УДК 519.95

СИСТЕМА АЛГОРИТМИЧЕСКИХ АЛГЕБР И НЕКОТОРЫЕ СХЕМЫ УПРАВЛЕНИЯ В ОДНОРОДНЫХ СТРУКТУРАХ

Г.Е.Цейтлин

1. В настоящее время получила распространение концепция структурного программирования [5], используемая в основе практически любого комплекса технологических средств, ориентированных на разработку больших программ, к которым относятся, в частности, системы математического обеспечения (СМО) для однородных вычислительных структур (ОВС) [7, 10, 11].

К числу основных конструкций структурного программирования относятся: композиция (последовательное применение программных модулей); альтернативный выбор из совокупности возможных продолжений процессов вычислений; повторение программных модулей вплоть до выполнения некоторого предиката.

Подобная стандартизация средств программирования способствует минимизации количества ошибок при разработке больших программ, ускорению темпов их создания, удобству в сопровождении и гибкой адаптации программ к изменению состава СМО вычислительной системы. Процесс разработки структурированных алгоритмов и программ тесно связан с развитием схематологии структурного программирования. При этом особенно важным является изучение логических схем больших программ, без учета которых невозможно эффективное решение задач, связанных с рассмотрением информационных зависимостей и структуры памяти. Таким образом, как и в случае традиционной схематологии [8] для структурного программирования весьма актуально развитие теории логических схем структурированных программ, аналогичных схемам Янова. В качестве методологической основы структурного программирования можно рассматривать предложенный В.М.Глушковым [1-3] аппарат алгоритмических алгебр, представляющий собой теорию схем структурированных алгоритмов и программ.

В настоящей работе рассматриваются возможности использования систем алгоритмических алгебр с целью формализации ряда важных системных процессов, ориентированных на мультиобработку. Приводится концепция автоматов с n-внутренними лентами, обобщающая известные автоматные структуры над различным образом организованной внутренней памятью. Решается проблема тождеств для $S(0)$ – алгебр, ориентированных на формализацию схем управления системными процессами.

2. Понятие систем алгоритмических алгебр ассоциируется с абстрактной моделью ЭВМ [I], которая представляет собой композицию операционного и управляющего автоматов. В качестве операционного автомата естественно рассматривать многомерные однородные структуры [I5], в качестве управляющего – конечный автомат.

Алгоритмические алгебры представляют собой многоосновную алгебраическую систему, состоящую из двух базовых множеств: множества операторов $\mathfrak{A}$ и множества логических условий $\mathcal{G}$. Операторы являются преобразованиями (вообще говоря, частичными) информационного множества $\mathcal{M}$ состояний операционной структуры в себя. Логические условия определены на множестве $\mathcal{M}$ и принимают в каждом состоянии одно из следующих истинностных значений: 0 – ложь, μ – неопределенность, 1 – истина.

В сигнатуру операций системы алгоритмических алгебр входят обобщения распространенных программистских конструкций, которые находятся в полном соответствии с концепцией структурного программирования:

Композиция $A \times B$ – последовательное применение $A, B \in \mathfrak{A}$;

α-дизъюнкция $[\alpha](A \vee B)$ – условный переход, передающий управление в зависимости от истинности (лжи) условия α оператору $A(B)$;

α-итерация $_{\alpha}\{A\}$ – циклическое повторение оператора A при $\alpha = 0$ до тех пор, пока условие α не примет значение I.

Пусть $\tilde{A} = \{A_i \mid i \in I\}$, $\tilde{\mathfrak{X}} = \{x_j \mid j \in \mathfrak{I}\}$ – совокупность элементарных операторов и условий, составляющих систему образующих системы алгебраических алгебр $\langle \mathfrak{A}, \mathcal{G} \rangle$. Представление оператора $A \in \mathfrak{A}$ посредством суперпозиции операции системы алгебраических алгебр над образующими $\tilde{A}, \tilde{\mathfrak{X}}$ называется регулярной схемой данного оператора. Условные переходы в регулярных схемах осуществляются с помощью α-дизъюнкций и α-итераций в направлении вперед; скачки назад возможны лишь при переходе от концов итерационных скобок к

соответствующим началам. Несмотря на ограниченность регулярных схем, по сравнению с традиционной схематологией имеет место следующее утверждение, принадлежащее В.М.Глушкову [I].

