

ПРОГРАММИРОВАНИЕ В ОГРАНИЧЕНИЯХ И НЕДООПРЕДЕЛЕННЫЕ МОДЕЛИ

А.С.Нариньяни, В.В. Телерман, Д.М. Ушаков, И.Е. Швецов

(Российский НИИ Искусственного Интеллекта)

Введение

В современном программировании можно выделить несколько основных парадигм: императивное или алгоритмическое программирование (C, Pascal), логическое программирование (Prolog), функциональное программирование (LISP, РЕФАЛ) и др. Важное место в этом ряду занимает программирование в ограничениях (constraint programming), которое за последние несколько лет развивается особенно активно и становится все более популярным.

Программирование в ограничениях является по своей сути максимально декларативным и основано на описании модели задачи, а не алгоритма ее решения [1, 2]. Модель специфицируется в виде неупорядоченной совокупности отношений, которые соответствуют связям, существующим между параметрами задачи. Эти отношения, называемые общим термином "ограничения" могут иметь вид уравнений, неравенств, логических выражений и т. п.

Замечательно то, что одну и ту же модель можно использовать для решения различных задач (например, прямых и обратных). При этом постановка той или иной задачи конкретизируется путем добавления в модель ограничений на допустимые значения параметров и/или формулирования дополнительных связей между ними.

В модели нет априорного разделения параметров на входные и выходные. В соответствии с требованиями решаемой задачи пользователь определяет, какие из параметров заданы точно, какие не известны совсем, а какие — приблизительно (исходная информация о таких параметрах задается в виде ограничений на множество их возможных значений). Используя модель задачи и исходную информацию о значениях ее параметров, методы программирования в ограничениях обеспечивают автоматическое нахождение решения.

В самом общем виде постановка задачи в парадигме программирования в ограничениях формулируется следующим образом. Пусть на переменные x_1, x_2, \dots, x_n , областями значений которых являются множества X_1, X_2, \dots, X_n , заданы ограничения $C_i(x_1, x_2, \dots, x_n), i = 1, k$. Требуется найти наборы значений $\langle a_1, a_2, \dots, a_n \rangle (a_i \in X_i)$, которые бы удовлетворяли всем ограничениям одновременно.

Такая постановка задачи называется проблемой удовлетворения ограничений, а для ее решения используются различные алгоритмы и методы. В частности проблема удовлетворения ограничений может формулироваться как система уравнений с числовыми параметрами, а для ее решения могут использоваться стандартные численные методы. Однако при решении многих реальных задач эти методы оказываются неприменимыми, особенно если модель включает и нечисловые параметры, а начальные данные могут задаваться приблизительно в виде множеств и интервалов, содержащих допустимые значения.

Реальное программирование в ограничениях особенно полезно там, где кончаются возможности обычной математики. Оно используется при решении задач планирования, проектирования, прогнозирования, в инженерных и экономических расчетах, при создании графических интерфейсов, в системах понимания естественного языка и др. Среди наиболее известных зарубежных систем, реализующих парадигму программирования в ограничениях, можно отметить Prolog III [3], CLP(R) [4], CLP(BNR) [5], clp(FD) [6], CHIP [7], ILOG Solver [8], Newton [9] и др.

Одним из наиболее развитых и практически значимых подходов, относящихся к программированию в ограничениях, являются *недоопределенные модели*.

Метод недоопределенных моделей (Н-моделей) был предложен в начале 80-х годов [10–12] для представления и обработки неполностью определенных знаний. Рассматриваемый вначале как оригинальный метод из области искусственного интеллекта, он трансформировался постепенно в прикладную технологию программирования в ограничениях. Технология Н-моделей выделяется среди других подходов вычислительной мощностью, универсальностью и эффективностью. Фактически она является единственной технологией, которая позволяет решать задачу удовлетворения ограничений в самой общей постановке.

К настоящему времени на базе Н-моделей создана многоуровневая технология программирования, позволяющая решать качественно новые классы задач в таких областях, как экономика, инженерные расчеты, календарное планирование, вычислительная математика, САПР, ГИС и др. Среди систем, реализованных на основе аппарата Н-моделей, можно отметить такие, как UniCalc [13], HeMo-TeK [14], LogiCalc [15] и Time-Ex [16].

Наш Институт планирует в журнале “Информационные Технологии” серию публикаций о наиболее интересных разработках, базирующихся на технологии Н-моделей и задача данной статьи дать читателю общую информацию о специфике и возможностях как самого этого аппарата, так и в целом о программировании в ограничениях.

Статья организована следующим образом. В первой главе рассматриваются основные понятия, лежащие в основе аппарата Н-моделей и пример, поясняющий идеи алгоритма вычислений. Во второй главе вводится понятие недоопределенного расширения и рассматриваются примеры таких расширений. В третьей главе описываются недоопределенные модели и типы данных, используемые в Н-моделях, а также приводится алгоритм вычислений. В четвертой - показаны некоторые возможные применения аппарата Н-моделей.

1. Основные понятия

1.1 Недоопределенная переменная

Напомним, что существуют по крайней мере два основных, качественно различных, понятия переменной.

Классическая переменная – базовое понятие математики, представляет некоторую неизвестную величину, связанную условиями задачи с другими известными и неизвестными величинами. При достаточной полноте условий задачи сопоставленная данной переменной величина, т.е. ее значение, может быть получена точно. Таким образом, значение классической переменной отражает некоторую конкретную, заданную условиями задачи сущность или *денотат*, представляемую в задаче именем данной переменной. В рамках одной задачи денотат-значение переменной не может меняться - он может быть либо известен либо неизвестен.

Алгоритмическая переменная, связанная с использованием алгоритмов и появлением языков программирования, является, по сути дела, ячейкой абстрактной памяти, в которую могут помещаться различные значения, меняющиеся по ходу исполнения соответствующей процедуры.

И в математике и традиционном программировании с каждой переменной можно связать только одно значение из области ее определения (или *универсума*).

