# Применение обратного метода Маслова и параллельных вычислений для решения задач искусственного интеллекта[1]

Петухова Н.Д., ст. преп. кафедры математики СПб ГМТУ,
ndpetukhova@gmail.com.

## Аннотация

Статья посвящена изложению разработанного алгоритма IAPTA решения задач искусственного интеллекта, допускающих формализацию в исчислении предикатов, с помощью модификации обратного метода Маслова и идей параллельных вычислений.

## Введение

Решение многих задач искусственного интеллекта, допускающих формализацию средствами языка исчисления предикатов, сводится к доказательству формул вида:

$$(\&S(\omega)) \Rightarrow \exists\overline{x}_{\neq}\, A(\overline{x}),$$

где $\omega = \{a_1, ..., a_k\}$ – набор констант, $S(\omega)$ – набор постоянных атомарных формул или их отрицаний, $A(\overline{x})$ – элементарная конъюнкция вида $P_{k_i}(\overline{x})$ [3]. Такое логическое следование равносильно истинности формулы:

$$(\&S(\omega)) \rightarrow \exists\overline{x}_{\neq}\, A(\overline{x})$$

при любых наборах значений $\omega$, которую, используя равносильные преобразования аппарата математической логики, можно свести к формуле вида:

$$\exists(x_1...x_n)\ _{\neq}(\&_{i=1}^{\delta} \vee \neg S(a_1, ..., a_k) \vee P_{k_i}(x_1, ..., x_n)) \qquad (1)$$

Эта формула выводима тогда и только тогда, когда существуют подстановки термов $t_1, ..., t_{n_i}$ вместо переменных $x_1, ..., x_{n_i}$ такие, что в каждом конъюнктивном члене найдется контрарная пара. Поиск таких термов требует экспоненциального числа шагов. Обратный метод ориентирован на существенное сокращение числа шагов поиска термов при переборе вариантов.

[1]Работа выполнена при поддержке гранта РФФИ 14-08-01276-а.

# Применение идей параллельных вычислений для решения задач логико-предметного распознавания образов

Создаем многопроцессорную систему, в которой действия между процессорами распределены поровну. Равное разделение действий необходимо для того, чтобы каждое действие имело возможность стать первым выполненным. Каждый процессор может выполнять следующие простые действия:

1. присвоение значения переменным;
2. замена всех вхождений переменной на её значение;
3. проверка формул на графическое совпадение;
4. отмена присвоения значения переменным;
5. отмена замены всех вхождений переменной;
6. изменение приоритета данного действия на 0.

Для начала работы алгоритма необходимо выбрать количество процессоров. Их, вообще говоря, может быть любое количество, но так как формула (1) содержит $\delta$ не повторяющихся дизъюнктов, то логично использовать $\delta$ процессоров.

Сформулируем алгоритм IAPTA (Inverse Ant Parallel Tactic Algorithm) решения задач логико-предметного распознавания образов, основанный на идеях параллельного вычисления и тактиках обратного метода Маслова. При этом при этом при каждом присвоении переменным значений процессоры связываются друг с другом и сравнивают результаты.

## Алгоритм IAPTA

**Определение 1.** *Список* $\Gamma$ *неповторяющихся формул вида* $\vee\neg S(\omega) \vee P_{k_i}(x_1.....x_{n_i})$ *называется* ***F-набором*** *для формул типа* (1) [2], [4].

**Определение 2.** *F-набор называется* ***пустым*** $\square$*, если все формулы, входящие в него, не имеют переменных и тавтологичны* [2], [4].

**Определение 3.** *F-набор называется* ***тупиковым****, если в него входит хотя бы одна формула, не имеющая переменных и являющаяся ложной или не являющаяся ни тавтологией, ни противоречием* [2].

1. Строим $\delta$-членный F-набор, формулы в котором не повторяются. То есть переписываем без конъюнкций все дизъюнкты вида $\vee\neg S(\omega) \vee P_{k_i}(x_1, ..., x_{n_i})$ при $i = 1, ..., \delta$. Создаем популяцию из $\delta$ процессоров.

Каждой паре потенциально контрарных формул $P_{k_i}(x_1, ..., x_{n_i})$ и $\neg P_{k_i}(a_{j_1}, ..., a_{j_{n_i}})$, входящих в один F-набор, назначаем приоритет их отождествления равным 1. Остальные приоритеты назначаем равными 0.

2. Копируем $\delta$-членный F-набор $\delta - 1$ раз. Получаем ровно $\delta$ одинаковых F-наборов.

Назначаем $i$-му процессору ($i = 1, ..., \delta$) свою начальную формулу $P_{k_i}(x_1, ..., x_{n_i})$ – формулу, с которой данный процессор начинает свой итерационный цикл, и потенциально контрарную ей постоянную формулу из $S(\omega)$, имеющую приоритет, равный 1.

Если какие-то два процессора начинают работу с формулой, начинающейся с одного и того же предикатного символа (таких формул не более $\delta$), то назначаем для них разные формулы из $S(\omega)$, потенциально контрарные данной. Если для каких-то двух процессоров не существует разных потенциально контрарных формул, то формула не выводима. Алгоритм заканчивает работу. Иначе, переходим к п. 3.3.

3. Параллельно работают $\delta$ процессоров. $i$-й процессор ($i = 1, ..., \delta$) осуществляет присвоение значений переменным.

3.1. Если в рабочей формуле данного процессора нет переменных, то в качестве рабочей для этого процессора выбираем формулу из следующей элементарной дизъюнкции, содержащую хоть одну переменную.

