

УДК 004.067

Н. А. Жукова

*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)*

## **О возможностях синтеза многоуровневых моделей объектов**

*Рассматривается новый метод синтеза моделей объектов, имеющих многоуровневую структуру. Предлагаемый метод развивает существующие методы индуктивного и дедуктивного синтеза формальных структур. Определяется многоуровневая структура моделей объектов, приводится описание индуктивно-дедуктивного метода синтеза. Метод предусматривает обработку поступающих от объектов данных, построение на их основе автомата дедуктивного синтеза и применение автомата для построения моделей объектов. При построении автомата выполняется индуктивный синтез многоуровневых структур в пространстве элементов исходных данных. На основе построенных структур определяются параметры автомата дедуктивного синтеза. Дедуктивный синтез моделей объектов с использованием построенного автомата предполагает формулировку теоремы о существовании моделей и их доказательство. Метод позволяет строить новые структуры для описания моделируемых объектов, имеет низкую вычислительную сложность. Приводится пример синтеза модели объекта, имеющего четырехуровневую структуру.*

### **Многоуровневый синтез, автоматная модель, моделирование объектов**

В настоящее время существенно увеличилось количество используемых технических объектов. Такие объекты применяются как при решении различных специализированных задач, так и в повседневной жизни. Высокий уровень развития искусственного интеллекта позволяет наделять даже достаточно простые объекты интеллектом. Объекты могут взаимодействовать между собой с использованием различных средств связи. В этих условиях возникает задача мониторинга и управления локальными и глобальными сетями объектов [1]. Основная сложность мониторинга и управления состоит в том, что объекты имеют различные структуры и поведение, которые изменяются во времени.

Основным источником информации об объектах являются поступающие от них данные. Объекты обладают достаточно широкими возможностями по сбору данных. Измерения выполняются с использованием установленных на объектах датчиков, которые позволяют измерять значения различных параметров объектов. Результаты измерений представляют собой многомерные вре-

менные ряды количественных, реже качественных значений. Объекты, как правило, имеют многоуровневую физическую и логическую структуры. Элементами их физической структуры являются узлы, агрегаты, блоки и т. д., к ним также относятся подсистемы, системы и т. д. Между структурными элементами объектов имеются множественные связи. Состав элементов, их состояния и связи между ними могут изменяться динамически.

Для решения задач мониторинга и управления объектами необходимо иметь структурные описания этих объектов, характеризующие их состояния на различные моменты времени. На каждый момент времени состояние объекта может описываться некоторой структурой, для которой определены элементы, установлены связи между ними. Такие описания можно рассматривать как модели объектов.

Модели объектов могут отражать их физическую или логическую структуру. Возможно построение моделей, рассматривающих объекты с других точек зрения. Требования к моделям

определяются составом решаемых задач. Источником данных для построения моделей являются собираемые об объектах данные. Построение моделей может быть обеспечено за счет их синтеза. Возможности синтеза ограничены отсутствием описаний множества возможных структур объектов, а также высокой вычислительной сложностью применяемых при синтезе методов. Известные методы синтеза не в полной мере удовлетворяют потребностям практики по точности и оперативности.

В статье рассматривается новый метод синтеза, позволяющий строить многоуровневые модели объектов по полученным от объектов данным.

**Анализ методов синтеза.** Синтез моделей может осуществляться на основе полных формальных спецификаций, в качестве которых могут использоваться, например, логические или математические функции. При описании логических функций, как правило, применяется логика первого порядка. Входными данными могут также являться примеры пар входных и выходных данных и примеры синтеза формальных структур. Отличие двух последних вариантов задания исходных данных состоит в том, что примеры синтеза дополнительно показывают способы преобразования данных, которые применяются при синтезе. Кроме того, в качестве входных данных могут выступать примеры синтезируемых структур.

**Классические методы синтеза.** Классическими методами синтеза являются методы индуктивного и дедуктивного синтеза.

*Дедуктивный синтез.* При дедуктивном синтезе [2], [3] модели строятся на основе спецификации и множества логических аксиом. В ходе синтеза строятся возможные варианты структур, соответствующих спецификации. Для дедуктивного синтеза требуется полная аксиоматика и полная формальная спецификация.

Идея построения интерпретируемых решений с использованием доказательств за счет объединения отдельных более частных решений рассматривалась еще в 1932 г. А. Н. Колмогоровым [4]. После разработки первого метода для автоматического доказательства теорем появилось много работ по дедуктивному синтезу ([5]–[8] и др.), в большинстве которых рассматриваются проблемы синтеза программных структур. Синтез предусматривает доказательство существования

структур на заданном множестве условий, которые связывают исходные данные с планируемым результатом, и извлечение результирующих структур из доказательств с применением таких методов, как метод резолюций [9] и метод обратного вывода [10].

К более поздним работам в этой области относятся [11]–[17] и др. Во многих из них рассматриваются возможности распараллеливания доказательств [12], а также использования многоуровневого подхода [16], [17].

*Индуктивный синтез* позволяет использовать менее строгие спецификации. При индуктивном синтезе вместо логического вывода на основе формальных спецификаций выполняется итеративный поиск требуемой структуры. Данный метод обладает высокой гибкостью, но сталкивается с проблемой масштабирования.

Индуктивный синтез получил развитие в области синтеза программных структур. Первыми работами в этой области стали [18]–[20].

При таком подходе выполняется синтез возможных структур, которые затем оцениваются. Если полученные оценки не отвечают требованиям, то синтезируются другие структуры. Исследовались различные возможные способы синтеза структур, в том числе с использованием генетического подхода [21], метода Монте-Карло по схеме марковской цепи [22], вероятностный вывод [23]. Для управления синтезом генерируются ограничения на основе неуспешных попыток синтеза, которые рассматриваются как контрпримеры (CEGIS – counterexample-guided inductive synthesis) [24]. Для упрощения процессов синтеза было предложено определять каркасы пространств возможных структур [25].

Наряду с дедуктивным и индуктивным методами также существует подход к синтезу на основе трансформаций [26]. Он предусматривает итеративную трансформацию верхнеуровневых описаний. В результате трансформаций обеспечивается переход от верхнеуровневых описаний к требуемым структурам.

