

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

DOI 10.17223/20710410/4/2

УДК 681.3.06

ЯЗЫК ПРОГРАММИРОВАНИЯ ЛЯПАС

Н. Р. Торопов

*Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь***E-mail:** toropov@newman.bas-net.by

Излагается история создания языка программирования ЛЯПАС, известного за рубежом как Russian Programming Language. Кратко описываются особенности языка и реализующих его систем программирования для различных типов компьютеров. Отражены основные этапы развития языка и его распространения. Показана возможность эффективного использования языка ЛЯПАС для разработки блоков систем программирования, в том числе трансляторов. Приводятся примеры пакетов прикладных программ и систем, созданных на базе языка ЛЯПАС.

Ключевые слова: язык программирования ЛЯПАС, автоматизация программирования.

1. Исторические предпосылки создания языка ЛЯПАС

Язык программирования ЛЯПАС (Логический Язык Представления Алгоритмов Синтеза дискретных автоматов) появился на заре распространения универсальных цифровых вычислительных машин (УЦВМ, ныне — компьютеров) в Советском Союзе, когда они были в диковинку даже в университетах, так как серийный выпуск УЦВМ только начинал налаживаться (Пенза, «Урал-1», 1957 г.). С появлением первой в Сибири УЦВМ «Урал-1», в Томском государственном университете, у А. Д. Закревского, тогда еще аспиранта кафедры радиофизики, родилась идея: автоматизировать не только синтез дискретных автоматов, но и само программирование алгоритмов. Тем самым решить проблему А — сократить разрыв между теорией, в рамках которой разрабатываются алгоритмы синтеза, и инженерной практикой конструирования реальных автоматов.

В рамках решения проблемы А велись серьезные исследования по пути создания специализированных «автоматических синтезаторов схем». Например, на базе графического метода Рогинского была построена первая машина, автоматически синтезирующая релейные схемы небольшой сложности [3]. Специально для минимизации булевых функций в классе ДНФ была создана машина [11], реализующая метод проб Гаврилова [4].

Однако путь «специализированных синтезаторов», реализующих фиксированные алгоритмы, оказался тупиковым, так как постоянно растущий круг задач, на которые разлагается проблема синтеза дискретных автоматов, настолько широк, что решать их можно только на машинах с гибким программным управлением.

УЦВМ того времени были слишком «заарифметизированы плавающей арифметикой» и плохо приспособлены для решения логических задач, каковыми являются большинство задач анализа и синтеза дискретных автоматов. Поэтому для повышения

логических способностей УЦВМ более приемлемым казался путь создания специализированных логических приставок к ним.

Такой проект был разработан в Сибирском физико-техническом институте при Томском государственном университете в 1961–1962 гг. под руководством А. Д. Закревского [15]. Трудоемкие чисто логические операции должны были выполняться L-приставкой, а управление всем вычислительным процессом возлагалось на УЦВМ.

В основу L-приставки положена идея распределенного выполнения логических операций в пространстве многомерных полей, структурно похожих на многомерные кубы. Такая структура обладает особыми удобствами при выполнении операций над булевыми функциями — особо важными объектами при решении логических задач. При решении трудоемких логических задач производительность УЦВМ, снабженной L-приставкой, поднималась на несколько порядков по сравнению с производительностью УЦВМ без приставки.

В качестве базовой была выбрана УЦВМ «Урал-1», скорость доступа к оперативному запоминающему устройству (ОЗУ) на магнитном барабане которой предполагалось увеличить в несколько раз за счет дополнительных считывающих головок по окружности барабана. Над реализацией проекта трудились несколько дипломников университета, однако из-за скудности материальной базы этому проекту не суждено было воплотиться в жизнь. Следует заметить, что идеи этого проекта были впоследствии реализованы в США в мультипроцессорной системе.

Оставался один путь — использовать УЦВМ. Но программировать их вручную обременительно. Для решения проблемы А нужен удобный, эффективный язык для представления алгоритмов синтеза дискретных автоматов, но отечественных языков, пригодных непосредственно для такой цели, не оказалось. Можно упомянуть лишь операторный метод Ляпунова — средство для представления логической структуры программ [2] и адресный язык Ющенко, обеспечивающий возможность выражения микроструктуры алгоритмов [12]. Только начали разворачиваться работы по созданию трансляторов для отечественных машин с пришедших из-за рубежа языков АЛГОЛ [6] и ФОРТРАН [1]. Но эти языки, «заарифметизированные» не меньше, чем УЦВМ, несмотря на разнообразные их расширения, были плохо приспособлены к решению логических задач. В модификации Алгола-60, в так называемом АЛЬФА-языке [16], было предусмотрено образование булевых векторов путем упаковки логических массивов в машинные слова (ради экономии памяти), но не обеспечивалась возможность непосредственного оперирования с этими векторами (без посредства механизма покомпонентной упаковки-распаковки).

Больше других для описания логических процедур подходил язык APL [9], но он даже на своей родине не был к тому времени реализован, да и сейчас больше распространены интерпретаторы его, чем компиляторы.

Нужно было разрабатывать новый алгоритмический язык, который был бы удобен для представления алгоритмов синтеза, допускал быструю трансляцию при получении компактных машинных программ и позволял бы эффективно использовать производительность УЦВМ.

Следует заметить, что к тому времени у автора будущего языка программирования, А. Д. Закревского, уже был накоплен солидный опыт теоретических разработок в области логического проектирования. Достаточно вспомнить предложенный им, ставший уже классическим, «визуально-матричный метод минимизации булевых функций» [5] и проект вычислительной машины, не уступающей по производительности УЦВМ «Урал-1» и требующей для своей реализации всего около двух десятков триг-

геров [7]. Был некоторый опыт и в программировании предлагаемых алгоритмов на машинном языке (например, [10] по методу [8]).

Работы по созданию языка ЛЯПАС и первой системы программирования с ним были начаты в 1962 г., после II Международного симпозиума по теории релейных устройств и конечных автоматов, где были высказаны первые идеи о развитии исследований в данном направлении [13]. Первой публикацией с кратким описанием языка ЛЯПАС была [14].

2. Характерные особенности языка

Алгоритмический язык ЛЯПАС имеет два уровня. Первый уровень более близкий к собственно машинному языку, более простой и предназначен для представления не слишком сложных алгоритмов [17]. На втором уровне операционные возможности языка расширяются практически неограниченно за счет включения в него Л-операторов, реализуемых подпрограммами на языке ЛЯПАС [18].

Из основных характерных особенностей, отличающих ЛЯПАС от других языков, нужно отметить следующие.

Л-программа (программа на языке ЛЯПАС) является моделью последовательной одноадресной машины. Минимальной лексической единицей такой программы является элементарная операция в линейной цепочке, а не иерархически сложное арифметическое выражение (с обилием скобок) в иерархии блоков программ на других языках. Такая особенность Л-программ упрощает не только их лексический анализ, но и процесс трансляции в целом. Высокая скорость трансляции делает возможным выполнять программы без предварительной их трансляции, что с успехом использовалось при разработке систем программирования с языком ЛЯПАС, но об этом речь впереди.

