

ОБ ОДНОМ ПОДХОДЕ К ПОСТРОЕНИЮ И ИСПОЛЬЗОВАНИЮ  
ОДНОРОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

Ю.Н.Корнев, С.В.Пискунов, С.Н.Сергеев

Один из возможных путей использования вычислительных сред состоит в следующем. По алгоритму решения некоторой задачи строится сеть автоматов, выполняющая этот алгоритм; затем эта сеть реализуется в среде.

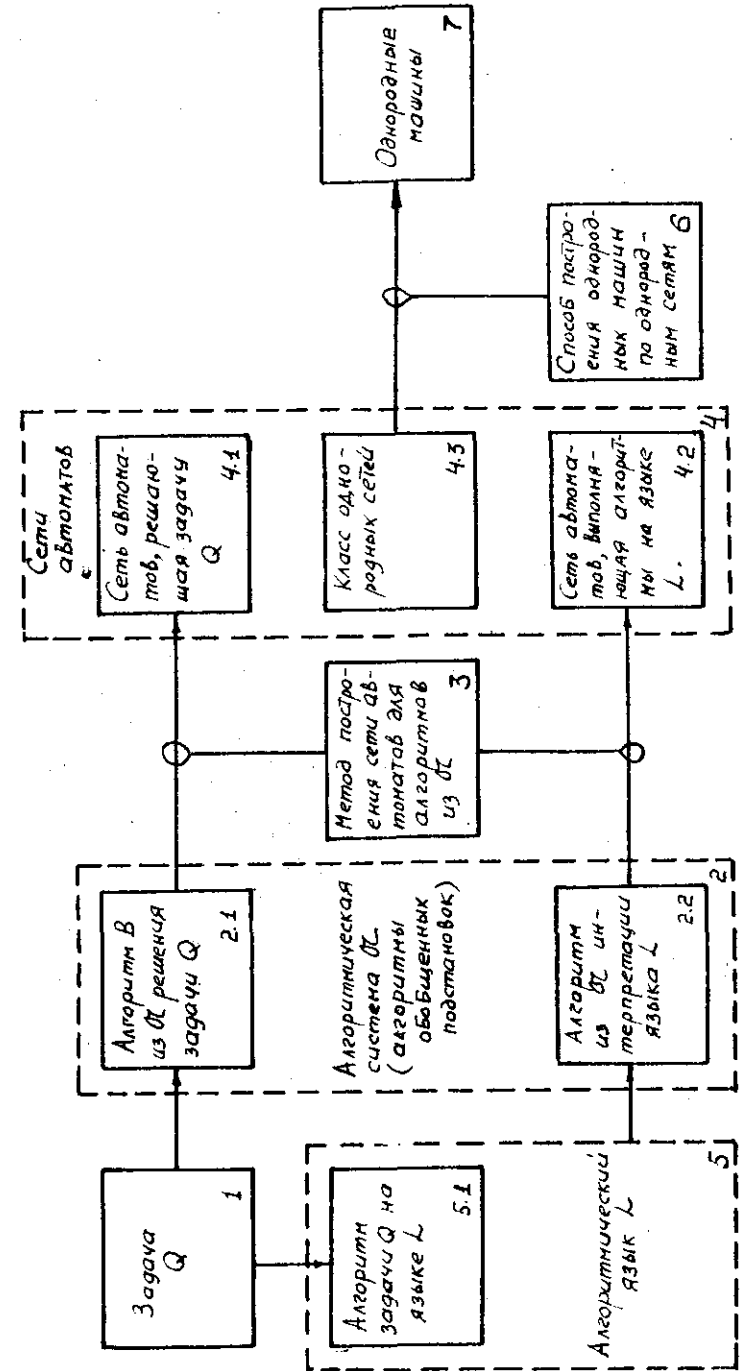
К настоящему времени достаточно хорошо разработаны методы реализации конечных автоматов в однородных средах [1,2]. Поэтому в работе обсуждается такой подход к построению вычислительных устройств, который дает способы перехода от записи алгоритма решения задачи к сети конечных автоматов, выполняющей этот алгоритм.

Отдельные части подхода проработаны довольно тщательно: алгоритмическая система  $\mathcal{A}$  (алгоритмы обобщенных подстановок [3]), способы построения сетей автоматов и однородных машин [3], алгоритмический язык  $\mathcal{L}$  [4]. Цель данной заметки: рассмотрение взаимосвязи отдельных частей предлагаемого подхода и их характерных черт.

I. Структура подхода изображена на рисунке. На этом рисунке можно выделить два пути построения вычислительных устройств.

I.1. Алгоритм решения задачи  $Q$  (или класса задач) записывается в виде алгоритма обобщенных подстановок, по этому алгоритму строится сеть автоматов, выполняющая этот алгоритм (блоки 1, 2.1, 3, 4.1 рисунка). Этот путь дает способ построения специализированного устройства для решения определенного класса задач.