На практике классические методы синтеза, как правило, применяются в комбинации с методами, позволяющими ограничить пространство поиска решений и тем самым снизить сложность поиска. К таким методам относятся: разрешение ограничений, статистические и методы машинного обучения.

Методы разрешения ограничений предусматривают два шага: генерацию ограничений и разрешение ограничений. Ограничения могут генерироваться на основе инвариантов, исходных данных, путей поиска. Недостатком первой группы методов [27] является сложность генерируемых ограничений. Это связано с тем, что индуктивные инварианты формулируются с использованием логик, имеющих большую выразительность, чем логика, применяемая при описании синтезируемых структур. Методы второй группы лишены этого недостатка, однако они гарантируют соответствие спецификации и синтезированных структур только на множестве исходных данных [28]. Методы этой группы используются совместно с методами поиска на основе контрпримеров. К третьей группе методов относятся промежуточные варианты генерации ограничений. Генерируемые ограничения обеспечивают соответствие спецификаций и синтезируемых структур на множестве исходных данных, при которых осуществляется проход по заданному набору путей [29]. Синтезируемые ограничения могут описываться с использованием логики второго порядка, содержать кванторы всеобщности. При разрешении ограничений осуществляется переход к логике первого порядка за счет использования шаблонов. Для исключения универсальных кванторов применяются алгоритмы покрытия, лемма Фаркаша и другие. Далее используются SAT/SMT-решатели.

Статистические методы и методы машинного обучения применяются для построения функций правдоподобия, которые могут использоваться при определении путей дальнейшего поиска на каждом из шагов синтеза формальных структур. Определение пути предполагает выбор не терминального пути. Функции правдоподобия могут строиться на основе исходных данных или данных, полученных из дополнительных источников [30].

**Новые методы синтеза.** На основе классических методов синтеза разработан ряд новых методов. Далее рассматриваются некоторые из них [31], [32].

*Управляемый нисходящий поиск.* При таком подходе выполняется нисходящий поиск с использованием численных методов [33]–[37]. Имеются различные способы оптимизации поиска, при которых могут применяться эвристики. Управление поиском осуществляется с помощью дедуктивного подхода или за счет построения

множества ограничений. Значительное внимание в рамках развития этого направления уделяется построению управляющих функций, предназначенных для прогнозирования для каждой ветки поиска на каждом шаге вероятности построения требуемой структуры. В качестве функций используются вероятностные модели или нейронные сети. Основным недостатком методов управляемого нисходящего поиска состоит в том, что применение нейронных сетей может оказаться неоправданным. При использовании сети может затрачиваться больше ресурсов, чем при полном поиске.

*Построение скетчей структур.* Построение структур на основе скетчей предусматривает выделение двух фаз синтеза: первая фаза – это синтез скетчей структур [25], вторая – синтез самих структур. Синтез на каждой фазе обеспечивается за счет обучения соответствующей модели. На различных фазах используются разные модели. Для обучения моделей задается язык написания скетчей, функции преобразования фрагментов кода в скетчи, база данных скетчей, ассоциированных со спецификациями. Управление синтезом осуществляется с использованием усовершенствованных методов управления на основе контрпримеров. В частности предложен метод синтеза, управляемого оракулом (OGIS – oracle-guided inductive synthesis) [38]. Как показала практика, использование промежуточного шага при синтезе структур позволяет получать очень хорошие результаты, что ожидаемо. Задача синтеза высокоуровневых структур по спецификациям оказывается значительно более простой, чем задача синтеза конечной структуры. Аналогично в разы меньших затрат требует построение структур по скетчам.

*Построение нейронных сетей.* Для синтеза моделей объектов необходимо, чтобы нейронные сети были способны обеспечивать оперативную и глубокую обработку данных. Наиболее подходящими нейросетевыми структурами являются рекуррентные нейронные сети с ассоциативной обработкой информации. Однако используемые в настоящее время рекуррентные нейронные сети [39]–[44] не позволяют одновременно выполнять требования по глубине и оперативности. В частности, сети Хопфилда, Коско позволяют глубоко, но не оперативно обрабатывать информацию. Известные нейронные сети реального времени (RMLP, RTRN, сеть Эльмана) [44] реализуют быструю, но не глубокую ассоциативную обработку. В последние годы получены некоторые

новые решения [45]–[48], которые могут позволить синтезировать модели объектов. Однако разработка новых нейронных сетей может потребовать значительного времени.

Существует ряд других новых методов синтеза формальных структур. В рамках методов, предусматривающих управление поиском на основе данных о предшествующих неудачных попытках синтеза, предложено учитывать неудачные попытки поиска за счет выявления причин и формулировки новых ограничений [49] на их основе. Такой подход получил название поиска, управляемого конфликтами. Значительный интерес представляют методы, позволяющие выявлять принципы построения и функционирования наблюдаемых объектов [50]. Эти методы предусматривают поиск правил поведения при обучении с подкреплением, а затем синтез структур по этим правилам. Несмотря на то что синтезируемые программы не оптимальны, они обладают существенным преимуществом – хорошей интерпретируемостью.

В настоящее время наиболее активно продолжают развиваться методы индуктивного синтеза формальных структур. Это обусловлено высокой заинтересованностью ведущих мировых компаний по разработке программного обеспечения в реализации возможностей автоматического синтеза при создании новых программных продуктов и сопровождении существующих.

Развитие методов дедуктивного синтеза происходит значительно более медленно. До настоящего времени не решены проблемы построения полных формальных спецификаций и высокой вычислительной сложности доказательства сформулированных теорем.

**Особенности синтеза многоуровневых моделей объектов.** При выборе методов синтеза формальных структур для построения моделей объектов учитывается их способность строить новые структуры и возможность перестраивать имеющиеся, а также вычислительная сложность методов. Требование к способности построения новых структур и перестройке существующих вызвано структурной сложностью моделируемых объектов, возможностью изменения структур объектов и их поведения. Для таких объектов нельзя построить множество всех возможных структур. Требование к низкой сложности определяется большим числом возможных состояний объектов и переходов между ними.