Далее в статье понятие переменной будет использоваться в расширенном классическом смысле: в Н-моделях переменной сопоставляется *недоопределенное значение* (или *Н-значение*), являющееся оценкой реального значения-денотата на основе доступной нам в данный момент информации. Н-значение является промежуточным между полной определенностью (точное значение) и полной неопределенностью (весь универсум) и может уточняться по мере получения более точных данных. Например, не зная точного возраста Петрова, мы можем оценить его как “между 35 и 40 годми”.

Таким образом, Н-значение есть непустое подмножество универсума, содержащее внутри себя значение-денотат, которое остается пока неизвестным (вернее, известным с точностью до данного недоопределенного значения) ввиду недостатка информации. В процессе уточнения, т.е. при поступлении более точных данных, Н-значение становится все более определенным и в пределе может стать точным, т.е. равным денотату данной недоопределенной переменной (*Н-переменной*).

Это означает, что для Н-переменной, вне зависимости от ее типа, следует различать два значения:

- * *реальное* неизвестное нам значение-денотат, которое она представляет, и
- * ее *текущее* Н-значение, являющееся доступной оценкой этого реального значения.

Недоопределенность может характеризовать не только значения параметров существующих объектов или процессов, но и виртуальных объектов, находящихся в процессе создания. В этом случае Н-значение выступает в качестве ограничения на вычисляемое значение. Примеры:

- здесь нужен провод диаметром от 0.25 до 0.32 мм;
- в этом редукторе придется использовать коническую или цилиндрическую передачу.

В процессе вычислений Н-значение может становиться только более точным, гарантируя тем самым монотонность вывода. Для завершаемости вычислений существенно, чтобы число различных Н-значений одного объекта было конечным.

1.2 Иллюстративный пример

Рассмотрим теперь пример, поясняющий алгоритм вывода, используемый в недоопределенных моделях.

Пусть требуется найти решение системы из двух линейных уравнений с двумя неизвестными:

$$y = x - 1; \quad (F_1)$$

$$2 * y = 3 * (2 - x); \quad (F_2)$$

Каждое уравнение можно рассматривать как неявную функцию (F_1 и F_2) от двух переменных (x и y). Графики этих функций изображены на Рис. 1.

Предположим, что известна начальная оценка значения переменной x : где-то между -1 и 4 (такую оценку можно рассматривать как интервал $[-1, 4]$).

Идея недоопределенных вычислений состоит в том, что по текущей оценке поочередно вычисляются проекции функций F_1 и F_2 на x и y . Например, проекция F_1 на y для x равного $[-1, 4]$ равняется интервалу $[-2, 3]$. Результат такой проекции изображен на Рис. 2.

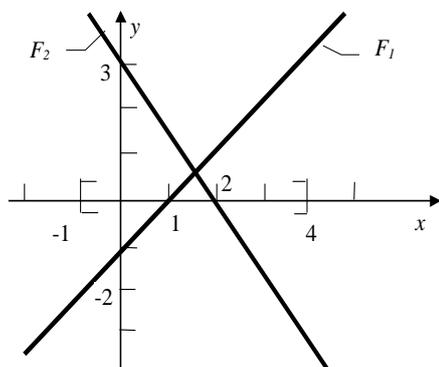


Рис. 1

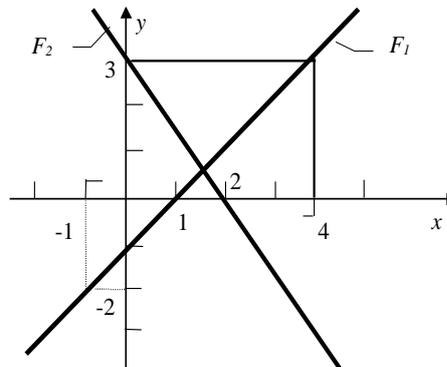


Рис. 2

Теперь, если по y вычислять проекцию F_2 на x , то получим новое значение x равное интервалу $[0, 10/3]$ (см. Рис. 3).

Продолжая этот процесс мы постепенно приближаемся к искомому решению. На Рис. 4 показаны две спирали, которые показывают, каким образом мы приближаемся к решению снизу и сверху, соответственно.

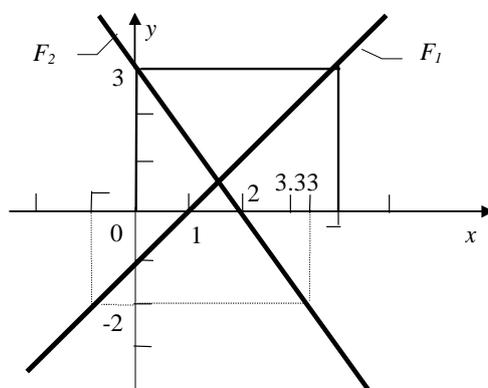


Рис. 3

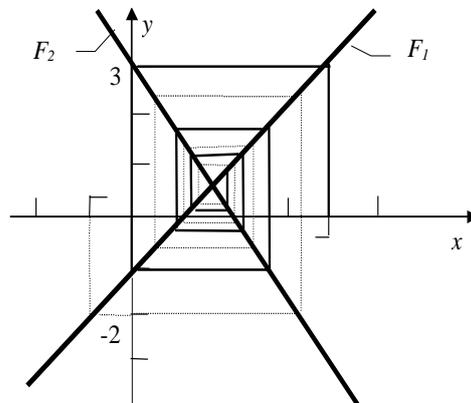


Рис. 4

Стоит отметить, что параметры реальных задач всегда имеют начальные оценки их значений, поскольку даже в тех случаях, когда решающий задачу затрудняется в определении исходных ограничений на область значений того или иного числового параметра, оценка его значения от минус до плюс бесконечности будет представлена в машине конкретными числами.

Таким образом, в нашем примере в общем случае одновременно будут закручиваться четыре спирали — две от x и две от y .

2. Недоопределенные расширения

При работе с недоопределенными значениями необходимо выбрать для каждого значения некоторый способ его представления. При этом необходимо представлять полностью неопределенное Н-значение (т.е. его универсум) и противоречивую оценку (т.е. пустое множество). Это приводит нас к следующему определению.