ТЕОРЕМА I. Любой алгоритм (в частности, программа или микропрограмма) представим некоторой регулярной схемой в системе алгоритмических алгебр. Существует процедура регуляризации, состоящая в построении регулярных схем для произвольных алгоритмов.

В сигнатуру операций системы алгоритмических алгебр входят также обобщенные булевы операции [I-3]. Для 3-значной логики $\mathcal{G}(E_3)$, порожденной обобщенными булевыми операциями, разработаны процедуры сведения выражений к формам α-полиномов, по аналогии с теорией д.н.ф. алгебры логики, и к совершенным α-полиномам (каноническим формам представлений в $\mathcal{G}(E_3)$); так что справедлива следующая

ТЕОРЕМА 2. Произвольное выражение алгебры $\mathcal{G}(E_3)$ единственным образом представимо в форме совершенного α-полинома.

В частности, совершенным α-полиномами является д.н.ф. $D = \bigvee_{i=1}^{k} \mathcal{U}_i$, где $\mathcal{U}_i$ – элементарная конъюнкция.

СЛЕДСТВИЕ. Если выражение $F(x_1, x_2, \ldots, x_n)$ алгебры $\mathcal{G}(E_3)$ представимо в д.н.ф., то подобное представление единственно.

Таким образом, в алгебре $\mathcal{G}(E_3)$ проблема представимости функций в классе д.н.ф. принципиально отличается от одноименной проблемы алгебры логики. В частности, для $\mathcal{G}(E_3)$ невозможна минимизация д.н.ф.

В силу принципа двойственности [I4], соответствующие результаты имеют место в связи с представлением выражений алгебры $\mathcal{G}(E_3)$ в форме конъюнктивных полиномов к.н.ф. Алгебра $\mathcal{G}(E_3)$ образует собственный подкласс в классе 3-значных функций, сохраняющих константу μ, причем получена характеристика данного подкласса в терминах таблиц истинности.

В [I4] сигнатура системы алгоритмических алгебр была дополнена операциями дизъюнкции $A \vee B$ (параллельного применения операторов A, B) и фильтрации, порождающей операторы-фильтры

$$\alpha = \begin{cases} E, \text{если } \alpha = 1, \\ N - \text{в противном случае}, \end{cases}$$

где E – тождественный и N – неопределенный операторы. Для дизъюнкции $A \vee B$ предполагается, что $\mathcal{M}(A,B) = \mathcal{M}(A) \cap \mathcal{M}(B) = \phi$, где $\mathcal{M}(A)$, $\mathcal{M}(B) \subseteq \mathcal{M}$ – информационные области применения операторов A и B. Если $\mathcal{M}(A,B) \neq \phi$, то в любом состоянии $q \in \mathcal{M}(A,B)$ операторы A, B функционируют одинаково, либо по крайней мере один из них неопределен в состоянии q.

При организации обменов между параллельными процессами в систему образующих системы алгоритмических алгебр могут быть включены операторы обменных взаимодействий, предложенных в [IO]. Ожидания в системе алгоритмических алгебр реализуются посредством операторов-семафоров, $F(\alpha) = \underset{\alpha}{\{E\}}$, которые останавливают параллельный процесс до тех пор, пока не выполняется условие α. Перечисленные средства представляют собой гибкий аппарат для формализации в рамках системы алгоритмических алгебр схем параллельных алгоритмов и программ.

3. Рассмотрим автоматы над внутренней памятью. Естественным обобщением ряда широко распространенных автоматных структур служат автоматы с n-вспомогательными лентами [3].

