

7599

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

ДУБНА



7599

Экз. чит. зала
11 - 7599

С.Х.Бычваров, И.Б.Бычварова

ПРОГРАММЫ ДЛЯ ПРОВЕРКИ КОРРЕКТНОСТИ
ОПИСАНИЯ КОНЦЕПЦИИ УМОЛЧАНИЯ
И ДЛЯ ПОЛУЧЕНИЯ И УПОРЯДОЧЕНИЯ
"РЕЗУЛЬТИРУЮЩИХ МНОЖЕСТВ"

1973

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКИ И АВТОМАТИЗАЦИИ

11 - 7599

С.Х.Бычваров, И.Б.Бычварова

ПРОГРАММЫ ДЛЯ ПРОВЕРКИ КОРРЕКТНОСТИ
ОПИСАНИЯ КОНЦЕПЦИИ УМОЛЧАНИЯ
И ДЛЯ ПОЛУЧЕНИЯ И УПОРЯДОЧЕНИЯ
"РЕЗУЛЬТИРУЮЩИХ МНОЖЕСТВ"

В настоящей работе дается краткое изложение программ проверки корректности описания концепции умолчания, а также программ получения и упорядочения результирующих множеств. Программы реализуют алгоритмы, обоснованные в работах /3,5/. Они составлены на языке ФОРТРАН для БЭСМ-6 (мониторная система "Дубна"). Длина всех программ около 16 тысяч символов. Была отработана техника программирования операций с множествами на языке ФОРТРАН.

Использованную в настоящей работе терминологию см. в /3,4,5/.

1. Проверка корректности описания концепции умолчания

Программы, описанные в этом разделе, проверяют полноту описания концепции умолчания и находят паразитные правила умолчания.

Если система правил умолчания (каскад простых правил умолчания) полная, то с её помощью всегда можно получить всю необходимую (полную) информацию. Полнота системы правил умолчания имеет отношение к семантической правильности описания концепции умолчания. Критерии полноты каскада простых правил умолчания обоснованы в работе /3/.

Эффективность применения системы правил умолчания можно повысить путем устранения паразитных правил. Правило умолчания называется паразитным, если оно не влияет на функционирование системы правил умолчания.

Программы Т131, Т132 и Т133 проверяют необходимые условия полноты системы правил умолчания (см. теорему Т13 в /3/). Для

этих программ совокупность максимальных множеств альтернативных элементов, множество дополнительных элементов и каскад простых правил умолчания с пустым ключевым множеством являются исходной информацией, заданной при помощи одномерных и двумерных массивов.

Программа TI31 проверяет первое условие теоремы TI3 /3/. Она находит то максимальное множество альтернативных элементов (если такое существует), которое, во-первых, не имеет общих элементов с остальными максимальными множествами альтернативных элементов и, во-вторых, содержит не только дополнительные элементы. Если произведение найденного максимального множества альтернативных элементов на объединение правых частей всех простых правил умолчания заданного каскада - пустое, то программа сообщает о неполноте этого каскада. В программе TI31 используются подпрограммы ADD, MULT, ZERO и PRINAD (см. 3).

Программа TI32 проверяет первую часть второго условия теоремы TI3.

В рабочем одномерном массиве MP получается произведение (обозначим его тоже MP) двух максимальных альтернативных множеств MK и MJ ($MK \neq MJ$). Если произведение MP не пустое, то программа проверяет выполнение следующих двух условий:

а) существует недополнительный элемент x из множества $MK \setminus MJ$, и не существует элемента z из дополнения множества $MK \cup MJ$ таких, что $\{x, z\}$ принадлежит совокупности множеств альтернативных элементов;

б) произведение множества $MK \setminus MJ$ на объединение правых частей всех простых правил умолчания заданного каскада - пустое.

Если условия а) и б) выполняются, то программа сообщает о неполноте заданного каскада. В программе TI32 используются подпрограммы ADD, MULT, SUBT, ZERO, IDENT и PRINAD (см. 3).

Программа TI33 проверяет вторую часть второго условия теоремы TI3.

В рабочем одномерном массиве MP получается произведение двух максимальных альтернативных множеств MK и MJ ($MK \neq MJ$). Если произведение MP не пустое, то программа проверяет выполнение следующих двух условий:

а) существуют недополнительный элемент x из множества $MK \setminus MJ$ и элемент y из множества $MJ \setminus P$ (множество $\{x, y\}$ не принадлежит совокупности множеств альтернативных элементов), такие, что для любого элемента z из дополнения множества $MK \cup MJ$, для которого множество $\{x, z\}$ принадлежит совокупности множеств альтернативных элементов, имеем: множество $\{y, z\}$ принадлежит совокупности множеств альтернативных элементов;

б) произведение множества $MK \setminus MJ$ на объединение правых частей всех простых правил умолчания заданного каскада - пустое.