2.1 Исходные определения

Для того, чтобы для данной традиционной формальной системы построить ее аналог, способный оперировать с Н-значениями, необходимо сформировать область значений для Н-переменных, представляющих переменные исходной системы.

Недоопределенным расширением (Н-расширением) произвольного универсального множества X является любая конечная система его подмножеств, замкнутая относительно операции пересечения и содержащая весь универсум и пустое множество. В случае бесконечного множества X в качестве универсума рассматривается некоторое его конечное подмножество $X_0 \subset X$. Например, если X – множество вещественных чисел, то X_0 может быть множеством чисел, представимых в памяти компьютера таких, что любые два числа отличаются не менее, чем на некоторый $\varepsilon > 0$.

Ниже мы будем использовать обозначения *Н-функция* и *Н-отношение* вместо *недоопределенная функция* и *недоопределенное отношение*.

Следует заметить, что для одного и того же универсума существуют различные возможные Н-расширения. Далее будем обозначать через $*X$ произвольное Н-расширение универсума X .

Независимо от вида выбранного Н-расширения, приведенное выше определение гарантирует *однозначное представление* любого множества X_0 из универсума X в его Н-расширении $*X$. Такое представление, обозначаемое $*[X_0]$, рассматривается как наименьший (в смысле включения) элемент Н-расширения, содержащий данное подмножество.

Рассмотрим некоторые виды Н-расширений, которые используются в программных технологиях, базирующихся на аппарате Н-моделей. Более подробное описание приведенных ниже Н-расширений можно найти в статье [17] и в докладе [18].

1) Наиболее простым является Н-расширение, в котором каждый его элемент представлен *точным* значением ($*X = X^{Single}$):

$$X^{Single} = \{ \{x\} \mid x \in X \} \cup \{\emptyset\} \cup \{X\}.$$

Данное Н-расширение добавляет в обычный универсум два специальных значения: *не определено* (X) и *противоречие* (\emptyset).

2) *Перечислимое* Н-расширение представляется множеством всех подмножеств (которое обозначим 2^X):

$$X^{Enum} = 2^X.$$

Данное Н-расширение можно применять лишь к конечным универсумам.

В случае, когда X является *решеткой* (множеством с определенными на нем ассоциативными и коммутативными операциями, подчиняющимися законам поглощения и идемпотентности), можно задать такие виды Н-расширений X , как *интервалы* и *мультиинтервалы* [17 – 22].

3) *Интервальное* Н-расширение:

$$X^{Interval} = \{ [x^{Lo}, x^{Up}] \mid x^{Lo}, x^{Up} \in X \}.$$

Здесь x^{Lo} обозначает нижнюю, а x^{Up} — верхнюю границу интервала. Пустое множество (\emptyset) представляется любым интервалом $[x^{Lo}, x^{Up}]$, где $x^{Lo} > x^{Up}$.

4) *Мультиинтервальное* Н-расширение:

$$X^{MultiInterval} = \{ x \mid x = \bigcup_k x_k, x_k \in X^{Interval}, k = 1, 2, \dots \}.$$

Пример. Пусть универсум переменной v - это множество целочисленных значений, а ее текущее значение равно множеству $\{3, -2, 7, 8, 9, 4\}$. Рассмотрим его недоопределенное представление в различных Н-расширениях множества целых чисел:

Н-расширение	Н-значение V
--------------	--------------

<i>Single</i>	(полная неопределенность)
<i>Enum</i>	{ 3, -2, 7, 8, 9, 4 }
<i>Interval</i>	[-2, 9]
<i>MultiInterval</i>	{ [-2, -2], [3, 4], [7, 9] }

5) Рассмотрим теперь универсум, заданный в виде декартова произведения множеств: $X = X_1 \times \dots \times X_n$. Как и к любому множеству, к нему применимы Н-расширения X^{Single} и X^{Enum} . Более того, если каждый из X_i является решеткой, то X тоже является решеткой. В этом случае к X также применимы Н-расширения $X^{Interval}$ и $X^{MultiInterval}$. Но к универсуму X применимо еще одно Н-расширение — *структурное*:

$$*X = *X_1 \times \dots \times *X_n.$$

Соотношения между Н-расширениями $*X_1 \times \dots \times *X_n$ и $*(X_1 \times \dots \times X_n)$ обсуждаются в работе [17].

2.2 Недоопределенные операции и отношения

2.2.1. Рассмотрим, каким образом над приведенными выше Н-расширениями можно определить операции и отношения.

Пусть в предметной области задана операция над обычными универсумами

$$f: X_1 \times \dots \times X_n \rightarrow X_{n+1},$$

и некоторые Н-расширения $*X_i$ ($i = 1, \dots, n+1$). Тогда *Н-расширение операции f* есть операция

$$*f: *X_1 \times \dots \times *X_n \rightarrow *X_{n+1},$$

результатом которой является представление образа функции f на множестве значений $X_1 \times \dots \times X_n$ в недоопределенном расширении $*X_{n+1}$:

$$*f(x_1, \dots, x_n) = *[\{ f(a_1, \dots, a_n) \mid a_i \in x_i, i = 1, \dots, n \}].$$

Под образом функции f на множестве значений X понимается множество значений $\{ f(x) \}$, для всех $x \in X$.

Например, в случае перечислений (*Enum*) вычисление Н-расширенной операции может заключаться в переборе возможных значений ее аргументов, а в случае интервалов (*Interval*) и мультиинтервалов (*Multiinterval*) можно воспользоваться широко известными правилами интервальной арифметики [22].

Пример. Пусть имеется операция извлечения корня из вещественного числа, которая вычисляет не только арифметическое значение корня, но и его отрицательный аналог. Областью определения этой функции является множество неотрицательных чисел. Посмотрим на некоторые виды ее Н-расширений на примере извлечения корня из недоопределенного вещественного числа, заданного в виде интервала [4, 25].