Существующие методы не обеспечивают возможность синтеза новых структур по исходным данным. Индуктивные методы позволяют синтезировать модели объектов по потокам исходных данных. Однако эти методы осуществляют поиск структур среди множества возможных и не предполагают построение новых. Построение новых структур возможно с применением методов дедуктивного синтеза. Эти методы требуют формальных спецификаций и не предусматривают синтез моделей по потокам исходных данных. Кроме того, как индуктивные методы, так и дедуктивные имеют высокую вычислительную сложность, не позволяющую применять их для синтеза моделей объектов.

Рассмотренные требования приводят к необходимости совместного использования индуктивных и дедуктивных методов при синтезе моделей объектов. Индуктивный синтез может позволить строить модели объектов по потокам исходных данных или их фрагментам. Множество построенных индуктивных моделей определяет пространство, в котором объекты имеют формальное описание. В этом пространстве возможно применение методов дедуктивного синтеза, что позволит строить новые дедуктивные модели объектов за счет формулирования и доказательства теорем об их существовании.

Для многоуровневых объектов проблема сложности синтеза их моделей может быть решена за счет синтеза моделей на различных уровнях, т. е. применения многоуровневого синтеза.

Таким образом, для построения моделей объектов требуется многоуровневый индуктивно-дедуктивный синтез. Для этого необходимо обеспечить возможность многоуровневого описания объектов за счет синтеза многоуровневых индуктивных моделей. Дедуктивный синтез должен обеспечить возможность синтеза дедуктивных моделей на различных уровнях описаний объектов.

В основу индуктивного синтеза моделей объектов могут быть положены методы синтеза формальных структур на основе построения скетчей. В настоящее время они предусматривают синтез двухуровневых структур. Однако ограничения, препятствующие переходу от двухуровневых структур к многоуровневым, отсутствуют.

В основу дедуктивного синтеза объектов могут быть положены классические методы, рассмотренные в [5], [7] с учетом идей по переходу от одноуровневого синтеза к многоуровневому [16], [17].

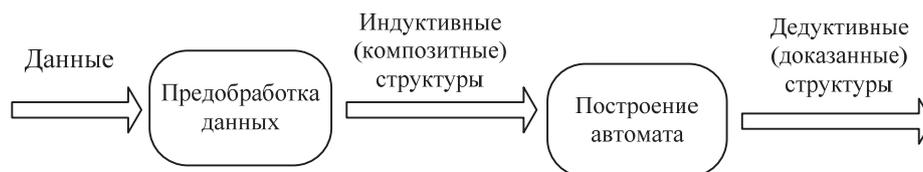


Рис. 1

**Метод автоматного индуктивно-дедуктивного синтеза многоуровневых моделей объектов.** Метод автоматного индуктивно-дедуктивного синтеза основан на построении автомата дедуктивного синтеза и его применении (рис. 1).

Исходными данными для автоматного синтеза являются исходные потоки данных.

Выходными данными являются синтезированные модели объектов.

Потоки данных представляют собой структуры, содержащие отдельные элементы данных о моделируемых объектах. Модели объектов описываются с помощью многоуровневых графовых структур, отражающих структуры моделируемых объектов, входящие в их состав элементы и связи между элементами.

Автоматный синтез моделей объектов предусматривает следующие шаги:

1. Предобработка исходных данных для построения автомата синтеза моделей объектов.
2. Построение автомата синтеза на основе предварительно обработанных данных.
3. Применение построенного автомата для синтеза моделей объектов.

При предобработке данных поступающий от объектов исходный поток представляется в виде последовательности фрагментов. Для каждого фрагмента строится индуктивная модель объекта. Индуктивная (композиционная) модель описывает объект по фрагменту исходных данных путем связывания элементов данных между собой по известным правилам. При построении таких моделей применяются методы индуктивного синтеза. Множество возможных моделей ограничено вариативностью порождающих их правил. Таким образом, задача построения модели сводится к поиску ее варианта в множестве возможных.

В результате предобработки последовательностей фрагментов данных формируются последовательности индуктивных моделей. Между элементами устанавливаются связи, которые определяют преобразования, позволяющие выполнять переходы между структурами.

На основе построенных индуктивных моделей и установленных между ними связей можно построить автомат для синтеза дедуктивных мо-

делей объектов. Их отличие от индуктивных состоит в том, что дедуктивные модели строятся путем доказательства их существования, в то время как индуктивные модели строятся по известным правилам.

При построении автомата определяется пространство состояний автомата и возможные переходы между ними. Пространство состояний автомата определяется в соответствии с множеством элементов построенных индуктивных структур. Множество переходов строится на основе установленных между элементами связей. Уточнение автомата синтеза осуществляется при поступлении каждого нового фрагмента данных.

Синтез дедуктивных структур с использованием автомата основан на дедуктивном выводе. При этом задаются исходная и целевая модели объектов. Исходная модель объекта может отражать его текущее состояние и быть построена по полученным от объекта данным путем индуктивного синтеза. Целевая модель формулируется в соответствии с требованиями, предъявляемыми конечными пользователями к результатам моделирования. Синтез дедуктивных структур предусматривает формулировку теоремы о существовании целевой модели и доказательства перехода от исходной модели к целевой. При успешном доказательстве целевая модель рассматривается как дедуктивная модель объекта, из результатов доказательства извлекаются возможные пути, позволяющие перейти от исходной модели к целевой.

**Вычислительная сложность автоматного индуктивно-дедуктивного синтеза многоуровневых моделей объектов.** Индуктивно-дедуктивный синтез позволяет строить модели объектов с многоуровневой структурой. Нижний уровень – это уровень исходных данных. Его элементами являются элементы исходных данных. Верхний уровень моделей – это уровень моделируемого объекта. Элементы этого уровня характеризуют объект в целом. Число и состав промежуточных уровней определяется структурой моделируемых объектов. Для каждого уровня модели определяется свое признаковое пространство, в котором описываются элементы и устанавливаемые между ними связи.

*Многоуровневый индуктивный синтез.* Исходными данными для индуктивного синтеза моделей являются элементы данных об объектах, а также множество правил связывания элементов между собой. Результатом данного синтеза являются многоуровневые индуктивные модели объектов. В построенных моделях для каждого уровня определены элементы данных и связи между ними, правила переходов между различными уровнями.