3.2. Ищем среди формул в $S(\omega)$ формулу $\neg P_{k_i}(a_{j_1}, ..., a_{j_{n_i}})$, имеющую приоритет, равный 1, и потенциально контрарную формуле $P_{k_i}(t_1, ..., t_{n_i})$, с которой работает этот процессор. Если нашли подходящую формулу, то переходим к п. 3.3. Если ее нет, то переходим к п. 4.

3.3. Решаем систему уравнений вида $t_l = a_{j_l}$ ($l = 1, ..., n_i$), унифицирующую список переменных и констант со списком констант. В случае, если эта система имеет решение, то переходим к п. 3.4. Если система решений не имеет, то понизить приоритет этого действия до 0 и переходим к п. 3.2.

3.4. Записываем результаты, полученные разными процессорами, и проверяем их на непротиворечивость следующим образом. Если процессоры одновременно присваивают одним и тем же переменным разные значения, то такие результаты считаются противоречивыми. Если результаты действий двух процессоров не противоречат друг другу, то присвоение полученных значений переменных осуществляется в формулах обоих процессоров.

3.5. Заменяем в F-наборе каждого процессора вхождения переменных из списка на их значения, полученные в п. 3.3 и 3.4, если успешно пройдена проверка на непротиворечивость.

3.6. Если для какого-либо процессора получился пустой F-набор, то алгоритм заканчивает работу. Формула выводима и найден набор значений переменных, существование которых утверждалось в формуле.

3.7. Если получился тупиковый F-набор, то переходим к п. 4.

3.8. Если для всех процессоров приоритеты всех действий равны 0, то формула не выводима. Алгоритм заканчивает работу.

3.9. Если в F-наборе какого-либо процессора существуют формулы, имеющие переменные, которым еще не присвоено значение, то переходим к п. 3.1.

4. Возвратная часть алгоритма.

4.1. Отменяем последнее действие п. 3.5, если это возможно, и переходим к п. 3.2.

4.2. Если для какого-либо процессора отмена последнего действия п. 3.5 невозможна, то алгоритм заканчивает работу.

4.3. Если все процессоры закончили работу, но пустой F-набор не получен, то алгоритм заканчивает работу. Формула не выводима.

**Теорема 1.** (Нижняя оценка числа шагов работы алгоритма.) Количество шагов решения любой из задач распознавания образов при использовании алгоритма IAPTA, основанного на тактиках обратного метода не менее $O(\delta s)$.

**Теорема 2.** (Верхняя оценка числа шагов работы алгоритма.) Количество шагов, затрачиваемых на решение любой из задач распознавания образов с помощью алгоритма IAPTA, а так же нахождения значений для переменных, существование которых утверждается в антецеденте задачи не превосходит $O(l(\max_k s_k)^\delta)$.

Где $\delta$ – количество дизъюнктивных членов в исходной формуле, $s+1$ – общее число атомарных формул в каждом дизъюнкте, $s_k$ – количество вхождений в исходную формулу атомарных формул с k-м предикатным символом, $l$ – наибольшее количество аргументов в атомарных формулах.

## Использование обратного метода для решения задачи выделения максимальной общей подформулы

Понятие неполной выводимости предикатной формулы было введено в [1] для распознавания объектов с неполной информацией. При этом рассматривается задача проверки того, что из истинности всех формул множества $S(\omega)$ следует истинность $A(\overline{y})$ или некоторой её максимальной подформулы $\tilde{A}(\overline{y})$ на наборе различных констант из $\omega$, где список переменных $\overline{y}$ является подсписком списка переменных $\overline{x}$.

Выражение $A(\overline{x}) \Rightarrow_p \exists \overline{y}_{\neq} \tilde{A}(\overline{y})$ означает не логическое следование, а то, что проводится выделение максимальной общей с точностью до имён переменных подформулы двух заданных элементарных конъюнкций $A(\overline{x})$ и $\tilde{A}(\overline{y})$, с нахождением такой подформулы $\tilde{A}'(\overline{y}')$ формулы $\tilde{A}(\overline{y})$, что имеет место

следствие $A(\overline{x}) \Rightarrow \exists \overline{y}'_{\neq} \tilde{A}'(\overline{y}')$, а также общего унификатора $\lambda$ формул $A(\overline{x})$ и $\tilde{A}'(\overline{y}')$.

Очевидно, что в этом следствии наборы $\overline{x}$ и $\overline{y}'$ являются наборами переменных. Тем не менее, алгоритм IAPTA можно применить и для решения задачи выделения максимальной общей подформулы, если внести следующие изменения:

- заменить действие «присвоение значений переменным» на «отождествление переменных»;
- «зациклить» алгоритм на полном переборе всех вариантов отождествлений переменных;
- ввести «хранилище» для длин фрагментов и самих фрагментов, соответствующих этим длинам.

## Литература


[1] Косовская Т. М. Частичная выводимость предикатных формул как средство распознавания объектов с неполной информацией // Вестник СПбГУ. Сер. 10. 2009. Вып. 1. С. 74-84.

[2] Косовская Т.М., Петухова Н.Д. Решение задач логико-предметного распознавания образов с использованием тактик обратного метода Маслова // Компьютерные инструменты в образовании – 2014. – Вып. 3. – С. 9-20.

[3] Косовская Т.М., Тимофеев А.В. Об одном новом подходе к формированию логических решающих правил – Вестник ЛГУ, 1985, №8.с.22-27.

[4] Оревков В.П., Обратный метод поиска вывода // В кн.: Адаменко А.Н., Кучуков А.М., Логическое программирование и Visual Prolog – Санкт-Петербург, БХВ, 2003, с. 952-965.