УДК 621.001.2

О. В. Малина, доктор технических наук, профессор;
О. Ф. Валеев, аспирант
Ижевский государственный технический университет имени М. Т. Калашникова

ПОДХОДЫ К МИНИМИЗАЦИИ РЕСУРСОВ ЭВМ, НЕОБХОДИМЫХ ДЛЯ РЕАЛИЗАЦИИ ПРОЦЕССА СТРУКТУРНОГО СИНТЕЗА ОБЪЕКТОВ, ПОСТРОЕННЫХ НА ДИСКРЕТНЫХ СТРУКТУРАХ

Реализация процессов структурного синтеза путем комбинаторного поиска сталкивается с проблемой проклятия размерности, что препятствует выполнению структурного синтеза на современных ЭВМ. Предлагаются способы минимизации вычислительной нагрузки и необходимой памяти при выполнении структурного синтеза, основанного на упорядочении признаков, описывающих класс объектов, при их перемножении.

Ключевые слова: системы автоматизированного проектирования, структурный синтез, комбинаторный поиск

Уровень интеллектуализации процессов структурного синтеза при конструировании изделий машиностроения в рамках существующих САПР достаточно низкий, поскольку низок уровень формализации интеллектуальной деятельности инженера. Существующие инженерные расчеты носят в основном проверочный характер, не покрывают весь процесс разработки конструкции изделия, привязаны к конкретной предметной области и не позволяют говорить о создании инвариантного алгоритма структурного синтеза.

Одним из направлений реализации процессов структурного синтеза является комбинаторный перебор на множестве модулей объекта. Множество возможных вариантов, согласно работам [1, 2, 3], может быть представлено следующей формулой:

$$F_{\text{в}} = \prod_{i=1}^{K} (p_i) - F_Z, \qquad (1)$$

где $F_{\text{в}}$ – множество реализуемых вариантов структурного синтеза; K – мощность множества признаков классификатора [4] данного объекта; p_i – i-й признак классификатора, представленный множеством своих значений (исходов): $p_i = \{a_{ij}\}$, где a_{ij} – j-й исход i-го признака; F_Z – множество нереализуемых вариантов, причина нереализуемости которых заключается в невозможности сочетания в рамках одного варианта отдельных значений нескольких признаков, получившего название запрещенной фигуры. Запрещенная фигура [5] – это нереализуемое множество исходов, каждое подмножество которого является реализуемым. По способу получения запрещенные фигуры делятся на эмпирические (полученные путем экспертного опроса) и функциональные (полученные путем вычисления функциональных зависимостей предметной области).

Согласно формуле (1) процесс структурного синтеза включает в себя два этапа: получение множества вариантов (синтез) и удаление запрещенных (анализ).

Для изделий средней степени сложности мощность множества признаков K достигает сотен, поэтому компьютерная реализация комбинаторного перебора по формуле (1) сталкивается с проблемой проклятия размерности, поскольку требует нереально больших объемов оперативной памяти и недостижимо высокой производительности ЭВМ для генерации и вычислений. Таким образом, для выполнения компьютерного синтеза необходима оптимизация, направленная на минимизацию объема расходуемой памяти и вычислительной нагрузки.

Чтобы уменьшить объем расходуемой памяти и снизить вычислительную нагрузку при выполнении синтеза, необходимо выполнить следующие шаги:

• разбить процесс синтеза на элементарные шаги, каждый из которых включает в себя этап синтеза (домножение на очередной признак с получением промежуточных вариантов) и этап анализа (усечение множества промежуточных вариантов путем удаления вариантов, содержащих запрещенные фигуры), что приведет к снижению максимального объема памяти, потребляемой в ходе синтеза;

• минимизировать количество элементарных вычислительных операций, выполняемых при проверке вариантов на запрещенные фигуры, что, очевидно, приведет к снижению вычислительной нагрузки;

• минимизировать количество промежуточных вариантов, формируемых в процессе синтеза, что поможет не только снизить объем памяти, требуемой для выполнения синтеза, но и сократить вычислительную нагрузку.

Результаты исследования показали, что эффективным инструментом реализации указанных выше шагов является упорядочение множества признаков при генерации. Выявим факторы, влияющие на порядок перемножения признаков в процессе генерации.

