

УДК 681.3

БЫСТРАЯ ОБРАБОТКА СИМВОЛЬНОЙ ИНФОРМАЦИИ В СИСТЕМАХ УПРАВЛЕНИЯ ДОКУМЕНТАМИ И ПРИНЯТИЯ РЕШЕНИЙ: РЕАЛИЗУЮЩИЕ УСТРОЙСТВА

Корольков О.Ф., Апальков В.В., Чаплыгин А.А., Королькова В.О.

ГОУ ВПО «Юго-Западный государственный университет», Курск, Россия (305040, г. Курск, ул. 50 лет Октября, 94)

Рассматриваются два варианта универсальных устройств символической обработки, реализующих модифицированную производственную алгоритмическую систему А.А. Маркова и принципы их работы, а также реализацию производств с алфавитными переменными и переменной длиной образца и модификатора, организацию памяти для обрабатываемого слова таким образом, чтобы не образовывалось «мусора». Производственные алгоритмы могут быть использованы для высокоскоростной обработки документов, формуляров, методов структурного распознавания образов и организации баз данных и знаний с высокоскоростным доступом. Работа с одной производственной алгоритма заключается в просмотре (сопоставлении) посимвольно обрабатываемого слова и образца продукции для обнаружения позиции его вхождения в обрабатываемое слово. Поиск вхождения образца в обрабатываемом слове выполняется посимвольно, как в устройстве с одним блоком памяти, так и с двумя блоками памяти. Результаты, полученные в данной работе, могут быть использованы в дальнейшем при разработке процессоров обработки символической информации.

Ключевые слова: продукция с алфавитными переменными, продукция с переменной длиной образца и модификатора, скорость обработки.

FAST SYMBOLIC INFORMATION PROCESSING IN DOCUMENTS CONTROL AND DECISION MAKING SYSTEMS: HARDWARE REALISATION

Korolkov O.F., Apalkov V.V., Chaplygin A.A., Korolkova V.O.

South-west state university, Kursk, Russia (305040, Kursk, 50 let Oktyabrya street, 94)

Realization variants and production kinds with variables in symbolic information computing devices with modified production algorithmic Markov's system, their work principles and production realization with alphabetic variables and also variable pattern and modifier length, such memory organization, that there will be no garbage are considered. Production algorithms may be used for high speed document processing, structured pattern recognition methods and database organization with fast access. The process of one algorithm's production begins with the symbol by symbol search in processing word and matching with the pattern for retaining of its position in processing word. The obtained results in this work may be used in the further symbolic information processing units development.

Keywords: productions with alphabetic variables, production of pattern and modification variable size, computing speed.

Введение

Системы поддержки принятия решений в социально-экономических сферах, использование систем компьютерной диагностики на основе структурного распознавания образов, экспертные системы, обработка документов – это далеко не полный перечень задач, для которых можно эффективно использовать аппарат символических вычислений, под которым понимается обработка символической информации в целом, включая компьютерную алгебру. Вместе с тем акселерация вычислений в существенной степени зависит от аппаратной платформы, поэтому в данной работе в прикладном аспекте рассматриваются устройства быстрых производственных вычислений. Названный новый класс устройств быстрых производственных вычислений не требует изменения существующей платформы массовой вычислительной техники, используемой в современных

учреждениях, и применяется в качестве периферийных процессоров-акселераторов с невысокой стоимостью. Доля символьных вычислений в медицинских учреждениях, в издательской деятельности и т.п. значительна, поскольку обработка числовой информации не является доминирующей. Например, в медицине обрабатывается большой объем электронных документов, таких как формуляры, карты больных, заключения и т.д.

Цель данной статьи заключается в создании новых инструментальных средств информационных технологий на основе продукционной парадигмы.

Принципы реализации устройств символьных вычислений

Основные положения теории обработки символьной информации приведены в работах [1–5], а важные для данной статьи положения теории нормальных модифицированных алгоритмов рассматриваются в [1; 4].

Рассмотрим продукцию с алфавитными переменными, которая представлена словом, имеющим следующий вид:

$$a\gamma\psi b \rightarrow d\psi\gamma a, \quad (1)$$

где символ « \rightarrow » – метасимвол, не принадлежащий рабочему алфавиту, отделяющий образец от модификатора; γ , ψ – алфавитные переменные; «a» и «b» и «d» – конкретные символы рабочего алфавита.

