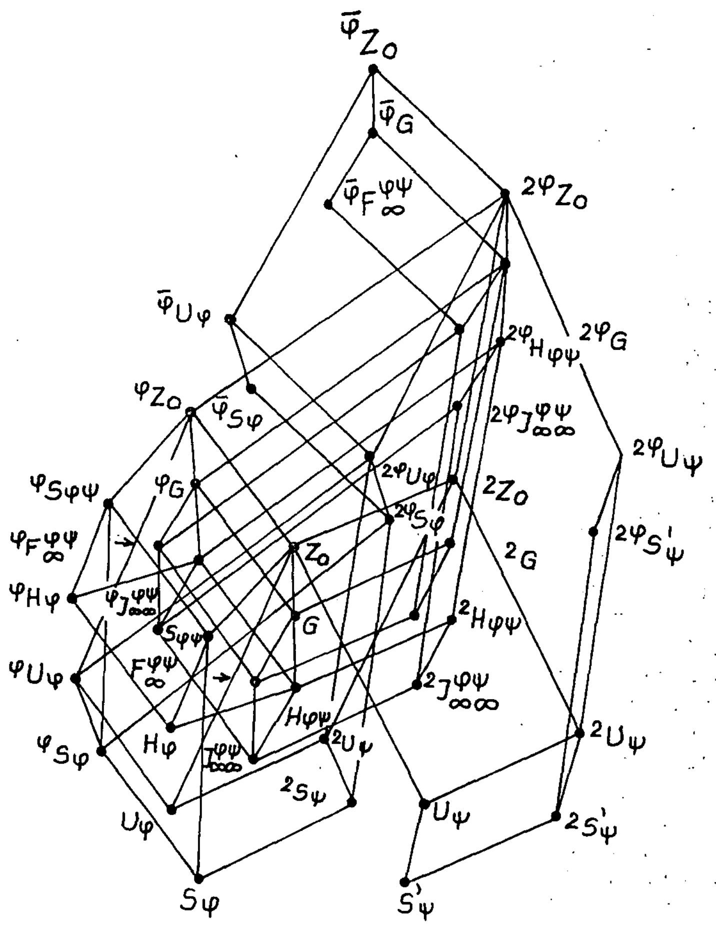


Рис. 11

всойством не обладают, и потому число новых клонов, получаемых с их участием, невелико.

В дальнейших рассуждениях без ссылок используется информация о подклонах клона Z_0, содержащаяся в таблицах 2 и 5. Для удобства читателя приводим также таблицу 6, содержащую результаты суперпозиций одноместных функций.

Таблица 6

ϱ	γ_0	δ_0	φ_0	$\bar\varphi_0$	ψ_0	$\bar\psi_0$	c_0	c_1
$\varrho*\gamma_2$	φ_0	$\bar\varphi_0$	c_0	c_1	φ_0	$\bar\varphi_0$	c_0	c_1
$\gamma_2*\varrho$	φ_2	$\bar\varphi_2$	γ_2	δ_2	ψ_2	$\bar\psi_2$	c_0	c_2
$\varrho*\psi_2$	ψ_0	$\bar\psi_0$	c_0	c_1	ψ_0	$\bar\psi_0$	c_0	c_2
$\psi_2*\varrho$	φ_2	$\bar\varphi_2$	γ_2	δ_2	ψ_2	$\bar\psi_2$	c_0	c_2
$\varrho*\delta_2$	$\bar\varphi_0$	φ_0	c_0	c_1	$\bar\varphi_0$	φ_0	c_0	c_1
$\delta_2*\varrho$	$\bar\varphi_2$	φ_2	δ_2	γ_2	$\bar\psi_2$	ψ_2	c_2	c_0
$\varrho*\bar\psi_2$	$\bar\psi_0$	ψ_0	c_0	c_1	$\bar\psi_0$	ψ_0	c_0	c_1
$\bar\psi_2*\varrho$	$\bar\varphi_2$	φ_2	δ_2	γ_2	$\bar\psi_2$	ψ_2	c_2	c_0

Пусть K-клон, порождаемый функциями u_0^3 и γ_2. Очевидно, $K\supset L_\varphi$. Так как $\varphi_0*\gamma_2=c_0$, то K содержит все функции u_0^n, то есть $K\supset S_\varphi$. Очевидно, клон K содержит также функции $\gamma_2*u_0^n$; других функций в нем нет. Клон K обозначим через ${}^\gamma S_\varphi$.

Клон ${}^\gamma U_\varphi$ порождается функциями $1+u_0^2, \gamma_2$. Содержит функции u_0^{2k+1}, $1+u_0^{2k}$, а потому и $u_0^{2k}, 1+u_0^{2k+1}$, $\gamma_2*u_0^n$, $\gamma_2*(1+u_0^n)$. Очевидно, $L'_\varphi\sqcup\gamma_2={}^\gamma U_\varphi$.

Клон ${}^\gamma S_{\varphi\psi}$ порождается функциями p_0^3, γ_2 и потому ${}^\gamma S_{\varphi\psi}\supset L_\psi$. Так как $\psi_0*\gamma_2=\varphi_0$, то ${}^\gamma S_{\varphi\psi}\supset L_\varphi$, отсюда ${}^\gamma S_{\varphi\psi}\supset S_\varphi$ и потому ${}^\gamma S_{\varphi\psi}\supset S_{\varphi\psi}$. Этот клон содержит функции $g\gamma_0+f\varphi_0+p\psi_0$, $\gamma_2(g\gamma_0+f\varphi_0+p\psi_0)$. Другие базисы этого клона — $\gamma_0+\gamma_0, \gamma_2$; $\varphi_0+\gamma_0, \gamma_2$; $\psi_0+\psi_0, \gamma_2$; $\varphi_0+\varphi_0, \gamma_2$.

Так как клоны L'_ψ, S'_ψ, $L^1_{\varphi\psi}$, U_ψ, $S'_{\varphi\psi}$ содержат L_ψ и не содержатся в $S_{\varphi\psi}$, то при добавлении γ_2 к базису любого из этих клонов получается $Z_0\sqcup\gamma_2=Z_0\cup Z_2$.

Клон, содержащий функции $1+\varphi_0+\gamma_0, \gamma_2$, содержит $L_{\varphi+\delta}$, то есть функции $c+\varphi_0+f_0^n$, $c+\psi_0+f_0^n$, а вместе с ними и все функции $c+g\gamma_0+f\varphi_0+p\psi_0, \gamma_2$ $(c+g\gamma_0+f\varphi_0+p\psi_0)$, и потому совпадает с $Z_0\cup Z_2$. Тот же результат получим, беря функции $1+\gamma_0+\gamma_0, \gamma_2$. Таким образом, семейство функций клона Z_0, содержащее существенно многоместную функцию, вместе с γ_2 может породить лишь один из четырех указанных выше клонов.

Клон ${}^\psi S_\varphi$ порождается функциями u_0^3, ψ_2. Так как $\psi_2*\varphi_0=\gamma_2$, то он содержит ${}^\gamma S_\varphi$, то есть функции u_0^n, $\gamma_2*u_0^n$ и ψ_2.

Клон ${}^\psi U_\varphi$ порождается функциями $1+u_0^2$, ψ_2. Содержит φ_0, а вместе с ней γ_2 и все функции из ${}^\gamma U_\varphi$, то есть u_0^n, $1+u_0^n$, $\gamma_2+u_0^n$, $\gamma_2*(1+u_0^n)$, ψ_2. Очевидно, этот же клон порождается функциями $1+u_0^3, \psi_2$.

Клон, порождаемый функциями $\gamma_0+\gamma_0$ и ψ_2 содержит $\psi_0+\gamma_0=(\gamma_0+\gamma_0)*\psi_2$, следовательно φ_0 и γ_2, и потому совпадает с ${}^\gamma S_{\varphi\psi}$. Тот же результат получим, взяв функции $\varphi_0+\gamma_0, \psi_2$ или u_0^3, p_0^3, ψ_2, так как $(\varphi_0+\gamma_0+\gamma_0)*\psi_2=\gamma_0+\gamma_0, \psi_2*u_0^3=\gamma_2$.

${}^\psi L_\psi=L_\psi\sqcup\psi_2$. Состоит из функций $p_0^{2k+1}, \psi_2*p_0^{2k+1}$ $(k\geqq 0)$.

${}^\psi S_\psi=S_\psi\sqcup\psi_2$. Содержит функции $p_0^n, \psi_2*p_0^n$, $n\geqq 0$.

${}^\psi S'_\psi=S'_\psi\sqcup\psi_2$. Содержит функции p_0^{2k+1}, $\psi_2*p_0^{2k+1}$, $1+p_0^{2k}$, $\psi_2*(1+p_0^{2k})$ $(k\geqq 0)$.

${}^{\psi}L'_{\psi}=L'_{\psi}\sqcup\psi_2$, состоит из функций p_0^{2k+1}, $1+p_0^{2k+1}$, $\psi_2*p_0^{2k+1}$, $\psi_2*(1+p_0^{2k+1})$ $(k\geqq 0)$.

${}^{\psi}U_{\psi}=U_{\psi}\sqcup\psi_2$, состоит из p_0^n, $1+p_0^n$, $\psi_2*p_0^n$, $\psi_2*(1+p_0^n)$ $(n\geqq 0)$.

Так как $L_{\varphi+\gamma}\sqcup\psi_2=L_{\varphi\psi}\sqcup\psi_2={}^{\psi}S_{\varphi\psi}$, а $S_{\varphi\psi}$-максимальный подклон клона Z_0, то все оставшиеся клоны, имеющие вид $K\sqcup\psi_2$, $K\nsubseteq S_{\varphi\psi}$, совпадают с $Z_0\cup Z_2$.

Из рассмотренных клонов вида ${}^{\psi}K$ не содержат γ_2 клоны ${}^{\psi}L_{\psi}$, ${}^{\psi}S_{\psi}$, ${}^{\psi}S'_{\psi}$, ${}^{\psi}L'_{\psi}$, ${}^{\psi}U_{\psi}$. Добавляя γ_2 к порождающим каждого из них, плучим ${}^{\gamma}S_{\varphi\psi}$ в первых двух случаях, и $Z_0\cup Z_2$-в остальных.

Очевидно, описывая клоны вида $K\sqcup\delta_2$, $K\subset Z_0$, содержащие существенно многоместные функции, достаточно рассмотреть клоны вида $K_1\sqcup\delta_2$, где $K_1\in\{{}^{\gamma}S_{\varphi}, {}^{\gamma}U_{\varphi}, {}^{\gamma}S_{\varphi\psi}\}$. Клон ${}^{\gamma}U_{\varphi}$ уже содержит δ_2, поэтому ${}^{\gamma}U_{\varphi}\sqcup\delta_2={}^{\gamma}U_{\varphi}$. Новым является клон ${}^{\delta}S_{\varphi}$, содержащий функции $u_0^n, \gamma_2*u_0^n, \delta_2*u_0^n$. Клон ${}^{\delta}S_{\varphi\psi}$ содержит функции $\varrho(g\gamma_0+f\varphi_0+p\psi_0)$, $\varrho\in\{\varphi_0, \gamma_2, \varrho_2\}$, и потому совпадает с $Z_0\cup Z_2$.

Займемся клонами вида $K\sqcup\bar{\psi}_2$, $K\in\{{}^{\psi}S_{\varphi}, {}^{\psi}U_{\varphi}, {}^{\psi}L_{\psi}, {}^{\psi}S_{\psi}, {}^{\psi}S'_{\psi}, {}^{\gamma}S_{\varphi\psi}, {}^{\psi}L_{\psi}, {}^{\psi}U'_{\psi}\}$. Клон ${}^{\bar{\psi}}S_{\varphi}$ порождается базисом u_0^2, $\bar{\psi}_2$. Содержит функции u_0^n, $\gamma_2*u_0^n$, $\delta_2*u_0^n$, ψ_2, $\bar{\psi}_2$.

${}^{\bar{\psi}}U_{\varphi}=U_{\varphi}\sqcup\bar{\psi}_2$. Содержит функции u_0^n, $1+u_0^n$, $\gamma_2*u_0^n$, $\gamma_2*(1+u_0^n)$, ψ_2, $\bar{\psi}_2$.

${}^{\bar{\psi}}L_{\psi}=L_{\psi}\sqcup\bar{\psi}_2$. Состоит из функций p_0^{2k+1}, $\psi_2*p_0^{2k+1}$, $\bar{\psi}_2*p_0^{2k+1}$.

${}^{\bar{\psi}}S_{\psi}=S_{\psi}\sqcup\psi_2$. Состоит из функций p_0^n, $\psi_2*p_0^n$, $\bar{\psi}_2*p_0^n$.

${}^{\bar{\psi}}S'_{\psi}=S_{\psi}\sqcup\psi_2$. Состоит из функций p_0^{2k+1}, $1+p_0^{2k}$, $\psi_2*p_0^{2k+1}$, $\psi_2*(1+p_0^{2k})$, $\psi_2*p_0^{2k+1}$, $\psi_2*(1+p_0^{2k})$.

Из $\psi_2*\bar{\psi}_0=\bar{\psi}_2$ заключаем, что ${}^{\psi}L'_{\psi}\sqcup\bar{\psi}_2={}^{\psi}L'_{\psi}$, ${}^{\psi}U'_{\psi}\sqcup\bar{\psi}_2={}^{\psi}U'_{\psi}$. Клон ${}^{\gamma}S_{\varphi\psi}\sqcup\bar{\psi}_2$ содержит $\delta_2=\bar{\psi}_2*\gamma_2$ и потому совпадает с $Z_0\cup Z_2$.

Клоны ${}^{\bar{\psi}}L_{\psi}$, ${}^{\bar{\psi}}S_{\psi}$, ${}^{\bar{\psi}}S'_{\psi}$ не содержат δ_2. Добавление к любому из них этой функции дает $Z_0\cup Z_2$.

Все вновь полученные клоны указаны в таблице 7.

Таблица 7

Клон	Базис	Содержит функции
${}^{\gamma}S_{\varphi}$	u_0^2, γ_2	$u_0^n, \gamma_2*u_0^n$
${}^{\gamma}U_{\varphi}$	$1+u_0^2, \gamma_2$	$u_0^n, 1+u_0^n, \gamma_2*u_0^n, \gamma_2*(1+u_0^n)$
${}^{\gamma}S_{\varphi\psi}$	p_0^2, γ_2	$g\gamma_0+f\varphi_0+p\psi_0, \gamma_2*(g\gamma_0+f\varphi_0+p\psi_0)$
${}^{\psi}S_{\varphi}$	u_0^2, ψ_2	$u_0^n, \gamma_2*u_0^n, \psi_2$
${}^{\psi}U_{\varphi}$	$1+u_0, \psi_2$	$c+u_0^n, \gamma_2*(c+u_0^n), \psi_2(c\in\{0, 1\})$
${}^{\psi}L_{\psi}$	p_0^3, ψ_2	$p_0^{2k+1}, \psi_2*p_0^{2k+1}$
${}^{\psi}S_{\psi}$	p_0^2, ψ_2	$p_0^n, \psi_2*p_0^n$
${}^{\psi}L'_{\psi}$	$1+p_0^3, \psi_2$	$c+p_0^{2k+1}, \psi_2*(c+p_0^{2k+1}), c\in\{0, 1\}$
${}^{\psi}S'_{\psi}$	$1+p_0^2, \psi_2$	$1+p_0^{2k}, \psi_2*(1+p_0^{2k}), p_0^{2k+1}, \psi_2*p_0^{2k+1}$
${}^{\psi}U_{\psi}$	$p_0^2, 1+p_0^2, \psi_2$	$c+p_0^n, \psi_2*(c+p_0^n), c\in\{0, 1\}$
${}^{\delta}S_{\psi}$	$u_0^2, \gamma_2, \delta_2,$	$u_0^n, \gamma_2*u_0^n, \delta_2*u_0^n$
${}^{\psi\delta}S_{\psi}$	$u_0^2, \gamma_2, \delta_2, \psi_2$	$u_0^n, \gamma_2*u_0^n, \delta_2*u_0^n, \psi_2$
${}^{\bar{\psi}}S_{\varphi}$	$u_0^2, \bar{\psi}_2$	$u_0^n, \gamma_2*u_0^n, \delta_2*u_0^n, \psi_2, \bar{\psi}_2$
${}^{\bar{\psi}}U_{\varphi}$	$1+u_0^2, \bar{\psi}_2$	$c+u_0^n, \gamma_2*(c+u_0^n), \psi_2, \bar{\psi}_2, c\in\{0, 1\}$
${}^{\bar{\psi}}L_{\psi}$	$p_0^3, \bar{\psi}_2$	$p_0^{2k+1}, \psi_2*p_0^{2k+1}, \psi_2*p_0^{2k+1}$
${}^{\bar{\psi}}S_{\psi}$	$p_0^2, \bar{\psi}_2$	$p_0^n, \psi_2*p_0^n, \bar{\psi}_2*p_0^n$
${}^{\bar{\psi}}S'_{\psi}$	$1+p_0^2, \bar{\psi}_2$	$h*p_0^{2k+1}, h*(1+p_0^{2k}), h\in\{\psi_0, \psi_2, \bar{\psi}_2\}$

Клоны из таблицы 8 легко получаются из клонов, отмеченных на рисунке 11, с учетом состава клонов из таблицы 7.

В таблице 9 указаны все клоны, состоящие из существенно одноместных функций и содержащие функции из множества $\{\gamma_2, \delta_2, \psi_2, \bar{\psi}_2\}$.

Таблица 8

Клон	Базис	Содержащиеся функции
${}^{\gamma\varphi}S_{\varphi}$	$u_0^2, \gamma_2, \varphi_2$	$u_0^n, \gamma_2 * u_0^n, \varphi_2$
${}^{\gamma\varphi}U_{\varphi}$	$1+u_0^2, \gamma_2$	$c+u_0^n, \gamma_2 * (c+u_0^n), \varphi_2, c \in \{0, 1\}$
${}^{\psi\varphi}S_{\varphi}$	u_0^2, φ_2, ψ_2	$u_0^n, \gamma_2 * u_0^n, \varphi_2, \psi_2$
${}^{\psi\varphi}U_{\varphi}$	$1+u_0^2, \varphi_2, \psi_2$	$c+u_0^n, \gamma_2 * (c+u_0^n), \varphi_2, \psi_2$
${}^{\bar{\psi}\varphi}S_{\varphi}$	$u_0^2, \varphi_2, \bar{\psi}_2$	$u_0^n, \gamma_2 * u_0^n, \delta_2 * u_0^n, \varphi_2, \bar{\varphi}_2, \psi_2, \bar{\psi}_2$
${}^{\bar{\psi}\varphi}U_{\varphi}$	$1+u_0^2, \varphi_2, \bar{\psi}_2$	$c+u_0^n, \gamma_3 * (c+u_0^n), \varphi_2, \bar{\varphi}_2, \psi_2, \bar{\psi}_2$
${}^{2\gamma}S_{\varphi}$	c_2, u_0^2, γ_2	$c_2, u_0^n, \gamma_2 * u_0^n$
${}^{2\gamma}U_{\varphi}$	$c_2, 1+u_0^2, \gamma_2$	$c_2, c+u_0^n, \gamma_2 * (c+u_0^n), c \in \{0, 1\}$
${}^{2\psi}S_{\varphi}$	c_2, u_0^n, ψ_2	$c_2, u_0^n, \gamma_2 * u_0^n, \psi_2$
${}^{2\psi}U_{\varphi}$	$c_2, 1+u_0^2, \psi_2, c_2,$	$c+u_0^n, \gamma_2 * (c+u_0^n), \psi_2$ $c \in \{0, 1\}$
${}^{2\gamma\varphi}S_{\varphi}$	$c_2, u_0^2, \gamma_2, \varphi_2$	$c_2, u_0^n, \gamma_2 * u_0^n, \varphi_2$
${}^{2\psi\varphi}S_{\varphi}$	$c_2, u_0^2, \varphi_2, \psi_2$	$c_2, u_0^n, \gamma_2 * u_0^n, \varphi_2, \psi_2$

Таблица 9

Клон	Базис	Содержащиеся функции
${}^{\gamma}I_{00}^{0}$	γ_2	γ_2, c_0
${}^{\gamma}I_{00}^{0\varphi}$	φ_0, γ_2	φ_0, c_0, γ_2
${}^{2\gamma}I_{00}^{0}$	γ_2, c_2	c_0, c_2, γ_2
${}^{\gamma}I_{00}$	γ_2, c_1	c_0, c_1, c_2, γ_2
${}^{2\gamma}I_{00}^{0\varphi}$	φ_0, c_2, γ_2	$c_0, c_2, \varphi_0, \gamma_2$
${}^{\gamma}I_{00}^{0\varphi\psi}$	φ_0, γ_2	$c_0, \varphi_0, \psi_0, \gamma_2, \psi_2$
${}^{\gamma}I_{00}^{\varphi}$	c_1, φ_0, γ_2	$c_0, c_1, c_2, \varphi_0, \gamma_2$
${}^{\gamma}I_{00}^{\varphi\psi}$	c_1, ψ_0, γ_2	$c_0, c_1, c_2, \varphi_0, \psi_0, \gamma_2, \psi_2$
${}^{\gamma}I_{10}^{0\varphi}$	γ_0, γ_2	$c_0, \gamma_0, \varphi_0, \gamma_2, \varphi_2$
${}^{\gamma}I_{10}^{0\varphi\psi}$	$\gamma_0, \psi_0, \gamma_2$	$c_0, \gamma_0, \varphi_0, \gamma_2, \varphi_2, \psi_2$
${}^{\gamma}F_{1}^{\varphi}$	δ_0, γ_2	$c_0, c_1, c_2, \gamma_0, \delta_0, \varphi_0, \bar{\varphi}_0, \varphi_2, \bar{\varphi}_2, \gamma_2, \delta_2$
${}^{\gamma}F_{1}^{\varphi\psi}$	$\delta_0, \psi_0, \gamma_2$	$c_0, c_1, c_2, \gamma_0, \delta_0, \varphi_0, \bar{\varphi}_0, \psi_0, \bar{\psi}_0, \gamma_2, \delta_2, \varphi_2, \bar{\varphi}_2, \psi_2, \bar{\psi}_2$
${}^{\gamma}I_{10}$	c_1, γ_0, γ_2	$c_0, c_1, c_2, \gamma_0, \gamma_2, \varphi_2$
${}^{\gamma}I_{10}^{\varphi}$	$c_1, \gamma_0, \varphi_0, \gamma_2$	$c_0, c_1, c_2, \gamma_0, \varphi_0, \gamma_2, \varphi_2$
${}^{\gamma}I_{10}^{\varphi\psi}$	$c_1, \gamma_0, \psi_0, \gamma_2$	$c_0, c_1, c_2, \gamma_0, \varphi_0, \psi_0, \gamma_2, \varphi_2, \psi_2$
${}^{\gamma}F_{0}^{\varphi}$	$\bar{\varphi}_0, \gamma_2$	$c_0, c_1, c_2, \varphi_0, \bar{\varphi}_0, \gamma_2, \delta_2$
${}^{\gamma}F_{0}^{\varphi\psi}$	$\bar{\psi}_0, \gamma_2$	$c_0, c_1, c_2, \varphi_0, \psi_0, \bar{\varphi}_0, \bar{\psi}_0, \gamma_2, \delta_2, \psi_2, \bar{\psi}_2$
${}^{2\gamma}I_{00}^{0\varphi\psi}$	ψ_0, c_2, γ_2	$c_0, c_1, c_2, \varphi_0, \psi_0, \gamma_2, \psi_2$
${}^{\gamma 2\varphi}I_{00}^{0}$	c_2, γ_2, φ_2	$c_0, c_2, \gamma_2, \varphi_2$
${}^{\gamma 2\varphi}I_{00}^{0\varphi}$	$c_2, \varphi_0, \gamma_2, \varphi_2$	$c_0, c_2, \varphi_0, \gamma_2, \varphi_2$
${}^{\gamma}I_{10}^{0\varphi\psi}$	$\gamma_0, \psi_0, c_2, \gamma_2$	$c_0, c_1, c_2, \gamma_0, \varphi_0, \psi_0, \gamma_2, \varphi_2, \psi_2$
${}^{\delta}E_{\varphi}$	φ_0, δ_2	φ_0, δ_2
${}^{\delta}I_{00}^{0}$	c_0, δ_2	c_0, c_2, δ_2
${}^{\delta}I_{00}^{0\varphi}$	c_0, φ_0, δ_2	$c_0, c_2, \varphi_0, \delta_2$

Таблица 9 (продолжение)

Клон	Базик	Содержащиеся функции
${}^{\delta}I_{00}^{\varphi\psi}$	c_0, ψ_0, δ_2	$c_0, c_1, c_2, \varphi_0, \bar{\varphi}_0, \psi_0, \bar{\psi}_0, \gamma_2, \delta_2, \psi_2, \bar{\psi}_2$
${}^{\delta}I_{00}^{1}$	c_1, δ_2	c_0, c_1, c_2, δ_2
${}^{\delta}I_{00}^{1\varphi}$	c_1, φ_0, δ_2	$c_0, c_1, c_2, \varphi_0, \delta_2$
${}^{\delta}I_{01}^{\varphi}$	δ_0, δ_2	$c_0, c_1, c_2, \delta_0, \varphi_0, \delta_2, \varphi_2$
${}^{\gamma\varphi}I_{00}^{0}$	γ_2, φ_2	c_0, γ_2, φ_2
${}^{\gamma\varphi}I_{00}^{0\varphi}$	$\varphi_0, \gamma_2, \varphi_2$	$c_0, \varphi_0, \gamma_2, \varphi_2$
${}^{\gamma\varphi}I_{00}$	c_1, γ_2, φ_2	$c_0, c_1, c_2, \gamma_2, \varphi_2$
${}^{\gamma\varphi}I_{00}^{\varphi}$	$c_1, \varphi_0, \gamma_2, \varphi_2$	$c_0, c_1, c_2, \varphi_0, \gamma_2, \varphi_2$
${}^{\gamma\varphi}F_{0}^{\varphi}$	$\bar{\varphi}_0, \gamma, \varphi_2$	$c_0, c_1, c_2, \varphi_0, \bar{\varphi}_0, \gamma_2, \delta_2, \varphi_2$
${}^{\gamma\bar{\varphi}}I_{00}^{0}$	γ_2, φ_2	$c_0, c_2, \gamma_2, \delta_2, \varphi_2, \bar{\varphi}_2$
${}^{\gamma\bar{\varphi}}I_{00}^{0\varphi}$	$\varphi_0, \gamma_2, \bar{\varphi}_2$	$c_0, c_2, \varphi_0, \gamma_2, \delta_2, \varphi_2, \bar{\varphi}_2$
${}^{\gamma\bar{\varphi}}I_{00}$	$c_1, \gamma_2, \bar{\varphi}_2$	$c_0, c_1, c_2, \gamma_2, \delta_2, \varphi_2, \bar{\varphi}_2$
${}^{\gamma\bar{\varphi}}I_{00}^{\varphi}$	$c_1, \varphi_0, \gamma_2, \bar{\varphi}_2$	$c_0, c_1, c_2, \varphi_0, \gamma_2, \delta_2, \varphi_2, \bar{\varphi}_2$
${}^{\gamma\bar{\varphi}}F_{0}^{\varphi}$	$\bar{\varphi}_0, \gamma_2, \bar{\varphi}_2$	$c_0, c_1, c_2, \varphi_0, \bar{\varphi}_0, \gamma_2, \delta_2, \varphi_2, \bar{\varphi}_2$
${}^{\psi}E_{\psi}$	ψ_0, ψ_2	ψ_0, ψ_2
${}^{\psi}E_{\varphi\psi}$	φ_0, ψ_2	$c_0, \varphi_0, \gamma_2, \psi_2$
${}^{\psi}I_{00}^{0}$	c_0, ψ_2	c_0, ψ_2
${}^{\psi}I_{00}^{0\psi}$	c_0, ψ_0, ψ_2	c_0, ψ_0, ψ_2
${}^{\psi\varphi}I_{10}^{0\psi}$	γ_0, ψ_2	$c_0, \gamma_0, \psi_0, \varphi_2, \psi_2$
${}^{\psi}I_{00}^{1}$	c_1, ψ_2	c_1, c_2, ψ_2
${}^{\psi}I_{00}^{1\psi}$	c_1, φ_0, ψ_2	c_1, c_2, ψ_0, ψ_2
${}^{\psi}I_{00}$	c_0, c_1, ψ_2	c_0, c_1, c_2, ψ_2
${}^{\psi}I_{00}^{\psi}$	c_0, c_1, ψ_0, ψ_2	$c_0, c_1, c_2, \psi_0, \psi_2$
${}^{\psi\varphi}I_{10}^{\psi}$	c_1, γ_0, ψ_2	$c_0, c_1, c_2, \gamma_0, \psi_0, \varphi_2, \psi_2$
${}^{\psi\varphi}I_{10}^{\psi}$	c_1, γ_0, ψ_2	$c_0, c_1, c_2, \gamma_0, \psi_0, \varphi_2, \psi_2$
${}^{\psi}F_{00}^{\psi}$	$\bar{\psi}_0, \psi_2$	$\psi_0, \bar{\psi}_0, \psi_2, \bar{\psi}_2$
${}^{\psi}F_{0}^{\varphi}$	$\bar{\varphi}_0, \psi_2$	$c_0, c_1, c_2, \varphi_0, \bar{\varphi}_0, \gamma_2, \delta_2, \psi_2, \bar{\psi}_2$
${}^{\psi}F_{0}^{\psi}$	$c_0, \bar{\psi}_0, \psi_2$	$c_0, c_1, c_2, \psi_0, \bar{\psi}_0, \psi_2, \bar{\psi}_2$
${}^{\psi\varphi}I_{00}^{0}$	c_0, φ_2, ψ_2	c_0, φ_2, ψ_2
${}^{\psi\varphi}I_{00}^{0\varphi}$	$c_0, \varphi_0, \varphi_2, \psi_2$	$c_0, \varphi_0, \gamma_2, \varphi_2, \psi_2$
${}^{\psi\varphi}I_{10}^{0\psi}$	$\gamma_0, \varphi_2, \psi_2$	$c_0, \gamma_0, \psi_0, \varphi_2, \psi_2$
${}^{\psi\varphi}I_{00}$	c_1, φ_2, ψ_2	$c_0, c_1, c_2, \varphi_2, \psi_2$
${}^{\psi\varphi}I_{00}^{\varphi}$	$c_1, \varphi_0, \varphi_2, \psi_2$	$c_0, c_1, c_2, \varphi_0, \gamma_2, \varphi_2, \psi_2$
${}^{\psi\varphi}I_{00}^{\psi}$	$c_1, \psi_0, \varphi_2, \psi_2$	$c_0, c_1, c_2, \psi_0, \varphi_2, \psi_2$
${}^{\psi\varphi}I_{00}^{\varphi\psi}$	$c_1, \varphi_0, \psi_0, \varphi_2, \psi_2$	$c_0, c_1, c_2, \varphi_0, \psi_0, \gamma_2, \varphi_2, \psi_2$
${}^{\psi\varphi}F_{1}^{\psi}$	δ_0, ψ_2	$c_0, c_1, c_2, \gamma_0, \delta_0, \psi_0, \bar{\psi}_0, \varphi_2, \bar{\varphi}_2, \psi_2, \bar{\psi}_2$
${}^{\psi\varphi}F_{0}^{\varphi}$	$c_0, \bar{\varphi}_0, \varphi_2, \psi_2$	$c_0, c_1, c_2, \varphi_0, \bar{\varphi}_0, \gamma_2, \delta_2, \varphi_2, \psi_2$
${}^{\psi 2}I_{00}^{0}$	c_0, c_2, ψ_2	c_0, c_2, ψ_2
${}^{\psi 2}I_{00}^{1}$	c_1, c_2, ψ_2	c_1, c_2, ψ_2
${}^{\psi 2}I_{00}^{0\varphi}$	$c_0, c_2, \varphi_0; \psi_2$	$c_0, c_2, \varphi_0, \gamma_2, \psi_2$
${}^{\psi 2}I_{00}^{\psi}$	c_1, ψ_0, ψ_2	$c_0, c_1, c_2, \psi_0, \psi_2$
${}^{\psi 2}I_{00}^{\varphi}$	$c_0, c_1, \varphi_0, \varphi_2$	$c_0, c_1, c_2, \varphi_0, \gamma_2, \psi_2$
${}^{\psi 2}I_{10}^{\psi}$	c_2, γ_0, ψ_2	$c_0, c_1, c_2, \gamma_0, \psi_0, \varphi_2, \psi_2$
${}^{\bar{\psi}}I_{00}^{0}$	$c_0, \bar{\psi}_2$	$c_0, c_2, \psi_2, \bar{\psi}_2$
${}^{\bar{\psi}}I_{00}^{0\varphi}$	$\varphi_0, \bar{\psi}_2$	$c_0, c_2, \varphi_0, \gamma_2, \delta_2, \psi_2, \bar{\psi}_2$

Таблица 9 (продолжение)

Клон	Базик	Содержащиеся функции
${}^{\bar\Psi}I_{00}$	$c_1, \bar\psi_2$	$c_0, c_1, c_2, \psi_2, \bar\psi_2$
${}^{\bar\Psi}F_0^{\psi}$	$c_0, \psi_0, \bar\psi_2$	$c_0, c_1, c_2, \psi_0, \bar\psi_0, \psi_2, \bar\psi_2$
${}^{\bar\Psi}F_0^{\varphi\psi}$	$c_0, \varphi_0, \psi_0, \bar\psi_2$	$c_0, c_1, c_2, \varphi_0, \bar\varphi_0, \psi_0, \bar\psi_0, \gamma_2, \delta_2, \psi_2, \bar\psi_2$
${}^{\bar\Psi\bar\Phi}I_{00}^{0}$	$c_0, \bar\varphi_2, \bar\psi_2$	$c_0, c_2, \varphi_2, \bar\varphi_2, \psi_2, \bar\psi_2$
${}^{\bar\Psi\bar\Phi}I_{00}^{0\varphi}$	$\varphi_0, \bar\varphi_2, \bar\psi_2$	$c_0, c_2, \varphi_0, \gamma_2, \delta_2, \varphi_2, \bar\varphi_2, \psi_2, \bar\psi_2$
${}^{\bar\Psi\bar\Phi}I_{00}$	$c_1, \bar\varphi_2, \bar\psi_2$	$c_0, c_1, c_2, \varphi_2, \bar\varphi_2, \psi_2, \bar\psi_2$
${}^{\bar\Psi\bar\Phi}I_{00}^{\varphi}$	$c_1, \varphi_0, \bar\varphi_2, \bar\psi_2$	$c_0, c_1, c_2, \varphi_0, \gamma_2, \delta_2, \varphi_2, \bar\varphi_2, \psi_2, \bar\psi_2$
${}^{\bar\Phi}F_0^{\varphi}$	$\bar\varphi_0, \bar\varphi_2, \bar\psi_2$	$c_0, c_1, c_2, \varphi_0, \bar\varphi_0, \gamma_2, \delta_2, \varphi_2, \bar\varphi_2, \psi_2, \bar\psi_2$

5. Подклоны клона $Z_0 \cup Z_2$, порождаемые с помощью многоместных функций из Z_2

Теперь мы рассмотрим возможность порождения новых клонов путем добавления к функциям из подклонов клона Z_0 существенно многоместных функций из Z_2. Начать естественно с более простого случая — с тех подклонов клона Z_0, которые содержат только существенно одноместные функции. Легко однако заметить, что мы будем получать лишь подклоны, изоморфные уже описанным. Действительно, пусть клон K порождается унарными функциями $\varrho_1, \ldots, \varrho_s$ из Z_0 и функцией $\varrho \in Z_2$. Рассмотрим функции, λ_0-двойственные указанным. Функции $\varrho_1^{\lambda_0}, \ldots, \varrho_s^{\lambda_0}$ принадлежат Z_2, ϱ принадлежит Z_0, и совместно они порождают клон, λ_0-двойственный клону K. Все клоны такого типа уже найдены выше.

