

# СРЕДСТВА ДЕДУКТИВНОГО СИНТЕЗА ПРЕОБРАЗОВАТЕЛЕЙ ДАННЫХ ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ В ИССЛЕДОВАНИЯХ ЭНЕРГЕТИКИ

О.В. Курганская

*Институт систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук*

**Введение.** В системных исследованиях топливно-энергетического комплекса и отраслевых систем энергетики математическое моделирование и вычислительный эксперимент являются важнейшими инструментами [1-3]. При этом вычислительные эксперименты в исследованиях системных исследованиях топливно-энергетического комплекса и систем энергетики имеют ряд особенностей. В частности, имеют место комплексные вычислительные эксперименты. В ходе комплексных экспериментов расчеты осуществляются последовательно с использованием нескольких математических моделей. Целью таких экспериментов могут быть исследования на разных уровнях – на уровне топливно-энергетического комплекса в целом или отдельных отраслевых систем энергетики с использованием соответствующих моделей, или исследования, сочетающие качественные и количественные методы (методы ситуационного анализа и традиционные методы математического моделирования).

Другой важной особенностью вычислительных экспериментов в таких исследованиях является использование внешних программных средств для расчета математических моделей и содержательной интерпретации результатов расчетов. В первую очередь, к таким средствам следует отнести оптимизаторы `lp_solve` или `MINOS` для расчета математических моделей и табличные процессоры `Microsoft Excel` или `OpenOffice Calc` [2, 3] для анализа полученных результатов. Кроме того, довольно широкое распространение имеют специализированные программные средства для качественного анализа исследуемых проблем, например, средства интеллектуальной ИТ-среды [4].

Кроме того, следует отметить, что преобразования данных, не обеспеченные уже имеющимися внешними программами и инструментами осуществляются по очень похожим схемам и строятся из похожих друг на друга элементов. К таким преобразованиям в первую очередь следует отнести процедуры построения файлов в формате внешнего программного средства, содержащих математическую модель или результаты эксперимента для их содержательной интерпретации. Как правило, такие преобразователи просты и не содержат условий, циклов или рекурсий.

Таким образом, несомненна возможность создания такого подхода к разработке программного обеспечения для поддержки вычислительного эксперимента в исследованиях энергетики, который бы позволял автоматизировать проведение сложных вычислительных экспериментов с помощью средств искусственного интеллекта. В [5] предлагается использовать методы дедуктивного синтеза программ для построения преобразователей данных в ходе вычислительного эксперимента. Преобразователи данных строятся обработкой декларативных представлений соответствующих процессов преобразования данных. Формально декларативные представления представляют собой формулы узкого исчисления предикатов, которые могут быть доказуемы (выводимы) или недоказуемы (невыводимы). Содержательно декларативные представления процессов преобразования данных – это правила преобразования данных, требования к структуре, формату и значениям преобразуемых данных. Дедуктивный синтез преобразователей данных, соответствующих заданным требованиям, заключается в автоматическом доказательстве декларативных представлений процессов преобразования данных. В процессе доказательства автоматически синтезируются сценарии преобразования данных или формирования протоколов диагностических сообщений.

В настоящем докладе представлен подход к построению преобразователей данных для обеспечения вычислительного эксперимента на основе декларативных представлений процессов преобразования данных. В первом параграфе вводится язык узкого исчисления предикатов для формализации декларативных представлений процессов преобразования данных и обосновывается использование XML для представления исходных данных. Во втором параграфе приводится структура и алгоритм построения декларативного представления процесса преобразования данных для вычислительного эксперимента. В третьем параграфе определяется способ применения декларативных представлений, который заключается в автоматическом доказательстве декларативного представления (теоремы существования преобразователя) и синтезе требуемого преобразователя в случае его существования.

**1. Исходные данные и язык для построения декларативных представлений.** Традиционно формат представления исходных данных для проведения вычислительных экспериментов в исследованиях топливно-энергетического комплекса и отраслевых систем энергетики существенно зависит от характера и особенностей проводимых исследований. Одним из самых распространенных форматов являются различные базы данных и электронные таблицы. Для унификации исходных и хранения промежуточных данных для вычислительного