В языке ЛЯПАС фиксированы число операндов и их имена:

64 *переменных* (32 основных и 32 дополнительных), представляющих булевы векторы стандартной размерности (32 компоненты);

64 *индекса* (32 основных и 32 дополнительных), представляющие булевы векторы стандартной размерности, чаще всего интерпретируемые как натуральные числа;

128 *натуральных* и 52 *специальных векторных констант*;

32 *общедоступных комплекса* (двумерные логические массивы);

16 *специальных комплексов*, представляющих стандартные константы различных типов, переменные, индексы, а также адреса начал и мощности других комплексов.

Особый статус придан так называемому *оперативному комплексу K*, элементы которого поставлены в однозначное соответствие с ячейками оперативной памяти УЦВМ. Теперь ячейки памяти становятся доступными программе, в том числе ячейки с командами самой этой программы. Такая степень доступа к оперативной памяти полезна при использовании языка ЛЯПАС для разработки системных программ, в том числе трансляторов и программ, управляющих исполнением машинных программ, полученных после трансляции.

Особое положение занимает так называемая собственная переменная τ , которая явно в программе никак не обозначается, но, играя роль сумматора-накопителя в модели последовательной одноадресной машины, служит левым операндом для большинства операций языка ЛЯПАС в линейных цепочках Л-программы.

Состав операций подобран с учетом как удобства оперирования с ним при программировании, так и простоты выражения выбранных операций через элементарные

операции типичной УЦВМ. Все операции распадаются на четыре группы: вычислительные, присвоения значений, переходов и обмена с внешней средой.

В группу вычислительных операций включены, например, покомпонентные логические операции, непосредственно реализуемые практически в любой УЦВМ; операции определения номера левой единицы в значении операнда и последовательного перебора таких единиц; операции инверсии порядка следования компонент в векторе и подсчета числа единиц в нем и т. п. Включены в состав также и четыре арифметические операции над натуральными числами, определяемые по модулю 2^{32} .

В группе операций переходов нетипичными являются операция *ухода* к другому участку программы по указанной метке с последующим *возвратом* в точку ухода и операция *ухода в машинный язык*, т. е. к участку оперативной памяти по указанному адресу с последующим возвратом в точку ухода.

На втором уровне языка ЛЯПАС предусмотрена возможность повышения размерности переменных и элементов комплексов (так называемая «штриховка» этих операндов) до величины 32ω , где $\omega \in \{\omega_1, \omega_2, \omega_3, \omega_4\}$ и ω_i — натуральные числа, задающие коэффициенты четырех типов штриховки: ' , " , ' ' , ' ' ' (такие символы ставятся перед операндами с повышенной размерностью). Забегая вперед, скажем, что дорогостоящий механизм штриховки операндов был реализован не во всех системах программирования с языком ЛЯПАС.

Из особенностей второго уровня языка ЛЯПАС, отличающих его от других языков программирования, отметим также то, что среди Л-операторов допускались многополюсники (подпрограммы, имеющие несколько входов и (или) несколько выходов), а в качестве фактических операндов Л-операторов могли быть символы других Л-операторов и даже целые выражения языка ЛЯПАС. Однако такое усложнение фактических операндов у Л-операторов, так же как многополюсники и «переход по переменной» не прижились. Они были реализованы лишь в первых системах программирования с языком ЛЯПАС и, не пользуясь особой популярностью, были исключены из последующих версий языка.

Уместно подчеркнуть еще одну особенность языка ЛЯПАС, вытекающую из его предназначения, — он служит средством публикации представляемых на нем алгоритмов. Еще до окончательного оформления языка проводилась апробация его на разработке нетривиальных алгоритмов синтеза дискретных устройств.

Фактически была установлена тесная связь между проводимыми параллельно исследованиями в областях алгоритмизации процесса логического проектирования, развития алгоритмического языка и построения программирующей системы с ним (ПС-ЛЯПАС). В результате взаимного влияния всех этих процессов друг на друга было достигнуто компромиссное решение, удовлетворяющее одновременно противоречивым требованиям выразительности входного языка, быстродействия транслятора и качества получаемых с его помощью машинных программ.

Такой параллелизм в разработке языка, ПС и прикладных алгоритмов позволил к началу открытия «Всесоюзного коллоквиума по языкам конечных автоматов» в Томске (март 1964 г.) не только продемонстрировать в действии первый отечественный язык программирования ЛЯПАС на первой в Сибири машине «Урал-1», но и представить на коллоквиуме серию докладов с результатами экспериментально-статистических испытаний оригинальных алгоритмов синтеза дискретных автоматов.

Кстати, заметим, что этот коллоквиум впоследствии стали считать первой школой М. А. Гаврилова [85]. В решении коллоквиума было, в частности, записано:

1. Организовать школу-семинар по проблемам синтеза дискретных автоматов и проводить ее тематические занятия регулярно — не реже двух раз в год (летние и зимние) в различных городах Союза под руководством члена-корреспондента АН СССР М. А. Гаврилова.

2. Настоящий коллоквиум считать Первой зимней школой-семинаром с тематикой «ЛЯПАС и его применение».

Вслед за книгой [18] выходит в свет сборник [19] с полным описанием первой версии языка ЛЯПАС. В этом же сборнике приводятся описания множества нетривиальных алгоритмов решения разнообразных логических задач, отладка которых была завершена к моменту сдачи в эксплуатацию систем программирования с языком ЛЯПАС.

Разработка языка велась параллельно с его реализацией сразу на двух машинах: «Урал-1» и М-20. Автор языка стремился повысить его уровень, чтобы облегчить программирование алгоритмов синтеза, а разработчики трансляторов [20, 21] стремились этот уровень понизить, чтобы упростить реализацию языка на машинах с малой производительностью. В то же время им хотелось, чтобы ЛЯПАС был самовыражаем, т. е. чтобы его можно было использовать при разработке отдельных блоков систем программирования.

В результате был найден компромиссный уровень языка ЛЯПАС, допускающий эффективную реализацию и использование его не только для разработки отдельных блоков программирующей системы [22–26], но и для записи на нем самих трансляторов с этого языка, что значительно облегчало разработку, отладку и документирование трансляторов [20, 21]. Правда, транслировать их на машинные языки приходилось вручную. В последующих версиях систем и трансляторы, написанные на ЛЯПАСе, раскручивались автоматически, но об этом речь впереди.

Насколько удачным оказался компромиссный уровень языка, можно судить по приводимым ниже кратким характеристикам первых трансляторов.

Транслятор для машины М20 (ОЗУ — 4096 45-разрядных слов, 20000 операций в секунду) занимает в памяти около 1600 слов и, работая без обращения к дополнительному ЗУ, синтезирует 250 машинных команд в секунду [20]. Высокое быстродействие транслятора (превышающее быстродействие транслятора с Алгола [16] на два порядка) позволило хранить программные модули (в том числе большинство блоков программирующей системы) в компактном исходном языке ЛЯПАС и транслировать их в момент обращения к ним.