Видим, что нам осталось описать все клоны K, порождаемые объединением базисов клонов $K_1 \leqq Z_0$ и $K_2 \leqq Z_2$, при условии, что эти клоны содержат существенно многоместные функции. Комбинации базисов, очевидным образом не порождающие новые клоны, рассматриваться не будут.

Пусть K_2 порождается либо функцией $u_2^3 = \varphi_2 \dotplus \varphi_2 \dotplus \varphi_2 = \gamma_2 * (\gamma_0 + \gamma_0 + \gamma_0)$, либо функцией $\bar u_2^3 = \gamma_2 * (1 + \gamma_0 + \gamma_0 + \gamma_0)$. Так как $\gamma_2 * \varrho_0 = c_0$, $\gamma_0 * \gamma_2 = \varphi_0$, для $\varrho_0 \in Z_0$, то $K \cap Z_2 = \{u_2^n,\ n = 0, 1, 2, \ldots\}$ в первом случае и $K \cap Z_2 = \{u_2^n, \bar u_2^n,\ n = 0, 1, 2, \ldots\}$ во втором. В таблице 10 приведены возможные варианты базиса клона K. Все построения при этом основываются на результатах суперпозиции $\varrho * \gamma_2$, $\varrho \in K_1$. Ввиду тривиальности мы их опустим. Вместо функций u_2^3, $\bar u_2^3$ в базисы включены функции $u_2^2, \bar u_2^2$, поскольку на результат такая замена не влияет.

Все клоны K, содержащие функцию $\bar u_2^2$, содержат и функцию u_2^2, поэтому случай, когда K_2 порождается функциями u_2^2, $\bar u_2^2$ совместно, не представляет интереса.

Пусть теперь K_2 порождается функцией $\varphi_2 \dotplus \gamma_2 = \gamma_2 * (\gamma_0 + \varphi_0)$. Из таблицы 4 видно, что K_2 состоит из функций $\varphi_2 \dotplus f_2^n = \gamma_2 * (\gamma_0 + u_0^n)$, $\psi_2 \dotplus f_2^n = \gamma_2 * (\psi_0 + u_0^n)$. Поскольку $\varphi_0 * \varrho = \varrho$ при $\varrho \in Z_0$, то если K содержит функцию γ_0 или ψ_0, то K содержит все функции $\gamma_2 * (g\gamma_0 + f\varphi_0 + p\psi_0)$. Если к тому же в K есть функция вида $1 + g_1\gamma_0 + f_1\varphi_0 + p_1\psi_0$, то $K > Z_2$. Если $\gamma_0 \in K$ или $\psi_0 \in K$, то из $\gamma_2 * \varrho \in K$ следует $\varrho \in K$. Учитывая сказанное и пользуясь данными таблиц 2, 6, 7, при-

Таблица 10

Клон	Базис	Вариант базиса	Содержащиеся функции
${}^{u}I^{0}_{\infty 0}$	$f_0^2,\ u_2^2$		$f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{0\varphi}_{\infty 0}$	$\varphi_0,\ f_0^2,\ u_2^2$		$\varphi_0,\ f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{0\psi}_{\infty 0}$	$\psi_0,\ f_0^2,\ u_2^2$		$\psi_0,\ f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{0\varphi\psi}_{\infty 0}$	$\varphi_0,\ \psi_0,\ f_0^2,\ u_2^2$		$\varphi_0,\ \psi_0,\ f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{1}_{0\infty}$	$1+f_0^2,\ u_2^2$		$1+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{1\varphi}_{0\infty}$	$\varphi_0,\ 1+f_0^2,\ u_2^2$		$\varphi_0,\ 1+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{1\psi}_{0\infty}$	$\psi_0,\ 1+f_0^2,\ u_2^2$		$\psi_0,\ 1+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{1\varphi\psi}_{0\infty}$	$\varphi_0,\ \psi_0,\ 1+f_0^2,\ u_2^2$		$\varphi_0,\ \psi_0,\ 1+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I_{\infty\infty}$	$f_0^2,\ 1+f_0^2,\ u_2^2$		$f_0^n,\ 1+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{\varphi}_{\infty\infty}$	$\varphi_0,\ f_0^3,\ 1+f_0^2,\ u_2^2$		$\varphi_0,\ f_0^n,\ 1+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{\psi}_{\infty\infty}$	$\psi_0,\ f_0^2,\ 1+f_0^2,\ u_2^2$		$\psi_0,\ f_0^n,\ 1+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}I^{\varphi\psi}_{\infty\infty}$	$\varphi_0,\ \psi_0,\ f_0^2,\ 1+f_0^2,\ u_2^2$		$\varphi_0,\ \psi_0,\ 1+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}S_{\varphi}$	$u_3^3,\ u_2^2$	$u_0^2,\ u_2^2$	$u_0^n,\ \gamma_2 * f_0^n$
${}^{u}U_{\varphi}$	$1+u_0^3,\ u_2^2$	$1+u_0^2,\ u_2^2$	$u_0^n,\ 1+u_0^n,\ \gamma_2 * f_0^n$
${}^{u}H_{\varphi}$	$\varphi_0+\gamma_0,\ u_2^2$	$\psi_0+\gamma_0,\ u_2^2$	$f_0^n,\ \varphi_0+f_0^n,\ \psi_0+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}S_{\varphi\psi}$	$p_0^3,\ u_2^2$		$g\gamma_0+f\varphi_0+p\psi_0,\ \gamma_2 * f_0^n$
${}^{u}G$	$1+\varphi_0+\gamma_0,\ u_2^2$		$c+f_0^n,\ c+\varphi_0+f_0^n,\ c+\psi_0+f_0^n,\ \gamma_2 * f_0^n\ (c\in\{0,1\})$
${}^{u}Z_0$	$1+\varphi_0+\psi_0+\psi_0,\ u_2^2$	$1+p_0^2,\ u_2^2$	$c+g\gamma_0+f\varphi_0+p\psi_0,\ \gamma_2 * f_0^n$
${}^{u}H_{\psi}$	$\psi_0+\gamma_0,\ u_2^2$		$1+f_0^n,\ \varphi_0+f_0^n,\ \psi_0+f_0^n,\ \gamma_2 * f_0^n$
${}^{u}H_{\varphi\psi}$	$\varphi_0+\gamma_0,\ c_1$		$c+f_0^n,\ \varphi_0+f_0^n,\ \psi_0+f_0^n,\ \gamma_2 * f_0^n$
${}^{\bar u}I_{\infty\infty}$	$f_0^2,\ \bar u_2^2$		$c+f_0^n,\ \gamma_2 * (c+f_0^n)(c\in\{0,1\})$
${}^{\bar u}I^{\varphi}_{\infty\infty}$	$\varphi_0,\ f_0^2,\ \bar u_2^2$		$\varphi_0,\ c+f_0^n,\ \gamma_2 * (c+f_0^n)$
${}^{\bar u}I^{\psi}_{\infty\infty}$	$\psi_0,\ f_0^2,\ \bar u_2^2$		$\psi_0,\ c+f_0^n,\ \gamma_2 * (c+f_0^n)$
${}^{\bar u}I^{\varphi\psi}_{\infty\infty}$	$\varphi_0,\ \psi_0,\ f_0^2,\ \bar u_2^2$		$\varphi_0,\ \psi_0,\ c+f_0^n,\ \gamma_2 * (c+f_0^n)$
${}^{\bar u}S_{\varphi}$	$u_0^3,\ \bar u_2^2,$	$u_0^2,\ \bar u_2^2$	$u_0^n,\ \gamma_2 * (c+f_0^n)$
${}^{\bar u}U_{\varphi}$	$1+u_0^3,\ \bar u_2^2$		$c+u_0^n,\ \gamma_2 * (c+f_0^n)$
${}^{\bar u}H_{\varphi\psi}$	$\varphi_0+\gamma_0,\ \bar u_2^2$	$\varphi_0+\gamma_0,\ u_2^2$	$c+f_0^n,\ \varphi_0+f_0^n,\ \psi_0+f_0^n,\ \gamma_2 * (c+f_0^n)$
${}^{\bar u}Z_0$	$p_0^3,\ \bar u_2^2$		$c+g\gamma_0+f\varphi_0+p\psi_0,\ \gamma_2 * (c+f_0^n)$
${}^{\bar u}G$	$1+\gamma_0+\varphi_0,\ \bar u_2^2$		$c+f_0^n,\ c+\varphi_0+f_0^n,\ c+\psi_0+f_0^n,\ \gamma_2 * (c+f_0^n)$.

ходим к выводу, что с помощью функции $\varphi_2 \dotplus \gamma_2$ порождается два новых клона: ${}^{\varphi+\gamma}S_{\varphi}$, ${}^{\varphi+\gamma}U_{\varphi}$ (таблица 11).

Если K_2 порождается функциями u_2^3 и p_2^3, то K_2 содержит функции

$$g\gamma_2 \dotplus f\varphi_2 \dotplus p\psi_2 = \gamma_2 * (g\varphi_0 + f\gamma_0 + p\psi_0), \quad f+p \doteq 2k+1.$$

Так как $K_1 \neq \emptyset$, то K содержит все функции $\gamma_2 * (g\varphi_0 + f\gamma_0 + p\psi_0)$. Получаем клоны ${}^{u,p}S_{\varphi}$, ${}^{u,p}U_{\varphi}$ (таблица 11). Очевидно, в случае, когда K_2 порождается функцией $\varphi_2 \dotplus \psi_2$ новых клонов не возникает.

Пусть K_2 порождается функцией $\gamma_2 \dotplus \gamma_2 = \gamma_2 * u_0^2$. В этом случае K_2 содержит только 3 существенно разные функции: c_0, $\gamma_2 + \varphi_0$, $\gamma_2 * u_0^2$. Если γ_0, δ_0, $\varphi_0 + \gamma_0$ или $\varphi_0 + \delta_0$ принадлежат K, то K содержит $\gamma_2 * (\varphi_0 + \gamma_0)$, и мы опять возвращаемся к рассмотренному случаю. Аналогично, из

$$\{\varphi_0 + \psi_0,\ p_0^3,\ \psi_0 + \gamma_0,\ 1 + \varphi_0 + \psi_0,\ 1 + p_0^3,\ 1 + \psi_0 + \gamma_0\} \cap K \neq \emptyset$$

следует принадлежность к K функций $\gamma_2*(g\gamma_0+f\varphi_0+p\psi_0)$, $g\gamma_0+f\varphi_0+p\psi_0$. Интерес могут представлять лишь клоны K, для которых $K\cap Z_2=\{\gamma_2*u_0^n\}$ или $K\cap Z_2=\{\gamma_2*(c+u_0^n)\}$, однако они уже указаны в таблице 7. Очевидно, те же клоны получим, взяв за основу функцию f_2^n. Если клон K_2 порождается функцией $\gamma_2\dotplus\delta_2=\gamma_2*(1+u_0^2)$, получаем клон ${}^{\gamma+\delta}S_\varphi$. Добавление к его базису функций $\varphi_2, \psi_2, \gamma_2\dotplus\gamma_2$ дает клоны

$${}^{\varphi,\gamma+\delta}S_\varphi,\quad {}^{\psi,\gamma+\delta}S_\varphi,\quad {}^{\varphi,\psi,\gamma+\delta}S_\varphi,\quad {}^{\varphi,\gamma+\gamma,\gamma+\delta}S_\varphi,\quad {}^{\varphi,\psi,\gamma+\gamma,\gamma+\delta}S_\varphi$$

(таблица 11).

Пусть клон K_2 порождается функциями $\varphi_2\dotplus\gamma_2, c_0$, тогда в нем содержатся также функции

$$f_2^n=\gamma_2*u_0^n,\quad \varphi_2\dotplus f_2^n=\gamma_2*(\gamma_0+u_0^n),\quad \psi_2\dotplus f_2^n=\gamma_2*(\psi_0+u_0^n).$$

Видим, что K содержит все функции одного из клонов ${}^{\varphi+\gamma}S_\varphi$, ${}^{\varphi+\gamma}U_\varphi$, ${}^{\gamma}S_{\varphi\psi}$, и если $K\neq Z_0\cup Z_2$, то совпадает с ним, поскольку каждый из этих клонов содержит c_0.

Если базис клона K_2 образован функцией $\varphi_2\dotplus\delta_2=\gamma_2*(1+\varphi_0+\gamma_0)$, то K_2 состоит из функций $\gamma_2*(c+\gamma_0+u_0^n)$, $\gamma_2*(1+\varphi_0+\gamma_0)$. При $K_1\in\{S_\varphi, L_\varphi, L'_\varphi, S'_\varphi, U_\varphi\}$ новых клонов не получаем. Если K_1 содержит γ_0 или ψ_0, то $K>Z_2$, более того, K совпадает с $Z_0\cup Z_2$.

Если $\varphi_2\dotplus\gamma_2, c_0, c_2$ — базис клона K_2, то K_2 состоит из функций $\gamma_2*(c+u_0^n)$, $\gamma_2*(\gamma_0+u_0^n)$, $\gamma_2*(\psi_0+u_0^n)$. При $K_1=S_\varphi$ получаем еще один клон ${}^{2,\varphi+\gamma}S_\varphi$ (таблица 11), в остальных случаях новых клонов не возникает.

Опираясь на проведенные рассуждения, легко убедиться в том, что новых клонов не возникает и в том случае, когда K_2 порождается одним из базисов $\bar\varphi_2\dotplus\psi_2\dotplus\psi_2$; $\varphi_2\dotplus\gamma_2$, $\bar\varphi_2$, c_0.

Еще несколько вариантов базисов клона K_2 дает нам таблица 5. В первых пяти случаях клон K либо содержит все функции клона ${}^{\gamma}S_{\varphi\psi}$, либо совпадает с одним из клонов ${}^{\psi}L_\psi$, ${}^{\psi}S_\psi$, ${}^{\psi}L'_\psi$, ${}^{\psi}S'_\psi$, ${}^{\psi}U_\psi$. Базис $\psi_2\dotplus\gamma_2$ дает клон ${}^{2,\psi+\gamma}S_\varphi$ (таблица 11), базис $\bar\varphi_2\dotplus\psi_2$ новых клонов не дает.

Таблица 11

Клон	Базис	Содержащиеся функции
${}^{\varphi+\gamma}S_\varphi$	u_0^2, $\varphi_2\dotplus\gamma_2$	u_0^n, $\gamma_2*u_0^n$, $\gamma_2*(\gamma_0+u_0^n)$, $\gamma_2*(\psi_0+u_0^n)$
${}^{\varphi+\gamma}U_\varphi$	$1+u_0^2$, $\varphi_2\dotplus\gamma_2$	$c+u_0^n$, $\gamma_2*(c+u_0^n)$, $\gamma_2*(c+\gamma_0+u_0^n)$, $\gamma_2*(c+\psi_0+u_0^n)$
${}^{u,p}S_\varphi$	u_0^2, u_3^2, p_2^3	u_0^n, $\gamma_2*(g\gamma_0+f\varphi_0+p\psi_0)$
${}^{u,p}U_\varphi$	$1+u_0^2$, u_2^3, p_2^3	$c+u_0^n$, $\gamma_2*(c+g\gamma_0+f\varphi_0+p\psi_0)$
${}^{\gamma+\delta}S_\varphi$	u_0^2, $\gamma_2\dotplus\delta_2$	u_0^n, $\gamma_2*(1+u_0^n)$
${}^{\varphi,\gamma+\delta}S_\varphi$	u_0^2, $\gamma_2\dotplus\delta_2$, φ_2	u_0^n, $\gamma_2*(1+u_0^n)$, φ_2
${}^{\psi,\gamma+\delta}S_\varphi$	u_0^2, $\gamma_2\dotplus\delta_2$, ψ_2	u_0^n, $\gamma_2*(1+u_0^n)$, ψ_2
${}^{\varphi,\psi,\gamma+\delta}S_\varphi$	u_0^2, $\gamma_2\dotplus\delta_2$, φ_2, ψ_2	u_0^n, $\gamma_2*(1+u_0^n)$, φ_2, ψ_2
${}^{\varphi,\gamma+\gamma,\gamma+\delta}S_\varphi$	u_0^2, $\gamma_2\dotplus\gamma_2$, $\gamma_2\dotplus\delta_2$, φ_2	u_0^n, $\gamma_2*(1+u_0^n)$, φ_2
${}^{\varphi,\psi,\gamma+\gamma,\gamma+\delta}S_\varphi$	u_0^2, $\gamma_2\dotplus\gamma_2$, $\gamma_2\dotplus\delta_2$, φ_2, ψ_2	u_0^n, $\gamma_2*(1+u_0^n)$, φ_2, ψ_2
${}^{\varphi+\delta}S_\varphi$	u_0^2, $\varphi_2\dotplus\delta_2$	u_0^n, $\gamma_2*(c+u_0^n)$, $\gamma_2*(c+\gamma_0+u_0^n)$, $\gamma_2*(c+\psi_0+u_0^n)$
${}^{2,\varphi+\gamma}S_\varphi$	u_0^2, $\varphi_2\dotplus\gamma_2$, c_2	u_0^n, $\gamma_2*(c+u_0^n)$, $\gamma_2*(\gamma_0+u_0^n)$, $\gamma_2*(\psi_0+u_0^n)$
${}^{2,\psi+\gamma}S_\varphi$	u_0^2, $\psi_2\dotplus\gamma_2$, c_2	u_0^n, $\gamma_2*(1+u_0^n)$, $\gamma_2*(\gamma_0+u_0^n)$, $\gamma_2*(\psi_0+u_0^n)$

Summary

The clone B consist of the essentially unary operations on $A=\{0, 1, 2\}$ and the operations of the form

$$f_0(f_1(x_1)+\dots+f_n(x_n)),$$

where f_0: $\{0, 1\}\to A$, $f_1, \dots, f_n$: $A\to\{0, 1\}$ and the addition is modulo 2. In the present paper we describe the lattice of subclones of the clone $Z_0\cup Z_2$ consisting of the projections and those members of clone B which take values 0 and 1 or 0 and 2 only.

Литература

[1] Bagyinszki J., Demetrovics J., The structure of linear classes in prime-valued logics (in Hungarian). — MTA SZTAKI Közl., 1976, N 16, p. 25—52.
[2] Bagyinszki J., Demetrovics J., The structure of linear classes in prime-valued logics. — Discrete Mathematics, Banach Center publications, Warsaw, 1982, p. 105—123.
[3] Berman J., McKenzie R., Clones satisfying the term condition. — Discrete Math., 1984, 52, p. 7—29.
[4] Бурле Г. А., Классы k-значных логик, содержащие все функции одной переменной. — Дискретный анализ, 1967, № 10, с. 3—7.
[5] Деметрович Я., Мальцев И. А., Подклоны клона Бурле на трехэлементном множестве. — Седьмая всесоюзная конференция по мат. логике, Новосибирск, 1984, с. 56.
[6] Деметрович Я., Мальцев И. А., О существенно минимальных ТС-клонах на трехэлементном множестве. — MTA SZTAKI Közl., 1984, № 32, с. 115—151.
[7] Деметрович Я., Мальцев И. А., ТУ-клоны на трехэлементном множестве. — ХУШ Всесоюзная алгебраическая конференция, тезисы сообщений, часть первая, Кишинев, 1985, с. 162.
[8] Lampe W. A., Abstract 77T—A120. — Notices Amer. Math. Soc. 1977, 24, A—371.
[9] А. И. Мальцев, Итеративные алгебры Поста. — Новосибирск, 1976.
[10] Мальцев И. А., Некоторые свойства клеточных подалгебр алгебр Поста и их основных клеток. — Алгебра и логика, 1972, № 11, с. 571—587.
[11] Мальцев И. А., Некоторые свойства клеток алгебр Поста. — Дискретный анализ, 1973, № 23, с. 24—31.
[12] Taylor W., Some applications of the term condition. — Algebra Universalis, 1982, 14, p. 11—24.
[13] Яблонский С. В., Габрилов Г. И., Кудрявцев В. Б., Функции алгебры логики и классы Поста. — М., Наука, 1966.
[14] Lau D., Klassen quasilinearen Functionen von P_3. — Rostocker Math. Koll., 1985, 28, 33—45.
[15] Марченков С. С., О замкнутых классах квазилинейных функций в P_3. — Препринт № 17 ИПМ АН СССР, 1986.

Received 9 February, 1988

Asymptotic analysis of some controlled finite-source queueing systems

By V. V. ANISIMOV and J. SZTRIK

KIEV STATE UNIVERSITY
FACULTY OF CYBERNETICS
VLADIMIRSKAJA 64.
KIEV—17, USSR

UNIVERSITY OF DEBRECEN
DEPARTMENT OF MATHEMATICS
DEBRECEN, PF. 12., 4010
HUNGARY

1. Introduction

Stochastic processes are powerful tools for the investigation of behavior of complex systems. Results of queueing theory are effectively used in solving problems encountered in practical applications. Different methods and approaches have been developed in order to that the involved models should be mathematically tractable. Since there exists an overwhelming body of literature devoted to the study of queues the interested reader is referred to, among others, Franken et. al. [4], Gnedenko and Kovalenko [5], Gnedenko [6], Gnedenko and König [7], Jaiswal [8], Kleinrock [9], Koroljuk and Turbin [11], Kovalenko [12], König and Stoyan [13], Lavenberg [14], Lifsic and Malc [15], Takács [16], White et. al. [17].

It is also well-known, that a great majority of problems can be treated by the help of Semi-Markov Processes (SMP). In many situations the distribution of time until the SMP gets out of a subset of its state space is of great practical importance. Recently, however, due to the rapid development of technical devices there are cases where the exit from a given subset is a "rare" event, that is, it occurs with a small probability. Thus, it is natural to investigate the asymptotic behaviour of sojourn time in a given subset, provided that the propability of exit from it tends to zero.

The purpose of the present paper is three-fold. On the one hand, without proofs we give a brief survey of preliminary results mainly due to Koroljuk and Anisimov. On the other hand, we deal with an asymptotic analysis of some controlled finite-source queueing systems under the assumption of "fast" service. We show that the time to the first system failure converges in distribution, under appropriate norming, to an exponentially distributed random variable. Finally, applications of these models in the field of reliability theory and computer performance are considered.

2. Preliminary results

Let us begin with the non-asymptotic case (see Koroljuk [11]).

(i) Let $(\xi(t),\ t\geqq 0)$ be a Semi-Markov Process with state space $\{0, 1, ..., r\}$ given by the embedded Markov chain $(X_n,\ n\geqq 0)$ and by transition matrix $\|p(i,j)\|$, $i,j=\overline{0,r}$. Furthermore, let $\tau(i,j)$ be mutually independent random variables denoting the time spent in state i, given that the next state is j, $i,j=\overline{0,r}$.

Let $\Omega(k)$ denote the sojourn time of $\xi(t)$ in subset $\{1, ..., r\}$ started in state k, that is

$$\Omega(k) = \inf\{t\colon t > 0,\ \xi(t) = 0/\xi(0) = k,\ k \neq 0\}.$$

For $\Omega(k)$ we have the following stochastic relations

$$\Omega(k) = \begin{cases} \tau(k,0) & \text{with probability (w.p.)} \quad p(k,0), \\ \tau(k,j)+\Omega(j) & \text{w.p.} \quad p(k,j), \quad j=\overline{1,r}. \end{cases} \tag{1}$$

Let us introduce some notations:

$$\varphi(u,k,j) = \mathbf{E}\exp\big(iu\tau(k,j)\big), \quad \psi(u,k) = \mathbf{E}\exp\big(iu\Omega(k)\big),$$

$$\underline{\varphi}(u) = \begin{pmatrix} p(1,0)\varphi(u,1,0) \\ \vdots \\ p(r,0)\varphi(u,r,0) \end{pmatrix}, \quad \underline{\psi}(u) = \begin{pmatrix} \psi(u,1) \\ \vdots \\ \psi(u,r) \end{pmatrix},$$

$$\Phi(u) = \|p(k,j)\varphi(u,k,j)\|, \quad k,j=\overline{1,r}.$$

When passing in (1) to characteristic functions we obtain

$$\underline{\psi}(u) = \Phi\underline{\psi}(u)+\underline{\varphi}(u). \tag{2}$$

Supposing that for any $j\in\{1, ..., r\}$ there exists a sequence of transitions with positive probabilities leading to $\{0\}$, that is $\{1, ..., r\}$ is not closed and does not contain any closed subset, from (2) we get

$$\underline{\psi}(u) = \big(E-\Phi(u)\big)^{-1}\underline{\varphi}(u).$$

In particular, for the mean sojourn times we have

$$\underline{M} = (E-P)^{-1}\underline{m},$$

where $\underline{M}$ and $\underline{m}$ are column vectors with components $\mathbf{E}\Omega(k)$, $m_k=\sum_{j=0}^{r} p(k,j)\mathbf{E}\tau(k,j)$ respectively, $k=\overline{1,r}$.

(ii) Suppose that $X_n=X_n(\varepsilon)$, $p(k,j)=p_\varepsilon(k,j)$ and $\tau(k,j)=\tau_\varepsilon(k,j)$, that is, $(\xi(t),\ t\geqq 0)$ depends on some small parameter ε, such that $p_\varepsilon(k,0)\to 0$ as $\varepsilon\to 0$. Therefore, it is natural to investigate the limit distribution of $\Omega(k)=\Omega_\varepsilon(k)$ as $\varepsilon\to 0$.

Assume that the following conditions are satisfied:

1. $p_\varepsilon(k,j)\to p_0(k,j)$, $k,j=\overline{1,r}$ and matrix $P_0=\|p_0(k,j)\|$, $k,j=\overline{1,r}$ corresponds to a single essential class.

2. $p_\varepsilon(k,0)/\varepsilon \to b_k < \infty$, $k=\overline{1,r}$, $\Sigma b_k \neq 0$.

3. There exists a normalizing factor β_ε such that

a) $\varphi_\varepsilon(\beta_\varepsilon u, k, j) = 1+\varepsilon a_{kj}(u)+o(\varepsilon), \quad k, j = \overline{1, r},$

b) $\varphi_\varepsilon(\beta_\varepsilon u, k, 0) \to \varrho_k(u), \quad k = \overline{1, r}.$

Theorem 1 (Koroljuk [10], [11]). If conditions (1)—(3) are satisfied, then independently of the initial state $j; j=\overline{1, r}$ the distribution of $\beta_\varepsilon \Omega_\varepsilon(j)$ converges weakly to a distribution with characteristic function

$$\Big(\sum_{k=1}^{r} \pi_k b_k \varrho_k(u)\Big)\Big/\Big(\sum_{i=1}^{r} \pi_i b_i - \sum_{i,j=1}^{r} \pi_i p_0(i,j) a_{ij}(u)\Big),$$

where $\{\pi_k,\ k=\overline{1, r}\}$ is the stationary distribution for the chain with matrix P_0.

Corollary 1. If the random variables $\tau_\varepsilon(k,j)$ do not depend on ε, that is $\tau_\varepsilon(k,j)=\tau_0(k,j)$, $k,j=\overline{0, r}$ and $\mathbf{E}\tau_0(k,j)=m_{kj}<\infty$, furthermore conditions (1), (2) are satisfied, then for any $j, j=\overline{1, r}$ we have

$$\mathbf{P}\{\varepsilon\Omega_\varepsilon(j) < x\} \to 1-\exp\left\{-\frac{b}{m}\,x\right\}, \quad x > 0$$

where

$$m = \sum_{k=1}^{r} \pi_k p_0(k,j) m_{kj}, \quad b = \sum_{k=1}^{r} \pi_k b_k.$$

(iii) Sometimes, however, there are practical situations (for example systems with "fast" service, or highly reliable systems) when the set $\{1, ..., r\}$ in the limit may split into several essential classes and, possibly, inessential states. To assert the correspondig theorem we need the notion of s-set, introduced by Anisimov (see [1]).

Let $(X_\varepsilon(k), k\geqq 0)$ be a Markov chain with state space $\{0, 1, ..., r\}$ and let $\|p_\varepsilon(i,j)\|$ denote its transition matrix, $i,j=\overline{0, r}$.

Furthermore, let $\langle\alpha\rangle$ be a subset from $\{1, ..., r\}$. Set

$$V_\varepsilon(i, \langle\alpha\rangle) = \min\{k\colon k > 0, X_\varepsilon(k)\notin\langle\alpha\rangle / X_\varepsilon(0) = i\in\langle\alpha\rangle\},$$

$q_\varepsilon(i,j,\langle\alpha\rangle)=\mathbf{P}\{X_\varepsilon(l)=j,$ for at least one l, $l<V_\varepsilon(i,\langle\alpha\rangle\}$ i.e. $q_\varepsilon(i,j,\langle\alpha\rangle)$ is the probability of a visit to j up the time when the chain exits from $\langle\alpha\rangle$, given that the initial state was i, $i,j\in\langle\alpha\rangle$.

Definition. A set of states

$$\langle\alpha\rangle = \{i_1, ..., i_l\}$$

is called an s-set (communicating set) if for any $i,j\in\langle\alpha\rangle$ $q_\varepsilon(i,j,\langle\alpha\rangle)\to 1$ as $\varepsilon\to 0$.

Practically, it means that initiated from any state the chain visits each state asymptotically infinitely many times before leaving. (The simplest example for an s-set is a set which in the limit forms a simple essential class.)

