УДК 004.415:658

В. А. ПОПОВ, М. В. МИЛАНОВ, Ю. В. МАРЧЕНКО

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина

ИССЛЕДОВАНИЕ МОДЕЛЕЙ АЛГОРИТМОВ УЧЕТА ТРАВМАТИЗМА СПОРТСМЕНОВ

Рассматривается актуальная задача моделирования и учета травматизма спортсменов. Анализируются различные схемы представления алгоритмов (граф-схема алгоритмов, матричная схема алгоритмов, автоматная схема алгоритмов, регулярная схема алгоритмов, логическая схема алгоритмов). Для каждой из схем разрабатываются методические рекомендации для перехода к любой другой. Предложенный метод используется для формирования алгоритмической модели учета травмированности спортсменов-культуристов при выборе для них тренировочных упражнений. Модель учета травматизма представляется в различных формах записи.

Ключевые слова: *алгоритмическое моделирование, логическая схема, спортивный травматизм.*

Введение и постановка задачи

В настоящее время наблюдается активное внедрение информационных технологий в практически все сферы человеческой жизни. Автоматизация производственных, обслуживающих и прочих процессов стала вполне обычным явлением. Такие тенденции в развитии автоматизированных систем приводят к их значительному усложнению. Эффективность работы сложной автоматизированной системы, в свою очередь, зависит от заложенной в ее основу алгоритмической базы [1].

Разнообразие задач, решаемых информационными системами и, как следствие, широкий диапазон алгоритмов, применяемых в них, привели к созданию ряда различных представлений для описания алгоритмических конструкций [2–5]. Наиболее актуальны на данный момент граф-схема алгоритмов, матричная схема алгоритмов, автоматная схема алгоритмов, регулярная схема алгоритмов и логическая схема алгоритмов. Такое разнообразие объясняется как тематической спецификой применения тех или иных систем, так и требованиями к простоте и удобству представления информации [6-8]. При этом часто ставится задача отразить алгоритмическую структуру, оформленную одним способом, в иной, более актуальной на данный момент, форме.

Предлагается систематизированное изложение в краткой форме процедур перехода от одной формы представления алгоритмов к другой на основе большого числа выполненных примеров из реальных систем.

Анализ известной литературы по алгебрам алгоритмов показал, что указанная выше задача в целом не раскрыта, поэтому нам представляется актуальной и научно новой разработка всех 20 процедур перехода из одной формы представления алгоритмов в другую [1-8].

Разработанная методика перехода из одной формы представления алгоритмов в другую применяется для создания алгоритмической модели учета травматизма при выборе упражнений спортсменами-культуристами [9].

При написании статьи был проанализирован ряд иностранных литературных источников. Были изучены западные подходы к построению, описанию и оптимизации алгоритмов и структур данных [6-8]. В качестве источника справочной информации по вопросам учета травматизма спортсменов было также использовано пособие [9].

Некоторые преобразования из одной схемы представления алгоритмов в другие изложены в [10]. В данной работе сделаны обобщения для переходов из одной формы в любую другую.

Для реализации поставленной цели необходимо выполнить следующее:

1. Провести анализ литературных источников по формам представления алгоритмов.
2. Выявить общие свойства для всех схем и построить обобщенный алгоритм перехода.
3. Определить рекомендуемый порядок преобразований.
4. Сформировать попарные переходы между всеми формами представления алгоритмов.
5. Применить полученный метод для конкретного алгоритма.

1. Анализ используемых схем алгоритмов

Граф-схема алгоритма [5] (ГСА) – конечный связный ориентированный граф, вершины которого соответствуют операторам, а дуги задают порядок следования операторов алгоритма. Основные виды вершин – операторные и условные. Операторные вершины обозначают выполнение какого-либо процесса. Условные служат для отображения ветвления алгоритма.

