

AGI For All (AGIFA)

Архитектура
Артюхов Виктор (@LiveBaster)

Цель проекта

Создать минимально необходимый и достаточный каркас для создания интеллектуальных систем, способных к развитию.

При этом, части проекта допускается реализовывать самыми разными методами.

Предполагается, что проект эволюционирует до системы, обладающей достаточным для полезного использования интеллектом.

Как в целом должен работать AGIFA?

Анохин П.К. А69 Избранные труды: Кибернетика функциональных систем/Под ред. К.В. Судакова

"На меня неизгладимое впечатление произвело одно случайное наблюдение в поле у пасущегося стада коров. Был жаркий летний день, и я заметил, как животные хлестали себя непрерывными ударами кончика хвоста по местам, на которые садились кровососущие насекомые – оводы. Но вот один овод, очевидно, сел и кусал на таком месте кожи, которое не могло быть, так сказать, "прострелено" ударом хвоста. Все туловище животного изогнулось дугой, голова сделала максимальный поворот в области шейных суставов, вытянулась шея, и язык, вытянувшись в неожиданно для меня длинную "палку", попытался достать то место туловища, где уселся овод. Все мышцы туловища находились в судорожном движении, однако все до единого сокращения были направлены в одну сторону – обеспечить прикосновение кончика языка к месту укуса. Одновременно подкожная мышца того места, на котором сидел овод, сильно сокращаясь, приближала овода к кончику языка. Здесь, таким образом, мы наблюдаем поразительное взаимодействие самых разнообразных мышц тела для обеспечения сбрасывания овода с кожи. Вот такая обширная организация, включающая мышцы, нервы, рецепторы, нервные центры и т.д., и может быть названа системой, поскольку она приводит к получению конечного полезного результата."

Из этого очень ёмкого описания видно, что AGIFA должен искать и успешно находить новые решения, а не пытаться повторять ранее запомненные. Память нужна для синтеза новых, улучшенных решений.

Требования к системе

1. Распределённость по схеме Peer-to-Peer
2. Масштабируемость
3. REST API для внешних пользователей
4. Простой веб-интерфейс для настроек и Базы Знаний (БЗ)
5. Возможность обмена структурированными знаниями
6. Возможность использовать разные хранилища данных, в том числе, крипто-хранилища (блокчейн)
7. Возможность расширения функциональности для работы с данными разных типов: текстами, изображениями, звуком, видео и т.п.
8. Использование C++ для максимизации производительности

Принципиальная схема AGIFA



Общая схема работы AGIFA

1. Все знания хранятся в структурированном виде Дерева Результатов ([ДР](#))
2. Образ требуемого результата приходит как управляющая часть входных данных
3. Блок синтеза, на основе знаний в ДР, создаёт набор действий для достижения образа требуемого результата
4. Блок исполнения исполняет действия
5. Окружающая среда изменяется и из неё приходят актуальные входные данные
6. Блок сравнения сравнивает поступившие данные с образом требуемого результата и в случае успеха, “зона внимания” переходит на родительский узел ДР, иначе Блок синтеза создаёт новые действия

Хранилище ДР

Возможные варианты реализации:

1. Файловое хранилище
2. MySQL
3. PostgreSQL
4. Blockchain

Дерево результатов

Описание методики:

<https://zen.yandex.ru/media/livebaster/metodika-dekompozicii-zadach-pri-pomosci-postroeniia-dereva-rezultatov-dr-602ea906ffa2d86ae452cbd2>

Основное полезное правило: “Нельзя получить Родительский Результат, без достижения всех дочерних результатов”.

Основные полезные свойства:

1. Узлы ДР просто сравнивать друг с другом, потому что все их дочерние элементы имеют строгие зависимости друг от друга.
2. ДР имеет строгую последовательность исполнения — от терминальных узлов вверх к родительским.
3. В ДР сразу видны зависимые (последовательные) друг от друга процессы и независимые (параллельные).

