© 2012 г. В.И. БОДЯКИН, канд. физ.-мат. наук
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

ПРОБЛЕМЫ АЛГОРИТМИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ И АЛГОРИТМ САМООБУЧЕНИЯ[1]

Анализируются вычислительные процессы на нейросемантических структурах и сравниваются с обработкой информации в многопроцессорных компьютерах. Рассматривается подход к построению алгоритма самообучения.

PROBLEMS OF ALGORITHMIZATION IN PARALLEL COMPUTING PROCESSES / V.I. Bodyakin, V.A. Gruzman (V.A. Trapeznikov Institute of Control Sciences, Profsoyuznaya 65, Moscow 117342, Russia, E-mail: body@ipu.ru). The computing processes on neurosemantical structures are analyzed and are compared to handling of the information in multiprocessor computers. The approach is considered to building of the algorithm of self-training.

1. Введение

Актуальность комплекса проблем, связанных с обработкой неструктурированной информации, экспоненциально нарастает и в ближайшее десятилетие угрожает глобальным системным кризисом. Вычислительное направление применения компьютеров всегда оставалось основным двигателем прогресса в компьютерных технологиях. Сегодня трудно переоценить роль высокопроизводительных вычислений, когда большинство сложных объектов проектируются виртуально и на первый взгляд очевидно, что число и производительность суперкомпьютеров должны и будут расти.

Показательно, что производительность самых первых компьютеров составляла всего около 100 операций в секунду, тогда как например на сегодняшний день, пиковая производительность отечественного суперкомпьютера МГУ «Ломоносов» после его модернизации выросла до 1700.21 TFlop/s и занимает 1-е место в СНГ и 12-е место в мировом рейтинге Top500 самых мощных высокопроизводительных систем. Т.е. произошло увеличение быстродействия в 17000 миллиардов раз. Невозможно назвать другую сферу человеческой деятельности, где прогресс был бы так велик. Причем примерно, 10^4-кратное увеличение скорости работы произошло за счет эволюционного развития электронных схем и 10^9-кратное – за счет максимально широкого распараллеливания обработки данных в супер-ЭВМ. (По вычислительной плотности на квадратный метр занимаемой площади 30 Тфлопс/м^2 «Ломоносов» превосходит все мировые аналоги.).

[1] Данная работа выполнена при частичной поддержке РФФИ, проект № 11-07-00007-а.

Если изначально основная задача суперкомпьютера заключалась в ускорении процесса виртуального моделирования, то теперь один из важных аргументов появления новой системы – успешное тестирование по Unpack с последующим попаданием в Top5OO и другие рейтинговые списки.

Большую часть времени современные суперкомпьютеры загружены небольшими задачами. Большие задачи, требующих для запуска нескольких тысяч процессорных ядер, и наиболее перспективные с точки зрения внутреннего параллелизма, представлены численными алгоритмами моделирования задач механики сплошных сред: расчеты течений жидкости и газа, метеорологические прогнозы. В которых, точность расчетов повышается путем перехода к более подробным дискретным моделям расчетных областей. Но такие задачи составляют лишь проценты от общего числа решаемых задач на суперкомпьютерах.

Кроме того, в проблеме повышения эффективности эксплуатации супер-ЭВМ, существует проблема подготовки исходных данных для крупномасштабных вычислительных экспериментов. Во многом это проблема существующих на сегодняшний день вычислительных алгоритмов, альтернативы которым пока нет. Поэтому на сегодняшний день, даже петафлопсная система как инструмент для решения одной важной задачи – пока не столь актуальна. А широкомасштабные вычислительные эксперименты с использованием десяти и более тысяч процессорных ядер, по сути, носят характер показательных мероприятий. В этой ситуации скептики считают дальнейший рост мощности суперкомпьютеров на практике бесполезным, когда намного проще управлять несколькими менее мощными компьютерами.

Как показывает практика, процесс разработки параллельной программы очень длителен и трудоемок, особенно когда к разрабатываемому параллельному программному обеспечению предъявляется требование его эффективности и мобильности. Это связано с тем, что универсальные средства, облегчающие труд программиста и обеспечивающие полноценный доступ к отладочной информации, на сегодняшний день находятся в стадии разработки. К тому же, проблема заключается в отсутствии стандартов в области создания и отладки программ для параллельных систем.

На сегодняшний день не существует универсальных средств адаптации программ к конкретной архитектуре супер-ЭВМ, поэтому большую часть этой проблемы приходится решать вручную, что делает процесс очень трудоемким [3]. Целью любой работы, связанной с параллельным программированием, является рассмотрение взаимосвязей между структурой математического алгоритма и архитектурой многопроцессорной вычислительной системы, которых большое разнообразие [3].