Let $(\xi_\varepsilon(t), t\geqq 0)$ be a SMP with state space $\{0, 1, ..., r\}$ given by the embedded Markov chain $(X_\varepsilon(n), n\geqq 0)$, the transition matrix $\|p_\varepsilon(k,j)\|$, $k,j=\overline{0, r}$ and the random variables $\tau_\varepsilon(k,j)$. Assume that the subset $\{1, ..., r\}$ forms an s-set.

Set

$$g_\varepsilon = \sum_{k=1}^{r} \pi_k p_\varepsilon(k, 0)$$

where $\{\pi_k,\ k=\overline{1, r}\}$ is the stationary distribution for the chain with transition matrix

$$\|p_\varepsilon(i,j)/(1-p_\varepsilon(i,0)),\quad i,j=\overline{1,r}.$$

Furthermore, suppose that

$$\pi_\varepsilon(k)p_\varepsilon(k,0)/g_\varepsilon \to b_k,\quad k=\overline{1,r},$$

and there exists a normalizing factor β_ε such that

a) $\varphi_\varepsilon(\beta_\varepsilon u, k, j) = 1+g_\varepsilon a_{kj}(u)+o(g_\varepsilon),\quad k,j=\overline{1,r},$

b) $\varphi_\varepsilon(\beta_\varepsilon u, k, 0) \to \varrho_k(u),\quad k=\overline{1,r}.$

Theorem 2 (Anisimov [2], [3]). If the above conditions are satisfied, then independently of the initial state $j, j=\overline{1,r}$ the distribution of $\beta_\varepsilon \Omega_\varepsilon(j)$ converges weakly to a distribution with characteristic function

$$\Big(\sum_{k=1}^{r} b_k\varrho_k(u)\Big)\Big/\Big(1-\sum_{k,j=1}^{r}\pi_0(k)p_0(k,j)a_{kj}(u)\Big),$$

where

$$\pi_0(k)=\lim_{\varepsilon\to 0}\pi_\varepsilon(k),\quad p_0(k,j)=\lim_{\varepsilon\to 0}p_\varepsilon(k,j).\quad k,j=\overline{1,r}.$$

The most crucial part of applying Theorem 2 to particular situations is finding the normalizing factor β_ε.

In the following an example is given on which our further considerations are based.

Example (see Anisimov et. al. [3] pp. 151).

Let $(X_\varepsilon(k),\ k\geqq 0)$ be a Markov chain with state space

$$E=\{(i,q),\ i=\overline{1,r},\ q=\overline{0,m+1}\}$$

defined by the transition matrix $\|p_\varepsilon[(i,q),(j,z)]\|$ satisfying the following conditions:

1. $p_\varepsilon[(i,0),(j,0)] \to p_{ij},\quad i,j=\overline{1,r},$

and the matrix $\|p_{ij}\|,\ i,j=\overline{1,r}$ is irreducible,

2. $p_\varepsilon[(i,q),(j,q+1)] = \varepsilon\alpha_{ij}^{(q)}+o(\varepsilon),\quad i,j=\overline{1,r},\quad q=\overline{0,m},$
3. $p_\varepsilon[(i,q),(j,q)] \to 0,\quad i,j=\overline{1,r},\quad q\geqq 1,$
4. $p_\varepsilon[(i,q),(j,z)] \equiv 0,\quad i,j=\overline{1,r},\quad |z-q|\geqq 2.$

In the sequel the set of states $\{(i,q),\ i=\overline{1,r}\}$ is called the q-th level of the chain, $q=\overline{0,m+1}$.

It is easy to see that conditions (1)—(4) have the following meaning. Level 0 in the limit forms an essential class, the transition probability from the q-th level to the $q+1$-th level is of order ε, on the q-th level the transition probability tends to zero, finally, the summarized one-step transition probability from the q-th level to a lower level tends to 1.

Let us single out the subset of states

$$\langle\alpha\rangle = \{(i, q),\ i = \overline{1, r},\ q = \overline{0, m}\}.$$

Denote by $\pi_\varepsilon(i, q)$ the stationary distribution of $X_\varepsilon(k)$ and by $g_\varepsilon(\langle\alpha\rangle)$ the steady state probability of exit from $\langle\alpha\rangle$, that is

$$g_\varepsilon(\langle\alpha\rangle) = \sum_{i=1}^{r} \pi_\varepsilon(i, m) \sum_{j=1}^{r} p_\varepsilon[(i, m), (j, m+1)].$$

Let

$$P = \|p_{ij}\|,\quad i, j = \overline{1, r},\quad A^{(q)} = \|\alpha_{ij}^{(q)}\|,\quad i, j = \overline{1, r},\quad q = \overline{0, m},$$

$\{\pi_k,\ k = \overline{1, r}\}$ the stationary distribution for the chain with matrix P,

$$\bar{\pi}_\varepsilon^{(q)} = \left(\pi_\varepsilon^{(q)}(i, q),\ i = \overline{1, m}\right),\quad \bar{\pi} = (\pi_1, \ldots, \pi_r)$$

row-vectors.

Conditions (1)—(4) enable us to compute the main terms of the asymptotic expression for $\bar{\pi}_\varepsilon^{(q)}$ and $g_\varepsilon(\langle\alpha\rangle)$, namely, we obtain

$$\begin{aligned} \bar{\pi}_\varepsilon^{(q)} &= \varepsilon^q \bar{\pi} A^{(0)} A^{(1)} \ldots A^{(q-1)} + o(\varepsilon^q),\quad q \geqq 1, \\ g_\varepsilon(\langle\alpha\rangle) &= \varepsilon^{m+1} \bar{\pi} A^{(0)} \ldots A^{(m)} \underline{1} + o(\varepsilon^{m+1}), \end{aligned} \tag{3}$$

where $\underline{1} = (1, \ldots, 1)^T$.

Now, making use of Theorem 2 and formula (3) we get the following asymptotic result. (See Anisimov [2], [3].)

Let $(\xi_\varepsilon(t),\ t \geqq 0)$ be a SMP given by the embedded Markov chain $(X_\varepsilon(k),\ k \geqq 0)$ satisfying conditions (1)—(4).

Let the times $\tau_\varepsilon((l, s), (j, z))$ transition time from state (l, s) to state (j, z) fulfill the condition

$$\mathbf{E} \exp\{i\theta\beta_\varepsilon \tau_\varepsilon((l, s), (j, z))\} = 1 + a_{lj}(s, z, \theta)\varepsilon^{m+1} + o(\varepsilon^{m+1}),$$

where β_ε is a normalizing factor.

Denote by $\Omega_\varepsilon(j, s)$ the instant at which the SMP reaches the $q+1$-th level for the first time, provided $\xi_\varepsilon(0) = (j, z)$, $s \leqq m$.

Corollary 2. If the above conditions are satisfied, then

$$\lim_{\varepsilon \to 0} \mathbf{E}\{iu\beta_\varepsilon \Omega(j, s)\} = (1 - A(\theta))^{-1},$$

where

$$A(\theta) = \Big(\sum_{k,j=1}^{r} \pi_k p_{kj} a_{kj}(0, 0, \theta)\Big) / (\bar{\pi} A^{(0)} \ldots A^{(m)} \underline{1}).$$

In particular, if $a_{ij}(s, z, \theta)=i\theta m_{ij}(s, z)$, then the limit is an exponentially distributed random variable with parameter

$$(\bar{\pi}A^{(0)}\dots A^{(m)}\underline{1})/(\sum_{k,j=1}^{r}\pi_k p_{kj} m_{kj}(0,0)).$$

3. The mathematical models

In this section we show how the above results for sojourn time problems can be applied to the asymptotic analysis of controlled finite-source queueing systems under the assumption of fast service.

3.1. System $\langle N/M_u/M_u/n\rangle$.

Let the requests emanate from a finite source of size N which are served by one of n $(n\leqq N)$ servers at a service facility according to a First-In-First-Out (FIFO) discipline. If there is no idle server, then a waiting line is formed and the customers are delayed. Suppose that the system is evolving in a random environment governed by an irreducible, aperiodic Markov chain $(x(t),\ t\geqq 0)$ with state space $\{1, \dots, r\}$ and transition density matrix

$$\{a_{ij},\ i, j=\overline{1, r},\quad a_{ii}=\sum_{j\neq i}a_{ij}\}.$$

Whenever the environmental process is in state i and there are s, $s=\overline{0, N-1}$ customers at the service facility, each request is assumed to stay in the source for a random time having exponential distribution with parameter $\lambda(i, s)$. Furthermore, the service time of each customer is supposed to be an exponentially distributed random variable with parameter $\mu_\varepsilon(i, s)$, $i=\overline{1, r}$, $s=\overline{1, N}$.

All random variables involved here are assumed to be independent of each other.

Let us consider the system under the assumption of fast service, that is, $\mu_\varepsilon(i, s)\to\infty$ as $\varepsilon\to 0$. For simplicity let $\mu_\varepsilon(i, s)=\mu(i, s)/\varepsilon$. Denote by $y_\varepsilon(t)$ the number of customers staying at the service facility at time t, and let

$$Z_\varepsilon(t)=(X(t), y_\varepsilon(t)).$$

Clearly, $(Z(t),\ t\geqq 0)$ is a two-dimensional Markov process with state space

$$E=\{(i, s),\ i=\overline{1, r},\ s=\overline{0, N}\}.$$

Let $\Omega_\varepsilon(k, q\colon m)$ denote the instant at which $y_\varepsilon(t)=m+1$ for the first time, provided that the initial state of $Z_\varepsilon(t)$ was (k, q), $k=\overline{1, r}$, $q=\overline{0, m}$, $m=\overline{1, N-1}$. That is,

$$\Omega_\varepsilon(k, q\colon m)=\inf\{t\colon t>0,\ y_\varepsilon(t)=m+1/Z_\varepsilon(0)=(k, q)\}$$

which is termed, in the sequel, as the time to the first system failure. It is easy to see, that $\Omega_\varepsilon(k, q\colon m)$ is the first exit time of $Z_\varepsilon(t)$ from the subset

$$\langle\alpha\rangle=\{(i, s),\ i=\overline{1, r},\ s=\overline{0, m}\},$$

provided that $Z_\varepsilon(0)=(k, q)$.

We are interested in the limiting distribution of the random variable $\Omega_\varepsilon(k, q: m)$ as $\varepsilon \to 0$.

It can easily be verified that the following transitions occur in an arbitrary time interval $(t, t+h)$

$$(i, s) \xrightarrow{h} \begin{cases} (j, s) & \text{w.p.} \quad a_{ij}h+o(h), \quad i \neq j, \\ (i, s+1) & \text{w.p.} \quad (N-s)\lambda(i, s)+o(h), \quad 0 \leqq s < N, \\ (i, s-1) & \text{w.p.} \quad (S_n\mu(i, s)/\varepsilon)h+o(h), \quad 1 \leqq s \leqq N, \\ (i, s) & \text{w.p.} \quad 1-h[(N-s)\lambda(i, s)+S_n\mu(i, s)/\varepsilon+a_{ii}]+o(h), \end{cases}$$

$$S_n = \min(s, n)$$

The sojourn time of $Z_\varepsilon(t)$ in state (i, s) is exponentially distributed with parameter

$$(N-s)\lambda(i, s)+S_n\mu(i, s)/\varepsilon+a_{ii}.$$

Thus, the transition probabilities for the embedded Markov chain are

$$p_\varepsilon[(i, s), (j, s)] = a_{ij}[(N-s)\lambda(i, s)+S_n\mu(i, s)/\varepsilon+a_{ii}]^{-1}, \quad 1 \leqq s \leqq N,$$

$$p_\varepsilon[(i, 0), (j, 0)] = a_{ij}[N\lambda(i, 0)+a_{ii}]^{-1}, \quad s = 0,$$

$$p_\varepsilon[(i, s), (i, s+1)] = (N-s)\lambda(i, s)[(N-s)\lambda(i, s)+S_n\mu(i, s)/\varepsilon+a_{ii}]^{-1}, \quad 1 \leqq s \leqq N,$$

$$p_\varepsilon[(i, 0), (i, 1)] = N\lambda(i, 0)[N\lambda(i, 0)+a_{ii}]^{-1}, \quad s = 0,$$

$$p_\varepsilon[(i, s), (i, s-1)] = \frac{S_n\mu(i, s)}{\varepsilon}[(N-s)\lambda(i, s)+S_n\mu(i, s)/\varepsilon+a_{ii}]^{-1}, \quad 1 \leqq s \leqq N.$$

As $\varepsilon \to 0$, this implies

$$p_\varepsilon[(i, s), (j, s)] = o(1), \quad s \geqq 1,$$

$$p_\varepsilon[(i, s), (i, s+1)] = \frac{\varepsilon(N-s)\lambda(i, s)}{S_n\mu(i, s)}(1+o(1)), \quad 1 \leqq s \leqq N,$$

$$p_\varepsilon[(i, s), (i, s-1)] \to 1, \quad 1 \leqq s \leqq N,$$

$$p_\varepsilon[(i, 0), (j, 0)] = a_{ij}/(N\lambda(i, 0)+a_{ii}), \quad i, j = \overline{1, r},$$

$$p_\varepsilon[(i, 0), (i, 1)] = N\lambda(i, 0)/(N\lambda(i, 0)+a_{ii}), \quad i = \overline{1, r}.$$

This agrees with the conditions (1)—(4) of Example, but here the zero level is the set $\{(i, 0), (i, 1), i=\overline{1, p}\}$ while the q-th level is $\{(i, q+1), i=\overline{1, r}\}$. Since the level 0 in the limit forms an essential class, the probabilities $\pi_0(i, q)$, $i=\overline{1, r}$, $q=\overline{0, 1}$ satisfy the following system of equations

$$\pi_0(j, 0) = \sum_{i \neq j} \pi_0(i, 0)a_{ij}/(N\lambda(i, 0)+a_{ii})+\pi_0(j, 1), \tag{4}$$

$$\pi_0(j, 1) = \pi_0(j, 0)N\lambda(j, 0)/(N\lambda(j, 0)+a_{jj}). \tag{5}$$

Substituting eq. (5) to eq. (4) we get

$$\pi_0(j,0)\frac{a_{jj}}{N\lambda(j,0)+a_{jj}} = \sum_{i\neq j}\pi_0(i,0)\frac{a_{ij}}{N\lambda(i,0)+a_{ii}}. \tag{6}$$

Denote by π_k, $k=\overline{1,r}$ the stationary distribution of the governing Markov chain $(x(t),\ t\geqq 0)$.

Since

$$\pi_j a_{jj} = \sum_{i\neq j}\pi_i a_{ij}, \quad j=\overline{1,r},$$

from (6) we have

$$\pi_0(i,0) = B\pi_i[N\lambda(i,0)+a_{ii}],$$

$$\pi_0(i,1) = B\pi_i N\lambda(i,0), \quad i=\overline{1,r},$$

where

$$B = \Big[\sum_{k=1}^{r}\pi_k\big(a_{kk}+2N\lambda(k,0)\big)\Big]^{-1}.$$

By using formula (3), it is easy to obtain that

$$\pi_\varepsilon(i,q) = \varepsilon^{q-1}B\pi_i N\lambda(i,0)\prod_{s=1}^{q-1}\frac{(N-s)\lambda(i,s)}{S_n\mu(i,s)}(1+o(1)),$$

$$\prod_{s=1}^{0} = 1, \quad q\geqq 1,$$

and

$$g_\varepsilon(\langle\alpha\rangle) = \varepsilon^m NB\sum_{i=1}^{r}\pi_i\lambda(i,0)\prod_{s=1}^{m}\frac{(N-s)\lambda(i,s)}{S_n\mu(i,s)}(1+o(1)).$$

Taking into consideration the exponentiality of $\tau_\varepsilon(i,s)$ for a fixed u we have

$$\mathbf{E}\exp\{i\varepsilon^m u\tau_\varepsilon(j,0)\} = 1+\varepsilon^m\frac{iu}{a_{jj}+N\lambda(j,0)}(1+o(1)),$$

$$\mathbf{E}\exp\{i\varepsilon^m u\tau_\varepsilon(j,s)\} = 1+o(\varepsilon^m), \quad s>0.$$

Notice, that $\beta_\varepsilon=\varepsilon^m$. Therefore, by the help of Corollary 2 we immediately get the following theorem.

Theorem 3. For the system $\langle N/M_u/M_u/n\rangle$ under the above conditions, for any $k=\overline{1,r}$, $q\leqq m$ the distribution of the normalized random variable $\varepsilon^m\Omega_\varepsilon(k,q\colon m)$ converges weakly to an exponentially distributed random variable with parameter

$$\Lambda = \sum_{i=1}^{r}\pi_i N\lambda(i,0)\prod_{s=1}^{m}\frac{(N-s)\lambda(i,s)}{S_n\mu(i,s)}.$$

Consequently, the distribution of time to the first system failure can be approximated by

$$\mathbf{P}\big(\Omega_\varepsilon(k,q\colon m)>t\big)\approx\exp(-\varepsilon^m\Lambda t).$$

In particular, for the system $\langle N/M/M/n\rangle$ the parameter Λ can be written as

$$\Lambda = \lambda\left(\frac{\lambda}{\mu}\right)^m N \prod_{s=1}^{m} \frac{(N-s)}{S_n}.$$

Furthermore, for the random variable $\varepsilon^{n+m}\Omega_\varepsilon(k, q\colon n+m)$, Λ is given by

$$\Lambda = \lambda\left(\frac{\lambda}{\mu}\right)^{n+m} \frac{N}{n!\,n^m} \prod_{s=1}^{n+m} (N-s). \tag{7}$$

It is well-known, that if $N\to\infty$ and $\lambda\to 0$ such that $N\lambda\to\lambda'$, then the stationary characteristics of the system $\langle N/M/M/n\rangle$ coincide with the corresponding characteristics of the system $M/M/n$ with Poisson arrivals with parameter λ' and exponentially distributed service times with parameter μ/ε. (See [8].) In fact, applying (7) as $N\to\infty$, $\lambda\to 0$, $N\lambda\to\lambda'$ we easily get

$$\Lambda = \frac{1}{n!\,n^m}\lambda'\left(\frac{\lambda'}{\mu}\right)^{n+m}$$

which agrees with the result of Anisimov [3] pp. 157.

3.2. The system $\langle N/\vec{M}_u/\vec{M}_u/1\rangle$.

Let us consider problem 3.1. with the following modifications. The requests are stochastically different, unit k is characterized by arrival rate $\lambda_k(i, s)$ and service rate $\mu_k(i, s)/\varepsilon$, provided that the underlying Markov chain is in state i and there are s customers at the service facility consisting of one server.

We are interested in the limiting distribution of $\Omega_\varepsilon(m)$ under the assumption of fast service, that is, as $\varepsilon\to 0$.

Let

$$Z_\varepsilon(t) = \{X(t),\ y_\varepsilon(t)\colon \gamma_1(t), \ldots, \gamma_{y_\varepsilon(t)}(t)\}$$

be a multi-dimensional Markov process with state space

$$E = \{(i, s\colon k_1, \ldots, k_s);\ i = \overline{1, r},\ s = \overline{0, N},\ (k_1, \ldots, k_s)\in V_N^s,\ k_0 = 0\},$$

where

$X(t)$ is the governing Markov chain,

$y_\varepsilon(t)$ is the number of customers staying at the service facility at time t,

$\gamma_1(t), \ldots, \gamma_{y_\varepsilon(t)}(t)$ are the indices of requests staying at the service facility at time t, ordered lexicographically,

V_N^s is the set of all variations of order k of integers $1, \ldots, N$.

Let us single out the subset of states

$$\langle\alpha\rangle = \{(i, q\colon k_1, \ldots, k_q),\ i = \overline{1, r},\ q = \overline{0, m},\ (k_1, \ldots, k_q)\in V_N^q\}.$$

Notice, that $\Omega_\varepsilon(m)$ is the first exit time of $Z_\varepsilon(t)$ from $\langle\alpha\rangle$. On the analogy of 3.1, it is not difficult to verify, that the transition probabilities of the embedded Markov chain as $\varepsilon \to 0$, are

$$p_\varepsilon[(i, 0\colon 0), (j, 0\colon 0)] = a_{ij}/[\sum_{l=1}^{N} \lambda_l(i, 0) + a_{ii}], \quad i = \overline{1, r}.$$

$$p_\varepsilon[(i, 0\colon 0), (i, 1\colon k)] = \lambda_k(i, 0)/[\Sigma\lambda_l(i, 0) + a_{ii}], \quad i = \overline{1, r}, \quad k = \overline{1, N},$$

$$p_\varepsilon[(i, s\colon k_1, \ldots, k_s), (j, s\colon k_1, \ldots, k_s)] = o(1), \quad s \geqq 1,$$

$$p_\varepsilon[(i, s\colon k_1, \ldots, k_s), (i, s+1\colon k_1, \ldots, k_{s+1})] = \frac{\varepsilon\lambda_{k_{s+1}}(i, s)}{\mu_{k_1}(i, s)}(1+o(1)),$$

$$p_\varepsilon[(i, s\colon k_1, \ldots, k_s), (i, s-1, k_2, \ldots, k_s)] \to 1, \quad 1 \leqq s \leqq N,$$

$$(i, 0\colon k_2, \ldots, k_1) = (i, 0\colon 0).$$

Now, we can obtain that

$$\pi_\varepsilon(i, q\colon k_1, \ldots, k_q) = \varepsilon^{q-1} B\pi_i \frac{\lambda_{k_1}(i, 0)\cdot\lambda_{k_2}(i, 1)\ldots\lambda_{k_q}(i, q-1)}{\mu_{k_1}(i, 1)\cdot\mu_{k_1}(i, 2)\ldots\mu_{k_1}(i, q-1)}(1+o(1)),$$

$$\pi_\varepsilon(i, q) = \sum_{(k_1, \ldots, k_q)} \pi_\varepsilon(i, q\colon k_1, \ldots, k_q),$$

and

$$g_\varepsilon(\langle\alpha\rangle) = \varepsilon^m B \sum_{i=1}^{r} \pi_i \sum_{(k_1, \ldots, k_{m+1})} \frac{\lambda_{k_1}(i, 0)\ldots\lambda_{k_{m+1}}(i, m)}{\mu_{k_1}(i, 1)\ldots\mu_{k_1}(i, m)}(1+o(1)),$$

where

$$B = [\sum_{i=1}^{r} \pi_i(a_{ii} + 2\sum_{l=1}^{N} \lambda_l(i, 0))]^{-1}.$$

Making use of Corollary 2 we are ready to get the following theorem.

Theorem 4. For the system $\langle N/\vec{M}_u/\vec{M}_u/1\rangle$ under the the above conditions, independently of the initial state, the distribution of the normalized random variable $\varepsilon^m\Omega_\varepsilon(m)$ converges weakly to an exponentially distributed random variable with parameter

$$\Lambda = \sum_{i=1}^{r} \pi_i \sum_{(k_1, \ldots, k_{m+1})} \frac{\lambda_{k_1}(i, 0)\ldots\lambda_{k_{m+1}}(i, m)}{\mu_{k_1}(i, 1)\ldots\mu_{k_1}(i, m)}.$$

4. Applications

In this section we show how the above results can be applied to problems arised in the field of reliability theory and computer performance.

4.1. Consider a repairable system operating in a random environment, which consists of N elements and one repairman. The expected life time of the k-th element is assumed to be an exponentialy distributed random variable with failure rate λ_k. When an element fails, it enters the repair facility and will be immediately repaired unless the repairman is busy, otherwise it will wait in a queue in the order of its arrival. The required repair time of the k-th element is supposed to be exponentially distributed random variable with parameter μ_k, $k=\overline{1, N}$. Furthermore, we assume that the failure and repair intensities depend on the state of the environmental process and the number of failed elements. Namely, whenever the environmental chain is in state i and there are s elements at the repair facility, the rate at which the remaining duration of life times decreases is $a(i, s)$ and the rate at which the remaining duration of repair time decreases is $b(i, s)/\varepsilon$.

The involved random variables are supposed to be independent of each other.

The system is said to be failed if the number of failed elements is $m+1$. Therefore, the instant $\Omega_\varepsilon(m)$ at which the queue length reaches the level $m+1$ for the first time is of great practical importance. Hence, the problem is to find the asymptotic distribution of the random variable under the assumption of fast repair, that is, as $\varepsilon \to 0$. This assumption is justified, since usually the average repair time is many times smaller than the average failure-free oparation time.

Clearly, this problem corresponds to the system $\langle N/\vec{M}_u/\vec{M}_u/1\rangle$, thus the distribution of $\Omega_\varepsilon(m)$ can be approximated by

$$\mathbf{P}(\Omega_\varepsilon(m) > t) \approx \exp(-\varepsilon^m \Lambda t),$$

where

$$\Lambda = \sum_{i=1}^{r} \pi_i \sum_{(k_1, \ldots, k_{m+1})} \frac{\lambda_{k_1} a(i, 0) \ldots \lambda_{k_{m+1}} a(i, m)}{\mu_{k_1} b(i, 1) \ldots \mu_{k_1} b(i, m)}.$$

4.2. Let us consider a CP-terminal system consisting of N terminals coupled to one Central Processor Unit (CPU). The system is operating in a random environment which influences the service rates at the terminals and at the CPU. At the terminals the think times are exponentially distributed random variables with parameter λ_k for terminal k, $k=\overline{1, N}$. The processing times for jobs at the CPU are exponentially distributed random variables with mean ε/μ_k, for the job from terminal k, $k=\overline{1, N}$, where ε is a small parameter. The service discipline at the CPU is FIFO. Whenever the environmental process is in state i the rate at which the remaining duration of think times, processing time decreases is $a(i)$, $b(i)$ respectively.

The think and processing times are supposed to be independent of each other. Let us assume that the average CPU times are many times smaller than the average think times, that is, $\varepsilon \approx 0$.

We are interested in the distribution of the instant $\Omega_\varepsilon(m)$ at which the number of jobs at the CPU reaches the level $m+1$, $1 < m < N$.

It is easy to see, that applying Theorem 4 we get the following approximation

$$\mathbf{P}(\Omega_\varepsilon(m) > t) \approx \exp\{-\varepsilon^m \Lambda t\}$$

where

$$\Lambda = \sum_{i=1}^{r} \pi_i \sum_{(k_1, \ldots, k_{m+1})} \frac{\lambda_{k_1} a(0) \ldots \lambda_{k_{m+1}} a(m)}{\mu_{k_1} b(1) \ldots \mu_{k_1} b(m)}.$$

Remark. We must admit that for terminal systems this characteristic is a less effective performance measure but sometimes it is useful to know.

Abstract

This paper is concerned with an asymptotic analysis of some controlled finite-source queueing systems under the assumption of fast service. Firstly, a brief summary of preliminary results related the asymptotic behavior of SMP is given. Secondly, models of queueing systems with fast service is treated. It is shown, that the time to the first system failure converges in distribution, under appropiate norming, to an exponentially distributed random variable. Finally, applications of the systems in the field of reliability theory and computer performance are considered.

Keywords: SMP, sojourn time, fast service, time to the first system failure, weak convergence.

References

[1] ANISIMOV V. V., Limit theorems for sums of random variables in an array of sequence defined on a subset of states of a Markov chain up to the exit time, Theor. Probability and Math. Statist. 1974 No 4, 15—22.
[2] ANISIMOV V. V., Asymptotic methods in analysing the behavior of stochastic systems, Mecniereba, Tbilisi, 1984 (in Russian).
[3] ANISIMOV V. V.—ZAKUSILO O. K.—DONCSENKO V. S., Elements of queueing theory and asymptotic analysis of systems, Visa Skola, Kiev, 1987 (in Russian).
[4] FRANKEN P.—KÖNIG D.—ARNDT U.—SCHMIDT V., Queues and Point Processes, Akademie-Verlag, Berlin, 1981.
[5] GNEDENKO B. V. and KOVALENKO I. N., Introduction to Queueing Theory, Nauka, Moscow, 1966 (in Russian).
[6] GNEDENKO B. V. (Editor), Mathematical problems in reliability theory, Radio i Svjaz, 1983 (in Russian).
[7] GNEDENKO B. V. and KÖNIG D., Handbuch der Bedienungstheorie, Vol. I., II., Akademie-Verlag, Berlin, 1983, 84.
[8] JAISWAL N. K., Priority Queues, Academic Press, New York, 1968.
[9] KLEINROCK L., Queueing Systems, Vol. II. Computer applications, Wiley-Interscience, New York, 1976.
[10] KOROLJUK V. S., Asymptotic behavior of the sojourn time of Semi-Markov Process in a subset of the states, Ukrain. Mat. Z. 21 (1969) 842—845 (in Russian).
[11] KOROLJUK V. S. and TURBIN A. F., Semi-Markov Processes and their applications, Naukova Dumka, Kiev, 1976 (in Russian).
[12] KOVALENKO I. N., Studies in reliability analysis of complex systems, Naukova Dumka, Kiev, 1975 (in Russian).
[13] KÖNIG D. and STOYAN D., Methoden der Bedienungstheorie, Academie-Verlag, Berlin, 1976.
[14] LAVENBERG S. S. (Editor), Computer Performance Modeling Handbook, Academic Press, New York, 1983.
[15] LIFSIC A. L. and MALC E. A., Simulation of queueing systems, Sovetskoe Radio, Moscow, 1978 (in Russian).
[16] TAKÁCS L., Introduction to the Theory of Queues, Oxford University Press, New York, 1962.
[17] WHITE J. A.—SCHMIDT J. W.—BENNETT G. K., Analysis of Queueing Systems, Academic Press, New York, 1975.

Received April 5, 1988

A Decomposition Theorem for a Class of Infinite Transformation Semigroups

RALPH P. TUCCI

DEPARTMENT OF MATHEMATICAL SCIENCES
LOYOLA UNIVERSITY
NEW ORLEANS, LA. 70118 USA

1. Introduction

In recent years there has been a great deal of interest in infinite automata; see for example Bavel [2] and Gacs [9] (automata theory), Biermann [3] and Scott [15] (semantics), Reeker [11] (formal languages), and Reeker and Tucci [12] (algorithms). One approach to the study of infinite automata is to see how much of the theory of finite machines extends to infinite machines. The purpose of this paper is to generalize results on the decomposition of finite automata to infinite automata. Previous results on the decomposition of infinite automata include the results of Bavel [2], in which he decomposes an infinite automaton into the union of certain sub-automata. Rhodes [13], [14] and Warne [18], [19] have developed decompositions for infinite semigroups similar to the classical Krohn—Rhodes decomposition (Arbib [1]). Ésik [6], [7] and Ésik and Gécseg [8] have studied decomposition from the point of view of varieties. Tucci [16] has developed a wreath product decomposition of infinite automata in terms of reset machines, group machines, and a third type of machine known as unique predecessor machines (see Bavel [2]). This decomposition is in the spirit of the Krohn—Rhodes decomposition, although the decomposition itself is much weaker. It seems necessary to make certain assumptions on an infinite automaton to obtain a stronger decomposition, and that is what we do in this article.

In this paper we work w th transformation semigroups rather than with automata because the notation is simpler. We develop a decomposition theorem for a certain class of unique predecessor transformation semigroups (Bavel [2]). The basic idea is to generalize the holonomy decomposition theorem of Eilenberg [5, theorem 7.1] to infinite transformation semigroups. We choose the holonomy decomposition theorem because it generalizes to the infinite case in a fairly natural manner. We follow closely the exposition presented in Holcombe [10], especially in the last section of this paper.

The second section of this paper develops some technical results on the skeleton of a transformation semigroup, as defined in Holcombe [10], and the third section describes what we call the depth function, which is the dual of the height function given in Holcombe [10]. The fourth section describes the structure of the semigroups into which a certain class of transformation semigroups can be decomposed. The final section contains the main decomposition theorem.

A **transformation semigroup** is an ordered pair $T=(Q, S)$ where Q is a set and S is a semigroup, together with a partial product $Q\times S\to Q$ denoted by concatenation, such that

(1) $(qs_1)s_2 = q(s_1s_2)$ for all $q\in Q$ and $s_1, s_2\in S$;
(2) if $s_1, s_2\in S$ and $qs_1=qs_2$ for all $q\in Q$, then $s_1=s_2$.

Throughout this paper the symbol T always stands for the transformation semigroup (Q, S). We assume that Q contains more than one element, and that Q and S are countable. If A is any subset of Q, then $|A|$ denotes the size of A. The semigroup S is called the **abstract** (or **action**) **semigroup** of S. We will assume that S contains an identity 1 which satisfies the property that $q\cdot 1=q$ for all $q\in Q$. If S is generated by elements $s_1, s_2, \dots, s_n, \dots$, then we denote this by writing $S=\langle s_1, s_2, \dots, \dots, s_n, \dots\rangle$.

