Лекция №3

ТЕХНОЛОГИЯ КОНЦЕПТУАЛЬНОГО ПРОГРАММИРОВАНИЯ (ТКП)

ТКП

Технология концептуального программирования (ТКП) – одна из старейших и наиболее развитых в ИИ как в теоретическом, так и в практическом аспектах. Она разработана советскими учеными из России и Эстонии.

ТКП предназначена для **синтеза программ решения задач по их описанию на ограниченном естественном языке (ОЕЯ) при некоторых ограничениях**.

Эти ограничения требуют:

1) точного указания предметной области (ПрО), к которой относится решаемая задача;

2) фиксации класса решаемых задач.

Последние получили название **вычислительных** или **расчетно-логических задач**.

Постановка задачи ТКП

В общем случае их описание на ОЕЯ имеет вид:

$$\textbf{Зная } \boldsymbol{M}\textbf{, вычислить } (\boldsymbol{y_1, \ldots, y_n}) \textbf{ по } (\boldsymbol{x_1, \ldots, x_m}) \qquad (1)$$

В выражении (1) ***M*** идентифицирует ПрО (например, тригонометрию, кинематику и т.д.).

Кортеж $(\boldsymbol{x_1, \ldots, x_m})$ содержит идентификаторы переменных с известными значениями, а кортеж $(\boldsymbol{y_1, \ldots, y_n})$ – идентификаторы переменных, значения которых требуется определить.

Существенным ограничением ТКП является предположение, что в компьютере имеется модель ПрО (МПрО), с которой можно манипулировать.

Подходы к синтезу программ в ТКП

В **ТКП** для представления МПрО используются **семантические сети специального вида**, называемые **вычислительными моделями (ВМ)**.

Известны четыре **подхода к синтезу программ**:

1) дедуктивный;
2) индуктивный;
3) трансформационный;
4) утилитарный.

В **ТКП** используются **первые два подхода**.

Основная идея ТКП

Основная идея ТКП состоит в следующем:

Пусть существует постановка задачи в виде (1), необходимо:

1) перейти от (1) к теореме существования решения данной задачи;

2) построить доказательство теоремы существования;

3) извлечь из доказательства программу решения задачи.

При реализации этого метода получаем два важных результата:

1) программа точно соответствует описанию задачи;

2) вместо отладки программы выполняется «отладка» описания задачи.

Концептуализацией называется **процесс перехода от описания ПрО на ОЕЯ к точной спецификации этого описания на некотором формальном языке, ориентированном на компьютерное представление.**

Математический аппарат концептуализации

В качестве математического аппарата концептуализации в рамках ТКП разработаны ВМ. Они являются разновидностями семантических сетей. **Семантическая сеть *S*** в общем виде определяется следующим образом:

$$\boldsymbol{S=(O,\ R) = (\{o_i|\ i = 1,2,\ \dots\ ,\ k\},\ \{r_i|\ j = 1,2,\ \dots\ ,\ l\}),} \qquad \textbf{(2)}$$

где $\boldsymbol{O}$ – множество объектов ПрО ($|\boldsymbol{O}|=\boldsymbol{k}$);

$\boldsymbol{R}$ – множество отношений между объектами ПрО ($|\boldsymbol{R}|=\boldsymbol{l}$).

ВМ для заданной ПрО определяется как кортеж:

$$\boldsymbol{(\{p_i\},\ \{f_j\},\ \{u_k\}),} \qquad \textbf{(3)}$$

где $\boldsymbol{p_i}$ — имя понятия ПрО;

$\boldsymbol{f_j}$ — функциональное отношение между понятиями;

$\boldsymbol{u_k}$ — управляющая структура.

Математический аппарат концептуализации

Функциональное отношение f_j задается тройкой **плекс-элементов**.

$$f_j = (X_j, F_j, Y_j), \qquad (4)$$

где $X_j = (x_{j1}, \ldots , x_{jmj})$ — набор входных переменных для f_j;
F_j — ссылка на процедуру, реализующую вычисление $Y_j = F_j(X_j)$;
$Y_j = (y_{j1}, \ldots , y_{jnj})$ — набор выходных переменных для f_j.