Матричной схемой алгоритма [4] (МСА) называется квадратная матрица, строки которой соответствуют исходным операторам $Y_0..Y_n$, а столбцы – операторам перехода $Y_1..Y_к$. В МСА пересечение строки y_i и столбца y_j записывается логическое условие (конъюнкция переменных X_i), соответствующее условию перехода от оператора y_i к оператору y_j. Если переход от некоторого оператора к другому отсутствует, то соответствующая клетка в МСА остается пустой. Если существует несколько альтернативных ветвей, по которым возможен переход из оператора Y_i в оператор Y_j, в соответствующую ячейку МСА заносится дизъюнкция их конъюнкций.

Логическая схема алгоритма [3] (ЛСА) представляет собой конъюнктивную форму записи алгоритма с указанием в ней условных и безусловных переходов.

Автоматная схема алгоритмов [3] (АСА, автомат Мура) – в теории вычислений – конечный автомат, выходное значение сигнала в котором зависит лишь от текущего состоянии данного автомата, и не зависит напрямую от входных значений.

Регулярная схема алгоритма [3] (РСА) представляет собой конъюнктивную форму записи алгоритма как описание собой совокупности всех ветвей переходов.

Рассмотрим переходы от одной схемы представления алгоритмов к другой.

1.1. Обоснование обобщенного метода и последовательности переходов между схемами представления алгоритмов

Нетрудно вычислить общее количество переходов между формами представления алгоритмов по известной формуле числа сочетаний из заданного множества по несколько элементов:

$$C_m^n = \frac{m!}{n!*(m-n)!},$$

где m – число исходных элементов заданного множества, равное 5;

n – количество элементов в комбинации, равное 2.

$$C_5^2 = \frac{5!}{2!*(5-2)!} = \frac{5!}{2!*3!} = \frac{4*5}{2} = \frac{20}{2} = 10.$$

Для учета обратных переходов нужно умножить полученный результат на 2. Получим общее количество переходов между формами представления алгоритмов равное 20.

Наиболее удобным представляется первоочередное описание взаимных переходов между ГСА, МСА и ЛСА ввиду их смысловой схожести и простоты формирования. Затем перейдем от каждой из этих схем к АСА, после чего – к РСА (рис. 1).

	ГСА	МСА	ЛСА	АСА	РСА
ГСА	-	1	3	7	15
МСА	2	-	5	9	17
ЛСА	4	6	-	11	19
АСА	8	10	12	-	13
РСА	16	18	20	14	-

Рис.1. Предлагаемая последовательность переходов между схемами алгоритмов.

В итоге становится возможным отображение одного и того же алгоритма в формах, применимых при реализации как различных программных, так и аппаратных систем.

В основе предлагаемого метода для перехода между формами представления алгоритмов лежит учет общих свойств всех схем, их анализа и выбора наиболее подходящей для решаемой задачи. После этого учитываются конкретные особенности выбранной и исходной формы представления и выполняется преобразование.

Таким образом, обобщенный метод перехода между двумя схемами алгоритмов можно представить следующим образом:

1. Определяется исходная форма представления алгоритма.
2. Анализируется специфика системы, в рамках которой будет применяться алгоритм.
3. Выбирается целевая форма представления алгоритма.
4. Определяются общие структурные элементы в исходной и целевой схемах алгоритмов.
5. Выявляются специфические особенности исходной и целевой схем алгоритмов.
6. Выполняются последовательные преобразования с целью представления алгоритма в целевой форме.

1.2. Формализация переходов между различными схемами алгоритмов

Граф-схема – матричная схема (переход 1)

1. В первую очередь именуются столбцы таблицы названиями вершин граф-схемы, в которые есть переходы.
2. Строки именуются названиями вершин, из которых есть переходы.
3. Последовательно построчно таблица заполняется значениями условий, необходимых для осуществления перехода:
   - в случае, если переход безусловный, в ячейку заносится 1;
   - в случае, если переход между 2 состояниями осуществляется по условию/ряду условий, значением ячейки будет дизъюнкция возможных ветвей, состоящая из конъюнктов операторов решения;
   - в случае, если переход между 2 активностями невозможен, ячейка оставляется пустой.