For each $s\in S$ we let F_s be the **partial function induced** by s, where F_s is given by the rule $F_s(q)=F(q, s)$ for all $q\in Q$. Note that F_s is single-valued where it is defined, but that F_s may not be defined on all of Q; the set $\{q\in Q|qF_s$ is defined$\}$ is the **domain of** s, denoted dom s. Sometimes for convenience we write qF_s, or simply qs, for $F_s(q)$. If $a, b\in S$ and qF_aF_b is undefined for all $q\in Q$ (i.e., the domain of F_b is disjoint from the range of F_a), then we adjoin a zero 0 to S and define $ab=0$. We can think of 0 as inducing a partial function on Q whose domain is $\emptyset$. A transformation semigroup $T=(Q, S)$ is a **unique predecessor transformation semigroup** if F_s is a $1-1$ map for each $s\in S$ (Bavel, [2, p. 576]). When T is a unique predecessor transformation semigroup, we can define the set $S^{-1}=\{s^{-1}|s\in S\}$ where $F_s-1= =(F_s)^{-1}$, the partial function which is defined by the rule $F_s-1(q)=q'$ if and only if $F_s(q')=q$ for all $q, q'\in Q$. Note that dom $s=Qs^{-1}$ for all $s\in S$. We define the **quotient of** T as the transformation semigroup $T'=(Q, S\cup S^{-1})$. The transformation semigroups we consider in this paper are all quotients of unique predecessor transformation semigroups. Note that if $s=s_1s_2\dots s_n\in S$, then S contains the element $s^{-1}=s_n^{-1}\dots s_2^{-1}s_1^{-1}$, where $ss^{-1}s=s$. Hence the action semigroup of the quotient of a unique predecessor transformation semigroup is regular.

Throughout this paper the symbol N stands for the set of positive integers. We assume that the reader is familiar with the definitions of the restricted direct product and the wreath product of transformation semigroups as given in Holcombe [10]. We also assume that the reader is familiar with the basic theory of semigroups as presented in Clifford and Preston [4].

2. The skeleton

To begin we need some preliminary definitions.

2.1. Definition. The **skeleton** (Holcombe [10, p. 119]) of a transformation semigroup T, denoted $I(T)$, is the collection of subsets of Q of the form $\emptyset$, $\{q\}$ (where $q\in Q$), or Qs for any $s\in S$. Since S contains an identity element by definition, we have that $Q\in I(T)$. Also, as we have observed earlier, if $s\in S$, then dom $s=Qs^{-1}$, so that dom $s \in I(T)$ for every $s \in S$. If A, B are two skeleton elements, then $A \leqq B$ if there is some $s \in S$ such that $A \subseteqq Bs$. Since S contains an identity element, and since this identity induces the identity map on Q, the condition $A \subseteqq B$ implies that $A \leqq B$. Two skeleton elements A, B are **equivalent** (Holcombe [10, p. 119]), denoted $A \equiv B$, if $A \leqq B$ and $B \leqq A$; i.e., $A \equiv B$ if there are elements $s, t \in S$ such that $A \subseteqq Bs$ and $B \subseteqq At$. If $A \leqq B$ and $A \not\equiv B$, then we indicate this by writing $A < B$. The elements A, B are **strongly equivalent**, denoted $A \equiv_s B$, if there are $s, t \in S$ such that $A = Bs$, $B = At$.

In a finite transformation semigroup, if $A, B \in I(T)$, $A \subseteqq B$, and $A \equiv B$, then $A = B$. However, in an infinite transformation semigroup we can have $A, B \in I(T)$, $A \subseteqq B$, and $A \equiv B$ but $A \neq B$.

2.2. Example. Let $T = (Q, S)$ where $Q = \{q_n | n \in N\}$, $S = \langle s, s^{-1} \rangle$, and $F_s(q_n) = q_{n+1}$ for all $q \in Q$. The partial function $F_s - 1$ is defined in the obvious manner. If we take $A = \{q_n | n > 1\} = Qs$ and $B = Q$ then $A, B \in I(T)$ and $A \equiv B$ but $A \neq B$. ∎

In a finite transformation semigroup, equivalent is the same as strongly equivalent (Holcombe [10, proposition 4.2.2]). This is not necessarily the case in an infinite transformation semigroup.

2.3. Example. Let $T = (Q, S)$ where $Q = \{p_n | n \in N\} \cup \{q_n | n \in N\}$ and $S = \langle a, b, s, t, a^{-1}, b^{-1}, s^{-1}, t^{-1} \rangle$. Define

(1) $F_a(p_j) = p_j$ for all $j \geqq 2$;

(2) $F_b(q_j) = q_j$ for all $j \geqq 2$;

(3) $F_s(q_j) = p_{j-1}$ for all $j \geqq 2$;

(4) $F_t(p_j) = q_{j-1}$ for all $j \geqq 2$.

The partial functions induced by $a^{-1}, b^{-1}, c^{-1}, t^{-1}$ are defined in the obvious way. For any other $x \in S$ and $q \in Q$, we have that $F_x(q)$ is undefined.

Let $Qa = A = \{p_j | j \geqq 2\}$, and let $Qb = B = \{q_j | j \geqq 2\}$. Then $A \subseteqq Bs$ and $B \subseteqq At$ but there are no elements $s', t' \in S$ such that $A = Bs'$ and $B = At'$. ∎

We now develop a simple but important criterion which we need in section 5 to make our decomposition work.

2.4. Definition. The skeleton $I(T)$ of the transformation semigroup T **satisfies the weak ascending chain condition,** or *WACC*, if every ascending chain of non-equivalent skeleton elements under the relation $\leqq$ halts after finitely many steps. Similarly, the semigroup S **satisfies the ascending chain condition on some class C of left (right, two-sided) ideals** if every increasing chain under inclusion of left (right, two-sided) ideals from class C halts after finitely many steps.

2.5. Example. The transformation semigroup in figure 2.1 does not satisfy *WACC* on its skeleton. The state set of this transformation semigroup consists of infinitely many components, where each component contains one more vertical edge labeled b than does the previous component. This transformation semigroup has the chain $Qaba \subsetneqq Qab^2a \subsetneqq \dots \subsetneqq Qab^na \subsetneqq \dots$ since $Qaba = \{q_1\}$, $Qab^2a = \{q_1, q_2\}$, etc.

(For simplicity we omit edges labeled by a^{-1} and b^{-1}.) ∎

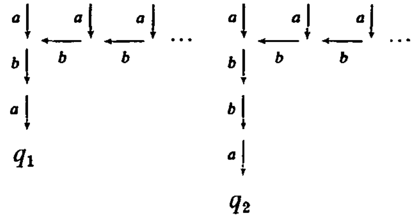


Figure 2.1

2.6. Proposition. Let T be a transformation semigroup, and let $Qa, Qb \in I(T)$. Then $Qa \cap Qb \in I(T)$.

Proof. It is easy to see that $Qa \cap Qb = Qab^{-1}b = Qba^{-1}a$. ∎

2.7. Proposition. The skeleton of the transformation semigroup $T=(Q, S)$ satisfies *WACC* if and only if S satisfies *ACC* on cyclic left ideals.

Proof. Let $A=Qa$, $B=Qb$ for some $a, b \in S$. Suppose first that $A \leqq B$. Then there is some $s \in S$ such that $Qa \subseteqq Qbs$. By proposition 2.6 $Qa = Qa \cap Qbs = Qa(bs)^{-1}bs$. It can be shown that $\operatorname{dom} a(bs)^{-1}bs = \operatorname{dom} a$, so that $a = a(bs)^{-1}bs$. Hence $SaS \subseteqq SbS$. Conversely, if $SaS \subseteqq SbS$, then $a = xby$ for some $x, y \in S$, so that $Qa = Qxby \subseteqq Qby$ and therefore $A \leqq B$. In a similar fashion, we can prove that $A \subseteqq B$ if and only if $Sa \subseteqq Sb$. Therefore, $A \equiv B$ if and only if $SaS = SbS$ or $Sa = Sb$. Hence, $I(T)$ satisfies *WACC* if and only if S satisfies *ACC* on cyclic left ideals and on cyclic two-sided ideals. The result now follows from the fact that every two-sided ideal is a left ideal. ∎

From now on, we assume that the skeletons of all transformation semigroups under consideration satisfy *WACC*. We conclude this section with a technical result.

2.8. Proposition. Let $T=(Q, S)$ be a transformation semigroup, and let $A, B \in I(T)$, where $A=Qa$, $B=Qb$ for some $a, b \in S$. For any $s \in S$, $(Qa \cap Qb)s = Qas \cap Qbs$.

Proof. Let $q \in (Qa \cap Qb)s$. Then $q \in Qas$ and $q \in Qbs$ so $q \in Qas \cap Qbs$. Conversely, suppose $q \in Qas \cap Qbs$. Then there exist $q_1, q_2 \in Q$ such that $q = (q_1 a)s = (q_2 b)s$. Since all elements of S induce $1-1$ maps on Q, we have $q_1 a = q_2 b$, so that $q_1 a \in (Qa \cap Qb)$. Thus, $q \in (Qa \cap Qb)s$. ∎

3. The depth function

3.1. Definition. Let $T=(Q, S)$ be a transformation semigroup. A **depth function** is a function d from $I(T)$ to the class of ordinals such that

(1) $d(Q)=0$.

(2) $d(\{q\}) > d(A)$ for all $q \in Q$ and $A \in I(T)$ with $|A|>1$, and $d(\emptyset) > d(\{q\})$ for all $q \in Q$.

(3) If $A \equiv B$ then $d(A)=d(B)$.
(4) If $A<B$ then $d(A)>d(B)$.
(5) If there are $A, B \in I(T)$ such that $d(A)=n$ and $d(B)=m$ then for every ordinal k with $n<k<m$ there is some $C \in I(T)$ such that $d(C)=k$.

The depth function as defined in definition 3.1 is the dual of the height function as given in Holcombe [10, p. 121]. To construct a depth function we need the following proposition.

3.2. Proposition. Let $T=(Q, S)$ be a transformation semigroup and let $A, B \in I(T)$. Define a function d from $I(T)$ to the class of ordinals as follows:
(1) $d(Q)=0$.
(2) Let $|A|>1$. If $A \equiv Q$ then $d(A)=0$; otherwise, $d(A)=\sup\{1+d(B)|A<B\}$.
(3) If $|A|=1$, then $d(A)=\sup\{1+d(B)||B|>1\}$.
(4) If $A=\emptyset$ then $d(A)=1+d(B)$, where B is any one-element set in the skeleton.
Then d is a depth function. (The **depth of the transformation semigroup** is $d(\{q\})$ for any $q \in Q$, which we denote by $d(T)$.)

Proof. We need to show that the conditions of definition 3.1 are satisfied.
3.1 (1) This follows from the definition of d.
3.1 (2) This follows from the definition of d.
3.1 (3) Let $A \equiv B$. If $|A|=|B|=1$ or $A=B=\emptyset$ then $d(A)=d(B)$ by definition. It is not possible that A is empty or a singleton while B is not, because of the definition of equivalence.
Suppose now that $|A|>1$, $|B|>1$. Then for any $C \in I(T)$ we have $A<C$ if and only if $B<C$.
3.1 (4) If $A<B$, then by definition $d(A)>d(B)$.
3.1 (5) This follows from the definition of d. ∎

3.3. Proposition. Let $T=(Q, S)$ be a transformation semigroup, let $A \in I(T)$, and let d be a depth function.
(1) If $d(A)=0$ then $A \equiv Q$.
(2) If $A, B \in I(T)$ with $A \not\equiv B$, $A \subsetneqq B$, and $|B|>1$ then $d(A)>d(B)$.

Proof. These facts follows immediately from the definition of d. ∎

4. The holonomy transformation semigroup

In this section we describe the structure of the transformation semigroups which we use in our decomposition.

4.1. Definition. Let $T=(Q, S)$ be a transformation semigroup and let $G(S)$ be the group of units of S. For each $A \in I(T)$, let $S_1(A)=\{s \in S | As=A \subseteqq \operatorname{dom} s, s \notin G(S)\}$, and let $S(A)=S_1(A) \cup G(S)$. Take $J(S_1(A))$ to be the ideal generated by $S_1(A)$ (if $S_1(A)=\emptyset$ then define $J(S_1(A))=0$), and let $J(A)=\bigcup\{J(S_1(A'))|A' \subseteqq A, A' \equiv A\}$. The **holonomy transformation semigroup of** A denoted $T(A)$, has as its state set the set $\{Bx | B \in I(T), B \subseteqq A, B \not\equiv A$, and $x \in J(A) \cup G(S)\}$; the action semigroup of $T(A)$ is defined by taking $J(A) \cup G(S)$ and identifying those elements which act the same on the state set of $T(A)$. We will denote the state set of $T(A)$ by $ST(A)$ and the action semigroup of $T(A)$ by $X(A)$.

We define $T(A)$ as we do for several reasons. First of all, if the transformation semigroup is finite then definition 4.1 reduces to $S_1(A)=\{s|As=A\}$, and this is the definition of $S(A)$ in the finite case in Holcombe [10, p, 123]. (Note that $T(A)$ contains the set of all permutations of A onto itself.) Second, in the finite case $S(A)$ is a set of permutations on the maximal skeleton elements contained in A (Holcombe [10, proposition 4.3.1]); in the infinite case, the elements of $S_1(A)$ map the skeleton elements in A to skeleton elements of A of the same depth in a $1-1$ fashion (proposition 4.5). Third, we define $ST(A)$ as we do because there are some elements in $X(A)$ which induce maps on A that do not necessarily preserve the depth of skeleton elements in A, and in fact may not map into A at all (proposition 4.5, example 4.6). Finally, we define $T(A)$ by using $J(A)\cup G(S)$ to assure us that $X(A)$ has at least as much structure as S (proposition 4.2 (1) and (2)) and to assure us that one important technical detail works out (proposition 4.2 (3)).

4.2. Proposition. Let $T(A)=(ST(A), X(A))$ be the holonomy semigroup of A.
(1) $X(A)$ is regular.
(2) $X(A)$ satisfies ACC on cyclic left ideals.
(3) If $As\equiv A$ for some $s\in S$ then s embeds in $X(A)$.

Proof. (1) The ideal $J(A)$ is a regular semigroup, for if $s\in J(A)$, then $s^{-1}=s^{-1}ss^{-1}\in J(A)$. Further, $J(A)\cup G(S)$ is a regular semigroup, and $X(A)$ is a homomorphic image of $J(A)\cup G(S)$.

(2) Any cyclic left ideal of $J(A)$ is of the form $J(A)a\cup\{a\}$ where $a\in J(A)$. But $J(A)a$ is also a left ideal of S. Hence $J(A)$ satisfies ACC on cyclic left ideals. Since every left ideal in $J(A)\cup G(S)$ is contained in $J(A)$, we have that $J(A)\cup G(S)$ satisfies ACC on cyclic left ideals. The result now follows because $X(A)$ is a homomorphic image of $J(A)\cup G(S)$.

(3) Suppose $As\equiv A$. If $s\in G(S)$, then $s\in T(A)$ by definition. If $s\notin G(S)$, let $K=A\cap \operatorname{dom} s$. By proposition 2.6 $K\in I(T)$. Also, $A\equiv As=Ks\leqq K\leqq A$ and hence $K\equiv A$. Further, $Kss^{-1}=K$ and $K\subseteq \operatorname{dom} ss^{-1}$, so that $ss^{-1}\in S_1(K)\subseteq J(K)\subseteq J(A)$. But $J(A)$ is an ideal, so that $s=ss^{-1}s\in J(A)$. Hence s embeds in $X(A)$. ∎

Although $ST(A)$ consists of more than just the skeleton elements of T in A, we concentrate our attention mostly on these latter elements. To study these skeleton elements we need the following notation.

4.3. Definition. Let T be a transformation semigroup and let $A', A\in I(T)$, $A'\subseteq A$. If $A\equiv A'$ then we define $d_A(A')=0$. If $A\not\equiv A'$ then $d_A(A')=\sup\{1+d_A(B)|B\in I(T), A'\subseteq B\subseteq A, A'\not\equiv B\}$. Note that for the function d_A we take chains under inclusion, not chains under the relation $\leqq$. Note also that $d_A(A')$ may be an infinite ordinal. When there is no ambiguity we abbreviate $d_A(A')$ by $d(A')$. Further, we define

$$I(A) = \{A'\in I(T)|A' \subseteq A\}$$
$$I_n(A) = \{A'\in I(T)|A' \subseteq A,\ d_A(A') = n\}$$
$$I_{n^+}(A) = \{A'\in I(T)|A' \subseteq A,\ d_A(A') \geqq n\}$$
$$I_{n'}(A) = \{A'\in I(T)|A' \subseteq A,\ d_A(A') > n\}.$$

4.4. Lemma. Let $T=(Q, S)$ be a transformation semigroup. If $A'\in I(A)$ and $Ax\subseteqq B$ for some $x\in S$ where $A\subseteqq \operatorname{dom} x$ and $B\in I(T)$, then $d_A(A')\leqq d_B(A'x)$.

Proof. Since $A\subseteqq \operatorname{dom} x$ we know that the map $C\to Cx$ for every $C\in I(A)$ is an isomorphism between $I(A)$ and $I(Ax)$ as posets ordered by inclusion. Moreover, $C\leqq D$ if and only if $Cx\leqq Dx$, for all $C, D\in I(A)$. But then $d_A(A')=d_{Ax}(A'x)\leqq$ $\leqq d_B(A'x)$ follows. This completes the proof. ∎

4.5. Proposition. Let $T=(Q, S)$ be a transformation semigroup and let $A\in I(T)$.

(1) Let $s\in S_1(A)$, and let $B\in I_n(A)$. Then the map $B\to Bs$ is a $1-1$ onto map from $I_n(A)$ to $I_n(A)$.

(2) If $s\in S$ and s induces a $1-1$ map from $I_n(A)$ to $I_n(A)$ for all n, then s embeds in $X(A)$.

Proof. (1) It is obvious that each $s\in S_1(A)$ maps $I(A)$ into $I(A)$. We first show that if $B\in I_n(A)$ then $Bs\in I_n(A)$. If $Bs=B$ then we are done. Otherwise, since $As=A\subseteqq \operatorname{dom} s$, we may directly infer from lemma 4.4 that $d_A(B)\leqq d_A(Bs)$. Now $A=As\subseteqq \operatorname{dom} s$ implies that $As^{-1}=Ass^{-1}=A$, and $A=As$ implies that $A\subseteqq \operatorname{dom} s^{-1}$, so that again by lemma 4.4 we have that $d_A(Bs)\leqq d_A(Bss^{-1})$. Since $Bss^{-1}=B$, we have that $d_A(B)=d_A(Bs)$, so $Bs\in I_n(A)$. (Note we have proven that if $s\in S_1(A)$ then $s^{-1}\in S_1(A)$.)

To show that the map $B\to Bs$ is onto, note that $B=(Bs^{-1})s$ where the argument of the previous paragraph shows that $Bs^{-1}\in I_n(A)$. To show that the map is $1-1$, suppose that $Bs=B's$ for some $B'\in I_n(A)$. Then $B=B'$ because every element of S is $1-1$ on its domain.

(2) Let s be any element of S which maps $I_n(A)$ to $I_n(A)$ in a $1-1$ onto fashion. If $s\in G(S)$, then automatically s embeds in $X(A)$. If $s\notin G(S)$, then by hypothesis the element s maps $I_0(A)$ to $I_0(A)$. In particular $As\equiv A$. By proposition 4.2, s embeds in $X(A)$. ∎

In the finite case the condition $As\subseteqq A\subseteqq \operatorname{dom} s$ implies that $As=A$, and hence that s permutes the elements of $I(A)$. This is not necessarily true in the infinite case; in particular, we may have that $As^{-1}\not\subseteq A$.

4.6. Example. Let $T=(Q, S)$ be a transformation semigroup where $Q=$ $=\{q_n|n\in N\}$ and $S=\langle a, s, a^{-1}, s^{-1}\rangle$. Define:

(1) $F_a(q_n)=q_n, \quad n\geqq 2;$

(2) $F_s(q_n)=q_{n+1}, \quad n\geqq 1.$

The partial functions induced by a^{-1} and s^{-1} are defined in the obvious manner. For any other $x\in S$ and $q\in Q$, we have that $F_x(q)$ is undefined.

The set $A=Qa=\{q_n|n\geqq 2\}$ is a skeleton element of this transformation semigroup, and $As\subseteqq A\subseteqq \operatorname{dom} s$. However, $q_1\in As^{-1}$ and $q_1\notin A$. ∎

If we assume that $As\subseteqq A\subseteqq \operatorname{dom} s$ and if we also assume that $As^{-1}\subseteqq A$, then s^{-1} may still not map $I_n(A)$ to $I_n(A)$ for all n.

4.7. Example. Let $T=(Q, S)$ be a transformation semigroup where $Q=$ $=\{q_n|n\in N\}$ and $S=\langle s, t, s^{-1}, t^{-1}\rangle$. Define:

(1) $F_s(q_n) = q_{n+1}, \quad n \geqq 1;$

(2) $F_t(q_1) = q_1$ and $F_t(q_2) = q_2$.

The partial functions induced by s^{-1} and t^{-1} are defined in the obvious manner. For any other $x \in S$ and $q \in Q$, we have that $F_x(q)$ is undefined.

Let $A=Q$, and note that $As \subseteqq A \subseteqq \text{dom}\, s$ (in fact, $A = \text{dom}\, s$) and $As^{-1} \subseteqq A$. Then $B = Qt = \{q_1, q_2\} \in I(A)$ and $Bs^{-1} = \{q_1\}$. Since $Bs^{-1} \subseteqq B$ but $Bs^{-1} \not\equiv B$, we have $d_A(Bs^{-1}) > d_A(B)$. In particular, s^{-1} does not map $I_n(Q)$ to $I_n(Q)$ for all n. ∎

We conclude this section by describing how the relation $A \equiv B$ induces a relation between $I(A)$ and $I(B)$.

4.8. Proposition. Let $T=(Q, S)$ be a transformation semigroup, and let $A, B \in I(T)$ where $A \equiv B$ with $A \subseteqq Bs$ and $B \subseteqq At$ for some $s, t \in S$.

(1) $I_n(A) \subseteqq I_{n^+}(B)s$ and $I_n(B) \subseteqq I_{n^+}(A)t$ for all n.

(2) If $A = Bs$ and $B = At$, then $I_n(A) \subseteqq I_n(B)s$ and $I_n(B) \subseteqq I_n(A)t$ for all n.

Proof. (1) Let $A' \in I_n(A)$, and let $B' = \{b \in B \mid bs = a$ for some $a \in A'\}$. Note that $B' = A's^{-1}$, so that $B' \in I(B) \subseteqq I(T)$. Since $A \subseteqq Bs$, we have $A' = B's$. Suppose that $B' \in I_m(B)$. We must show that $m \geqq n$.

Choose any chain of non-equivalent skeleton elements from A to A', say

$$A \supset A_1 \supset \ldots \supset A'.$$

Since $A \subseteqq Bs$ we have that $A_j \subseteqq \text{dom}\, s^{-1}$ for all A_j. Therefore

$$B \equiv As^{-1} \supseteqq A_1 s^{-1} \supseteqq \ldots \supseteqq A's^{-1} = B'.$$

By the argument of lemma 4.4, the elements of this chain are all non-equivalent. Therefore $m \geqq n$.

(2) Suppose now that $A = Bs$ and $B = At$. We prove that $m \leqq n$. Choose any chain of non-equivalent skeleton elements from B to B', say

$$B \supset B_1 \supset \ldots \supset B'.$$

Since $B \subseteqq At$, for each $B_j \subseteqq B$ there exists some $A_j \subseteqq A$ such that $B_j = A_j t$, and there is also some $A'' \subseteqq A$ such that $B' = A''t$; that is, $A_j = B_j t^{-1}$ and $A'' = B't^{-1}$. Thus, we can rewrite this chain as

$$B \supset A_1 t \supset \ldots \supset A''t = B'$$

This gives rise to a chain in A, namely

$$A \equiv Bt^{-1} \supseteqq A_1 \supseteqq \ldots \supseteqq A''.$$

But then $A' = B's = A''ts$ and $A = Bs = Ats$ which yields the chain

$$A = Ats \supseteqq A_1 ts \supseteqq \ldots \supseteqq A''ts = A'.$$

Again, by the argument of lemma 4.4, the elements of this chain are all non-equivalent. Therefore $m \leqq n$. It follows that $m = n$. ∎

The containments $I_n(A) \subseteqq I_{n^+}(B)s$ and $I_n(B) \subseteqq I_{n^+}(A)t$ in proposition 4.8 (1) cannot be replaced by $I_n(A) \subseteqq I_n(B)s$ and $I_n(B) \subseteqq I_n(A)t$.

4.9. Example. Let $T=(Q, S)$ be a transformation semigroup where

$$Q = \{p_n | n \in N\} \cup \{q_n | n \in N\}$$

and

$$S = \langle a, a', b, b', s, t, a^{-1}, a'^{-1}, b^{-1}, b'^{-1}, s^{-1}, t^{-1} \rangle.$$

Define:

(1) $F_a(p_n) = p_n$ for all $n \geqq 2$;

(2) $F_{a'}(p_n) = p_n$ for all $n \geqq 3$;

(3) $F_b(q_n) = q_n$ for all $n \geqq 2$;

(4) $F_{b'}(q_n) = q_n$ for all $n \geqq 3$;

(5) $F_s(q_n) = p_{n-1}$ for all $n \geqq 2$;

(6) $F_t(p_n) = q_{n-1}$ for all $n \geqq 2$.

The partial functions induced by a^{-1}, a'^{-1}, b^{-1}, b'^{-1}, s^{-1}, t^{-1} are defined in the obvious manner. For all other $x \in S$ and $q \in Q$, we have that $F_x(q)$ is undefined.

Let $A=Qa$ and let $B=Qb$. Then $A \subseteqq Bs$ and $B \subseteqq At$. By definition, $A \in I_0(A)$. If we let $B_1=Qb'=\{q_n | n \geqq 3\}$ then $A=B_1 s$ but $B_1 \notin I_0(B)$ because $B_1 \not\equiv B$. Note also that $A \neq B's$ for any other $B' \in I(B)$. ∎

The containments $I_n(A) \subseteqq I_n(B)s$ and $I_n(B) \subseteqq I_n(A)t$ in proposition 4.8 (2) cannot be replaced with equalities.

4.10. Example. Let $T=(Q, S)$ be a transformation semigroup where

$$Q = \{p_n | n \in N\} \cup \{q_n | n \in N\}$$

and

$$S = \langle a, b, c, x, s, t, a^{-1}, b^{-1}, c^{-1}, x^{-1}, s^{-1}, t^{-1} \rangle.$$

Define:

(1) $F_a(p_j) = p_j$ for $j \geqq 1$;

(2) $F_b(q_j) = q_j$ for $j \geqq 1$;

(3) $F_c(p_j) = p_{(j/2)}$ if j is even;

(4) $F_x(p_j) = p_{j+1}$ if j is odd;

(5) $F_s(q_j) = p_j$ for all $j \geqq 1$;

(6) $F_t(p_2) = q_1$, $F_t(p_{2j+1}) = q_{j+1}$ for $j \geqq 1$.

The partial functions induced by a^{-1}, b^{-1}, c^{-1}, s^{-1}, t^{-1}, and x^{-1} are defined in the obvious manner. For all other $y \in S$ and $q \in Q$, the expression $F_y(q)$ is undefined.

The sets $A=Qa=\{p_j | j \geqq 1\}$ and $B=Qb=\{q_j | j \geqq 1\}$ are strongly equivalent skeleton elements, with $A=Bs$ and $B=At$. Let $X=Ax=\{p_{2j} | j \geqq 1\}$. Then $X \subseteqq A$, $A \subseteqq Xc$, so that $X \in I_0(A)$ but $Xt=\{q_1\} \neq B$, so that $Xt \notin I_0(B)$. ∎

5. The decomposition theorem

We now develop the decomposition theorem. The basic idea for this decomposition is the same as in the finite case; we start off with a "coarse" decomposition and refine it until we get the result we desire. Throughout this section we follow closely the presentation in Holcombe [10, chapter 4].

5.1. Definition. (1) (Holcombe [10, p. 102]). Let $T=(Q, S)$ be a transformation semigroup. Let $\pi=\{H_j\}_{j\in I}$ be a collection of subsets of Q such that $Q=\bigcup_{j\in I}H_j$, where I is some indexing set for this collection. Then π is an **admissible subset system** if for any $i\in I$ and $s\in S$ there exists $j\in I$ such that $H_iF_s\subseteq H_j$.

(2) Let π, π' be two admissible subset systems. Then we say that $\pi'\leqq\pi$ if for every $H'\in\pi'$ there is some $H\in\pi$ such that $H'\subseteq H$.

5.2. Definition. Let $T=(Q, S)$ and $T'=(Q', S')$ be two transformation semigroups.

(1) (Holcombe [10, p. 43]). A partial function $\alpha\colon Q'\to Q$ is a **covering of T by T'** if

(a) α is surjective;

(b) for every $s\in S$ there is some $t_s\in S'$ such that either $\alpha(q')s$ is undefined or $\alpha(q')s=\alpha(q't_s)$ for every $q'\in Q'$.

We denote the fact that T' covers T by writing $T\leqq T'$.

(2) (Holcombe [10, p. 116]). A relation α on $Q'\times Q$ is called a **relational covering of T by T'** if

(a) α is surjective;

(b) for every $s\in S$ there is some $t_s\in S'$ such that $\alpha(q')s\subseteq\alpha(q't_s)$ for every $q'\in Q'$.

We denote the fact that α is a relational covering of T by T' by writing $T\leqq_\alpha T'$.

5.3. Definition. (Holcombe [10, p. 122]). Let $T=(Q, S)$ and $T'=(Q', S')$ be two transformation semigroups, where T has depth function d. Let α be a relational covering of T by T'. Then α has **rank** i (with respect to d) if

(1) $\alpha(q')\in I(T)$ for all $q'\in Q'$;

(2) $d(\alpha(q'))\geqq i$ for all $q'\in Q'$ and $d((\alpha(q'))=i$ for at least one $q'\in Q'$ where $0\leqq i\leqq d(T)$.

5.4. Definition. (Holcombe [10, p. 35]). Let $T=(Q, S)$ be a transformation semigroup. The **closure of** S is the set $\bar{S}=S\cup\{\bar{q}|q\in Q\}$ where, for each $q\in Q$, $\bar{q}$ is the constant map defined by $x\bar{q}=q$ for all $x\in Q$. The **closure of** T is $\bar{T}=(Q, \bar{S})$.

For each ordinal j, $0\leqq j\leqq d(T)$, we divide the set of skeleton elements of T at depth j into equivalence classes under the relation $\equiv$, and we take a set of representatives from these classes, say $A_1^j, A_2^j, \ldots$. We form the holonomy transformation semigroups for all A_k^j and take their join $T(A_1^j)\vee T(A_2^j)\vee\ldots$. This is denoted by $T_j^\vee(T)$, a transformation semigroup with state set denoted by $ST_j^\vee(T)$ and action semigroup denoted by $X_j^\vee(T)$. Note that the sets at depth 0 are all equivalent to Q, so we can choose $A_1^0=Q$, and hence $T_0^\vee(T)=T(Q)$. To ensure that the state sets of the $T(A_k^j)$'s are disjoint we will consider the state set of $T(A_k^j)$ to be $\{k\}\times ST(A_k^j)$ instead of $ST(A_k^j)$. Thus, a typical element of the state set of $T(A_k^j)$ is denoted (k, B_k^j) where $k\geqq 1$, $B_k^j\in ST(A_k^j)$.

The next definition generalizes a definition in Holcombe [10, p. 126].

5.5. Definition. For a transformation semigroup $T=(Q, S)$, the set π^j is $\{A\in I(T)|d(A)\geqq j\}$.

Note that π^j is an admissible subset system. We have $Q=\bigcup_{H\in\pi} jH$ because if $j\leqq d(T)$ then $d(\{q\})\geqq j$ for all $q\in Q$ and so $\{q\}\in\pi^j$. Also, if $B\in\pi^j$ and $s\in S$, then $d(Bs)\geqq d(B)\geqq j$ so that $Bs\in\pi^j$.

5.6. Theorem. Let $T=(Q, S)$ be a transformation semigroup of depth at least 1. Then there is a relational covering $T\leqq_\alpha \mathbf{T}_0^{\vee}(T)$ of rank 1.