Алфавитные переменные конкретизируются значениями символов из соответствующих им позиций в обрабатываемом слове (ОС). Необходимо уточнить, что алфавитные переменные в продукциях могут принимать любые значения из алфавита либо значения из его определенной части.

Введем продукцию с переменной длиной образца и модификатора, которая представлена словом, имеющим следующий вид:

$$a\#b \rightarrow d\#g, \quad (2)$$

где символ «#» в образце означает то, что в ОС между символами «a» и «b» имеется цепочка символов переменной длины, в том числе и пустая. Этот же символ «#» в модификаторе означает то, что он может формироваться как из символов «b» и «g», так и из цепочки, полученной при обнаружении позиции вхождения образца в ОС. Для того чтобы осуществить поиск позиции вхождения образца в ОС и модификацию, аналогичную формуле (2), необходимо применить эквивалентный алгоритм из продукции с алфавитными переменными. Такой алгоритм представляется конечным списком слов, имеющим следующий вид:

$$\begin{aligned} a\gamma\psi\beta b \rightarrow d\gamma\psi\beta g, \\ a\gamma\psi b \rightarrow d\gamma\psi g, \\ a\gamma b \rightarrow d\gamma g. \end{aligned} \quad (3)$$

Из алгоритма (3) следует: чтобы обнаружить продукцию, в которой образец совпадет с фрагментом ОС, необходимо затратить время для выбора соответствующей продукции в процессе работы алгоритма.

Правила работы продукционных устройств с одним и двумя блоками памяти для ОС, реализующих продукцию канонического вида [4], описаны в работах [1]. Структурная схема устройства с двумя блоками памяти для ОС представлена на рисунке 1.

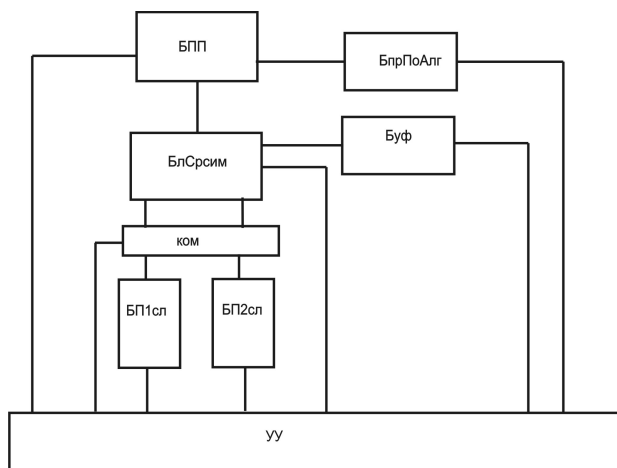


Рис. 1. Структурная схема устройства с двумя блоками памяти для обрабатываемого слова:

УУ – устройство управления; БПП – блок памяти продукций;

БП1сл, БП2сл – блоки памяти обрабатываемого слова;

Ком – коммутатор; БлСрсим – блок сравнения символов;

Буф – буфер; БпрПоАлг – блок перехода по продукциям в алгоритме.

Работа с одной продукцией алгоритма заключается в просмотре (сопоставлении) посимвольно ОС и образца продукции для обнаружения позиции его вхождения в ОС. Такое сопоставление осуществляется в блоке «БлСрсим» (рис. 1) во время перезаписи символов ОС из одного блока памяти в другой, причем считанный символ образца записывается во второй блок. В случае обнаружения позиции вхождения образца в ОС на место совпавшего фрагмента во второй блок памяти ОС записывается модификатор.

Работа продукционных алгоритмов выполняется по модифицированной схеме, которую можно представить конечным списком продукций следующего вида:

$$M1:\gamma_i \rightarrow \beta_j : M2, \quad (4)$$

где γ_i , β_j – цепочки символов заданного рабочего алфавита, несущие тот же смысл, как и у продукции (1), $M1$, $M2$ – метки (натуральные числа) перехода на определенные номера продукции из списка; $M1$ – если продукция не обнаружила своего вхождения при просмотре слова, $M2$ – соответственно если обнаружила, такой алгоритм заканчивает свою работу если продукция имеет

где «a, d» – конкретные символы рабочего алфавита; символ «*» – алфавитная переменная; скобки {} указывают на область изменения значений алфавитной переменной (любой символ алфавита, который располагаются в нем между символами «C₁» и «C₂»); β – модификатор.