Матричная схема – граф-схема (переход 2)

1. Наименование 1 строки матрицы переходов переносится на схему как первый элемент.
2. В этой строке производится поиск элемента, для которого указано значение для перехода.
3. На граф-схему добавляется конечная для данного перехода вершина.
4. В случае, если значение в ячейке – 1, элементы соединяются стрелкой напрямую.
5. Если значением является логическая функция, то для каждого конъюнкта на схему наносится элемент условного перехода, и элементы соединяются по указанным ветвям (если конъюнкт указан без отрицания – по ветке 1, если со знаком – 0).
6. Действия, описанные в пунктах 2-5, последовательно выполняются для всех пар в матрице.

Граф-схема – логическая схема (переход 3)

1. Из граф-схемы последовательно выписываются все элементы по мере их появления при линейном прохождении алгоритма.
2. Прохождение ветвления по позитивному сценарию записывается как последовательность операций.
3. Для указания условного перехода по негативному сценарию после оператора ставится стрелка вверх, нумеруемая от 1 до n, в целевом месте перехода ставится стрелка вниз, нумеруемая тем же значением.

Логическая схема – граф-схема (переход 4)

1. На граф схему последовательно наносятся все элементы, указанные в логической схеме.
2. Последовательно расположенные в логической схеме элементы соединяются друг с другом на граф-схеме.
3. Для каждого перехода по негативному сценарию в логической схеме (для пары разнонаправленных одинаково нумерованных стрелок) строится соответственная связь на граф-схеме.

Матричная схема – логическая схема (переход 5)

1. Из матричной схемы последовательно выписываются все элементы $Y_1..Y_n$, при этом между ними оставляются достаточные пустые промежутки, если переход в элемент не является безусловным;
2. Построчно перебираются ячейки, в результате чего логическая схема заполняется операторами $X_1..X_n$.
3. Если оператор x_i был встречен впервые, он записывается в логическую схему между парой операторов Y_j и Y_k, в ячейке с переходом между которыми он расположен.
4. Если оператор X_i встречен повторно, при этом с другим знаком, после него на логической схеме ставится нумерованная стрелка вверх, а перед оператором, в который должен осуществиться переход – стрелка вниз с таким же индексом.

Логическая схема – матричная схема (переход 6)

1. Столбцы и строки матричной схемы именуются названиями элементов $Y_1..Y_n$, встречающихся в логической схеме.
2. Выполняют последовательное заполнение ячеек матричной схемы значениями условий переходов между ними.
3. Если между парой операторов Y_i и Y_j записаны некоторые операторы условного перехода, они последовательно записываются в соответствующую ячейку матрицы, соединяемые знаком конъюнкции.
4. Если описано несколько альтернативных вариантов перехода из одного оператора в другой, все эти варианты записываются в соответствующую ячейку матрицы, разделяемые знаком дизъюнкции.

Граф-схема – автоматная схема (переход 7)

1. На граф-схеме каждый переход со стороны конечного элемента помечается как $A_1..A_n$.

2. Последовательно на автоматной схеме отображаются элементы $A_1..A_n$.

3. Элементы схемы соединяются связями, при этом:

– если пара переходов на граф-схеме соответственно входят и выходят в и из элемента процесса, то на автоматной схеме они соединяются стрелкой, подписываемой названием этого узла;

– если пара переходов на граф-схеме соответственно входят и выходят в и из элемента решения, то на автоматной схеме они соединяются стрелкой, подписываемой конъюнкцией позитивного или отрицательного значения этого условия с пустым подмножеством e.

Автоматная схема – граф-схема (переход 8)

1. На граф-схему наносятся все элементы $Y_{1..n}$ и $X_{1..n}$ в виде элементов процессов и решений соответственно.

2. Далее выполняется попарное соединение элементов.

– элемент, связь с чьим названием входит в узел автоматной схемы, соединяется с элементом, чье название обозначает выходной для данного узла переход;

– если начальный элемент-ветвление, то переход обозначается как 1 или 0 в зависимости от отсутствия или наличия отрицания данного элемента на автоматной схеме;

3. На граф-схему наносятся начальный элемент Y_0 и конечный элемент $Y_к$, соединяемые соответственно с Y_1 и Y_n.