Proof. By the argument of the previous paragraph, the set π^1 is an admissible subset system. To specify the covering, let $B'\in\pi^1$, let $s\in S$, and define

$$B' * s = \begin{cases} B's & \text{if } s\in T_0^{\vee}(T) \\ Qs & \text{otherwise.} \end{cases}$$

By proposition 4.2, if $s\notin T_0^{\vee}(T)$ then $Qs\not\equiv Q$, so that Qs is of depth 1 or greater. The pair (π^1, S) gives rise to the transformation semigroup $(\pi^1, S/\sim)$ where the congruence $\sim$ identifies any two elements of S which act identically on π^1. Denote $(\pi^1, S/\sim)$ by $T/\langle\pi^1\rangle$. If we define a relation $\alpha\colon \pi^1\to Q$ by $\alpha(B)=B$ for all $B\in\pi^1$ then we obtain a relational covering $T\leqq_\alpha T/\langle\pi^1\rangle$ of rank 1. We can in turn cover $T/\langle\pi^1\rangle$ by $\mathbf{T}_0^{\vee}(T)$. ∎

The proof of theorem 5.7 follows Holcombe [10, theorem 4.3.4].

5.7. Theorem. Let $T=(Q, S)$ and let d be a depth function. Let π be an admissible partition of rank j, where $j<d(T)$. Then there is an admissible subset system π' of rank $j+1$ with $\pi'\leqq\pi$.

Proof. Let $I_j(\pi)=\{A\in\pi|d(A)=j\}$, let $I_{j+}(\pi)=\{A\in\pi|d(A)>j\}$, and $I_{j++}(\pi)=$ $=\{A\in I(T)|A\in I_{1+}(Y)$ for some $Y\in I_j(\pi)\}$. Define $\pi'=I_{j+}(\pi)\cup I_{j++}(\pi)$. Then $\pi'\leqq\pi$ and rank $(\pi')=j+1$. We must show that π' is admissible.

We first show that $Q=\bigcup_{H\in\pi'} H$. Let $q\in Q$. If $q\in A\in L_{j+}(\pi)$ then there is nothing more to prove. If $q\notin A$ for any $A\in I_{1+}(\pi)$, then $q\in A$ for some $A\in I_j(\pi)$. But $d(\{q\})>j$ by definition so $\{q\}\in I_{j++}(\pi)$ by definition of $I_{j++}(\pi)$. Hence $Q=\bigcup_{H\in\pi'} H$.

Now let $B\in\pi'$, $s\in S$. We must show that $Bs\subseteqq A$ for some $A\in\pi'$. There are two main cases to consider.

Case I: $B\in I_{j+}(\pi)$. Then $B\in\pi$ so $Bs\subseteqq A$ for some $A\in\pi$ because π is admissible. There are two subcases to examine.

Subcase 1: $A\in I_{j+}(\pi)$. Then $Bs\subseteqq A\in\pi'$.

Subcase 2: $A\in I_j(\pi)$. Since $Bs\leqq B$ and $d(B)>j$ we have $d(Bs)>j$ and so $Bs\in I_{1+}(A)$ for $A\in I_j(\pi)$. Therefore Bs is an element of π'.

Case II: $B\in I_{j++}(\pi)$. Then $B\in I_{1+}(Y)$ for some $Y\in I_j(\pi)$ and $Bs\subseteqq Ys$. There are two subcases to consider.

Subcase 1: $Ys\subseteqq A$ for some $A\in I_j(\pi)$. Then $Bs\subseteqq A\in\pi'$.

Subcase 2: $Ys\subseteqq A$ for some $A\in I_j(\pi)$. Then $Bs\subseteqq Ys\subseteqq A$ and $d(Bs)\geqq d(B)>j$ so that $Bs\in I_{1+}(A)$. Hence $Bs\in\pi'$. ∎

5.8. Theorem. Let $T=(Q, S)$ and $T'=(Q', S')$ be transformation semigroups, and let $T\leqq_\alpha T'$ be a relational covering of rank j, $j<d(T)$, such that the image of α is π^j. Then there exists a relational covering $T\leqq_{\alpha'} \mathbf{T}_j^\vee(T)\circ T'$ such that
(1) the rank of α' is $j+1$;
(2) the image of α' is π^{j+1}.

Proof. Since $T\leqq_\alpha T'$, for every $s\in S$ there is some $t_s\in S'$ such that $\alpha(q')s=\alpha(q't_s)$ for all $q'\in Q'$. Let $A_1^j, \ldots, A_k^j, \ldots$ be a set of representatives of equivalence classes under $\equiv$ of skeleton elements in T of depth j. Recall that $ST_j^\vee(T)=\bigcup_{1\leqq k}(\{k\}\times ST(A_k^j))$ denotes the state set of $\mathbf{T}_j^\vee(T)$ and that $X_j^\vee(T)$ denotes the action semigroup of $\mathbf{T}_j^\vee(T)$.

To define the relation α' from $ST_j^\vee(T)\times Q'$ to Q consider an element $((k, B_k^j), q')\in \in K\times Q'$. If $d(\alpha(q'))=j$ then there is some A_k^j such that $\alpha(q')\equiv A_k^j$; in particular, $\alpha(q')\subseteqq A_k^j x$ for some $x\in S$. We define

$$\alpha'((k, B_k^j), q') = \begin{cases}\alpha(q') & \text{if } d(\alpha(q')) > j \\ B_k^j x\cap\alpha(q') & \text{if } \alpha(q') \equiv A_k^j, \quad \alpha(q') \subseteqq A_k^j x \\ \emptyset & \text{otherwise.}\end{cases}$$

By proposition 2.6, the image of α' is a skeleton element in all cases. Clearly, the rank of α' is greater than j.

We now show that the image of α' is π^{j+1}. Writing π^j as $I_j(\pi)\cup I_{j+}(\pi)$ as in theorem 5.7, we have $\pi^{j+1}=I_{j+}(\pi)\cup I_{j++}(\pi)$. Since the image of α is π^j, suppose that $Z\in\pi^{j+1}$. If $Z\in\pi^j$ then we have $d(Z)>j$ and $Z=\alpha(q')$ for some $q'\in Q'$ and so $\alpha'((k,B_k^j), q')=\alpha(q')=Z$ for any $(k,B_k^j)\in K$. If $Z\in I_{1'}(Y)$ for some $Y\in I_j(\pi)$ then $Y\in\pi^j$ and $Y=\alpha(q')$ for some $q'\in Q'$. Now $Y=\alpha(q')\equiv A_k^j$ for some $1\leqq k$, so that $Y\subseteqq A_k^j x$ for some $x\in S$. Then $Z=B_k^j x$ for some $B_k^j\in I_{1+}(A_k)$ by proposition 4.8, and also $Z\subseteqq Y=\alpha(q')$. Therefore $Z=B_k^j x\cap\alpha(q')=\alpha'((k, B_k^j), q')$. Hence the image of α' equals π^{j+1}.

We now prove that α' is a relational covering. The crucial part is the definition of the element of the action semigroup $\mathbf{T}_j^\vee(T)\circ T'$ which covers a given element of S.

Let $s\in S$ and suppose that t_s covers s with respect to the relational covering α. Thus $\alpha(q')s\subseteqq\alpha(q't_s)$ for all $q'\in Q'$. As before, $\mathbf{T}_j^\vee(T)$ denotes the closure of the join of all the $T(A_k^j)$ for $k\geqq 1$. Now the action semigroup of $\mathbf{T}_j^\vee(T)\circ T'$ consists of all ordered pairs (f, t) where $t\in S'$ and $f\colon Q'\to X_j^\vee(T)$. Having chosen our element $s\in S$ we define a function $f_s\colon Q'\to X_j^\vee(T)$ in the following way. Let $q'\in Q'$. Three possibilities arise:

Case I: $\alpha(q't_s)\in I_{j+}(\pi)$. Then f_s is chosen arbitrarily.

Case II: $\alpha(q't_s)\in I_j(\pi)$ and $\alpha(q')s\not\equiv\alpha(q't_s)$. Then $\alpha(q't_s)\equiv A_k^j$ for some $k\geqq 1$, so there is some $y\in S$ such that $\alpha(q't_s)\subseteqq A_k^j y$, so that $\alpha(q't_s)y^{-1}\subseteqq A_k^j$. Now $\alpha(q')sy^{-1}\leqq \leqq\alpha(q')s\not\equiv A_k^j$ and so $\alpha(q')sy^{-1}\subseteqq B'$ for some $B'\in I_{1+}(A_k^j)$. We put $f_s=C(B')$, the constant map which maps everything to B'.

Case III: $\alpha(q't_s)\in I_j(\pi)$ and $\alpha(q')s\equiv\alpha(q't_s)$. Then $\alpha(q')\equiv\alpha(q't_s)$ since $\alpha(q')s\leqq \leqq\alpha(q')$ and yet $\alpha(q')$ is of depth at least j. Now, as stated in the definition of

α', $A_k^j \equiv \alpha(q')$ implies that there is some $x \in S$ such that $\alpha(q') \subseteq A_k^j x$. Thus $A_k^j xs \supseteq \supseteq \alpha(q')s \equiv \alpha(q't_s)$.

Now $\alpha(q')s \equiv \alpha(q't_s)$ implies that $\alpha(q')s \subseteq \alpha(q't_s)x'$ for some $x' \in S$, so that in particular $\alpha(q')s(x')^{-1} \equiv \alpha(q')s$. Therefore $A_k^j xs(x')^{-1} \supseteq \alpha(q')s(x')^{-1} \equiv \alpha(q')s \equiv \alpha(q') \equiv \equiv A_k^j$ which implies that $A_k^j xs(x')^{-1} \equiv A_k^j$. It follows from proposition 4.2 that $xs(x')^{-1} \in T(A_k^j) \subseteq T_j^\vee(T)$, so we put

$$f_s(q') = xs(x')^{-1}.$$

This defines the function f_s: $Q' \to X_j^\vee(T)$. What remains is the task of showing that (f_s, t_s) covers s with respect to α'. Let $((l, B_l^j), q') \in K \times Q'$. We prove that

$$\alpha'((l, B_l^j), q')s \subseteq \alpha'(((l, B_l^j), q')(f_s, t_s)). \qquad (*)$$

Case I: $\alpha(q't_s) \in I_{j+}(\pi)$. If $\alpha(q') \in I_{j+}(\pi)$ then

$$\alpha'((l, B_l^j), q')s = \alpha(q')s \subseteq \alpha(q't_s).$$

In all other cases

$$\alpha'((l, B_l^j), q')s \subseteq \alpha(q')s \subseteq \alpha(q't_s).$$

Since $\alpha(q't_s) \in I_{j+}(\pi)$, we have that $f_s(q')$ is arbitrary and that

$$\alpha'(((l, B_l^j), q')(f_s, t_s)) = \alpha'((l, B_l^j), f_s(q'), q't_s) = \alpha(q't_s).$$

Therefore the inequality (*) holds in this case.

Case II: $\alpha(q't_s) \in I_j(\pi)$ and $\alpha(q')s \not\equiv \alpha(q't_s)$. Now $f_s(q') = C(B')$, where $B' \in I_{1+}(A_k^j)$ and $\alpha(q')sy^{-1} \subseteq B'$, where y is defined in Case II above to be the element such that $\alpha(q')s \subseteq \alpha(q't_s) \subseteq A_k^j y$. Therefore

$$\alpha'(((l, B_l^j), q')(f_s, t_s)) = \alpha((k, B'), q't_s) = B'y \cap \alpha(q't_s)$$

By definition of α', we have $\alpha'((l, B_l^j), q') \subseteq \alpha(q')$, so that

$$\alpha'((l, B_l^j), q')s \subseteq \alpha(q')s = \alpha(q')sy^{-1}y \subseteq B'y \cap \alpha(q't_s)$$

and so (*) holds again.

Case III: $\alpha(q't_s) \in I_j(\pi)$ and $\alpha(q')s \equiv \alpha(q't_s)$. If $\alpha(q') \equiv A_k^j$ then

$$\alpha'(((l, B_l^j), q')(f_s, t_s)) = \alpha'((l, B_l^j xs(x')^{-1}, q't_s) = B_l^j xs(x')^{-1}x' \cap \alpha(q't_s).$$

Recall that x is the element of S for which $\alpha(q') \subseteq A_k^j x$ and that x' is the element of S for which $\alpha(q't_s) \subseteq A_k^j x'$; this latter inequality implies that $\alpha(q't_s) \subseteq \mathrm{dom}\,(x')^{-1}$. In particular, $\alpha(q't_s)(x')^{-1}x' = \alpha(q't_s)$. Using these facts together with proposition 2.8 we can rewrite the last expression as

$$B_l^j xs(x')^{-1}x' \cap \alpha(q't_s)(x')^{-1}x' = (B_l^j xs \cap \alpha(q't_s))(x')^{-1}x' =$$
$$= B_l^j xs \cap \alpha(q't_s) \supseteq B_l^j xs \cap \alpha(q')s =$$
$$= (B_l^j x \cap \alpha(q'))s = \alpha'((1, B_l^j), q')s$$

and so (*) holds. Finally, if $\alpha(q') \not\equiv A_l^j$ then $\alpha'((1, B_l^j), q') = \emptyset$ and so

$$\alpha'((1, B_l^j), q')s \subseteq \alpha'(((1, B_l^j), q')(f_s, t_s))$$

as required. ∎

5.9. Main Decomposition Theorem. Let $T=(Q, S)$ be a transformation semigroup which is the quotient of a unique predecessor transformation semigroup, and let S satisfy ACC on cyclic left ideals. For each ordinal j, $0\leqq j\leqq d(T)$, let $A_1^j, \ldots, A_k^j, \ldots$ be a set of representatives of equivalence classes under $\equiv$ of skeleton elements of T of depth j. Then T is covered by a wreath product of transformation semigroups which are of one of the two following forms:

(1) (A_k^j, C_k^j) where C_k^j is the set of all constant maps from A_k^j to itself;

(2) $T_0^{\vee}(Q) = T(A_1^j)\vee\ldots\vee T(A_k^j)\vee\ldots$.

Further, if A stands for any A_k^j, then $T(A)=(ST(A), X(A))$, where

(a) $ST(A)=\{B|B\in I(T); B=B'x$ for some $B'\subseteqq A$, $B\not\equiv A$, and some $x\in X(A)\}$;

(b) $X(A)=(J(A)\cup G(S))/\sim$, where

(i) $J(A)$ is the ideal of S generated by the elements of S which induce a permutation on some $B\subseteqq A$, $B\equiv A$;

(ii) $G(S)$ is the group of units of S;

(iii) the congruence $\sim$ identifies elements of $J(A)\cup G(S)$ which act identically on $ST(A)$;

(iv) $X(A)$ is regular and satisfies ACC on cyclic left ideals.

Proof. Let $n=d(T)$ be the depth of the transformation semigroup T. We prove by transfinite induction that for every $1\leqq j\leqq n$ there is a relational covering $T\leqq_\alpha\ldots\circ\overline{\mathbf{T}}_1^{\vee}(T)\circ\overline{\mathbf{T}}_0^{\vee}(T)$ of depth j.

Base case $j=1$. This is theorem 5.6.

Inductive step. Assume that there is an ordinal J such that the theorem is true for all $j<J$, and assume that $j=J$. There are two cases to consider.

Case I: J is a non-limit ordinal. Then the result follows from theorem 5.6.

Case II: J is a limit ordinal. Note that $\pi^J=\bigcap_{k<J}\pi^k$. Therefore, the image of $\ldots\circ\overline{\mathbf{T}}_1^{\vee}(T)\circ\overline{\mathbf{T}}_0^{\vee}(T)$ is π^J, where the terms in this product are indexed by all the ordinals j such that $0\leqq j<J$. This proves the existence of the relational covering.

Now each $\overline{\mathbf{T}}_j^{\vee}(T)=T_j^{\vee}(T)\vee(K, C)$ where $K=\bigcup_{k\geqq 1}A_k^j$ and C is the set of constant maps on K. We can in turn decompose (K, C) as $(A_1^j, C_1^j)\vee\ldots\vee(A_k^j, C_k^j)\vee\vee\ldots$ and $T_j^{\vee}(T)$ as $T(A_1^j)\vee\ldots\vee T(A_k^j)\ldots$. The remainder of the theorem follows from proposition 4.5. ∎

If we assume that S has a composition series — that is, a sequence of two sided ideals $I_1\supset I_2\supset\ldots\supset I_n\supset\ldots$ such that each I_{j+1} is a maximal ideal in the semigroup I_j — then it seems possible to replace the factors $T(A_k^j)$ of the decomposition by simple semigroups. (The author has not checked this fact. For a possible method of proof of this conjecture, see Tucci [17].)

We conclude with two trivial examples to show that the decomposition can be either finite or infinite.

5.10. Example. Consider the transformation semigroup of example 2.2. Any skeleton element is either equivalent to Q or is a singleton. Hence the depth of the transformation semigroup T is 1, and so $T\leqq(Q, \overline{\mathbf{Z}})=\overline{\mathbf{T}}$; that is, the decomposition is trivial in this case. ∎

5.11. Example. Let $T=(Q, S)$ where $Q=\{q_n|n\in N\}$, and

$$S=\langle (x_j, x_j^{-1}, a_j, a_j^{-1}|j\in N)\rangle$$

where

(1) $F_{x_n}(q_n)=q_{n+1}$ for all $n\geqq 1$;

(2) $F_{a_k}(q_n)=q_n$ for all $1\leqq k\leqq n$.

The functions induced by all x_j^{-1} and a_j^{-1}, $j\geqq 1$, are defined in the obvious manner. For all other $s\in S$ and $q\in Q$, the expression $F_s(q)$ is undefined.

The skeleton elements of T are either singletons or of the form $A_n=\{q_j|j\geqq n$ for some integer $n\}$. Hence there is an infinite descending chain of non-equivalent skeleton elements $Q=A_1\supset A_2\supset\ldots$ which yields an infinite decomposition $T\leqq_\alpha\ldots$ $\ldots\circ\bar{\mathbf{T}}_1^{\vee}(T)\circ\bar{\mathbf{T}}_0^{\vee}(T)$. ∎

Acknowledgements

The author would like to thank the referee for his helpful comments which have led to substantial improvements in this paper. The author would also like to thank his colleagues, Jeff Connor and Irene Loomis, for several stimulating conversations on this paper.

References

[1] Arbib, M. A., *Algebraic Theory of Machines, Languages, and Semigroups,* Academic Press, New York, 1968.
[2] Bavel, Z., *Introduction to the Theory of Automata,* Reston Publishing Company, Inc., Reston, Va., 1983.
[3] Biermann, A., "A Fundamental Theorem for Real Time Programs", Technical Report, Dept. of Computer Science, Duke University, 1976.
[4] Clifford, A. H., and Preston, G. B., *The Algebraic Theory of Semigroups, Vol. I.,* Math. Surveys Am. Math. Soc., Providence, R. I., 1961.
[5] Eilenberg, S., *Automata, Languages and Machines, Vol. B,* Academic Press, New York, 1976.
[6] Ésik, Z., "On Identities Preserved by General Products of Algebras", *Acta Cybernetica,* Tom. 6, Fasc. 3, Szeged, 1983, pp. 285—289.
[7] Ésik, Z., "Varieties and General Products of Top-Down Algebras", *Acta Cybernetica,* Tom. 7, Fasc. 3, Szeged, 1986, pp. 293—298.
[8] Ésik, Z., and Gécseg, F., "General Products and Equational Classes of Automata", *Acta Cybernetica,* Tom. 6, Fasc. 3, Szeged, 1983, pp. 281—284.
[9] Gacs, P., "Reliable Computation with Cellular Automata", *Journal of Computer and Systems Sciences* v. 32, 1985, pp. 79—90.
[10] Holcombe, W. M. L., *Algebraic Automata Theory,* Cambridge University Press, 1982.
[11] Reeker, L., "State Graphs and Context Free Languages", *Proceedings of the International Symposium on Theory of Machines and Computations,* ed. by Kohlavi and Paz, Academic Press, 1971, pp. 143—151.
[12] Reeker, L. and Tucci, R. P., "Information and Algorithms", submitted to *Information Sciences.*
[13] Rhodes, J., "Infinite Iteration of Matrix Semigroups I. Structure Theorem for Torsion Semigroups", *Journal of Algebra* v. 98, 1986, pp. 422—451.
[14] Rhodes, J., "Infinite Iteration of Matrix Semigroups II. Structure Theorem for Arbitrary Semigroups up to Aperiodic Morphism", *Journal of Algebra* v. 100, 1986, pp. 25—137.

[15] Scott, D., "The Lattice of Flow Diagrams", *Symposium on the Semantics of Algorithmic Languages*, ed. by E. Engeler, Springer Verlag, Lecture Notes in Mathematics v. 188, 1971, pp. 309—372.
[16] Tucci, R. P., "A Note on the Decomposition of Infinite Automata", *International Journal of Computer Mathematics* 24 (1988), 141—149.
[17] Tucci, R. P., "A New Proof of the Krohn—Rhodes Theorem for Transformation Semigroups", *Applied Mathematics Letters*, to appear.
[18] Warne, R. J., "Embedding of Regular Semigroups in Wreath Products", *Journal of Pure and Applied Algebra* v. 29, 1983, pp. 177—207.
[19] Warne, R. J., "Embedding of Regular Semigroups in Wreath Products, II", *Journal of Pure and Applied Algebra* v. 40, 1986, pp. 185—207.

Received August 16, 1988

Two Transformations on Attribute Grammars Improving the Complexity of their Evaluation

ÉVA GOMBÁS and MIKLÓS BARTHA

A. JÓZSEF UNIVERSITY
BOLYAI INSTITUTE
SZEGED, ARADI VÉRTANÚK TERE 1.
6720 — HUNGARY

1. Introduction

Several papers have been written recently on designing efficient evaluators for attribute grammars (*AGs*). Some of these papers (e.g. [6], [7]) provide techniques to optimize the time complexity of the evaluators for certain classes of *AGs* (the class of absolutely noncircular *AGs* in the referenced papers), other ones (e.g. [5], [9]) try to reduce the storage requirement of the evaluators. The same goal of these papers is, however, to optimize evaluation by improving the evaluator itself in some respect. Our aim is to improve the *AG* to be evaluated — by the application of a suitable transformation — not the evaluator (of a fixed type) by which the evaluation is actually performed. Of course, this approach cannot provide general optimization results as the previous one, but in some cases it can be quite powerful. In this paper we present two transformation techniques and show how they work in restricted classes of *AGs*.

It is known that every *AG* can be converted to an equivalent one which uses only synthesized (s-)attributes. The underlying idea is the following: the value of a new s-attribute computed at any (nonterminal) node of a derivation tree becomes a function that describes how the corresponding old s-attribute depends on the old inherited (i)-attributes at the same node of the tree in the original *AG*. An exact algebraic formulation of this method, which will be referenced as the "convert to functional domains" (c.f.d.) principle, can be found e.g. in [2]. On the one hand it is clear that for an *AG* having s-attributes only, the structure of the evaluator is the simplest possible (only one left-to-right pass is needed). On the other hand, it is in general much more costly to deal with functional domains during evaluation, than to make multiple visits to the nodes of the derivation trees. Therefore, the c.f.d. principle cannot be used as a general transformation technique improving the complexity of evaluation. But it can be used successfully in a less drastical form for restricted classes of *AGs*. Indeed, both transformations presented in this paper are eventually applications of the c.f.d. principle.

The trick we are going to apply in the first transformation is based on the following well-known method of designing a one-pass assembler. If a post-definite label occurs in some instruction of the source program, then the assembler will translate an incomplete object code from that instruction, and it will update the adress field of this object code instruction (chained together with all those instructions referring to the same postdefinite label) as soon as the referenced label becomes definite. The term "reference to a post-definite label" corresponds to the term "reference to an *i*-attribute occurrence on the right-hand side of the semantic rules" in an *AG*, thus, we simplify evaluation by ignoring or postponing the computation of certain *i*-attribute instances at the nodes of the derivation trees. An example for this transformation is given in Sect. 2, and it is generalized in Sect. 4. To characterize the *AGs* for which the transformation is applicable, we introduce the class of *VSE AGs* in Sect. 3 and investigate the basic properties of this class. The class of *VSE AGs* is the visit-oriented counterpart of the class of *ASE AGs* [8], and it is in strong connection with the classes of *OAG* [10] and simple multi-visit *AGs* [3].

The second transformation technique is described in Sect. 5. It uses the c.f.d. principle with its full power, i.e. all the *i*-attributes are eliminated. The transformation can be applied, however, for a more restricted class of *AGs*, the class of linear string-valued *AGs*. By linearity we mean that in the Bochmann normal form of the semantic rules every attribute occurrence can be referenced at most once on the right-hand sides. This concept was originally defined for attributed tree transducers in [1].

In Sect. 4 we introduce two complexity measures for the evaluation of the complete derivation trees (*cd*-trees) of an *AG* under a fixed visit-oriented evaluator (cf. [9]). The visit complexity of a *cd*-tree t is the average number of visits made to a node during the evaluation of t. The pure computation complexity of t is the total amount of computation needed to assign value to all the attribute instances of t. The collection of pairs constructed from these two numbers for all the *cd*-trees, together with the type of the evaluator characterizes the evaluation complexity of the *AG* in a satisfactory way. We shall show that our transformations indeed reduce the complexity of evaluation in this sense.

2. Definitions and Examples

Although we assume familiarity with attribute grammars [11], we repeat some of the basic concepts here to fix our notations for the forthcoming sections.

An attribute grammar is a 5-tuple

$$\mathcal{G} = (G, A, v, \{D_a | a \in A\}, \{r_p | p \in P\}),$$

where

$G=(N, T, P, S')$ is a context-free grammar, called the underlying *CF*-grammar of $\mathcal{G}$. N and T denote the set of nonterminal and terminal symbols, respectively; P is the set of productions and $S' \in N$ is the start symbol. We assume that G is "augmented" by the top-production $S' \to S$ ($S \in N$ as well), so that S' does not appear in any other production. We shall write a production $p \in P$ in the form

$$p\colon\ F_0 \to w_0 F_1 w_1 \dots F_n w_n,$$

where $F_j \in N$ and w_j is a string of terminal symbols for each $j \in [0, n]$. For nonnegative integers k, l, $[k, l]$ denotes the set $\{k, k+1, \dots, l\}$; $[k]$ is a shorthand for $[1, k]$, as usual. Since terminal symbols play no essential role in attribute grammars, the above production will rather be written as p: $F_0 \to F_1 \dots F_n$. Accordingly, by a node of a derivation tree we always mean a nonterminal node.

$A = A_S \cup A_I$ is a finite set of attributes, $A_S \cap A_I = \emptyset$. The elements of A_S and A_I are called synthesized (s-) and inherited (i-) attributes, respectively.

v: $N \to 2^A$ is a mapping; if $a \in v(F)$, then we say that $F \in N$ has attribute a. $S(F)$ and $I(F)$ will denote the sets $v(F) \cap A_S$ and $v(F) \cap A_I$, respectively. We assume that every nonterminal has at least one attribute, S' has only s-attributes.

$\{D_a | a \in A\}$ is the family of attribute domains. An attribute $a \in A$ takes its value from the set D_a.

$\{r_p | p \in P\}$ is the family of semantic rules. If p: $F_0 \to F_1 \dots F_n$, then r_p consists of the following rules (equations):

$$a_0(F_{j_0}) = f(a_1(F_{j_1}), \dots, a_m(F_{j_m}))$$

for each

$$a_0 \in \begin{cases} S(F_0) & \text{if } j_0 = 0; \\ I(F_j) & \text{if } j = j_0 > 0, \end{cases}$$

where $j_i \in [0, n]$ and $a_i \in v(F_{j_i})$ for every $i \in [0, m]$; f: $D_{a_1} \times \dots \times D_{a_m} \to D_{a_0}$ is a (computable) function. The above rule will be abbreviated later on as $a_0(F_{j_0}) = rhs\ (a_0, F_{j_0})$. We say that $a_i(F_{j_i})$ is a definition or a reference to attribute occurrence a_i of nonterminal (occurrence) F_{j_i} in a rule corresponding to production p depending on whether it occurs on the left-hand side or right-hand side of the rule. If there are several occurrences of the same nonterminal in p, then these occurrences will be distinguished by subscripts, as usual. The condition that if $a_i(F_{j_i})$ is referred on the right-hand side in any rule of r_p, then

$$a_i \in \begin{cases} S(F_j) & \text{if } j = j_i > 0 \\ I(F_0) & \text{if } j_i = 0 \end{cases}$$

is the well-known Bochmann normal form (n.f.) condition. We shall violate this condition only if this makes the semantic rules shorter to write down.

The underlying idea of the following example is well-known from compiler literature (see e.g. [12]). We show how to compile a Boolean expression so that the length of the generated code depends only on the relations which the expression is built up from.

Example 2.1. Let $\mathcal{G}$ be the following AG. The underlying CF-grammar G has productions:

$$B' \to B;\ B \to B \text{ or } D | D;\ D \to D \text{ and } C | C;\ C \to \text{not}\, C | (B) | R,$$

where B', B, D, C and R are all the nonterminals with B' being the start symbol. (Note that the syntax satisfies both the LR-1 and operator precedence conditions.) Of course, G is incomplete in the sense that it is impossible to generate any terminal string using the above productions only. Therefore we assume that the grammar G is "continued" in such a way that the nonterminal R derives relations e.g. between arithmetic expressions. This part of the grammar is, however, not relevant from the point of view of our example. In this way G generates well-formed Boolean expressions, and by $\mathscr{G}$ we would like to translate these expressions to assembly language code. To this end we define the following attributes and corresponding domains:

code: string of assembly instructions, the generated code;
len: integer, the length of the generated code;
loc: integer, the location (or adress) of the first instruction of the generated code;
$\uparrow$: integer, the location where control should be passed if the corresponding Boolean expression is true;
$\downarrow$: integer, the location where control should be passed if the corresponding Boolean expression is false.

code and len are s-attributes, while loc, $\uparrow$ and $\downarrow$ are i-attributes. Every nonterminal, except B' has all these attributes, $v(B')=\{\text{code}\}$. The semantic rules corresponding to the productions are listed below.

$B'\to B$ code $(B')=$ code (B),
loc $(B)=l_0$, $\uparrow(B)=\uparrow_0$, $\downarrow(B)=\downarrow_0$

(l_0, $\uparrow_0$ and $\downarrow_0$ are constant locations).

$B_1\to B_2$ or D code $(B_1)=$code (B_2) code (D), len $(B_1)=$len $(B_2)+$len (D),
loc $(B_2)=$loc (B_1), loc $(D)=$loc $(B_1)+$len (B_2),
$\uparrow(B_2)=\uparrow(D)=\uparrow(B_1)$, $\downarrow(B_2)=$loc (D), $\downarrow(D)=\downarrow(B_1)$.
$D_1\to D_2$ and C code $(D_1)=$code (D_2) code (C), len $(D_1)=$len $(D_2)+$len (C),
loc $(D_2)=$loc (D_1), loc $(C)=$loc $(D_1)+$len (D_2),
$\uparrow(D_2)=$loc (C), $\uparrow(C)=\uparrow(D_1)$, $\downarrow(D_2)=\downarrow(C)=\downarrow(D_1)$.
$C_1\to$not C_2 code $(C_1)=$code (C_2), len $(C_1)=$len (C_2),
loc $(C_2)=$loc (C_1), $\uparrow(C_2)=\downarrow(C_1)$, $\downarrow(C_2)=\uparrow(C_1)$.

In the remaining four productions: $B\to D$, $D\to C$, $C\to(B)|R$ the value of the attributes is transferred without any change from one nonterminal to the other. □

Let t be a cd-tree of $\mathscr{G}$, and assume that the value code (u) of attribute instance code at any node u labelled by R is such a code that, when executed, it passes control to $\uparrow(u)$ or $\downarrow(u)$ depending on whether the corresponding relation below u is true or false, respectively. Then it is obvious from the semantic rules that this property is inherited by all the nodes of t. Clearly, the augmentation $B'\to B$ is not necessary in practice, because $\mathscr{G}$ is just a portion of a large AG defining the compiler semantics of a programming language, and the locations l_0, $\uparrow_0$, $\downarrow_0$ are inherited from the context.

To define dependency relations between the attribute occurrences of a production in an AG we assume that the family of domains $\{D_a|a\in A\}$ is extended to a many sorted algebra, which is called the attribute algebra of $\mathscr{G}$, and the functions f on the right-hand sides of the semantic rules are polynomials of the attribute algebra. Thus, if p: $F_0\to F_1...F_n\in P$, then we say that attribute occurrence $a(F_i)$ depends on $b(F_k)$ in p if there is an equation of the form $a(F_i)=f(...b(F_k)...)$ among the n.f. of the rules in r_p. The dependency graph for production p (denoted by $dp(p)$) is the graph having as nodes the attribute occurrences of all nonterminals F_j of p, $j\in[0, n]$, and in which there is an arc running from node $b(F_k)$ to node $a(F_i)$ iff $a(F_i)$ depends on $b(F_k)$ in p. See Fig. 1 for the dependency graph for production $B\to B$ or D of Example 2.1.

