

# Обоснование 3.2В

## Семантический ассоциативный компьютер

В настоящее время в ходе расширения областей применения ЭВМ имеет место тенденция более быстрого роста объема неарифметических вычислений (т.е. объема переработки нечисловой информации) по сравнению с ростом объема вычислений арифметического характера. Кроме того, ощущается острая потребность в повышении уровня “интеллекта” различных систем переработки информации, т.е. потребность в расширении их логических возможностей. Отмеченные тенденции проявляются в автоматизированных системах управления, системах принятия решений, системах управления поведением очерствевших роботов, системах структурного распознавания образов, системах автоматизированного проектирования, автоматизированных обучающих системах, системах обслуживания неподготовленных пользователей ЭВМ, системах автоматического перевода, диалоговых информационно-справочных и информационно-логических системах, оперирующих сложными базами знаний и обеспечивающих взаимодействие с пользователем на языке, близком к естественному. Таким образом, в настоящее время определился очень важный для практики класс задач, предполагающих переработку нечисловой сложноструктурированной информации и допускающих при этом отсутствие точного алгоритма их решения. Такие задачи, следуя ряду авторов [Видоменко В.П. О структурной интерпретации языков информационно-логического программирования. – Программирование, 1980, №5, с. 56-62], будем называть информационно-логическими и комбинаторными задачами по переработке сложноструктурированных баз данных или, более кратко, просто информационно-логическими задачами. Заметим также, что понятие информационно-логических задач фактически совпадает по смыслу с широко используемым в последнее время понятием задач искусственного интеллекта, что позволяет использовать оба эти термина.

Разработка средств решения задач того или иного класса в настоящее время обычно осуществлявшаяся путем создания языка программирования высокого уровня, ориентированного на этот класс задач, и путем реализации такого языка на современных ЭВМ, т.е. путем создания транслятора. Для задач искусственного интеллекта в настоящее время предложено довольно много языков программирования [132, с. 58–59, 102–103; 24; 162]. Большинство из них построено путем расширения используемых в настоящее время языков программирования. Обычно в качестве такого базового языка выбирается язык LISP. Но, поскольку логика решения задач искусственного интеллекта плохо согласуется с современными языками программирования (в том числе и с языком LISP), более целесообразной является разработка принципиально новых языков, отражающих на уровне их элементарных операций основные логические механизмы решения задач рассматриваемого класса. Такие языки программирования обычно называют языками сверх-высокого уровня или непроцедурными языками [101]. Реализация языков сверх-высокого уровня на современных ЭВМ представляется весьма сложной в силу большого разрыва между языками этого класса и внутренними языками современных ЭВМ, для преодоления которого создание эффективного транслятора оказывается практически невозможным.

Таким образом, состояние проблемы автоматизации решения задач искусственного интеллекта (информационно-логических задач) в настоящее время можно охарактеризовать тем, что эта проблема входит в противоречие с принципами логической организации современных ЭВМ и, в первую очередь, с используемыми в современных ЭВМ внутренними языками. Наглядным подтверждением этому является то, что построенные на базе современных ЭВМ автоматизированные системы, осуществляющие переработку больших сложноструктурированных баз данных и имеющие высокоразвитые логические возможности, во-первых, требуют огромных программистских усилий, во-вторых, оказываются весьма громоздкими и, следовательно, ненадежными и, в-третьих, являются медленнодействующими, что полностью

исключает возможность их использования в реальном масштабе времени, например, для управления оцувствленными интеллектуальными роботами. Поскольку отмеченное противоречие между возможностями современных ЭВМ и острой практической потребностью в указанных выше автоматизированных системах обусловлено плохой приспособленностью современных ЭВМ к такого рода использованию, становится актуальным создание принципиально нового класса нетрадиционных машин по переработка дискретной информации, специально предназначенных для этой цели. Очевидно, что эффективность этих машин будет прежде всего определяться степенью близости их внутреннего языка к языкам непроцедурного типа (к языкам сверх-высокого уровня). Учитывая указанное назначение таких машин, их естественно назвать не вычислительными машинами, а математическими машинами или даже “мыслительными” машинами.