Рассмотрим конечный автомат, который, кроме входного и выходного каналов, имеет каналы для работы с n-внутренними лентами $(n \geq 0)$. Каждая из подобных лент представляет собой память, разделенную на ячейки, которые последовательно занумерованы положительными целыми числами от I до ∞. Номер ячейки является ее адресом при обращении к памяти. Любая ячейка может содержать один символ (букву) из множества $\mathcal{Z}$, не обязательно конечного и названного алфавитом данной ленты. Содержимое ленты располагается в ее верхних ячейках, остальные ячейки остаются пустыми. Пусть $x = a_1 a_2 \ldots a_m$ – содержимое ленты. При записи в ячейку i символа a "хвост" $a_i a_{i+1} \ldots a_m$ смещается на одну ячейку вниз и в освободившуюся ячейку i помещается символ a. В режиме чтения из ячейки i происходит ее опустошение, сопровождающееся сдвигом подцепочки $a_{i+1} \ldots a_m$ на одну ячейку вверх. Рассмотренная лента служит естественным обобщением

структур памяти, положенных в основу известных концепций бесконечных автоматов (магазин, бобслей, счетчик и пр.) [3]. Кроме вспомогательных (внутренних) лент, имеются также входная и выходная ленты, на которых соответственно размещаются входные и накапливаются выходные цепочки.

Автомат A с n-внутренними лентами представляет собой объект

$$A = (S, \mathcal{X}, \mathcal{Y}, \mathcal{Z}_1, \ldots, \mathcal{Z}_n, \varphi, \varphi_i, \psi_i, \delta, s_0, F),$$

где S – конечное множество состояний автомата A, $\mathcal{X}$ – входной, $\mathcal{Y}$ – выходной алфавиты; $\mathcal{Z}_i$ – алфавит i-й внутренней ленты; $\varphi: S \times \{\mathcal{X} \cup \Lambda\} \to S$ – функция переходов, связанная с чтением входной цепочки (символ Λ используется для переключений автомата в новое состояние без обращения к входной ленте); $\varphi_i: S \times \mathcal{Z}_i \to S$ – функция переходов, связанная с чтением из i-й внутренней ленты; $\psi_i: S \to S \times \mathcal{Z}_i$ – функция записи на i-ю внутреннюю ленту; $\rho: S \to \mathcal{Y} \times S$ – функция выходов; $s_0 \in S$ – начальное состояние; $F \subset S$ – множество заключительных состояний автомата A, $i = 1, 2, \ldots, n$.

В каждый момент времени автомат A осуществляет одно из следующих элементарных действий: чтение с входной ленты; чтение с внутренней ленты i; запись на внутреннюю ленту i; запись на выходную ленту; групповую операцию, состоящую в одновременном обращении к нескольким различным лентам (или отдельным участкам памяти на одной из них); переключение автомата A в очередное состояние без обращения к его лентам.

Перечисленные действия сопровождаются однозначными переходами автомата A в очередное состояние.

Конкретизация памяти, используемой в автоматах с n-внутренними лентами, приводит к известным концепциям бесконечных автоматов. В частности, если в качестве внутренних лент выбраны магазины, получаем класс n-магазинных автоматов. Наряду с магазинной при реализации ряда важных системных процессов используется память типа бобслей, организованная по принципу циклической очереди. В [I7] исследовалась проблема представимости языков в δc-автоматах с одним внутренним бобслеем. В частности, имеет место следующая

ТЕОРЕМА 3. *Для любой машины Тьюринга T существует эквивалентный δc-автомат A, воспринимающий то же множество цепочек, что и машина T, так* что $L(A) = L(T)$, где $L(A), L(T)$ – языки, представимые автоматом A и машиной T соответственно.

Из данной теоремы следует, что в классе автоматов с одной внутренней лентой представимы произвольные рекурсивно-перечислимые языки.

Организация взаимодействия автоматных структур над внутренней памятью может быть использована при реализации системных процессов, возникающих в связи с конструированием СМО для вычислительных систем и однородных структур [6,7,II]. Проиллюстрируем возможности применения аппарата системы алгоритмических алгебр для формализации алгоритмов функционирования автоматных структур на следующих важных примерах.

