Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

по курсу «Функциональное и логическое программирование»

Студент Рунов К.А.

Группа ИУ7-64Б

Оценка (баллы) _____

Преподаватели Толпинская Н. Б., Строганов Ю. В.

2024 г.

СОДЕРЖАНИЕ

1 Практические задания

1.1 Задание 1

Представить следующие списки в виде списочных ячеек:

1) '(open close halph)

2) '((open1)(close2)(halph3))

3) '((one) for all (and (me (for you))))

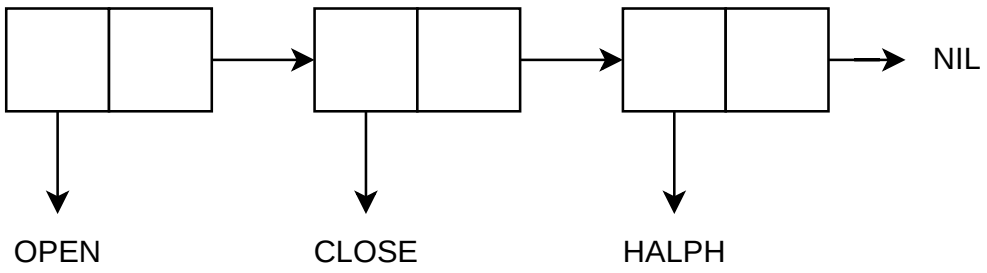
4) '((TOOL)(call))

5) '((TOOL1)((call2))((sell)))

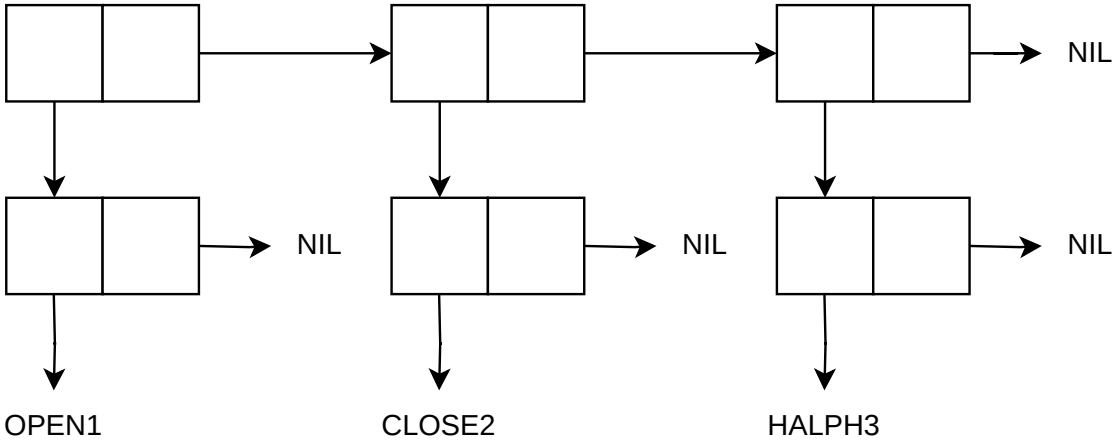
6) '(((TOOL)(call))((sell)))

Решение

1) '(open close halph)



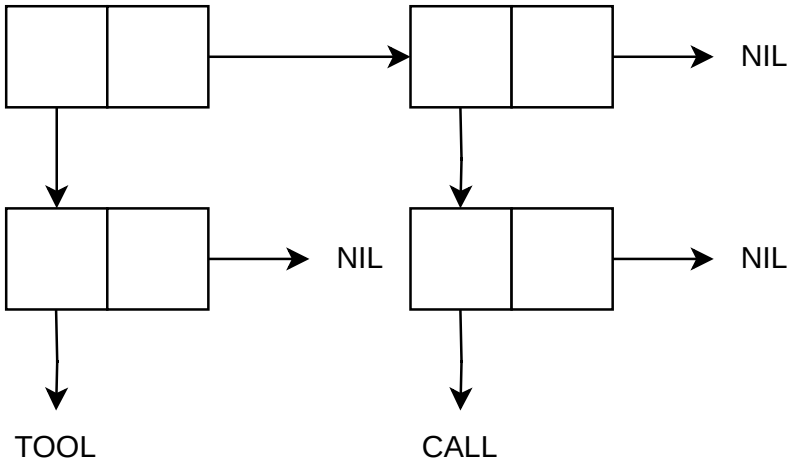
2) '((open1)(close2)(halph3))