эксперимента используем XML [6]. Выбор XML продиктован его широкой распространенностью и универсальностью. В частности, исходные данные для проведения вычислительного эксперимента предлагается представлять в виде XML-прототипа - предметно-ориентированного описания объекта исследования в формате XML. XML-прототип должен содержать все параметры объекта исследования, необходимые для проведения вычислительного эксперимента, т.е. описание всех данных, применяемых в ходе такого вычислительного эксперимента. Важнейшей особенностью XML-прототипа является его предметно-ориентированная структура, независимая от характера математической модели. XML-прототип может быть как целиком сгенерирован из различных источников данных, например, из баз данных, электронных таблиц, онтологий, так и представлять собой ядро, связывающее данные из различных источников. Использование XML-прототипа в качестве источника данных для формирования файлов, обрабатываемых внешними программными средствами (решателями или средствами содержательной интерпретации), позволяет унифицировать представление данных и при необходимости многократно использовать данные для решения различных задач.

Для задания требований к структуре используемых XML-документов, их элементам, содержанию и атрибутам предлагается использовать специальные XML-документы, называемые в дальнейшем XML-схемами. С помощью XML-схем, в частности, задаются требования к структуре и содержанию файлов, содержащих результаты преобразований (файлов математической модели и средств содержательной интерпретации).

При обработке схем и XML-документов для определения отношений смежности между их элементами воспользуемся элементами интерфейса DOM (Document Object Model) [6]. Примеры отношений и их обозначения проиллюстрированы на рис. 1.

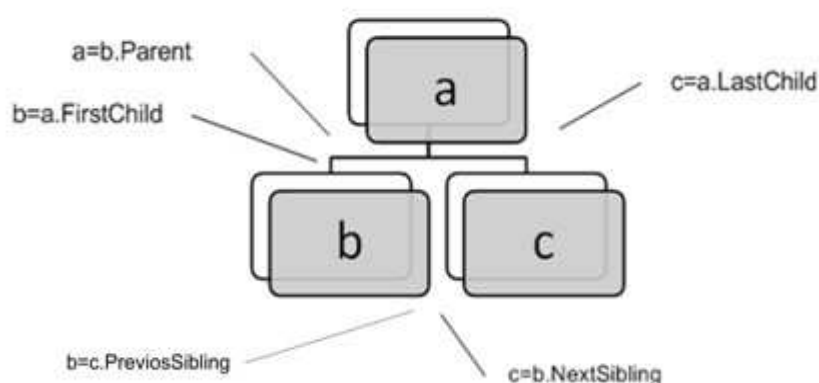


Рис. 1. Отношения элементов в Document Object Model.

Для построения декларативных представлений используем язык узкого исчисления предикатов [9]. Основными элементами языка являются предметные переменные ( $x, y, z, \dots$ ), константы (*element*, *object*, *resource*, ...) для обозначения элементов XML-документов и схем, типов значений атрибутов и содержимого этих элементов, атомы (см. таблица 1), логические связки  $\wedge, \vee, \Rightarrow$  и кванторы  $\forall, \exists$ .

Декларативное представление процесса преобразования данных представляет собой формулу узкого исчисления предикатов. Эта формула содержит требования к искомому преобразованию данных. Она может доказуемой (выводимой) или недоказуемой (невыводимой). Недоказуемость формулы означает, что заданные требования противоречат друг другу, и поэтому преобразователь не может быть построен. В случае выводимости требуемый преобразователь существует и строится в процессе доказательства формулы.

Таблица 1. Значение атомов, используемых в декларативных представлениях.

Атом	Значение
$Vd(y,x)$	элемент XML-документа $x$ предшествует элементу $y$ , $x$ и $y$ - смежные по вертикали.
$H(x,y)$	элемент XML-документа $x$ предшествует элементу $y$ , $x$ и $y$ - смежные горизонтали.
$Vp(y,x)$	элемент XML-документа $y$ следует за элементом $x$ , $x$ и $y$ - смежные по вертикали.
$Q(x,y)$	существует элемент сценария преобразования элемента XML-документа $x$ в элемент $y$ .
$Z(x)$	для элемента XML-документа $x$ не задан никакой элемент сценария
$VVd(x,y)$	элемент схемы XML-документа $x$ предшествует элементу $y$ , и $x$ и $y$ - смежные по вертикали.
$VH(y,x)$	элемент схемы XML-документа $x$ предшествует элементу $y$ , и $x$ и $y$ - смежные по горизонтали
$VVp(y,x)$	элемент схемы XML-документа $x$ предшествует элементу $y$ , и $x$ и $y$ - смежные по вертикали.
$E_i(x)$	элемент $x$ имеет тип $E_i$ .
$P(x), R(x), V(x)$	элемент $x$ доступен для обработки, находится в обработке, успешно обработан соответственно.