Двусторонний параллельный анализ. В работе [I7] была выдвинута принципиально новая стратегия параллельного анализа, состоящая во встречной обработке входной цепочки с противоположных ее концов двойственными методами развертки и свертки. Была разработана автоматная модель двустороннего ΠP-анализатора, которая нашла свое дальнейшее развитие в [I3 и I8].

Представим функционирование двупроцессорного ΠP-анализатора над общей памятью в системе алгоритмических алгебр посредством следующей регулярной схемы:

$$\overleftrightarrow{S}_1 = \{ R \vee C \}_{L\rho \leq k} \{ R \}_{L\rho = 0}\ 'm_\tau = 'm_c ,$$

где R, C – процессы развертки и свертки соответственно; $L\rho$ – расстояние на входной ленте между указателями, фиксирующими фазу двустороннего анализа; k – коэффициент синхронизации, зависящий от характеристик двупроцессорной системы, на которой реализован ΠP-анализатор; $'m_\tau \ ('m_c)$ – содержимое магазина развертки (свертки). При выполнении условия $L\rho \leq k$ процесс C отключается, в то время как R продолжает работу до слияния указателей на входной ленте. Анализируемая цепочка принадлежит входному языку, если в момент слияния указателей совпадает содержимое магазинов $\mathcal{M}_\tau$ и $\mathcal{M}_c$.

Регулярная схема ΠP-анализатора, реализованного на двупроцессорной системе с разделенной памятью, имеет вид

$$\overleftrightarrow{S}_2 = \{ R \}_{\alpha_1} F(\alpha_2) \{ R \}_{L\rho = 0}\ \underline{'m_\tau = 'm_c} \vee \{ C \}_{\alpha_3}\ 'm_c \Rightarrow R .$$

Процессы R и C работают параллельно и независимо друг от друга, выполняя циклы развертки и свертки по условиям α_1 и α_3, соответственно. Затем процесс R осуществляет ожидание по семафору $F(\alpha_2)$, который сигнализирует о завершении процесса C и о поступлении по каналу связи в R содержимого $'m_c$ посредством засылки $'m_c \Rightarrow R$. Далее R продолжает работу вплоть до слияния указателей на входной ленте.

В соответствии с концепцией программирования "сверху-вниз" дальнейшая детализация схем $\overleftrightarrow{S}_1$ и $\overleftrightarrow{S}_2$ связана с уточнением процессов R и C и организацией проверки условий, входящих в $\overleftrightarrow{S}_1$ и $\overleftrightarrow{S}_2$. На основании приведенных регулярных схем разработаны программные модели ΠP-анализаторов для ЭВМ "Мир-2" [I2] и вычислительной системы "Мир-2 – БЭСМ-6" [I6]. Изучению теории грамматических автоматных моделей параметрического типа, ориентированных на эффективный двусторонний анализ и его обобщения, посвящены [I3,I7,I8].

Распределение ресурсов. Рассмотрим ленту D с алфавитом из $\tilde{\mathcal{V}} = \mathcal{V} \cup P$. Пусть имеет место разбиение $R(\mathcal{V}) = \{ \mathcal{V}_i \mid i \in I \}$ такое, что каждому классу $\mathcal{V}_i$ сепоставлен маркер из множества $P = \{ *_i \mid i \in I \}$, названный приоритетом данного класса. Приоритетное множество P упорядочено по возрастанию индексов так, что $*_i \leq *_j$, если $i \leq j$. Лента D допускает чтение из вершины (ячейка с номером I); запись символов из класса $\mathcal{V}_i$ возможна лишь в маркерные ячейки τ, содержащие приоритет данного класса $'\tau = *_i$. При необходимости инициировать очередь с максимальным приоритетом соответствующий маркер записывается в вершину ленты. Кроме того, в ячейку τ' (с номером $\tau+1$), соседнюю с маркерной τ, где $'\tau = *_i$, допускается запись маркера $*_j \leq *_i$, который удовлетворяет соотношению $*_j > '\tau_k$ для любых маркерных ячеек τ_k, расположенных ниже τ и таких, что $'\tau_k = '\tau$. Ленту, организованную подобным образом, назовем метаочередью.