Матричная схема – автоматная схема (переход 9)

1. На автоматную схему наносятся элементы $A_0..A_n$, где n-сумма элементов Y и X в матрице переходов.

2. Состояния автомата последовательно соединяются таким образом:

– безусловному переходу соответствует 1 связь, именуемая названием конечного процесса;

– для каждого элемента $X_1..X_n$ строится 2 связи с различными состояниями автомата, при этом для этого состояния исходящая связь будет помечена как еще не отмеченный конъюнкт из матрицы переходов или конечный процесс.

Автоматная схема – матричная схема (переход 10)

1. Сперва заполняются переходы к первому и конечному процессам, для этого определяется начальное состояние автомата, значение исходящего из него перехода – название процесса, в который безусловно осуществляется переход из процесса Y_0, аналогично название перехода в конечное состояние автомата – название процесса, из которого осуществляется безусловный переход к процессу Y_k.

2. Далее для того, чтобы определить условия переходов для матричной схемы последовательно рассматривают пары процессов и определяют промежуточные для них условные переходы, при этом обозначение пустого множества e опускают.

3. Если для перехода к какому-либо состоянию из другого необходимо выполнить несколько условных переходов, в соответствующую ячейку матрицы заносится их конъюнкция.

4. В случае, если из одного процесса в другой можно попасть несколькими различными способами, результирующим значением становится дизъюнкция ветвей.

Логическая схема – автоматная схема (переход 11)

1. На автоматную схему последовательно наносятся элементы, указанные в логической схеме.

2. Последовательно расположенные в логической схеме элементы соединяются друг с другом на автоматной схеме.

3. Для каждого перехода по негативному сценарию в логической схеме (для пары разнонаправленных одинаково нумерованных стрелок) строится соответственная связь на автоматной схеме.

Автоматная схема – логическая схема (переход 12)

1. Из автоматной схемы последовательно выписываются все элементы по мере их появления при линейном прохождении алгоритма.

2. Прохождение ветвления по позитивному сценарию записывается как последовательность операций.

3. Для указания условного перехода по негативному сценарию после оператора ставится стрелка вверх, нумеруемая от 1 до n, в целевом месте перехода ставится стрелка вниз, нумеруемая тем же значением.

Автоматная схема – регулярная схема (переход 13)

1. Для каждого условного перехода записывают функцию R_k, представляющую собой дизъюнкцию значений каждой его ветви, при этом, если ветвь пустая, она заменяется знаком пустого множества e, если же она содержит в себе другой услов-

ный переход, его первично указывают соответствующей функцией R_k.

2. Затем производят подстановку простых функций в сложные, тем самым заменяя наименования этих функций на соответствующие им дизъюнкты.

3. Записывают регулярное выражение как конъюнкцию всех безусловных переходов, обозначаемых названием процесса, и сложных функций, покрывающих все варианты перехода.

Регулярная схема – автоматная схема (переход 14)

Автоматная схема строится на основе регулярной путем последовательного восстановления ветвей переходов при поочередном переборе конъюнктов регулярного выражения:

1. Если конъюнкт простой, из состояния автомата проводится только 1 связь, обозначаемая названием соответствующего процесса.

2. Если конъюнкт представлен дизъюнкцией 2 элементов, из состояния автомата проводят 2 связи, представляющие собой 2 дальнейшие ветви переходов, при этом каждая ветвь может быть представлена в свою очередь дизъюнкцией, что приводит к дальнейшему ветвлению.

Граф-схема – регулярная схема (переход 15)

Регулярное выражение представляет собой совокупность конъюнктов:

1. Если переход в элемент безусловный, конъюнкт представляет собой наименование исходного узла.

2. Конъюнкт ветвления обозначается как дизъюнкция двух его ветвей, при этом, если какая-либо из ветвей не содержит элементов, она обозначается пустым множеством е; дизъюнктом также может быть и другой составной элемент.

3. Скобки, ограничивающие дизъюнкт, помечаются названием условного оператора.

Регулярная схема – граф-схема (переход 16)

1. Граф-схема строится на основе регулярной схемы путем последовательного отображения конъюнктов.