Резюмируя сложности программирования многопроцессорных систем отметим, что:

– относительно мало распараллеливаемых задач (например, логические задачи сугубо последовательны);

– отсутствие устоявшихся стандартов и норм параллельного программирования;

– сложность обучения и подготовки программистов для параллельного программирования;

– необходимо совмещение структуры алгоритма и вычислительной структуры супер-ЭВМ;

– эксплуатационная дороговизна Супер-ЭВМ;

– и др.

Несмотря на несомненные успехи в применении мультипроцессорных систем, имеет место недостаточная эффективность их практического использования. При разработке

реальных параллельных программ, как правило, достижение высокой эффективности требует многократных изменений программы для поиска наилучшей схемы ее распараллеливания. Успешность такого поиска определяется простотой модификации исходной программы.

2. Нейросемантика

2.1. Актуальность распараллеливания вычислительных процедур

Рассмотрим отдельно проблему распараллеливания вычислительного алгоритма. Исторически сложилось так, что развитие начального этапа вычислительной техники происходило в условиях острого дефицита вычислительных ресурсов, в частности, числа процессоров в ЭВМ. В основном, парк первых ЭВМ был однопроцессорным. Поэтому алгоритмы решения реальных задач предметной области, состоящих, естественно, из параллельных процессов, программистами вынужденно подстраивались под архитектуру однопроцессорных ЭВМ, что создавало им на том этапе определенные трудности.

С широким распространением в последнее время многопроцессорных ЭВМ, появилась возможность ускорить процесс решения задач и, соответственно, встала задача распараллеливания больших массивов накопленных алгоритмов, первоначально ориентированных на архитектуру однопроцессорных ЭВМ. Задача оказалась неавтоматизируемой и трудоемкой и до сих пор окончательно не решена.

Попытки распараллеливать алгоритм задачи, ранее сведенный к однопроцессорному варианту, аналогичны попыткам восстановления многомерного объекта по его неполным проекциям. Также, как и например, попытки функционального расширения алгоритмических «черно-белых» наработок с помощью различных эвристик до исходного «цветного» – практически невозможны.

Но если отбросить груз наработок и посмотреть на задачу распараллеливания заново, то мы увидим абсолютно корректное исходное «цветное изображение» задачи предметной области (ПО), готовое к многопроцессорной обработке. Далее дело за малым, за мощным инструментарием и средствами его настройки на параллельную обработку. Отметим, что на сегодня инструментария, предназначенного для целостной обработки такого сложного «цветного изображения», пока нет. Как показано в последних наших работах [1], в качестве инструментария предназначенного для обработки параллельных вычислений вполне может подойти прототип информационно-управляющей системы (ИУС) на базе нейросемантической[2] парадигмы (НСС).

Все естественные процессы произвольной предметной области причинно-связаны и, естественно, свободно параллельны. НСС представляет собой многодольный иерархический граф, в котором можно осуществлять максимальное распараллеливание задачи. При этом, одним из значимых свойств НСС является возможность автоматического формирования в ней графа, гомоморфного причинно-следственной структуре отображаемых физических процессов в задачах любой предметной области. Для этого необходимо лишь минимизировать суммарные ресурсные затраты памяти, расходуемые на формирование вершин (N-элементов) и дуг (связей) в НСС при отображении в ней не-

[2] *нейросемантика* – совокупность математических методов стягивающих взаимоотображения процессов (семантики) произвольной предметной области в нейроподобные элементы ИУС, т.е. «нейрон ↔ процесс».

ограниченного символьного информационного потока из произвольной ПО [2]. В результате автоматически формируется топологически гомоморфная информационная модель произвольной ПО, см. рис.1.

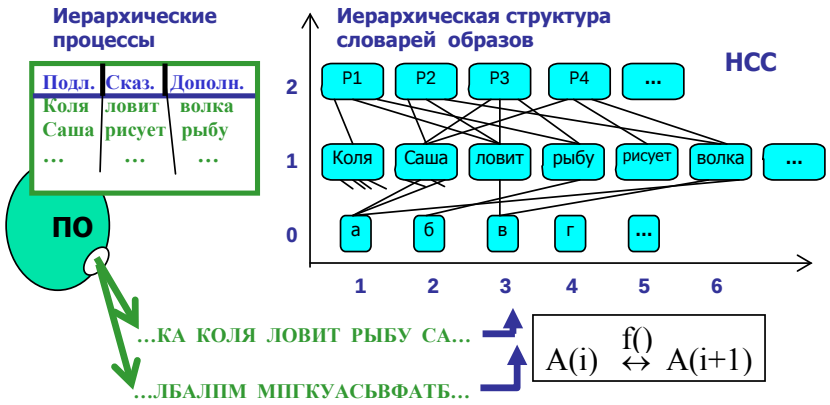

Рис. 1. Формирование многодольного иерархического графа НСС.