Даже на такой малопроизводительной машине «Урал-1» (ОЗУ — 1024 36-разрядных слова, 100 операций в секунду) транслятор [21], не выходя за пределы ОЗУ, способен был обрабатывать Л-программы, объем которых ограничен 32 предложениями и 408 символами. Была предусмотрена также возможность трансляции в два приема, увеличивающая длину транслируемой Л-программы вдвое и доводящая объем получаемой при этом машинной программы до предела возможного (2000 коротких 18-разрядных команд). Быстродействие «Урал-1» слишком мало для решения нетривиальных логических задач, но оказалось достаточным для эффективной отладки алгоритмов решения многих из них. Поэтому в дополнение к транслятору и корректору [21] в программирующую систему для машины «Урал-1» был включен отладчик [23].

Отладчик по краткой инструкции программиста подготавливает отлаживаемую Л-программу к отладке путем вставки в начало каждого предложения нескольких команд, обеспечивающих выдачу на печать траектории процесса выполнения программы (в виде последовательности номеров проходимых при этом предложений) и процесса изменения значений подлежащих наблюдению операндов. При этом объем

выдаваемой информации разумно ограничивается, так как чрезмерный избыток информации иногда хуже ее недостатка. Предложенный принцип отладки выгодно (на порядок! [18]) отличается от широко распространенного принципа «прокрутки» отлаживаемой программы по скорости контрольной реализации, что особенно важно для такой тихоходной машины, как «Урал-1». Заметим, однако, что описываемый отладчик разумно использует даже «тихоходность» машины, позволяя программисту визуально наблюдать траекторию реализации Л-программы по сигнальным лампочкам регистров машины.

3. Состав ПС с языком ЛЯПАС

Прежде чем описывать состав программирующей системы с алгоритмическим языком ЛЯПАС, напомним, что УЦВМ того времени были чисто цифровыми. Поэтому всякая информация (вводимая, выводимая и обрабатываемая) кодировалась цифровыми кодами, в том числе и Л-программы, каждый символ которых кодировался девятиразрядным кодом.

При хорошем распределении труда делал это, как правило, специально выделенный человек-кодировщик. Он же перфораторщик. Он же курьер с пакетами заданий на машину. В лучшем случае он же (если имел допуск, иначе появлялся еще один посредник) оператор за пультом машины, вводящий задания в машину и получающий распечатки с результирующими колонками цифр на бумаге, которые курьер должен был доставить с нетерпением ожидающим их программистам.

Такое пояснение делает понятным наличие блока «*корректор*» [21] в ПС-ЛЯПАС [21, 22], который помогает программисту исправить уже закодированную Л-программу (иногда большого объема) по краткой «инструкции», показывающей, что где подправить, а что оставить без изменения.

Кроме упоминавшихся уже *транслятора* и *корректора*, в состав первой версии ПС-ЛЯПАС [22] входят также следующие основные блоки: *дирижер*, *синош*, *подот* и *отинф*, *ликш*, *библиотекарь* и *Л-компилятор*.

Дирижер организует взаимодействие блоков, работа которых необходима в указанном программистом режиме использования ПС.

Синош осуществляет синтаксический контроль выполняемой Л-программы и если обнаруживает ошибки, то печатает их перечень и блокирует работу других блоков [24].

Подот готовит к отладке Л-программу на основании кратких указаний программиста (какие участки программы и изменения каких операндов его будут интересовать), а *отинф* обеспечивает выдачу отладочной информации по установленной форме.

Ликш «ликвидирует штрихи», заменяя операции над операндами повышенной размерности соответствующими подпрограммами, реализующими эти операции путем программного объединения соответствующих совокупностей машинных слов в памяти.

Библиотекарь находит в библиотеке нужные Л-программе Л-модули и упорядочивает их надлежащим образом, чтобы ускорить процесс их компиляции в тело Л-программы [26].

Л-компилятор обеспечивает переход от Л-программы второго уровня, содержащей символы Л-операторов, к эквивалентной по действию Л-программе первого уровня, уже не содержащей символов Л-операторов [25].

После Л-компилятора вступает в работу *транслятор*. Это единственный блок ПС, который хранится в машинном языке. Все остальные блоки, будучи разработанными

в языке ЛЯПАС, преобразуются в машинную форму лишь в процессе подготовки ПС к сеансу.

Компактность ПС давала возможность каждому программисту иметь свой личный дубликат ее и легко переносить его на любую доступную ему машину указанного выше типа.

4. Распространение языка

Момент ввода ПС-ЛЯПАС в эксплуатацию совпал с пиком серийного выпуска машин «Урал-1» (Пенза) и М-20 (Казань) и широкого распространения их по Союзу. В результате ЛЯПАС оказался востребован во многих городах Союза (Москве, Ленинграде, Риге, Новосибирске, Свердловске, Киеве, Севастополе, Таганроге, Кишиневе, Фрунзе и др.), причем в большинстве случаев не столько сам ЛЯПАС, сколько прикладные программные продукты, созданные на его базе.

ЛЯПАС многократно был темой широких дискуссий на конференциях и семинарах, например на семинаре по языку ЛЯПАС в Ленинграде (1966 г.); на Международной школе по ЛЯПАСу в Праге (1967 г.); на Workshop в Любляне (Чехословакия, 1979 г.), где ЛЯПАС был признан лучшим для логического синтеза среди десятка других языков [41]; на семинаре в Томске [35] и др.

После издания английского перевода [27] сборника [19] с полным описанием языка ЛЯПАС, опирающихся на него ПС на машинах «Урал-1» и М-20, а также набора алгоритмов на этом языке, решающих широкий круг задач из области анализа и синтеза дискретных устройств, о ЛЯПАСе узнают за рубежом.

В США автор перевода [27] Norton Nadler организует «User Group for Russian Programming Language»¹, в анонсе о которой он, в частности, дает такую оценку языку ЛЯПАС: «Having been designed for implementation on Soviet computers, it is remarkably efficient, both in compiler memory and running time requirements, and in object program parameters. Suffice it to say that there exist a Ural-1 version (as if APL had been implemented on the IBM 650!)»² [29].

Появляются трансляторы в Польше [28], Югославии [41], Чехословакии, ГДР, интерпретатор в США [33].

5. Развитие языка и систем программирования

По мере широкого распространения ПС-ЛЯПАС и ее интенсивной эксплуатации на разных типах машин постоянно высказывались различные пожелания по модернизации как языка ЛЯПАС, так и базирующейся на нем ПС.

5.1. Л Я П А С - 7 1

В дальнейшем язык был расширен в сторону вычислительных операций над числами различных типов. Наибольшее распространение получили при этом версии ЛЯПАС-70 на машине М-20, БЭСМ-3М, БЭСМ-4, СМ-4, «Минск-2», «Минск-22» [30] и ЛЯПАС-71 [31, 32] на машинах М-220 и БЭСМ-6.