Н-расширение	Значение корня
<i>Single</i>	(полная неопределенность)
<i>Enum</i>	{ -5, -5+ε, -5+2ε, ..., -2-ε, -2, 2, 2+ε, 2+2ε, ..., 5-ε, 5 }
<i>Interval</i>	[-5, 5]
<i>MultiInterval</i>	{ [-5, -2], [2, 5] }

2.2.2. Пусть теперь в предметной области задано отношение $r(x_1, \dots, x_n)$, определяющее некоторое подмножество декартова произведения $X_1 \times \dots \times X_n$. Проблема организации автоматического решения для моделей традиционно связывается с функциональной интерпретацией всякой отношения $r(X')$, которое является набором функций, вычисляющих значение каждой переменной из X' на основании заданных значений других переменных. Таким образом, функция f_i ($i = 1, \dots, n$) называется *функцией интерпретации* отношения r , если

$$x_i = f_i(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n),$$

на множестве $r(x_1, \dots, x_n)$.

Например, уравнение

$$x + y = z;$$

интерпретируется тремя функциями:

$$\begin{aligned} z &= x + y; \\ x &= z - y; \\ y &= z - x; \end{aligned}$$

Такие функционально интерпретируемые отношения в дальнейшем будем называть *ограничениями*.

Задача удовлетворения ограничений на Н-моделях описывается как множество объектов предметной области, каждому из которых сопоставлено недоопределенное значение, и множество ограничений на этих объектах. При этом каждому объекту приписывается универсум и вид Н-расширения.

Все функции интерпретации ограничений, если их рассматривать как операторы, действующие на множестве Н-значений объектов модели, являются монотонными по включению и нерасширяющими. Двух этих свойств и конечности множества всех Н-значений достаточно для существования *наибольшей* (по включению) *неподвижной точки* произвольной системы таких операторов и гарантированного достижения этой точки за конечное число шагов методом последовательных приближений [17]. Его разновидностью является используемый в Н-моделях потоковый алгоритм вычислений, который подробнее будет рассмотрен в следующей главе.

2.3 Свойства недоопределенных расширений

Заметим, что некоторые свойства обычных операций существенно меняются в соответствующих недоопределенных расширениях. Рассмотрим например, хорошо известное свойство аддитивных операций над числами (целых или вещественных), что сумма любого числа и обратного ему равна единственному нулевому элементу

$$a + (-a) = 0. \quad (1)$$

Пусть в качестве Н-расширения множества чисел рассматривается Н-расширение *Interval*, а соответствующие Н-расширения операций минус (*-) и плюс (*+) определены согласно известным правилам интервальной арифметики [21]. Таким образом, Н-расширение унарного минуса имеет вид

$$*- : [a^{Lo}, a^{Up}] = [-a^{Up}, -a^{Lo}];$$

а Н-расширение сложения ($a = b *+ c$) вычисляется следующим образом:

$$*+ : [a^{Lo}, a^{Up}] = [b^{Lo} + c^{Lo}, b^{Up} + c^{Up}];$$

В таком случае, Н-расширение выражения (1) имеет следующий вид:

$$\begin{aligned} [a^{Lo}, a^{Up}] *+ (*-[a^{Lo}, a^{Up}]) &= [a^{Lo}, a^{Up}] *+ [-a^{Up}, -a^{Lo}] = \\ &= [a^{Lo} + (-a^{Up}), a^{Up} + (-a^{Lo})] = [a^{Lo} - a^{Up}, a^{Up} - a^{Lo}]. \end{aligned}$$

Достаточно очевидно, что когда нижняя и верхняя границы Н-числа не совпадают (т.е. число недоопределено), интервал $[a^{Lo} - a^{Up}, a^{Up} - a^{Lo}]$ всего лишь *содержит* нулевой элемент, но не равен в точности ему.

Такое изменение свойства унарного минуса приводит к тому, что закон дистрибутивности для обычных аддитивных и мультипликативных операций переходит в так называемый закон *субдистрибутивности*:

$$*a * \times (*b *+ *c) \subseteq *a * \times *b *+ *a * \times *c.$$

Из сказанного выше можно сделать вывод, что в Н-моделях существенно то, в каком виде представлены условия задачи: какие Н-расширения выбраны для переменных и каким образом представлены ограничения (уравнения и неравенства).

2. Недоопределенные модели

2.1 Обобщенные вычислительные модели

Недоопределенные модели являются частным случаем *обобщенных вычислительных моделей (ОВМ)* [23, 24], которые имеют более широкую область применения, чем решение задач удовлетворения ограничений. Ниже мы даем определение ОВМ и алгоритма вычислений на них, указывая при необходимости отличия Н-моделей от ОВМ.

Обобщенная вычислительная модель $M = (V, W, C, R)$ состоит из следующих четырех множеств:

- V – множество объектов из заданной предметной области,
- R – множество ограничений на значениях объектов из V ,
- W – множество *функций присваивания*, и
- C – множество *функций проверки корректности*.

Каждому объекту $v \in V$ сопоставлены:

- универсум X_v ;
- начальное значение из универсума (точное, недоопределенное, или полная неопределенность);
- функция присваивания W_v ;
- функция проверки корректности C_v .

Функция присваивания — это двухместная функция, работающая при каждой попытке присваивания очередного значения объекту $v \in V$ и определяющая новое значение объекта как функцию от текущего и присваиваемого значений.

Функция проверки корректности — это унарный предикат, который выполняется в случае, если значение объекта x изменилось, и проверяет правильность этого нового значения.

Ограничения из R должны быть функционально интерпретируемыми.

На уровне интерпретации ОВМ представляется двудольным ориентированным графом (ОВМ-сеть), в котором выделены два типа вершин: объекты и функции. Дуги связывают функциональные и объектные вершины. Входящие в вершину-функцию дуги соотносят с ней объекты, значения которых выступают в качестве входных аргументов для функции, исходящие — указывают на объекты, в которые должна производиться запись вырабатываемых функцией результатов.

Каждой объектной вершине сопоставляются тип и значение, а также связываются функции присваивания и проверки корректности.

С каждой функциональной вершиной соотнесены целое число, играющее роль приоритета, и разметка входящих и исходящих дуг.

