**Современные языки и системы логического программирования.**

Логическое программирование - один из видов "парадигм программирования", которое в большей степени основывается на математической логике.

Парадигмы программирования - это способ классификации языков программирования по их возможностям, при этом они могут одновременно относиться сразу к нескольким парадигмам.

Общие парадигмы программирования включают в себя:

1) " Императивные – мы говорим компьютеру, как решать задачу (что делать)"

1.1) Процедурные языки, которые группируют инструкции в процедуры

1.2) Объектно-ориентированные языки, которые группируют инструкции вместе с частью состояний

2) Декларативные, в которых просто объявляется свойства желаемого результата, а не то как его получить( " При декларативном программировании мы (на некотором формальном языке) описываем результат (его свойства), а не способ его достижения").

2.1) Функциональные языки, в них результат объявлен как значение функций

2.2) Математические языки, желаемый результат объявляется как решение задачи оптимизации

2.3) Логические, в которых желаемый результат объявляется как ответ на вопрос о системе фактов и правил.

**Опр.** Логическое программирование - это тип парадигмы программирования, который в значительной степени основан на формальной (математической) логике. Факты о проблемной области выражаются в виде логических формул, и программы выполняются путем применения к ним правил вывода до тех пор, пока не будет найден ответ на проблему или пока набор формул не окажется несовместимым.

Логическая парадигма программирования:

- Абдуктивное логическое программирование (ALP) - это представление высоко уровневых знаний, которое используется для решения задач, основанных на абдуктивном рассуждении.

**Опр.** Абдуктивное рассуждение - это логическое умозаключение, основанное на наборе наблюдений, которое предоставляет наиболее вероятное объяснение данных наблюдений. То есть данный процесс предоставляет правдоподобное объяснение, но без подтверждения его правильности.

Используется для решения проблем в диагностике, планировании, естественных языках и машинном обучении.

Большинство реализаций ALP просто расширяют вычислительную модель логического программирования на основе SLD (Selective Linear Definite). Также ALP может быть реализован с помощью связи его с ASP (Answer Set Programming).

Примерами первого способа являются ACLP, Asystem, CIFF, SCIFF, ABDUAL и ProLogICA.

Asystem - система объединившая идеи трех других систем: SLDNFA [Denecker and De Schreye], IFF [Fung] и ACLP [Kakas], в более эффективном виде. По сути, она формирует правила вывода SLDNFA в форме правил, которые используются в IFF процедурах. Исходные правила SLDNFA дополняются правилами, которые обрабатывают выражения ограничений, как предложено в ACLP. На уровне реализации, Asystem использует схему, аналогичную решателям конечных доменных ограничений: она будет оценивать всю детерминированную информацию, прежде чем сделать недетерминированный выбор. Когда все недетерминированные варианты успешно оценены и все хранилища ограничений согласованы, решение найдено.

Asystem написана как мета-аргумент (meta reasoner) в Sicstus Prolog.

- Answer set programming - это форма декларативного программирования, ориентированная на сложные (в первую очередь NP-сложные) задачи поиска.

В ASP задачи поиска сводятся к вычислению устойчивых моделей, а набор ответов, задействованные для генерации устойчивых моделей, используются для выполнения поиска. Вычислительный процесс, используемый при разработке многих решателей для наборов ответов, является усовершенствованием алгоритма DPLL и, в принципе, он всегда завершается (в отличие от оценки запроса Пролога, которая может привести к бесконечному циклу).

- Параллельное логическое программирование - это один из видов лог. программирования, в котором программы представляют собой наборы guarded Horn clauses, вида:

H :- G1, …, Gn | B1, …, Bn.

Где соединение G1, …, Gn называется guard утверждения, а символ "|" - оператор обязательства.

**Опр.** Horn clause - это утверждение (дизъюнкция литералов), содержащая не более одного положительного литерала.

Данное выражение декларативно читается как " H если G1 и ... и Gn или B1 и ... и Bn.".

Но если есть несколько вариантов, дающих правильное решение, тогда все эти выражения выполняются параллельно, проверяя удерживаются ли их guards G1, ..., Gn.