Пусть необходимо распределить поток L приоритетных заявок, поступающих в метаочередь D, на множество свободных процессоров, имена которых хранятся в очереди Π. Несколько модифицируя модель распределителя памяти, предложенную в [I7] и запрограммированную в [I2], рассмотрим следующую регулярную схему распределения заявок по процессорам:

$$\mathcal{W}: \{ \underbrace{\alpha_5}_{\alpha_1 \wedge \alpha_2} W_1(\Pi) \vee \underbrace{\bar{\alpha}_1}_{} W_1(D) \vee F(\alpha_5)\{ \underbrace{D \Rightarrow \Pi}_{\alpha_3 \vee \alpha_4} \}\}!,$$

где α_1 - условие пустоты потока L; α_2 - все процессоры свободны; α_3 - метаочередь D пуста; α_4 - очередь Π пуста, α_5 - существуют свободные процессоры; $W_1(D)(W_1(\Pi))$ - загрузка метаочереди D (очереди Π); $D \Rightarrow \Pi$ - отсылка очередной заявки из D на свободный процессор, имя которого считано из Π; $F(\alpha_5)$ - семафор, осуществляющий ожидание при отсутствии свободных процессоров; $!$ - оператор останова.

Отличительной чертой схемы $\mathcal{W}$ является распределение заявок по процессорам с параллельной подкачкой потока L в метаочередь D и свободных процессоров - в очередь Π. Дальнейшая детализация схемы $\mathcal{W}$ связана с организацией проверки логических условий, входящих в $\mathcal{W}$, и описанием операторов $W_1(D), W_1(\Pi), D \Rightarrow \Pi, F(\alpha_5)$.

Динамический обмен. Пусть $\tilde{\Pi} = \{P_i | i \in I\}$ - совокупность процессоров, на которых реализован большой программный комплекс процессов, тесно связанных друг с другом по операциям обмена. При взаимодействии параллельных процессов и организации обмена данными между ними часто, в силу динамики взаимодействия, невозможно заранее установить момент выработки данного на одном из процессоров и момент, когда оно может потребоваться другим процессорам. Вместе с тем решение проблемы оперативного обмена существенно влияет на скорость вычисления. Поэтому наряду с универсальным набором обменных взаимодействий [10] целесообразно рассмотреть автоматную схему (динамический обмен) ДО. В качестве внутренней памяти автомата ДО выберем ленту, разделенную на совокупность полей $\{D_j | j \in \mathcal{J}\}$ по типам пересылаемых данных. Каждое поле D_j состоит из двух зон: зоны P_j, в которой хранятся данные j-го типа, и зоны P'_j с однотипными запросами. Автомат ДО, воспринимая группу данных, определяет их типы и просматривает соответствующие зоны запросов. Информация, при наличии запросов на нее, отсылается в процессоры, от которых поступили указанные запросы. Данные, не нашедшие адресата, размещаются в зонах P_j определенного типа. Аналогично, при поступлении группы запросов, автомат ДО просматривает однотипные зоны данных и при наличии необходимой информации отсылает ее в соответствующие процессоры. Запросы, не нашедшие данных, размещаются в однотипных зонах P'_j, и для их удовлетворения вырабатываются заявки, которые отсылаются распределителю, описанному ранее, либо прямо процессорам с необходимой функциональной настройкой.

Предложенные схемы могут быть использованы при конструировании параллельных трансляторов [16], а также в связи с организацией эффективного анализа системных ситуаций, приоритетном распределении ресурсов, при создании информационных систем и банков данных и для решения других важных задач разработки СМО для современных вычислительных систем [11].