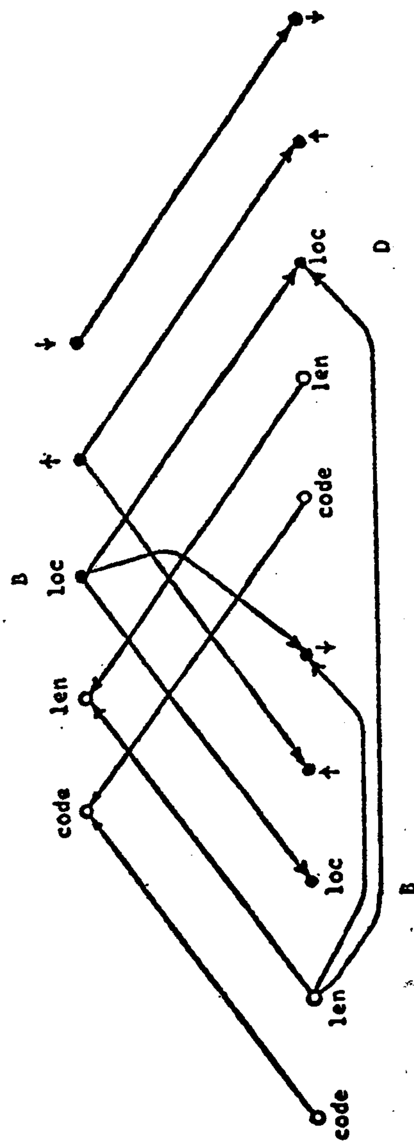


Fig. 1. The dependency graph for production $B \to B$ or D.

Several tree walking strategies exist for evaluating the *cd*-trees of an *AG*. The reader is assumed to be familiar with the notion of visit and pass, and with at least some of the papers [3], [4], [8], [10]. It can be seen directly from Fig. 1 that the *cd*-trees of our example *AG* $\mathcal{G}$ cannot be evaluated in one pass, nor in one visit ($\downarrow(B_2)$ depends on len (B_2) in the production $B_1 \to B_2$ or D). On the other hand it is clear that $\mathcal{G}$ satisfies the *ASE* property [8]. loc and len can be computed in the first left-to-right pass, and the remaining attributes in the second pass (which can be either left-to-right or right-to-left).

$\mathcal{G}$ can be transformed into an equivalent one-pass *AG* $\mathcal{G}'$ by the following trick. We drop the *i*-attributes $\uparrow$ and $\downarrow$, and compute the code of any sub-Boolean expression by leaving holes in the adress field of the crucial "jump to $\uparrow(R)$" and "jum to $\downarrow(R)$" instructions generated while compiling the relations contained in that subexpression. At the same time, we maintain two chains to register the locations of the "$\uparrow$-holes" and "$\downarrow$-holes" in the code. The holes will be filled in by the "old" values of $\uparrow$ and $\downarrow$ computed in $\mathcal{G}$ at the rootnode of the subexpression, but in $\mathcal{G}'$ we compute and fill in these values only later, when it becomes possible moving upwards in the derivation tree. The explicit construction is the following.

Example 2.2. The underlying *CF*-grammar of $\mathcal{G}'$ is the same grammar *G*, and it is equipped with the following attributes and corresponding domains:

$\langle$code, $\uparrow c$, $\downarrow c\rangle$:	a triple consisting of the generated code and two chains containing the locations of $\uparrow$-holes and $\downarrow$-holes in the code,
len:	integer, the length of the code,
loc:	integer, the location of the first instruction of the code.

Again, every nonterminal except B' has all these attributes, but now loc is the only *i*-attribute; $v(B')=\{\text{code}\}$. In fact code, $\uparrow c$ and $\downarrow c$ will be treated as three different *s*-attributes; we gathered them up just for the sake of the generalization we are going to introduce in Sect. 4. The semantic rules of $\mathcal{G}'$ are the following.

$B' \to B$ code $(B')=\textit{rollup}(\textit{rollup}(\text{code}(B), \uparrow c(B), \uparrow_0), \downarrow c(B), \downarrow_0)$,
loc $(B)=l_0$;

where *rollup* (w, c, x_0) is a function which substitutes iteratively a constant string x_0 into another string w at all the locations registered in a chain c.

$B_1 \to B_2$ or D code $(B_1)=\textit{rollup}(\text{code}(B_2), \downarrow c(B_2), \text{loc}(D))$ code (D),
$\uparrow c(B_1)=\uparrow c(B_2)\uparrow c(D)$, $\downarrow c(B_1)=\downarrow c(D)$, loc $(B_2)=$loc (B_1),
len $(B_1)=$len $(B_2)+$len (D), loc $(D)=$loc $(B_1)+$len (B_2).

$D_1 \to D_2$ and C code $(D_1)=\textit{rollup}(\text{code}(D_2), \uparrow c(D_2), \text{loc}(C))$ code (C),
$\uparrow c(D_1)=\uparrow c(C)$; $\downarrow c(D_1)=\downarrow c(D_2)\downarrow c(C)$, loc $(D_2)=$loc (D_1),
len $(D_1)=$len $(D_2)+$len (C), loc $(C)=$loc $(D_1)+$len (D_2).

$C_1 \to$ not C_2 code $(C_1)=$code (C_2), $\uparrow c(C_1)=\downarrow c(C_2)$, $\downarrow c(C_1)=\uparrow c(C_2)$,
len $(C_1)=$len (C_2), loc $(C_2)=$loc (C_1).

The rules corresponding to the remaining four productions are again "simple" rules. □

It is clear that $\mathcal{G}$ and $\mathcal{G}'$ are equivalent in the sense that they compute the same code for every Boolean expression. To compare the evaluation complexity of $\mathcal{G}$ and $\mathcal{G}'$ we make the following two observations,

a) $\mathcal{G}'$ is clearly one-pass.

b) In $\mathcal{G}'$ we have to compute the "old" values of attribute instances ↑ and ↓ only at certain nodes of a *cd*-tree (i.e. at exactly those points where a chain must be rolled up), and not both at all nodes as we do it in $\mathcal{G}$. For this reason we can say that, although the operations of maintaining the chains and rolling them up (which corresponds to a substitution of depth one) bring some extra cost into the evaluation, the total amount of computation needed to evaluate a *cd*-tree by $\mathcal{G}'$ is approximately the same as by $\mathcal{G}$.

Thus, by a) $\mathcal{G}'$ is more efficient than $\mathcal{G}$.

3. The Visit-Oriented Semantic Evaluator

We would like to extend the transformation technique outlined in Example 2.2 to a restricted subclass of simple multi-visit *AGs* (see [3]). The class that we are going to introduce — the class of *VSE AGs* — is the visit-oriented counterpart of the class of *ASE AGs* introduced in [8]. Those familiar with this work of Jazayeri and Walter know that it is not clear from the paper whether the authors mean the *ASE* property in a global sense, i.e. for all the attributes, or in a local sense, i.e. for all the attribute occurrences of the semantic rules. We assume here that they mean it in the global sense. Anyway, this question is not too relevant, and Proposition 3.2 shows the obvious connection between the two alternatives. The local version of the *ASE* property was redefined in [4] in a generalized form, with the new name simple multi-*ALT*.

The *VSE* Property.

In the sequel let $\mathcal{G}=(G, A, v, \{D_a | a\in A\}, \{r_p | p\in P\})$ with $G=(N, T, P, S')$ be a fixed *AG*.

Definition 3.1. The localized grammar of $\mathcal{G}$ is the *AG*

$$\mathrm{lc}(\mathcal{G}) = (G, A', v', \{D_{a'} | a'\in A'\}, \{r'_p | p\in P\}),$$

where

— $A' = \{(a, F)\in A\times N | F\in N,\ a\in v(F)\}$;

— $v'(F) = \{(a, F) | a\in v(F)\}$ for all $F\in N$;

— $D_{(a,F)} = D_a$ for all $F\in N$, $a\in v(F)$;

— if $p\in P$ is of the usual form and

$$a_0(F_{j_0}) = f(\dots a_i(F_{j_i})\dots)$$

is in r_p, then there exists a corresponding rule in r'_p of the form:

$$(a_0, F_{j_0})(F_{j_0}) = f(\dots(a_i, F_{j_i})(F_{j_i})\dots)$$

and r'_p consists of exactly these rules. □

Proposition 3.2. $\mathcal{G}$ is strictly alternating simple multi-*ALT* iff $lc(\mathcal{G})$ is *ASE*.

Proof. Obvious. □

Let B be a finite set. By an ordered partition of B we mean a finite sequence $(B_1, ..., B_m)$ of subsets of B such that $\bigcup_{i=1}^{m} B_i = B$. (Note that any of the B_i might be $\emptyset$.) Recall from [3] that a set of ordered partitions for $\mathcal{G}$ is a set Π containing an ordered partition $\pi(F)$ of $v(F)$ for each $F \in N$. Π is called a simple multi-visit (*smv*) set of (ordered) partitions if for every *cd*-tree there is a computation sequence (cf. [3]) for it respecting Π. $\mathcal{G}$ is *smv* if there exists an *smv* set of partitions for it.

Definition 3.3. Let Π be an *smv* set of partitions for $\mathcal{G}$ and $\varphi = (A_1, ..., A_m)$ an ordered partition of A. $\mathcal{G}$ is *m-VSE* with respect to (w.r.t.) (Π, φ) if the following condition holds. For every $F \in N$, if $\pi(F) = (B_1, ..., B_{k_F})$, then there exists an injective and monotonic mapping $\varrho_F\colon [k_F] \to [m]$ such that $B_k \subseteq A_{\varrho_F(k)}$ for each $k \in [k_F]$. That is, $\pi(F)$ is the projection of φ to $v(F)$. $\mathcal{G}$ is *(m-)VSE* if it is *(m-)VSE* w.r.t. some (Π, φ). In this case φ is called a *VSE* partition for $\mathcal{G}$. □

Example 3.4. The *AG* of Example 2.1 is 2-*VSE* w.r.t. (Π, φ), where

$$\varphi = (\{\text{ioc}, \text{ien}\}, \{\uparrow, \downarrow, \text{code}\}),$$

and for each $F \in N \setminus \{B'\}$, $\pi(F) = \varphi$; $\pi(B') = (\{\text{code}\})$. □

Let $\mathcal{G}$ be *m-VSE* w.r.t. (Π, φ). The sets B_k, $k \in [k_F]$ in $\pi(F) = (B_1, ..., B_{k_F})$ are called the local visit-sets of F in contrast with the "global" visit-sets $A_1, ..., A_m$ in φ. If $\varrho_F(k) = c$, then we shall say that c is the global visit-number (*gv*-number) of the local visit-set $B_k(F)$. Let $p\colon F_0 \to F_1 ... F_n \in P$ with

$$\pi(F_j) = (B_1^j, ..., B_{k_{F_j}}^j)$$

for each $j \in [0, n]$. The visit sequence (see [3]) of the visit-set $B_k^0(F_0)$, $k \in [k_{F_0}]$ in p (denoted by $Vs_p(B_k^0)$) is a concrete list of descendant visit-sets, i.e. a list consisting of some visit-sets $B_l^j(F_j)$, $j \geq 1$. Let c be the *gv*-number of $B_k^0(F_0)$. We associate a global visit sequence $Gvs_p(c)$ with $Vs_p(B_k^0)$ in a natural way: $Gvs_p(c)$ is a list of pairs of integers such that to any member $B_l^j(F_j)$ on $Vs_p(B_k^0)$ there corresponds a member (d, j) on $Gvs_p(c)$ (at the same position, of course), where d is the *gv*-number of $B_l^j(F_j)$. In this way we can consider Gvs_p as a vector of m lists. For each $c \in [m]$, if $c = \varrho_{F_0}(k)$ for some $k \in [k_{F_0}]$, then $Gvs_p(c)$ is the above list, otherwise $Gvs_p(c)$ is the empty list.

The *cd*-trees of $\mathcal{G}$ can now be evaluated by the help of the following procedure:

procedure *visit* (c, u); *integer c; node u;*
comment c is a global visit-number;
comment let $p\colon F_0 \to F_1 ... F_n$ be the production applied at u;
begin
  compute the instances of $I(F_0) \cap A_c$ at node u;
  for $i = 1$ *to length* $(Gvs_p(c))$ *do*
  begin
    comment take the i-th member of the list $Gvs_p(c)$;
    $(d, j) = take\,(i, Gvs_p(c))$;
    comment make a visit to the j-th son of u;
    visit $(d, son\,(j, u))$
  end;
  compute the instances of $S(F_0) \cap A_c$ at node u
end

The part of the "main program" which evaluates a *cd*-tree t can be written as:
for $c=1$ *to* m *do*
    visit $(c, root(t))$.

The main point of the procedure visit above is that we evaluate the local visit-sets at each node as being the projections of the corresponding known global visit-sets. This makes the procedure so simple compared with e.g. the simple multi-visit evaluation procedure in [4].

There are some situations when it is more appropriate to compute the final value of certain attribute instances by several assignments placed in different visits. To handle such situations we allow some instances of $v(F_0)\cap A_c$ at node u to be "marked" in the procedure *visit* (c, u) above. The value of these attribute instances can be updated later by the call of the following procedure.

procedure *update* (c, u); *integer c; node u;*
comment let F_0 be the label of u;
begin
    recompute the marked instances of $v(F_0)\cap A_c$ at node u;
    modify the present
    marking for the sake of further updates to u, if necessary;
end

Since the procedure *update* can be considered to be a visit of depth zero, we assume that update visits are also placed as distinguished members onto the global visit-sequence lists. Update visits will be used in the next section.

The following two definitions are adopted from [3]. An ordered partition $\pi=(B_1, \dots, B_m)$ of a subset of A is reduced if $B_k\neq\emptyset$ for any $k\in[m]$. π is good if, whenever $m\geqq 2$, B_1 contains at least one *s*-attribute, B_m contains at least one *i*-attribute and for every $k\in[2, m-1]$, B_k contains both *i*- and *s*-attributes.

Lemma 3.5. If $\mathcal{G}$ is *VSE* w.r.t. (Π, φ), then φ can be assumed to be reduced.

Proof. The result is a direct consequence of Lemma 2.1 in [3]. □

Theorem 3.6. If $\mathcal{G}$ is VSE w.r.t. (Π, φ), then φ can be assumed to be good.

Proof. By Lemma 3.5 we can assume that $\varphi=(A_1, \dots, A_m)$ is reduced. Moreover, we can assume that Π is also reduced, i.e. $\pi(F)$ is reduced for every $F\in N$. Suppose first that A_c contains only *s*-attributes for some $c\in[2, m]$. Let F be a nonterminal such that $\pi(F)=(B_1, \dots, B_{k_F})$ and there exists $k\in[k_F]$ with $c=\varrho_F(k)$. Define

$$\pi'(F) = \begin{cases} \pi(F) & \text{if} \quad k=1 \quad \text{or} \quad \varrho_F(k-1) < c-1, \\ (B_1, \dots, B_{k-1}\cup B_k, \dots, B_{k_F}) & \text{if} \quad \varrho_F(k-1) = c-1. \end{cases}$$

By virtue of Theorem 2.2 in [3], the replacement of $\pi(F)$ by $\pi'(F)$ in Π preserves the *smv* property. Thus, making this replacement for all appropriate $F\in N$ we get a set Π' which is still *smv* and, together with $\varphi'=(A_1, \dots, A_{c-1}\cup A_c, \dots, A_m)$ it satisfies the *VSE* condition. The same argument shows that, if A_c contains only *i*-attributes for some $c\in[m-1]$, then the partition $\varphi''=(A_1, \dots, A_c\cup A_{c+1}, \dots, A_m)$ together with its projections $\{\pi''(F)|F\in N\}$ remains *VSE*. In this way it is clear that, applying a finite number of such transformations on φ and Π, we shall end up with a good ordered partition $\tilde{\varphi}$. It must be noted, however, that the corresponding set of *smv* partitions $\tilde{\Pi}$ need not be good at all. □

Testing the *VSE* Property.

Lemma 3.7. $\mathcal{G}$ is *smv* iff $lc(\mathcal{G})$ is *VSE*.

Proof. Obvious. □

Theorem 3.8. The following problems are *NP*-complete:
(i) deciding whether an arbitrary *AG* is *VSE*,
(ii) deciding whether an arbitrary *AG* is 2-*VSE*.

Proof. (i) is an immediate consequence of Lemma 3.7 and Theorem 4.1 in [3], because the size of $lc(\mathcal{G})$ is polynomially related to the size of $\mathcal{G}$. To prove (ii) it is enough to observe that the example *AG* $\mathcal{G}(F_0)$ constructed in the proof of Theorem 4.1 in [3] for a Boolean expression F_0 is simple 2-visit iff $lc(\mathcal{G}(F_0))$ is 2-*VSE*. (Of course, this is not true for an arbitrary *AG*.) □

In spite of these negative results it is worth computing the relation "forced before" (see [3]) of the attributes as it was done also in [10]. Let $r \subseteq A \times A$ be any relation and $p\colon F_0 \to F_1 \ldots F_n \in P$. Define the graph $idp(p, r)$ to be an extension of $dp(p)$ with the arcs

$$a(F_j) \to b(F_j) \quad \text{iff } arb$$

for all $j \in [0, n]$, and let $idp(p, r)^+|F_j$ denote the restriction of the transitive closure of $idp(p, r)$ to the attribute occurrences of the j-th nonterminal (occurrence) in p. Then the relation forced before is the smallest relation $fb \subseteq A \times A$ such that

$$idp(p, fb)^+|F_j \subseteq fb|v(F_j)$$

for all productions $p\colon F_0 \to F_1 \ldots F_n$ and each $j \in [0, n]$. Clearly, fb must be respected by any *VSE* partition φ for $\mathcal{G}$. It is easy to give an algorithm which computes the relation fb in polynomial time (see [10] for a similar construction). When fb is computed and it is cycle free, then we can either attempt to construct a *VSE* partition φ directly, as it was done in [10], or to design a more complicated backtrack algorithm to search for a suitable φ. We must know, however by Theorem 3.8, that a really good backtrack algorithm will presumably have exponential time complexity.

4. Improving the evaluation of VSE AGs

We start this section by introducing two complexity measures for the evaluation of the *cd*-trees of an *AG*. For a *cd*-tree t define the visit complexity of t to be the ratio of the total number of visits to the nodes of t during a concrete visit-oriented evaluation (recall that every noncircular *AG* is at least pure multi-visit), and the number of (nonterminal) nodes of t. The pure computation complexity of t is the total amount of computation needed to assign value to all the attribute instances of t (the number of visits is irrelevant here). Suppose that $\mathcal{G}$ and $\mathcal{G}'$ are two equivalent *AG*, i.e. they have the same underlying *CF*-grammar and they compute the same values at the root S' of every *cd*-tree; just by the help of different sets of attributes. To compare the efficacy of $\mathcal{G}$ and $\mathcal{G}'$ we have to consider three points.

1. The evaluator applied for $\mathcal{G}$ and $\mathcal{G}'$ (i.e. pure multivisit, simple multi-visit, *ASE*, *VSE*, etc.).
2. The visit complexity of each *cd*-tree in $\mathcal{G}$ and $\mathcal{G}'$.
3. The pure computation complexity of each *cd*-tree in $\mathcal{G}$ and $\mathcal{G}'$.

We say that $\mathcal{G}'$ is more efficient than $\mathcal{G}$ if $\mathcal{G}'$ is not worse than $\mathcal{G}$ in any of the above three respects, and it is strictly better in at least one of them. From this point of view the *AG* $\mathcal{G}'$ of Example 2.2 is indeed more efficient than the *AG* $\mathcal{G}$ of Example 2.1, because of the reasons a) and b) explained at the end of Sect. 2.

To generalize the transformation technique described in Example 2.2 assume that $\mathcal{G}$ is *m-VSE* w.r.t. (Π, φ), furthermore it satisfies the following three conditions.

(C1) For every $p \in P$ and $c \in [m]$, if $(d; j)$ is a member of $Gvs_p(c)$, then $d \leqq c$.

(C2) There exists a distinguished *gv*-number $v \in [2, m]$ with the following property. Let p: $F_0 \to F_1 \ldots F_n$ be any production, and suppose that F_0 has a local visit set $B(F_0)$ the *gv*-number of which is v. If $C(F_j)$, $j \in [n]$, is a member of $Vs_p(B)$ such that some *i*-attribute occurrence $b(F_j) \in C(F_j)$ depends on an attribute occurrence $b'(F_l) \in D(F_l)$ in p, then the *gv*-number of $D(F_l)$ is not equal to (or equivalently, it is strictly less than) v, except when all the three conditions below are satisfied:

a) $D(F_l) \equiv B(F_0)$,
b) the *gv*-number of $C(F_j)$ is v,
c) b' is also an *i*-attribute and the semantic rule for $b(F_j)$ is: $b(F_j) = b'(F_0)$.

We shall say that such an exception rule is a simple rule.

(C3) $A_v \cap A_S$ *contains only string-valued attributes.*

Construction of the simplified *AG* $\mathcal{G}'$.

On the analogy of Example 2.2 we define the *AG*

$$\mathcal{G}' = (G, A', v', \{D'_a | a \in A'\}, \{r'_p | p \in P\})$$

as follows:

$A' = A \setminus A_v \cup A'_v$, where if

$$A_v \cap A_s = \{\alpha_1, \ldots, \alpha_{s_v}\} \quad \text{and} \quad A_v \cap A_I = \{\beta_1, \ldots, \beta_{i_v}\},$$

then

$$A'_v = \{(\alpha_y, c_1, \ldots, c_{i_v}) | y \in [s_v]\} \cup \{\beta_1, \ldots, \beta_{i_v}\}.$$

$A'_I = A_I$, $A'_S = A' \setminus A'_I$. In the "chained" *s*-attribute $(\alpha_y, c_1, \ldots, c_{i_v})$, c_z $(z \in [i_v])$ represents the chain of those locations which point to the "β_z-holes" in the string corresponding to attribute α_y. Note that by Theorem 3.6 we can assume that $i_v \geqq 1$.

For any $F \in N$ we first define the set

$$v''(F) = v(F) \setminus A_v \cup \{(\alpha_y, c_1, \ldots, c_{i_v}) | y \in [s_v],\ \alpha_y \in v(F)\},$$

then define

$$v'(F) = \begin{cases} v''(F) & \text{if } v \text{ is the greatest } gv\text{-number at } F, \\ v''(F) \cup \{\beta_z | z \in [i_v],\ \beta_z \in v(F)\} & \text{otherwise.} \end{cases}$$

That is, we supply F with the i-attributes of $A_v \cap v(F)$ iff there exists a local visit-set of F with gv-number greater than v. In fact, these attributes will always be evaluated in the $(v+1)$-th global visit.

If $a \in A \cap A'$ then $D'_a = D_a$, else (i.e. if a is a chained s-attribute $(\alpha_y, c_1, \ldots, c_{i_v})$) D'_a is the cartesian product of D_{α_y} and i_v chains (strings) of integer locations.

Let $p\colon F_0 \to F_1 \ldots F_n \in P$ and consider a rule

$$r\colon\ a_0(F_{j_0}) = rhs\,(a_0, F_{j_0})$$

in r_p (now we assume that r_p is in n.f.). To construct the corresponding rule r' in r'_p we distinguish two cases.

Case (a). $a_0(F_{j_0}) = \alpha_x(F_0)$ for some $x \in [s_v]$. In this case the left-hand side of r' is $(\alpha_x, c_1, \ldots, c_{i_v})(F_0)$, and the right-hand side of r' is obtained from $rhs\,(a_0, F_{j_0})$ by

(i) replacing any reference to an s-attribute $\alpha_y(F_j)$, $j \geqq 1$ by

$$(*)\qquad rollup\,\big(\alpha_y(F_j), c_{z_1}(F_j), rhs\,(\beta_{z_1}, F_j), \ldots, c_{z_l}(F_j), rhs\,(\beta_{z_l}, F_j)\big),$$

where $z_1, \ldots, z_l$ are all the numbers z such that the (existing) rule in r_p defining β_z is not a simple rule. The function *rollup* is the obvious generalization of the one used in Example 2.2, allowing several chains to be rolled up at the same call.

(ii) Ignoring (i.e. replacing with marked holes) all the references to the i-attributes $\beta_z(F_0)$, $z \in [i_v]$, and adjusting the value of the chains in $(\alpha_x, c_1, \ldots, c_{i_v})(F_0)$ in an appropriate way (obvious details are omitted).

Case (b). $a_0(F_{j_0}) \neq \alpha_x(F_0)$ for any $x \in [s_v]$. In this case r' is of the form

$$r'\colon\ a_0(F_{j_0}) = rhs'(a_0, F_{j_0}),$$

where $rhs'(a_0, F_{j_0})$ is obtained from $rhs\,(a_0, F_{j_0})$ by replacing any reference to an s-attribute $\alpha_y(F_j)$ $(j \geqq 1)$ by

$$(**)\qquad rollup\,\big(\alpha_y(F_j), c_1(F_j), rhs'(\beta_1, F_j), \ldots, c_{i_v}(F_j), rhs'(\beta_{i_v}, F_j)\big).$$

(Note that all the chains are rolled up here.)

Although we construct a "corresponding" r' for each rule r in r_p (this is necessary because of the recursion in $(**)$ above), r'_p should contain only those rules r' which define correct attribute occurrences in $\mathscr{G}'$ (an occurrence $a_0(F_{j_0})$ is correct in $\mathscr{G}'$ if $a_0 \in v'(F_{j_0})$).

Proposition 4.1. $\mathscr{G}'$ is correct and it is equivalent to $\mathscr{G}$.

Proof. By the correctness of $\mathscr{G}'$ we mean that all the attribute occurrences in the semantic rules r' of $\mathscr{G}'$ are correct. The left-hand side of the rules is clearly correct by construction, so we only have to prove that the references on the right-hand sides are also correct. In Case (a) it is enough to check that the expressions $rhs\,(\beta_z, F_j)$ in $(*)$ $(z \in \{z_1, \ldots, z_l\})$ do not refer to incorrect attribute occurrences. Indeed, in this case $Gvs_p(v)$ contains (v, j) by condition (C1), hence by (C2) $rhs\,(\beta_z, F_j)$ is always correct (note that the rules of $\mathscr{G}$ are in n.f., as we assumed). In Case (b) observe that, by (C1) and (C2), if $a_0(F_{j_0})$ is correct on the left-hand side of r', then $a_0(F_{j_0})$ is computed in such a local visit of F_0 which follows the one with gv-number v.

Consequently, F_0 has β_z in $\mathscr{G}'$ for each $z \in [i_v]$ by the construction of v'. We have to consider the expressions *rhs'* (β_z, F_j) in $(**)$. Two subcases are possible.

(i) The rule defining $\beta_z(F_j)$ in r_p is a simple rule. In this case *rhs'* (β_z, F_j) is correct by the above observation.

(ii) The rule defining $B_z(F_j)$ is not a simple rule. An easy inductive argument shows that *rhs'* (β_z, F_j) contains only correct attribute occurrences.

Note that the semantic rules of $\mathscr{G}'$ are also in n.f. The rest of the proof, i.e. that $\mathscr{G}$ and $\mathscr{G}'$ are equivalent, is left to the reader. □

Proposition 4.2. $\mathscr{G}'$ is $(m-1)$-*VSE*.

Proof. Let

$$\varphi' = (A_1, \ldots, A_{v-1} \cup (A'_v \cap A'_S), A_{v+1} \cup (A'_v \cap A'_I), \ldots, A_m),$$

and for each $F \in N$ let $\pi'(F)$ be the projection of φ' to $v'(F)$. In the proof of Proposition 4.1 we observed already that only those rules refer to occurrences of i-attributes β_z $(z \in [i_v])$ in $\mathscr{G}'$ which define occurrences of attributes computed in local visits following the ones with *gv*-number v in $\mathscr{G}$. This shows that Π' is also an *smv* set of partitions, thus $\mathscr{G}'$ is *VSE* w.r.t. (Π', φ'). □

Evaluating the *AG* $\mathscr{G}'$.

We must admit that, although the reduction achieved in the visit complexity, the pure computation complexity of the *cd*-trees has increased. This is due to the fact that, while rolling up chains we have to recompute the "old" value of certain i-attribute occurrences several times. To solve the problem we use update visits introduced in Sect. 3. To this end we put the attributes β_z, $z \in [i_v]$, forward into the joint global visit-set $A'_v \cap A_{v-1}$ of φ' (and, of course, into the corresponding joint local visit-set of each nonterminal, too). However, at the call *visit* $(v-1, u)$ to a node u we do not compute the instances of these i-attributes (or just give some unimportant initial value to them), but mark them together with all the instances of the chained s-attributes belonging to A'_v at u. Then, *update* $(v-1, u)$ should be called instead of the first *rollup* call of type $(**)$ — detailed in Case (b) of the construction of the rules of $\mathscr{G}'$ — for a chained s-attribute instance α_y at node u. Note that this *update* call can always be designed as a fixed member in that global visit-sequence of the father of u which contains also the pair $(v-1, j)$ corresponding to the call *visit* $(v-1, u)$. Then, every further *rollup* call for a chained s-attribute instance at u can be replaced by a simple reference to the corresponding "old" s-attribute instance. The problem of recomputation is not solved completely, however, because we did not handle *rollup* calls of type $(*)$ in Case (a). This would require a more sophisticated marking procedure for the *update* visits, the details of which is left to the reader.

Theorem 4.3. $\mathscr{G}'$ is (generally) more efficient than $\mathscr{G}$.

Proof. By Propositions 4.1 and 4.2, $\mathscr{G}$ and $\mathscr{G}'$ are equivalent, and they are both *VSE*. Once we have eliminated recomputation, we can say that the pure computation complexity of every *cd*-tree is approximately the same in both *AG*. It would not be honest, however, to state categorically that, by Proposition 4.2, the visit complexity of the *cd*-trees in $\mathcal{G}'$ is less than that in $\mathcal{G}$. To be exact, an *update* visit is also a visit, although it concerns only one node of a derivation tree. There are extreme examples of *cd*-trees where an *update* visit must be made to every node of the tree. Such an example is illustrated in Fig. 2. Circles represent local visit-sets in the graph of the figure, and the *gv*-number of the visit-sets is written inside the circles. Members of the visit-sequences are represented as descendants of the corresponding circles. It can be seen that the visit complexity of such kind of *cd*-trees remains the same in $\mathcal{G}'$. But, taking into account all the *cd*-trees of the grammar we can in general say that $\mathcal{G}'$ is indeed better than $\mathcal{G}$ from the point of view of visit complexity. This is always the case when e.g. v is the greatest *gv*-number, like in Example 2.1.

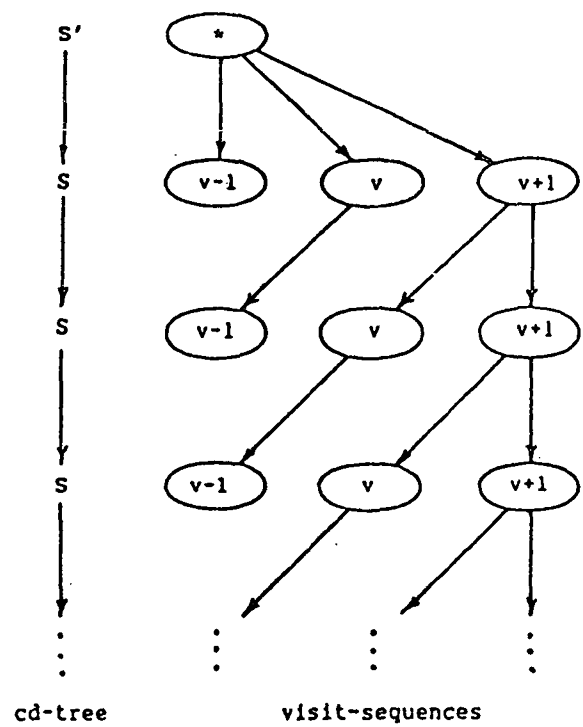


Fig. 2. An extreme example.

5. Improving the Evaluation of Linear String-valued AGs

Our second transformation technique eliminates all the *i*-attributes of $\mathcal{G}$, therefore it is more powerful than the chaining technique. This transformation can be applied, however, for a more restricted class of *AGs*, the class of linear string-valued *AGs*. Although we shall use the c.f.d. principle here with its full power, the attribute domains (as well as the algebra!) except for one attribute will not be changed.