Многоуровневый индуктивный синтез моделей объектов осуществляется в направлении от нижнего уровня к верхнему. Построение уровней выполняется последовательно.

Индуктивный синтез  $i$ -го уровня проходит по следующей схеме. Предполагается, что уровни от 0 до  $i - 1$  построены. В результате синтеза предшествующего ( $i - 1$ ) уровня построено признаковое пространство для  $i$ -го уровня, определены элементы данных  $i$ -уровня.

В этих условиях при синтезе  $i$ -уровня выполняются следующие шаги:

1. Между элементами  $i$ -уровня устанавливаются связи. При этом рассматриваются различные связи из допустимого множества. Проверяется наличие таких связей между элементами  $i$ -уровня.

2. Формируются группы из элементов  $i$ -уровня. Группы формируются исходя из выявленных между элементами связей. Связанные элементы относятся к одной группе.

3. Перекодирование элементов  $i$ -уровня. При перекодировании каждая группа элементов рассматривается как отдельный новый элемент. Определяются характеристики новых элементов. В соответствии с характеристиками строится признаковое пространство описания новых элементов, которые описываются в новом пространстве.

4. Новые элементы назначаются элементами уровня ( $i + 1$ ). Новое признаковое пространство назначается признаковым пространством уровня ( $i + 1$ ).

Далее осуществляется переход к синтезу ( $i + 1$ )-уровня.

Среди перечисленных шагов синтеза вычислительно сложным является шаг установления связей между элементами. При одноуровневом синтезе при построении моделей объектов необходимо рассматривать связи на множестве всех исходных элементов данных. При многоуровневом синтезе учитывается структура объекта. Это

позволяет при определении связей на нижних уровнях модели независимо рассматривать группы элементов, которые характеризуют отдельные составляющие объектов. На верхнем уровне моделей рассматриваются объекты в целом, однако на этом уровне элементами описания объекта являются модели отдельных составляющих объектов. Во всех случаях поиск зависимостей осуществляется на множествах, число элементов в которых меньше числа исходных элементов данных.

*Многоуровневый дедуктивный синтез.* Исходными данными для него являются исходная модель объекта и целевая модель объекта. В результате этого синтеза требуется доказать существование целевой модели и определить возможные пути перехода от исходной модели к целевой.

Многоуровневый дедуктивный синтез состоит из трех основных шагов:

1. Формулировка теоремы о существовании целевой модели при известной исходной модели.

2. Попытка доказательства теоремы на основе прямого вывода.

3. Восстановление путей перехода от исходной модели к целевой из результатов доказательства на основе обратного вывода.

Формулировка теоремы предусматривает построение множества исходных данных и множества целевых данных. Элементы этих множеств определяются в соответствии с элементами и связями, имеющимися в исходной и целевой моделях.

Цель дедуктивного синтеза состоит в доказательстве всех элементов данных из множества целевых данных при условии, что известны элементы множества исходных данных.

Для доказательства теоремы должны быть заданы множества внутренних элементов данных и правила, определяющие возможные переходы от исходных элементов к целевым. Путь между исходными и целевыми элементами может включать многие промежуточные элементы.

Множества внутренних элементов и правила переходов строятся и перестраиваются на основе индуктивных моделей, построенных по потокам исходных данных. При дедуктивном синтезе моделей объектов считается, что эти элементы и правила известны.

Доказательство сформулированной теоремы путем прямого вывода предполагает, что имеется множество доказанных элементов данных и множество не доказанных элементов данных. В нача-

ле доказательства множество доказанных элементов содержит только элементы из множества исходных данных. На каждом шаге доказательства доказываются все возможные элементы из множества не доказанных. Для доказательства элемента необходимо, чтобы существовали правила, позволяющие перейти от доказанных элементов к доказываемому. Если такие правила найдены, то доказываемый элемент перемещается из множества не доказанных элементов в множество доказанных. Устанавливаются связи между доказанным элементом и элементами, которые использовались при доказательстве. Теорема считается доказанной, если доказаны все целевые элементы.

В результате прямого доказательства оказываются доказанными многие элементы, которые не требуются для перехода от элементов из множества исходных данных к элементам из множества целевых данных. Такие элементы исключаются из рассмотрения за счет применения обратного вывода. На основе пути, построенного при обратном выводе, определяется последовательность переходов между элементами множества исходных данных и элементами множества целевых данных. Эта последовательность переходов определяет путь от исходной модели объекта к целевой.

Вычислительно сложным является шаг дедуктивного синтеза, на котором выполняется доказательство сформулированной теоремы. В случае одноуровневого синтеза доказательство осуществляется на множестве всех элементов данных и установленных для них связей. В случае многоуровневого синтеза доказательство осуществляется на различных уровнях. При доказательстве на некотором уровне рассматриваются только те элементы данных и правила переходов между ними, которые определены в моделях объектов этого уровня.

При многоуровневом синтезе доказательство выполняется, начиная с верхнего уровня.

Доказательство на  $i$ -м уровне осуществляется по следующей схеме. Предполагается, что попытки доказательства на уровнях, начиная с наиболее высокого и до  $(i + 1)$ , признаны неуспешными. Для  $(i + 1)$ -уровня определены множества доказанных и множества недоказанных элементов данных. В этих условиях выполняется:

1. На основе множеств доказанных и недоказанных элементов  $(i + 1)$ -уровня определяются множества доказанных и недоказанных элементов  $i$ -уровня. При этом используются правила переходов между различными уровнями, установленные при индуктивном синтезе моделей.

2. Выполняется попытка доказательства теоремы на  $i$ -м уровне.

3. В соответствии с результатами доказательства перестраиваются множества доказанных и недоказанных элементов данных. Если множество недоказанных элементов данных пусто, то теорема считается доказанной. Иначе осуществляется переход на более низкий уровень.

Если достигнут нулевой уровень и на нем попытка доказательства оказалась неуспешной, то теорема считается недоказанной.

При многоуровневом синтезе во многих случаях оказывается возможным доказать все или значительную часть целевых элементов данных на верхних уровнях. С учетом того что число элементов на более высоких уровнях меньше числа элементов на более низких уровнях, вычислительная сложность дедуктивного синтеза на верхних уровнях меньше, чем на нижних. Численные значения сложности зависят от числа уровней моделей и соотношения количества элементов между уровнями.