4. Рассмотрим произвольное логическое условие $\alpha \in \mathcal{C}$, которому сопоставим область определенности $\mathcal{M}_0(\alpha) = \{q | \alpha(q) \neq \mu\}$ и область истинности $\mathcal{M}_1(\alpha) = \{q' | \alpha(q') = 1\}$, где $\mathcal{M}_0(\alpha), \mathcal{M}_1(\alpha) \subseteq \mathcal{M}$. Информационное множество $\mathcal{M}' \subseteq \mathcal{M}$ - замкнуто, если для любого оператора $A \in \mathcal{Y}$ имеет место $A(q) \in \mathcal{M}'$ и для всякого $q \in \mathcal{M}'$ такого, что оператор A определен в состоянии q. Множество $\mathcal{M}'$ изолировано, если его дополнение $\mathcal{M} \setminus \mathcal{M}'$ замкнуто. Понятие изолированного множества является частным случаем отношения изолированности, рассмотренного в [2], для универсальных алгебр.

Логическое условие $\alpha \in \mathcal{C}$ назовем замкнутым, если его область определенности $\mathcal{M}_0(\alpha)$ изолирована, а область истинности $\mathcal{M}_1(\alpha)$ замкнута. Содержательно, замкнутые условия характеризуют ситуации, факт свершения которых не зависит от дальнейшего развития процесса. Так, истинность условия, связанного с решением i-й задачи в некоторый момент времени t, не зависит от состояния вычислительной системы в последующие моменты $t' > t$. Замкнутыми также являются условия, входящие в рассмотренные ранее схемы двустороннего анализа. Замкнутые условия могут быть использованы в качестве при организации схем управления в структурированных программах. Систему алгебраических алгебр $\langle \mathcal{Y}, \mathcal{C} \rangle$ с замкнутыми логическими условиями назовем $S(0)$ -алгеброй.

Для $S(0)$ -алгебр разработано аксиоматическое исчисление, в котором доказан ряд тождеств, оформленных в качестве технических лемм и теорем. На основании техники тождественных преобразований для $S(0)$ -алгебр $\langle \mathcal{Y}, \mathcal{C} \rangle$ получены процедуры сведения регулярных схем к стандартным полиномам (каноническим формам в $\langle \mathcal{Y}, \mathcal{C} \rangle$) так, что справедлива следующая

ТЕОРЕМА 4. *Произвольная регулярная схема F в $S(0)$-алгебре единствен-*

ным образом представима в форме стандартного полинома; существует алгоритм стандартизации регулярных схем в $S(O)$ -алгебрах.

В рамках проведенного исследования для $S(O)$ -алгебр доказана разрешимость проблемы тождеств.

ТЕОРЕМА 5. Для $S(O)$ -алгебр разработана конечная полная аксиоматика, характеризующая класс всех истинных тождеств в данных алгебрах; существует алгоритм, на основе которого в $S(O)$ -алгебрах разрешима проблема тождеств.

Полученные результаты использованы при разработке пакета программ для ЭВМ серии "Мир", ориентированного на автоматизацию процесса доказательства теорем (тождеств) в построенном аксиоматическом исчислении. Техника тождественных преобразований в системе алгебраических алгебр может быть использована в качестве подхода к решению проблемы оптимизации структурированных программ по выбранным критериям.

Рассмотренная методология является естественным сочетанием схемных и программных методов конструирования СМО в однородных структурах. На ее основе разработаны методы реализации в однородных структурах средств системы алгебраических алгебр и, в частности, многорегистровых операторов [9], в терминах которых может быть осуществлено семантическое погружение языков программирования в эту систему и их последующая схемная интерпретация. С использованием концепции автоматов над внутренней памятью появляется возможность синтезировать в однородных структурах ряд важных системных процессоров на основе непосредственного вложения в структуру соответствующих регулярных схем. В частности, принципы параллелизма и программной настройки однородных структур позволяют реализовать отдельные модули параметрических систем программирования, ориентированных на мультиобработку.

Идеологическая близость однородных структур к рекурсивным машинам [4], однородным вычислительным сетям с растущим количеством процессоров способствует применению рассмотренных методов при конструировании СМО для рекурсивных машин.