In this section $\mathcal{G}$ will be a purely string-valued *AG*. By this we mean that the domain of all the attributes is Σ^*, the set of all strings over Σ, for some finite alphabet Σ, and the only operation applied in the semantic rules is concatenating strings. As usual, Σ^* denotes also the free monoid generated by Σ, thus, we can say that the attribute algebra of $\mathcal{G}$ is Σ^*. Dealing with polynomials over Σ^* we shall use different copies of the set $Z=\{z_1, ..., z_m, ...\}$ as variable symbols (e.g. $X=\{x_1, ..., x_m, ...\}$, $Y=\{y_1, ..., y_m, ...\}$). Z_m will denote the set $\{z_1, ..., z_m\}$. These sets of variables are assumed to be disjoint from Σ. For simplicity assume that every nonterminal, except S', has the same set of attributes consisting of k synthesized and l inherited attributes, and these attributes are numbered from 1 to k and from 1 to l, respectively. S' has only *s*-attributes 1, 2, ..., k. (The name of the attributes is irrelevant in this section.) Now, since the right-hand side of the semantic rules are polynomials over Σ^*, and these polynomials can also be represented as strings in $(\Sigma\cup Z)^*$, the semantic rules r_p corresponding to production p: $F_0\to F_1...F_n$ can be condensed into a sequence of $k+l\cdot n$ strings:

$$\alpha(p)\in((\Sigma\cup X_{k\cdot n}\cup Y_l)^*)^{k+l\cdot n},$$

where each occurrence of any variable in $\alpha(p)$ corresponds to a reference to an appropriate attribute occurrence: any occurrence of $x_{k\cdot(j-1)+i}$, where $j\in[n]$ and $i\in[k]$ is a reference to the i-th *s*-attribute occurrence of F_j, while y_r $(r\in[l])$ corresponds to the r-th *i*-attribute occurrence of F_0. The first k components of $\alpha(p)$ define the *s*-attribute occurrences of F_0, and the following components define the *i*-attribute occurrences of $F_1, ..., F_n$ from left to right in n segments each containing l components. Observe that r_p is in n.f.

Definition 5.1. A sequence of strings $\gamma\in((\Sigma\cup Z_m)^*)^s$ (s is a nonnegative integer) is linear if each variable occurs at most once in γ. $\mathcal{G}$ is linear if $\alpha(p)$ is linear for every $p\in P$.

Let Dt (F) denote the set of all derivation trees with root $F\in N$. It is clear that — provided $\mathcal{G}$ is noncircular — the value of all the attribute instances of the nodes of a tree $t\in$Dt (F) is uniquely determined by fixing the values of the *i*-attribute instances at the root of t. Let us fix these values to $y_1, ..., y_l$, respectively ($y_1, ..., y_l$ are variable symbols, as we agreed), and compute the value of the *s*-attribute instances at the root with the attribute domains enlarged to $(\Sigma\cup Y_l)^*$ during this computation. We obtain a sequence:

$$\beta(t)\in((\Sigma\cup Y_l)^*)^k.$$

Clearly, $\beta(t)$ represents the sequence of polynomials that describe how the *s*-attributes depend on the *i*-attributes at the root of t. It is important to note that if $\mathcal{G}$ is linear, then it is essentially noncircular. By this we mean that, although there might be circles in the dependency graph of a derivation tree (cf. [3]), these circles are "self-contained", i.e. they do not bother the computation of the attribute instances of the root. (In other words, the attribute instances contained in the circles are always useless.) For this reason, if $\mathcal{G}$ is linear, then $\beta(t)$ always exists, moreover, it is easy to see that $\beta(t)$ is always linear.

We are able to compute the polynomials $\beta(t)$ by the help of the following algorithm.

Algorithm 5.2.

Input: a production $p\colon F_0\to F_1\dots F_n\in P$,
strings $\beta_j\in((\Sigma\cup Y_l)^*)^k$ for each $j\in[n]$
($\beta_j=\beta(t_j)$ for some hypothetical $t_j\in \mathrm{Dt}\,(F_j)$).
Output: $\beta\in((\Sigma\cup Y_l)^*)^k$
($\beta=\beta(t)$ for $t=p(t_1, \dots, t_n)\in \mathrm{Dt}\,(F_0)$).
Method: Suppose that

$$\alpha(p)=(a_1, \dots; a_k, b_{1,1}, \dots, b_{1,l}, \dots, b_{n,1}, \dots; b_{n,l}),$$

and set the initial value of the string variables $w_{i,j}$ and u_i ($i\in[k]$, $j\in[n]$) to the i-th component of β_j and to a_i, respectively. Then apply the following procedure and set $\beta=(u_1, \dots, u_k)$.

procedure *substitute;*
begin for all $j\in[n]$ *do*
  for $i=1$ *to* k *do*
  comment substitute $b_{j,r}$ for y_r in $w_{i,j}$ for each $r\in[l]$;
  $w_{i,j}=w_{i,j}[y_r\leftarrow b_{j,r},\ r=1\ \textit{to}\ l]$;
  repeat
    for $i=1$ *to* k *do*
    $u_i=u_i[x_{k\cdot(j-1)+s}\leftarrow w_{j,s},\ j=1\ \textit{to}\ n,\ s=1\ \textit{to}\ k]$
    until ($u_i\in(\Sigma\cup Y_l)^*$ for every $i\in[k]$)
end

Lemma 5.3. Let $\mathcal{G}$ be noncircular. If $\beta_j=\beta(t_j)$ for some derivation trees $t_j\in \mathrm{Dt}\,(F_j)$ ($j\in[n]$), then after the execution of Algorithm 5.2, $\beta=\beta(t)$ for $t=p(t_1, \dots, t_n)$.

Proof. Immediate from the construction. □

Further on we assume that $\mathcal{G}$ is linear. Then $\beta(t)$ can always be "splitted" by the partial mapping

$$\xi\colon ((\Sigma\cup Y_l)^*)^k \to (\Sigma^*)^{k+l}\times(Y_l\cup\{\#\})^{*(k+l+1)}$$

which we define as follows ($\#$ is a new symbol, $\Delta^{*(s)}$ denotes the set of strings over Δ not longer than s). Let $\gamma=(c_1, \dots, c_k)\in((\Sigma\cup Y_l)^*)^k$ be linear. Put

$$\gamma_\# = \# c_1 \# c_2 \# \dots \# c_k \#,$$

and consider the symbols of $Y_l\cup\{\#\}$ occurring in $\gamma_\#$ as delimeters. Then the last component of $\xi(\gamma)$ is the string of delimeters occurring in $\gamma_\#$ read from left to right, while the first $k+l$ components of $\xi(\gamma)$ are those substrings of $\gamma_\#$ lying between the delimeters. If there are less than $k+l$ such substrings, then the remaining components of $\xi(\gamma)$ are set to λ (λ denotes the empty string). ξ is partial, since it can be applied only for linear elements. On the other hand, ξ is clearly injective, so we can speak of ξ^{-1}, the inverse of ξ.

Now we are ready to define the transformed AG $\mathcal{G}'$, which has only $k+l+1$ s-attributes, numbered again from 1 to $k+l+1$. Every nonterminal, except S' has all these attributes, S' has only the first k ones. For the sake of uniformness, however, we shall construct the semantic rules for the production $S'\to S$ in such a way that they define the "dummy" attribute occurrences $k+1, \ldots, k+l+1$ of S', too. The first $k+l$ attributes are the so called "derived" attributes with domain Σ^*, while the last one is the "control string" (denoted by cs) having $(Y_l \cup \{\#\})^{*(k+l+1)}$ as its domain. We define the semantic rules corresponding to a production $p: F_0 \to F_1 \ldots \ldots F_n$ in the following way. With the notation of Algorithm 5.2 set

$$\beta_j = \xi^{-1}(x_{(j-1)\cdot(k+l+1)+1}, \ldots, x_{(j-1)\cdot(k+l+1)+d_j-1}, \lambda, \ldots, \lambda, \mathrm{cs}(F_j)),$$

where $d_j = length(\mathrm{cs}(F_j))$, (supposing just for the moment that $X_{(k+l+1)\cdot n} \subseteq \Sigma$) and apply Algorithm 5.2. Then $\xi(\beta)$ represents the polynomials over Σ^* that define the derived attribute occurrences of F_0 in $\mathscr{G}'$, and it gives the definition of $\mathrm{cs}(F_0)$, too, in the last component.

Theorem 5.4. $\mathscr{G}$ and $\mathscr{G}'$ are equivalent.

Proof. Let $t \in \mathrm{Dt}(F)$ be any derivation tree. It can be proved by a straightforward induction on the depth of t using Lemma 5.3 that $\xi(\beta(t)) = \beta'(t)$, where $\beta'(t)$ is the sequence of values of all the attribute instances computed at the root of t in $\mathscr{G}'$. Observe that the (dummy) value of $\mathrm{cs}(S')$ is always $(\#, \ldots, \#)$. Hence, if $t \in \mathrm{Dt}(S')$, then $\beta(t)$ and the first k components of $\beta'(t)$ coincide. □

It is evident that the visit complexity of the *cd*-trees in $\mathscr{G}'$ is the best possible. On the other hand, since the form of the semantic rules became more complicated, one would think that the pure computation complexity has also increased in $\mathscr{G}'$. But this is not true. Indeed, let t be a *cd*-tree and suppose for simplicity that all the attribute instances of t are useful in the evaluation of t. If $\beta(t) = (w_1, \ldots, w_k)$ for some strings $w_i \in \Sigma^*$ $(i \in [k])$, then each w_i is the concatenation of such "atomic" strings in Σ^* that can be found on the two sides of the variable occurrences, or stand as constants on the right-hand side of the semantic rules. If w_i consists of m_i atomic strings, then $\sum_{i=1}^{k} m_i$ is a reasonable estimate for the pure computation complexity of t in $\mathscr{G}$. The same reasoning holds for $\mathscr{G}'$, too, moreover by construction, the atomic strings in $\mathscr{G}'$ are already some composites of the ones in $\mathscr{G}$. Thus, by Theorem 5.4 we obtain that the corresponding sum $\sum_{i=1}^{k} m_i'$ in $\mathscr{G}'$ is generally less than $\sum_{i=1}^{k} m_i$. The difference is compensated, however, by the extra cost of computing the control strings at each node of t.

Finally, let us mention that it is also possible to restrict the scope of our second transformation to one or more visits. For example, if $\mathscr{G}$ is *ASE* and the restriction of the semantic rules to the attribute occurrences contained in the same pass (say the m-th) is linear, then we can eliminate the i-attributes from the m-th pass and postpone their evaluation to the $(m+1)$-th pass (if any). At the same time the s-attributes of the m-th pass can be put forward to the $(m-1)$-th pass. The elaboration of a similar condition for *VSE AG* is left to the reader.

Summary

The design of efficient attribute evaluators is one of the most important requirements for compilers based on attribute grammars. Another interesting question is that, given a fixed evaluator, is it possible to optimize the attribute grammar (*AG*) to be evaluated by performing a suitable transformation on it. In this paper we present two such transformation techniques and show how they optimize evaluation by a simple visit-oriented attribute evaluator, called the *VSE* evaluator (Visit-oriented Semantic Evaluator). The class of *VSE AGs* is introduced as the visit-oriented counterpart of the class of *ASE AG*s. We also study the basic properties of *VSE AGs* relying on the strong connection between the class of *VSE AG*s and the classes of *OAG* and simple multi-visit *AG*s.

References

[1] BARTHA, M.: Linear attributed tree transformations. Acta Cybernet. 6, 125—147 (1983).
[2] BARTHA, M.: An algebraic definition of attributed transformations. Acta Cybernet. 5, 409—421 (1982).
[3] ENGELFRIET, J., FILE, G.: Simple multi-visit attribute grammars. J. Comput. System Sci. 24, 283—314 (1982).
[4] ENGELFRIET, J., FILE, G.: Passes, sweeps and visits in attribute grammars. Technische Hogeschool Twente, Onderafdeling der Informatica Memorandum Nr. INF—82—6, 1982.
[5] FARROW, R., YELLIN, D.: A comparison of storage optimizations in automatically-generated attribute evaluators. Acta Informat. 23, 393—427 (1986).
[6] FILE, G.: Attribute evaluation by recursive procedures. Theoret. Comput. Sci. 53, 25—65 (1987).
[7] GOMBÁS, E., BARTHA, M.: A multi-visit characterization of absolutely noncircular attribute grammars, Acta Cybernet. 7, 19—31 (1985).
[8] JAZAYERI, M., WALTER, K. G.: Alternating semantic evaluator. Proc. ACM Annual Conference 1975, pp. 230—234 (1975).
[9] KASTENS, U.: Lifetime analysis for attributes. Acta Informat. 24, 633—651 (1987).
[10] KASTENS, U.: Ordered attribute grammars. Acta Informat. 13, 229—256 (1980).
[11] KNUTH, D. E.: Semantics of context-free languages. Math. Systems Theory 2, 127—145 (1968); Correction: Math. Systems Theory 5, 95—96 (1971).
[12] WAITE, W. M., GOOS, G.: Compiler Construction. Berlin—Heidelberg—New York: Springer 1983.

Received June 27, 1988

Key and superkey for a closure function

By LE VAN BAO and HO THUAN

INSTITUTE OF COMPUTER SCIENCE AND CYBERNETICS, HANOI
VIÊN TINH TOAN VA ĐIEU KHIEN
LIEU GIAI, BA ĐINH, HANOI VIETNAM

In this paper, we investigate some characteristic properties of keys and superkeys for a closure function, defined on the power set of a finite set U. In particular we give a necessary condition under which a subset X of U is a key and explicit formula to compute the intersection of all keys for f, a necessary and sufficient condition for which a closure function f has precisely one key.

Moreover, the translation of a closure function f which, in some sense, preserves the keys for f, as well as the relationship between the keys for a closure function and the keys for the corresponding relation scheme are also considered.

These results are closely related to those presented in [1].

1. Keys of closure function

In this section after proving some lemmas, we give a characteristic condition under which a subset K can be a key for a closure function.

Definition [2]. Let $U=\{a_1, a_2, ..., a_n\}$ be a set of n elements (attributes) and 2^U its power set. The function $F: 2^U \to 2^U$ is called a closure function or closure iff for every $X, Y \in 2^U$

a) $X \subseteqq F(X)$,

b) $F(F(X)) = F(X)$,

c) $Y \subseteqq X \Rightarrow F(Y) \subseteqq F(X)$.

Let $K \in 2^U$. We say that K is a superkey of the closure function F if $F(K)=U$, and K is said to be a key of F if $F(K)=U$, but $F(X) \neq U$ for any proper subset X of K. We set $\overline{X}=F(X) \setminus X$.

We define two sets P and T for the closure function F as follows

a) $T = \cup\{X: X \in 2^U \text{ and } F(X) \neq X\}$,

b) $P = \cup\{\overline{X}: X \in 2^U \text{ and } F(X) \neq X\}$.

Lemma 1.1. Let F be a closure function and $X \subseteq U$ then:

$$F(X) \subseteq X \cup P.$$

Lemma 2.1. Let F be a closure function and $X \subseteq U$ and $a \notin T$. Then

$$F(X \setminus a) = F(X) \quad \text{or} \quad F(X \setminus a) = F(X) \setminus a.$$

Proof. There are two possible cases:
a) If $a \notin X$ then $X = X \setminus a$. It follows that $F(X) = F(X \setminus a)$.
b) If $a \in X$ then from the definition of T we have $F(X) = X$, and $F(X) \setminus a = X \setminus a$. Thus $F(X \setminus a) = F(F(X) \setminus a)$. On the other hand we have

$$F(X) \setminus a \subseteq F(F(X) \setminus a) = F(X \setminus a) \subseteq F(X).$$

It is clear that $F(X \setminus a) = F(X)$ or $F(X \setminus a) = F(X) \setminus a$. The proof is complete.

Lemma 3.1. Let F be a closure function. Then:

$$P \subseteq F(T).$$

Proof. Let $a \in P$. From the definition of P there would exist an $X \subseteq U$ such that $F(X) \neq X$ and $a \in F(X) \setminus X$. Clearly $X \subseteq T$. Hence $F(X) \subseteq F(T)$, showing that $a \in F(T)$.

Lemma 4.1. Let F be a closure function. If $a \in F(X) \setminus P$ then

$$a \in X.$$

Proof. We have $F(X) = X \cup (F(X) \setminus X) \subseteq X \cup P$. On the other hand $a \in F(X)$ and $a \notin P$. This implies $a \in X$.

Lemma 5.1. Let F be a closure function. If $a \notin T$ and $F(Y) \subseteq F(X)$ then

$$F(Y \setminus a) \subseteq F(X \setminus a).$$

Proof. Since $a \notin T$, taking account of Lemma 2.1 we get $F(X \setminus a) = F(X)$ or $F(X \setminus a) = F(X) \setminus a$.

a) If $F(X \setminus a) = F(X)$, it is obvious that

$$Y \setminus a \subseteq F(Y) \setminus a \subseteq F(X) \setminus a \subseteq F(X) = F(X \setminus a)$$

implies

$$F(Y \setminus a) \subseteq F(F(X \setminus a)) = F(x \setminus a).$$

b) If $F(X \setminus a) = F(X) \setminus a$ we have

$$Y \setminus a \subseteq F(Y) \setminus a \subseteq F(X) \setminus a = F(X \setminus a).$$

Clearly

$$F(Y \setminus a) \subseteq F(F(X \setminus a)) = F(X \setminus a),$$

the proof is complete. As an immediate consequence of Lemma 5.1 we have the following.

Lemma 6.1. Let F be a closure function. If $a \notin T$ and $F(Y) = F(X)$ then

$$F(X \setminus a) = F(Y \setminus a).$$

Lemma 7.1. Let F be a closure function. If $a \in K \cap F(K \setminus a)$, then K is not a key of F.

Proof. From $a \in K$ we have $K \setminus a \subsetneqq K$. Thus $F(K \setminus a) \subseteqq F(K)$. On the other hand $a \in F(K \setminus a)$. Clearly $a \cup (K \setminus a) \subseteqq F(K \setminus a)$, thus $K \subseteqq F(K \setminus a)$. It is obvious that $F(K) \subseteqq F(F(K \setminus a)) = F(K \setminus a) \subseteqq F(K)$. Therefore $K \setminus a \subsetneqq K$ and $F(K) = F(K \setminus a)$. From this we get that K is not a key of the closure function F.

Theorem 1.1. Let F be a closure function and K a key of F then:

$$U \setminus P \subseteqq K \subseteqq (U \setminus P) \cup (P \cap T).$$

Proof. We shall begin with showing that $U \setminus P \subseteqq K$. Assume to the contrary, that is $U \setminus P \nsubseteqq K$. From this, there would exist an $a \in (U \setminus P) \setminus K$. Clearly $a \notin P$ and $a \notin K$. On the other hand, since K is a key of the closure function F, it is obvious that $a \in F(K)$. From $a \notin P$ and taking account of Lemma 4.1, we get $a \in K$ which conflicts with $a \notin K$.

To complete the proof it remains to show that $K \subseteqq (U \setminus P) \cup (P \cap T)$. We know that $U = (U \setminus P) \cup P = (U \setminus P) \cup (P \cap T) \cup (P \setminus T)$. Suppose to the contrary, i.e. $K \nsubseteqq (U \setminus P) \cup (T \cap P)$. Then there exists an $a \in K \cap (P \setminus T)$.

From this we have $a \in K$, $a \in P$ and $a \notin T$. Because K is a key of F, we have $F(K) = U = F(U)$. From $a \notin T$, taking account of Lemma 6.1, we get $F(K \setminus a) = F(U \setminus a)$. From $a \notin T$, evidently $T \subseteqq U \setminus a$. It is obvious that $F(T) \subseteqq F(U \setminus a)$. In view of the Lemma 3.1, $P \subseteqq F(T)$. Combining this with $a \in P$ we obtain

$$a \in P \subseteqq F(T) \subseteqq F(U \setminus a) = F(K \setminus a).$$

It is clear that $a \in K \cap F(K \setminus a)$ showing, by Lemma 7.1, that K is not a key of F. We thus arrive to a contradiction. The proof is complete.

2. Intersection of all keys for a closure function

We propose in this section to describe the intersection of all key of a closure function.

Lemma 8.2. Let F be a closure function and $a \in P$. Then there exists a key K of F such that $a \notin K$.

Proof. Becauese $a \in P$, there exists an $X \subseteqq U$ such that $a \in F(X) \setminus X$. Let $C \subseteqq U$ such that $F(X) \cup C = U$ and $F(X) \cap C = \emptyset$. Clearly, $U \subseteqq F(X) \cup C \subseteqq F(X \cup C) \subseteqq U$. Thus $F(X \cup C) = U$ and there exists a key $K \subseteqq X \cup C$. It is clear that $a \notin K$.

Theorem 2.2. Let F be a closure function and let I be the intersection of all keys of F. Then

$$I = U \setminus P.$$

Proof. From Theorem 1.1 we have $U \setminus P \subseteqq I$. To complete the proof, it remains to show that $I \subseteqq U \setminus P$. In view of Lemma 8.2 we obtain $I \cap P = \emptyset$ showing that $I \subseteqq U \setminus P$.

Hence $I = U \setminus P$. The proof is complete.

3. Sufficient and necessary condition under which a closure function has precisely one key

In this section we present a theorem which gives a sufficient and necessary condition for a closure function F to have precisely one key.

Theorem 3.3. Let F be a closure function. Then F has precisely one key iff $T \cap P \subseteq F(T \setminus P)$.

Proof. Sufficiency: Let $T \cap P \subseteq F(T \setminus P)$. From this we have $T = (T \setminus P) \cup \cup (T \cap P) \subseteq F(T \setminus P)$. It is clear that $F(T \setminus P) \subseteq F(T) \subseteq F(F(T \setminus P)) = F(T \setminus P)$. Thus $F(T) = F(T \setminus P)$.

By Lemma 3.1, $P \subseteq F(T)$. It is clear that: $F(T \cup P) \subseteq F(T)$. From this, $F(T) = F(T \cup P)$. Taking account of Lemma 1.1 we find $F(T \cup P) \subseteq (T \cup P) \cup P = = T \cup P$. Thus $T \cup P = F(T \cup P)$. Consequently $F(T \setminus P) = F(T) = F(T \cup P) = T \cup P$. On the other hand we have $T \setminus P \subseteq U \setminus P$.

Thus $T \cup P = F(T \setminus P) \subseteq F(U \setminus P)$. From this we find $U = (U \setminus P) \cup (T \cup P) \subseteq \subseteq F(U \setminus P) \subseteq U$. Finally we have $U = F(U \setminus P)$.

Now we shall show that $U \setminus P$ is the unique key of F. If $U \setminus P$ is not a key of F then there exists a key X of F such that $X \subsetneqq U \setminus P$. By Theorem 1.1 we have $U \setminus P \subseteq X \subsetneqq U \setminus P$ showing that $U \setminus P$ is the unique key of the closure function F.

Necessity: Let F be a closure function that has precisely one key K. We invoke Theorem 2.2 to deduce that $I = U \setminus P = K$, showing that $U \setminus P$ is a key of F. Thus $F(U \setminus P) = U$. There are two possible cases.

a) If $U \setminus P \neq U$ then from the definition of T we have $U \setminus P \subseteq T$. Thus $U \setminus P \subseteq \subseteq T \setminus P$ and clearly $U = F(U \setminus P) \subseteq F(T \setminus P) \subseteq U$. This implies $U = F(T \setminus P)$. Consequently $T \cap P \subseteq F(T \setminus P)$.

b) If $U \setminus P = U$ then clearly $P = \emptyset$. From this we have $\emptyset = T \cap P \subseteq F(T \setminus P)$. The proof is complete.

Example. Let $U = \{a, b, c\}$.
$F: 2^U \to 2^U$ is a closure function,

$F(\emptyset) = \emptyset,$

$F(a) = ab,$

$F(b) = b,$

$F(c) = abc,$

$F(ab) = ab,$

$F(ac) = abc,$

$F(cb) = cba,$

$F(abc) = abc.$

From this we have:

$F(a) = ab \neq a, \quad \bar{a} = b,$

$F(c) = abc \neq c, \quad \bar{c} = ab,$

$F(ac) = abc \neq ac, \quad \overline{ac} = b, \quad f(cb) \neq abc \Rightarrow \overline{cb} = a.$

We obtain:

$T = acb,$

$P = ab,$

$T \cap P = ab.$

a) If K is a key of the closure function F then:

$$U \setminus P \subseteq K \subseteq (U \setminus P) \cup (P \cap T).$$

Thus $c \subseteq K \subseteq cab.$

b) The intersection of all keys of F is

$$I = U \setminus P = c.$$

c) $T \cap P = ab$, $F(T \setminus P) = F(c) = abc \Rightarrow T \cap P \subseteq F(T \setminus P)$. From this, F has precisely one key $K = U \setminus P = c$.

4. Translations of closure functions

In this section we shall be concerned with a class of translations of closure functions. Starting from a given closure function, translations make it possible to obtain more simple closure functions so that the key — finding problem becomes less cumbersome, etc. On the other hand, from the set of key for the new, closure function obtained in this way the corresponding keys of the original closure function can be found by a single translation.

Let $C(F)$ denote the family of all keys for the closure F. We define two sets H and G as follows:

$$G = \cap \{K \mid K \in C(F)\},$$

$$H = \cup \{K \mid K \in C(F)\}.$$

Lemma 9.4. Let F be a closure function in U, and $A \subseteq U$. We define a new F_A by

$$F_A(E) = F(E \cup A) \setminus A \quad \text{for} \quad E \subseteq U \setminus A.$$

Then: F_A is a closure function in $U \setminus A$.

Proof.

a) Let $E \subseteq U \setminus A$. Since F is a closure function, $E \subseteq F(E \cup A)$ and $E \cap A = \emptyset$. Clearly $E \subseteq F(E \cup A) \setminus A$. Consequently $E \subseteq F_A(E)$.

b) Let $E_1 \subseteq E_2 \subseteq U \setminus A$. Clearly, $F(E_1 \cup A) \subseteq F(E_2 \cup A)$, which implies $F_A(E_1) = F(E_1 \cup A) \setminus A \subseteq F(E_2 \cup A) \setminus A = F_A(E_2)$.

c) Let $E \subseteq U \setminus A$. To complete the proof it remains to show that $F_A(E) = F_A(F_A(E))$. We have $F_A(F_A(E)) = F_A(F(E \cup A) \setminus A) = F(F(E \cup A) \setminus A) \cup A) \setminus A$. Since $A \subseteq F(E \cup A)$, $F(F(E \cup A) \setminus A) \cup A) \setminus A = F(F(E \cup A)) \setminus A = F(E \cup A) \setminus A = F_A(E)$. From a), b), and c), we conclude that F_A is a closure function.

Lemma 10.4. Let F be a closure function in X, $A \cap X = \emptyset$. We define a new F^A by:

$$F^A(E) = F(E \setminus A) \cup A \quad \text{for} \quad E \subseteq X \cup A.$$

Then F^A is a closure function in $X \cup A$.

Proof.

a) Let $E\subseteq X\cup A$. We have $E=(E\setminus A)\cup(E\cap A)$. On the other hand $E\setminus A\subseteq F(E\setminus A)$ and $E\cap A\subseteq A$, showing that $E\subseteq F(E\setminus A)\cup A=F^A(E)$.

b) Let $E_1\subseteq E_2\subseteq X\cup A$. This implies $F(E_1\setminus A)\subseteq F(E_2\setminus A)$ and $F^A(E_1)=F(E_1\setminus A)\cup A\subseteq F(E_2\setminus A)\cup A\subseteq F^A(E_2)$.

c) Let $E\subseteq X\cup A$. Since F is a closure function in X and $A\cap X=\emptyset$, we have $F(E\setminus A)\cap A=\emptyset$. It is clear that: $F^A(F^A(E))=F^A(F(E\setminus A)\cup A)=F((F(E\setminus A)\cup A)\setminus A)\cup A=F(F(E\setminus A))\cup A=F(E\setminus A)\cup A=F^A(E)$. Consequently $F^A(F^A(E))=F^A(E)$ and F is a closure function in $X\cup A$.

Lemma 11.4. Let F be a closure function in U, $A\subseteq U$. Then:

1. $F(X)\setminus A\subseteq F_A(X\setminus A)$ for all $X\subseteq U$, and
2. $F_A(X)\cup A=F(X\cup A)$ for all $X\subseteq U\setminus A$.

Proof.

1. From the definition of F_A we have $F_A(X\setminus A)=F((X\setminus A)\cup A)\setminus A=F(X\cup A)\setminus A$. On the other hand $F(X)\subseteq F(X\cup A)$. Thus $F(X)\setminus A\subseteq F(X\cup A)\setminus A$. Consequently $F(X)\setminus A\subseteq F_A(X\setminus A)$.

2. We have $F_A(X)=F(X\cup A)\setminus A$. Since $A\subseteq F(X\cup A)$, we get $F_A(X)\cup A=F(F(X\cup A)\setminus A)\cup A=F(X\cup A)$.

Theorem 4.4. Let F be a closure function in U, $A\subseteq G$. Then: K is a key of F_A if and only if $A\cap K=\emptyset$ and $K\cup A$ is a key of F.

Proof. We first prove the necessity: Suppose that K is a key of F_A. Obviously $F_A(K)=U\setminus A$ and $A\cap K=\emptyset$. Taking Lemma 11.4 into account we get:

$$U=(U\setminus A)\cup A=F_A(K)\cup A\subseteq F(K\cup A)\subseteq U,$$

showing that $K\cup A$ is a superkey of F. If $K\cup A$ were not a key of F then there would exist a key $\bar{K}$ of F such that $A\subseteq\bar{K}\subsetneqq K\cup A$. Consequently there would exist an $K_1\subsetneqq K$ such that: $\bar{K}=K_1\cup A$, $K_1\cap A=\emptyset$. Since $\bar{K}$ is a key for F, $F(K_1\cup A)=U$. Applying Lemma 11.4, clearly $U\setminus A=F(K_1\cup A)\setminus A\subseteq F_A(K_1\cup A\setminus A)=F_A(K_1)$. So we have $K_1\subsetneqq K$, $F_A(K_1)=U\setminus A$. This contradicts the hypothesis that K is a key of F_A.

We now turn to the proof of sufficiency. Suppose that $K\cap A=\emptyset$ and $K\cup A$ is a key for F_A. We have to show that K is a key for F_A. Since $K\cup A$ is a key for F, we have $F(A\cup K)=U$. By virtue of Lemma 11.4 and $K\cap A=\emptyset$, we get $U\setminus A=F(K\cup A)\setminus A\subseteq F_A(K\cup A\setminus A)=F_A(K)\subseteq U\setminus A$. Thus $U\setminus A=F_A(K)$, showing that K is a superkey for F_A. Assume that K is not a key of F_A, then there would exist a key $\bar{K}$ of F_A such that $\bar{K}\subsetneqq K$ and $F_A(\bar{K})=U\setminus A$. Applying Lemma 11.4, it follows $U=F_A(\bar{K})\cup A=F(\bar{K}\cup A)$ where $\bar{K}\cup A\subsetneqq K\cup A$. This contradicts the fact that $K\cup A$ is a key for F, that completes the proof.

Theorem 5.4. Let F be a closure function in U, $A\subseteq U$ and $A\cap H=\emptyset$. Then K is a key of F_A iff K is a key of F.

Proof.

1. The necessity: Suppose that K is a key for F_A. Obviously $F_A(K)=U\setminus A$. By virtue of Lemma 11.4 we have $F(K\cup A)=F_A(K)\cup A=(U\setminus A)\cup A=U$, showing

that $K \cup A$ is a superkey for F. Hence, there exists a key $\bar{K}$ of F such that $\bar{K} \subseteq K \cup A$. Since $A \cap H = \emptyset$ then $\bar{K} \cap A = \emptyset$. From this, it is easy to see that $\bar{K} \subseteq K$. There are two possible cases:

a) $\bar{K} = K$. Then obviously K is a key for F.

b) $\bar{K} \subsetneq K$. Since $\bar{K}$ is a key for F, $F(\bar{K}) = U$. Applying Lemma 11.4, we have $U \setminus A = F(\bar{K}) \setminus A \subseteq F_A(\bar{K} \setminus A) \subseteq U \setminus A$ and $\bar{K} \cap A = \emptyset$, that is $F_A(\bar{K}) = U \setminus A$. This contradicts the fact that K is a key for F_A.