2. Простой конъюнкт переносится на граф-схему в виде блока деятельности.

3. Составной конъюнкт отображается на граф-схеме как элемент решения, при этом дальнейшее его описание элементами граф-схемы происходит уже по 2 ветвям, описанным дизъюнктами в регулярной схеме.

4. Ветвь, помеченная пустым множеством е, не содержит в себе никаких элементов, и означает прямой переход к следующему конъюнкту.

Матричная схема – регулярная схема (переход 17)

Регулярное выражение представляет собой совокупность конъюнктов:

1. Если переход в элемент безусловный, конъюнкт представляет собой наименование исходного узла.

2. Элемент условного перехода $X_1..X_n$ обозначается как дизъюнкция двух его ветвей, при этом, если какая-либо из ветвей не содержит элементов, она обозначается пустым множеством е; дизъюнктом также может быть и другой составной элемент.

3. Скобки, ограничивающие дизъюнкт, помечаются названием условного оператора.

Регулярная схема – матричная схема (переход 18)

1. Матричная схема заполняется на основе регулярной схемы путем последовательного отображения конъюнктов.

2. Простой конъюнкт переносится в матрицу как безусловный переход, обозначаемый 1.

3. Составной конъюнкт заносится в матрицу переходов как элемент решения, описываемый конъюнкцией условных переходов, разделяющих эти 2 процесса, при этом переход к процессу, являющемуся частью составного дизъюнкта по отношению к текущему элементу, не возможен.

Логическая схема – регулярная схема (переход 19)

1. Элементы логической схемы последовательно переносятся в качестве конъюнктов в регулярную схему;

2. Если после элемента логической схемы появляется оператор условного перехода (нумерованная стрелка вверх), вся ветвь, следующая за этим оператором до соответствующего оператора вхождения (нумерованная стрелка вниз) записывается как дизъюнкция составляющих ее элементов с оператором пустого множества е.

Регулярная схема – логическая схема (переход 20)

1. Элементы регулярной схемы последовательно переносятся в качестве конъюнктов в логическую схему;

2. Если встречается составной конъюнкт, представляющий собой дизъюнкцию операторов с

пустым множеством е, на логической схеме вся эта ветвь обособляется операторами условного перехода и вхождения (нумерованные стрелки, направленные вверх и вниз соответственно).

2. Разработка алгоритмической модели для учета травмированности спортсменов

Применение вышеописанных способов представления алгоритмов рассмотрим на примере учета травматизма спортсменов-культуристов при выборе для них тренировочных упражнений. Для этого введем следующие обозначения:

– k_1– поправочный коэффициент для рабочего веса при травме колен [9];

– k_2 – поправочный коэффициент для рабочего веса при травме плечевого пояса [9];

– Y_0 – начало;

– Y_1 – ввод исходных данных;

– Y_2 – форма упражнения=”тренажер”;

– Y_3 – рабочий вес*k_1;

– Y_4 – тип упражнения=”изоляция”;

– Y_5 – рабочий вес*k_2;

– Y_6 – найти упражнение;

– $Y_к$ – конец;

– X_1 – мышечная группа=”ноги”?;

– X_2 – травма поясницы?;

– X_3 – травма колен?;

– X_4 – мышечная группа=”грудь” или “плечи”?;

– X_5 – травма плечевого пояса?;

– X_6 – тип упражнения=”базовое”?;

– X_7 – травма локтя?.

Используя данные обозначения, построим граф-схему алгоритма (рис. 2).

Матричную схему преобразуем в логическую (рис. 3) (переход 5):

$$L = Y_0 \wedge Y_1 \wedge X_1 \uparrow^1 \wedge X_2 \uparrow^2 \wedge Y_2 \downarrow^2 \wedge$$
$$\wedge X_3 \uparrow^1 \wedge Y_3 \downarrow^1 \wedge X_4 \uparrow^3 \wedge \neg X_5 \uparrow^4 \wedge$$
$$\wedge X_6 \uparrow^3 \wedge X_7 \uparrow^1 \wedge Y_5 \wedge$$
$$\wedge \neg Y_6 \uparrow^3 \downarrow^4 \wedge Y_4 \downarrow^3 \wedge Y_к.$$

