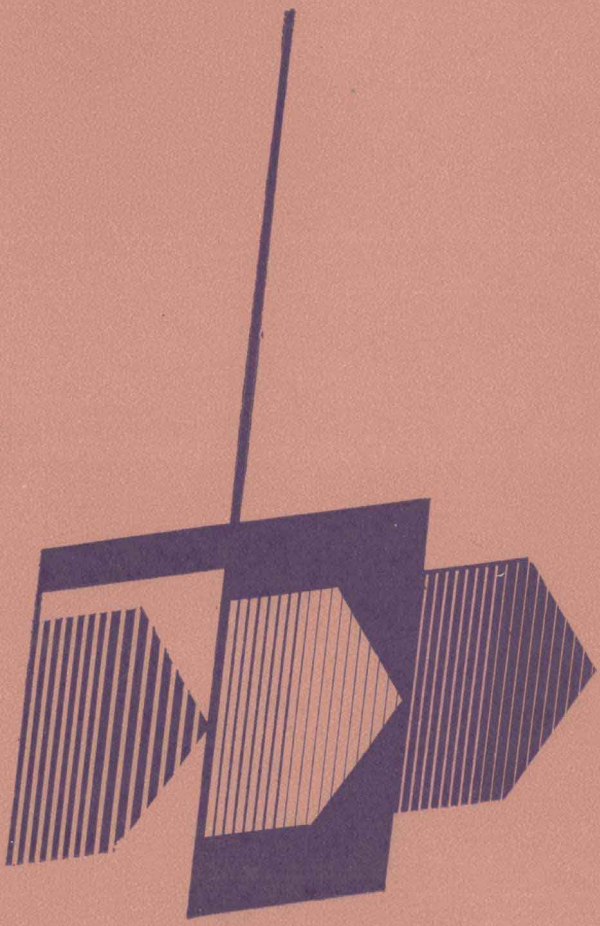
ТЕЛЕОБРАБОТКА ПЛАНОВО-ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ



ГОСПЛАН МОЛДАВСКОЙ ССР
Научно-исследовательский институт планирования

ТЕЛЕОБРАБОТКА ПЛАНОВО— ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ

КИШИНЕВ „ШТИИНЦА" 1985

В сборнике рассматриваются вопросы развития и использования фактографических информационных систем, алгоритм автоматического формирования запросов и метод синтеза программ расчета значений агрегативных показателей для задач АСУ. Приводятся модели прогнозирования, планирования и имитации развития народного хозяйства в интерактивном режиме, модели центрального вычислительного комплекса ВЦКП двухуровневой иерархической структуры.

Книга рассчитана на работников органов управления, специалистов, занимающихся разработкой и эксплуатацией автоматизированных систем, на аспирантов и студентов соответствующих специальностей.

С.К.Корня

ИНФОРМАЦИОННАЯ СИСТЕМА С МНОГОЯЗЫКОВЫМ СЕМАНТИЧЕСКИМ УРОВНЕМ

Тенденция построения баз данных с многоуровневой архитектурой, наметившаяся в последние годы в нашей стране и за рубежом,полностью оправдала себя. Сегодня имеется множество информационно-вычислительных систем с надстройкой в виде семантического уровня, использующих формализованные, близкие к естественному языки описания данных. Языки описания, а также языки общения (запросов) с базами данных, созданные на их основе, являются удобным средством для работы конечного пользователя. В большинстве случаев языки создаются путем формализации естественного русского языка и используются в различных информационных системах, функционирующих в СССР. Однако всестороннее развитие процессов экономической интеграции в рамках СЭВ, а также различных видов интернациональных взаимосвязей (торговля, научно-техническое сотрудничество и др.) ставят на повестку дня их автоматизацию. Предлагаемая работа представляет собой результат эксперимента по созданию средств автоматизации упомянутых процессов. Основой системы обработки информации подобного типа является проблемно-ориентированная база данных. Пользователями такой базы могут выступать представители различных государств. Следовательно, одним из обязательных требований к подобным системам должно быть обеспечение возможности идентичного семантического описания одних и тех же данных на языке того государства, представителем которого является пользователь. Очевидно, что создание таких средств сводится к многоязыковому семантическому описанию данных. Эксперимент проводился на информационной системе СПИРИТ, логическая схема организации которой дана в [I]. База данных организована разработанным нами программно-ассоциативным методом и представлена семантической, ассоциативной и собственной областями. При создании многоязыковых средств представления и описания данных ассоциативная и собственная области не претерпевают никаких изменений. Модификации подвергается лишь семантическая область, в частности словари наименований показателей и значений признаков (описательных терминов).

Словарь наименований показателей $S=\{s_l; l=\overline{1,l_0}\}$ представляется множеством $S=\{s_l^e; l=\overline{1,l_0}; e=\overline{1,e_0}\}$, где s - конкретное наименование отдельной позиции словаря; l - шифр наименования показателя; e - номер языка, на котором дается формализованная позиция наименования показателя или описательного термина.