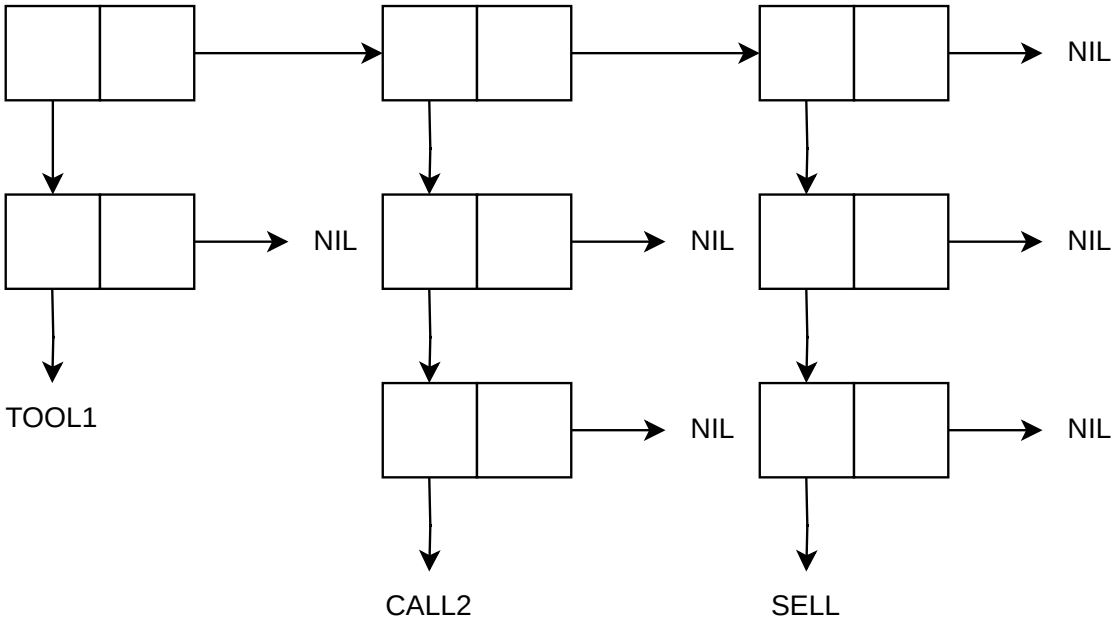
3) '((one) for all (and (me (for you))))

См. Рисунок 2 в приложении А.

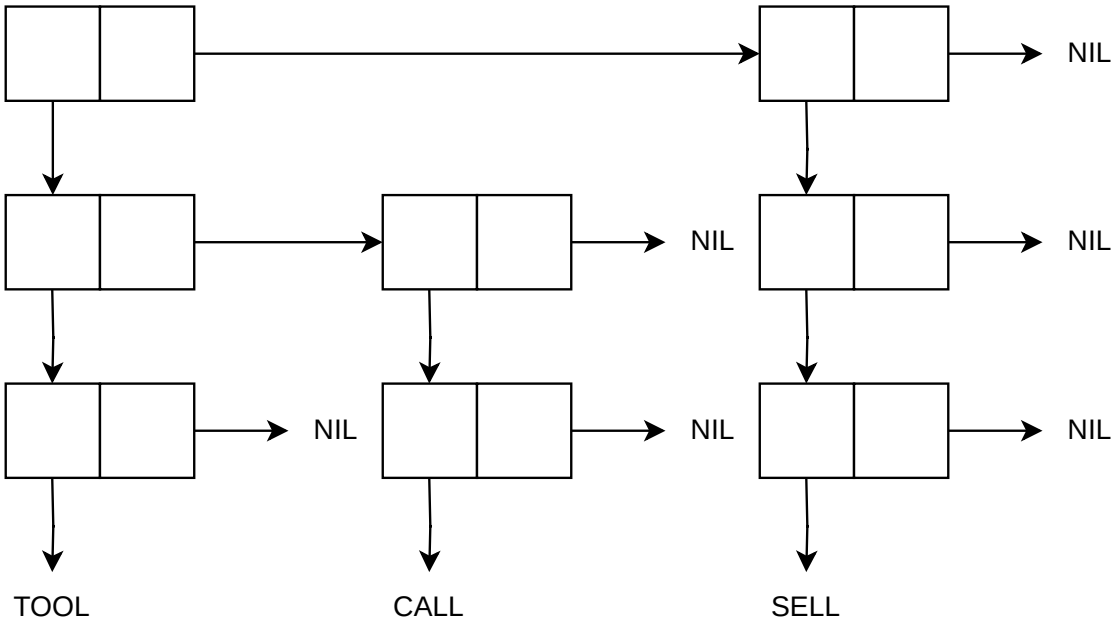
4) '((TOOL)(call))