Примеры ДР

ДР для сперматозоида

Оплодотворить яйцеклетку

Проникнуть внутрь яйцеклетки

Приблизиться к яйцеклетке

ДР для дверного ключа

Открыть дверной замок

Проникнуть внутрь замка

Приблизиться к отверстию замка

ДР для воды

H₂O

H

H

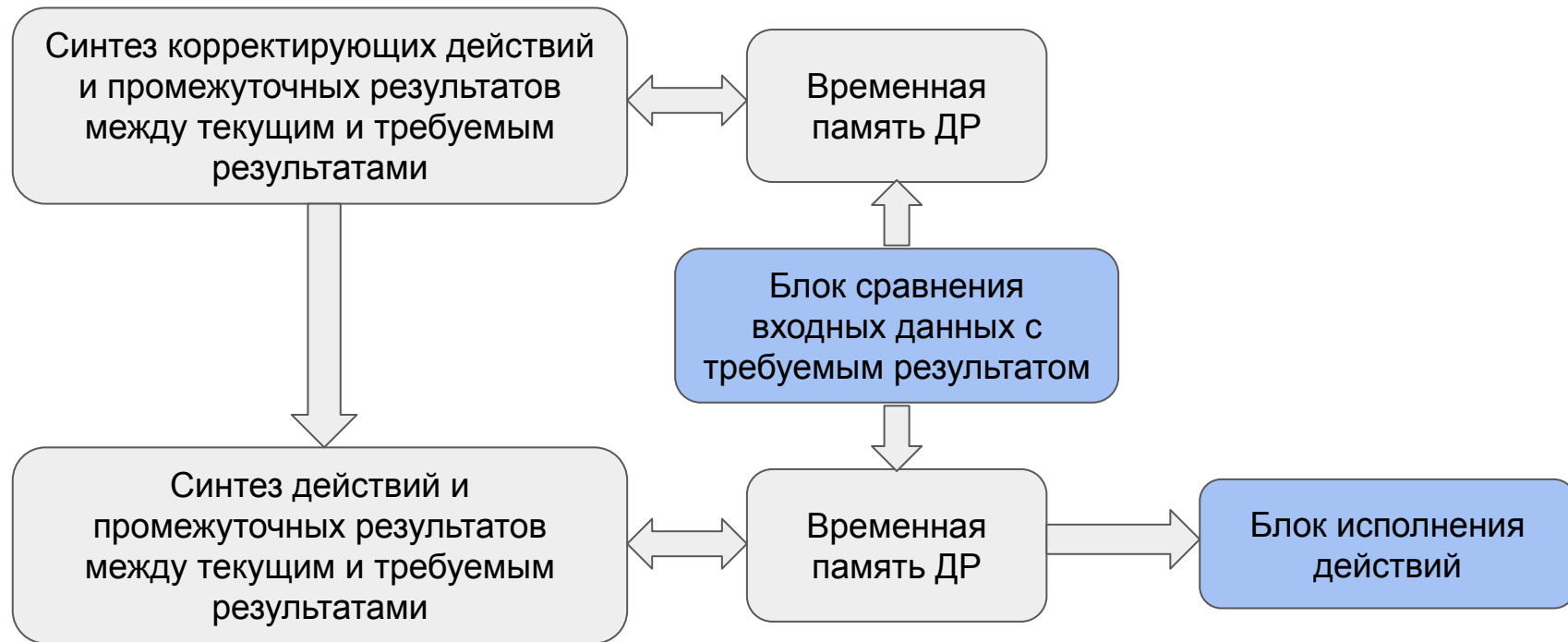
O

Нагреть

H₂

O₂

Блок синтеза образа требуемого результата и действий для его достижения



Общая схема работы блока синтеза

1. Синтез действий происходит во временной памяти ДР от узла ДР с требуемым результатом до текущего результата, в итоге формируется набор проверенных действий, который исполняется Блоком исполнения действий. Работу данного блока, человек идентифицирует, как “Подсознание” или “Автопилот”.
2. Если промежуточных действий недостаточно или исполненные проверенные действия не достигают требуемого результата, то блок синтеза корректирующих действий создаёт новые варианты действий, в рамках возможного набора действий. Работу данного блока, человек идентифицирует, как “Сознание”.