Первый фактор – это множество функциональных запрещенных фигур (ФЗФ), участвующих в анализе. Данное предположение объясняется тем, что в отличие от эмпирических запрещенных фигур (ЭЗФ), сформированных до начала синтеза в результате проведения экспертного опроса, ФЗФ получаются посредством проведения расчетов. Мощность множества ФЗФ значительно превышает мощность множества ЭЗФ, поэтому признано целесообразным хранить расчетные процедуры, использование которых в процессе синтеза позволяет сформировать множество ФЗФ, а не само множество ФЗФ.

Согласно определению, любая запрещенная фигура, в том числе ФЗФ, – это неупорядоченное множество исходов. Однако расчет (способ получения ФЗФ) является направленной последовательностью действий, преобразующей исходные данные в результат. Таким образом, становится очевидным, что признаки, составляющие множество исходных данных, должны участвовать в декартовом произведении раньше, чем признак-результат. Сделанный вывод позволяет предположить, что, имея множество функциональных зависимостей, признаки можно упорядочить однократно до наложения технического задания на процесс синтеза. Однако на практике часто встречается ситуация, когда техническое задание на синтез объекта содержит значения признаков-результатов, не определяя при этом признаки-аргументы.

Так, например, имея функцию C = f_1(A, B), где A={a_1,a_2}, B={b_1,b_2}, C={c_1,c_2}, естественным порядком признаков для перемножения следует считать A, B, C (или B, A, C). Если техническим заданием задано значение признака A – a_1, то исход a_2 становится запрещенным, и в процессе синтеза расчет C = f_1(A, B) будет выполнен один раз для каждого набора (a_1b_1 и a_1b_2), т. е. 2 раза. Если техническим заданием определено значение признака C – c_1, то в процессе синтеза необходимо будет выполнить расчет по разу для каждого набора (a_1b_1, a_1b_2, a_2b_1, a_2b_2), т. е. 4 раза, что в 2 раза увеличивает вычислительную нагрузку.

Естественно, для сокращения вычислительной нагрузки было бы целесообразно использовать при синтезе вместо функции C = f_1(A, B) функцию A = f_2(B, C), где функция f_2 – обратная к f_1 относительно признака A. Идеальным случаем для минимизации вычислительной нагрузки при любом содержании технического задания является построение семейства функций F, связывающих между собой множество признаков {A, B, C}, отличающихся признаком-результатом: F = {C = f_1(A, B), A = f_2(B, C), B = f_3(A, C)}. Очевидно, что мощность семейства функций равна количеству признаков, на множестве которых определены его функции.

В приведенном примере рассматривались функции, имеющие в качестве результата значение одного признака. Однако зачастую зависимости предметной области представляют собой комплексные расчеты, имеющие в качестве результатов значения нескольких признаков. Например,

$$\begin{cases} p_1 = \sqrt{ab} + \sin(c) \\ p_2 = \dfrac{p_1}{a} + \mathrm{tg}(c) \end{cases} \qquad (2)$$

и

$$p_3 = \sqrt{ab} + \sin(c)\,, \qquad (3)$$

при выполнении неявно вычисляющий значение признака p_4, если существует расчет $p_4 = \sin(c)$.

В этом случае необходимо решить задачу разбиения комплексных расчетов на простые. Для расчета (2) результатом разбиения будут расчеты $p_1 = \sqrt{ab} + \sin(c)$ и $p_2 = \dfrac{p_1}{a} + \mathrm{tg}(c)$, а для расчета (3) – расчет $p_3 = \sqrt{ab} + p_4$ и уже существующий расчет $p_4 = \sin(c)$.

Анализ простых расчетов показал, что расчеты могут различаться вычислительной сложностью, зависящей в первую очередь от типа расчета. Выделены следующие типы расчетов:

• аналитический – выражен формулой, состоящей из элементарных математических операций, реализуемых программно;

• численный – представленный суммой рядов;

• итеративный – реализуемый методом последовательных приближений.

Очевидно, что аналитический расчет имеет меньшую вычислительную сложность, чем численный, а численный – меньшую, чем итеративный. Внутри каждого типа расчеты также имеют различную вычислительную сложность, связанную с количеством элементарных математических операций, называемых атомарными расчетами.

В аналитических расчетах вычислительная сложность определяется количеством элементарных математических операций, в численных – точностью, а в итеративных – точностью, количеством признаков-аргументов и степенью дискретизации каждого признака (длиной интервала и шагом дискретизации значений).