2. The sufficiency: Suppose that K is a key for F. We have to prove that K is also a key for F_A. We have, by the definition of keys, $F(K) = U$. Applying Lemma 11.4, $U \setminus A = F(K) \setminus A \subseteq F_A(K \setminus A) \subseteq U \setminus A$. Thus $F_A(K \setminus A) = U \setminus A$. Since $A \cap H = \emptyset$, it follows $K \cap A = \emptyset$. Consequently $F_A(K) = U \setminus A$ showing that K is a superkey of F_A. Now assume to the contrary, that K is not a key for F_A. Then, there would exist a key $\bar{K}$ of F_A such that $\bar{K} \subsetneq K$. Obviously $F_A(\bar{K}) = U \setminus A$. We invoke Lemma 11.4 to deduce $F(\bar{K} \cup A) = F_A(\bar{K}) \cup A = (U \setminus A) \cup A = U$, showing that $\bar{K} \cup A$ is a superkey of F. Consequently, there exists a key $\bar{\bar{K}}$ of F such that $\bar{\bar{K}} \subseteq \bar{K} \cup A$, $\bar{\bar{K}} \cap A = \emptyset$. From this $\bar{\bar{K}} \subseteq \bar{K} \subsetneq K$. This contradicts the hypothesis that K is a key for F.

This completes the proof.

To continue let us recall a result from § 1. Let F be a closure function in U. Let us set

$$T = \bigcup \{X \mid X \in 2^U \text{ and } F(X) \neq X\},$$

$$P = \bigcup \{\bar{X} \mid X \in 2^U \text{ and } F(X) \neq X\}.$$

Then, the necessary condition under which K is a key for F is

1. $U \setminus P \subseteq K \subseteq (U \setminus P) \cup (T \cap P)$, and

2. the intersection I of all keys for F is $I = U \setminus P$. We have the following theorems.

Theorem 6.4. Let F be a closure function in U and $I = U \setminus \bigcup \{F(X) \setminus X \mid X \in Z^U$ and $F(X) \neq X\}$. Then K is a key of F_I if and only if $K \cap I = \emptyset$ and $K \cup I$ is a key of F.

Theorem 7.4. Let F be a closure function in U, and $N = P \setminus T$. Then K is a key of F_N if and only if K is a key of F.

Lemma 12.4. Let F be a closure function in U, $U \cap A = \emptyset$. Then

1. $F^A(X) \setminus A \subseteq F(X \setminus A)$, $X \subseteq U \cup A$,

2. $F(X) \cup A = F^A(X \cup A)$, $X \subseteq U$.

Proof. We first prove

1. Let $X \subseteq U \cup A$. From the definition of F^A we have:

$$F^A(X) \setminus A = (F(X \setminus A) \cup A) \setminus A = F(X \setminus A) \setminus A \subseteq F(X \setminus A).$$

2. Let $X \subseteq U$. We have $F^A(X \cup A) = F(X \cup A \setminus A) \cup A = F(X \setminus A) \cup A$. Since $A \cap U = \emptyset$, $A \cap X = \emptyset$. It is clear that $F^A(X \cup A) = F(X \setminus A) \cup A = F(X) \cup A$. This completes the proof.

Theorem 8.4. Let F be a closure function in U and $A \cap U = \emptyset$. Then $K \cap A = \emptyset$ and K is a key of F^A iff K is a key of F.

Proof. We first prove the necessity: Suppose that K is a key of F^A and $K \cap A = \emptyset$. Obviously $F^A(K) = U \cup A$. Taking Lemma 12.4 we get:

$$U = U \cup A \setminus A = F^A(K) \setminus A \subseteq F(K \setminus A) \subseteq U.$$

Obviously, $F(K) = F(K \setminus A) = U$, showing that K is a superkey of F. If K were not a key of F, then there would exist a key $\bar{K}$ such that $\bar{K} \subsetneqq K$ and $F(\bar{K}) = U$. From the definition of F^A we find: $F^A(\bar{K}) = F(\bar{K} \setminus A) \cup A = F(\bar{K}) \cup A = U \cup A$. This contradicts the hypothesis that K is a key for F. We now turn to the proof of the sufficiency. Suppose that K is a key for F. We have to show that $K \cap A = \emptyset$ and K is a key for F^A. Since K is a key of F, we have $F(K) = U$ and $K \subseteq U$. Thus $K \cap A = \emptyset$. On the other hand $F^A(K) = F(K \setminus A) \cup A = F(K) \cup A = U \cup A$ showing that K is a superkey of F^A. If K is not a key of F^A, then there would exist a key $\bar{K}$ such that $\bar{K} \subsetneqq K$ and $F^A(\bar{K}) = U \cup A$. We have $U = F^A(\bar{K}) \setminus A \subseteq F(\bar{K} \setminus A) = F(\bar{K}) \subseteq U$. Thus $F(\bar{K}) = U$. This contradicts the hypothesis that K is a key of F. Hence K is a key of F^A. The proof is complete.

5. On a relationship between keys for relation scheme and keys for closure function

Let us recall some necessary notions and definitions. Definition of a closure function: Let $U = \{A_1, A_2, \ldots, A_n\}$ be a set of n elements (attributes) and 2^U its power set. The function $f\colon 2^U \to 2^U$ is called a closure function or closure iff for every $X, Y \in 2^U$,

a) $X \subseteq f(X)$,

b) $f(f(X)) = f(X)$,

c) if $X \subseteq Y$ then $f(X) \subseteq f(Y)$.

Let $K \subseteq U$, K is said to be a superkey for the closure function f if $f(K) = U$. K is said to be a key for the closure function f if K is a superkey for f but $f(X) \neq U$ for any proper subset X of K. Let $C(f)$ denote the family of all keys for the closure function f.

Definition of a relation scheme: [3].

Armstrong's axioms [4]. Let $X, Y, Z \subseteq U$;

Rule 1: (Reflexivity) if $Y \subseteq X$ then $X \to Y$;

Rule 2: (Transitivity) if $X \to Y$ and $Y \to Z$ then $X \to Z$;

Rule 3: (Augmentation) if $X \to Y$ then $X \cup Z \to Y \cup Z$.

Relation scheme:

A relation scheme is a 2-tuple (U, F) where:

a) U is a finite set (of attributes),

b) F is a finite set of functional dependencies (FD).

Let F be a given set of FD's of a relation scheme. We can apply these rules to the FD's in F to derive new FD's.

The set of all *FD*'s that are derivable from F by repeated applications of Armstrong's rules (including the *FD*'s in F) is called the closure of F and is denoted by F^+.

Let $X \subseteq U$ be a given set of attributes. We define the closure of X (relative to F), denote by X^+, to be the set of all attributes that are functionally dependent on X:

$$X^+ = \{A | (X \to A) \in F^+\}.$$

Algorithm for finding X^+:

$$X^{(0)} = X,$$

$$X^{(i+1)} = X^{(i)} \cup \{R_j | L_j \to R_j \in F \text{ and } L_j \subseteq X^{(i)}\}.$$

There exists an N such that $X^{(N)} = X^{(N+1)}$. Then $X^+ = X^{(N)}$. We have $X \to Y \in F^+$ iff $Y \subseteq X^+$.

Let (U, F) be a relation scheme and let X be a subset of U. We say that X is a superkey of (U, F) if every attribute in U functionally depends on X. If the set X is a superkey and it does not properly contain any superkey then X is a key for (U, F).

$C(U, F)$ denotes the set of all keys of a relation scheme (U, F).

Theorem 9.5. Let (U, F) be a relation scheme. We define the function $f: 2^U \to 2^U$ as follows:

$$X \in 2^U: f(X) = X^+.$$

Then 1. f is a closure function;
2. $C(f) = C(U, F)$.

Proof. We first prove 1.

a) $X \subseteq X^+$. Clearly $X \subseteq f(X)$.

b) $X = (X^+)^+$ implies $f(X) = f(f(X))$.

c) $X \subseteq Y \Rightarrow X^+ \subseteq Y^+$ implies $f(X) \subseteq f(Y)$.

Consequently f is a closure function.

2. Now let K be a key of the relation scheme (U, F). Obviously $K^+ = U$. Thus we have $f(K) = U$, showing that K is a superkey for f. Now assume to the contrary that, K is a not a key for f. Then there would exist a key $\bar{K}$ of f such that $\bar{K} \subsetneq K$ and $f(\bar{K}) = U$. From the definition of f we have $\bar{K}^+ = U$. Thus $\bar{K} \to U$. This contradicts the hypothesis that K is a key of (U, F).

Now let K be a key of the closure function f. Obviously $f(K) = U$. Thus $K^+ = U, K$ is a superkey for (U, F). Now assume to the contrary, that K is not a key for (U, F). Then there would exist a key $\bar{K}$ of (U, F) such that $\bar{K} \subsetneq K$ and $\bar{K} \to U$. We have $K^+ = U$. Thus $f(\bar{K}) = U$. This contradicts the hypothesis that K is a key for f.

Theorem 10.5. Let f be a closure function in U. We define the relation scheme (U, F) as follows:

$$F = \{X \to f(X) | X \in 2^U\}.$$

Then

$$C(f) = C(U, F).$$

Proof. From the definition of F we have $X \to f(X) \in F$. Thus $f(X) \subseteq X^+$. Now we have to prove $X^+ \subseteq f(X)$.

We proceed by induction on n. If $n=0$ we have $X^{(0)}=X\subseteq f(X)$. Assume it is true for n i.e. $x^{(n)}\subseteq f(X)$. In fact we have $X^{(n+1)}=X^{(n)}\cup\{\bigcup Y|Z\to Y\in F, Y=F(Z)$ and $Z\subseteq X^{(n)}\}$. From $Z\in X^{(n)}$ we have $f(Z)\subseteq f(X^{(n)})\subseteq f(f(X))=f(X)$. Obviously $X^{(n+1)}\subseteq f(X)$. Finally, we find $f(X)=X^+$. Applying Theorem 9.5, we have $C(f)=$ $=C(Z, F)$. The proof is complete.

References

[1] DEMETROVICS, J., HO THUAN, NGUYEN XUAN HUY and LE VAN BAO: Translations of relation Schemes, Balanced relation schemes and the problem of key representation, J. Inf. Process. Cybern. EIK 23 (1987), N 2/3, pp 81—97.

[2] BÉKÉSSY, A., DEMETROVICS, J., et. al., On the number of Maximal Dependencies in a Data base relation of fixed order. Discrete Mathematics 30 (1980), pp 83—88.

[3] HO THUAN and LE VAN BAO: Some results about keys of relational schemas. Acta Cybernetica. Tom. 7, Fasc. 1, Szeged, Hungary, 1985, pp 99—113.

[4] ARMSTRONG, W. W.: Dependency Structures of Data Base Relationship, Information Processing 74, North Holland Pub. Co. Amsterdam 1974, pp 580—583.

Received August 12, 1988

K. G. Murty: Linear Complementarity, Linear and Nonlinear Programming (Sigma series in applied mathematics, 3), XLVIII+629 pages, Heldermann Verlag, Berlin, 1988.

This is an extended and up-to-dated book of "Linear and Combinatorial Programming" published in 1976.

"The book begins with a section titled 'notation' in which all the symbols and several terms are defined. It is strongly recommended that the reader peruse this section first at initial reading, and refer to it whenever there is a question about the meaning of some symbol or term.

Chapter 1 presents a clear geometric interpretation of the Linear Complementarity Problem (LCP) through the definition of the system of complementary cones as a generalization of the set of orthants in R^n. Applications to Linear Programming (LP), Quadratic Programming (QP), and nonzero sum game problems are discussed. There is a complete discussion of positive definiteness and positive semidefiniteness of square matrices, their relationship to convexity, together with efficient pivotal methods for checking whether these properties hold for a given matrix. Various applications of QP are discussed, as well as the recursive quadratic programming method for solving Nonlinear Programming (NLP) models.

Chapter 2 presents a complete discussion of the many variants of the complementary pivot method and proofs of its convergence on different classes of LCPs. Section 2.7. contains a very complete, lucid, but elementary treatment of the extensions of the complementary pivot method to simplicial methods for computing fixed points using triangulations of R^n, and various applications of these methods to solve a variety of general NLP models and nonlinear complementarity problems.

Chapter 3 covers most of the theoretical properties of the LCP. There is extensive treatment of the various separation properties in the class of complementary cones, and a complete discussion of principal pivot transforms of matrices. In this chapter we also discuss the various classes of matrices that arise in the study of the LCP. Chapter 4 provides a survey of various principal pivoting methods for solving the LCP. Algorithms for parametric LCP are presented in Chapter 5.

Chapter 6 contains results on the worst case computational complexity of the complementary and the principal pivoting methods for the LCP. Chapter 7 presents a special algorithm for the LCP associated with positive definite symmetric matrices, based on orthogonal projections, which turned out to be very efficient in computational tests. Chapter 8 presents the polynomially bounded ellipsoid method for solving LCPs associated with positive semidefinite matrices, or equivalently convex QPs.

Chapter 9 presents various iterative methods for LCPs. In Chapter 10 we present an extensive survey of various descent methods for unconstrained and linearly constrained minimization problems; these techniques provide alternative methods for solving quadrativ programming problems. In Chapter 11 we discuss some of the newer algorithms proposed for solving linear programming problems and their possible extensions to solve LCPs, and we discuss several unsolved research problems in linear complementarity.

To make the book self-contained, in the appendix we provide a complete treatment of theorems of alternatives for linear systems, properties of convex functions and convex sets, and various optimality conditions for nonlinear programming problems."

This is a high quality book. It may be recommended for a great number of people from researchers to graduate students.

J. Csirik

Chew Soo Hong, Zheng Quan: Integral Global Optimization — Theory, Implementation and Applications (Lecture Notes in Economics and Mathematical Systems, Vol 298), VII+179 pages. Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo, 1988.

The book discusses a global optimization method studied and presented by the second author in numerous papers in Chinese only. Thus it is the first detailed publication of this algorithm in English. The book consists of five chapters: Preliminary, Integral Characterizations of Global Optimality, Theoretical Algorithms and Techniques, Monte Carlo Implementation, and Applications.

The method is based on the idea that for each level set the mean value of the objective function is not greater than the function value to which this level set belongs. At the same time, the mean value is always not less than the global minimum in the mentioned level set. With the mean value we can form a new, smaller level set, find the new mean value on this level set, and so on. This simple but fairly original idea results in a global optimization algorithm which can provide the set of global minimum points for all robust sets and all continuous objective functions. The word "integral" in the title of the book hints at the method with which one can determine the mean function value on a set.

The problem is, however, that each step of this algorithm (and what is more, separately the determination of the mean value and the level set) has the same computational complexity as that of the original global optimization problem. Because of this, one has to give up the guarantee for obtaining the set of global minimum points for all global optimization problems, and to produce a reasonable heuristic based on the theoretical algorithm mentioned before. This procedure to find an implementable algorithm that is reliable and effective enough is rather common. In all such cases the question is how to modify the theoretical algorithm to get an implementable version retaining the most of the desirable features of the original one.

In this case the above problem is solved in a rather rude way. The determination of a level set is padded in the algorithm with the selection of the set of points of a random sample that have function values not greater than the specified level, and with the calculation of an n-dimensional box containing this set of points, with a certain tolerance. The determination of the mean function value on a level set is substituted by the finding out of the mean function value on the random sample. These steps are easily implementable and of low complexity.

The reliability of this simplified algorithm depends heavily on the size of the random sample. From a practical point of view, the numerical testing of the reliability for all global optimization methods is more important than the theorems about the theoretical algorithms in the background.

The algorithm was tested with two of the standard global optimization test problems. Unfortunately two very simple problems were chosen: their global minima can be located by the majority of global optimization methods with high probability. It would have been interesting to see the performance of the new method on other test problems that have more hidden global minima. The test results are compared with those of Törn's method, and it is found that the method of Zheng is about five times more effective. This comparison is rather unfair, since the method of Törn is to find all local minima (not only the global one). In addition, the method of De Biase and Frontini (to be found in the same book dealing with the method of Törn) is more effective than that of Zheng, not to speak about later publications than the mentioned book published in 1978.

Some (not printer's or typing) errors make the reading difficult. In spite of the unsatisfactory and not convincing numerical testing, the book is worth reading for those interested in optimization.

Tibor Csendes

A. S. House: The recognition of speech by machine — A bibliography. Academic Press, Harcourt Brace Jovanovich, Publishers. London—San Diego—New York—Boston—Sydney—Tokyo—Toronto, 1988.

"The book is divided into three parts. Part 1 is a subject index that attempts to provide the names of authors who have written on a particular subject. Names listed here are sole or primary authors, making it possible for the user to search directly for citations in Part 2, the bibliography proper. An attempt has been made to provide enough detail and crossreference in the subject index to make the user's task manageable, but the well-known weaknesses of subject indexes are still obvious.

In Part 2 the citations are arranged alphabetically starting with the surnames of the sole or the primary authors. The alphabetical listing constitutes the working part of the bibliography.

Part 3 is an alphabetical list of authors; it will help is finding work for which an author is not the initially listed author, that is, neither a sole nor a primary author. Instructions for the use of the listing are included at the start of the section. It is hoped that the author index, along with the subject index, will provide adequate means of entry into the listing of citations in Part 2. It may be helpful to remember that the alphabetic ordering of names that have been transliterated from languages that do not use the roman alphabet can be surprising at time.

The titles include articles that appeared in periodicals, conference proceedings, institutional or laboratory reports that appear on a regular basis, as well as extended abstracts from proceedings, abstracts of oral presentations, etc., and books."

This bibliography include more than 4000 titles. It is surely very helpful for researchers of speech recognition.

J. Csirik

A. Kurzhanski, K. Neumann, D. Pallaschke (Editors): **Optimization, Parallel Processing and Applications** (Lecture Notes in Economics and Mathematical Systems, Vol. 304), VI+292 pages, Springer-Verlag, Berlin—Heidelber—New York—London—Paris—Tokyo, 1988.

This book contains selected papers that were presented at the Oberwolfach Conference on Operations Research, February 16—21, 1987 and the Workshop on Advanced Computation Techniques, Parallel Processing and Optimization, held at Karlsruhe, West Germany, February 22—25 1987.

On the basis of their subject, the papers form two groups: one discussing new optimization methods, and another studying the possible impact of recent computer advances (such as parallel computation, interval arithmetic and automatic differentation) on optimization. The contents of the book are the following: B. Bank, R. Mandel: Quantitative Stability of (Mixed-) Integer Linear Optimization Problems; O. Burdakov, C. Richter: Parallel Hybrid Optimization Methods; V. F. Demyanov: Continuous Generalized Gradients for Nonsmooth Functions; R. Horst: Outer Cut Methods in Global Optimization; P. S. Kenderov, N. K. Ribarska: Generic Uniqueness of the Solution of "Max Min" Problems; M. Schäl: Optimal Stopping and Leavable Gambling Models with Observation Costs; S. Schaible: Multi-Ratio Fractional Programming — A Survey; D. Conforti, L. Grandinetti: An Experience of Advanced Computation Techniques in the Solution of Nonlinearly Constrained Optimization Problems; L. C. W. Dixon: Automatic Differentiation and Parallel Processing in Optimisation; Y. Evtushenko: V. Mazourik, V. Ratkin: Multicriteria Optimization in the DISO System; R. De Leone, O. L. Mangasarian: Serial and Parallel Solution of Large Scale Linear Programs by Augmented Lagrangian Succesive Over-relaxation; K. Schittkowski: EMP: An Expert System for Mathematical Programming; R. G. Strongin, Y. D. Sergeev: Effective Algorithm for Global Optimization with Parallel Computations; M. Bartusch, R. H. Möhring, F. J. Radermacher: M-Machine Unit Time Scheduling: A Report on Ongoing Research; S. Perz, S. Rolewicz: On Inverse-Image of Non-Oriented Graphs; M. Kisielewicz: Existence of Optimal Trajectory of Mayer Problem for Neutral Functional Differential Inclusions; D. Przeworska-Rolewicz: Smooth Solutions of Linear Equations with Scalar Coefficients in a Right Invertible Operator; G. Feichtinger: Production-Pollution Cycles; B. Mazbic-Kulma, E. Komorowska, J. Stepien: Location Problem and its Applications in Distribution of Petrol.

The papers of Horst, Dixon, Evtushenko et al., Schittkowski and Strongin et al. are of great importance, and certainly worth reading. The book can be recommended for those willing to keep abreast with the new trends of optimization.

Tibor Csendes

ECOOP '88 European Conference on Object-Oriented Programming, Oslo, Norway, August 1988. Proceedings, Editors: S. Gjessing and K. Nygaard (Lecture Notes in Computer Science Vol. 322), VI+410 pages, Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo, 1988.

"Object oriented seems to be becoming in the 1980s what structured programming was in the 1980s", (B. Randell and P. Lee), quoted in the motto of the conference.

The volume contains 22 papers from 13 countries, read at the second conference on object oriented programming. The papers included in the volume were selected by the program committee from 103 contributions. The editors "believe that the papers contain a representative sample of the best works in object oriented programming today".

We can find papers on various topics including theoretical and practical aspects such as teaching object oriented programming, handling data bases, debugging, developing distributed systems, algebraic specification languages, etc.

Contents of the Volume:

What Object-Oriented Programming May Be—and What It Does Not Have to Be, O. Lehrmann Madsen, B. Moller-Pedersen; Teaching Object-Oriented Programming Is More Than Teaching Object-Oriented Programming Languages, J. Lindskov Knudsen, O. Lehrmann Madsen; The Mjolner Environment: Direct Interaction with Abstractions, G. Hedin, B. Magnusson; Inheritance as an Incremental Modification Mechanism or What Like Is and Isn't Like, P. Wegner, S. B. Zdonik; GSBL: An Algebraic Specification Language Based on Inheritance, S. Clerici, F. Orejas; Name Collision in Multiple Classification Hierarchies, J. Lindskov Knudsen; Reflexive Architecture: From ObjVLisp to CLOS, N. Graube; Nesting in an Object-Oriented Language is NOT for the Birds, P. A. Buhr, C. R. Zarnke; An Object-Oriented Exception Handling System for an Object-Oriented Language, C. Dony; On the Darker Side of C++, M. Sakkinen; Prototyping an Interactive Electronic Book System Using an Object-Oriented Approach, J. Pasquier-Boltuck, E. Grossman, G. Collaud; SCOOP, Structured Concurrent Object-Oriented Prolog, J. Vaucher, G. Lapalme, J. Malenfant; The Implementation of a Distributed Smalltalk, M. Schelvis, E. Bledoeg; Implementing Concurrency Control in Reliable Distributed Object-Oriented Systems, D. G. Parrington S. K. Shrivastava; An Implementation of an Operating System Kernel Using Concurrent Object-Oriented Language ABCL/c+, N. Doi, Y. Kodama, K. Hirose; Debugging Concurrent Systems Based on Object Groups, Y. Honda, A. Yonezawa; Fitting Round Objects Into Square Databases (invited paper), D. C. Tsichritzis, O. M. Nierstrasz; Database Concepts Discussed in an Object-Oriented Perspective, Y. Lindsjorn, D. Sjoberg; Object-Oriented Programming and Computerised Shared Material, P. Sorgaard; Asynchronous Data Retrieval from an Object-Oriented Database, J. P. Gilbert, L. Bic; An Overview of OOPS+, An Object-Oriented Database Programming Language, E. Laenens, D. Vermeir; PCLOS: A Flexible Implementation of CLOS Persistence, A. Paepcke; A Shared, Persistent Object Store, C. Low.

The Volume is recommanded all people interested in object oriented programming. The proceeding gives a good review on this topic.

János Toczki

J. Balcázar, J. Diaz, J. Gabarró: Structural Complexity I (EATCS Monographs on Theoretical Computer Science, Vol 11). IX+191 pages, Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo, 1988.

"This book assumes as a prerequisite some knowledge of the basic models of computation, as taught in an undergraduate course on Automata Theory, Formal Language Theory, or Theory of Computation. Certainly, some mathematical maturity is required, and previous exposure to programming languages and programming techniques is desirable. Most of the material of Volume I can be successfully presented in a senior undergraduate course; Volume I and II should be suitable for a first graduate course. Some sections lead to a point in which very little additional work suffices to be able to start research projects. In order to ease this step, an effort has been made to point out the main references for each of the results presented in the text.

Thus, each chapter ends with a section entitled "Bibliographical Remarks", in which the relevant references for the chapter are briefly commented upon. These sections might also be of interest to those wanting an overview of the evaluation of the field. Additionally, each chapter (excluding the first two, which are intended to provide some necessary background) includes a section of exercises."

The contents of the book are the following:

— Introduction,
— Time and Space Bounded Computations,
— Central Complexity Classes,
— Time Bounded Turing Reducibilities,
— Nonuniform Complexity,
— Probabilistic Algorithms,

— Uniform Diagonalization,
— The Polinomial Time Hierarchy.

The book is clearly written. It can be recommended as a text for a graduate course and for people interested in Complexity Theory.

J. Csirik

P. Deransart, M. Jourdan and B. Lorho: Attribute Grammars, Definitions, Systems and Bibliography. (Lecture Notes in Computer Science. Vol. 323), IX+232 pages, Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo, 1988.

Attribute Grammars have proved to be an efficient tool for the description of syntax-directed computations. This book gives a concize and comprehensive survey of attribute grammars and their applications. The book consists of three parts:

1. In the first part (Definitions and Main Results) the theoretical results achieved by attribute grammars are presented.

2. In the second part more than 40 systems are described. These descriptions contain the following parts:

— a list of the members of the project,
— the birthdate and deadline of the project,
— general features of the system,
— a scheme of the internal organization of the systems,
— optimizations implemented in the system,
— applications and performances of the systems,
— future projects,
— references.

3. In the third part a bibliography of about 600 titles pertaining to AG-s is included.

T. Gyimóthy

H. A. Eiselt, G. Pederzoli (Editors): **Advances in Optimization and Control** (Lecture Notes in Economics and Mathematical Systems, Vol. 302), VIII+371 pages, Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo, 1988.

This book provides a collection of refereed papers presented at the Conference "Optimization Days 85" held at Montreal, Canada, April 30—May 2, 1986. It is first time that the proceedings of the Optimization Days is published in this way.

The papers cover a fairly wide variety of fields in optimization and control. The contents of the book are the following: J. Jones, Jr.: Matrix Differential Equations and Lyapunov Transformations; Quan Zheng: Theory and Methods for Global Optimization—an Integral Approach; E. A. Galperin: The Beta-Algorithm for Mathematical Programming; M. J. Todd: Polynomial Algorithms for Linear Programming; F. Chauny, R. Loulou, S. Sandones, F. Soumis: A Class of Asymptotically Optimal Strip-Packing Heuristics; M. Gendreau, J.-C. Picard, L. Zubieta: An Efficient Implicit Enumeration Algorithm for the Maximum Clique Problem; A. G. Ferreira: An Optimal $O(n)$-Algorithm to Fold Special PLA's; M. P. Helme: A Mixed Integer Programming Model for Planning an Integrated Services Network; G. Lapalme, J.-Y. Potvin, J.-M. Rousseau: A General Heuristic for Node Routing Problems; J. Desrosiers, Y. Dumas: The Sortest Path Problem for the Construction of Vehicle Routes with Pick-Up, Delivery and Time Constraints; G. Laporte, Y. Nobert: A Vehicle Flow Model for the Optimal Design of a Two-Echelon Distribution Problem; L. Jenkins: An Approximate Solution to a Capacitated Plant Location Problem Under Uncertain Demand; I. J. Curiel, G. Pederzoli, S. H. Tijs: Reward Allocations in Production Systems; M. Breton, P. L'Ecuyer: On the Existence of Sequential Equilibria in Markov Renewal Games; P. L'Ecuyer J. Malenfant: Computing Optimal Checkpoint Policies: A Dynamic Programming Approach; S. P. Sethi, C. Bes: Dynamic Stochastic Optimization Problems in the Framework of Forecast and Decision Horizons; J. B. Lasserre: Decision Horizon, Overtaking and 1-Optimality Criteria in Optimal Control; O. Hajek, K. A. Loparo: Bilinear Control: Geometric Properties of Reachable Sets; D. A. Carlson: Sufficient Conditions for Optimality and Supported Trajectories for Optimal

Control Problems Governed by Volterra Integral Equations; G. Bojadzeiv: Behavioural Strategy of Some Controlled Predator-Prey Systems; H. Y. Wan, Jr., S. Clemhout: A General Dynamic Model of Bargaining—The Perfect Information Case; C. Deissenberg: Long-Run Macroeconometric Stabilization Under Bounded Uncertainty; Z. Ritz, D. Sudharshan: An Evolutionary Analysis of Product-Preference Structure: Toward Managerial Control; J.-P. Amigues, G. Gaudet, M. Moreaux: Bertrand and Cournot Equilibrium Price Paths in a Nonrenewable Resource Differentiated Product Duopoly; B. Tolwinski: A Renegotiation—Proof Solution for a Price Setting Duopoly.

These papers provide a relatively quick access to some of the most important new ideas in optimization. For example, the paper of Zheng is one of the first publications in English about this new global optimization method. Todd's survey on polynomial algorithms for linear programming is excellent. Overall, the book gives a good selection of the latest research results in optimization, and it can be recommended for those interested in the state of the art in this field.

Tibor Csendes

H. Edelsbrunner: Algorithms in Combinatorial Geometry (EATCS Monographs on Theoretical Computer Science, Vol. 10) XV+423 pages, Springer-Verlag, Berlin—Heidelberg—New York—London—Paris—Tokyo, 1987.

From the preface of the book: "Computational geometry as an area of research in its own right emerged in the early seventies of this century. Right from the beginning, it was obvious that strong connections of various kinds exist to questions studied in the considerably older field of combinatorial geometry. For example, the combinatorial structure of a geometric problem usually decides which algorithmic method solves the problem most efficiently. Furthermore, the analysis of an algorithm often requires a great deal of combinatorial knowledge. As it turns out, however, the connection between the two research areas commonly referred to as computational geometry and combinatorial geometry is not as lop-sided as it appears. Indeed, the interest in computational issues in geometry gives a new and constructive direction to the combinatorial study of geometry.
and combinatorial geometry is not as lop-sided as it appears. Indeed, the interest in computational issues in geometry gives a new and constructive direction to the combinatorial study of geometry.

It is the intention of this book to demonstrate that computational and combinatorial investigations in geometry are doomed to profit from each other. To reach this goal, I designed this book to consist of three parts, a combinatorial part, a computational part, and one that presents applications of the results of the first two parts. The choice of the topics covered in this book was guided by my attempt to describe the most fundamental algorithms in computational geometry that have an interesting combinatorial structure. In this early stage geometric transforms played an important role as they reveal connections between seemingly unrelated problems and thus help to structure the field. These transforms led me to beleive that arrangements of hyperplanes are at the very heart of computational geometry- and this is my belief now more than ever.

As mentioned above, this book consists of three parts: I. Combinatorial Geometry, II. Fundamental Geometric Algorithms, and III. Geometric and Algorithmic Applications. Each part consists of four to six chapters. The non-trivial connection pattern between the various chapters of the three parts can be somewhat untangled if we group the chapters according to four major computational problems. The construction of an arrangement of hyperplanes is tackled in Chapter 7 after Chapters 1, 2, and 5 provide preparatory investigations. Chapter 12 is a collection of applications of an algorithm that constructs an arrangement. The construction of the convex hull of a set of points which is discussed in Chapter 8 builds on combinatorial results presented in Chapter 6. Levels and other structures in an arrangement can be computed by methods described in Chapter 9 which bears a close relationship to the combinatorial studies undertaken in Chapter 3. Finally, space cutting algorithms are presented in Chapter 14 which is based on the combinatorial investigations of Chapter 4 and the computational results of Chapter 10. The above listing of relations between the various chapters is by no means exhaustive. For example, the connections between Chapter 13 and the other chapters of this book come in too many shapes to be described here. Finally, Chapter 15 reviews the techniques used in the other chapters of this book to provide some kind of paradigmatic approach to solving computational geometry problems."

The monograph contains a rich and deap material, which is well-arranged. It can be recommended as an excellent summary of algorithms in combinatorial geometry for a large number of people from students to researchers.

J. Csirik

Subscription information:

For Albania, Bulgaria, China, Cuba, Czechoslovakia, German Democratic Republic, Korean People's Republic, Mongolia, Poland, Romania, USSR, Vietnam
orders should be addressed to:

Kultura
Hungarian Foreign Trading Co.
H—1389 Budapest 62
P. O. Box 149
Hungary

For all other countries orders should be adressed to:

J. C. Baltzer AG
Scientific Publishing Company
Wettsteinplatz 10
CH—4058 Basel
Switzerland

Mailing address for editorial correspondence:

Acta Cybernetica
Aradi vértanúk tere 1.
Szeged
H—6720 Hungary

INDEX — TARTALOM

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Gécseg Ferenc
A kézirat a nyomdába érkezett: 1988 november
Terjedelem: 7,7 (A/5) ív
Készült monószedéssel, ives magasnyomással
az MSZ 6601 és az MSZ 5602—55 szabvány szerint
88-3934 — Szegedi Nyomda — Felelős vezető: Surányi Tibor igazgató