Разработка и исследование архитектурной и логической организации высокопроизводительных машин, ориентированных на решение информационно-логических задач, т.е. хорошо приспособленных к переработке нечисловой сложноструктурированной информации и осуществляющих аппаратную интерпретацию непроцедурного языка программирования, специально созданного для этой цели. Разработка такого непроцедурного языка, специально предназначенного для аппаратной интерпретации, является основной частью данного проекта, ибо разработка машин, ориентированных на решение информационно-логических задач, в проекте методологически строится “от языка”, что означает отсутствие каких бы то ни было ограничений, изначально накладываемых на внутренний язык и обусловленных ориентацией на те или иные технические средства, на ту или иную элементную базу.

Прежде чем сформулировать основные задачи, решения которых требует цель данного проекта, кратко охарактеризуем развиваемый нами подход к достижению этой цели.

Анализ причин плохой приспособленности традиционной, восходящей еще к фон Нейману архитектуры и логической организации современных ЭВМ к решению информационно-логических задач позволяет сделать следующие выводы:

- в современных ЭВМ при работе со сложноструктурированными базами данных время, затрачиваемое на информационный поиск, на 2–3 порядка превышает время, затрачиваемое на собственно переработки;
- в современных ЭВМ имеет место весьма низкий уровень аппаратно реализуемых операций над нечисловыми данными;
- представление информации в памяти современных ЭВМ имеет уровень весьма далекий от семантического, что делает переработку информации довольно громоздкой, требующей учета большого количества деталей, касающихся не смысла перерабатываемой информации, а способа ее представления в памяти;
- в современных ЭВМ громоздка реализуются даже простейшие процедуры логического вывода.

Перечисленные причины плохой приспособленности современных ЭВМ к решению информационно-логических задач, по существу, не устраняются также и в развиваемых в настоящее время подходах к построению нетрадиционных высокопроизводительных ЭВМ, ибо, в основном, все эти подходы (даже если они достаточно далеко отходят от предложенных фон Нейманом принципов организации вычислительных машин) неявно сохраняют точку зрения на ЭВМ как на большой арифмометр и тем самым сохраняют ее ориентацию на задачи числового характера. В качестве примера, подтверждающего эту мысль, можно отметить то, что потоковый принцип обработки информации (управление потоком данных или потоком запросов), всеми (2 признанный как наиболее перспективный принцип распараллеливания, до сих пор не был исследован для случая данных нечислового характера и сложной структуры [3, с. 158].

Анализ причин непригодности современных ЭВМ к решению информационно-логических задач, т.е. критический анализ положенных в их основу принципов фон Неймана, является предметом исследования целого ряда работ [I42, с. 3-10; 47, с. 9; дд, с. 169-171, 178-180; 37; 45; 33, с. 58-61} 112; 135; 136; 79, с. 4; 2, 6. 5-7, 11-16, 31, 36; 95, с. 7-8, 80-81; 31; 152, 0. 3-16; 34, с. 56-57; I49, 0. 322-326; 76].

Сутью развиваемого в данной работе подхода к построению машин, ориентированных на решение информационно-логических задач, является стремление так организовать процесс переработки информации, чтобы он был наиболее близок к семантическому (содержательному) уровню. Очевидно, что такой подход требует, во-первых, разработки семантического способа представления перерабатываемой информации в памяти машины и, во-вторых, разработки такого внутреннего языка, запись программ на котором была бы максимально близка к тому, что называют записью алгоритма на содержательном уровне. Исследования по системам искусственного интеллекта убедительно показали, что способ представления знаний в их памяти, точнее степень его близости к семантическому, является фактором, во многом определяющим эффективность таких систем [I18; 132, с. 38-40; I54, с. 245-249; 42. с. 15-18].