**Пример индуктивно-дедуктивного синтеза многоуровневой модели.** Рассмотрим четырехуровневый объект, на котором установлено восемь датчиков. Датчики измеряют параметры, относящиеся к двум различным системам. Такими датчиками могут являться датчики температуры и давления. В состав каждой системы входят две подсистемы. Структура объекта показана на рис. 2.

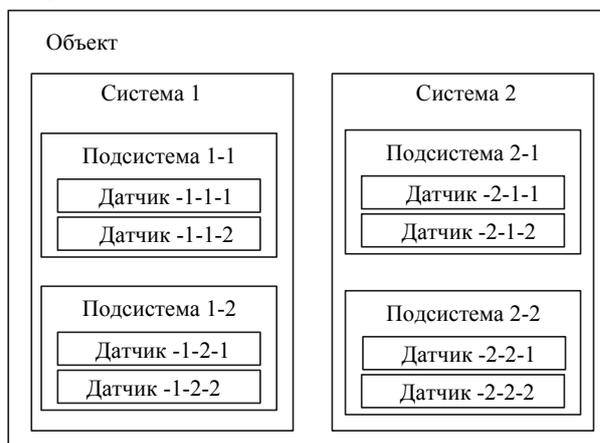


Рис. 2

Рассмотрим индуктивный синтез модели объекта. Исходными данными для него являются результаты измерений значений параметров объекта, которые представляют собой временные ряды. Измерения выполняются с помощью датчиков. Синтез осуществляется за четыре шага (рис. 3). На первом шаге строится нулевой уровень модели. Элементами этого уровня являются исходные временные ряды значений параметрами А31–А38.