Отметим, что N-элементы в каждой доле графа НСС независимы и, соответственно, могут обрабатываться параллельно во всех вершинах-процессорах [2]. НСС представляет собой ассоциативную память. Поток входного сигнала, через связи терминального слоя (0, см. рис. 1) ассоциативно активирует N-элементы (образы-процессы) в 1-ом слое НСС. Последовательность активированных N-элементов 1-го слоя, в свою очередь, активирует N-элементы во 2-ом слое и т.д., пока ассоциативно не обработается весь поток входного сигнала (задача), который будет отображен активацией некоторого N-элемента, на некотором n-ом слое $НСС_S$, см. рис. 2.

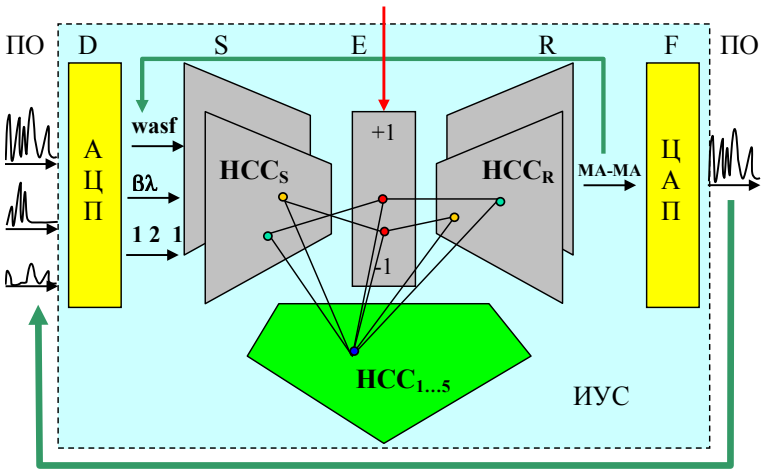

Рис. 2. Архитектура рекурсивной модели ИУС.

Отметим, что при числе процессоров (RISC – процессоры), равном максимальному числу N-элементов в слое $НСС_S$ (аналог аппаратной реализации ЭВМ), процесс распараллеливания условий задачи будет проходить естественно параллельно и, соответственно, автоматически параллельно. Если же, у нас ограниченное число процессоров, то в них, в порядке освобождения процессоров, будут загружаться полуактивированные

N-элементы. Обычно их число в реальных задачах не превышает – 20-50. В них будет загружена одна вычислительная программа и данные состояния конкретного N-элемента [1].

Аналогичные вычислительные процессы будут происходить и при обучении реакции рекурсивной модели ИУС, т.е. выборе N-элемента в $НСС_R$, см. рис. 2.

Таким образом, построенная из НСС архитектура рекурсивной модели ИУС (см. рис. 2), позволяет автоматически максимально возможно распараллеливать любую задачу, в частности, задачу распознавания образов. При этом процесс решения однотипных задач, при некоторой многократности их повторения, будет асимптотически сводиться к одноактному обращению к формируемому обобщающему нейрону-процессору в ассоциативной НСС-памяти.

Предлагаемый нейросемантический подход полностью укладывается в современную естественно-научную мировоззренческую парадигму и опирается на базовые философские установки и принципы. Это опора на всеобщее свойство отражения материи и ее интервальной параметрической устойчивости, а также, на физический принцип наименьшего действия и др. [1,2].

3. Подход к понятию самообучающегося алгоритма на базе активно-субъектного самоподдерживающегося процесса

3.1. Неформализованная постановка задачи

Как уже отмечалось во введении, современные проблемы информационной цивилизации выражаются в экспоненциальном нарастании потоков неструктурированной информации, которое, как следствие, ведет к усилению неуправляемости социально-экономическими процессами. Что в итоге, с большой вероятностью, приведет в ближайшее десятилетие к глобальному системному кризису [1]. Предлагается решение этого комплекса проблем путем создания ИУС на базе современных средств многопроцессорной вычислительной техники с реализацией в самой ИУС механизмов самообучения, которые бы функционально не уступали когнитивным возможностям человека. В качестве испытательного полигона для подобной ИУС предлагается в будущем рассмотреть следующие примеры.