Аналогично представляется словарь описательных терминов

$$Z=\{z_\beta^e; \beta=\overline{1,\beta_0}; e=\overline{1,e_0}\},$$

где z - конкретное наименование описательного термина отдельной позиции словаря; β - шифр описательного термина.

В общем случае $S=\bigcup_e s_i^e$; $Z=\bigcup_e z_\beta^e$.

Остальные моменты формального представления языка описания данных, равно как и логические процедуры вычислительного процесса обработки данных, достаточно полно описанные в [1; 2], остаются без изменения.

Эксперименты проводились на восьми языках (иностранных и народов СССР): русском, английском, французском, немецком, испанском, румынском, украинском и молдавском.

Процедуры обработки осуществляются по принципу многоаспектного избирательного поиска. Исходным является код запроса. Основные элементы кода следующие: номер класса показателей; позиции горизонтальной и вертикальной разверток; номера признаков; количество разворачиваемых позиций; начальная позиция развертки;

по аспектным признакам: номера признаков; количество разворачиваемых позиций; конкретные позиции на выбор;

по дополнительным признакам: номера признаков; конкретные позиции развертки;

по языковому признаку: общее количество языков, на которых должно выдаваться семантическое сопровождение результатов поиска; позиции конкретных языков при выборочном запросе.

Алгоритм поиска.

1. Начало.

2. Ввод кодового запроса.

3. Определение номера набора признаков искомого класса показателей.

4. Определение начальных позиций описательных терминов по всем признакам набора.

5. Селекция терминов требуемого языка описания выходных результатов.

6. Селекция строк ассоциативных матриц, соответствующих описательным терминам.

7. Определение адреса оснований показателей.

8. Селекция оснований по определенным адресам.

9. Формирование выходных результатов.

10. Конец.

В разработанной программе, реализующей приведенный укрупненный алгоритм, учтены лишь основные моменты многоаспектной селекции, наиболее приемлемые для справочного режима. Простота алгоритма и структуры организации базы данных позволяет реализовать множество версий, где могут быть учтены и другие, более детальные требования пользователей. Аналогичным образом разрабатываются алгоритмы обработки и формирования более сложных выходных структур.

Следует отметить некоторые затруднения при реализации системы. Почти каждый язык имеет свои алфавитные особенности, например, немецкий - $\ddot{u}, \ddot{a}$, румынский - $\hat{\imath}, \c{s}, \c{t}, \breve{a}$, молдавский - ӂ и др. Для данных символов не предусмотрены соответствующие коды, и аппаратура не может ни считывать их при вводе, ни печатать при выводе результатов на АЦПУ или на экран видеотерминала. Представляется целесообразным более глубокое исследование особенностей алфавитов различных языков и учет их как при производстве аппаратурных вычислительных средств, так и при разработке программного обеспечения вычислительных систем.

Для иллюстрации получаемых результатов в терминах различных языков приведем описание единичного показателя на формализованных языках.

На русском языке:

ЗАТРАТЫ / СОЗДАНИЕ И ВНЕДРЕНИЕ АСУ И ВТ / МИНИСТЕРСТВО ЛЕГКОЙ ПРОМЫШЛЕННОСТИ / 1982 ГОД / ФАКТ/ТЫСЯЧ РУБЛЕЙ / - R_1.

(R_1 - числовое основание искомого показателя).

Дробной чертой отделены друг от друга наименования показателя и описательные термины. В данном случае языком описания является язык с ограниченными грамматическими средствами, но пользователь легко преодолевает эти ограничения путем самостоятельного добавления недостающих союзов и необходимых изменений грамматических форм. Например, в описании показателя R_1 пользователь легко определит, что это затраты на создание АСУ и ВТ по министерству легкой промышленности за 1982 год, фактически в тысячах рублей.

На английском языке:

EXPENDITURES/CREATION AND INCULCATION OF ACS AND CALCULATION MACHINERY/MINISTRY OF LIGHT INDUSTRY/1982 YEAR/ FACT/THOUSAND ROUBLES/ - R_1.

На французском языке:

LES DEPENSES/LA CREATION ET L'INCULCATION DU SAC ET DE LA TECHN.A CALC./LE MINISTERE DE L'INDUSTRIE LEGERE/L'ANNEE 1982/LE FAIT/MILLIES RUBLES/ - R_1.

На немецком языке:

AUSGABEN/SCHAFFUNG UND EINFUHRUNG DER ASS UND DER RECHENTECHNIK/MINISTERIUM DER LEICHTINDUSTRIE/1982 JAHR/FAKT/TAUSAND RUBEL/ - R_1.