Минимизация вычислительной нагрузки зависит от двух факторов:

1) вычислительной сложности функций, участвующих в процессе синтеза;

2) мощности множества наборов, для которых эта функция будет реализована.

Исходя их этих посылов, необходимо из каждого семейства функций выбрать функцию, участвующую в синтезе. Для решения этой задачи используем графовую модель представления данных, поскольку функция и семейство функций удобно представляются в виде графа. Для этого создадим граф признаков, множество вершин которого соответствует множеству признаков, описывающих класс объектов, а множество ребер – пустое, поскольку связи между признаками не установлены (рис. 1).

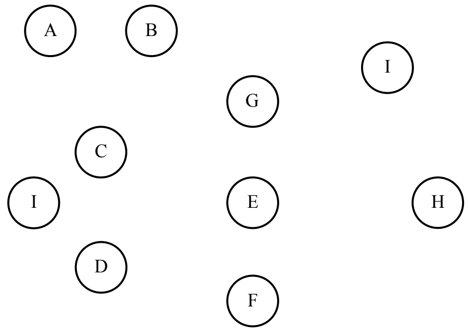

Рис. 1. Исходное множество признаков

Затем дополним граф признаков множеством семейств функций и получим граф, множество вершин которого соответствует множеству признаков и множеству семейств функций, а множество ребер показывает участие признаков в функциях, составляющих семейство функций (рис. 2):

$$G = (V_0, F, E),$$

где V_0 – множество вершин признаков; F – множество вершин семейств; E – множество ребер.

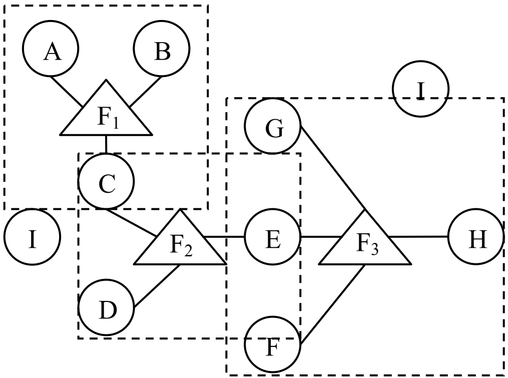

Рис. 2. Совмещение графов семейств функций с множеством признаков

Анализ полученного графа демонстрирует присутствие вершин-признаков, не связанных с вершинами-семействами функций. Это означает, что данные признаки не являются результатами или аргументами расчетных зависимостей предметной области, а следовательно, не оказывают влияния на упорядочение функциональных зависимостей в процессе синтеза. В результате исключения указанных признаков получим граф, множество вершин которого – множество семейств функций и множество признаков, принадлежащих множеству аргументов или результатов функциональных зависимостей, а множество ребер показывает участие признаков в функциях, составляющих семейство функций (рис. 3):

$$G = (V, F, E),$$

где V – множество вершин признаков, участвующих в расчетах; F – множество вершин семейств; E – множество ребер.

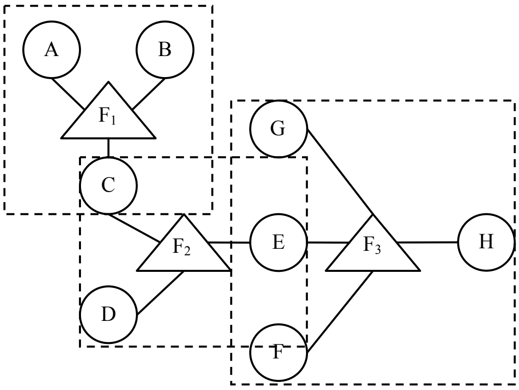

Рис. 3. Граф семейств функций

Задача выбора функции из каждого семейства фактически сводится к задаче ориентации ребер графа семейства функций.

Для выбора функции из каждого семейства введем понятие комплексной оценки вычислительной сложности (КОВС) функции, которую определим как произведение вычислительной сложности функции и мощности множества наборов значений аргументов. Вычислительная сложность функции зависит от типа и определяется количеством атомарных расчетов.

Таким образом, выбор расчета из семейства функций сводится к вычислению КОВС каждого из расчетов семейства и выбору расчета с наименьшей КОВС.

Теорема: пусть семейство функций F, определенное на N признаках, состоит из расчетов $f_1, f_2, ..., f_N$, где расчет f_1 является прямым (аналитическим или численным) с вычислительной сложностью BC_{f_1}, а все остальные расчеты – итеративными. КОВС любого итеративного расчета меньше или равна КОВС прямого расчета f_1, если множество возможных исходов признака, являющегося результатом f_1, состоит из одного элемента.