Блок сравнения входных данных с требуемым результатом

1. Ветви ДР, требующие сравнения, выполняются во временной памяти до получения образа результата
2. Образ результата сравнивается с образами от входных данных
3. Варианты реализации алгоритмов сравнения могут быть разные, от самых простых до сложных

Этот способ сравнения активный, т.е. система имеет возможность оперировать образами результатов до тех пор, пока они не станут похожими с требуемой точностью. Например, можно покрутить образ пирамидки во временной памяти, одновременно перемещая тело и точку зрения, пока он не совпадёт с образом от входных данных.

Это можно представить, как две молекулы соединяются в более крупную молекулу - обе стороны взаимодействуют в получении требуемого результата.

Молекулы тоже “думают”

Если рассмотреть процессы синтеза, начиная от атомов и выше, то хорошо видно, что это один и тот же процесс, который постепенно усложняется вплоть до афферентного синтеза и выше.

Здесь важно создать правильные условия для целенаправленного развития процесса.

Например, для синтеза воды нужно обеспечить наличие молекул кислорода и водорода, затем поджечь.

Клетки создают ещё более сложные условия для ещё более сложных процессов синтеза внутри клеток.

Нейроны создают условия для быстрой передачи сигнала от одной клетки к другой.

Нервная система организма в целом создаёт условия для афферентного синтеза.

Блок исполнения действий

Передаёт команды из временной памяти на внешние исполнительные механизмы.

Временная память ДР

Это память, в которой выполняются различные операции с частями основного ДР:

1. Сравнение веток
2. Выполнение веток
3. Операции перестановок частей веток - синтез новых веток
4. Удаление ненужных веток

Временная память - это “воображение” и место, где наблюдается процесс “мышления”. Если отключить входные и выходные данные, то у системы начнутся галлюцинации/сны, т.к. процессы синтеза веток ДР будут продолжаться, но без синхронизации с окружающим миром.

Коррекция ошибок при обучении

Свойство ДР: “Нельзя получить Родительский Результат, без достижения всех дочерних результатов”.

Если исполнить ветку ДР, то в итоге получим набор (лес) соседних, пересекающихся веток, корешки которых стремятся к нормальному распределению. Каждый раз пытаюсь выполнить ветку, которая наиболее близка к вершине кривой Гаусса, система будет стремиться к улучшению требуемого результата, за счёт работы блока корректирующих действий (“сознательная” коррекция).

Это можно представить как: “перелёт”, “недолёт” - “попал”.

Повторяемость результатов

Свойство ДР: “Нельзя получить Родительский Результат, без достижения всех дочерних результатов”.

При выполнении веток ДР, движение вверх по ДР в сторону корня, невозможно, пока не достигнут ближайший требуемый результат.

Это свойство обеспечивает *повторяемость* всех промежуточных результатов при движении к конечному требуемому результату.

При этом точность достижения дочерних результатов возможно “сознательно” регулировать при помощи блока синтеза корректирующих действий.

Отличие от генетических алгоритмов (ГА)

1. ГА - это алгоритм оптимизации.
2. ГА может подобрать оптимальные параметры для решения какой-либо отдельной задачи, но не может работать с иерархиями задач
3. ГА, как и прочие оптимизирующие алгоритмы, возможно использовать в AGIFA по своему прямому назначению - для оптимизации параметров действий и образов требуемых результатов

Типичные задачи для AGIFA

Задача падающего ключа.

Требуется открыть дверной замок ключом, но в процессе ключ падает на пол.

Здесь сложность в том, что невозможно заранее обучиться решению задачи, т.к. ключ каждый раз падает на пол в разное место.

В этой задаче бесполезны ГА, нейросети и прочие подобные методы.

Однако, человек эту задачу решает ежедневно и очень быстро.

AGIFA тоже может её успешно решать.

Типичные задачи для AGIFA

Задача управления движением глаз.

Требуется управлять глазами так, чтобы две картинки выглядели как одна, за исключением частей, разных из-за разных точек зрения.

Здесь сложность в том, что траектории движения глаз будут всё время разные, т.к. объект в поле зрения может находиться ближе, дальше, левее, правее и т.п. от глаз - невозможно запомнить все возможные варианты.

Единственное, что здесь постоянно - образ требуемого результата (величина допустимых отклонений картинок друг от друга), а значит AGIFA все необходимые дочерние варианты веток ДР может синтезировать за ограниченное время.