**2. Структура и алгоритм построения декларативного представления процесса преобразования данных.** Будем строить декларативное представление процедуры преобразования XML-документа  $D$  со схемой  $S_D$  в документ, удовлетворяющий условиям, заданным в схеме  $S_M$ . Такое декларативное представление будем в дальнейшем называть теоремой синтеза.

Принципиальной особенностью теоремы синтеза является использование нелогических элементов – шаблонов команд запуска элементов сценария преобразования данных  $T$ . Элементы сценария преобразования представляют собой подпрограммы, задающие элемент преобразования данных, содержащихся в документе  $D$ , в XML-файл или текстовый файл, удовлетворяющий всем требованиям, заданным в  $S_M$ .

Теорема синтеза  $G(S_D, D, T)$  имеет следующий вид:

$$((\bigwedge_i S_{D_i}) \wedge (\bigwedge_j D_j)) \rightarrow (B_0 \rightarrow V(a))$$

где

- формулы  $S_{D_i}$  строятся при обходе схемы  $S_D$ ;
- формулы  $D_j$  строятся при обходе XML-прототипа  $D$ ;
- $B_0 = (ND \wedge P(a))$ , причем формула  $ND$  описывает элементы документа  $D$  и связи между ними, а также связи между элементами документа  $D$  и схемы  $S_M$ .

Алгоритм построения теоремы синтеза  $G(S_D, D, T)$  состоит из трех этапов. На первом этапе в ходе обработки схемы XML-документа  $S_D$  строятся формулы  $S_{D_i}$ ; на втором и третьем этапах в ходе обработки XML-документа  $D$  строятся формулы  $D_j$  и следствие теоремы  $B_0 = (ND \wedge P(a))$  соответственно.

Этап 1. Обработка схемы прототипа, построение формул  $S_{D_i}$ .

1. Для всякого элемента  $x \in S_D$  положим  $S_i = \forall x : x \in D (\exists y : y \in S (E_y(x) \wedge P(x) \rightarrow R(x)))$ .
2. Для всякой пары элементов  $x, y \in S_D$  таких, что  $y = x.FirstChild$  положим  $S_i = \forall z, w, z, w \in D : (\exists x, y : x, y \in S : (Vd(z, w) \wedge E_y(w) \wedge E_x(z) \wedge R(z) \rightarrow Vvd(x, y) \wedge R(z)))$
3. Для всякой пары элементов  $x, y \in S_D$  таких, что  $y = x.LastChild$  положим:  $S_i = \forall z, w : z \in D, w \in D : (\exists x, y : x, y \in S (R(z) \wedge E_y(z) \wedge E_x(w) \wedge Vp(w, z) \rightarrow Vvp(w, z) \wedge R(z)))$
4. Для всякой пары элементов  $x, y \in S_D$  таких, что  $y = x.PreviousSibling$  положим  $S_i = \forall z, w : z \in D, w \in D : (\exists x, y : x, y \in S (H(z, w) \wedge E_x(z) \wedge E_y(w) \rightarrow VH(z, w)))$ .

Этап 2. Построение элементов теоремы синтеза  $D_j$ :

1. Для всякого элемента документа  $x \in D$  такого, что  $x.FirstChild = Null$  и  $\exists T_i : T_i \in T : m_i = T_i(x)$ , запись  $\exists T_i : T_i \in T : m_i = T_i(x)$  символизирует построение элемента  $m_i$  в результате обработки элемента  $x$  подпрограммой  $T_i$ .

положим  $D_j = \forall x, m : (\exists : (R(x) \wedge Q(x, m_i)) \rightarrow V(x))$ .