На испанском языке:

GASTOS/CREACION E INTRODUCION SAD Y TECNICA DE COMPUTACION/MINISTERIO DE LA INDUSTRIA LIVIANA/ANO 1982/FACTO/MIL RUBLOS/ - R_1.

На румынском языке:

CHELTUELI/CREAREA SI APLICAREA IN PRACTICA A SAC SI TEHNIC.DE CALCUL/MINISTERUL INDUSTRIEI USOARE/ANUL 1982/REAL/MII RUBLE/ - R_1.

На украинском языке:

ЗАТРАТИ/ СТВОРЕНИЯ І ВПРОВАДЖЕНЯ АСУ І ВТ/ МІНІСТЕРСТВО ЛЕГКОІ ПРОМИСЛОВОСТІ/ 1982 РІК/ФАКТ/ ТИС.КАРБ./ - R_1.

На молдавском языке:

КЕЛТУЕЛЬ/КРЕАРЯ ШИ АПЛИКАРЯ ЫН ПРАКТИКЭ А САК ШИ ТЕХНИЧИЙ ДЕ КАЛКУЛ/ МИНИСТЕРУЛ ИНДУСТРИЕЙ УШОАРЕ/ АНУЛ 1982/ РЕАЛ/ МИЙ РУБЛЕ/ - R_1.

Программная реализация позволяет осуществлять многоаспектное информационное обслуживание на любом языке или наборе языков на выбор как в пакетном режиме, так и в режиме теледоступа.

Литература

1. Корня С. К. Инструментарий построения безызбыточных баз данных. - В кн.: Информационные языки и базы данных в планировании. М., ЦЭМИ АН СССР, 1980.

2. Корня С. К. Информационная совместимость взаимодействующих автоматизированных систем. - В кн.: Обработка информации в АСУ-Молдавия. Кишинев, Штиинца, 1980.

В.П.Недельчук, В.С.Гыля

АЛГОРИТМ СИНТЕЗА ЗАПРОСОВ К СИСТЕМЕ ОБРАБОТКИ ДАННЫХ

В практике планирования и управления значительное число задач заключается в том, чтобы определить и в форме таблиц представить

численные значения показателей, характеризующих объекты управления, т.е. сводится или может быть сведено к заполнению заданных таблиц (форм) T числами, полученными в результате обработки другой информации или их поиска в различных фондах. Такое представление показателей привычно для широкого круга специалистов, и потому большинство задач, решаемых сегодня в рамках АСУ, является задачами автоматизации процесса получения именно такого рода таблиц.

Анализ структур различных форм представления показателей свидетельствует, что подлежащие заполнению таблицы T, как правило, состоят из следующих компонентов:

надтабличная текстовая часть A;

табличная текстовая часть B;

собственно числовая часть C.

Надтабличная текстовая часть A в общем случае содержит наименование таблицы (A_1), дополняющую информацию (A_2) и вспомогательную информацию (A_3). Содержание A_1 и/или A_2 иногда необходимо для семантического однозначного описания элементов-клеток (чисел) части C, причем A_1 задает наименование однородной совокупности показателей, а A_2 – конкретизирующие общие признаки данной совокупности (например, единицу измерения, рассматриваемый период и пр.); A_3 содержит номер таблицы, наименование органа, утвердившего ее, и тому подобную информацию, которая с точки зрения информативности алгоритма определения показателей T является лишней. Зачастую, однако, табличная текстовая часть B, заданная в виде вертикальных (B_1) и/или горизонтальных (B_2) граф, подграф (в дальнейшем – доменов), семантически однозначно описывает содержание всех элементов (чисел) C.

Таким образом, T можно представить в виде

$$T = A \cup B \cup C, \qquad (1)$$

тогда как на практике при решении конкретных задач таблицы T состоят из сочетаний (компонентов):

$$T = A_1 \cup A_2 \cup B \cup C, \qquad (2)$$

$$T = A_{1(2)} \cup B \cup C, \qquad (3)$$

$$T = A_{1(2)} \cup B_{1(2)} \cup C, \qquad (4)$$

$$T = B \cup C, \qquad (5)$$

$$T = B_{1(2)} \cup C, \qquad (6)$$

где $1(2)$ означает возможность наличия одного из индексов при A и B.

Если $B_{1(2)}$ в (6) соответствует одному домену, имеем одностолбцовую (однострочную) таблицу, где содержимое $B_{1(2)}$ описывает

запрос на множество показателей; если же содержимое $B_{1(2)}$ описывает запрос на один показатель, то имеем случай таблицы размером IxI (бестабличный запрос).

Нетрудно заметить, что (2) - (4) могут быть сведены к (5), так как содержимое $A_1 \cup A_2$ можно отнести к одному верхнему домену части B_1, ширина которого равна ширине B_1. В связи с этим все рассуждения приводятся для таблицы, заданной формулой (5).