Анализ задач всевозможных видов позволяет сделать следующие два вывода:

- все задачи допускают естественную и удобную теоретико-графовую трактовку, т.е. все задачи можно рассматривать как задачи на переработку графовых структур того или иного вида;
- процесс решения любой задачи можно трактовать как процесс логического вывода в некоторой формальной теории.

Об удобстве теоретико-графовых методов свидетельствует интенсивное расширение областей их применения [98, с. 7; 38; 141, с. 351; 83, с. 477-501], а также активная

разработка специальных языковых [109; 144], программных [12] и аппаратных [165] средств. Кроме традиционных приложений в органической химии (при анализе молекулярных структур), в электротехнике (при анализе электрических сетей), теоретико-графовые методы в настоящее время широко используются в социологии и экономике, в теории вероятностей (например, при анализе цепей Маркова), в генетике [119]. в математической лингвистике, в исследовании операций, в статистической и теоретической физике, в проектировании дискретных устройств [115]. Об эффективности теоретико-графовых методов говорят также результаты, полученные при разработке структурного подхода к распознаванию образов, который позволяет сократить сложность процедуры распознавания по сравнению с другими подходами и обеспечивает распознавание в тех случаях, когда другие подходы оказываются неприемлимыми [153]. Как считается в работе [11, с. 3], “возможность приложения теории графов к столь различным областям заложена, в сущности, уже в самом понятии графа, сочетающего в себе теоретико-множественные, комбинаторные и топологические аспекты”. Говоря о теоретико-графовой трактовке различных задач, необходимо подчеркнуть то, что такая трактовка всегда неявно подразумевает формализацию перерабатываемых данных не просто в виде некоторой графовой структуры, а в виде такой графовой структуры, которая обладает вполне определенными семантическими свойствами, а именно: её вершины однозначно соответствуют понятиям предметной области, в которой задана задача, а её дуги, ребра, гиперребра и т.д. соответствуют связям между указанными понятиями. В теории искусственного интеллекта такие графовые структуры называют семантическими сетями [161], а также ассоциативными сетями, смысловыми графами, семантическими представлениями, семантическими графами, концептуальными графами, концептуальными моделями данных, сетями концептуализаций, сетями концептуальной зависимости.

Итак, теоретико-графовая трактовка задач по сути дела есть не что иное, как формализация перерабатываемых в задаче данных в виде семантических сетей. Естественность теоретико-графовой трактовки задач определяется в первую очередь имен – но этим обстоятельством и обуславливает то, что семантические сети являются весьма удобным способом семантического представления информации. Последнее подтверждается также и в работах по теории искусственного интеллекта, в которых семантические сети используются как основа формального аппарата исследования, формализации и моделирования сложных (“интеллектуальных”) процессов переработки информации, таких, как доказательство теорем, распознавание образов, планирование действий, обнаружение закономерностей [159; 100; 70]. В заключение рассмотрения теоретико-графовой трактовки задач отметим, что обеспечение возможности такой трактовки для любой задачи требует обобщения классического понятия графа. Такое обобщение прежде всего необходимо для того, чтобы иметь возможность вершины графа связывать не только бинарными связями, но и связями произвольной арности.

Теперь перейдем к рассмотрению второго вывода, сделанного на основе анализа всевозможных задач. Трактовка процесса решения задачи как процесса логического вывода осуществляется в непроцедурных языках программирования типа PROLOG [175; 173; 176]. Языки этого класса строятся на базе логических языков (обычно языков первого порядка), т.е. являются расширениями этих языков. Программы, написанные на языках типа PROLOG, представляют собой специальным образом устроенные формальные теории в соответствующих логических языках, а элементарные операции при их реализации представляют собой процессы выполнения различных правил логического вывода.

Выводы, сделанные на основе анализа задач всевозможных видов, позволяют сформулировать общие принципы построения машин, ориентированных на решение информационно–логических задач:

- такие машины целесообразно строить как машины,

манипулирующие графовыми структурами непосредственно на физическом уровне;

- в качестве внутреннего языка таких машин целесообразно использовать язык типа PROLOG.