I.2. В виде алгоритма обобщенных подстановок записывается алгоритм интерпретации некоторого алгоритмического языка, и по алгоритму строится сеть, выполняющая этот алгоритм (блоки 1,



5.1, 2.2, 3, 4.2 рисунка). Полученная таким образом сеть будет *hardware* интерпретатором и по сути будет являться универсальной вычислительной машиной. Построение такого интерпретатора возможно для любого алгоритмического языка, однако практический интерес представляет рассмотрение языков с достаточным запасом изобразительных средств и имеющих простой интерпретатор (малое число подстановок в алгоритме интерпретатора). Языком, удовлетворяющим этим требованиям, и является язык из [4]. Динамическая правильность структуры, в которой выполняются операторы, произвольность алфавита и набор операторов — все это выбрано так, что на этом языке можно записывать задачи довольно широкого класса, например, задачи интерпретации алгоритмических языков.

2. Основу излагаемого подхода составляет алгоритмическая система  $\alpha$  — алгоритмы обобщенных подстановок (блок 2). Каждый такой алгоритм перерабатывает исходную информацию (слово  $W$ ), которая представляет собой некоторое конечное множество наименованных клеток, в каждую из которых вписан символ из конечного алфавита  $A$ .

Алгоритм задается конечной системой подстановок вида  $S_1 * S_2 \rightarrow S_3$ , где  $S_j$  — конфигурация — функция от натурального аргумента  $i$ , элементами области значений которой являются подмножества наименованных клеток с одним и тем же числом элементов. Эти значения будем обозначать через  $W_{ji}$ .

Процесс переработки исходного слова  $W$  согласно некоторому алгоритму заключается в следующем. Для каждой подстановки вычисляются значения конфигурации  $S_1 * S_2$  такие, что все  $W_{1i} \cup W_{2i} \subseteq W$ , и соответствующие им  $W_{3i}$ . Затем из слова  $W$  сначала вычитаются все  $W_{1i}$ , а затем добавляются все  $W_{3i}$ . Такое преобразование слова  $W$  производится до тех пор, пока существует хотя бы одно непустое слово  $W_{1i} \cup W_{2i} \subseteq W$ , которое является одним из значений какой либо левой части  $S_1 * S_2$  подстановки из данного алгоритма.

3. Выбор представления информации и правил применения схем алгоритма к этой информации произведены так, что обеспечивают возможность построения формального метода (блок 3) перехода от алгоритма обобщенных подстановок к сети автоматов, выполняющей этот алгоритм. Этот метод пригоден не для всех алгоритмов под-

становок, но для достаточно широкого класса таких алгоритмов (например, сети, предложенные в [5], выполняют алгоритмы из этого класса). Каждый автомат сети имеет число состояний меньшее или равное мощности алфавита  $A$ ; каждый автомат обменивается информацией с "соседями". Число этих соседних автоматов и их относительное расположение в сети зависит от конфигурации подстановок схемы алгоритма. Отметим также, что сеть автоматов адекватно выполняет алгоритм: преобразование информации на одном шаге применения схемы алгоритма совпадает с преобразованием состояний автоматов сети за такт работы этой сети.

4. В этой работе мы не будем рассматривать вложение сетей автоматов в однородные среды. Заметим только, что в том случае, когда сеть автоматов, выполняющая алгоритм из  $\alpha$ , получилась существенно однородной (почти все автоматы сети с точностью до обозначения входных, выходных переменных одинаковы), возможен переход к однородной вычислительной машине, которая моделирует работу исходной сети (блоки 6,7 рисунка). Этот переход базируется на представлении автоматов сети в виде устройств, состоящих из трех частей: памяти для хранения таблицы переходов автомата, устройства сравнения строк левой части таблицы переходов и входов автомата и памяти (внутреннее состояние автомата), в которую записывается строка правой части таблицы переходов в случае успешного сравнения. В этом случае представляется возможность поместить таблицу переходов одинаковых автоматов в одну общую для них всех память. Более того, записывая в такую память различные таблицы переходов, можно заставить сеть выполнять различные алгоритмы. Устройство, построенное таким способом, назовем однородной вычислительной машиной.

#### Л и т е р а т у р а

1. ЕВРЕЙНОВ Э.В., КОСАРЕВ Ю.Г. "Однородные универсальные вычислительные системы высокой производительности", Наука, 1967.
2. ПРАНГИВИЛИ И.В., АБРАМОВА Н.А., БАБИЧЕВА Е.В., ИГНАТУШЕНКО В.В. "Микроэлектроника и однородные структуры для построения логических и вычислительных устройств". Наука, 1967.

3. КОРНЕВ Ю.Н., ПИСКУНОВ С.В., СЕРГЕЕВ С.Н. "Алгоритмы обобщенных подстановок и вопросы их интерпретации" - "Теоретическая кибернетика", труды семинара, вып.4, 1970. Киев.

4. КОРНЕВ Ю.Н., ПИСКУНОВ С.В., СЕРГЕЕВ С.Н. "Язык *JP-I*. Язык программирования задач переработки слов в произвольной алфавите". - В сб.: Вычислительные системы, вып.41, Новосибирск, Наука, 1971.

5. Yamada H., Amoroso S. "Tessellation Automata". Information and Control, vol. 14, N 3, 1969.