Несмотря на успехи, достигнутые в области создания баз данных (БД) и средств теледоступа, в АСУ еще доминирует позадачный подход, когда для каждой Т составляются специальные программы, определяющие и при необходимости обрабатывающие первичные показатели с целью вычисления значений агрегируемых показателей заданной Т. При этом роль БД сводится к хранению и выдаче значений первичных показателей, а роль средств теледоступа - к ускорению процесса подготовки и ввода-вывода программ и данных.

Жесткая зависимость программ решения задач АСУ от структуры Т и содержания $B \subset T$, а отсюда - необходимость привлечения специалиста программиста при всевозможных изменениях Т, значительный коэффициент дублирования программных средств и памяти ЭВМ, привязка пользователей системы обработки данных (СОД) к фиксированному набору таблиц представления задач являются причинами, сдерживающими внедрение диалоговых методов решения задач непосредственно пользователями непрограммистами.

В последние годы предложены новые подходы к решению задачи автоматизации процесса заполнения Т. Так, в [1] рассматриваются дополнительные средства к БД фактографического типа, построенной согласно [2], которые позволяют пользователям непрограммистам заданием одних только запросов на формализованном языке, близком к естественному, формируемых ими на базе $B \subset T$, получить значение первичных и агрегируемых показателей данной предметной области.

Решение задач в режиме взаимодействия пользователей с СОД, организованными в соответствии с [1; 2], оставляет за пользователями право на выполнение процедуры "расшивки" требуемых выходных таблиц (ТВТ) с целью составления из $B \subset T$ запросов $\mathcal{L}$ на формализованном языке, которые однозначно описывают все показатели, содержащиеся в Т. Это усложняет процесс формирования и ввода запросов (т.е. задач) в ЭВМ как при пакетной, так и при диалоговой обработке данных. К тому же процедуры формирования и вывода ТВТ сильно усложняют программы обработки Т.

Таким образом, возникает необходимость разработки средств автоматического преобразования $(B \subset T) \rightarrow \mathcal{L}$, в результате чего стало бы возможным инвариантное по отношению к Т решение задач извест-

ными средствами с одновременным исключением процедур формирования выходных таблиц.

Постановка задачи

На практике используемые таблицы часто содержат в доменах B лишние подграфы, вспомогательную информацию в виде аббревиатур, комментариев и т.п. Легко заметить, что все таблицы могут быть приведены к нормализованному виду.

Определение I. ТВТ называется нормализованной, если в ней: а) отсутствуют домены $D_k \in B_{1(2)}$ одинаковой ширины (высоты), охваченные общими вертикальными (горизонтальными) линиями; б) каждый домен содержит только комбинации элементов заданного тезауруса рассматриваемой предметной области.

Постановка задачи I. На формализованном языке, близком к естественному, задан тезаурус рассматриваемой предметной области [I], в котором выделяются множества наименований подлежащих S, признаков X и значений признаков Z показателей, элементы которых достаточны для однозначного описания всего множества L первичных и агрегируемых показателей.

Пользователи задают незаполненные нормализованные ТВТ всевозможных конфигураций T, также, где каждый запрос, формируемый содержимым соответствующих доменов частей $B_{1(2)}$, содержит не более одного элемента множества S.

Необходимо установить алгоритм, предусматривающий автоматическое преобразование $(B \subset T) \rightarrow L$, в результате чего станет возможным преобразование $B \rightarrow C$ без участия пользователя СОД.

Постановка задачи 2. Заданы условия задачи I с тем лишь отличием, что ТВТ заполнены значениями первичных показателей.

Необходимо установить алгоритм, предусматривающий автоматическое преобразование $B \rightarrow L$ и запись чисел множества C в БД без участия пользователя СОД.

Решение задачи 2 сводится в основном к решению задачи I, но после этапа синтеза запросов не выполняется процедура нахождения значений показателей, а производится запись в БД значений соответствующих показателей.

Описывая алгоритм, будем исходить из того, что значения первичных (неагрегируемых) показателей хранятся в БД фактографического типа, построенной согласно [2] или [4]. В целом же сам алгоритм не зависит от типа используемой БД.

Определения и условные обозначения

Прежде чем приступить к рассмотрению самого алгоритма, введем некоторые новые понятия и обозначения.

Для просмотра и анализа Т с целью формирования запросов (предложений), однозначно описывающих показатели $C \subset T$, воспользуемся известным из теории распознавания зрительных образов способом сканирования входного образа (в данном случае - нормализованной ТВТ) и некоторыми положениями работы [3]. Причем, можно применять как построчную, так и контурную развертку (т.е. развертку по доменам).

Вначале рассмотрим алгоритм синтеза запросов с использованием построчной развертки Т, а затем отметим особенности синтеза запросов для случая контурной развертки.