2.2 Процесс удовлетворения ограничений на ОВМ

Процесс вычислений на ОВМ имеет потоковый характер, - изменение объектных вершин сети активирует (вызывает к исполнению) функциональные вершины, для которых эти объектные вершины являются входными аргументами, а исполнение функциональных вершин в свою очередь может вызывать изменение результирующих объектных вершин. Вычисления заканчиваются тогда, когда либо не останется активных функциональных вершин (*УСПЕХ*), либо функция проверки корректности вырабатывает значение ложь (*НЕУДАЧА*).

Допустим, что вместо обычных универсумов X_v рассматриваются некоторые их недоопределенные расширения $*X_v$. Пусть все функции присваивания в ОВМ производят пересечение N -значений:

$$w_i(\xi^{old}, \xi^{new}) = \xi^{old} \cap \xi^{new};$$

а функции проверки корректности – проверку N -значения на непустоту:

$$corr_i(\xi) = \text{if } \xi \neq \emptyset \text{ then true else false fi.}$$

Именно этот класс обобщенных вычислительных моделей называется недоопределенными моделями или N -моделями.

В работе [17] доказаны следующие утверждения:

(i) *Процесс удовлетворения ограничений в N -моделях завершается за конечное число шагов.*

(ii) *Достижение процессом НЕУДАЧИ или УСПЕХА предопределено входными данными (начальными N -значениями переменных и ограничениями) и не зависит от конкретной стратегии выбора очередного ограничения для интерпретации.*

(iii) *В случае УСПЕХА процесса, при одних и тех же входных N -значениях переменных их выходные N -значения не зависят от конкретной стратегии выбора очередного ограничения для интерпретации.*

(iv) *В случае УСПЕХА процесса, решение задачи (если оно существует) лежит внутри полученного результата (декартова произведения N -значений).*

2.3 Пример работы процесса

Поясним работу процесса удовлетворения ограничений для N -моделей на примере достаточно простой системы из двух линейных уравнений с двумя целочисленными неизвестными:

$$\begin{cases} x + y = 12; \\ \\ 2 * x = y; \end{cases} \quad (2)$$

Предположим, что значения x и y ограничены следующими неравенствами:

$$0 \leq x \leq 100; 0 \leq y \leq 100. \quad (3)$$

Множество объектов X данной модели содержит две целочисленные переменные (x, y) , множество ограничений R — два уравнения и четыре неравенства.

Для определения соответствующей H -модели требуется выбрать некоторый вид H -расширения целого числа. Пусть недоопределенное целое число (H -число) будет представлено H -расширением *Interval* (см. раздел 1). Функция присваивания (обозначим ее $PRint$) реализует следующее правило изменения значений H -числа: нижняя граница может только увеличиваться, а верхняя — только уменьшаться. Функция проверки корректности (обозначим ее $PRDint$) проверяет, чтобы нижняя граница H -числа не превосходила верхнюю. Операции для таких H -чисел определяются в соответствии с правилами интервальной математики [21].

Так как множество X содержит две переменные типа H -число, множество функций присваивания (W) и множество функций проверки корректности (C) содержат по два элемента:

$$W = \{ PRint(x), PRint(y) \}; \quad C = \{ PRDint(x), PRDint(y) \};$$

Множество функций интерпретации уравнений (2) состоит из четырех элементов (f_1-f_4):

$$\begin{aligned} f_1: & \quad y \leftarrow 12 - x; & f_2: & \quad x \leftarrow 12 - y; \\ f_3: & \quad y \leftarrow 2 * x; & f_4: & \quad x \leftarrow y / 2; \end{aligned}$$

Согласно правилам интервальной арифметики, семантика функций интерпретации (f_1-f_4) представляется следующим образом:

$$\begin{aligned} f_1: & [y^{Lo}, y^{Up}] \leftarrow [12 - x^{Up}, 12 - x^{Lo}]; \\ f_2: & [x^{Lo}, x^{Up}] \leftarrow [12 - y^{Up}, 12 - y^{Lo}]; \\ f_3: & [y^{Lo}, y^{Up}] \leftarrow [\min \{ 2 * x^{Lo}, 2 * x^{Up} \}, \max \{ 2 * x^{Lo}, 2 * x^{Up} \}]; \\ f_4: & [x^{Lo}, x^{Up}] \leftarrow [\min \{ x^{Lo} / 2, x^{Up} / 2 \}, \max \{ x^{Lo} / 2, x^{Up} / 2 \}]; \end{aligned}$$

Четыре ограничения (3) можно проинтерпретировать на этапе генерации H -сети, и тогда нижние и верхние границы x и y станут равными 0 и 100 соответственно. Таким образом, на первом шаге процесса потоковых вычислений имеем

$$x = [0, 100]; \quad y = [0, 100];$$

В табл. 1 приведена последовательность шагов исполнения потокового процесса для данной задачи. В таблице предполагается, что на каждом шаге исполняется первая функция из множества активных функций (она отделена от остальных символом '|'). На первом шаге итерации исполняется функция f_1 . В результате работы этой функции получаем значение $y = [-8, 12]$. После этого вызывается функция присваивания $PRint(y)$. Текущее значение y равно $[0, 100]$, новое — $[-88, 12]$. Функция присваивания вырабатывает значение $[0, 12]$ и присваивает его y . Как видим, y переменной y изменилась только верхняя граница (так как новая нижняя граница (-88) меньше текущей (0)). Изменение значения y в целом все-таки произошло (Флаг = ∂a), поэтому вызывается функция проверки корректности $PRDint(y)$. Условие корректности для y ($0 \leq 12$) не нарушается, и процесс исполнения H -модели продолжается, т.е. происходит активация функций интерпретации f_2 и f_4 , для которых y является входным аргументом. Ввиду того, что эти функции уже входят в множество активных функций, их повторная активация не происходит. Далее исполняется следующая функция (f_2) из очереди.

Вычисления заканчиваются тогда, когда нижняя и верхняя границы как x , так и y становятся равными друг другу. Это произойдет при значениях x^{Lo} и x^{Up} равными 8, а y^{Lo} и y^{Up} — равными 4. При таких значениях исполнение любой функции f_1-f_4 не изменяет значения своего результата и множество активных функций становится пустым.