В версии ЛЯПАС-71 существенное развитие получили второй и третий уровни языка, обеспечивающие удобства поблочной разработки суперпрограмм. Развитый в этой

¹Группа пользователей русского языка программирования.

²Будучи разработанным для советских компьютеров, он исключительно эффективен по компактности и быстрдействию как самих компиляторов, так и получаемых с их помощью объектных программ. Достаточно сказать, что существует версия компилятора для машины «Урал-1» (что равносильно реализации APL на IBM 650!).

версии механизм автоматической сегментации программ позволяет разрабатывать и реализовывать в единой системе программы, содержащие десятки тысяч команд.

Полагается, что сегментом может быть любая подпрограмма, если только соответствующая Л-операция будет отмечена в некоторой обратившейся к ней программе дополнительно символом *. Любая подпрограмма, содержащая некоторую подпрограмму-сегмент, также может использоваться как сегмент, т. е. число уровней сегментирования практически не ограничивается. В общее число Л-сегментов и М-сегментов (сегментов на машинном языке), которыми может располагать программист, входят и все блоки СП. Ознакомившись с их описаниями, программист может эффективно использовать их в своих программах.

5.2. Л Я П А С - М

Богатая практика использования языка ЛЯПАС и развитие средств вычислительной техники подсказали пути дальнейшего его совершенствования. Наконец-то УЦВМ перестали быть чисто цифровыми и стали воспринимать, кроме цифр, также буквы и символы, научились измерять время! В конце 1974 г. появилась новая версия языка, названная ЛЯПАС-М [34], существенно отличающаяся от всех предыдущих. От прежнего языка ЛЯПАС осталась лишь концептуальная основа. Кардинально поменялась символика, в результате чего была утрачена совместимость между программами, представленными в ЛЯПАС-М, и Л-программами, написанными на прежних его версиях.

Совершенствование языка пошло по пути приближения его символики к стандартным алфавитам отечественных устройств отображения информации, расширения языка путем включения в него операций над символами, операций измерения времени. Расширены чисто вычислительные возможности языка: включены операции над действительными числами с плавающей запятой.

Большие удобства для программирования представляют также операции над комплексами (двумерными логическими массивами), которые включены в язык на уровне элементарных операций (сбылась давнишняя мечта программистов!). Получили дальнейшее развитие средства модульного программирования. Существенно расширены возможности применения языка не только для представления алгоритмов синтеза, но и для решения логико-комбинаторных задач широкого класса, в том числе для создания самих систем программирования, уже без каких-либо оговорок и ограничений, как это было в прежних версиях.

Наличие в языке средств оперирования с двоичными векторами, символами и битами приближает его по эффективности к машинно-ориентированным языкам и дает возможность писать на нем программы, не уступающие по компактности и быстродействию программам, составленным на автокодах, что позволяет использовать язык при разработке программирующих и операционных систем. Исключительная простота синтаксиса языка, сохранившаяся от предыдущих его версий, допускает создание простых и эффективных интерпретирующих систем [36], удобных для проведения отладки программ в режиме диалога. В то же время высокая скорость трансляции, являющаяся следствием простоты языка, делает возможным включение процесса трансляции во внутренние циклы программ, что с успехом используется при построении на базе языка ЛЯПАС-М эффективных диалоговых систем.

С момента появления первой публикации с описанием новой версии языка ЛЯПАС-М в 1974 г. [34] начались исследования по его реализации и созданию систем программирования для различных машин параллельно несколькими коллективами: для ЭВМ «Минск-32» — в Минске, для ЕС и СМ ЭВМ — в Томске и для БЭСМ-6

— в Москве. Краткие сведения об этих системах содержатся в книге [40], а также в [65, 73].

Каждый коллектив шел своим путем, и поэтому созданные ими системы программирования имеют свои уникальные особенности. Например, в Белорусском государственном университете пошли традиционным путем создания трансляторов, когда продуктом трансляции является загрузочный модуль той же структуры, что и загрузочные модули, получаемые с других языков программирования. Такой путь дает возможность легко «вписаться» в действующие операционные системы и разрешить вопрос межъязыковой стыковки без существенных затрат. Однако для синтаксически простого языка ЛЯПАС-М (по сравнению с другими языками программирования общего назначения) многие другие вопросы реализации могут быть решены гораздо проще и лучше, если ориентироваться на специфику языка и не придерживаться традиционных приемов.

В Институте технической кибернетики АН БССР были проведены исследования по двум направлениям — создание интерпретатора с языка ЛЯПАС-М на машине «Минск-32» [36] и разработка мобильного (не зависящего от машины) транслятора, написанного на языке ЛЯПАС-М [57]. Блоки транслятора, написанные на языке ЛЯПАС-М, приводились в рабочее состояние на машине «Минск-32» методом «самораскрутки» с помощью интерпретатора [46, 79].

Созданный таким образом транслятор явился ядром для системы ДИСМО (диалоговая система мгновенного обслуживания) [39, 42, 43, 50], реализованной на машине «Минск-32», оснащенной терминалами типа пишущей машинки. ДИСМО показала неплохие эксплуатационные характеристики по обслуживанию параллельно многих абонентов, но не получила должного распространения, как и другие системы программирования для устаревшей и вскоре снятой с эксплуатации машины «Минск-32».

Несколько дольше эксплуатировалась система, разработанная в Томске для ДОС ЕС ЭВМ [44], но и ее пришлось реконструировать, чтобы вписать в более совершенную операционную систему. Хотя внешних побудительных причин для реконструкции системы ПРОЛОГ [51], разработанной в Москве для машины БЭСМ-6, и не было, опыт эксплуатации ее стимулировал стремление к совершенствованию, породив несколько версий, расходившихся даже по входному языку. В одну из последних версий был включен транслятор с эталонной версии языка ЛЯПАС-М, полученный из мобильного транслятора [57].

5.3. Система программирования ЛЕС

На базе опыта разработки и пробной эксплуатации упомянутых систем в рамках ОС ЕС ЭВМ создается диалоговая система программирования ЛЕС (ЛЯПАС-М для ЕС ЭВМ) [47, 48, 52, 53, 66]. В ее состав включаются два типа трансляторов с языка ЛЯПАС-М: быстродействующий — для разработки и отладки программ в языке ЛЯПАС-М (создается в Томске [59]) и оптимизирующий — для оптимизации отлаженных программ (создается в Минске [38, 60, 61]). Система ЛЕС предназначена для разработки, отладки [52], оптимизации и эксплуатации [53, 66] программ на языке ЛЯПАС-М в диалоговом [48] и пакетном режимах работы.