1. Структуризация неструктурированных текстовых потоков на произвольном естественном языке, отражающем причинно-следственный характер некой совокупности описываемых процессов в некоторой открытой предметной области. При этом, задача ИУС заключается в разбиении (структуризации) текстовых потоков на текстовые фрагменты: описания задач и решения, на которых ИУС будет самообучаться поиску алгоритмов (в виде графовых структур), отображающих причинно-следственные связи между текстами задач и текстами решений. В результате обучения ИУС качество структуризации описания задач и прогнозирования ею решений (при их наличии в потоке) будет монотонно стремиться к 1.0.

После обучения ИУС сможет сама выдавать решения в текстовой поток, тем самым запуская новые причинно-следственные процессы и выполняя собственную управленческую функцию. Все причинно-следственные тройки: <задача><решение><оценка>, отображаемые в текстовом потоке, имеют цель максимизировать <оценку>. Из возможных вариантов <решений> ИУС будет предпочитать (выдавать) дающие максимальную <оценку>. Тем самым, при достаточной информационной мощности ИУС, она сможет

переломить современные нарастающие тенденции хаотизации информационных потоков и положительно разрешить назревающий системный кризис.

2. Структуризация экспериментальных потоков произвольных данных, отражающих причинно-следственный характер исследуемых процессов в некоторой открытой предметной области. В результате ИУС сформирует информационную модель исследуемых процессов данной предметной области в виде многодольного иерархического графа, отражающего причинно-следственные связи исследуемых процессов.

Кратко обсудим формализованную постановку задачи по формированию ИУС или *саморазвивающегося универсального алгоритма* на базе активно-субъектного самоподдерживающегося процесса (АСП). Предлагается в качестве базиса взять два понятия: *процесс (П)* и *целевую функцию (ЦФ)*, а в качестве базовых переменных: pt – пространство-время и e – энергия. При этом: $П = f_1(e, pt)$, $ЦФ = f_2(e, f_3(П))$, где: f_1, f_2, f_3 – есть П.

3.2. Формализованная постановка задачи

Базовые понятия: *процесс* и *целевая функция (ЦФ)*.

Процесс рассматривается как изменение *Е-параметра* ЦФ в *пространственно-временных координатах (Р-Т)*, т.е. в двух-трехмерном пространстве *(Р-Т,Е)*. Под процессами будем понимать процессы молекулярного уровня типа физического, химического или биологического характеров.

Каждый процесс отображается *цепочкой символов-знаков* (текстом) *алфавита А*. Текст процесса *отражает динамику* изменения его Е-параметра в *Р-Т–координатах*.

Все процессы имеют *причинно-следственную* и *иерархическую* природу.

Причинно-следственный характер процессов отображается в *принципе локального детерминизма*. Т.е. если в локальной области пространств (Р-Т,Е) наблюдается повторение параметров (p,t,e), то также с большой вероятностью будет наблюдаться и повторение пространственно-временного развития параметров (p,t+Δ,e), где Δ достаточно мало.

Принцип локального детерминизма имеет нарушения, т.к. при расширении локальной области пространств (Р-Т,Е) в реальных предметных областях всегда можно найти *не тождественность* параметров (p,t,e) предыдущим состояниям и, как следствие, *не повторение* временной динамики параметров (p,t+Δ,e) процессов.

Взаимодействие двух и более процессов отражает *иерархически более высокий* процесс данной предметной области.

Принцип локального детерминизма объединяет все процессы предметной области в иерархический *глобальный процесс процессов* (ГПП). ГПП – *открытая, но счетная* система процессов. Примеры ГПП: Вселенная, Земля, произвольная предметная область (ПО). Далее везде под обозначением ПО будем подразумевать ГПП.

В ПО существует *расходуемый аддитивный ресурс Е* (энергия). Параметр Е характеризует *количественную меру* любого процесса по возможности его влияния (управления) на другой процесс при их взаимодействии.

Изначально (p,t0,e), в ПО задано некоторое количество ресурса – *Емах*. Емах рассматривается как текущая суммарная *энергия всех процессов*. При взаимодействии нескольких процессов происходит *перераспределение* их суммарной энергии. При каждом процессе *безвозвратно теряется* часть ресурса *ΔЕ*.

