

# Лекция 6а. Понятие свёртки

Коновалов А. В.

27 сентября 2022 г.

**Свёртка** — объединение нескольких значений одной операцией.  
Примеры: вычислить сумму нескольких чисел, произведение нескольких чисел и т.д.

$$a \cdot b \cdot c \cdot \dots \cdot k$$

Здесь знаком  $\cdot$  обозначена некоторая двуместная операция.

Свёртка может быть *правой* и *левой*.

Правая свёртка:

$$a \cdot (b \cdot (c \cdot (\dots \cdot k)\dots))$$

Левая свёртка:

$$((\dots(a \cdot b) \cdot c) \dots \cdot k)$$

В Scheme некоторые операции обладают свойством свёртки, такие функции могут принимать произвольное количество параметров.

1. Сложение:  $(+ \ 1 \ 2 \ 3 \ 4) \rightarrow 10$
2. Умножение:  $(* \ 1 \ 2 \ 3 \ 4) \rightarrow 24$
3. Вычитание:  $(- \ 10 \ 5 \ 3) \rightarrow 2$
4. Деление:  $(/ \ 120 \ 6 \ 5) \rightarrow 4$
5. Функции min и max:  $(\min \ 3 \ 8 \ 2 \ 5) \rightarrow 2, (\max \ 3 \ 8 \ 2 \ 5) \rightarrow 8.$
6. Конкатенация списков:  $(\text{append} \ '(a \ b) \ '(c \ d \ e) \ '(f \ g)) \rightarrow (a \ b \ c \ d \ e \ f \ g).$
7. Конкатенация строк:  $(\text{string-append} \ "ab" \ "cde" \ "fg") \rightarrow "abcdefg".$

Особые формы (and ...) и (or ...) тоже обладают свойством свёртки, не смотря на то, что это не функции.

Для свёрток определена такая аксиома, что если or — свёрточная операция, то

$$(or\ x) \equiv x$$

кроме - и / — ( $-\ x$ ) меняет знак числа, ( $/\ x$ ) — вычисляет обратное значение.

Иногда для свёртки может быть определён некоторый нейтральный элемент  $z$ :

$$((a \cdot b) \cdot c) = (((z \cdot a) \cdot b) \cdot c)$$

$$(a \cdot (b \cdot c)) = (a \cdot (b \cdot (c \cdot z)))$$

Для ряда свёрточных операций существует нейтральный элемент. Он возвращается при вызове операции без параметров:

<code>(+)</code>	$\rightarrow 0$
<code>(*)</code>	$\rightarrow 1$
<code>(append)</code>	$\rightarrow '()$
<code>(string-append)</code>	$\rightarrow ""$
<code>(and)</code>	$\rightarrow \#t$
<code>(or)</code>	$\rightarrow \#f$

Помимо свёрточных операций (+, \* и т.д.) в Scheme произвольное число параметров принимают операции арифметических отношений:

(= 1 1 1)	→ #t
(= 1 2 1)	→ #f
(> 7 5 2)	→ #t
(> 7 5 5)	→ #f
(≥ 7 5 5 2)	→ #t
...	

Операции арифметических отношений не являются свёрточными, т.к. запись вида  $(a < b) < c$  бессмысленна, но здесь упоминаются для полноты картины.

Для вызова свёрточных операций часто используется функция `apply`:

```
(define xs '(1 2 3 4))
```

```
(apply * xs)           → 24
```

```
(apply + xs)           → 10
```

```
(apply ≤ xs)           → #t
```

Назначение функции `apply`:

1. Вызов свёрточных операций.
2. Вызов функции с неизвестным числом аргументов.

При помощи `apply` невозможно вызвать `and` и `or`, поскольку они не функции (попытка вызова будет ошибкой синтаксиса).