Если условия а) и б) выполняются, то программа сообщает о неполноте заданного каскада. В программе TI33 используются подпрограммы ADD, MULT, SUBT, ZERO, IDENT и PRINAD (см. 3).

Программа TI4 проверяет достаточное условие полноты системы правил умолчания (см. теорему TI4 в /3/). Для этой программы исходными данными являются совокупность максимальных множеств альтернативных элементов и каскад простых правил умолчания с пустым ключевым множеством, заданные при помощи двумерных массивов, а также

множество дополнительных элементов, заданное при помощи одномерного массива.

Программа TI4 получает в рабочем одномерном массиве MP произведение двух максимальных альтернативных множеств MK и MJ ($MK \neq MJ$), в рабочем одномерном массиве MRF - разность $MK \setminus MP$ и в рабочем одномерном массиве MRIGHT - объединение правых частей всех простых правил умолчания, имеющих пустую левую часть. Программа проверяет, выполняется ли хотя бы одно из следующих условий:

- а) множество MP содержит только дополнительные элементы;
- б) множество MP содержит недополнительный элемент x , который принадлежит MRIGHT, и не существует элемента y из дополнения множества MK такого, что множество $\{x, y\}$ принадлежит совокупности множеств альтернативных элементов;
- в) множество MP содержит недополнительный элемент x из MRIGHT и для любого элемента y из дополнения MK, для которого множество $\{x, y\}$ принадлежит совокупности множеств альтернативных элементов, а также для любого недополнительного элемента z из $MK \setminus MP \setminus MRIGHT$ множество $\{y, z\}$ принадлежит совокупности множеств альтернативных элементов.

Если хотя бы одно из условий а), б) и в) удовлетворяется для всех возможных множеств MP, то программа TI4 сообщает о полноте заданного каскада простых правил умолчания. Если для всех возможных множеств MP удовлетворяется условие а) и имеются недополнительные элементы, то тогда сообщение о полноте заданного каскада выдается, если $MRIGHT \neq 0$. В программе TI4 используются

подпрограммы ADD, MULT, SUBT, ZERO, IDENT и PRINAD (см.3).

Программы TI5 и T20 проверяют достаточные условия паразитности простого правила умолчания.

В программе TI5 исходными данными являются совокупность максимальных множеств альтернативных элементов, простое правило умолчания из каскада с непустым ключевым множеством и само непустое ключевое множество. В этой программе левая часть исследуемого простого правила умолчания задана при помощи одномерного массива MLEFT, а правая часть - при помощи переменной MRIGHT. Непустое ключевое множество исследуемого каскада задано при помощи одномерного массива MKEY.

Программа TI5 проверяет, выполняется ли хотя бы одно из следующих трех условий:

- а) множество $MLEFT \cup MKEY$ не является допустимым (в смысле /4/) для заданного каскада,
- б) множество $MRIGHT \cup MKEY$ не является допустимым для заданного каскада,
- в) множество MRIGHT является подмножеством MKEY.

Если хотя бы одно из условий а), б) и в) удовлетворяется, то программа TI5 сообщает о паразитности исследуемого простого правила умолчания. В программе TI5 используются подпрограммы ADD, PRINAD, DOP

Программа T20 проверяет достаточное условие паразитности простого правила умолчания из каскада с пустым ключевым множеством. Исходными данными являются совокупность максимальных множеств альтернативных элементов, каскад простых правил умолча-

ния и простое правило умолчания из этого каскада. Левая часть исследуемого простого правила умолчания задана при помощи одномерного массива MLEFT, а правая часть — при помощи переменной MRIGHT. Программа T20 получает в одномерном массиве MQ множество, которое содержит элемент MRIGHT (т.е. единственный элемент, который задан переменной MRIGHT) и любой другой элемент, который вместе с элементом MRIGHT образует множество альтернативных элементов. Программа также получает в одномерном массиве MQI произведение MQ на объединение правых частей простых правил умолчания с пустой левой частью, которые расположены выше исследуемого простого правила умолчания.

Программа T20 проверяет, выполняется ли хотя бы одно из следующих условий:

- а) элемент MRIGHT принадлежит множеству MLEFT,
- б) множество $MLEFT \cup MRIGHT$ не является допустимым (здесь MRIGHT — множество из единственного элемента, который задан переменной MRIGHT);
- в) существует элемент r из множества MQI, и не существует элемента u из дополнения множества MQ таких, что множество $\{u, r\}$ принадлежит совокупности множеств альтернативных элементов;
- г) исследуемое простое правило умолчания является паразитным для каскада, полученного из заданного путем отбрасывания первых q простых правил умолчания, расположенных выше исследуемого простого правила умолчания, где либо $q = j$, если существуют r (r — правая часть простого правила умолчания, расположенного выше j -го простого правила умолчания) и u (u — эле-

мент из ключевого множества вновь полученного каскада) такие, что $r = u$; либо $q = 0$ — в противном случае и, кроме того, вновь полученный каскад имеет следующую совокупность ключевых множеств: $\{\{u\} : \{u, r\}\}$ — множество альтернативных элементов, элемент u из дополнения множества MQ и элемент r из множества MQI.