Процессы ПО делятся на два класса: *простые* – причинно-взаимосвязанные (например, физические) и сложные – *активно-субъектные самоподдерживающиеся процессы (АСП)*, например, биологические высокомолекулярные соединения. АСП характеризуются функционально-процессной целостностью.

Ресурс Е является одной из компонент *комплексной целевой функции (ЦФ) для АСП – «эволюционного потенциала»*. Ресурс Е целенаправленно используется АСП для запуска внутренних процессов по *самоподдержанию целостного иерархического процесса*, т.е. АСП компенсирует деструктивные воздействия из ПО.

АСП может *избирательно* взаимодействовать с окружающими его процессами ПО *(задачами)*. При каждом взаимодействии АСП *теряет ΔЕ*. В то же время, *управляя* посылаемыми в ПО процессами *(решениями)* из собственной *библиотеки (процессов)*, АСП может «*приобрести*» (часть) ресурса процесса-задачи, на который направлено его взаимодействие, или же «*потерять*», при *неверно* выбранном из библиотеки процессе-решении.

АСП может *пополнять* свою библиотеку значимыми (по Е) для него *тройками: <задача><решение><+E>*, где *+E ≠ 0*.

Тексты процессов данных троек назовем *образами*, а тройку образов – *информацией*.

Внутренние процессы анализа информации в библиотеке троек и применения к ним *набора из 7-ми базисных правил* (см. рис. 3) могут породить *знание*, приводящее к функциональному их покрытию и, как следствие, свертыванию числа избыточных троек в библиотеке без ухудшения прежнего усредненного *ресурсного баланса – E(t)* АСП.

Обработка внешних процессов ($П_{ПО}$) порождает в АСП и внутренние процессы ($П_{АСП}$). Мощность множества $П_{АСП}$ существенно больше $П_{ПО}$. Использование $П_{АСП}$ расширяет признаковое пространство задачи и у АСП появляется возможность формировать *абстрактные понятия*. Знание = $f_З(П_{ПО}, П_{АСП})$.

Знание – *наращиваемый неаддитивный ресурс* АСП. Его характеризующая величина – *сумма <+E> пересечения* по <задачам> троек из библиотеки АСП и <задачам> в зоне возможного взаимодействия с ними АСП.

Знание является *второй* компонентой «эволюционного потенциала» АСП.

Ресурсный баланс АСП – это текущее интегральное значение ресурса Е, при взаимодействии АСП по собственно выработанным правилам с окружающими задачами.

Любой процесс в библиотеке, такой как сравнение текста задачи с каждой тройкой библиотеки, имеет стоимость для АСП – -ΔЕ, поэтому АСП стремится перейти от библиотеки информации к *библиотеке знаний*. Если же ресурсный баланс АСП становится меньше *Epor*, то АСП *распадается* на простые процессы.

Первоначально АСП могут *образовываться случайно* среди всего многообразия процессов ПО. Например, в качестве самых простейших АСП можно рассматривать супрамолекулярные соединения, находящиеся между живыми и не живыми органическими соединениями.

Если в окрестности двух или более АСП окажется задача, то ее ресурс (Е) перейдет к АСП *с наиболее адекватной тройкой* к данной задаче. В случае же равенства адекватности троек – к АСП *с наибольшим ресурсным балансом*.

3.3. Выводы и следствия из подхода к понятию самообучающегося алгоритма

Отметим, что в работе формализовано продемонстрировано возможное построение саморазвивающегося универсального алгоритма основываясь только *на базисе из двух понятий*: процесс и целевая функция, рассматриваемых в координатах физических переменных: pt – пространство-время и e – энергия.

При определенных начальных условиях (параметрах) ПО в ней могут успешно развиваться АСП, поглощая весь наличный ресурс Емах в *ресурсный баланс одной АСП* (1-й вариант), либо, при других начальных условиях, вся ПО *деградирует в Емах = 0 и*

A=0 (2-й вариант). Все это напоминает физические основы «Антропного принципа» из космологии.

Относительно постановки задачи проблем информационной цивилизации, можно отметить, что не решив ее, мы скатываемся ко 2-му варианту развития процессов в ПО.

Доказательства и промежуточные выводы из вышеприведенной аксиоматической базы возможны с помощью компьютерного моделирования.

