

*Е. В. АЛИСЕЙКО, Н. Н. БУСЛИК, Л. Б. КЛИМЕНКО*

### **КОНТЕКСТНЫЙ МЕТОД УТОЧНЕНИЯ ИМЕН ЭЛЕМЕНТОВ В ИЕРАРХИЧЕСКИХ СТРУКТУРАХ ДАННЫХ**

Одним из распространенных способов представления данных в языках программирования и языках определения данных (ЯОД) является использование именованных иерархических структур. Последние включают в себя несколько подструктур, каждая из них также может включать в себя несколько подструктур нижнего уровня и т. д. до уровня элементов данных, в качестве которых выступают переменные, массивы, атрибуты базы данных. Каждому элементу структуры (включая собственно структуру) сопоставляется имя, служащее для идентификации данного элемента в программе при описании операций, выполняемых над ним. При этом в большинстве языков, работающих со структурами, допускается омонимия элементов, т. е. употребление одного и того же имени для обозначения нескольких элементов при объявлении структуры.

Естественно, что на этапе выполнения операций над элементами требуется однозначная их идентификация, которая традиционно осуществляется с использованием так называемых уточняющих имен: кроме собственного имени элемента в идентификатор входят имена включающих подструктур. Например, в ПЛ/1 такой идентификатор [1] имеет синтаксис (используется модифицированная форма Бэкуса — Наура):

$\langle \text{идентификатор} \rangle ::= [ \langle \text{идентификатор} \rangle ] . \langle \text{имя} \rangle$

Искомый элемент занимает крайнюю правую позицию в идентификаторе; символ «.» (точка) понимается как предикат «входит»; слева от точки — имя подструктуры (структуры), включающей искомый элемент. Таким образом, два элемента с одинаковым именем могут быть однозначно идентифицированы, если они входят в подструктуры, имеющие различные имена.

Другим эквивалентным способом идентификации является групповое уточнение. Например, в ЯОД СИНТА [2] используются конструкции со следующим синтаксисом:

<именованная группа> ::= <идентификатор> (<набор идентификаторов>)  
 <набор идентификаторов> ::= <идентификатор> [, <набор идентификаторов>]

Групповое уточнение позволяет сделать запись идентификатора наглядной и более лаконичной в случае, когда в операции одновременно участвует несколько элементов структурных данных. Эквивалентность обоих способов уточнения определяется правилом «раскрытия скобок»:

$$A(B, C) = A.B, A.C$$

Здесь A, B, C — идентификаторы; символ «=» понимается как «эквивалентно».

Различные версии языков, допускающих омонимию элементов структуры данных, позволяют в зависимости от семантической мощности транслятора проводить полное или частичное уточнение: в первом случае в идентификаторе перечисляются имена всех включающих подструктур, во втором достаточно перечислить имена, однозначно идентифицирующие искомый элемент. В обоих случаях для рассмотренного метода уточнения «сверху» основным требованием к объявляемой структуре остается следующее: *всякие два элемента структуры, непосредственно входящие в одну и ту же подструктуру, должны иметь различные имена.*

В языках программирования, где явление омонимии вводится для удобства программиста, указанное ограничение можно считать приемлемым. В то же время для структур данных с которыми работают пользователи ЯОД, выдержать такое ограничение удастся не всегда. В ЯОД структура, как правило отражает традиционно сложившиеся формы «ручных» документов, для которых характерным является омонимия реквизитов. Например, в одном документе могут иметь место подструктуры (таблица):

Сдано			. . .	Сдано		
По цеху всего	В том числе			По участку всего	В том числе	
	готовых изделий	комп- лектую- щих			готовых изделий	комп- лектую- щих

1) СДАНО (ПО-ЦЕХУ-ВСЕГО, В-ТОМ-ЧИСЛЕ (ГОТОВЫХ-ИЗДЕЛИЙ, КОМПЛЕКТУЮЩИХ));

2) СДАНО (ПО-УЧАСТКУ-ВСЕГО, В-ТОМ-ЧИСЛЕ (ГОТОВЫХ-ИЗДЕЛИЙ, КОМПЛЕКТУЮЩИХ)).

Традиционные методы уточнения в данном случае не позволя-

ют однозначно идентифицировать элементы с именами: ГОТОВЫХ-ИЗДЕЛИЙ, КОМПЛЕКТУЮЩИХ, В-ТОМ-ЧИСЛЕ, СДАНО.

Мы ставим задачу расширить класс допустимых омонимических структур без усложнения синтаксиса и семантики конструкций ЯОД. Конкретизируя последнее требование, формулируем следующие ограничения на модификацию ЯОД: 1) лексика языка не расширяется; 2) правила построения идентификаторов не меняются или меняются незначительно; 3) вновь вводимые предикаты на именах элементов структуры являются такими же элементарными, как предикат «входит».