Отметим, что при проверке условия г) необходимо использовать программу T15.

Если хотя бы одно из условий а), б), в) или г) удовлетворяется, то программа T20 сообщает о паразитности исследуемого простого правила умолчания. В программе T20 используются подпрограммы ADD, SUBT, MULT; ZERO, IDENT, PRINAD, DOP (см. 3).

Для того, чтобы удовлетворить достаточные условия полноты, иногда приходится добавлять новые правила в систему правил умолчания. Некоторые из добавленных правил умолчания могут оказаться паразитными. Большинство добавленных паразитных правил умолчания обычно выбрасывается в результате применения критериев паразитности. Поэтому для системы правил умолчания необходимо сначала проверить достаточные условия полноты, а затем достаточные условия паразитности. Для языка PL / I ^{1,2/} все добавленные паразитные правила умолчания при проверке достаточных условий полноты были выброшены в результате применения критериев паразитности.

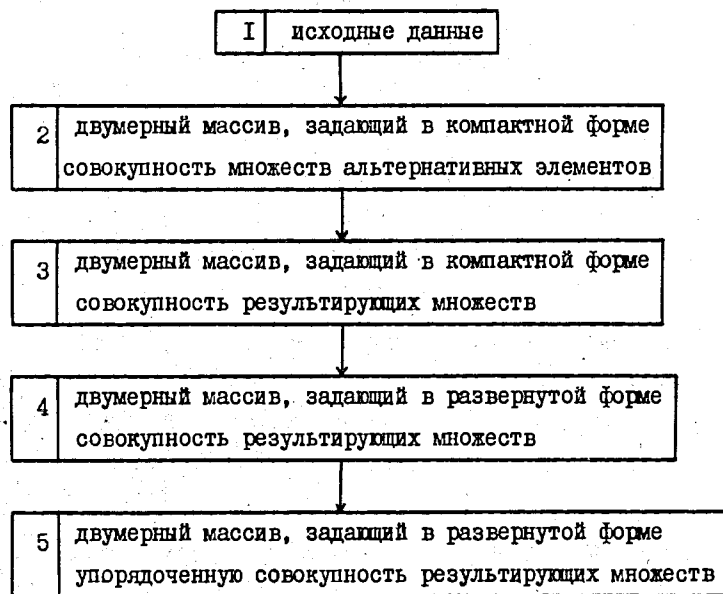
2. Получение и упорядочение результирующих множеств

Получение и упорядочение результирующих множеств является основным преобразованием при получении информации о концепции умол-

чания в форме, удобной для использования в трансляторах^{/5/} :

Программа SAPPU осуществляет получение и упорядочение результирующих множеств полного каскада простых правил умолчания. Проверяются критерии применимости программы SAPPU, обоснованные в теореме T1^{/5/} (программа TDI) и в теореме T2^{/5/} (программа TD2). Если упорядочение результирующих множеств невозможно, программа SAPPU выдает соответствующее сообщение.

Блок-схема основных этапов преобразования информации программой SAPPU



2.1. Исходными данными являются:

а) совокупность максимальных множеств альтернативных элементов, которая задана при помощи двумерного массива $MA(I, J)$, где I - номер очередного максимального множества альтернативных элементов, а J - номер очередного элемента в I -ом множестве;

б) множество дополнительных элементов, заданное при помощи одномерного массива $MDOP$,

в) полный каскад простых правил умолчания, который задан при помощи двумерного массива $MLEFT$ (левые части простых правил умолчания) и одномерного массива $MRIGHT$ (правые части простых правил умолчания).

2.2. Программа $POLEEM$ получает в массиве $MA(I_A, J_A)$ в компактной форме совокупность множеств альтернативных элементов. Элементы массива $MA(I_A, J_A)$ имеют следующие значения:

а) $MA(I_A, J_A) = 0$, если $I_A = J_A$;

б) $MA(I_A, J_A) = I$, если элементы с номерами I_A и J_A образуют множество альтернативных элементов (заранее все элементы множеств альтернативных элементов пронумерованы последовательными натуральными числами, начиная с 1);

в) $MA(I_A, J_A) = -I$ в противном случае.

2.3. Подпрограмма $POLEEM$ также получает в массиве $MAO(I_0, J_0)$ в компактной форме совокупность результирующих множеств, а именно: если $MAO(I_0, J_0) = I$, то элементы с номерами I_0 и J_0 принадлежат некоторому результирующему множеству. Значения элементов массива MAO получаются в результате умножения на $-I$ значений соответствующих элементов массива MA .