5) '((TOOL1)((call2))((sell)))



6) '(((TOOL)(call))((sell)))



1.2 Задание 2

Используя только функции CAR и CDR, написать выражения, возвращающие

1) второй,

2) третий,

3) четвертый

элементы заданного списка L.

Решение

1) Выражение, возвращающее второй элемент L:

```
(car (cdr L))
```

2) Выражение, возвращающее третий элемент L:

```
(car (cdr (cdr L)))
```

3) Выражение, возвращающее четвертый элемент L:

```
(car (cdr (cdr (cdr L))))
```

1.3 Задание 3

Что будет в результате вычисления выражений?

1) `(CAADR '((blue cube)(red pyramid)))`

2) `(CDAR '((abc)(def)(gji)))`

3) `(CADR '((abc)(def)(ghi)))`

4) `(CADDR '((abc)(def)(ghi)))`

Решение

1) Результат выполнения (CAADR '((blue cube)(red pyramid))) — RED;

2) Результат выполнения (CDAR '((abc)(def)(ghi))) — NIL;

3) Результат выполнения (CADR '((abc)(def)(ghi))) — DEF;

4) Результат выполнения (CADDR '((abc)(def)(ghi))) — GHI.

1.4 Задание 4

Напишите результат вычисления выражений и объясните, как он получен:

1) `(list 'Fred 'and 'Wilma)`

2) `(list 'Fred '(and Wilma))`

3) `(cons Nil Nil)`

4) `(cons T Nil)`

5) `(cons Nil T)`

6) `(list Nil)`

7) `(cons '(T) Nil)`

8) `(list '(one two) '(free temp))`

9) `(cons 'Fred '(and Wilma))`

10) `(cons 'Fred '(Wilma))`

11) `(list Nil Nil)`

12) `(list T Nil)`

13) `(list Nil T)`

14) `(cons T (list Nil))`

15) `(list '(T) Nil)`

16) `(cons '(one two) '(free temp))`

Решение

1) Выражение: (list 'Fred 'and 'Wilma);

   Аргументы list:

   — Fred,

   — and,

   — Wilma;

   Результат вычисления: (FRED AND WILMA);

   Объяснение: Функция list всегда возвращает список из своих аргументов. В данном выражении ей были переданны аргументы Fred, and и Wilma.

2) Выражение: (list 'Fred '(and Wilma));

   Аргументы list:

   — Fred,

   — (and Wilma);

   Результат вычисления: (FRED (AND WILMA));

   Объяснение: В данном выражении list были переданны аргументы Fred и (and Wilma).

3) Выражение: (cons Nil Nil);

   Результат вычисления: (NIL);

   Объяснение: Функция cons принимает два аргумента, и возвращает точечную пару или списковую ячейку, в которой указатель CAR указывает на первый аргумент, а указатель CDR — на второй аргумент. В данном выражении cons были переданны аргументы Nil и Nil, в результате чего была возвращена списковая ячейка, в которой указатели CAR и CDR указывают на Nil, а (Nil . Nil) эквивалентно (Nil).

4) Выражение: (cons T Nil);

   Результат вычисления: (T);

Объяснение: В данном выражении cons были переданны аргументы T и Nil, в результате чего была возвращена списковая ячейка, в которой указатель CAR указывает на T, а указатель CDR — на Nil. (T . Nil) эквивалентно (T).

5) Выражение: (cons Nil T);

   Результат вычисления: (NIL . T);

   Объяснение: В данном выражении cons были переданны аргументы Nil и T, в результате чего была возвращена точечная пара, в которой указатель CAR указывает на Nil, а указатель CDR — на T.

6) Выражение: (list Nil);

   Аргументы list:

   — Nil;

   Результат вычисления: (NIL);

   Объяснение: В данном выражении list был передан только один аргумент — Nil, следовательно, в результате вычисления выражения был создан список из одного элемента (Nil).