Введем обозначения:

T_{ij} - заданная ТВТ (вся совокупность входных символов таблицы), где i, j - соответственно номера строк и столбцов развертки ТВТ;

k, p - порядковый номер доменов в процессе сканирования T_{ij} слева направо: D_k соответствует доменам i-й (текущей) строки, а D_p - доменам $(i-1)$-й (предыдущей) строки;

h - номер вертикальной линии, пересекающей текущую сканируемую строку;

n_k, u_k - значения j, соответствующие началу и концу текста k-го домена текущей строки;

$y(2k-1)$, $y(2k)$ и $\hat{y}(2p-1)$, $\hat{y}(2p)$ - значения j, указывающие границы доменов соответственно по текущей и предыдущей строкам (в дальнейшем - вход и выход доменов);

$\varphi_k(\varphi_p)$ - признак наличия (1) или отсутствия (0) горизонтальной линии в k-м (p-м) домене текущей (предыдущей) строки;

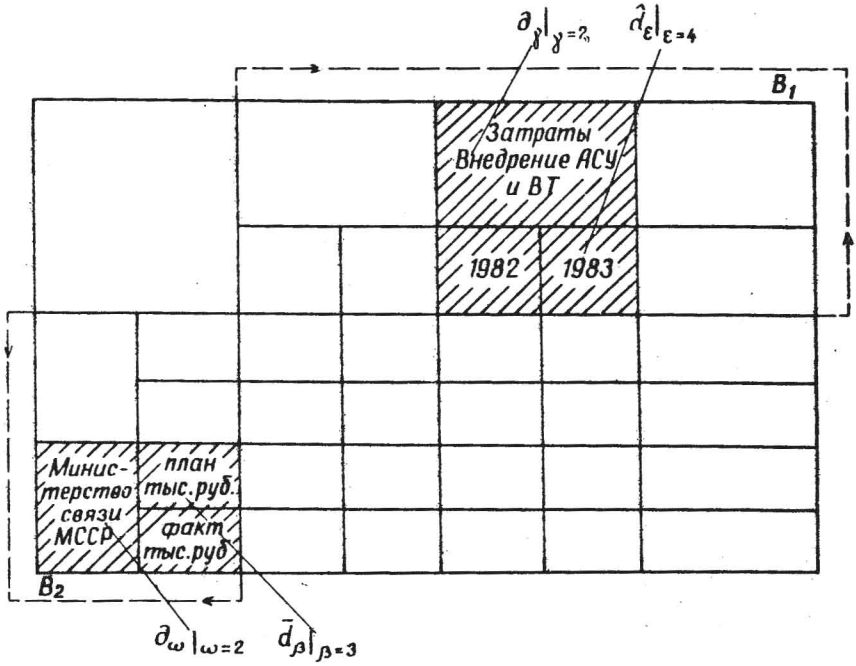
F_k - текст k-го домена текущей строки; после обработки информации текущей строки - текст k-го домена по i-ю строку включительно;

$\hat{F}_p$ - текст p-го домена по $(i-1)$-ю строку включительно;

w - значение j, соответствующее левой границе $C \subset T$;

$$w = y(2k-1)\Big|_{\substack{k=1 \\ D_k \in B_1}}.$$

Для доменов D_k и D_p введем понятие перекрытия, определяемое как $\sigma_{kp} = sign\left[y(2k) - \hat{y}(2p-1)\right] \cdot sign\left[\hat{y}(2p) - y(2k-1)\right]$. При этом, если $\sigma_{kp} = 1$, следует говорить о наличии перекрытия между доменами k и p и их содержимое относить к одному и тому же запросу (предложению); если $\sigma_{kp} \neq 1$, то домены k и p не имеют пе-



Р и с. I. Пример нормализованной ТВТ

рекрытия, и их содержимое относится к разным запросам.

Определение 2. Домен $D_k \in B$, внутри которого $\varphi_k \neq 1$, назовем доменом I-го рода (d_k).

Определение 3. Домены I-го рода, граничащие с $C \subset T$, назовем базовыми вертикальными $\partial_\varepsilon \in B_1$ и базовыми горизонтальными $\bar{d}_\beta \in B_2$.

Определение 4. Доменом 2-го рода $\partial_\gamma (\partial_\omega)$ назовем составной домен, образованный одним из крайних сверху (слева) доменов I-го рода части $B_{I(2)}$ и множеством всех доменов I-го рода, которыми он перекрывается.

Домен d_k, имеющий высоту (ширину), равную высоте (ширине) части $B_1 \subset T$ ($B_2 \subset T$), также отнесем к доменам 2-го рода.

Значения индексов $k, p, \gamma, \varepsilon$ растут слева направо, а β, ω – сверху вниз (рис.I).