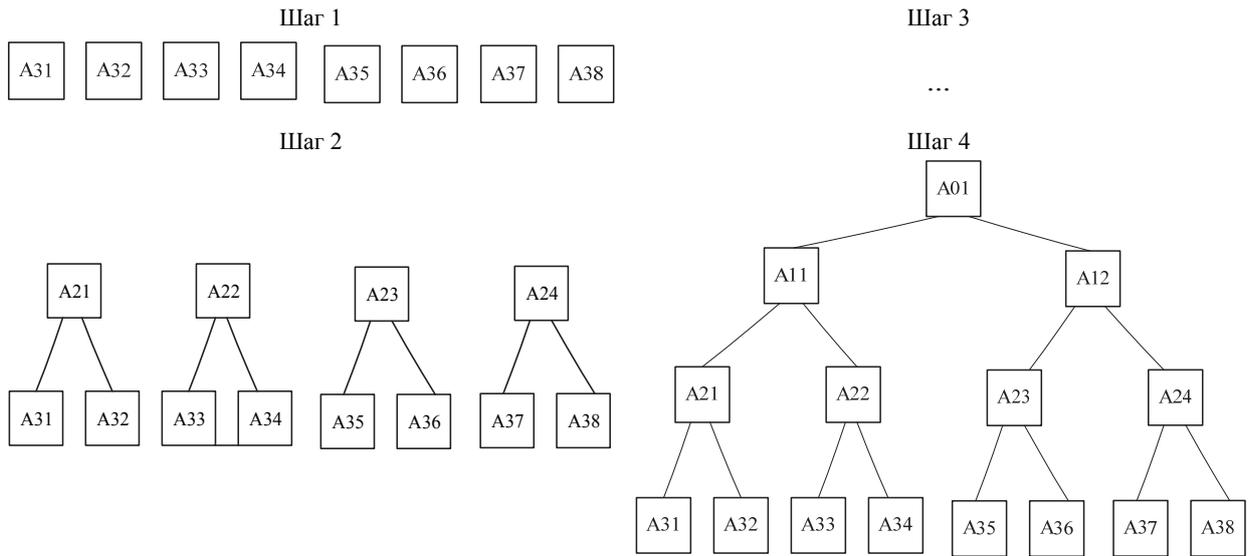


Рис. 3

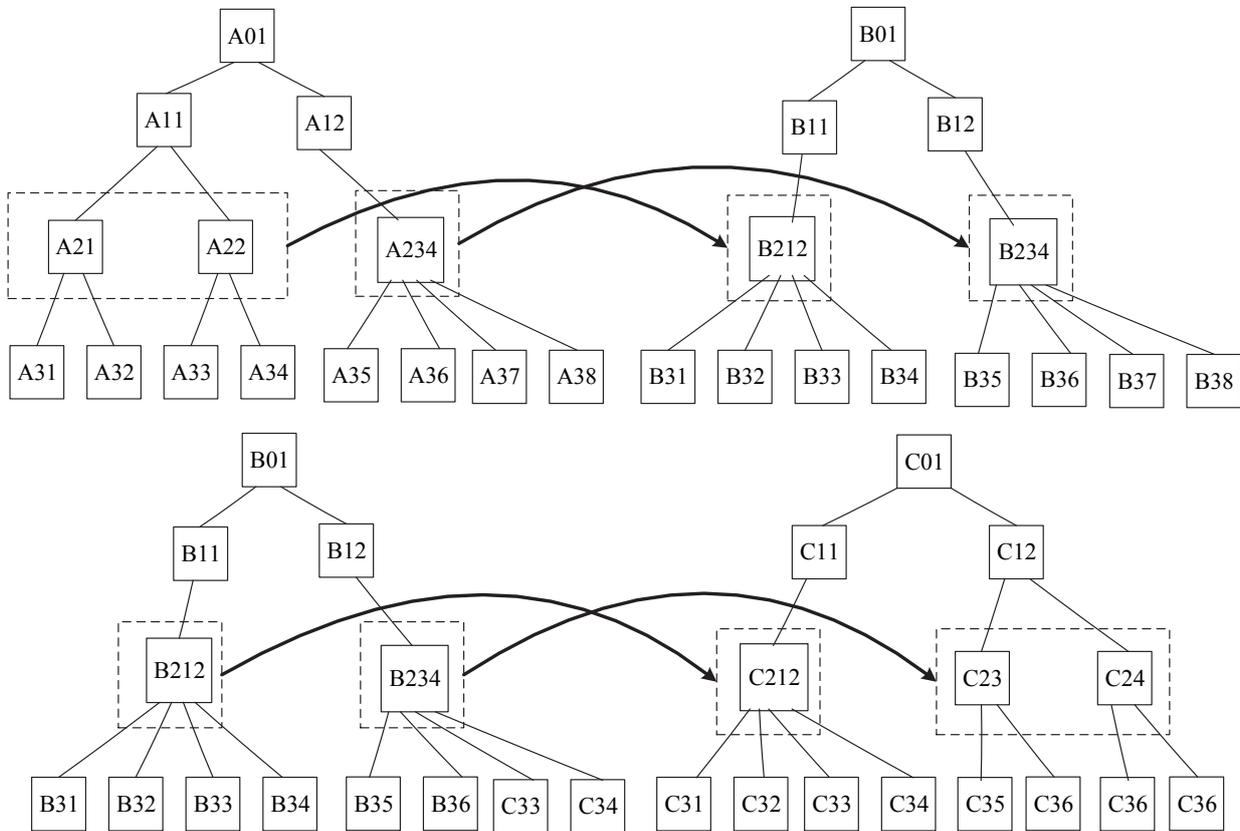


Рис. 4

Устанавливается, что между ними имеются следующие парные связи:  $A31 \leftrightarrow A32$ ,  $A33 \leftrightarrow A34$ ,  $A35 \leftrightarrow A36$ ,  $A37 \leftrightarrow A38$ . Зависимости отражают изменения между температурами и давлениями элементов подсистем моделируемого объекта. На втором шаге связанные элементы объединяются и формируются новые элементы A21–A24. Элементы модели второго уровня описывают состояние подсистем 1–1, 1–2, 1–3 и 1–4. Устанавливается, что связанными являются эле-

менты A21–A22 и A23–A24. На третьем уровне на основе этой связи определяются новые элементы A11 и A12, характеризующие системы объекта. Элемент A01 четвертого уровня характеризует моделируемый объект в целом, исходя из описаний его систем.

Рассмотрим построение автомата синтеза моделей объектов. Пусть в результате наблюдения за объектом получен поток данных. Этот поток представлен в виде трех фрагментов данных, со-

ответствующих трех различным состояниям объекта. Пусть в первом состоянии (состоянии А) система 1 активна, а система 2 пассивна, во втором состоянии (состоянии В) обе системы пассивны, а в третьем (состоянии С) – активна только система 2. В состоянии, когда система активна, процессы активно протекают во всех ее подсистемах. В системах, которые находятся в пассивном состоянии, процессы неактивны. При этом система может рассматриваться как единый элемент без разделения на подсистемы.

Пусть по данным каждого из фрагментов синтезированы индуктивные модели. На рис. 4 такими моделями являются А, В и С, которые имеют разные структуры. В соответствии с моделями с течением времени в структуре объекта наблюдаются следующие изменения:

– при переходе из состояния А в состояние В наблюдается склеивание элементов А21 и А22; в состоянии В им соответствует один элемент В212;

– при переходе из состояния В в состояние С наблюдается разделение элемента В234; в состоянии В ему соответствуют элементы С23 и С24.

Построенные индуктивные модели и связи между ними позволяют определить следующие параметры автомата: допустимое множество состояний и допустимое множество правил перехода между состояниями. Допустимые множества определяются для различных уровней модели. Это позволяет построить многоуровневый автомат.

В соответствии с рис. 4 автомат характеризуется четырьмя допустимыми множествами состояний:

$$\{ \{E31, E32, E33, E34, E35, E36, E37, E38\}; \\ \{E21, E22, E23, E24\}; \{E11, E12\}; \{E01\} \}.$$

Установленные между элементами связи позволяют определить следующее множество допустимых связей между элементами первого уровня:

$$\{ \{E21, E22\} \leftrightarrow E212 \}; \\ \{ E234 \leftrightarrow \{E23, E24\} \}.$$

Эти зависимости представляются в виде допустимых правил:  $\{ \{E_{xy}, E_{xz} \} \rightarrow E_{xyz} \}; \{ E_{xyz} \rightarrow \{E_{xy}, E_{xz} \} \}$ , которые определяют возможность независимого перехода подсистем из активного состояния в пассивное и обратно.

Рассмотрим дедуктивный синтез модели объекта, у которого все подсистемы находятся в активном состоянии, при условии, что в исходный момент времени все подсистемы неактивны.

Тогда модель объекта в исходном состоянии имеет вид:  $\{ \{ E31, E32, E33, E34, E35, E36, E37,$

$E38\}; \{E212, E234\}; \{E11, E12\}; \{E01\} \}$ . Целевой моделью является следующая:  $\{ \{E31, E32, E33, E34, E35, E36, E37, E38\}; \{E21, E22, E23, E24\}; \{E11, E12\}; \{E01\} \}$ .

Доказательство существования целевой модели осуществляется на первом уровне за два шага. Применение первого правила обеспечивает преобразование исходной модели к виду  $\{ \{ E31, E32, E33, E34, E35, E36, E37, E38 \}; \{ E21, E22, E234 \}; \{ E11, E12 \}; \{ E01 \} \}$ . Применение второго правила обеспечивает доказательство существования целевой модели объекта. Путь между исходной и целевой моделями описывается в виде  $\{ E212 \rightarrow \{ E21, E22 \}; \{ \{ E23, E24 \} \rightarrow E234 \}$ .

В рассматриваемом примере задача дедуктивного синтеза решается на первом уровне модели. Процесс доказательства начинается с верхнего, третьего уровня. Однако попытки доказательства на третьем и втором уровнях будут неуспешными, поскольку векторы характеристик элементов в исходной и целевой моделях на этих уровнях будут отличаться. Доказательство на нулевом уровне проводить не требуется, поскольку целевая модель оказывается доказанной на более высоком первом уровне.

Предлагаемый автоматный индуктивно-дедуктивный синтез направлен на решение задачи синтеза моделей объектов по поступающим от них данным при мониторинге и управлении этими объектами. Рассматриваются объекты, имеющие многоуровневую структуру. Модели объектов строятся на основе поступающих от объектов данных.

Предложен новый метод индуктивно-дедуктивного синтеза моделей объектов, который предусматривает построение автомата синтеза и его применение для построения моделей объектов.