Первый из указанных принципов предполагает создание структурно-перестраиваемой (“графовой”) памяти. Такая память состоит из ячеек, однозначно соответствующих вершинам хранимого в памяти графа, но, в отличие от обычной памяти, эти ячейки связываются не фиксированными условными связями, задающими фиксированную последовательность (порядок) ячеек в памяти, а реально (физически) проводимыми связями произвольной конфигурации. Эти связи соответствуют дугам, ребрам, гиперребрам записанного в памяти графа. Очевидно, что в ходе переработки информации в структурно-перестраиваемой памяти меняются не только и даже не столько состояния ячеек памяти, как это имеет место в обычной памяти, сколько конфигурация связей между этими ячейками. Т.е. в структурно-перестраиваемой памяти в ходе переработки информации не только перераспределяются метки на вершинах записанного в памяти графа, но и меняется структура самого этого графа.

Разработка языка типа PROLOG, предназначенного к использованию в качестве внутреннего языка программирования для машин со структурно-перестраиваемой памятью, требует решения нетривиальной задачи согласования графового способа представления данных в структурно-перестраиваемой памяти и способа записи в этой же памяти самих программ, описывающих переработку этих данных. В предлагаемом проекте указанная задача решается путем разработки графового варианта языка типа PROLOG, обеспечивающего запись программ в виде специальным образом устроенных графовых структур. Переход на “графовый” способ кодирования программ и данных в структурно-перестраиваемой памяти обеспечивает компактность их представления и существенно упрощает аппаратную реализацию операций над сложными структурами. Говоря об аппаратной интерпретации языка типа PROLOG, необходимо подчеркнуть



следующее. На уровне любого языка типа PROLOG, т.е. на уровне абстрактной PROLOG – машины, естественным образом реализуется эффективное распараллеливание процесса переработки сложных структур, организованное по принципу управления потоком запросов или управления потоком перерабатываемых сложноструктурированных данных. Управление потоком сложноструктурированных данных при этом основывается на использовании развитой формы ассоциативного доступа, а именно, доступа к произвольному фрагменту перерабатываемых данных (фрагменту, имеющему произвольный размер и произвольную структуру). Из вышесказанного следует, что создание машины, аппаратно интерпретирующей язык типа PROLOG, есть не что иное, как создание параллельной машины, управляемой потоком сложноструктурированных данных и имеющей развитую ассоциативную память.

При постановке задачи на техническую реализацию структурноперестраиваемой памяти вполне естественно выглядит желание упростить эту задачу путем упрощения синтаксиса внутреннего языка, т.е. путем сведения всевозможных графовых структур, представляющих как данные, так и программы, к структурам некоторого частного вида. В проекте в качестве внутреннего языка разрабатываемой машины предлагается язык именно с таким упрощенным синтаксисом.

Следующим используемым в данном проекте принципом построения машины, ориентированной на решение информационно-логических задач, является принцип организации переработки информации непосредственно в памяти [121]. Этот принцип обеспечивает значительное ускорение переработки информации благодаря исключению этапов передачи информации из памяти в процессор и обратно, но оплачивается ценой большой избыточности функциональных (процессорных) средств, равномерно распределяемых по памяти. При распределении функциональных средств по структурно-перестраиваемой памяти каждая ячейка дополняется функциональным (процессорным) элементом, а перестраиваемые связи между ячейками становятся

коммутируемыми

каналами связи между функциональными элементами. Каждый функциональный элемент при этом имеет свою специальную внутреннюю регистровую память, отражающую важные для данного функционального элемента аспекты текущего состояния процесса выполнения элементарных операций внутреннего языка. Рассмотренный архитектурный принцип позволяет предлагаемые в проекте машины, ориентированные на решение информационно-логических задач, назвать Однородными параллельными Структурами, осуществляющими Переработку Семантических Сетей (сокращенно – ОСПСС).