Исходя из сформулированных определений, количество запрашиваемых показателей ТВТ m определяем произведением числа базовых вертикальных ε_0 на число базовых горизонтальных β_0 доменов: $m = \varepsilon_0 \cdot \beta_0$.

Алгоритм синтеза запросов по заданной ТВТ

При описании рассматриваемого алгоритма приняты следующие допущения, которые не носят принципиального характера, но упрощают его машинную реализацию: а) между терминами (словами) – элементами множеств S, X, Z, расположенными в одной строке, разрешается только один пробел; б) наименования подлежащих (т.е.элементы множества S) в заданной ТВТ включаются либо в крайние сверху домены I-го рода части B_1, либо в крайние слева домены I-го рода части B_2.

Алгоритм основан на следующем утверждении: для однозначного определения содержащихся в ТВТ запросов достаточно один раз сканировать $B_{1(2)}$ сверху вниз и слева направо и анализировать выделяемые в процессе сканирования значения установленных признаков только для i-й и $(i-1)$-й строк, а также содержимое просмотренных горизонтальных базовых доменов $\bar{d}_\beta$ заданного сканируемого домена 2-го рода для $\partial_\omega \in B_2$.

Необходимость хранения содержимого просмотренных горизонтальных базовых доменов заданного сканируемого домена 2-го рода объясняется неодинаковым размещением доменов частей $B_{1(2)}$ таблицы ($D_k \in B_1$ размещены вертикально, а $D_k \in B_2$ – горизонтально). Если текстовую информацию $D_k \in B_2$ разместить по столбцам (снизу вверх), как это делается в ряде случаев, то необходимость в хранении такой информации отпадает: остается лишь поменять местами значения параметров i и j при анализе частей $B_{1(2)}$ таблицы.

В реализованной версии данного алгоритма нахождение значений ранее введенных параметров осуществляется путем обработки литерных строк $B \subset T$ согласно следующим соотношениям:

$$h = \begin{cases} h+1, & \text{если } T_{i,j} = T_{i-1,j} \vee T_{i+1,j} = \text{„|“} \\ h & \text{в противном случае,} \end{cases};$$

$$\varphi_k = \begin{cases} 1, & \text{если } T_{i,j} = T_{i,j+1} \vee T_{i,j-1} = \text{„—“} \\ 0 & \text{в противном случае} \end{cases};$$

$$k = \begin{cases} k+1, & \text{если } T_{i,j} = \text{„|“}, h>1|_{w\neq 0}, \\ & \text{или } T_{i,j} = \text{„|“}, T_{i,j-1} \neq \text{„|“} \\ k & \text{в противном случае} \end{cases};$$

$$n_k = \begin{cases} j, & \text{если } T_{i,j-2} = T_{i,j-1} = \text{„␣“}, T_{i,j} \neq \text{„␣“} \vee \text{„|“} \vee \text{„—“} \\ & \text{или } T_{i,j-1} = \text{„|“}, T_{i,j} \neq \text{„␣“} \vee \text{„—“} \\ n_k & \text{в противном случае} \end{cases};$$

$$u_k = \begin{cases} j, \text{ если } T_{i,j+1} = T_{i,j+2} = \text{„␣“}, T_{ij} \neq \text{„␣“} \vee \text{„|“} \vee \text{„—“} \\ \quad \text{или } T_{i,j} \neq \text{„␣“} \vee \text{„|“} \vee \text{„—“}, T_{i,j+1} = \text{„|“} \\ u_k \quad \text{в противном случае} \end{cases} ;$$

$$y(2k-1) = \begin{cases} j, \text{ если } y(2k-1)=0, k\big|_{T_{i,j-1}} = a, k\big|_{T_{ij}} = a+1 \\ 0 \quad \text{в противном случае} \end{cases} ;$$

$$y(2k) = \begin{cases} j, \text{ если } y(2k-1)\neq 0, k\big|_{T_{i,j-1}} = a, k\big|_{T_{ij}} = a+1 \\ 0 \quad \text{в противном случае} \end{cases} ;$$

$$w = y(2k-1)\Big|_{\substack{k=1 \\ h=2}} = y(1);$$

$$F_k = F_k \parallel T_{ij} \quad ,$$
$$j=\overline{n_k, u_k}$$

где „|“, „—“ – элементы соответственно вертикальных и горизонтальных линий; „␣“ – пробелы; $\vee$, $\parallel$ – операции соответственно дизъюнкции (ИЛИ) и сцепления.

Алгоритм синтеза запросов

I. Ввод $T_{ij}\ (i=\overline{1,i_0}, j=\overline{1,j_0})$. Установление исходных значений рассмотренных ранее параметров.

2. Выделение и анализ признаков доменов $D_k \in B_1$.

2.1. $i=i+1$, $j=j+1$. Поиск первого элемента $T_{ij} \neq$ „␣“. Если найденный элемент $T_{ij} =$ „|“ $\vee$ „—“, переход к п.2.3.

