

§ 9. Полиморфизм

От типа данных зависят допустимые методы обработки этих данных.

Цель: использовать один код (метод) для работы с разными типами данными.

Опр. Полиморфным называется любой структурный элемент программы, если он может выполняться с аргументами разных типов.

Структурными элементами программы можно называть выражения, инструкции, функции (в смысле подпрограммы), классы, методы и т.п..

В данном параграфе рассматривается «параметрический полиморфизм», использующий параметры или переменные типа.

Опр. Переменной типа называется переменная, значением которой является выражение типа.

В атрибутивной грамматике APG значение атрибута `.type` тоже является переменной типа. Но для полиморфных операторов понадобятся другие переменные.

Пример 1.

17	$E \rightarrow E_1[E_2]$	$E.type := \begin{cases} t, & \text{если } E_2.type = \text{int}, E_1.type = \text{array}(N, t) \\ \text{type_error} \end{cases}$
----	--------------------------	--

Выражение $E_1[E_2]$, выводимое правилом вывода 17 в грамматике APG, называется индексирования массива. Переменная t в соответствующем семантическом правиле является переменной типа.

Оператору индексирования массива соответствует функция (отображение) $(\text{array}(N, t) \times \text{int}) \mapsto t$.

Более точно выражение типа данной функции записывают с использованием квантора общности:

$\forall t ((\text{array}(N, t) \times \text{int}) \mapsto t)$.

Такое выражение будем называть полиморфным типом.

Пример 2.

16	$E \rightarrow E_1 \uparrow$	$E.type := \begin{cases} t, & \text{если } E_1.type = ptr(t) \\ type_error \end{cases}$
----	------------------------------	--

Выражение $E_1 \uparrow$, выводимое правилом вывода 16 в грамматике APG, называется разыменованное указателя. Переменная t в соответствующем семантическом правиле является переменной типа.

Оператору разыменованное указателя соответствует функция (отображение) $ptr(t) \mapsto t$.

Соответствующий полиморфный тип:

$\forall t (ptr(t) \mapsto t)$.

В каждом полиморфном операторе связанная переменная может быть заменена на другую переменную.

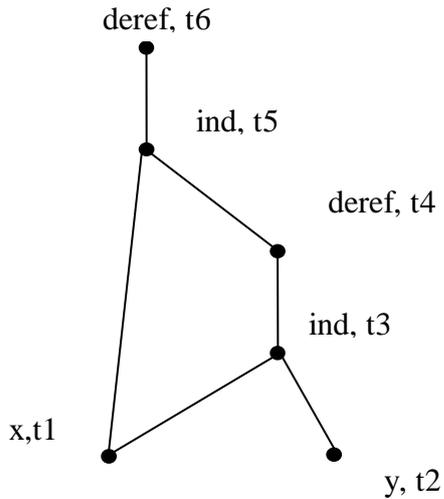
Например,

$\forall p (ptr(p) \mapsto p)$ эквивалентно полиморфному типу и примера 2.

Замена связанных переменных применяется при проверке типов.

Пример 3. Вычисление типа в выражении $x[x[y]^\uparrow]^\uparrow$

Обозначим `ind` оператор индексирования, `deref` оператор разыменования.
Тогда выражению соответствует даг.



Каждому узлу сопоставим переменную типа t_1, t_2, \dots, t_6 , отвечающую за тип подвыражения в поддереве с корнем в этом узле.

Покажем процесс вычисления значения переменных снизу вверх.

Оператор $\text{ind} = \forall t ((\text{array}(\mathbb{N}, t) \times \text{int}) \mapsto t)$ преобразуется в $(\text{array}(\mathbb{N}, t_3) \times \text{int}) \mapsto t_3$.

Следовательно, $t_1 = \text{array}(\mathbb{N}, t_3)$, $t_2 = \text{int}$.

Оператор $\text{deref} = \forall t (\text{ptr}(t) \mapsto t)$ преобразуется в $\text{ptr}(t_4) \mapsto t_4$.

Следовательно, $t_3 = \text{ptr}(t_4)$.

Второй оператор ind преобразуется в $(\text{array}(\mathbb{N}, t_5) \times \text{int}) \mapsto t_5$.

Следовательно, $t_4 = \text{int}$, $t_3 = t_5 = \text{ptr}(\text{int})$, $t_1 = \text{array}(\mathbb{N}, \text{ptr}(\text{int}))$.

Второй оператор deref преобразуется в $\text{ptr}(t_6) \mapsto t_6$. Следовательно,

$t_5 = \text{ptr}(\text{int}) = \text{ptr}(t_6)$, $t_6 = \text{int}$.

Следовательно, для правильного вычисления выражения идентификатор x должен иметь тип `array(N, ptr(int))`, а y – тип `int`.

В процессе анализа типов мы нашли подстановку, унифицирующую множество полиморфных типов.

Опр. Пусть w – выражение типа, содержащее переменные типа, t – набор переменных типа, S – набор замен переменных из t на выражения типа. $S(w)$ – результат замены каждого вхождения в w переменной из t на выражение из S .

S называется подстановкой.

$S(w)$ – экземпляром w .

Опр. Подстановка S называется унификатором выражений w_1 и w_2 , если $S(w_1) = S(w_2)$.

Опр. Подстановка S называется наиболее общим унификатором выражений w_1 и w_2 , если для любого унификатора S' выполняется, что $S'(w_1)$ является экземпляром $S(w_1)$ и $S'(w_2)$ является экземпляром $S(w_2)$.