Управляющие структуры u_k реализуют отображения X_j и Y_j в множество разрешенных типов данных, а также они позволяют приписывать переменным известные и вычисленные значения.

Концептуализация ПрО

Графически **концептуализация ПрО** в рамках ВМ изображается **графом *G***:

$$\boldsymbol{G = (V, U) = (\{x_i\} * \{y_j\} * \{F_l\}, \{u_k\})} \qquad \textbf{(5)}$$

Процесс доказательства теоремы существования решения задачи (1) отображается на **графе** как **«волновой процесс»**, начинающийся в вершинах $\boldsymbol{(x_1, \ldots, x_m)}$ и заканчивающийся, когда **«волна»** достигнет всех вершин $\boldsymbol{(y_1, \ldots, y_n)}$. При волновой интерпретации можно детализировать постановку задачи (1) и выделить четыре класса задач.

1. **Задачи на доказательство.**
2. **Задачи на вычисление значений переменных.**
3. **Задачи на прогнозирование.**
4. **Задачи планирования эксперимента.**

Теорема существования решения

Рассмотрим теорему существования решения задачи в постановке (1).

Обозначим $\boldsymbol{P(x)}$ предикат входных условий, а $\boldsymbol{R(x, y)}$ — предикат выходных условий, $\boldsymbol{x = (x_1, \dots , x_m), y = (y_1, \dots , y_n)}$.

Запишем **теорему существования** в виде

$$\forall \boldsymbol{x}(\boldsymbol{P}(\boldsymbol{x}) \Rightarrow \exists \boldsymbol{y}\, \boldsymbol{R}(\boldsymbol{x},\boldsymbol{y})) \qquad (6)$$

Будем рассматривать только конструктивные логические теории. Н.Н. Непейвода доказал, что различные определения реализуемости эквивалентны. Он же показал, что существует реализуемость, при которой формулам вида $\exists \boldsymbol{y}\, \boldsymbol{R(y)}$ будет соответствовать либо программа вычисления $\boldsymbol{y}$, либо само значение $\boldsymbol{y}$. Тогда **любой доказуемой формуле будет соответствовать программа**.

Теорема существования решения

Предполагается, что реализации всех аксиом заданы априорно. Для каждого правила вывода

$$\Pi \quad \frac{\boldsymbol{A_1}, \dots, \boldsymbol{A_k}}{\boldsymbol{A}} \qquad \text{(или просто } \frac{\boldsymbol{A_1}, \dots, \boldsymbol{A_k}}{\boldsymbol{A}}\text{)}$$

заданы правила построения реализации выводимой по этому правилу формулы $\boldsymbol{A}$ по реализациям формул $\boldsymbol{A_1}, \dots, \boldsymbol{A_k}$.

Тогда реализация любой выводимой формулы может быть построена прямо по выводу формулы.

Обычно в качестве конструктивной логической теории используют интуиционистскую логику, в которой неприменимы законы снятия двойного отрицания и закон исключенного третьего.

Для каждого правила вывода в ней записываются программные конструкции, дающие реализации формул, выводимых по этому правилу.

Конструктивные доказательства

Конструктивные доказательства имеют следующие **особенности**:

• на каждом шаге доказательства применяется некоторое правило вывода;

• в качестве посылок используются только аксиомы или ранее доказанные формулы;

• в доказательстве отсутствуют циклы;

• некоторые шаги доказательства могут использовать леммы, для которых строятся вспомогательные доказательства.

Каждый шаг доказательства преобразуется во фрагмент программы отдельно от других шагов.

Построенные таким способом программы являются хорошо структурированными.

Существуют **два способа извлечения программы из доказательства**.

При первом реализации формул используются непосредственно, поэтому **программой является реализация теоремы существования решения**.

Второй способ извлечения программы заключается в составлении ее оператор за оператором из шагов доказательства теоремы существования (линейный вывод). В этом случае **программа состоит из операторов присваивания и операторов вызова процедур**.

Рассмотренная система правил вывода не содержала правил для индукции, поэтому в программах не было циклов, но применяя разные схемы индукции, можно получить разные схемы циклов.