2.2. Случай $A_{1(2)} \neq \varnothing$. $\hat{F}_1 = \hat{F}_1 \parallel T_{ij}$, $T_{ij} \in A_{1(2)}$. В процессе сцепления элементов T_{ij} исключаются двойные пробелы. Переход к п.2.I.

2.3. $i=i+1$, $j=1$. Сканируется строка i, пока $T_{ij}\big|_{h=1} =$ „|“. Если $T_{i,j+1} = T_{i,j+2} =$ „—“ & $w \neq 0$, переход к п.2.7.

2.4. Находится $T_{ij}\big|_{h=2} =$ „|“. Далее в процессе сканирования i-й строки фиксируются значения $y(2k-1)$, $y(2k)$, n_k, u_k, φ_k и F_k для всех k. При этом, если $w=0$, то $w=y(1)$.

2.5. $\forall k$ и p определяются σ_{kp}. Если $\sigma_{kp}=1$, то $F_k = \hat{F}_p \parallel F_k$.

2.6. Выполняются действия

$$\left.\begin{array}{c} \hat{y}(2a-1)=y(2a-1), \hat{y}(2a)=y(2a), \hat{\varphi}_a=\varphi_a, \hat{F}_a=F_a \\ y(2a-1)=y(2a)=\varphi_a=0; F_a=\text{“} \end{array}\right\} a=\overline{1,k}$$
$$p=k,\ k=0.$$ Переход к п.2.3.

2.7. $e=p$.

$$\left.\begin{array}{c} \varphi_p=\hat{F}_p, \hat{F}_p=\text{“} \\ y(2p-1)=y(2p)=\varphi_p=0 \end{array}\right\} p=\overline{1,e} \quad ,$$
$$p=0$$

где $\Phi_p (p=\overline{1,\varepsilon})$ - приведенное содержимое базовых вертикальных доменов; " - строка нулевой длины для переменных строкового типа.

3. Выделение и анализ признаков доменов $D_k \in B_2$.

3.1. $i=i+1$, $j=1$. Находится $T_{ij}|_{h=1} = "|"$. Если $T_{ij} \neq "|"$ для $j=\overline{1,j_0}$, переход к п.4.

3.2. Сканируется строка до $j=\omega$, фиксируя при этом значения $y(2k-1)$, $y(2k)$, n_k, u_k, φ_k и F_k для всех выделенных k.

3.3. $\forall k$ и p определяются σ_{kp}. Если $\sigma_{kp}=1$, то $F_k = \hat{F}_p \| F_k$.

3.4. Анализируется значение признаков домена, для которого $y(2k)=\omega$. Если $\varphi_k \neq 1$, переход к п.3.8.

3.5. $\beta=\beta+1$,

$\lambda_\beta = k-1$,

$G_\beta = F_k$,

$$q=\begin{cases}\beta, & \text{если } q=0\\ q & \text{в противном случае}\end{cases};$$

где λ_β - количество доменов, расположенных левее k-го домена $D_k \in B_2$; G_β - содержимое β-го горизонтального базового домена; q - номер первого горизонтального базового домена рассматриваемого домена 2-го рода.

3.6. Если $\prod_{t=1}^{k} \varphi_t \neq 1$, то $F_k = ", k=k-1$. Переход к п.3.8.

3.7. $G_r = \left\{ \|F_t \right\}_{t=\overline{1,\lambda_r}} \| G_r$, $r=\overline{q,\beta}$;

$$\left.\begin{array}{l} y(2a-1)=\hat{y}(2a-1)=\hat{y}(2a)=y(2a)=\varphi_a=0 \\ \hat{F}_a = F_a = " \\ p=\hat{k}=0 \\ q=\beta+1. \end{array}\right\} a=\overline{1,k}$$

Переход к п.3.1.

3.8.
$$\left.\begin{array}{l} \hat{y}(2a-1)=y(2a-1) \\ \hat{y}(2a)=y(2a) \\ \hat{\varphi}_a=\varphi_a \\ \hat{F}_a=F_a \\ y(2a-1)=y(2a)=\varphi_a=0 \\ F_a=" \end{array}\right\} a=\overline{1,k}$$

3.9. Переход к п.3.1.

4. Формирование результирующих запросов ТВТ:

$$L_{ij}=\begin{cases}\Phi_j \| G_i, & \text{если } s_i \in S \text{ включены в } B_1 \\ G_i \| \Phi_j, & \text{если } s_i \in S \text{ включены в } B_2\end{cases},$$

где $i=\overline{1,\beta}$, $j=\overline{1,\varepsilon}$; $\mathcal{L}_{ij}$ - результирующее семантическое описание (i,j)-го показателя (запроса) ТВТ.