Расширение сферы применения языка ЛЯПАС-М, создание новых систем программирования на различных типах машин силами различных коллективов и накопление опыта эксплуатации этих систем в различных условиях так или иначе влияют на устойчивость конструкций языка, побуждая его развитие. Хотя все упоминавшиеся выше системы программирования базируются на одном и том же алгоритмическом языке

ЛЯПАС-М, каждая из них специфична не только по принципам организации и внутренней структуре, но и по версии входного языка, хотя и незначительно, но отличающейся от других версий.

Наряду с нежелательной специализацией версий входного языка по отношению к классу машин наметилась тенденция к созданию систем программирования, инвариантных относительно исходных текстов программ. Чтобы удержать в определенном русле фантазию разработчиков систем программирования при выборе ими версий входного языка, зафиксирована так называемая эталонная версия языка ЛЯПАС-М, описанная в [40], которая должна была быть обязательным подмножеством входного языка каждой из разрабатываемых систем программирования с языком ЛЯПАС-М.

Ближе других к эталону находятся входные языки систем программирования, описанные в [43, 44, 47], а также входной язык транслятора [57], работающего в рамках системы ПРОЛОГ (на БЭСМ-6) [51], а дальше всех от эталона отстоит ветвь языка, являющаяся входным языком московского варианта транслятора той же системы ПРОЛОГ.

Для операционной системы ОС ЕС программирующая система ЛЕС-4 [66] представляет собой автономный программный комплекс, функционирующий в роли обычного задания, а с точки зрения пользователя она является иерархической многоязыковой диалоговой системой, обеспечивающей удобные средства для разработки, отладки и эксплуатации программ в языке ЛЯПАС-М одновременно с восемью терминалов. Система обеспечивает два режима работы: диалоговый и пакетный. При этом функционирование основных блоков ПС не зависит от режима, т. е. работа ведется одинаково: на тех же входных данных, с получением тех же выходных результатов. От пакетного режим диалога отличается лишь тем, что ведется он экспромтом, а не по заготовленному заранее жесткому сценарию. Однако из-за недостаточного количества терминалов система ЛЕС-4 эксплуатировалась больше в пакетном, чем в диалоговом режиме.

С момента выхода в свет первой публикации с описанием языка ЛЯПАС-М [34] успело смениться целое поколение машин: ушла со сцены «Минск-32», оказавшая определенное влияние на структуру, состав и синтаксис языка; ЕС ЭВМ повсеместно вытеснены персональными компьютерами.

Назрела необходимость в очередной ревизии языка ЛЯПАС, чтобы полнее учесть особенности современных ЭВМ и предотвратить дальнейшее расхождение его версий путем удовлетворения содержащихся в них пожеланий. Такая ревизия была проведена в книге [68], содержащей полные описания модернизированного языка ЛЯПАС-М и реализующей его системы программирования ЛЕС, базирующейся на ОС ЕС и существенно ориентированной на диалоговый режим работы.

Изменения в языке коснулись прежде всего операций обмена с внешними устройствами, операций над комплексами, организации межмодульных связей, а также введения дополнительных средств для повышения степени инвариантности программ к различным типам ЭВМ. Особое внимание в системе ЛЕС уделяется комплексной отладке многомодульных программ.

Ядром системы является макроинтерпретатор [53], который управляет выполнением макроопераций в языке ЛЯПАС-М и освобождает программиста от предварительной трансляции Л-модулей, обеспечивая поиск в библиотеке нужного модуля (при необходимости включая в работу транслятор) и проверяя соответствие типов формальных операндов найденного модуля и фактических операндов выполняемой макрооперации. Попутно макроинтерпретатор исполняет роль сегментатора: если имя очередной выполняемой макрооперации отмечено в списке сегментов или реализующий ее

М-модуль не помещается в отведенном для М-модулей участке оперативной памяти, то последний очищается для размещения там нового сегмента, возглавляемого найденным модулем.

Когда требуется максимально повысить быстродействие уже отлаженной программы, некоторые из ее модулей, выполняющихся во внутренних циклах, лучше оттранслировать предварительно, а еще лучше собрать их в М-теку (совокупность М-модулей, жестко связанных в единый сегмент, целиком помещающийся в отведенном участке оперативной памяти). Конструирование жестко связанных в М-теку модулей осуществляется М-компилятором [62].

Предусмотрены средства, обеспечивающие программную совместимость с загрузочными модулями ОС ЕС, полученными трансляторами с других языков программирования [49, 63].

подавляющее большинство блоков системы ЛЕС разрабатывалось с использованием машинно-независимого языка ЛЯПАС-М, что существенно упростило перенос ее на персональные компьютеры [79]. Лишь с переносом системы ЛЕС на персональные компьютеры в полной мере оказались востребованными ее диалоговые средства [48, 78]. Получили дальнейшее развитие средства диалогового программирования [69], в том числе комплексной динамической отладки и редактирования программ в языке ЛЯПАС-М [52, 74], впервые в системе ЛЕС введена в действие динамическая сегментация данных [70, 71].

Несколько слов об отладке программ в языке ЛЯПАС-М. В системе ЛЕС предусмотрены два режима отладки: детальная отладка в режиме макродиалога в специальном отладочном языке ОПЛ и простая отладка в режиме микродиалога в контрольных точках отлаживаемой Л-программы [74]. Отладка в режиме макродиалога в оригинальном языке ОПЛ внешне для программиста ничем не отличается от отладки с помощью отладочной программы на базе интерпретатора языка ЛЯПАС-М [36], кроме того, что в нашем случае отладка базируется на быстродействующем трансляторе.

Однако наибольшей популярностью среди программистов пользуется режим отладки с контрольными точками, напоминающий режим отладки в языке С++. Основные отличия их сводятся к следующему. В системе ЛЕС:

- имеется возможность автоматической расстановки контрольных точек в начале каждого предложения отлаживаемой Л-программы;
- в момент прерывания в каждой контрольной точке программист может выполнить любую цепочку операций в языке ЛЯПАС-М, в том числе изменяющую значения любых операндов отлаживаемой программы и содержащую макрооперации обращения к другим подчиненным модулям;
- введен специальный режим визуализации процесса выполнения отлаживаемого Л-модуля.

Возможность выполнения в любой выделенной контрольной точке отлаживаемого Л-модуля произвольной цепочки операций языка ЛЯПАС-М, в том числе макроопераций (чего не допускается в других системах программирования), позволяет программисту не только отлаживать программу, но и исследовать ее возможности. Он может за один «прогон» программы испытать множество ее вариантов без изменения исходного текста, подставляя в цепочку макрооперации, инициирующие выполнение различных модулей, в том числе не транслированных предварительно.

Режим визуализации процесса выполнения отлаживаемого Л-модуля, также отсутствующий в других системах программирования, заключается в следующем. В про-

цессе выполнения Л-модуля на экране высвечивается его активный участок (с особой подсветкой номера активного предложения), а также значения всех его переменных. В этом режиме легко наблюдать за ходом выполнения программы, особенно явно видны циклы, в том числе нежелательные. Функциональные возможности режима еще больше возрастут, если реализовать управляемое масштабирование скорости выполнения наблюдаемого модуля.