Синтез моделей объектов осуществляется в два шага. Вначале проводится построение моделей объектов по исходным данным с использованием методов индуктивного синтеза. Множество индуктивных моделей позволяют определять возможные состояния объектов и правила их связывания. Далее применяются методы дедуктивного синтеза. Он дает возможность строить дедуктивные модели объектов за счет формулирования и доказательства теорем об их существовании.

Проблема сложности синтеза моделей объектов решается за счет перехода от одноуровневого синтеза к многоуровневому. При многоуровневом синтезе решение общей задачи синтеза моделей объектов состоит в решении многих отдельных задач. Число

элементов, рассматриваемое при решении отдельных задач синтеза, меньше, чем число элементов, рассматриваемое при синтезе моделей объектов без учета их многоуровневой структуры.

На практике многоуровневый синтез может позволить в разы снизить требования к вычислительным ресурсам.

## СПИСОК ЛИТЕРАТУРЫ

1. Huang Jun, Hua Kun. Managing the Internet of Things: Architectures, Theories and Applications. UK: The Institution of Engineering and Technology, 2016. 226 p.
2. Green C. Cordell. Application of theorem proving to problem solving // Proc. of the 1<sup>st</sup> Intern. Joint Conf. on Artificial intelligence (IJCAI'69). San Francisco, Washington: Morgan Kaufmann Publishers Inc., 1969. P. 219–240.
3. Manna Z., Waldinger R. J. Toward automatic program synthesis // Communications of the ACM. Vol. 14, № 3. P. 151–165.
4. Kolmogorov A. N. Zur deutung der intuitionistischen logic // Math. Zeitschr. 1932. № 35. P. 58–65.
5. Robinson J. A. A machine – oriented logic based on resolution principle // J. of the ACM. 1965. № 12. P. 23–41.
6. Waldinger R., Lee R. PROW: A step toward automatic program writing // Proc. of the 1<sup>st</sup> Intern. Joint Conf. on Artificial intelligence San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1969. P. 241–252.
7. Chang C., Lee R. Symbolic Logic and Mechanical Theorem Proving. New York: Academic, 1973.
8. Kant E. On the efficient synthesis of efficient programs // Artificial Intelligence. 1983. № 20 (3). P. 253–305.
9. Tyugu E. Kh., Kharf M. Ya. Algorithms for structural synthesis of programs // Программирование. 1980. № 4. P. 3–13.
10. Маслов Ю. С. Теория дедуктивных систем и ее применения. М: Радио и связь, 1986. 136 с.
11. Korukhova Yu. An approach to automatic deductive synthesis of functional programs // Annals of Mathematics and Artificial Intelligence. 2007. № 50 (3–4). P. 255–271.
12. Novoseltsev V. B. Synthesis of parallel recursive programs in structural functional models // Program. Comput. Software. 2007. № 33(5). P. 293–299.
13. Giacomo G., Patrizi F., Sardina S. Automatic behavior composition synthesis // Artificial Intelligence. 2013. № 196. P. 106–142.
14. Kreitz C. Program Synthesis. Chapter III.2.5 of Automated Deduction – A Basis for Application. Kluwer Publ. 1998. P. 105–134.
15. Avellone A., Ferrari M., Miglioli P. Synthesis of Programs in Abstract Data Types // Proc. of the 8<sup>th</sup> Intern. Workshop on Logic Programming Synthesis and Transformation. London, UK: Springer-Verlag, 1990. P. 81–100.
16. A Multi-level Approach to Program Synthesis / W. Bibel, D. Korn, C. Kreitz, F. Kurucz, J. Otten, S. Schmitt, G. Stolpmann // Lecture Notes in Computer Science. 1997. Vol. 1463. P. 1–27.
17. Fu P., Komendantskaya E. A Type – theoretic Approach to Resolution // Revised Selected Papers of the 25<sup>th</sup> Intern. Symposium on Logic-Based Program Synthesis and Transformation. Berlin, Heidelberg: Springer-Verlag, 2015. P. 91–106.
18. Shaw D., Swartout W., Green C. Inferring LISP programs from examples // In Proc. of the 4<sup>th</sup> Intern. Joint Conf. on Artificial Intelligence. Morgan Kaufmann Publishers Inc., 1975. Vol. 1. P. 260–267.
19. Summers Ph. A methodology for LISP program construction from examples // J. of the ACM. 1977. Vol. 24 (1). P. 161–175.
20. Pygmalion S. D. A Creative Programming Environment. Stanford: Stanford University, 1975.
21. Koza J. Genetic programming as a means for programming computers by natural selection // Statistics and Computing. 1994. Vol. 4 (2). P. 87–112.
22. Schkufza E., Sharma R., Aiken A. Stochastic superoptimization // Proc. of the Eighteenth Intern. Conf. on Architectural Support for Programming languages and operating systems. Houston, 2013. P. 305–316.
23. Gulwani S., Jovic N. Program verification as probabilistic inference // In Proc. of the 34<sup>th</sup> Symposium on Principles of Programming Languages. Nice, 2007. P. 277–289.
24. Combinatorial sketching for finite programs / A. Solar-Lezama, L. Tancau, R. Bodik, S. Seshia, V. Saraswat // Proc. of the 12<sup>th</sup> Intern. Conf. on Architectural support for programming languages and operating systems. New York, 2006. P. 404–415.
25. Syntax-guided synthesis / R. Alur, R. Bodik, G. Junwal, K. Martin Milo, M. Raghothaman, S. Seshia, R. Singh, A. Solar-Lezama, E. Torlak, A. Udupa // Proc. of the IEEE Intern. Conf. on FormalMethods in Computer-Aided Design (FMCAD). Portland, OR, USA, 2013. P. 1–8.
26. Manna Z., Waldinger R. Knowledge and reasoning in program synthesis // Artif. Intell. 1975. Vol. 6 (2). P. 175–208.
27. Srivastava S., Gulwani S., Foster J. From program verification to program synthesis // In Proc. of the 37<sup>th</sup> ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. Madrid, Spain, 2010. P. 313–326.
28. Solar-Lezama A. Doctoral Dissertation in Philosophy in Engineering-Electrical Engineering and Computer Science // Program synthesis by sketching. University of California at Berkeley, CA, USA, 2008. URL: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-176.html> (дата обращения 20.11.2019).
29. Path-based inductive synthesis for program inversion / S. Srivastava, S. Gulwani, S. Chaudhuri, J. Foster // In Proc. of the Conf. on Programming Language Design and Implementation. San Jose, CA, USA, 2011. P. 492–503.
30. A machine learning framework for programming by example / A. Menon, O. Tamuz, S. Gulwani, B. Lampson, A. Kalai // In Proc. of the 30<sup>th</sup> Intern. Conf. on Machine Learning. Atlanta, GA, USA, 2013. P. 187–195.