Нахождение значений c_{ij} (в БД или в результате обработки значений других показателей) требует, однако, перехода от семантического к кодовому описанию $\mathcal{L}_{ij} \in \mathcal{L}$. Это предполагает разбиение предложения $\mathcal{L}_{ij}$ на соответствующие элементы множеств заданного тезауруса.

В общем случае предложение $\mathcal{L}_{ij}$ представлено совокупностью элементов $\mathcal{L}_{ij}=\left\{s_t, \bigcup\limits_{n\in\Omega_z(ij)} z_n, \bigcup\limits_{t\in\Omega_x(ij)} x_t\right\}$ или символов $\mathcal{L}_{ij}=\{l_k;\ k=\overline{1,\xi}\}$, где t, n, t - конкретные для $(i\ j)$-го показателя номера элементов множеств S, Z, X; l_k - символы формализованного языка запросов; ξ - длина запроса. Заметим, что для любых $\mathcal{L}_{ij}$ $\Omega_z(i\ j)\neq\varnothing$, тогда как иногда $\Omega_x(i\ j)=\varnothing$.

Разбиение $\mathcal{L}_{ij}$ на элементы множеств S, Z, X осуществляется следующим образом.

Предварительно массивы S, Z, X ранжируются в порядке убывания длин их элементов (количества образующих их символов); значения длин хранятся в соответствующих элементах массивов длин данных множеств v_s, v_z, v_x.

С учетом того, что s_t всегда находится в начале $\mathcal{L}_{ij}$, а $\Omega_z(ij)\neq\varnothing$, наборы символов l_k сравниваются до первого совпадения последовательно с элементами S, Z, X.

Определение кодов $\mathcal{L}_{ij}$ предусматривает выполнение следующих шагов:

I. $E=S=\{s_k\}$, $k=0$, $\rho=1$.

2. $k=k+1$.

3. Если $k>k_{max}$, переход к п.6.

4. Если $e_k=\left\{\bigcup\limits_{n=\overline{\rho,v_E^i}} l_n\right\}$, то

$$\rho=v_E^i+1,$$

$$\mathcal{L}_{ij}=\left\{\bar{\theta}\,\|\,\theta_k\,\|\left(\bigcup_{m=\overline{\beta,q}} l_m\right)\right\},$$

где $\bar{\theta}$ - коды элементов, найденные на предыдущих шагах; θ_k - код элемента e_k. Переход к п.8.

5. Переход к п.2.

6. Если $E=S$, то $E=Z=\{z_k\}$, $k=0$, переход к п.2.

7. Если $E=Z$, то $E=X=\{x_k\}$, $k=0$, переход к п.2.

8. Если $\rho\neq q$, то $E=Z=\{z_k\}$, $k=0$, переход к п.2.

9. Конец.

Реализация алгоритма определения содержимого L_{ij} способом контурной развертки основывается также на выделении значений установленных признаков согласно формуле (7) за исключением σ_{kp}. Отличие заключается лишь в том, что значения указанных признаков определяются только для одного, сканируемого домена; необходимо дополнительно хранить полученное содержимое всех доменов I-го рода заданного домена 2-го рода, пока не закончится его сканирование.

Таким образом, реализация данного алгоритма позволяет наряду со средствами, рассмотренными в [I; 2], использовать следующую технологию обработки задач АСУ: а) составление на экране терминала ТВТ; б) приведение ТВТ к нормализованному виду; в) запуск программы синтеза запросов; г) запуск программы определения значений показателей; д) вывод на экран терминала значений показателей c_{ij}; е) добавление (или изменение) при необходимости информации в $A_{1(2)}$ и/или $B_{1(2)}$; ж) распечатка результирующей ТВТ на АЦПУ.

Составлены и апробированы *PL/1*-программы, реализующие обе версии данного алгоритма, для различного рода ТВТ как в пакетном, так и в диалоговом режиме. Апробирована работа данного алгоритма совместно с известными средствами при решении в пакетном режиме конкретной задачи для одного из отделов Госплана МССР.

Литература

I. Бужор П. Ф. Обработка информации в АСУ с применением баз данных фактографического типа и семантической идентификации информации. Деп.рукопись. МолдНИИНТИ Госплана МССР. - Библиографический указатель "Депонированные рукописи", 1983, № I.

2. Корня С. К. Инструментарий построения безызбыточных баз данных. - В кн.: Информационные языки и базы данных в планировании. М., ЦЭМИ АН СССР, 1980.

3. Недельчук В. П. Вопросы идентификации фигур с использованием оптоэлектронных параллельных преобразователей изображения. - Сб.трудов МАИ, вып.461. М., 1978.

4. Недельчук В. П. Использование методов ассоциативной и прямой адресации для организации фактографических БД АСУ.- В кн.: Методология АСУ. Кишинев, Штиинца, 1982.