Таблица 1. Протокол исполнения H -модели

N	Активные функции	H-значения текущее новое	Флаг	Добавить функции
1	$f_1 f_2 \text{ } f_4$	$y=[0,100] [0,12]$	да	f_2, f_4

2	$f_2 f_3, f_4$	$x=[0,100] [0,12]$	да	f_1, f_3
3	$f_3 f_4, f_1$	$y=[0,12] [0,12]$	нет	
4	$f_4 f_1$	$x=[0,12] [0,6]$	да	f_1, f_3
5	$f_1 f_3$	$y=[0,12] [6,12]$	да	f_2, f_4
6	$f_3 f_2, f_4$	$y=[6,12] [6,12]$	нет	
7	$f_2 f_4$	$x=[0,6] [0,6]$	нет	
8	$f_4 $	$x=[0,6] [3,6]$	да	f_1, f_3
9	$f_1 f_3$	$y=[6,12] [6,9]$	да	f_2, f_4
10	$f_3 f_2, f_4$	$y=[6,9] [6,9]$	нет	
11	$f_2 f_4$	$x=[3,6] [3,6]$	нет	
12	$f_4 $	$x=[3,6] [3,4]$	да	f_1, f_3
13	$f_1 f_3$	$y=[6,9] [8,9]$	да	f_2, f_4
14	$f_3 f_2, f_4$	$y=[8,9] [8,8]$	да	f_2, f_4
15	$f_2 f_4$	$x=[3,4] [4,4]$	да	f_1, f_3
16	$f_4 f_1, f_3$	$x=[4,4] [4,4]$	нет	
17	$f_1 f_3$	$y=[8,8] [8,8]$	нет	
18	$f_3 $	$y=[8,8] [8,8]$	нет	

3. Примеры использования Н-моделей

Так как процесс вычислений в Н-моделях не зависит от типа задачи, аппарат Н-моделей используется для решения не только численных, но и задач из других предметных областей. Н-модели могут достаточно просто настраиваться на любые области. Для этого необходимо определить типы данных и ограничения характерные для той или иной предметной области. Кроме того, в отличие от других решателей, Н-модели могут использоваться и для решения так называемых смешанных задач, содержащие как численные (вещественные и целые) переменные, так и множества, перечисления, дискретные объекты и т.п.

Ввиду ограниченного объема статьи ниже рассмотрим только относительно небольшие примеры, показывающие основные возможности аппарата Н-моделей.

3.1 Решение комбинаторных задач

Комбинаторные задачи представляют собой предмет интенсивных исследований в области искусственного интеллекта. Сложность решения таких задач объясняется огромным пространством поиска при относительно малом количестве вычислений. Постоянно изобретаются все новые и новые эвристики, позволяющие сократить пространство поиска и находить решение за приемлемое время. Н-модели, за счет использования Н-значений вместо точных значений могут решать комбинаторные задачи и без привлечения такого рода эвристик.

Классическим примером комбинаторной задачи являются арифметические (буквенно-арифметические) головоломки. Одна из таких головоломок представляет собой шутивную телеграмму: SEND MORE MONEY. Если предположить, что слова в этой телеграмме представляют собой зашифрованные десятичные записи натуральных чисел, в которых каждая буква обозначает одну и ту же цифру (от 0 до 9), а разные буквы обозначают разные цифры и, наконец, старшие разряды чисел (обозначенные в данном примере буквами S и M) не равны нулю, то необходимо найти такую подстановку цифр вместо букв, при которой сумма двух первых чисел равна третьему:

$$\begin{array}{rcccc}
 c_4 & c_3 & c_2 & c_1 & 0 \\
 & S & E & N & D \\
 + & M & O & R & E \\
 \hline
 M & O & N & E & Y
 \end{array}$$

// c_i - это перенос из i -го столбца

За простотой формулировки скрывается трудоемкость решения: действуя простым перебором, мы должны рассмотреть максимум $8! = 40320$ вариантов различных подстановок.

Для того, чтобы решить такую задачу с использованием аппарата Н-моделей необходимо сопоставить каждой букве Н-значение, содержащее все цифры от 0 до 9. Кроме букв в Н-модели должны присутствовать переносы из одного столбца в другой. Их Н-значения равны множеству $\{0, 1\}$. Ограничения

представляются в виде линейных целочисленных уравнений, связывающих буквы из одного и того же столбца (с учетом переносов):

$$\begin{aligned} S + M + c3 &= O + 10*c4; \\ E + O + c2 &= N + 10*c3; \\ N + R + c1 &= E + 10*c2; \\ D + E + 0 &= Y + 10*c1; \\ \text{alldiff}(S, E, N, D, M, O, R, Y); \end{aligned}$$

Последнее ограничение означает, что значения всех букв должны быть различными. Заметим, что во втором и третьем столбцах уравнения выглядят (упрощенно) следующим образом: $E+O=N$ и $N+R=E$. Отсюда легко сделать вывод, что сумма O и R должна быть равна 10. С учетом соответствующих переносов такое уравнение выглядит следующим образом:

$$O + R + c1 = 9*c2 + 10*c3;$$

Описанный выше процесс удовлетворения ограничений за небольшое число шагов находит следующее решение:

$$S = 9; E = 5; N = 6; D = 7; M = 1; O = 0; R = 8; Y = 2; c1 = 1; c2 = 1; c3 = 0; c4 = 1.$$

3.2 Решение теоретико-множественных задач

Одним из основных нечисловых типов данных, используемых в Н-моделях являются множества. Впервые понятие недоопределенного множества было предложено А.С.Нариньяни в работе [10]. Существует несколько различных способов представления недоопределенного множества (см. [10], [15], [18]). Ниже мы используем способ, предложенный в [10].

Недоопределенное множество A представляется тремя компонентами (слотами):

$$A = \langle A^+, A^-, M \rangle,$$

где A^+ – множество элементов, которые точно принадлежат A ;

A^- – множество элементов, которые точно не принадлежат A ;

M – кардинал A , представленный недоопределенным целым числом.