2.4. Далее подпрограмма POIREM получает в массиве MEG(IG, JG) в развернутой форме совокупность результирующих множеств, а именно: IG - номер очередного результирующего множества, а JG - номер очередного элемента в IG-ом результирующем множестве. Для этого подпрограмма POIREM находит все максимальные цепочки элементов. Элементы с номерами IO и JO являются соседними в некоторой из найденных максимальных цепочек тогда и только тогда, когда MAO(IO, JO) = I. При получении максимальных цепочек используется техника магазина.

2.5. Подпрограмма TABUL получает в массиве MТАВ(IT, JT) всевозможные отношения предшествования или следования между результирующими множествами. Элементы массива MТАВ(IT, JT) имеют следующие значения:

а) MТАВ(IT, JT) = I, если результирующее множество с номером IT предшествует результирующему множеству с номером JT;

б) MТАВ(IT, JT) = -I, если результирующее множество с номером JT предшествует результирующему множеству с номером IT;

в) MТАВ(IT, JT) = 0 - в других случаях.

Сначала подпрограмма TABUL аннулирует массив MТАВ. Затем она получает в одномерном массиве NMEG произведение двух результирующих множеств, например, с номерами IT и JT ($IT \neq JT$). Подпрограмма TABUL получает в одномерном массиве MSF результат применения заданного полного каскада простых правил умолчания к множеству NMEG. Если множество MSF

совпадает с результирующим множеством с номером IT, то подпрограмма TABUL заносит I в MТАВ(IT, JT). Если множество совпадает с результирующим множеством с номером JT, то подпрограмма TABUL заносит -I в MТАВ(IT, JT).

Подпрограмма STRING получает в массиве MREZ в развернутой форме упорядоченную совокупность результирующих множеств. Она находит столбец в массиве MТАВ, например, с номером IT, все элементы которого не имеют значения -I, и заносит результирующее множество с номером IT из массива MEG в массив MREZ как очередное. Далее она заносит -I в MТАВ(IT, IT) и вновь повторяет описанные выше действия с массивом MТАВ. Если до заполнения всех диагональных элементов массива MТАВ с -I невозможно найти столбец, все элементы которого не имеют значения -I, то это является указанием о наличии цикла.

3. Основные подпрограммы, осуществляющие действия с множествами

3.1. Подпрограмма ADD

Исходными данными являются одномерный массив МК, задающий множество МК и одномерный массив МJ, задающий множество МJ. В одномерном массиве MR подпрограмма получает сумму множеств МК и МJ.

3.2. Подпрограмма SUBT

Исходными данными являются одномерный массив МК, задающий множество МК и одномерный массив МJ, задающий множество МJ. В одномерном массиве MR подпрограмма получает разность множеств МК и МJ.

3.3. Подпрограмма MULT

Исходными данными являются одномерный массив МК , задающий множество МК и одномерный массив МГ , задающий множество МГ . В одномерном массиве MR подпрограмма получает произведение множеств МК и МГ.

3.4. Подпрограмма ZERO

Исходным является одномерный массив МК , задающий исследуемое множество МК . Подпрограмма проверяет, пустое ли множество МК .

3.5. Подпрограмма PRINAD

Исходными данными являются одномерный массив МК , задающий множество МК и одномерный массив МХ , задающий множество МХ . Подпрограмма проверяет, является ли МХ подмножеством множества МК .

3.6. Подпрограмма IDENT

Исходными данными являются одномерный массив МК , задающий множество МК и одномерный массив МГ , задающий множество МГ . Подпрограмма проверяет, совпадают ли множества МК и МГ .

3.7. Подпрограмма DOP

Исходными данными являются двумерный массив МАЛТ , задающий совокупность множеств альтернативных элементов и одномерный массив MD , задающий исследуемое множество MD . Подпрограмма проверяет, содержатся ли хотя бы два элемента MD в одном и том же альтернативном множестве.

Авторы выражают благодарность Н.Н.Говоруно и В.П.Ширикову за помощь и внимание к работе, Е.А.Жоголеву за полезные обсуждения.

ния, Е.Д.Федюнькину - за редакционные замечания.

Л И Т Е Р А Т У Р А

1. Универсальный язык программирования PL /I. (Перевод с английского под редакцией В.М.Курочкина, Москва, Мир, 1968).
2. PL/I Language Specifications Order number GY 33-6003-2 (Major Revision June 1970) IBM 1970.
3. С.Х.Бычваров, ОИЯИ, II-6737, Дубна, 1972.
4. С.Х.Бычваров, ОИЯИ, II-6738, Дубна, 1972.
5. С.Х.Бычваров, ОИЯИ, II-6875, Дубна, 1973.

Рукопись поступила в издательский отдел
13 декабря 1973 года.