Затем представим рассматриваемый алгоритм в автоматной схеме (рис. 4) (переход 11). Введем следующие обозначения для состояний автомата:

– A_0 – ожидание ввода исходных данных;

– A_1 – определение мышечной группы №1;

– A_2 – определение травмы №1;

– A_3 – модификация параметра «форма упражнения»;

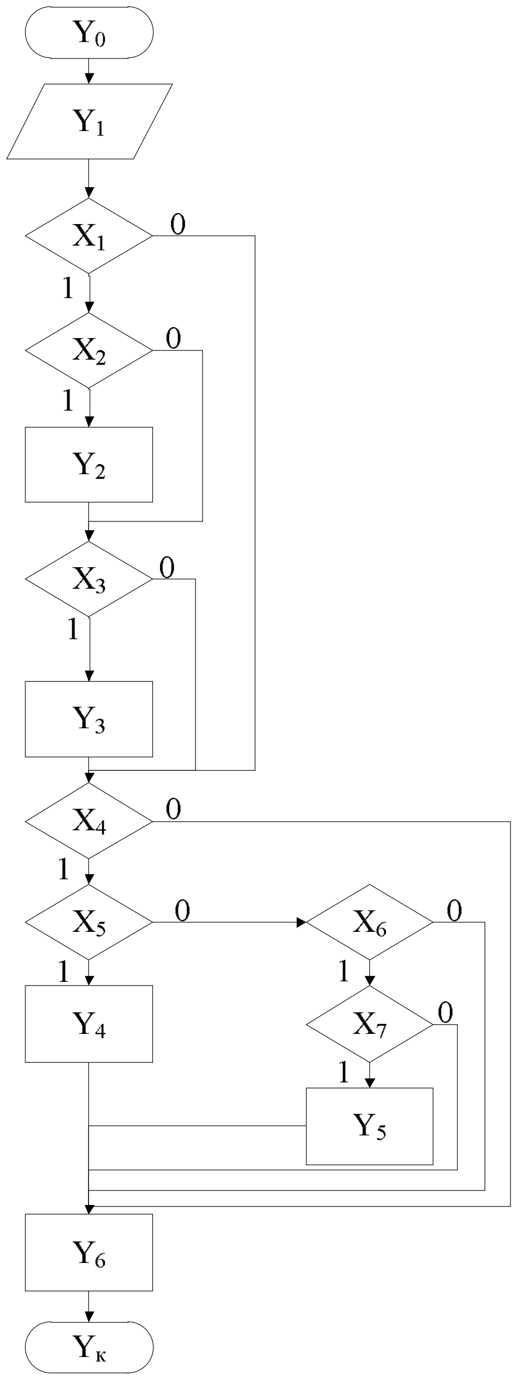

Рис. 2. Граф-схема алгоритма

– A_4 – определение травмы №2;

– A_5 – модификация параметра «рабочий вес» №1;

– A_6 – определение мышечной группы №2;

– A_7 – определение травмы №3;

– A_8 – модификация параметра «тип упражнения»;

– A_9 – формирование поискового запроса;

– A_{10} – конечное состояние;

– A_{11} – определение типа упражнения;

– A_{12} – определение травмы №4;

– A_{13} – модификация параметра «рабочий вес» №2.

Из ГСА перейдем к матричной схеме (рис. 3) (переход 1).