Отметим, что условию 3 удовлетворяет предикат «включает» (по сути, «входит» и «включает» — один и тот же предикат с перестановкой переменных). Основная идея решения поставленной задачи заключается в использовании в идентификаторах обоих предикатов одновременно. В самом деле, всякая подструктура может быть уточнена именами не только включающих, но и включаемых подструктур (элементов данных). Введем следующие правила построения идентификаторов:

<идентификатор> ::= <идентификатор> [ <структурное имя> ]

<структурное имя> ::= <имя> [( <структурное имя> ) ]  
Здесь предикат «включает» обозначается круглыми скобками, предикат «входит», как в традиционных способах — точкой.

Расширяет ли предложенное нововведение класс допустимых именованных структур? Ответ положительный: это видно хотя бы из приведенного выше примера документа, для которого 2 реквизита КОМПЛЕКТУЮЩИХ теперь могут быть идентифицированы однозначно: СДАНО (ПО-ЦЕХУ-ВСЕГО). КОМПЛЕКТУЮЩИХ и СДАНО (ПО-УЧАСТКУ-ВСЕГО). КОМПЛЕКТУЮЩИХ. С одной стороны, возникает вопрос, существуют ли ограничения на объявляемую структуру при предлагаемом способе уточнения (который назовем контекстным уточнением)? Ответ также положительный. Легко показать, что в этом случае именованная структура должна удовлетворять следующему ограничению: *в любой подструктуре не существует двух или более одинаково именуемых полных путей*. Под именованнием пути понимается упорядоченная последовательность имен элементов структуры, принадлежащих данному пути. Сформулированное ограничение на употребление омонимов при объявлении структуры существенно слабее ограничения, накладываемого методами уточнения «сверху».

Используя комбинацию символов точки и круглых скобок, т. е. не расширяя лексику языка, введем правило группового уточнения:

$\langle \text{именованная группа} \rangle ::= \langle \text{идентификатор} \rangle ; (\langle \text{набор идентификаторов} \rangle)$

Возвращаясь к примеру, для группы реквизитов ГОТОВЫХ-ИЗДЕЛИЙ, КОМПЛЕКТУЮЩИХ идентификаторы для 1-й и 2-й подструктур СДАНО имеют вид:

1) СДАНО (ПО-ЦЕХУ-ВСЕГО). (ГОТОВЫХ-ИЗДЕЛИИ, КОМПЛЕКТУЮЩИХ)

2) СДАНО (ПО-УЧАСТКУ-ВСЕГО). (ГОТОВЫХ-ИЗДЕЛИИ, КОМПЛЕКТУЮЩИХ).

При определенных соглашениях в ЯОД имя группы может уточняться именами входящих в группу элементов. Так, запись СДАНО (ПО-ЦЕХУ-ВСЕГО, КОМПЛЕКТУЮЩИХ)

при этих соглашениях может считаться эквивалентной записи СДАНО (ПО-ЦЕХУ-ВСЕГО). ПО-ЦЕХУ-ВСЕГО, СДАНО (ПО-ЦЕХУ-ВСЕГО). КОМПЛЕКТУЮЩИХ.

Таким образом, в нашем примере удастся однозначно идентифицировать реквизит КОМПЛЕКТУЮЩИХ по контексту.

Контекстное уточнение элементов структуры данных практически не усложняет конструкции ЯОД, позволяя в то же время существенно расширить класс допустимых именованных структур с омонимией элементов. Анализ алгоритмов интерпретации новых конструкций ЯОД показывает, что методы синтаксического разбора, ориентированные на грамматику предшествования и применявшиеся в ЯОД СИНТА [2], остаются приемлемыми. При этом алгоритм лексического разбора дополняется функцией распознавания символов «.» как одного терминального символа группового уточнения. Наибольшие изменения претерпевают алгоритмы семантического разбора именованных групп и идентификаторов, т. е. собственно алгоритм распознавания искомых элементов структуры. При использовании контекстного уточнения алгоритм распознавания состоит в поиске подграфа, а не в поиске пути графа, как в традиционных методах трансляции. Определение эффективности того или иного алгоритма распознавания искомых элементов структуры представляет материал для отдельного исследования.

**Список литературы:** 1. Скотт Р., Сондак Н. ПЛ/1 для программистов.— М.: Статистика, 1977.—223 с. 2. Буслик Н. Н., Дедиков Э. А., Ченцов В. Ф. Определение данных в системе управления базой данных СИНТА // АСУ приборы автоматки.— 1982.— Вып. 62.— С. 52—59.

Поступила в редколлегию 21.02.84