2. Для всякого элемента документа  $x \in D$  такого, что  $x.FirstChild = Null$  и  $\exists T_i : T_i \in T : m_i \neq T_i(x)$ , положим  $D_j = \forall x, m : (\exists : (R(x) \wedge Z(x)) \rightarrow V(x))$ .
3. Для всяких элементов  $x, y \in D$ ,  $m \in S_M$  таких, что  $y = x.FirstChild$  и  $\exists T_i : T_i \in T : m_i = T_i(x)$  положим  $D_j = \forall x, y, m : (\exists : ((Vvd(y, x) \wedge R(x) \wedge Q(y, m_i)) \rightarrow P(x)))$ .
4. Для всяких элементов  $x, y \in D$ , таких, что  $y = x.FirstChild$  и  $\exists T_i : T_i \in T : m_i \neq T_i(x)$ , положим  $D_j = \forall x, y, m : (\exists : (Vvd(x, y) \wedge R(x) \wedge Z(x)) \rightarrow P(y))$
5. Для всякой пары элементов  $x, y \in D$  таких, что  $y = x.LastChild$  и  $\exists T_i : T_i \in T : m_i = T_i(x)$  положим:  $D_j = \forall x, y, m : (\exists : (V(x) \wedge VVp(y, x) \wedge Q(y, m_i)) \rightarrow V(x))$
6. Для всякой пары элементов  $x, y \in D$  таких, что  $y = x.LastChild$  и  $\exists T_i : T_i \in T : m_i \neq T_i(x)$ , положим  $D_j = \forall x, y, m : (\exists : ((V(y) \wedge VVp(y, x) \wedge Z(y)) \rightarrow V(x)))$
7. Для всякой пары элементов  $x, y \in D$  таких, что  $y = x.PreviousSibling$  положим  $D_j = \forall x, y : (\exists : ((V(y) \wedge VH(y, x)) \rightarrow P(y)))$ .

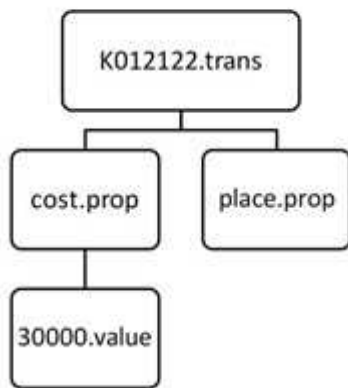
Этап 3. Построение формулы  $ND = \wedge_k ND_k$ .

1. Для всякой пары элементов  $x, y \in D$  таких, что справедливо  $y = x.FirstChild$  положим  $ND_k = Vd(x, y)$ .
2. для всякой пары элементов  $x, y \in D$  таких, что справедливо  $y = x.PreviousSibling$  положим  $ND_k = H(x, y)$ .
3. Для всякой пары элементов  $x, y \in D$  таких, что справедливо  $y = x.LastChild$  положим  $ND_k = Vp(y, x)$ .
4. Для всякого элемента  $x \in D$  такого, что  $E_i = x.NodeType$  положим  $ND_k = E_i(x)$ .
5. Для всякого элемента  $x \in D$  такого, что  $\exists T_i : T_i \in T : m_i = T_i(x)$  положим  $ND_k = Q(x, m_i)[m_i = T_i(x)]$ .
6. Для всякого элемента  $x \in D$  такого, что  $\exists T_i : T_i \in T : m_i \neq T_i(x)$ , положим  $ND_k = Z(x)$ .
7. Служебный нелогический элемент  $[Start()]$  свяжем с атомом  $P(a)$ , а элемент сценария  $[Finish()]$  свяжем с атомом  $False$

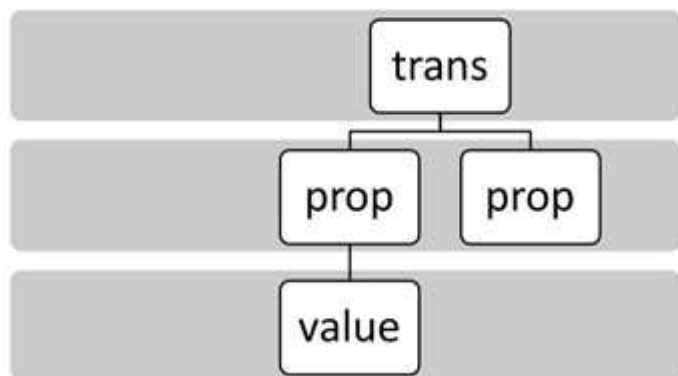
Рассмотрим пример. На рис. 2 приведено диаграммное представление XML-документа DE, его схемы SE, на рис. 3. – диаграммное представление схемы SM результата преобразования. Преобразуем данные из XML-прототипа DE со схемой SE в XML-документ, содержащий фрагмент математической модели со схемой SM.

Документ DE представляет собой упрощенный фрагмент XML-прототипа ТЭК, содержащий агрегированное описание технико-экономических характеристик транспорта природного газа из Северо-Западного федерального округа в Центральный. Корневой элемент типа trans группирует технико-экономические характеристики транспорта газа. Содержимое этого элемента (искомая переменная K012122) соответствует объему транспортируемого ресурса. Первый элемент типа prop с содержимым cost описывает стоимость единицы транспортируемого ресурса. Потомок этого элемента типа value с содержимым 30000 задает значение этого показателя. Элемент типа property с содержимым place описывает пункт назначения транспортируемого ресурса.