6. Прикладные пакеты и системы на базе языка ЛЯПАС

Язык ЛЯПАС и реализующие его программирующие системы никогда не были самоцелью для их создателей, а были лишь инструментальным средством для разработки логико-комбинаторных алгоритмов широкого применения и систем логического проектирования дискретных устройств.

Заметим, что описания большинства из ранних систем автоматизации проектирования, разработанных на базе языка ЛЯПАС, содержались лишь в отчетах заказчикам и не подлежали открытой публикации. Впервые полное описание одной из таких систем появилось в открытой печати в книге «Синтез асинхронных автоматов на ЭВМ» [37], содержащей детальное описание алгоритмов, выполняющих все этапы сквозного синтеза: от формального описания условий работы синтезируемого автомата до получения реализующей его схемы из элементов заданного базиса. Приводятся оценки эффективности отдельных алгоритмов и системы в целом.

Из наиболее значимых продуктов коллективного труда следует назвать ИСАПР — инструментальную исследовательскую систему автоматизации логического проектирования [77, 80–83], разработанную на базе языка ЛЯПАС-М. ИСАПР представляет собой информационно-программный комплекс, служащий технологической средой для разработки и полигоном для испытания как отдельных алгоритмов решения частных задач логического проектирования, так и сквозных технологий автоматизированного проектирования дискретных устройств различных классов. Система может служить также средством для создания на ее базе автономных специализированных САПР (как для работы в пакетном режиме, так и в диалоговом) и инструментом для проектирования конкретных устройств управления.

Система эксплуатируется на персональных компьютерах. В качестве базовых языков программирования при разработке прикладного программного обеспечения в рамках ИСАПР допускается использование ЛЯПАС-М и С, поддерживаемых системами программирования ЛЕС-7 [78] и TURBO-C в рамках операционной системы MS-DOS.

Не задаваясь целью сделать обзор всех программных продуктов, созданных на базе языка ЛЯПАС, сошлемся лишь на некоторые примеры пакетов программ, автоматизирующих логическое проектирование дискретных управляющих устройств [45, 58, 67, 72, 75, 76], а также анализ и моделирование логических схем [54, 55]. Включение в язык ЛЯПАС-М операций над символами и символьными комплексами открыло для него новую сферу применения [56, 64, 84].

Заключение

Долгое время (более 30 лет) ЛЯПАС не имел себе равных в решении логико-комбинаторных задач, имеющих дело с объектами типа булевы и троичные векторы и матрицы, вплоть до появления языка C++ Visual после пополнения его библиотеки специальными классами векторно-матричных объектов [86, 87]. Но и в таких условиях представленные на языке ЛЯПАС-М алгоритмы не уступают, а порой и превосходят по быстрдействию те же алгоритмы, написанные на C++.

Справедливости ради, однако, заметим, что ЛЯПАС-М уступает С++ в читабельности программ и, как следствие, в удобстве разработки больших программ. Виной тому является оставшееся от первой версии языка ограничение на число операндов и фиксирование их имен. Для машин первого поколения с малым объемом их оперативной памяти, когда приходилось экономить каждый байт, такие ограничения, повышающие компактность Л-программ, были благом. Программы были обозримы одним взглядом (помещались на одном листе бумаги), и компактность не мешала их читабельности. Другое дело теперь. Поскольку нет острого дефицита памяти, то, естественно, возросли размерности решаемых задач и, следовательно, объемы выполняемых программ. На первый план выдвигается читабельность программ, а компактность их теряет актуальность.

Свидетельством того, что ЛЯПАС-М успешно функционирует, является публикация [88] с описанием большого набора разработанных на языке ЛЯПАС-М алгоритмов для оптимальной полиномиальной реализации системы частичных булевых функций. Приведены результаты многочисленных экспериментов по испытанию алгоритмов, которые доказывают их высокую эффективность.

ЛИТЕРАТУРА

1. *Backus J. W., Beeber R. J. and other.* The FORTRAN automatic coding system // Proc. Of the Western Joint Computer Conference. Los Angeles, Calif., February 1957. P. 188–198.
2. *Ляпунов А. А.* О логических схемах программ // Проблемы кибернетики. Вып. 1. М.: Физматгиз, 1958. С. 46–74.
3. *Архангельский А. А., Лазарев В. Г., Рогинский В. Н.* Машина для синтеза контактных схем // Проблемы передачи информации. Вып. 1. М.: Изд-во АН СССР, 1959. С. 41–52.
4. *Гаврилов М. А.* Минимизация булевых функций, характеризующих релейные цепи // Автоматика и телемеханика. 1959. № 9. С. 1217–1238.
5. *Закревский А. Д.* Визуально-матричный метод минимизации булевых функций // Автоматика и телемеханика. 1960. Т. 21. № 3. С. 369–373.
6. *Бэкус Дж. В. и др.* Сообщение об алгоритмическом языке Алгол-60 // ЖВМ и МФ. 1961. Т. 1. № 2. С. 308–342.
7. *Закревский А. Д.* Формализация синтеза электронной цифровой вычислительной машины // Вычислительная техника. Автоматика, теория информации. Томск: Изд-во Том. ун-та, 1961. С. 64–72.
8. *Закревский А. Д.* К синтезу последовательностных автоматов // Труды СФТИ. Вып. 40. Томск: Изд-во Том. ун-та, 1961. С. 73–88.
9. *Iverson K. E.* A programming language. New York: Wiley, 1962.
10. *Бутаков Е. А., Закревский А. Д.* Минимизация числа состояний релейной схемы на универсальной вычислительной машине «Урал-1» // Проблемы передачи информации. Вып. 11. М.: Изд-во АН СССР, 1962. С. 66–76.
11. *Тимофеев Б. Л.* Машина для минимизации булевых функций // Структурная теория релейных устройств. М.: Изд-во АН СССР, 1963. С. 242–249.
12. *Ющенко Е. Л.* Адресное программирование. Киев: ГТИ УССР, 1963.
13. *Zakrewskij A. D.* On conference papers by S. Waligorskij and A. Stogny. Relay systems and finite automata // Translated proceedings, Burrough Corp., 1964. P. 33–34.
14. *Закревский А. Д.* ЛЯПАС — логический язык представления алгоритмов синтеза // Теория автоматов. Киев: ИК АН УССР, 1964. С. 3–29.