	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	$Y_к$
Y_0	1						
Y_1		$X_1 \wedge X_2$	$X_1 \wedge \neg X_2 \wedge X_3$	$(X_1 \wedge \neg X_2 \wedge \neg X_3 \vee \neg X_1) \wedge X_4 \wedge X_5$	$(X_1 \wedge \neg X_2 \wedge \neg X_3 \vee \neg X_1) \wedge X_4 \wedge \neg X_5 \wedge X_6 \wedge X_7$	$(X_1 \wedge \neg X_2 \wedge \neg X_3 \vee \neg X_1) \wedge (\neg X_4 \vee X_4 \wedge \neg X_5 \wedge (\neg X_6 \vee X_6 \wedge \neg X_7))$	
Y_2			X_3	$\neg X_3 \wedge X_4 \wedge X_5$	$X_4 \wedge \neg X_5 \wedge X_6 \wedge X_7$	$\neg X_3 \wedge (\neg X_4 \vee (X_4 \wedge \neg X_5 \wedge (\neg X_6 \vee X_6 \wedge \neg X_7)))$	
Y_3				$X_4 \wedge X_5$	$X_4 \wedge \neg X_5 \wedge X_6 \wedge X_7$	$\neg X_4 \vee X_4 \wedge \neg X_5 \wedge (\neg X_6 \vee X_6 \wedge \neg X_7)$	
Y_4						1	
Y_5						1	
Y_6							1

Рис. 3. Матричная схема алгоритма

Пустые ветви алгоритма обозначим с помощью оператора e.

В форме регулярной схемы данный алгоритм будет выглядеть следующим образом (переход 13):

$$R = Y_0 \wedge Y_1 \wedge X_1(e \vee X_2 \wedge (Y_2 \vee e) \wedge X_3 \wedge (Y_3 \vee e)) \wedge$$
$$\wedge X_4 \wedge (e \vee X_5 \wedge (Y_4 \vee X_6 \wedge (X_7 \wedge (Y_5 \vee e) \vee e)))) \wedge$$
$$\wedge Y_6 \wedge Y_к.$$

Полученные алгоритмические модели могут применяться как основа рекомендательных информационных систем, в частности для автоматизированной системы информационной поддержки силового тренинга. Для этой цели наиболее подходят граф-схема и матричная схема алгоритма.

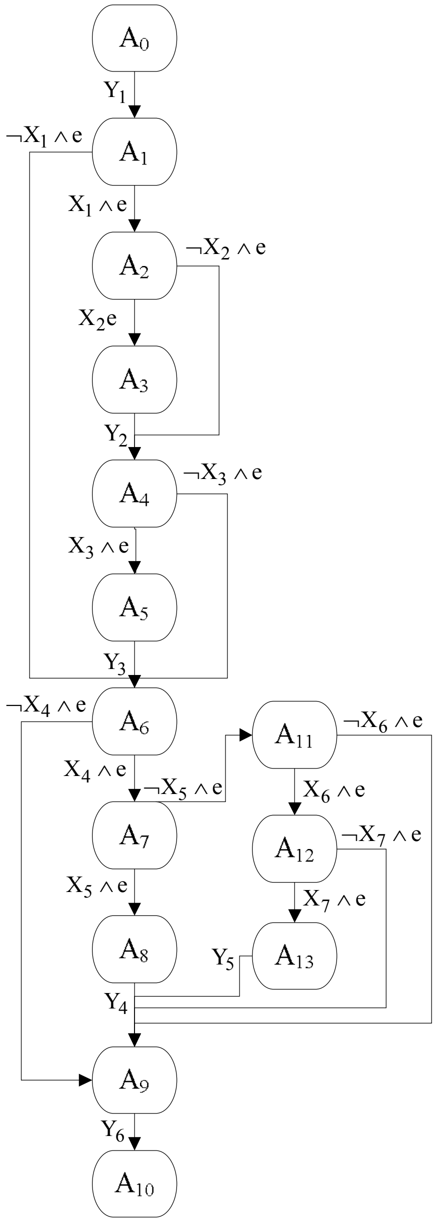

Рис. 4. Автоматная схема алгоритма

Заключение

Проанализированы различия форм представления алгоритмов. Сформирован метод для перехода от одной формы представления алгоритмов к другой. В основе каждого перехода лежит использование общих свойств процедуры и затем учет конкретных свойств для каждой пары форм представления. Алгоритмические модели были применены для формирования алгоритма учета травмированности спортсменов-культуристов при выборе для них тренировочных упражнений, что популизирует силовые виды спорта. При этом учитывается состояние здоровья, и восстанавливаются тренировочные показатели после перенесенной травмы.