К числу систем, в той или иной мере похожих на предлагаемую машину для решения информационно–логических задач, можно отнести:

- машины с развитой ассоциативной памятью [95; 96; 152];
- ассоциативные параллельные матричные процессоры типа STARAN [а, с. 15–16];
- машины с аппаратурной интерпретацией сложных структур данных [168; 169; 170];
- машины, управляемые потоком данных [3, 0. 73-103; 136];
- рекурсивные вычислительные машины [142];
- процессоры реляционных баз данных [79];
- предложенные Л.С.Берштейном с соавторами однородные параллельные структуры для решения комбинаторно-логических задач на графах и гиперграфах [18];
- всевозможные устройства переработки графов [165; 122, с. 28–34];
- абстрактная параллельная машина А.М.Степанова [146; 147];
- абстрактная вегетативная машина на клубных системах, предложенная В.Б.Борщевыи [23];
- вычислительные машины со структурно-перестраиваемой памятью [138; 139; 42, с. 15-19, 30];

- системы, осуществляющие переработку информации непосредственно в памяти путем равномерного распределения функциональных средств по памяти [I21] и, в частности, предложенная М.Н. Вайнцвайгом процессоропамять, ориентированная на решение задач искусственного интеллекта [32];
- активные семантические сети (М-сети) [4];
- ассоциативные однородные среды, предложенные И.П. Кузнецовым и Е.В. Золотовым [82];
- нейроподобные структуры [120].

К числу работ, составляющих теоретическую основу развиваемого нами подхода к построению аппаратурного интерпретатора языка типа PROLOG, можно отнести:

- работы по алгоритмам Колмогорова [92; 68; 69];
- теоретические исследования абстрактных машин с модифицируемой памятью [I81];
- работы по графовым грамматикам и графовым автоматам [129; 177];
- работы Ю.Г. Гостева по использованию аппарата графовых грамматик для описания сложноструктурированных данных и для описания семантики программ [65; 66; 67];
- работы по непроцедурным языкам программирования [101; I48; I9; 88];
- работы В.Б. Борщева, развивающие теоретико-модельный (непроцедурный) стиль программирования на клубных системах (на структурах данных весьма сложного вида) [21; 23];
- работы по реляционным моделям баз данных [22; 158; 10];
- работы по семантическим сетям, в особенности, работы по операционной семантике семантических сетей [179; 180; 161; 155; 130; I72; 13#, с. 57-59; 26; 91, с. 22, 45-94; 35; 132, с. 60-62, 118-140; 154, 0. 449-454; ч., с. 19-31; 102].

Необходимо подчеркнуть, что сама постановка задачи на проектирование аппаратурного интерпретатора

непроцедурного языка программирования стала возможной только сейчас, когда микроэлектронная технология достигла достаточно высокого уровня. Это тем более справедливо для предлагаемого в данном проекте подхода к построению такого интерпретатора, подхода, который требует существенного усложнения логики памяти, а, точнее, равномерного распределения по памяти всех функциональных и управляющих средств, т.е. превращения памяти в процессоро-память.

Итак, для достижения поставленной цели в данном проекте решаются следующие основные задачи:

- разработка и исследование способов семантического представления информации различного вида;
- разработка и исследование принципов организации развитой ассоциативной памяти для непосредственного хранения семантического представления информации;
- разработка и исследование языка программирования высокого уровня, который (1) ориентирован на решение информационно-логических задач; (2) обеспечивает непосредственную реализацию простых процедур логического вывода. (3) согласован с выбранным способом семантического представления перерабатываемой информации, (4) приспособлен к использованию в качестве внутреннего языка параллельной однородной структуры, имеющей распределенную ассоциативную память для хранения сложноструктурированных данных;
- разработка и исследование принципов построения и, принципов параллельного взаимодействия функциональных средств, обеспечивающих непосредственную переработку семантического представления информации в распределенной ассоциативной памяти и реализующих управление потоком словноструктурированных данных;
- экспериментальная проверка полученных результатов.