15. *Закревский А. Д.* Машина для решения логических задач типа синтеза релейных схем // Синтез релейных устройств: Труды Междунар. симп. по теории релейных устройств и конечных автоматов. М.: Наука, 1965. С. 346–356.
16. *Ершов А. П., Кожухин Г. И., Поттосин И. В.* Обзор особенностей АЛЬФА-языка. АЛЬФА — система автоматизации программирования / Под ред. А. П. Ершова. Новосибирск: РИО СО АН СССР, 1965. С. 14–30.
17. *Закревский А. Д.* Первый уровень языка ЛЯПАС // Труды СФТИ. Вып. 48. Томск: Изд-во Том. ун-та, 1966. С. 12–22.
18. *Закревский А. Д.* Алгоритмический язык ЛЯПАС и автоматизация синтеза дискретных автоматов. Томск: Изд-во Том. ун-та, 1966. 266 с.
19. Логический язык для представления алгоритмов синтеза релейных устройств / Под ред. М. А. Гаврилова. М.: Наука, 1966. 342 с.
20. *Товштейн М. Я.* Транслятор для быстродействующих УЦВМ // Там же. С. 52–69.
21. *Торопов Н. Р.* Транслятор для машины УРАЛ-1 // Там же. С. 99–126.
22. *Закревский А. Д., Товштейн М. Я., Торопов Н. Р.* Программирующая система ПС-ЛЯПАС // Там же. С. 47–51.
23. *Закревский А. Д.* Отладка Л-программ на машине «Урал-1» // Там же. С. 79–90.
24. *Усачева Н. А.* Поиск синтаксических ошибок в Л-программах // Там же. С. 75–78.
25. *Закревский А. Д.* Компилятор ЛЯПАСа // Там же. С. 70–74.
26. *Быкова С. В., Воробьев В. А., Закревский А. Д.* Библиотека подпрограмм системы ЛЯПАС // Докл. Всесоюзн. коллоквиума по автоматизации синтеза дискретных вычислительных устройств. Новосибирск: СО АН СССР, 1966. С. 4–10.
27. LYaPAS: A programming language for logic and coding algorithms / Ed. by M. A. Gavrilov and A. D. Zakrevskii. New York and London: Academic Press, 1969. 475 p.
28. *Michalski A., Wiewiorowski T.* Odra Ljapas. Warszawa: Computation Centre Polish Academy of Sciences, 1970. 33 p.
29. *Nadler N.* User Group for Russian Programming Language // IEEE, Newsletter for Computer-Aided Design. ISSUE № 3, may/june, 1971.
30. *Усачева Н. А.* Система автоматического программирования с языком ЛЯПАС для ЭВМ «Минск-22», работающей в режиме Т. М.: ВНТИЦ, 1971. 151 с.
31. *Закревский А. Д.* Алгоритмы синтеза дискретных автоматов. М.: Наука, 1971. 512 с.
32. *Закревский А. Д., Белоусова Н. А., Жирова Л. П. и др.* Программирующая система ЛЯПАС-71 // Системное программирование: Материалы Всес. симп., март 1973). Ч. 2. Новосибирск, 1973. С. 84–95.
33. *Charles J., Albright Jr.* An interpreter for the language LYaPAS. University of North Carolina at Chapel Hill: Department of Computer Science, 1974. 127 p.
34. *Закревский А. Д.* Язык программирования ЛЯПАС-М // Вычислительная техника в машиностроении. Минск: Ин-т техн. кибернетики АН БССР, 1974. С. 99–111.
35. *Закревский А. Д.* Сообщение о школе-семинаре, посвященной рассмотрению системы ЛЯПАС-71 (Томск, 1972) // Вопросы кибернетики. Теория релейных устройств и конечных автоматов. М.: ВИНТИ, 1975. С. 35–37.
36. *Томашев В. Ф.* Отладочная интерпретирующая шаговая трассировочная программа для языка ЛЯПАС-М // Управляющие системы и машины. 1975. № 1. С. 29–31.
37. *Закревский А. Д., Балаклей Л. И., Елисеева Н. А. и др.* Синтез асинхронных автоматов на ЭВМ. Минск: Наука и техника, 1975. 184 с.

38. *Закревский А. Д., Поляков А. С.* О повышении качества машинных программ путем оптимизации использования регистров // *Управляющие системы и машины.* 1976. № 6. С. 105–111.
39. *Закревский А. Д., Торопов Н. Р.* Программное обеспечение разработки диалоговой системы логического синтеза // *Дискретные системы.* Т. 1. Дрезден, 1977. С. 164–173.
40. *Закревский А. Д., Торопов Н. Р.* Система программирования ЛЯПАС-М. Минск: Наука и техника, 1978. 240 с.
41. *Tratnik I.* Seminar «Analiza in primerjava jezikov za podro'je digitalne tehnike». Ljubljani: Univerzav, 1979. 63 с.
42. *Романов В. И., Торопов Н. Р.* Организация информационного обмена в ДИСМО // *Вычислительная техника в машиностроении.* Вып. 1. Минск: Ин-т техн. кибернетики АН БССР, 1979. С. 122–128.
43. *Торопов Н. Р.* Организация взаимодействия мониторов в ДИСМО // *Вычислительная техника в машиностроении.* Вып. 1. Минск: Ин-т техн. кибернетики АН БССР, 1979. С. 129–135.
44. *Быкова С. В., Головчинер М. Н.* Инструкция к пользованию системой программирования ЛЯПАС-М для ЕС ЭВМ // *Алгоритмы решения задач дискретной математики.* Томск: Изд-во Том. ун-та, 1979. С. 4–13.
45. *Закревский А. Д.* О пакете программ ЛОГИКА-I // *Алгоритмы решения логикокомбинаторных задач.* Вып. 5. Минск: Ин-т техн. кибернетики АН БССР, 1979. С. 3–9.
46. *Томашев В. Ф.* Об опыте разработки, развертывания и тестирования транслятора и системы ЛЯПАС-М // *Автоматизация логического проектирования дискретных устройств.* Вып. 2. Минск: Ин-т техн. кибернетики АН БССР, 1980. С. 127–134.
47. *Торопов Н. Р.* Система программирования ЛЯПАС-М в рамках ОС ЕС ЭВМ // *Автоматизация логического проектирования дискретных устройств.* Вып. 2. Минск: Ин-т техн. кибернетики АН БССР, 1980. С. 99–104.
48. *Торопов Н. Р.* Унификация диалоговых языков в системе ЛЕС // *Автоматизация логического проектирования дискретных устройств.* Вып. 2. Минск: Ин-т техн. кибернетики АН БССР, 1980. С. 105–110.
49. *Романов В. И.* О совместимости языка ЛЯПАС-М с другими языками программирования // *Автоматизация логического проектирования дискретных устройств.* Вып. 2. Минск: Ин-т техн. кибернетики АН БССР, 1980. С. 121–126.
50. *Романов В. И., Торопов Н. Р.* Простой способ планирования в диалоговых системах // *Материалы семинара «Диалог в автоматизированных системах».* М., 1981. С. 126–134.
51. *Автоматизация проектирования цифровых устройств /* Под ред. С. С. Бадулина. М.: Радио и связь, 1981. 238 с.
52. *Торопов Н. Р.* Комплексная отладка программ на языке ЛЯПАС-М // *Управляющие системы и машины.* 1981. № 2. С. 91–101.
53. *Торопов Н. Р.* Динамическая сегментация в программирующей системе ЛЯПАС-М // *Управляющие системы и машины.* 1981. № 3. С. 67–70.
54. *Погарцев А. Г., Уткин А. А.* Экспериментальная диалоговая система ЭДА // *Автоматизация анализа и моделирования логических сетей.* Минск: Ин-т техн. кибернетики АН БССР, 1981. С. 17–30.
55. *Енин С. В., Василенок В. К.* Структура и принципы построения диалоговой системы ДИМОД-1 для моделирования дискретных устройств с неисправностями // *Автоматизация анализа и моделирования логических сетей.* Минск: Ин-т техн. кибернетики АН БССР, 1981. С. 129–137.