3.4. Базовые когнитивные правила АСП

Настала пора переходить от изобретения «эвристик аксиоматики абсолютной абстракции» к поиску и открытию «аксиоматики Природной». На рис. 3 кратко рассмотрены семь базовых когнитивных правил <задача> <решение> <+E> самообучаемой АСП на базе «мини-макси-минного» принципа:

1. Минимизация ресурса НСС

$$\lim_{t\to\infty} \frac{\text{Объем НСС (в битах)}}{\text{Объем КФ (в битах)}} \to 0$$

2. Максимизация эволюционного потенциала

```
<ABC> <c> <E–>
<ABC> <d> <E+>
------------------------
<ABC> <d> <E+>
```

3. Адаптация алфавита дискретизации

Детализация A

```
<AB>  <d> <E+>  A2={AB}
<AB>  <d> <E–>  A2={AB}
-------------------------------------
<ABC>           A3={ABC}
```

Загрубление A

```
<ABC> <d> <E+>  A3={ABC}
<BBC> <d> <E+>  A3={ABC}
-------------------------------------
<BC>            A2={BC}
```

4. Построение закономерностей в У-Р

```
<ABCD> <d> <E+>
<AB>   <d> <E+>
------------------------
<AB>   <d> <E+>
```

5. Построение закономерностей в Рк

```
<УЛКС дом > <домик >  <E+>
<УЛКС слон> <слоник > <E+>
-----------------------------------------
<УЛКС   X >  < XΔt ик> <E+>
```

6. Минимизация ресурса $НСС_{1\text{-}5}$

7. Повтор <d> → Stop, решение !

Рис. 3. Совокупность базовых когнитивных правил <задача><решение><+E> самообучаемой АСП на базе «мини-макси-минного» принципа:

Правило 1 – автоматическое формирование информационной модели ПО, где НСС – многодольный граф, отображающий ПО и КФ – текстовой поток из ПО;

Правило 2 – максимизация АСП ЦФ(Е), что принципиально отличается от таблицы состояний машины Тьюринга и других методов отображения классического понятия алгоритм;

Правило 3 – адаптация алфавита структуризации по приращению ЦФ;

Правило 4 – формирование простого правила через пересечение условий задачи;

Пример 5-го правила – формирование правил решения задач только на примерах их решений, используя понятие переменной. По 2-3 структурированным текстам задачи с

решениями (S-R-E, т.е. <задача><решение><+E>), например, сомообучение ИУС правилу формирования уменьшительно-ласкательного к данному слову (УЛКС):

<УЛКС *дом*> $\Rightarrow$ <*домик*> < *E+*>

<УЛКС *слон*> $\Rightarrow$ <*слоник*> <*E+*>

|-- ------- S --- --- | ----- R -----| -- E -- |

в ИУС(АСП) формируется интегрированный Р-оператор из имен функций и переменных:

$P(P_1,P_2,P_3)=P_1^{УЛКС}(X_1), P_2^{\Rightarrow}, P_3(X_1,C^{ИК})$ <E+>

Если теперь подать текст новой задачи (S) этого же класса, например:

<УЛКС *ёж*>

то получим завершение Р-оператора в виде ответа:

$\Rightarrow$ <*ёжик*>< *E+*> и т.д. …

Правило 6 – построение информационной модели самой АСП через рекурсивную архитектуру ИУС. Переход работы ИУС в категорию - знание;

Правило 7 – признак найденного решения любой задачи отражает повторение состояния процессов в ИУС.

3.5. Универсальный НСС-алгоритм рекурсивной ИУС (см. рис. 2)

1. Если на входе нет новых образов, то переход в режим («Сон») оптимизации НСС-структур.
   Если в режиме оптимизации найден образ с большим приращением E+, то перейти к 2.
   Перейти к 1.
2. Если найден образ общих признаков (образов) ПО в нескольких (2-х или более) похожих (ассоциирующих) примерах (N-элементах), то перейти к 4.
   Выполнить найденный образ.
3. Увеличить дискретизацию алфавита ПО.
   Если ресурс ЦФ < 0, то перейти к 5.
   Вычесть из целевой функции ΔЦФ_А. Перейти к 1.
4. Добавить к целевой функции соответствующее приращение E+.
   Если не было, то сформировать новое абстрактное понятие.
   Если ресурс ЦФ ≥ ЦФ-порог, то породить нового НСС-вычислителя или увеличить собственный вычислительный ресурс. Вычесть из ЦФ ΔЦФ_НСС.
   Перейти к 1.
5. EXIT (Эволюционный тупик)