Методы исследования в данном проекте основаны на сочетании теории множеств, теории алгебраических систем, теории графов,

математической логики, теории вычислений, методов машинного моделирования.

Основные положения проекта:

- теорию наиболее общего вида структур данных (квази-графов), используемых в задачах искусственного интеллекта и являющихся обобщением классических алгебраических моделей;
- ориентированный на аппаратную интерпретацию способ кодирования сложноструктурированных данных (квази-графов) в виде однородных семантических сетей специального вида;
- новый логический язык, являющийся модификацией классического и обеспечивающий описание квази-графов;
- графовые варианты логического языка описания квази-графов и, в частности, ориентированный на аппаратную интерпретацию способ кодирования логических формул в виде однородных семантических сетей;
- непроцедурный язык программирования типа PROLOG, отличающийся удобством работы со сложными структурами данных (квази-графами) и развитостью средств управления вычислительным процессом;
- предлагаемый в качестве внутреннего аппаратно интерпретируемого языка программирования способ записи непроцедурных программ в виде однородных семантических сетей;
- архитектуру и принципы организации однородной ассоциативной параллельной структуры, ориентированной на переработку семантических сетей и обеспечивающей аппаратную интерпретацию непроцедурного языка программирования типа PROLOG.

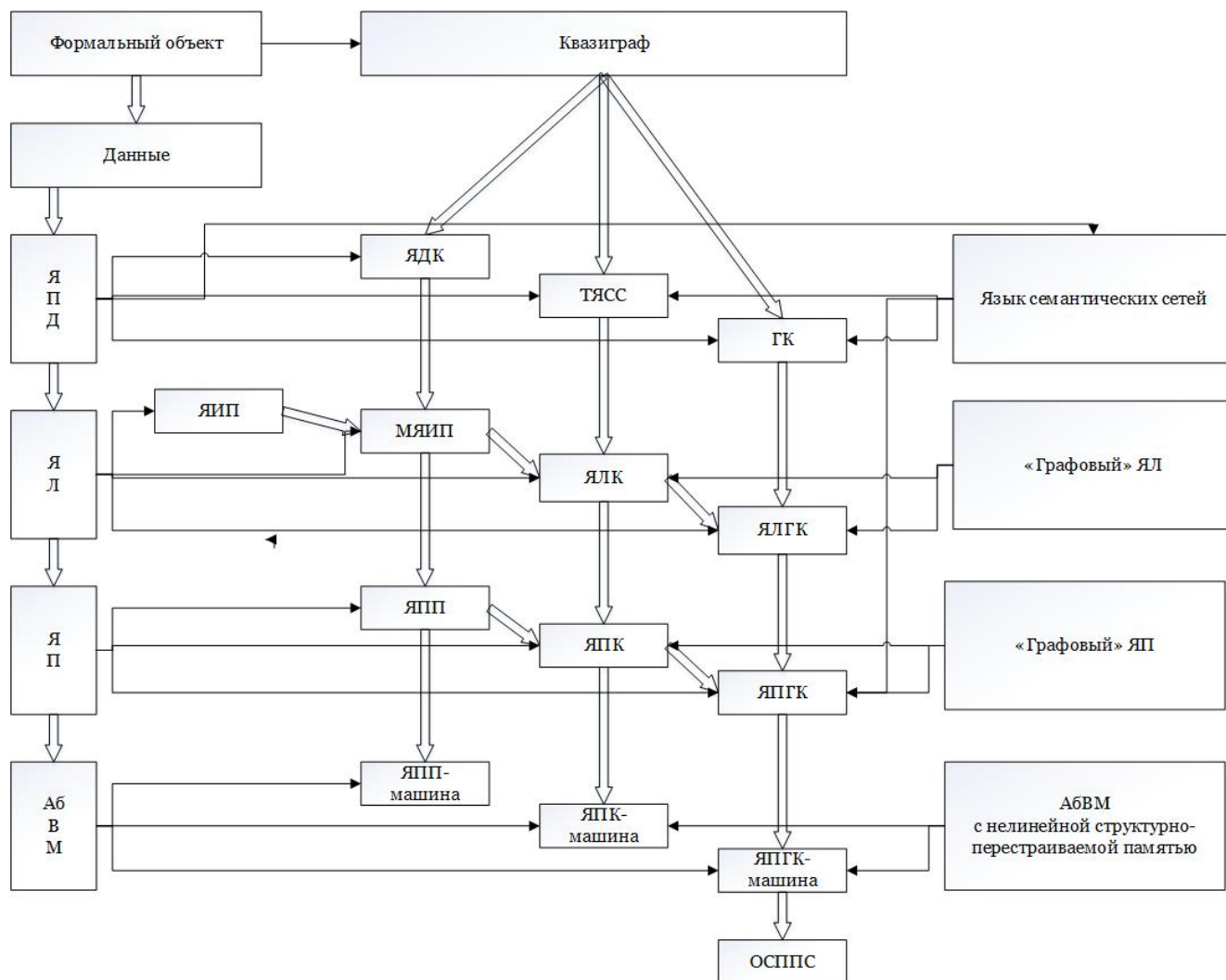


Диаграмма основных используемых понятий

ЯПД – Язык Представления Данных

ЯДК – Язык Диаграмм Квазиграфов

ГК – Графовый Код

ЯЛ – Язык Логики

ТЯСС – Тривиальный Язык Семантических Сетей

ЯИП – Язык Исчисления Предикатов (классический)

МЯИП – Модифицированный Язык Исчисления Предикатов

ЯЛК – Язык Логики Квази-графов

ЯЛГК – Язык Логики текстов Графового Кода

ЯП – Язык Программирования

ЯПП – Язык Программ на Предикатах

ЯПК – Язык Программ на Квази-графах

ЯПГК – Язык Программ на текстах Графового Кода

АБВМ – Абстрактная Вычислительная Машина

ОСПС – Однородная Структура для Переработки Семантических