56. *Черемисинов Д. И.* Инструментальный комплекс для разработки программного обеспечения микропроцессоров // Управляющие системы и машины. 1981. № 5. С. 68–71.
57. *Томашев В. Ф.* Мобильный ЛЯПАС-М – транслятор (вариант для БЭСМ-6). Минск: Ин-т техн. кибернетики АН БССР, 1982. 105 с.
58. *Закревский А. Д., Енин С. В., Поттосин Ю. В.* Диалоговый пакет программ для автоматизации логического проектирования дискретных устройств // Автоматизация проектирования систем управления. Вып. 4. М.: Финансы и статистика, 1982. С. 29–40.
59. *Белюсова Н. А.* Транслятор с алгоритмического языка ЛЯПАС-М для ЕС ЭВМ // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 94–111.
60. *Поляков А. С.* Синтаксический контроль программ в системе ЛЕС // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 73–80.
61. *Кириченко Н. А.* Оптимизирующий транслятор с языка ЛЯПАС-М для ЕС ЭВМ // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 81–93.
62. *Романов В. И.* М-компиляция в системе программирования ЛЕС // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 62–72.
63. *Романов В. И.* Использование ЛЯПАС-М-программ из программ на других языках программирования // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 119–126.
64. *Черемисинов Д. И.* Транслятор языка регулярных выражений // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 127–134.
65. *Быкова С. В., Головчинер М. Н., Жуковский О. И. и др.* ТОМЛЕС — система программирования с базовым языком ЛЯПАС-М для ЕС ЭВМ // Программирование. 1984. № 5. С. 56–63.
66. *Торопов Н. Р., Романов В. И.* Система программирования ЛЕС 4 (инструкционно-методические материалы). Минск: Ин-т техн. кибернетики АН БССР, 1984. 134 с.
67. *Закревский А. Д., Бибило П. Н., Дудкин А. А., Шнейдер А. А.* Пакет программ для решения комбинаторных задач над булевыми и троичными матрицами // Управляющие системы и машины. 1984. № 3. С. 115–118.
68. *Торопов Н. Р.* Диалоговая система программирования ЛЕС. Минск: Наука и техника, 1985. 263 с.
69. *Торопов Н. Р.* Диалоговое программирование в языке ЛЯПАС-М // Кибернетика. 1986. № 1. С. 25–30.
70. *Торопов Н. Р.* «Уборка мусора» в системе ЛЕС // Проектирование систем логического проектирования. Минск: Ин-т техн. кибернетики АН БССР, 1986. С. 91–96.
71. *Торопов Н. Р.* Динамическая сегментация данных в системе ЛЕС. Минск, 1987. 12 с. / Препринт Ин-та техн. кибернетики АН БССР № 24.
72. *Закревский А. Д., Василенок В. К., Черемисинов Д. И., Черемисинова Л. Д.* Система Логика-М синтеза управляющих устройств в базисе ПЛМ и микропроцессоров // Управляющие системы и машины. 1987. № 3. С. 31–35.
73. *Белюсова Н. А., Питосин В. Т.* Реализация ЛЯПАС-М на СМ ЭВМ // Программирование. 1988. № 4. С. 106–108.

74. *Торопов Н. Р.* Диагностика программ в режиме микродиалога. Минск, 1988. 16 с. / Препринт Ин-та техн. кибернетики АН БССР № 12.
75. *Черемисинова Л. Д., Василенок В. К., Желудко Е. В., Красильникова Л. В.* Система логического синтеза устройств управления на базе программируемых контроллеров. Минск: Ин-т техн. кибернетики АН БССР, 1988. 100 с.
76. *Дудкин А. А., Поттосин Ю. В., Сичичка А. А., Черемисинова Л. Д.* Комплекс программ синтеза комбинационных схем в базисе ПЛМ и МЛВ. Минск: Ин-т техн. кибернетики АН БССР, 1988. 68 с.
77. *Торопов Н. Р., Шестаков Е. А.* Инструментальная система логического проектирования. Минск: Ин-т техн. кибернетики АН БССР, 1989. 70 с.
78. *Торопов Н. Р., Романов В. И., Кириенко Н. А. и др.* Система программирования ЛЕС-7. Минск: Ин-т техн. кибернетики АН БССР, 1990. 64 с.
79. *Томашев В. Ф.* Перенос ЛЯПАС-М-предтранслятора с ЕС ЭВМ на ЕС 1840 // Автоматизация логического проектирования дискретных устройств. Минск: Ин-т техн. кибернетики АН БССР, 1991. С. 63–71.
80. *Закревский А. Д., Торопов Н. Р.* Проектирование систем логического управления // Фундаментальные и поисковые исследования в интересах обороны страны. М.: Министерство обороны СССР, 1991. Вып. 73–74. С. 36–43.
81. *Торопов Н. Р.* Организация диалога через иерархию вложенных меню в ИСАПР // Автоматизация логического проектирования дискретных систем. Минск: Ин-т техн. кибернетики АН БССР, 1991. С. 81–95.
82. *Торопов Н. Р.* Формирование составных проектных операций в ИСАПР // Формализация и автоматизация логического проектирования. Минск: Ин-т техн. кибернетики НАН Беларуси, 1993. С. 193–146.
83. *Торопов Н. Р.* Исследовательская САПР дискретных управляющих устройств // Материалы по математическому обеспечению ЭВМ. Минск: Ин-т техн. кибернетики НАН Беларуси, 1994. 60 с.
84. *Торопов Н. Р.* Тренажер решения логических задач методом комбинаторного поиска. Минск, 1994. 16 с. / Препринт Ин-та техн. кибернетики НАН Беларуси № 22.
85. *Поспелов Д. А.* Школа МАГа // Новости искусственного интеллекта. 1997. № 3. С. 80–130.
86. *Романов В. И., Василькова И. В.* Булевы векторы и матрицы в С++ // Логическое проектирование. Вып. 2. Минск: Ин-т техн. кибернетики НАН Беларуси, 1998. С. 150–158.
87. *Черемисинов Д. И., Черемисинова Л. Д.* Троичные векторы и матрицы // Логическое проектирование. Вып. 3. Минск: Ин-т техн. кибернетики НАН Беларуси, 1998. С. 146–155.
88. *Закревский А. Д., Торопов Н. Р.* Полиномиальная реализация частичных булевых функций и систем. Минск: Ин-т техн. кибернетики НАН Беларуси, 2001. 200 с.