3.6. Подходы к построению самообучающегося алгоритма

На рубеже XX века Давид Гильберт сформулировал задачу «автоматизации» процесса вывода математических теорем: «Можно ли построить алгоритм, строящий необходимый алгоритм решения любой точно поставленной задачи?». Программа Д. Гильберта была направлена на получение универсального алгоритма, сводившего деятельность математиков к формальной механической игре, в которой произвольную формализованную задачу (точно описанную на языке математики) можно было решить механически, используя процедуры универсального алгоритма. Парадокс программы Д. Гильберта обнаружился практически сразу и состоит он в том, что для доказательства существования такого универсального алгоритма требуется его строгое математическое описание. Этот парадокс был обнаружен Б. Расселом и сводился к хорошо известному еще древним грекам парадоксу брадобрея, «который бреет всех тех и только тех мужчин деревни, которые не бреются сами». Вопрос «Кто бреет брадобрея?» ставит любую «формальную» машину в тупик, так как такой брадобрей существует в том и только в том случае, когда его нет. Аналогичная ситуация складывается и с «множеством всевозможных подмножеств, которые не включают себя в качестве элемента», что является необходимым условием существования универсального алгоритма Гильберта.

Чтобы обойти возникшую парадоксальную ситуацию, было предложено большое количество уточнений понятия алгоритма, так А.Н. Колмогоров построил схему, лежащую в основе любого уточнения этого понятия. С помощью этой схемы можно строить новые уточнения как угодно долго.

3.7. Анализ понятия «алгоритм»

Понятие алгоритма - одно из базовых понятий программирования и математики. Примеры определения этого понятия. «Алгоритм – это всякая система вычислений, выполняемых по строго определённым правилам, которая после какого-либо числа шагов заведомо приводит к решению поставленной задачи». (А. Колмогоров)

«Алгоритм – это конечный набор правил, который определяет последовательность операций для решения конкретного множества задач и обладает пятью важными чертами: конечность, определённость, ввод, вывод, эффективность». (Д. Э. Кнут) и др.

В общем случае, под алгоритмом понимается четкое описание последовательности действий (операций) исполнителю, которые необходимо выполнить при решении задачи. Алгоритм записывается на формальном языке, исключающем неоднознач-ность толкования для исполнителя. Кратко можно сказать, что алгоритм описывает процесс преобразования исполнителем исходных текстов задачи в тексты решения. В качестве испол-нителя может быть человек, компьютер, автоматическое устройство и т.п. Исполнитель должен уметь выполнять все команды, составляющие алгоритм.

Формировать алгоритмы, в широком смысле, пока может только человек. Он и определяет «под исполнителя» необходимое понимание «точности» данных, «четкости» команд, «уровни» определенности и эффективности, а также «признаки решения поставленной задачи». И эти все параметры существенно зависят от возможностей (ТТХ характеристик) исполнителя.

Если исполнителем будет человек, то алгоритм может быть менее четким, по сравнению с ЭВМ, для которой должна быть написана точная программа на определенном алгоритмическом языке, процесс написания которой представляет определенные сложности. Ранее все социально-экономическое управление предусматривало в качестве исполнителя человека и алгоритмы писались на естественных или профессиональных языках. Но современные требования (скорости приращения информационных потоков и их объемы) существенно превышают ТТХ человека, как «алгоритмизатора», так и исполнителя.

В математике рассматриваются различные виды алгоритмов – программы для машин Тьюринга, алгоритмы Маркова (нормальные алгоритмы), частично рекурсивные функции и т.п. Знаменитый тезис Чёрча утверждает, что все виды алгоритмов эквивалентны друг другу, т.е. классы задач, решаемых разными типами алгоритмов, всегда совпадают. Тезис этот недоказуем (можно лишь доказать совпадение для двух конкретных типов алгоритмов, например, машин Тьюринга и нормальных алгоритмов), но никто в его верности не сомневается. Так что все языки программирования эквивалентны друг другу и различаются лишь тем, насколько они удобны для решения конкретных классов задач.

Разработка алгоритма решения задачи, обычно, это разбиение задачи на иерархически составляющие ее «решения-задачи» вплоть до терминальной и с обязательной оценкой корректности решения.

3.8. Отличие подхода ИУС (АСП) от «алгоритмической классики»

ИУС построена на многозначной «прагматической» логике <задача><решение><+E>. Например, правило 2 (см. рис. 3) демонстрирует недетерминизм, гибкость решения – максимизацию ЦФ(E).

Далее, правило 1 – это автоматическое формирование открытых областей применимости и результатов ИУС.

Используется минимум основных понятий: «процесс» и «целевая функция (ЦФ)».