## Сетей

Совокупность полученных результатов составляет основу, необходимую для построения формальных моделей и исследования нового класса вычислительных машин, что открывает новое перспективное направление.

Научная новизна проекта. Впервые исследованы пути построения параллельных машин, управляемых потоком сложноструктурированных данных. В качестве памяти таких машин впервые предложена структурно-перестраиваемая запоминая среда, обеспечивающая непосредственное хранение графовых структур и манипулирование ими, а также обеспечивающая ассоциативный

доступ к произвольным фрагментам перерабатываемых графовых структур (фрагментам, имеющим произвольный вид и произвольный размер). Впервые проведено исследование путей и принципов аппаратной интерпретации непроцедурных языков программирования типа PROLOG. В качестве интерпретируемого (внутреннего) языка впервые предложен и исследован язык программирования графового типа, являющийся способом записи программ в виде однородных семантических сетей.

Практическая ценность проекта. Результаты работы могут быть использованы при выборе внутреннего языка, способа организации памяти и архитектуры вычислительных машин 5-го поколения, ориентированных на решение задач искусственного интеллекта и приспособленных к переработке сложноструктурированных данных. Результаты работы могут служить основой для формулирования заданий на опытно-конструкторские разработки по созданию вычислительных машин указанного класса. Использование результатов работы позволит:

- существенно расширить класс аппаратно интерпретируемых (непосредственно перерабатываемых) структур данных;
- обеспечить высокую скорость переработки сложноструктурированных данных, благодаря (1) “укрупнению” аппаратно реализуемых операций

преобразования структур данных, (2) глубокому распараллеливанию процесса переработки сложных структур как на программном, так и на микропрограммном уровне, (3) организации переработки информации непосредственно в памяти;

- существенно расширить логические возможности вычислительных машин благодаря использованию логического языка в качестве основы внутреннего языка программирования;
- обеспечить достаточно высокую технологичность вычислительных машин рассматриваемого класса благодаря их организации как однородных структур.