Литература

I. ГЛУШКОВ В.М. Теория автоматов и формальные преобразования микропрограмм. -"Кибернетика", 1965, № 5.с. I-9.

2. ГЛУШКОВ В.М., ЦЕЙТЛИН Г.Е., ЮЩЕНКО Е.Л.Алгебра,языки,программирование. Киев, "Наукова думка", 1974.

3. ГЛУШКОВ В.М., ЦЕЙТЛИН Г.Е., ЮЩЕНКО Е.Л. Теория автоматов и некоторые вопросы синтеза структур языковых процессоров. -"Кибернетика", 1975, № 5, с. I-20.

4. GLUSHKOV V.M., IGNATIEV M.B., MYASNIKOV V.A., TORGACHEV V.A. Rxcursivx Machinxs and computing. Technology IFYP Congress 74. Stockholm,1 974.

5. ДАЛ, ДЕЙКСТРА Э., ХООР И. Структурное программирование.М., "Мир", 1975.

6. ЕВРЕИНОВ Э.В. Теоретические основы построения универсальных вычислительных сред. -В кн.: Вычислительные системы. Вып.16. Новосибирск, 1965, с. 3-72.

7. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск,"Наука", 1966.

8. ЕРШОВ А.П. Об операторных схемах Янова. -В кн.: Проблемы кибернетики. Вып. 20. М., "Наука", 1968, с. I8I-200.

9. КЕКЕЛИЯ В.И., ЦЕЙТЛИН Г.Е. К реализации многорегистровых периодически определенных преобразований в одной абстрактной модели вычислительной среды. -В кн.: Вычислительные системы.Вып.47. Новосибирск, 197I, с. 87-I02.

IO. КОСАРЕВ Ю.Г. О схемах обмена между ветвями параллельных алгоритмов. -В кн.: Вычислительные системы. Вып. 5I. Новосибирск, 1972, с.70-75.

II. МИРЕНКОВ Н.Н. Структурное параллельное программирование.- "Программирование", 1975, № 3, с. 3-I4.

I2. МИЩЕНКО В.В., ЦЕЙТЛИН Г.Е., ШАПОВАЛОВА Н.Н. К вопросам автоматного взаимодействия при реализации некоторых системных процессов. -В кн.:Вопросы обучения языкам программирования. Киев (Институт кибернетики АН УССР), 1974.

I3. ПЕРЕВОЗЧИКОВА О.Л., ЦЕЙТЛИН Г.Е., ШЕВЧЕНКО В.В. О моделях грамматик, ориентированных на двусторонний синтаксический анализ языков. -"Кибернетика", 1976, № 3, с. I-II.

I4. ЦЕЙТЛИН Г.Е. Канонические представления логических условий в системе алгоритмических алгебр. -В кн.: Языки программирования и методы их реализации. Киев (Институт кибернетики АН УССР), 1973.

I5. TSEYTLIN G.E. The the of the modified Post alcebras and multidimxnsioanl automata structures. MFCS 75. Lecture Notes in Computer Science № 32. Prague Czechoslovakia, 1975.

I6. ЦЕЙТЛИН Г.Е., ДОВГОПОЛАЯ Л.И. О моделировании некоторых системных процессов, ориентированных на мультиобработку. -В кн.: Труды школы семинара "Языки обработки данных". Киев, 1976.

17. ЦЕЙТЛИН Г.Е., ЮЩЕНКО Е.Л. О представлении языков в δc-автоматах. -"Кибернетика", 1974, № 6, с. 40-51.

18. ЦЕЙТЛИН Г.Е., ЮЩЕНКО Е.Л. Некоторые вопросы теории параметрических моделей языков и параллельный синтаксический анализ. -В кн.: Труды Всесоюзного симпозиума по методам реализации новых алгоритмических языков. Ч. II. Новосибирск, 1975.

Поступила в ред.-изд.отд.
30 августа 1976 года