Доказательство: обозначим множество возможных исходов признака X как K_X КОВС расчета f_1 равна произведению его вычислительной сложности BC_{f_1} и мощностей множеств возможных исходов признаков, являющихся его аргументами: $\prod_{j=2}^{N} \left|\mathrm{K}_{X_j}\right|$. КОВС расчета $f_i, i \neq 1$ равна произведению его вычислительной сложности BC_{f_i} и мощностей множеств возможных исходов признаков, являющихся его аргументами: $\prod_{j=1}^{N} \left|\mathrm{K}_{X_j}\right|, j \neq i$. Вычислительная сложность BC_{f_i} расчета f_i равна произведению BC_{f_1} на число от 1 до K_{X_i}, поскольку для каждого набора аргументов расчета f_i необходимо выполнить расчет f_1 для каждого исхода признака, являющегося результатом расчета f_i, проверяя таким образом тождество до тех пор, пока не будет получен положительный результат. Очевидно, что КОВС расчета f_1 не зависит от $\left|\mathrm{K}_{X_1}\right|$ и равна $BC_{f_1} \prod_{j=2}^{N} \left|\mathrm{K}_{X_j}\right|$. В то же время КОВС расчета f_i равна $BC_{f_1} \prod_{j=1}^{N} \left|\mathrm{K}_{X_j}\right|, j \neq i$, что при $\left|\mathrm{K}_{X_1}\right|$ = 1 обращается в $BC_{f_1} \prod_{j=2}^{N} \left|\mathrm{K}_{X_j}\right|, j \neq i$, что меньше КОВС расчета f_1 в (от 1 до K_{X_i}) раз, что и требовалось доказать.

Таким образом, признаки в пределах семейства функций упорядочены. Однако необходимо упорядочить признаки не только в пределах каждого семейства функций, а все признаки, участвующие в синтезе, т. е. выбрать порядок выполнения расчетов. Порядок выполнения расчетов также следует выбрать таким образом, чтобы сократить объем памяти и вычислительных операций, выполняемых в ходе синтеза. И то, и другое зависит от количества обрабатываемых наборов.

Мощность множества наборов уменьшается тогда, когда некоторые наборы признаются запрещенными, так как содержат в себе запрещенные фигуры:

– эмпирических относительно запрещенных, возникающих вследствие наложения технического задания;

– эмпирических абсолютно запрещенных, полученных до начала процесса синтеза и до наложения ТЗ путем экспертного опроса;

– функциональных запрещенных фигур, поскольку произведенный ранее расчет сужает множество возможных исходов признака, являющегося результатом этого расчета и аргументом данного семейства зависимостей.

Количество промежуточных наборов в результате расчета, как правило, резко сокращается. Поэтому не имеет смысла результат одного расчета домножать по очереди на признаки другого, пока не домножим на последний признак, и только после этого сократим множество промежуточных вариантов за счет выполнения расчета. Логичнее сначала признаки каждого семейства перемножить друг на друга, получив множество промежуточных наборов, разрешенных в рамках данного семейства функций, после чего упорядочить перемножение получившихся множеств вариантов.

Снижение вычислительной нагрузки при формировании множества промежуточных вариантов возможно при снижении количества наборов аргументов, для которых выполняется расчет. Количество наборов аргументов расчета сокращается в большей степени при уменьшении количества возможных исходов каждого признака-аргумента расчета и в меньшей степени при наложении абсолютных ЭЗФ на множество наборов-аргументов расчета (при котором также иногда уменьшается количество возможных исходов какого-либо признака). Учитывая возможность возникновения информационного взрыва уже на стадии формирования наборов аргументов расчета, порядок перемножения признаков, являющихся аргументами расчета, предлагается выбирать таким образом, чтобы приблизить к началу этого процесса проверку на возможно большее число ЭЗФ, определенных на множестве признаков-аргументов расчета.