Схемой SE заданы требования к отношениям смежности элементов XML-документа DE и требования к содержимому, атрибутам и их значению для каждого элемента DE.



а) Диаграмма исходного XML-документа



б) Диаграмма схемы исходного XML-документа

Рис. 2. Диаграммное представление исходного XML-документа и его схемы.

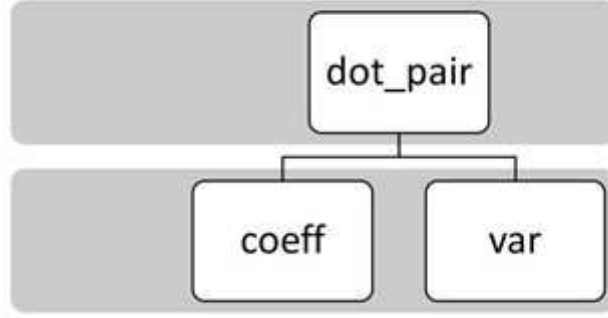


Рис. 3. Диаграммное представление схемы результата преобразования.

Схема  $SM$  математической модели задает требования к отношениям смежности элементов выходного документа и требования к содержимому, атрибутам и их значениям для каждого элемента выходного документа. Элемент схемы  $dot\_pair$  группирует элементы математической модели  $coeff$  и  $var$ , служащие для задания коэффициента и переменной соответственно.

Будем преобразовывать данные из XML-прототипа в математическую модель следующим образом. Содержимое 30000 элемента прототипа типа  $value$  будем преобразовывать в элемент математической модели типа  $coeff$ , содержимое K012122 элемента прототипа типа  $trans$  в элемент  $var$ .

Для этого с элементами схемы  $coeff$ ,  $var$  свяжем нелогические элементы - элементы сценария преобразования данных  $Add(y,x)$ . Элемент сценария  $Add(y,x)$  преобразует (копирует) содержимое элемента XML-документа  $x$  в элемент XML-документа  $y$ .

Кроме того, для построения сценария нам потребуются т.н. служебные элементы – элементы сценария, задающие начало сценария, осуществляющие построение документа в соответствии с имеющейся схемой и завершающие сценарий. Элемент сценария  $Start()$  задает начало сценария преобразования, элемент  $CreateDoc(x,y)$  формирует XML-документ  $x$  по схеме  $y$ , элемент  $Finish()$  завершает сценарий преобразования.

Для сокращения записи обозначим через  $E_0$  тип элемента прототипа  $trans$ ,  $E_1$  - тип элемента прототипа  $prop$ ,  $E_2$ - тип элемента прототипа  $value$ ,  $a$  - содержимое элемента прототипа типа  $trans$ ,  $b,c$ - содержимое элементов прототипа типа  $prop$ ,  $d$  - содержимое элемента прототипа типа  $value$ ,  $M_0$  - тип элемента модели  $dot\_pair$ ,  $M_1$ - тип элемента  $coeff$ ,  $M_2$  - тип элемента  $var$ .

Теорема синтеза для XML-документа  $D_E$  и схем  $S_M$   $S_E$  имеет вид:

$$G(S_E, D_E, T) = ((S_E \rightarrow D_E) \rightarrow ((ND \wedge P(a)) \rightarrow V(a))),$$

где

$$S_E = \left[ \begin{array}{l} \bigwedge \left[ \bigvee x: x \in D (\exists y: y \in S_E (P(x) \wedge E_2(x) \rightarrow R(x))) \right] \\ \bigwedge \left[ \bigvee z, w: (\exists x, y: (R(z) \wedge E_1(z) \wedge E_2(w) \wedge Vd(z, w) \rightarrow VVd(z, w) \wedge R(z))) \right] \\ \bigwedge \left[ \bigvee z, w: (\exists x, y: (R(z) \wedge E_2(z) \wedge E_1(w) \wedge Vp(w, z) \rightarrow VVp(w, z) \wedge R(w))) \right] \\ \bigwedge \left[ \bigvee x: x \in D (\exists y: y \in S_E (\bar{P}(x) \wedge E_1(x) \rightarrow \bar{R}(x))) \right] \\ \bigwedge \left[ \bigvee z, w: (\exists x, y: (R(z) \wedge E_0(z) \wedge E_1(w) \wedge Vd(z, w) \rightarrow VVd(z, w) \wedge R(z))) \right] \\ \bigwedge \left[ \bigvee z, w: (\exists x, y: (R(z) \wedge E_1(z) \wedge E_0(w) \wedge Vp(z, w) \rightarrow VVp(z, w) \wedge R(w))) \right] \\ \bigwedge \left[ \bigvee z, w: (\exists x, y: (E_1(z) \wedge E_1(w) \wedge H(z, w) \rightarrow VH(z, w))) \right] \\ \bigwedge \left[ \bigvee x: x \in D (\exists y: y \in S_E (P(x) \wedge E_0(x) \rightarrow R(x))) \right] \end{array} \right]$$