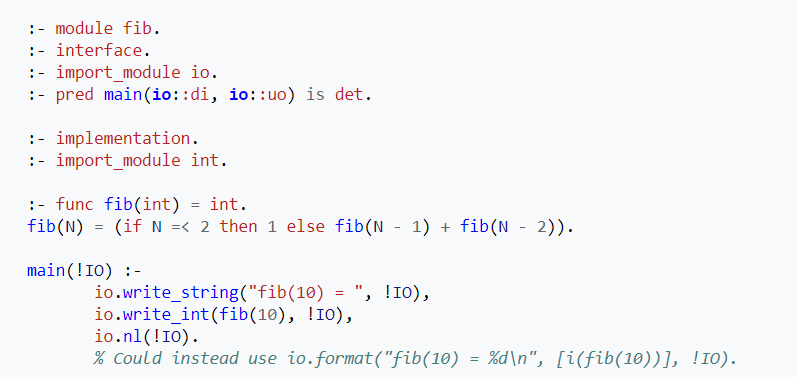
Если guards более чем одного выражения выполняется, то одно из них выбирается и выполнение продолжается с подцепями B1, ..., Bn для выбранного выражения. Эти подзадачи, так же могут выполняться параллельно.

" Thus concurrent logic programming implements a form of "don't care nondeterminism", rather than "don't know nondeterminism". "

- Функциональное логическое программирование - это комбинация в одном языке программирования парадигм финкционального и логического программирования. Например, такие языки как Curry и Mercury.

Mercury основывается на языке логического программирования Prolog, но является чисто декларативным языком, например, в нем нету выражения "!". Он может рассматриваться как Prolog с режимами и сильными типами. В отличие от оригинального Prolog'а, он имеет отдельную фазу компиляции, а не интерпретируется напрямую.

Пример (вычисление чисел Фибоначи)



- Индуктивное логическое программирование (ILP)

При кодировании известных базовых знаний и набора примеров, представленных в виде логической базы данных фактов, система ILP выведет гипотетическую логическую программу, которая влечет за собой все положительные и ни один из отрицательных примеров. Алгоритм ILP состоит из поиск гипотез и выбор гипотез. После выполнения поиска гипотез, с помощью процедуры с использованием индуктивной логики. Затем мнодество найденных гипотез выбираются алгоритмом выбора.

Одни из современных систем ILP -Hail, Imparo и Prolog.

Prolog хорошо подходит для конкретных задач, которые выгоднее решать с помощью логических запросов на основе правил, таких как системы голосового управления, шаблоны заполнения и поиск в базах данных.

Литература:

1) Nørmark, Kurt. Overview of the four main programming paradigms. Aalborg University, 9 May 2011. Retrieved 22 September 2012.

2) Frans Coenen (1999-10-11). "Characteristics of declarative programming languages". cgi.csc.liv.ac.uk. Retrieved 2014-02-20.

3) B. Van Nuffelen, "Abductive constraint logic programming: implementation and applications" , Ph.D. Thesis, Department of Computer Science, K.U.Leuven, Leuven, Belgium, June, 2004.

4) B. Van Nuffelen, and A. Kakas, A-system : Programming with abduction, Logic Programming and Nonmonotonic Reasoning, LPNMR 2001, Proceedings (T. Eiter, W. Faber and M. Truszczynski, eds.), LNAI, vol. 2173, Springer Verlag, 2001, pp. 393-396.

5) Marc Denecker and Antonis Kakas "Abduction in Logic Programming"

A.C. Kakas, F. Sadri (Eds.): Computat. Logic (Kowalski Festschrift), LNAI 2407, pp. 402–436, 2002.

c

Springer-Verlag Berlin Heidelberg 2002

A.C. Kakas, F. Sadri (Eds.): Computat. Logic (Kowalski Festschrift), LNAI 2407, pp. 402–436, 2002.

c

Springer-Verlag Berlin Heidelberg 2002

A.C. Kakas, F. Sadri (Eds.): Computat. Logic (Kowalski Festschrift), LNAI 2407, pp. 402–436, 2002.cSpringer-Verlag Berlin Heidelberg 2002.

6) Antoy, Sergio, and Michael Hanus. "Functional logic programming." Commun. ACM 53.4 (2010): 74-85.

7) Hanus, Michael, Herbert Kuchen, and Juan Jose Moreno-Navarro. "Curry: A truly functional logic language." Proc. ILPS. Vol. 95. No. 5. 1995.

8) Muggleton, S.H. (1991). "Inductive logic programming". New Generation Computing.