Количество возможных исходов признака уменьшается непосредственно при наложении технического задания (т. е. относительных ЭЗФ) и опосредованно при применении расчетов (поскольку из множества исходов признака, являющегося результатом расчета, разрешены только значения-результаты расчета). Поскольку каждый признак может участвовать в нескольких семействах функций, то семейства функций взаимосвязаны – сокращение числа возможных исходов признака в рамках одного семейства функций напрямую отражается на количестве промежуточных вариантов, формируемых в рамках других семейств функций, в которых участвует тот же признак. Наличие такой связи между семействами функций позволяет предположить, что обход графа семейств функций позволит минимизировать количество промежуточных вариантов, формируемых в рамках каждого из семейств функций. Для минимизации вычислительной нагрузки обход следует начать с семейства функций с наибольшим коэффициентом сокращения множеств возможных исходов признаков, который определим как $\prod_i \frac{|\text{МПВ}_{i0}|}{|\text{МПВ}_i|}$, где МПВ_{i0} = множество всех исходов i-го признака; МПВ_i = множество возможных исходов i-го признака. После формирования множества промежуточных вариантов по данному семейству функций и вычисления возможных исходов каждого из его признаков логично рассмотреть семейства функций, имеющих в составе один из признаков только что обработанного, поскольку множество его возможных исходов могло быть сокращено. Таким образом, последовательно выбирая семейство функций с наибольшим коэффициентом сокращения множеств возможных исходов признаков, получим множества промежуточных вариантов для всех семейств функций, каждое из которых состоит из вариантов, содержащих исходы признаков семейства функций, и максимально сужено применением ФЗФ, относительных ЭЗФ и абсолютных ЭЗФ, определенных на множестве признаков семейства функций.

Далее необходимо определить порядок перемножения множеств промежуточных вариантов, сформированных в рамках семейств функций.

Проиллюстрируем перемножение двух множеств промежуточных вариантов следующим примером. Пусть в качестве аргументов перемножения имеем множество промежуточных вариантов $\Pi_1 = \{a_1b_1c_2, a_1b_2c_1, a_2b_1c_3, a_3b_2c_1\}$ и $\Pi_2 = \{c_1d_2e_1, c_1d_3e_1, c_3d_1e_2, c_3d_3e_3\}$. Тогда построим матрицу синтеза (рис. 4), в которой в строках и столбцах размещены промежуточные варианты множеств Π_1 и Π_2, а на их пересечении проставлены единицы, если синтезируемый вариант реализуем, и нули – в противном случае. Синтезируемый вариант признается нереализуемым, если образующие его варианты содержат различные исходы какого-либо из признаков (в данном случае исходы признака С).

			a_1 b_1 c_2	a_1 b_2 c_1	a_2 b_1 c_3	a_3 b_2 c_1
c_1	d_2	e_1	0	1	0	1
c_1	d_3	e_1	0	1	0	1
c_3	d_1	e_2	0	0	1	0
c_3	d_3	e_3	0	0	1	0

Рис. 4. Матрица синтеза при перемножении множеств промежуточных вариантов

Таким образом, результатом перемножения множеств промежуточных вариантов Π_1 и Π_2 станет множество:

$$\Pi_{12} = \begin{Bmatrix} a_1b_2c_1d_2e_1, a_1b_2c_1d_3e_1, a_3b_2c_1d_2e_1, a_3b_2c_1d_2e_1, \\ a_2b_1c_3d_1e_2, a_2b_1c_3d_3e_3 \end{Bmatrix}.$$

Чтобы предотвратить разрастание множества промежуточных вариантов и возможность в связи с этим возникновения информационного взрыва, определим принцип, в соответствии с которым будет идти процесс синтеза. Предотвратить разрастание можно, если удастся на каждом шаге синтеза максимально сокращать множество промежуточных вариантов. Для этого был разработан следующий алгоритм:

1. Начало.
2. С каждым множеством промежуточных вариантов Π_i соотнесем множество абсолютных ЭЗФ Ψ_i, состоящее из ЭЗФ, которые содержат хотя бы один исход a_i такой, что a_i – исход признака A и $A \in \Omega_i$, где Ω_i – множество признаков, из исходов которых состоят варианты множества Π_i, и хотя бы один исход b_j такой, что b_j – исход признака B и $B \notin \Omega_i$.
3. Для максимального сокращения множества промежуточных вариантов на текущем этапе синтеза из текущего набора множеств промежуточных вариантов $\Pi = \{\Pi_1, \Pi_2, \ldots, \Pi_i\}$ выбирается пара множеств промежуточных вариантов Π_i и Π_j, в результате перемножения которых возможно проверить наибольшее число абсолютных ЭЗФ на получившемся множестве промежуточных вариантов Π_{ij}. Количество таких запрещенных фигур найдем следующим образом: $K_{ij} = |Z_{ij}| = |\Theta_{ij} - \Upsilon_{ij}|$, $\Theta_{ij} = \Psi_i \cap \Psi_j$, где K_{ij} – искомое количество ЭЗФ; Z_{ij} – множество ЭЗФ, доступных для анализа при перемножении множеств промежуточных вариантов Π_i и Π_j, Θ_{ij} – множество ЭЗФ-претендентов на возможность проверки, Ψ_i и Ψ_j – множества ЭЗФ, соотнесенные соответственно с множествами промежуточных вариантов Π_i и Π_j, $\Upsilon_{ij} = \{\upsilon_{ij}\}$, где при $\upsilon_{ij} = \{a_k\}$ $\exists a_k : a_k \notin (V_i \cup V_j)$, где a_k – k-й исход запрещенной фигуры υ_{ij}; V_i, V_j – множества исходов, характеризующие множества промежуточных вариантов Π_i и Π_j соответственно.
4. В результате перемножения пары выбранных множеств промежуточных вариантов Π_i и Π_j получим новое множество промежуточных вариантов Π_{ij}, соотносящееся с множеством абсолютных ЭЗФ $\Psi_{ij} = (\Psi_i \cup \Psi_j) \setminus (\Psi_i \cap \Psi_j)$.
5. Множества промежуточных вариантов Π_i и Π_j удаляются из набора Π, а полученное множество промежуточных вариантов Π_{ij} добавляется в него и наравне с другими множествами промежуточных вариантов участвует в выборе следующей пары перемножаемых множеств.
6. Если в наборе Π остался один элемент, то это – искомый результат перемножения множеств промежуточных вариантов. Иначе к п. 3.
7. Конец.

Для завершения синтеза остается перемножить получившееся множество промежуточных вариантов на признаки, не участвующие в расчетах, но участвующие в ЭЗФ, и на признаки, не участвующие в каких-либо запрещенных фигурах. Для минимизации количества промежуточных вариантов на каждом этапе перемножения следует приблизить к началу перемножения наибольшее количество запрещенных фигур. Таким образом, на каждом этапе перемножения следует выбирать такой признак, чтобы мощность множества общих ЭЗФ текущего множества промежуточных вариантов и перемножаемого признака была наибольшей. Указанное множество определим следующим образом: $Z_A = \Theta_A - \Upsilon_A$, $\Theta_A = \Psi_\Pi \cap \Psi_A$, где Z_A – множество ЭЗФ, доступных для анализа при перемножении множества промежуточных вариантов Π и признака A, Θ_A – множество ЭЗФ-претендентов на возможность проверки, Ψ_Π и Ψ_A – множества ЭЗФ, соотнесенные соответственно множеством промежуточных вариантов Π и признаком A, $\Upsilon_A = \{\upsilon_A\}$, где при $\upsilon_A = \{a_k\}$ $\exists a_k : a_k \notin (V_\Pi \cup V_A)$, где a_k – k-й исход запрещенной фигуры υ_A; V_Π – множество исходов, характеризующее множество промежуточных вариантов Π; V_A – множество исходов признака A.

Затем перемножаем на признаки, не участвующие в каких-либо запрещенных фигурах (перемножать не обязательно, вместо этого итоговые варианты изделий будут содержать исход «любое значение» по данному признаку).

В итоге получаем возможные варианты реализуемых изделий, минимизировав объем вычислений и потребление памяти в процессе синтеза.

Проведенные исследования и полученные результаты позволяют сделать следующие выводы:

– предлагаемый подход к упорядочению признаков в процессе перемножения является эффективным инструментом минимизации ресурсов компьютера при реализации процесса структурного синтеза;

– учет характера запрещенных фигур и их разбиение на подмножества является необходимым условием оптимизации процесса структурного синтеза.

Библиографические ссылки

1. *Малина О. В.* Теория и практика автоматизации структурного синтеза объектов и процессов с использованием методов характеризационного анализа : автореф. дис. … д-ра техн. наук : 05.13.12 / [Ижев. гос. техн. ун-т]. – Ижевск, 2002. – 36 с.

2. *Малина О. В., Уржумов Н. А.* Математическая модель процесса структурного синтеза объектов на дискретных структурах, исключающая порождение запрещенных

вариантов // Информац. математика. – 2005. – № 1. – С. 114–120.