Литература

1. Галиев, Ш. И. Математическая логика и теория алгоритмов [Текст] / Ш. И. Галиев. – Казань : Издательство КГТУ им. А.Н. Туполева, 2002. – 270 с.

2. Акуловский, В. Г. Реализация комплексного подхода к описанию алгоритмов информационно-управляющих систем в рамках алгебраического аппарата [Текст] / В. Г. Акуловский, А. Е. Дорошенко // Управляющие системы и машины. – 2013. – № 5. – С. 46-52.

3. Глушков, В. М. Теория автоматов и формальное преобразование микропрограмм [Текст] / В. М. Глушков // Кибернетика. – 1965. – № 5. – С. 1–10.

4. *Куц, А. К. Математическая логика и теория алгоритмов [Текст] / А. К. Куц. – Омск : Издательство Наследие, 2003. –108 с.*

5. *Цейтлин, Г. Е. Введение в алгоритмику. [Текст] / Г. Е. Цейтлин. – Киев : «Сфера», 1998. – 310 с.*

6. *Wirth, N. Algorithms and data structures [Текст] / N. Wirth. – United States : Prentice Hall, 2004.– 288 p.*

7. *Aho, A. V. Data Structures and Algorithms [Текст] / A. V. Aho, J. D. Ullman, J. E. Hopcroft. – United States : Addison-Wesley, 1983. –620 с.*

8. *Introduction to Algorithms [Текст] / T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. – Cambridge : MIT Press, 2009. – 1296 с.*

9. *Delavier, F. Strength Training Anatomy [Текст] : / F. Delavier. – United States : Human Kinetics, 2010. – 144 p.*

10. *Дискретные устройства автоматизированных систем управления [Текст] : учеб. для вузов / А. Е. Амбросов, А. П. Плахтеев, Г. Н. Тимонькин, М. П. Ткачев, В. С. Харченко ; под ред. Г. Н. Тимонькина, В. С. Харченко. – ХВКИУ, 1990. – 512 с.*

Поступила в редакцию 11.06.2015, рассмотрена на редколлегии 18.11.2015

ДОСЛІДЖЕННЯ МОДЕЛЕЙ АЛГОРИТМІВ ОБЛІКУ ТРАВМАТИЗМУ СПОРТСМЕНІВ

В. О. Попов, М. В. Міланов, Ю. В. Марченко

Розглядається актуальна задача моделювання та обліку травматизму спортсменів. Аналізуються різні схеми представлення алгоритмів (граф-схема алгоритмів, матрична схема алгоритмів, автоматна схема алгоритмів, регулярна схема алгоритмів, логічна схема алгоритмів). Для кожної з схем розробляються методичні рекомендації для переходу до будь-якої іншої. Запропонований метод використовується для формування алгоритмічної моделі обліку травмованості спортсменів-культуристів при виборі для них тренувальних вправ. Модель обліку травматизму представляється в різних формах запису.

Ключові слова: алгоритмічне моделювання, логічна схема, спортивний травматизм.

RESEARCH OF MODELS OF THE ATHLETES TRAUMATISM ACCOUNTING ALGORITHMS

V. A. Popov, M. V. Milanov, Y. V. Marchenko

The actual problem of modeling and the accounting of traumatism of athletes is considered. Various schemes of representation of algorithms (a flowgraph of algorithms, the matrix scheme of algorithms, the automatic scheme of algorithms, the regular scheme of algorithms, the logical scheme of algorithms) are analyzed. Methodical recommendations for transition to any other are developed for each of schemes. The offered method is used for formation of algorithmic model of the accounting of injuries of athletes-bodybuilders at a choice for them training exercises. The model of the accounting of injuries is represented in various forms of record.

Key words: algorithmic modeling, logical scheme, sports injuries.

Попов Вячеслав Алексеевич – канд. техн. наук, профессор, профессор кафедры «Информационные управляющие системы», Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков, Украина.

Миланов Михаил Владимирович – канд. техн. наук, доцент, доцент кафедры «Информационные управляющие системы», Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков, Украина.

Марченко Юрий Вячеславович – магистрант кафедры «Информационные управляющие системы», Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков, Украина.