Продукцию (2) можно представить словом, имеющим следующий вид:

$$a\#b\$ \rightarrow d\#g\$, \quad (9)$$

где символ «#» в образце означает то, что в ОС между символами «a» и «b» может находиться цепочка символов любой конечной длины; «\$» – признаки конца образца и модификатора.

Сопоставление для символа «*» всегда завершается успешно, а символ ОС, соответствующий этой переменной, записывается как во второй блок памяти ОС, так и в буфер. Замена совпавшей части ОС с образцом осуществляется из символов модификатора продукции, считываемого из блока памяти продукции и записываемого посимвольно во второй блок памяти. Если в модификаторе встречается символ «*», то из блока памяти продукции считывается следующий символ модификатора «2», являющийся порядковым номером, по которому в буфере записан символ из ОС, конкретизирующий алфавитную переменную образца. Этот символ из буфера затем записывается следующим во второй блок памяти ОС вслед за предыдущим.

Структурная схема устройства с одним блоком памяти для ОС представлена на рисунке 3.

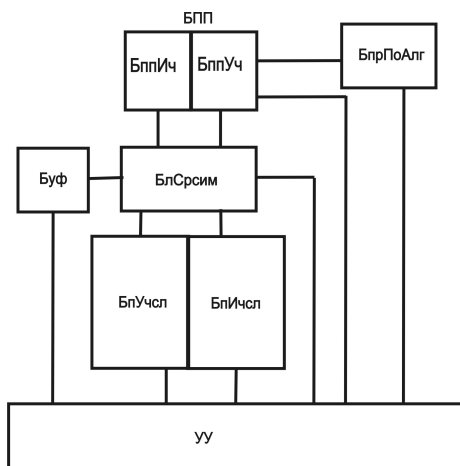


Рис. 3. Структурная схема устройства с одним блоком памяти для ОС: УУ – устройство управления; БПП – блок памяти продукции (Ич – информационная часть, Уч – управляющая часть); ПлПсл – блок памяти ОС (Ич – информационная часть, Уч – управляющая часть); БлСрсим – блок сравнения символов; Буф – буфер.

Рассмотрим структурную организацию варианта устройства, при котором ОС размещено в одном блоке памяти (рис. 3).

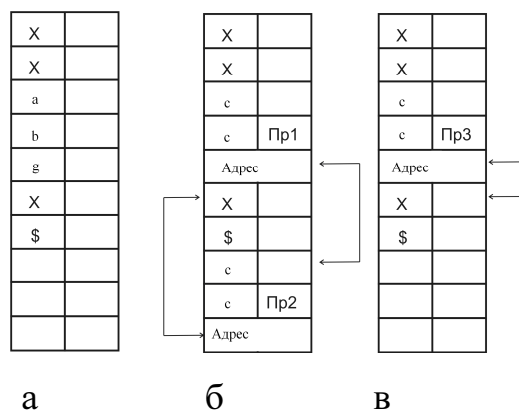


Рис. 4. Схемы расположения ОС в памяти для устройства с одним блоком памяти ОС:
 а – исходное слово; б – окончательный результат, случай, когда длина модификатора больше длины образца; в – окончательный результат, случай, когда длина модификатора меньше длины образца.

Блок памяти для продукций и ОС двухбайтные (информационный и управляющий байты). Признаки концов слова образца и слова модификатора представлены дополнительным управляющим байтом, который записывается в памяти продукции над позициями их последних символов соответственно.

Размещение слова в памяти при различных длинах образцов и модификаторов представлены рис 4. Если длина модификатора больше длины образца, то в управляющей части слова модификатора ставится специальный признак в позиции, равной длине образца, уменьшенной на единицу (рис. 4), продукция (1), признак Пр1. Другой признак Пр2 ставится в конце модификатора (рис. 4), продукция (1), признак Пр2. Если длина модификатора меньше длины образца, то в конце образца в управляющем байте ставится специальный признак Пр3, показывающий, что это признак окончания модификатора (рис. 4), продукция (2), признак Пр3.