Типичные задачи для AGIFA

Задача обучения речи.

С точки зрения архитектуры AGIFA, “слышать” и “говорить” - это одинаковый процесс синтеза ДР.

“Слышать” - синтез веток ДР, максимально похожих на входные данные от “уха”.

“Говорить” - синтез веток ДР, опять максимально похожих на входные данные от “уха”, но с задержкой по времени.

Это можно представить, как режим “попугая”: услышал, сказал, сравнил, сказал чуть иначе, пока образы результатов не станут максимально похожи друг на друга.

Типичные задачи для AGIFA

Задача распознать цифры “на спине” - это известная детская игра.

Что делать, если обучение цифрам произошло по одному каналу входных данных (“глаза”), а требуется распознавать по другому каналу (“спина”)?

На самом деле, обучение произойдёт не на “сетчатке глаз” и в зрительной коре, а синтезируется ветка ДР, управляющая движением “глаз”.

Эта ветка ДР будет инвариантной к “изображению” на “спине” и на “сетчатке глаз”.

Типичные задачи для AGIFA

Задача обучения чему-нибудь.

Из архитектуры AGIFA, видно, что данная система потенциально может научиться решать практически любые задачи.

1. Образ требуемого результата ничем не ограничен и может быть сколько угодно сложным.
2. Дерево результатов ничем не ограничивается и может быть сколько угодно большим и сложным.
3. Ветка ДР - это фактически привычный всем алгоритм, имеющий ряд ограничений для более простого его использования, т.е. допускает реализацию на любой привычной компьютерной архитектуре.

Реализация блока сравнения входных данных с требуемым результатом

Базовый алгоритм сравнения может быть реализован просто как сравнение двух многомерных множеств.

Сложность реализации будет зависеть от того, сколько различных параметров требуемого результата надо извлечь из входных данных.

Например, для человека требуется извлекать много параметров (сложная зрительная кора), для лягушки - мало, для мухи ещё меньше.

Во всех случаях, AGIFA одинаков - меняются только способы реализации отдельных модулей.

Реализация предсказаний в ДР

Основной способ планирования будущего в ДР - это синтез новых веток ДР, на основе предыдущих успешно выполненных.

Это выглядит как продвижение “зоны внимания” от кончиков веток к корням деревьев.

“Зона внимания” - это часть ДР, которая находится во временной памяти.

Т.к. ветки ДР упорядочены, то части предыдущих веток могут быть использованы, как параметры, определяющие выбор направления, для синтезируемых последующих.

Более интересно это работает в многомодальной реализации - позволяет блокировать синтез заведомо нереалистичных предсказаний (“Нельзя получить Родительский Результат, без достижения всех дочерних результатов”).

Реализация AGIFA для систем реального времени

Из принципиальной схемы AGIFA видно, то наиболее ресурсоёмкий компонент - это блок сравнения входных данных с требуемым результатом.

В “живом мире” прослеживается прямая зависимость сложности “блока сравнения” от сложности организма - у простых организмов простая реализация, у сложных - сложная.

Таким образом, всегда можно выбрать такую реализацию отдельных компонентов AGIFA, которая обеспечивает решение требуемых задач в требуемое время.

Передача памяти по наследству

Для передачи наследственной информации используется ДНК, которая имеет ограниченную информационную ёмкость.

При этом видно, что простые существа сразу приспособлены к выживанию, а сложные наоборот совершенно беспомощны и без родителей не могут выжить.

Что может передаваться посредством ДНК?

Если есть механизм синтеза промежуточных действий, как в AGIFA, то достаточно в ДНК запомнить только ствол Дерева Результатов, состоящий из компактных образов стволовых требуемых результатов.

“Крону” дерева запоминать не обязательно - её всегда можно синтезировать по мере роста и развития организма.

Что из AGIFA используется в Самодостаточном ИИ?

Проект самодостаточного ИИ:

<http://www.livebaster.ru>

<https://zen.yandex.ru/livebaster>

1. Представление данных в виде ДР
2. Многомерные образы требуемых результатов
3. Распознавание через синтез
4. Решения в реальном времени

LiveBaster - это следующий шаг, после AGIFA, т.е. полностью самодостаточный, саморазвивающийся разум.