31. Polozov O. Program Synthesis in 2017–18. URL: <https://alexpolozov.com/blog/program-synthesis-2018/> (дата обращения 20.11.2019).
32. Gulwani S., Polozov O., Singh R. Program Synthesis // Foundations and Trends® in Programming Languages. 2017. Vol. 4, № 1–2. P. 1–119.
33. Accelerating Search-Based Program Synthesis using Learned Probabilistic Models / W. Lee, K. Heo, R. Alur, M. Naik // Trends in Programming Languages. 2017. Vol. 4, №. 1-2. P. 1–119.
34. Leveraging Constraint Logic Programming for Neural Guided Program Synthesis / L. Zhang, G. Rosenblatt, E. Fetaya, R. Liao, W. Byrd, R. Urtasun, R. Zemel // URL: <https://arxiv.org/abs/1809.02840> (дата обращения 20.11.2019).
35. Neural-Guided Deductive Search for Real-Time Program Synthesis from / A. Kalyan, A. Mohta, A. Polozov, D. Batra, P. Jain, S. Gulwani. URL: <https://arxiv.org/abs/1804.01186> (дата обращения 20.11.2019).
36. Polosukhin I., Skidanov A. Neural Program Search: Solving Programming Tasks from Description and Examples. URL: <https://arxiv.org/abs/1802.04335> (дата обращения 20.11.2019).
37. Feng Y., Martins R., Bastani O. I. Dillig Program Synthesis using Conflict-Driven Learning. URL: <https://arxiv.org/abs/1711.08029> (дата обращения 20.11.2019).
38. Oracle-guided component-based program synthesis / S. Jha, S. Gulwani, S. Seshia, A. Tiwari // IEEE Intern. Conf. on Software Engineering. New York, USA. 2010. Vol. 1. P. 215–224.
39. Kotseruba I., Tsotsos J. K. 40 years of cognitive architectures: core cognitive abilities and practical applications. URL: <https://link.springer.com/article/10.1007%2Fs10462-018-9646-y> (дата обращения 20.11.2019).
40. Cognitive Control / H. Simon, M. Fatemi, P. Setoodeh, Y. Xue // Proc. of the IEEE. 2012. Vol. 100. P. 3156–3169.
41. Palm G. Neural associative memories and sparse coding // Neural Networks. 2013. Vol. 37. P. 165–171.
42. Haykin S. Neural Networks and Learning Machines, 3rd ed. PrenticeHall, NewYork, 2008.
43. A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications / C. D. James, J. B. Aimone, N. E. Miner, C. M. Vineyard, F. H. Rothganger, K. D. Carlson, S. A. Mulder, T. J. Draelos, A. Faust, M. J. Marinella, J. H. Naegle, S. J. Plimpton // Biologically Inspired Cognitive Architectures. 2017. Vol. 19. P. 49–64.
44. A world survey of artificial brain projects. Part II / B. Goertzel, R. Lian, I. Arel, Hugo de Garis, S. Chen // Biologically inspired cognitive architectures. Neurocomputing. 2010. Vol. 74, № 1–3. P. 30–49.
45. Osipov V., Osipova M. Space-time signal binding in recurrent neural networks with controlled elements // Neurocomputing. 2018. Vol. 308. P. 194–204.
46. Osipov V., Nikiforov V. Formal aspects of streaming recurrent neural networks / Advances in Neural Networks // Lecture Notes in Computer Science. 2018. Vol. 10878. P. 29–36.
47. Osipov V. Space-time structure of recurrent neural networks with controlled synapses // Lecture Notes in Computer Science. 2016. Vol. 9719. P. 177–184.
48. Osipov V. Structure and basic functions of cognitive neural network machine. URL: [https://www.mateconferences.org/articles/mateconf/abs/2017/27/mateconf\\_er2017\\_02011/mateconf\\_er2017\\_02011.html](https://www.mateconferences.org/articles/mateconf/abs/2017/27/mateconf_er2017_02011/mateconf_er2017_02011.html) (дата обращения 30.10.2019).
49. Feng Y., Martins R., Bastani O., Dillig I Program synthesis using conflict-driven learning // Proc. of the 39<sup>th</sup> ACM SIGPLAN Conf. on Programming Language Design and Implementation. New York, USA, 2018. P. 420–435.
50. Programmatically Interpretable Reinforcement Learning / A. Verma, V. Murali, R. Singh, P. Kohli, S. Chaudhuri // Proc. of the 35<sup>th</sup> Intern. Conf. on Machine Learning. URL: <http://proceedings.mlr.press/v80/verma18a.html> (дата обращения 30.10.2019).

N. A. Zhukova

Saint Petersburg Electrotechnical University

## ABOUT THE POSSIBILITIES OF SYNTHESIS OF MULTILEVEL MODELS OF OBJECTS

*A new method for synthesis models of objects that have multilevel structure is considered. The proposed method develops existing approaches to inductive and deductive synthesis of formal structures. A multilevel structure of object models is defined, a description of the proposed method for inductive-deductive synthesis is given. The method assumes processing of data received from objects, construction of a deductive synthesis machine on their basis, and application of the machine for building models of objects. When constructing the machine, inductive synthesis of multilevel structures in the space of elements of the source data is performed. Based on the constructed structures, the parameters of the deductive synthesis machine are determined. Deductive synthesis of object models using the constructed machine involves formulation of a theorem on the existence of a model and its proof. The method allows build new structures to describe the objects, it has low computational complexity. An example of synthesis of a model of an object with a four-level structure is given.*

**Multilevel synthesis, state machine model, objects modeling**