$$D_E = \left[ \begin{array}{l} \bigwedge \left[ \bigvee x, y: (\exists: (R(c) \rightarrow V(c))) \right] \\ \bigwedge \left[ \bigvee x, y: (\exists: (Vvp(c, a) \wedge V(c) \wedge Z(c) \rightarrow V(a))) \right] \\ \bigwedge \left[ \bigvee x, y: (\exists: (R(d) \rightarrow V(d))) \right] \\ \bigwedge \left[ \bigvee x, y: (\exists: (Vvd(b, d) \wedge R(b) \wedge Z(b) \rightarrow P(d))) \right] \\ \bigwedge \left[ \bigvee x, y: (\exists: (Vvp(d, b) \wedge V(d) \wedge Q(d, M_1) \rightarrow V(d))) \right] \\ \bigwedge \left[ \bigvee x, y: (\exists: (Vvd(a, b) \wedge R(a) \wedge Q(a, M_0) \rightarrow P(b))) \right] \\ \bigwedge \left[ \bigvee x, y: (\exists: (V(b) \wedge VH(b, c) \rightarrow P(c))) \right] \end{array} \right]$$

$$ND = \left[ \begin{array}{l} P(a)[Start()][CreateDoc(S_D)] \wedge Vd(a, b) \wedge Q(a, M_1)[M_0 = Add(a)] \wedge Vd(b, d) \\ \wedge Z(b) \wedge Vp(d, b) \wedge Q(d, M_2)[M_2 = Add(d)] \wedge H(b, c) \wedge Vp(a, c) \\ \wedge Z(c) \wedge E_0(a) \wedge E_1(b) \wedge E_2(d) \wedge E_3(c) \end{array} \right] \rightarrow V(a)$$

Нетрудно заметить, что построенная в соответствии с алгоритмом теорема синтеза представляет собой дизъюнкт Хорна [10, 11]. К атомам вида  $Q(x,y)$  приписаны нелогические элементы  $y=T_i(x)$ , которые указывают

на то, какую подпрограмму преобразования входного элемента  $x$  нужно выполнить, чтобы получить элемент  $y$ . Сценарий преобразования данных представляет собой последовательность команд запуска этих подпрограмм, которая строится в процессе вывода теоремы синтеза. Значения параметров запуска подпрограмм также определяется в процессе вывода.

Рассмотрим модификацию декларативного представления процесса преобразования данных – декларативное представление  $F(S,D)$  процесса контроля соответствия данных, заданных в документе  $D$ , требованиям, представленным в схеме  $S$ . В дальнейшем будем называть его теоремой достоверности. Структура и алгоритм построения теоремы достоверности в целом аналогичны теореме синтеза. Принципиальное отличие заключается в том, что нелогические элементы, используемые при его построении, предназначены для запуска контролируемых подпрограмм, задающих требования к составу и содержанию элементов. Применение этого декларативного представления позволяет проконтролировать используемые в ходе вычислительного эксперимента XML-файлы.

**3. Синтез сценариев преобразования данных и автоматическое доказательство декларативных представлений.** Применение декларативных представлений преобразований данных для обеспечения вычислительного эксперимента заключается в их автоматическом доказательстве и исполнении полученного в ходе доказательства сценария преобразования данных (рис. 4). Автоматическое доказательство теорем синтеза и достоверности может быть выполнено как методом резолюций [8], так и в исчислении позитивно-образованных формул [9]. В ходе автоматического доказательства декларативных представлений синтезируются сценарии преобразования данных или протоколы контроля достоверности XML-документов.