Естественно, что количество элементов в A^+ и A^- может только возрастать. Если обозначить через X – универсум множества A , а через card – кардинал множества, то справедливы следующие ограничения:

$$\begin{aligned} M &\geq \text{card}(A^+); & M &\leq \text{card}(X) - \text{card}(A^-); \\ A^+ &\subseteq X; & A^- &\subseteq X; & A^+ \cap A^- &= \emptyset; \end{aligned}$$

Рассмотрим один простой пример. Предположим, что в универсуме, состоящем из 20 букв (от a до t), имеются 4 множества (A , B , C и D) со следующими значениями:

имя	элементы, принадлежащие множеству	элементы, не принадлежащие множеству	кардинал
A	$\{a, b, c, d, e, f, k, l, p\}$	$\{h, i\}$	
B	$\{k, l\}$	$\{g, h, i, j\}$	
C	$\{c, d, e, f\}$	$\{a, b, g\}$	
D	$\{o, p, q, r, s, t\}$	$\{\}$	

Также предположим, что модель состоит из следующих ограничений:

$$C = A \setminus B; \quad D \subseteq C; \quad \text{card}(A) \leq 14; \quad \text{card}(B) > 5;$$

После применения к данной Н-модели процесса удовлетворения ограничений, получаем следующий результат.

имя	элементы, принадлежащие множеству	элементы, не принадлежащие множеству	кардинал
A	$\{a, b, c, d, e, f, k, l, o, p, q, r, s, t\}$	$\{g, h, i, j, m, n\}$	$[14, 14]$
B	$\{a, b, k, l, m, n\}$	$\{c, d, e, f, g, h, i, j, o, p, q, r, s, t\}$	$[6, 6]$
C	$\{c, d, e, f, o, p, q, r, s, t\}$	$\{a, b, g, h, i, j, k, l, m, n\}$	$[10, 10]$
D	$\{o, p, q, r, s, t\}$	$\{a, b, g, h, i, j, k, l, m, n\}$	$[6, 10]$

Таким образом значения множеств A , B и C полностью уточнились. Значение множества D уточнилось, но не полностью: при текущих ограничениях осталось неясным принадлежат ли множеству D элементы c , d , e и f .

3.3 Решение смешанных задач

Основное преимущество Н-моделей состоит в том, что они способны решить задачи, которые не могут быть решены никаким другим способом (кроме, возможно, полного перебора). Это относится в первую очередь к так называемым *смешанным* задачам: когда в модели присутствуют переменные и ограничения самой различной природы (численные, дискретные, множественные, таблицы и т.п.).

Предположим, что к Н-модели из предыдущего раздела добавляются следующие два уравнения, равенство и логическая импликация над вещественными (x, y), целым (k) и множествами (D, A, C):

$$x^2 + 6*x = y - 2^k;$$

$$k*x + 7.7*y = 2.4;$$

$$k < 3;$$

$$(x < 2.5*y) \& (D \subset A) \rightarrow (k*y \leq 3) \& (k > y + 1) \& (C \subseteq D).$$

В такой модели результат будет следующим:

имя	элементы, принадлежащие множеству	элементы, не принадлежащие множеству	кардинал
A	{a,b,c,d,e,f,k,l,o,p,q,r,s,t}	{g,h,i,j,m,n}	[14,14]
B	{a,b,k,l,m,n}	{c,d,e,f,g,h,i,j,o,p,q,r,s,t}	[6,6]
C	{c,d,e,f,o,p,q,r,s,t}	{a,b,g,h,i,j,k,l,m,n}	[10,10]
D	{c,d,e,f,o,p,q,r,s,t}	{a,b,g,h,i,j,k,l,m,n}	[10,10]

$$k = 2; \quad x = -0.6586; \quad y = 0.48261.$$

Заключение

В статье рассмотрена технология программирования, основанная на концепции недоопределенности значений. Данная технология существенно повышает уровень спецификации проблемы, а также позволяет решать новые классы задач. Недоопределенные модели имеют широкий спектр приложений и предоставляют принципиально новые возможности для решения задач в таких областях, как вычислительная математика, инженерия знаний, проектирование, планирование и др. Они используются как база в ряде основных проектов, разрабатываемых в РосНИИ искусственного интеллекта:

* _____ NeMo+ (см. [25]) – объектно-ориентированная программная обстановка для недоопределенных вычислений с широким спектром типов данных,

* _____ SemP-ТАО (см. [26]) – технология создания сложных систем обработки знаний,

* _____ ТАО (см. [27]) – мультиагентная программная технология активных объектов,

* UniCalc (см.[13]) – решение алгебраических задач произвольной сложности,

* _____ Time-EX (см.[16]) – календарное планирование на основе неполных данных,

* Экономика – модель макроэкономики и Госбюджета РФ,

* ФинПлан – технология финансового планирования нового поколения

и некоторые другие проекты.

Опыт применения Н-моделей показал, что они являются не только удобным высокоуровневым средством спецификации задач, но и позволяют организовать вычисления достаточно эффективным способом. Поскольку в основе Н-моделей лежит недетерминированный параллельный вычислительный процесс, эффективность их работы может быть повышена во много раз при реализации на компьютерах с параллельной архитектурой. Проведение соответствующих экспериментов является одним из направлений дальнейшего развития.

Еще одно направление исследований связано с разработкой моделей и систем, обеспечивающих обработку недоопределенной информации в сочетании с неточностью, неоднозначностью и т. п. [28].

Для расширения класса решаемых задач перспективной является интеграция Н-моделей с другими технологиями программирования. Вопрос интеграции с объектно-ориентированным подходом достаточно хорошо проработан и нашел свое отражение в первых трех из перечисленных выше проектов.

В настоящее время ведутся работы по интеграции недоопределенных вычислений с системами логического и продукционного программирования и с базами данных.