Правило 3 – адаптация алфавита по приращению ЦФ, под текущую ПО;

Правило 4 – формирование простого (терминального) алгоритма через пересечение текстов условий задачи;

Правило 5 – формирование алгоритмов решения задач только на примерах их решений используя абстрактное понятие переменной. По 2-3 структурированным текстам задачи с решениями (S-R-E), формируется алгоритм оператор в виде графа, например:

$$P(P_1,P_2,P_3,P_4)=P_1^{ВСЕ}(X_1,C^{И},X_2,C^{Ы}),\ P_2^{ЕСТЬ}(X_3,X_1),\ P_3^{ЗНАЧИТ}(X_3,C^{ЕСТЬ}),\ P_4(X_2,C^{Ы},C^{Й})\ <E+>$$

который далее будет решать все задачи (S) этого же класса.

Правило 6 – построение информационной модели самой АСП через рекурсивную архитектуру ИУС. Переход работы ИУС в категорию - знание;

Правило 7 – признак найденного решения любой задачи «проблема останова – STOP» – повторение состояния процессов в ИУС.

Все правила ИУС можно объединить одним «мини-макси-минным» принципом. Минимизация отображения процессов ПО в процессах ИУС – приводит к автоматическому формированию структурированной информационной модели в виде многодольного иерархического графа.

Максимизация ЦФ – это монотонное повышение качества решений ИУС.

Процесс «мини/максимизации» в ИУС легко решается простейшими «генетическими» алгоритмами и вариационными методами. Таким образом, можно сказать, что весь базис алгоритмов и правил «саморазвивающегося алгоритма» построен на базе активно-субъектных самоподдерживающихся процессов.

Ассоциативность НСС эквивалентна понятию множества, где каждый элемент встречается только один раз.

В отличие от оснований математики, построенной на базе теории множеств, которой сопутствуют парадоксы при рассмотрении бесконечных мерностей и задаванием множества произвольными словосочетаниями, НСС близка к способу Кантора придумавшего «наивную теорию множеств», в которой запрещаются все действия и операции, ведущие к парадоксам. Идея в следующем: разрешается работать со множествами, кото-

рые «встречаются в природе», также разрешается работать со множествами, которые получаются из них разумными теоретико-множественными операциями.

Например, «Парадокс брадобрея» рассматривается как безрекурсивный процесс по параметру время и в каждый конкретный временной квант парадокса нет. Также как и в парадоксе: «Я – лжец». При характерной неполноте информации в НСС не работает «Закон исключённого третьего» («верно либо утверждение, либо его отрицание»)

Дополнительная переменная процесса – время (квант времени), при данном подходе снимает все парадоксы классической математики. Отталкиваясь от процессов естественно непротиворечивой ПО, можно проводить этап обучения и самообучения АСП. При этом, результат решения будет монотонно сходится к глобально верному, а самодостаточность графового представления образа-процесса, одновременно отображающая его динамические характеристики и целевую функцию, позволяет запустить механизмы автономного самообучения, открывая путь к практическому построению крупномасштабных ИУС.

Сегодня уже очевидно, что теория передачи информации (сигнала) К.Шеннона не есть теория семантической информации. Из рассмотренного же подхода вытекают формально-конструктивные определения понятий «информация» и «знание».

4. Заключение

В ближайшие десятилетия добьются благополучия и процветания те страны, которые сумеют создать «инновационную индустрию», на базе когнитивных технологий. Процесс восходящего развития техносферы идет через инновации, которые по сути являются «прибавочным продуктом» нашей цивилизации и, в соответствии с этим, чтобы максимизировать суммарную инновационность цивилизации, необходимо создать крупномасштабные самообучаемые ИУС.

Сегодняшнее развитие технологий построения искусственного разума вышло на уровень, когда уже необходимо создавать инфраструктуру для повышения эффективности и безопасности их разработки, т.к. они принадлежат к технологиям глобального действия. Такая инфраструктура может быть создана при участии государства и общественных организаций. В рамках этой инфраструктуры должны быть решены многие социально-экономические вопросы, вопросы формирования естественно-научной картины мира, вопросы нравственности и морали, а также ряд других важных общечеловеческих вопросов.

СПИСОК ЛИТЕРАТУРЫ

1. *В сети интернет* http://www.informograd.narod.ru/Liter.zip
2. *Бодякин В.И.* Концепция построения крупномасштабных информационно-управляющих систем на базе нейросемантических структур // настоящий сборник трудов РАСО’2012
3. *Воеводин В.В., Воеводин Вл.В*. Параллельные вычисления. – СПб:. БХВ-Петербург, 2002, – 608с.