3. *Малина О. В., Уржумов Н. А.* Математическое обеспечение оптимизации процесса структурного синтеза объектов средней степени сложности // Современные технологии : сб. науч. тр., посвящ. 50-летию каф. ТРП. – Ижевск : Изд-во ИжГТУ, 2006. – С. 32–44.

4. Подходы к организации экспертного опроса подсистемы формирования классификатора системы структурного синтеза конечных объектов, построенных на дискретных структурах / О. В. Малина, О. Ф. Валеев, С. А. Морозов и др. // Вестн. Ижев. гос. техн. ун-та. – 2012. – № 1. – С. 126–129.

5. *Малина О. В.* Анализ множества запрещенных фигур структурного синтеза объектов и процессов // Информац. математика. – 2003. – № 1. – С. 138–143.

* * *

O. V. Malina, DSc in Engineering, Professor, Kalashnikov Izhevsk State Technical University
O. F. Valeyev, Post-graduate, Kalashnikov Izhevsk State Technical University

Approaches to minimizing computer resources necessary to implement the process of structural synthesis of objects consisting of discrete structures

Implementation of structural synthesis processes by combinatorial search faces the problem of dimensionality curse which prevents executing the structural synthesis by means of advanced personal computers. The paper proposes the methods of minimizing the computational load and consumed memory when performing the structural synthesis, the methods being based on feature ordering when multiplying.

Keywords: computer-aided design, structural synthesis, combinatorial search

Получено: 03.04.13

УДК 510.62

Д. В. Пархоменко, аспирант
Московский государственный университет имени М. В. Ломоносова

АВТОМАТНЫЕ *P*-ЯЗЫКИ

*Ранее автор ввел понятие **p**-языка, как множества слов, возникающих на выходе некоторого автомата не менее **p** раз. Было показано, что каждый такой язык регулярен, но оставался вопрос, проверяемо ли свойство произвольного регулярного языка быть языком **p**-типа. Эта проблема решена в данной статье. Регулярные языки с частотными свойствами ранее рассматривались в работе* [1].

Ключевые слова: спектральная функция конечных автоматов, мультимножества конечных автоматов, выходные языки конечных автоматов

Основные определения и результаты

Рассмотрит конечные алфавиты A,B: $A = \{a_1, \ldots, a_{|A|}\}$, $B = \{b_1, \ldots, b_{|B|}\}$ и конечный детерминированный инициальный автомат

$$V = (A, Q, B, \varphi, \psi, q_0).$$

Заданный автомат порождает ограниченно детерминированную словарную функцию

$$f_V: A^* \to B^*.$$

Определение 1:

Пусть задан конечный детерминированный инициальный автомат V. Гистограммной автоматной функцией автомата V назовем функцию

$$\boldsymbol{\kappa_V}: B^* \to N \cup \{0\},$$

определенную по правилу:

$$\boldsymbol{\kappa_V(\beta) = |\{\ \alpha \in A^* \mid f_V(\alpha) = \beta\ \}|}.$$

Функция $\boldsymbol{\kappa_V}$ каждому слову выходного алфавита сопоставляет мощность множества его прообразов при отображении f_V. Фактически, если использовать множество с кратностями $f_V(A^*)$, можно каждому слову из B^* поставить в соответствие его кратность во множестве $f_V(A^*)$. Другими словами, $\boldsymbol{\kappa_V}$ – это функция кратности выходных слов автомата V.

Определение 2:

Для конечного детерминированного автомата V и натурального p обозначим

$$\boldsymbol{L_p(V) = \{\ \beta \in B^* \mid \kappa_V(\beta) \geq p\ \}}.$$

$L_p(V)$ – множество выходных слов, которые на выходе автомата V встречаются не менее p раз. В работе [2] установлено, что для любого конечного детерминированного автомата V, $L_p(V)$ – продолжаемый регулярный язык, причем язык либо бесконечный, либо пустой. Далее нам потребуется свойство языков $\boldsymbol{L_p(V)}$, непосредственно вытекающее из определения

$$\boldsymbol{L_1(V) \supseteq L_2(V) \supseteq L_3(V) \supseteq \ldots}.$$

Имеет место

Утверждение 1:

Если автомат $V = (A, Q, B, \varphi, \psi, q_0)$ *таков, что для любого натурального i: множество* $L_i(V)$ *– не пусто, то для всякого n найдется m>n, что* $L_m(V) \neq L_n(V)$.