Литература

1. Montanari U. Networks of Constraints: Fundamental Properties and Applications to Picture Processing // *Inform. Sci.* –V.7, 1974. – P. 95 – 132.
2. А.С.Нариньяни. Модель или алгоритм: новая парадигма информационной технологии.// “Информационные технологии”, № 4, М., 1997
3. Colmerauer A. An introduction to Prolog III. *Communications of the ACM*, 33(7), July 1990. – P. 69 – 90.
4. Jaffar J., Michayov S., Stuckey P., Yap R. The CLP(R) language and system. *ACM Transactions on Programming Languages and Systems*, 14(3), July 1992. – P. 339 – 395.
5. Benhamou F., Older W.J. Applying Interval Arithmetic to Real, Integer and Boolean Constraints. *Journal of Logic Programming*, 1996.
6. Diaz D., Codognot P. A minimal extension of the WAM for clp(FD). *Proceedings of the 10th International Conference on Logic Programming*, 1994.
7. Van Hentenryck P. Constraint Satisfaction in Logic Programming. *Logic Programming Series*. MIT Press, Cambridge, MA, 1989.
8. Puget J.-F.. A C++ Implementation of CLP. *Tech. Report*, Plog, January 1994.
9. Benhamou F., McAllester D., Van Hentenryck P. CLP(Intervals) Revisited. *Proceedings of ILPS'94*, Ithaca, NY, 1994. – P. 124 – 138.
- 10.Нариньяни А.С. Недоопределенные множества – новый тип данных для представления знаний. – Новосибирск, 1980. – 28с. (*Препр./АН СССР. Сиб. отд-ние. ВЦ; № 232*).
- 11.Нариньяни А.С. Недоопределенные модели и операции с недоопределенными значениями. – Новосибирск, 1982. – 33с. (*Препр./ АН СССР. Сиб. отд-ние ВЦ; № 400*).
- 12.Нариньяни А.С. Недоопределенность в системах представления и обработки знаний // *Изв. АН СССР. Техн. кибернетика*, 1986. № 5. – С. 3 – 28.
- 13.Semenov A.L., Leshchenko A.S. Interval and Symbolic Computations in the UniCalc Solver // *Inter. Conf. on Interval and Computer-Algebraic Methods in Science and Engineering (INTERVAL-94): Abstracts*. St-Petersburg, Russia, 1994. – P. 206 – 208.
- 14.Телерман В.В., Дмитриев В.Е. Технология программирования на основе недоопределенных моделей. – Новосибирск, 1995. – 43 с. (*Препр./ РАН. Сиб. отд-ние. ИСИ; № 25*).
- 15.Yakhno T.M., Petrov E.S. Application of Subdefinite Models for Solving Constraint Satisfaction Problems // *Perspectives of Systems Informatics, Lecture Notes in Computer Science*, **1181**, 1996. – P. 80 – 90.
- 16.Нариньяни А.С., Иванов Д.А., Седреев С.В., Фролов С.А. Недоопределенное календарное планирование: новые возможности. // “Информационные технологии”, № 1, М., 1997
- 17.Телерман В.В., Ушаков Д.М. Недоопределенные модели: формализация подхода и перспективы развития // *Проблемы представления и обработки не полностью определенных знаний*, сб. трудов РосНИИ Искусственного Интеллекта / ред. И.Е. Швецов, Москва – Новосибирск. – 1996. – С. 7 – 30.
- 18.Telerman V.V., Ushakov D.M. Subdefinite Models as a Variety of Constraint Programming // *Proc. of the Intern. Conf. Tools of Artificial Intelligence (ICTAI'96)*, Toulouse, 1996.
- 19.Hyvonen E. Constraint Reasoning Based on interval Arithmetic: the tolerance propagation approach, *Artificial Intelligence*, V.58, 1992, P. 71 – 112.
- 20.Яковлев А.Г. Машинная арифметика мультиинтервалов // *Вопросы кибернетики. Проблемно-ориентированные вычислительные системы*. 1987. – С. 66 – 81.

- 21.Телерман В.В. Использование мультиинтервалов в недоопределенных моделях // *X Всесоюз. семинар Параллельное программирование и высокопроизводительные системы: Методы представления знаний в информационных технологиях* (Уфа, 19 – 26 июня): Тез. докл. – Киев, 1990. – С. 128 – 129.
- 22.Алефельд Г., Херцбергер Ю. Введение в интервальные вычисления. – М.: Мир, 1987. – 260 с.
- 23.Нариньяни А.С., Дмитриев В.Е., Телерман В.В. Специализированный виртуальный потоковый процессор для баз знаний // *Системы обработки знаний и изображений: Тр. / Междунар. рабочей конф. по комплексным научным проектам КНП-1 и КНП-2, Смоленице, ноябрь 1986.* – Смоленице, 1986. – Том II. – С. 19 – 22.
- 24.Телерман В.В. Применение обобщенных вычислительных моделей для реализации вывода/вычислений в базах знаний // *Проблемы развития и освоения интеллектуальных систем: Тез. докл. Всесоюз. конф., Секция II: Методы и модели освоения интеллектуальных систем.* – Новосибирск, 1986. – С. 80 – 81.
- 25.Shvetsov I.E., Telerman V.V., Ushakov D.M. NeMo+: Object-Oriented Constraint Programming Environment Based on Subdefinite Models // *Third Intern. Conf. on Principles and Practice of Constraint Programming (CP'97), Lecture Notes in Comp. Sci. N 1330, Linz, Austria, P. 373 -- 385.*
- 26.Загорулько Ю.А., Попов И.Г. Представление знаний в интегрированной технологической среде Semp-ТАО // *Проблемы представления и обработки не полностью определенных знаний, сб. трудов РосНИИ Искусственного Интеллекта / ред. И.Е. Швецов, Москва – Новосибирск. – 1996.* – С. 59 – 74.
- 27.Швецов И.Е., Нестеренко Т.В., Старовит С.А., Титова М.В. Технология активных объектов: от концепции к реализации // *Там же.* – С. 88 – 100.
- 28.Нариньяни А.С. Неточность как НЕ-фактор. Попытка доформального анализа.– Новосибирск, 1994.– 34 с. (Препр./Рос. НИИ ИИ; 2).