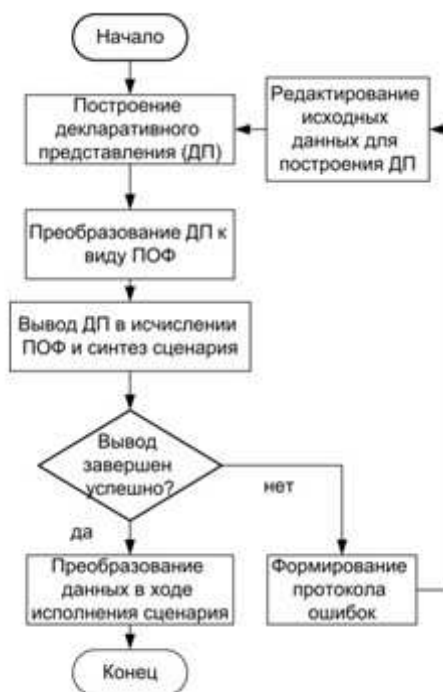


Рис. 4. Алгоритм применения декларативных представлений.

При разработке научно-исследовательского прототипа программного обеспечения для автоматизации проведения комплексных вычислительных экспериментов в исследованиях ЭБ автоматическое доказательство декларативных представлений было реализовано в исчислении позитивно-образованных формул. Применение механизма вывода на основе исчисления позитивно-образованных формул продиктовано более широкими возможностями управления процедурой вывода в исчислении позитивно-образованных формул по сравнению с методом резолюций [10].

Рассмотрим подробнее процедуру логического вывода построенных декларативных представлений в исчислении позитивно-образованных формул. В этом случае логический вывод теорем синтеза и достоверности осуществляется после их преобразования к виду позитивно-образованной формулы (ПОФ)  $\Phi = B_0 \{B_Q\}$ . Вывод ПОФ представляет собой ее многократное преобразование по правилу  $\omega$  [9]. Исходя из особенностей выводимой ПОФ, используем дополнительную стратегию вывода  $\tau$ , введенную в [11].

Элемент  $\rho$  стратегии  $\tau$  заключается в разделении конъюнкта базы ПОФ на конъюнкт вывода  $K_D$ , конъюнкт синтеза  $K_S$ , конъюнкт типов  $K_T$ . Конъюнкт вывода  $K_D$  составляют атомы, которые можно использовать в данный момент при выводе, конъюнкт синтеза  $K_S$  - атомы, уже использованные при выводе, конъюнкт типов  $K_T$  - атомы, постоянно используемые при выводе и не подлежащие преобразованию. Перед началом вывода

полагаем  $K_S = True$ . После каждого успешного шага вывода использованные согласно правилу  $\omega$  атомы из конъюнкта  $K_D$  перемещаются в конъюнкт  $K_S$ . Элемент  $\xi$  стратегии  $\tau$  заключается в поиске подходящего запроса путем просмотра базы слева направо при применении правила  $\omega$ . Глубина просмотра базы при проверке выполнимости запроса определяется мощностью множества атомов, составляющих конъюнкт предусловия запроса. Элемент  $\psi$  стратегии  $\tau$  заключается в пополнении базы новыми конъюнктами примененного запроса, путем их приписывания к базе только слева.

Доказательство ПОФ заключается в циклической обработке ПОФ в соответствии со стратегией  $\tau$  до тех пор, пока это возможно или  $K_D \neq False$ . Невозможность применения стратегии означает недоказуемость ПОФ. Случай  $K_D = False$  означает, что ПОФ успешно доказана.

В соответствии с особенностями вывода ПОФ и введенной стратегией по результатам вывода конъюнкт синтеза  $K_S$  содержит те шаблоны и параметры элементов сценария, которые составляют сценарий преобразования или контроля XML-документа. Результатом исполнения этого сценария будет текстовый или XML-файл, сформированный из элементов сценария  $T$ , соответствующий всем требованиям, заданным в схеме  $S_M$ .

Отметим важные свойства построенного декларативного представления. Введем следующее определение. Структурно достоверным относительно схемы  $S_D$  будем называть такой XML-документ  $D$ , что тип любого его элемента совпадает с типом одного из элементов схемы  $S_D$  и для любой пары элементов документа отношения смежности совпадают с отношениями смежности соответствующей пары элементов в схеме  $S_D$ .

Отметим также, что вообще говоря, набор элементов сценария  $T$  задает отображение схемы  $S_D$  на схему  $S_M$ . При этом важными свойствами этого отображения являются его сюръективность и функциональность [12].