Теперь из блока памяти продукций считываются символы модификатора со своими управляющими байтами и записываются в блок памяти ОС на место символов совпавшего фрагмента образца в ОС. Если в управляющем байте определяется один из признаков Пр1, Пр2 или Пр3, то он будет записан в управляющий байт ОС, а в следующие по порядку два байта в ОС будет записан адрес перехода (рис. 4). Если длина модификатора больше длины образца, то таких адресов в ОС будет два: первый адрес – переход в свободный фрагмент памяти, второй адрес – возврат на символ ОС, следующий за позицией окончания совпавших фрагментов ОС и образца (рис. 4б). Если длина модификатора меньше длины образца, то такой адрес один (рис. 4в).

Для реализации на устройстве с одним блоком памяти для ОС продукция (1) будет иметь следующий вид (таблица 1)

Таблица 1 – Побайтное представление продукции

			1				1
		1			2	1	
a	*	*	в	в	*	*	a

Первые две строки таблицы 1 представляют управляющие байты (четыре бита – первая, четыре бита – вторая строки). Третья строка (байт) – символы продукции. Признак «1» в первой строке означает конец образца или модификатора. В этом варианте реализации устройства появляется новая возможность определять равные значения разных алфавитных переменных при их конкретизации при работе продукции. Признак «1» во второй строке в образце указывает, что первая и вторая алфавитные переменные имеют равные значения при их конкретизации. Цифровые символы в позиции записи модификатора в таблице 1 обозначают номера алфавитных переменных в образце, которые считываются из буфера в процессе модификации. Продукция (1) с ограничением на алфавит для этого устройства будет иметь следующий вид (таблица 2).

Таблица 2 – Побайтное представление продукции

				1	1		
	2	2					
a	C ₁	C ₂	*	d	β		

Признак «1» в верхней строке обозначает конец образца и конец модификатора. Признак «2» в следующей строке указывает на то, что код символа, соответствующий «*», может быть любым в диапазоне кодов между C₁ и C₂.

При аппаратной реализации продукции вида (2) на устройстве с одним блоком памяти для ОС (рис. 2) с дополнительным байтом для каждого ее символа организация памяти продукции иллюстрируется таблицей 3.

Таблица 3 – Побайтное представление продукции

		1			1				
a	#	в	b	#	g				

Две первые строки таблицы 3 по аналогии с таблицей 1 представляют собой управляющие байты (четыре бита – первая строка, четыре бита – вторая). Третья строка (байт) – символы рабочего алфавита продукции.

Поиск вхождения образца в ОС выполняется посимвольно, как и в устройстве с двумя блоками памяти (рис. 1). В этом устройстве появляется новая возможность определять равенство значений алфавитных переменных за счет введения дополнительного (управляющего) байта в памяти продукции и, как следствие, изменять обработку символа слова, что позволит вводить специализированные продукции для обработки специализированных текстов.

Заключение

Таким образом, предложены способы реализации продукции с переменными для двух вариантов устройств. Результаты, полученные в данной работе, могут быть использованы в дальнейшем при разработке процессоров обработки символьной информации в качестве периферийных устройств-акселераторов. Это дает возможность существенно повысить скорость обработки символьной информации в системах подготовки и обработки документов в социально-экономических сферах.

Список литературы

1. Корольков О.Ф., Старков Ф.А. Устройство символьной обработки с одним и двумя блоками памяти для обработки слова // Курск. Известия КГТУ.
2. Корольков О.Ф. [и др.] Методы и алгоритмы ветвящихся продукционных вычислений в информационных системах // Курск: ЮЗГУ. – 2011. – 123 с.
3. Корольков О.Ф., Довгаль В.М. Быстрые символьные вычисления: механизмы реализации специальных видов продукции // Информационно-измерительные и управляющие системы. – 2010. – № 7. – Т. 8. – С. 67-71.
4. Марков А.А. Теория алгоритмов / А.А. Марков, Р.М. Нагорный. – М. : Наука, 1984. – 432 с.
5. Успенский В.А. Теория алгоритмов: основные успехи и достижения / В.А. Успенский, А.А. Семенов. – М., 1987. – 288 с.

Рецензенты

Зотов И.В., д.т.н., профессор кафедры «Информационные системы и технологии», Юго-Западный государственный университет, г. Курск.

Дегтярев С.В., д.т.н., профессор кафедры «Информационные системы и технологии», Юго-Западный государственный университет, г. Курск.