*Утверждение 1.* Теорема достоверности  $F(S, D)$  выводима в исчислении ПОФ тогда и только тогда, когда документ  $D$  структурно достоверен относительно схемы  $S$ .

Доказательство утверждения 1 выполняется построением взаимно-однозначного соответствия между процедурой обхода элементов документа  $D$  и шагами вывода теоремы  $F(S, D)$ .

*Утверждение 2.* Если теорема синтеза  $G(S_D, D, T)$  выводима в исчислении ПОФ и соответствие, задаваемое элементами  $T$  - сюръективно и функционально, то выходной XML-файл и входной XML-документ структурно достоверны относительно заданных схем  $S_M, S_D$  соответственно.

Доказательство этого утверждения основывается на свойствах сюръективности и функциональности набора элементов сценария  $T$  и построении взаимно-однозначного соответствия между шагами вывода теоремы синтеза и процедурой обхода документа  $D$  в прямом порядке.

Таким образом, можно утверждать, что построенные в результате применения декларативных представлений преобразователи обеспечивают соответствие результатов преобразования заданным при построении требованиям.

**Заключение.** В статье разработан подход к построению преобразователей данных на основе применения декларативных представлений процесса преобразования данных. Предложены структура и алгоритм построения декларативных представлений процессов преобразования и контроля данных, рассмотрены вопросы построения преобразователей данных в ходе автоматического доказательства декларативных представлений. Применение построенных таким образом преобразователей позволяет формировать файлы заданного формата, структуры необходимые для обеспечения вычислительного эксперимента – в первую очередь, файлы, содержащие математическую модель или результаты расчета, нуждающиеся в дополнительной интерпретации.

Приведенные в статье результаты используются для автоматизации вычислительного эксперимента в исследованиях проблем энергетической безопасности в Институте систем энергетики им. Л.А. Мелентьева СО РАН. Автор считает, что предложенный подход может использоваться в других предметных областях.

Работа выполняется при поддержке Программы Президиума РАН №2-2.29, грантов РФФИ № 14-07-31268, №14-07-00116, №12-07-00359.

#### ЛИТЕРАТУРА:

1. Теоретические основы системных исследований в энергетике / А.З. Гамм, А.А. Макаров, Б.Г. Санеев и др. – Новосибирск: Наука, 1986 – 331 с.
2. Системные исследования в энергетике: Ретроспектива научных направлений СЭИ–ИСЭМ / отв. ред. Н.И. Воропай. – Новосибирск: Наука, 2010. – 686 с.
3. Интеграция информационных технологий в системных исследованиях энергетики / Л.В. Массель, Е.А. Болдырев, А.Ю. Горнов и др.; под ред. Н.И. Воропай. – Новосибирск: Наука, 2003. – 320 с.
4. Массель Л.В. Интеграция семантического и математического моделирования в исследованиях проблем энергетической безопасности/ Труды Международной конференции «Моделирование-2012», Киев: ИПМЭ НАН Украины, 2012. – С. 270-273

5. Массель Л.В., Курганская О.В. Автоматизация вычислительного эксперимента на основе логических моделей - Вестник Иркутского государственного технического университета, №2(49). – Иркутск: Издательство Иркутского государственного университета, 2011. – с. 8-14
6. Хантер Д., Рафтер Д. и др. XML. Базовый курс = Beginning XML — М.: Вильямс, 2009. — 1344 с.
7. Ершов Ю.Л., Палютин Е.А. Математическая логика. – М.: Наука, 1979. – 320 с.
8. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем - М.: Наука, 1983. - 360 с.
9. Васильев С.Н. и др. Интеллектуальное управление динамическими системами. – М.: Физматлит, 2000. – 352 с.
10. Ларионов А.А. и др. Программная система КВАНТ/4 для автоматического доказательства теорем / А.А. Ларионов, И.Н. Терехин, Е.А. Черкашин, А.В. Давыдов // Труды ИМЭИ ИГУ. Математика и информатика: сб. науч. тр. – Иркутск: Изд-во ИГУ, 2011. – С.77-86.
11. Бутаков М.И. и др. Контроль диалога объектных программ на основе позитивно-образованных формул / М. И. Бутаков, В. И. Курганский // Вопросы современной науки и практики. Университет им. В.И. Вернадского, 2010, № 4-6(29). – С. 106-115.
12. Кузнецов О.Н., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергоатомиздат, 1988. – 480 с.