7) Выражение: (cons '(T) Nil);

   Результат вычисления: ((T));

   Объяснение: В данном выражении cons были переданны аргументы (T) и Nil, в результате чего была возвращена списковая ячейка, в которой указатель CAR указывает на список (T), состоящий из одного элемента (T), а указатель CDR — на Nil. ((T) . Nil) эквивалентно ((T)).

8) Выражение: (list '(one two) '(free temp));

   Аргументы list:

   — (one two),

   — (free temp);

   Результат вычисления: ((ONE TWO) (FREE TEMP));

Объяснение: В данном выражении list были переданны аргументы (one two) и (free temp).

9) Выражение: (cons 'Fred '(and Wilma));

   Результат вычисления: (FRED AND WILMA);

   Объяснение: В данном выражении cons были переданны аргументы Fred и (and Wilma), в результате чего была возвращена списковая ячейка, в которой указатель CAR указывает на Fred, а указатель CDR — на список (and Wilma), состоящий из двух списковых ячеек, в котором в первой ячейке указатель CAR указывает на and, а CDR — на список (Wilma), в котором указатель CAR указывает на Wilma, а CDR — на Nil. (Fred . (and Wilma)) эквивалентно (Fred and Wilma).

10) Выражение: (cons 'Fred '(Wilma));

    Результат вычисления: (FRED WILMA);

    Объяснение: В данном выражении cons были переданны аргументы Fred и (Wilma), в результате чего была возвращена списковая ячейка, в которой указатель CAR указывает на Fred, а указатель CDR — на список (Wilma), в котором указатель CAR указывает на Wilma, а CDR — на Nil. (Fred . (Wilma)) эквивалентно (Fred Wilma).

11) Выражение: (list Nil Nil);

    Аргументы list:

    — Nil,

    — Nil;

    Результат вычисления: (NIL NIL);

    Объяснение: В данном выражении list были переданны два аргумента: Nil и Nil.

12) Выражение: (list T Nil);

    Аргументы list:

    — T,

— Nil;

Результат вычисления: (T NIL);

Объяснение: В данном выражении list были переданны аргументы T и Nil.

13) Выражение: (list Nil T);

Аргументы list:

— Nil,

— T;

Результат вычисления: (NIL T);

Объяснение: В данном выражении list были переданны аргументы Nil и T.

14) Выражение: (cons T (list Nil));

Результат вычисления: (T NIL);

Объяснение: При вычислении выражений в языке Lisp сначала вычисляются их аргументы. В результате вычисления T будет получено T. В результате вычисления (list Nil) будет получен список (NIL) из одного элемента — Nil. В результате вычисления (cons T (NIL)) будет получена списковая ячейка, в которой указатель CAR указывает на T, а указатель CDR — на список (Nil), в котором указатели CAR и CDR указывают на Nil. (T . (NIL)) эквивалентно (T NIL).

15) Выражение: (list '(T) Nil);

Аргументы list:

— (T),

— Nil;

Результат вычисления: ((T) NIL);

Объяснение: В данном выражении list были переданны аргументы (T) и Nil.

16) Выражение: (cons '(one two) '(free temp));

Результат вычисления: ((ONE TWO) FREE TEMP);

Объяснение: В данном выражении cons были переданны аргументы (one two) и (free temp), в результате чего была возвращена списковая ячейка, в которой указатель CAR указывает на список (one two), а указатель CDR — на список (free temp).

1.5 Задание 5

Написать лямбда-выражение и соответствующую функцию:

1) Написать функцию (f ar1 ar2 ar3 ar4), возвращающую список ((ar1 ar2) (ar3 ar4)).

2) Написать фукнцию (f ar1 ar2), возвращающую ((ar1) (ar3)).

3) Написать функцию (f ar1), возвращающую (((ar1))).

4) Представить результаты в виде списочных ячеек.

Решение

1) Функция:

```
(defun f (ar1 ar2 ar3 ar4) (cons (cons ar1 (cons ar2 nil))
  (cons (cons ar3 (cons ar4 nil)) nil)))
```

Лямбда-выражение:

```
(lambda (ar1 ar2 ar3 ar4) (cons (cons ar1 (cons ar2 nil))
  (cons (cons ar3 (cons ar4 nil)) nil)))
```

2) Функция:

```
(defun f (ar1 ar2) (cons (cons ar1 nil) (cons (cons ar2 nil)
  nil)))
```

Лямбда-выражение:

```
(lambda (ar1 ar2) (cons (cons ar1 nil) (cons (cons ar2 nil)
  nil)))
```

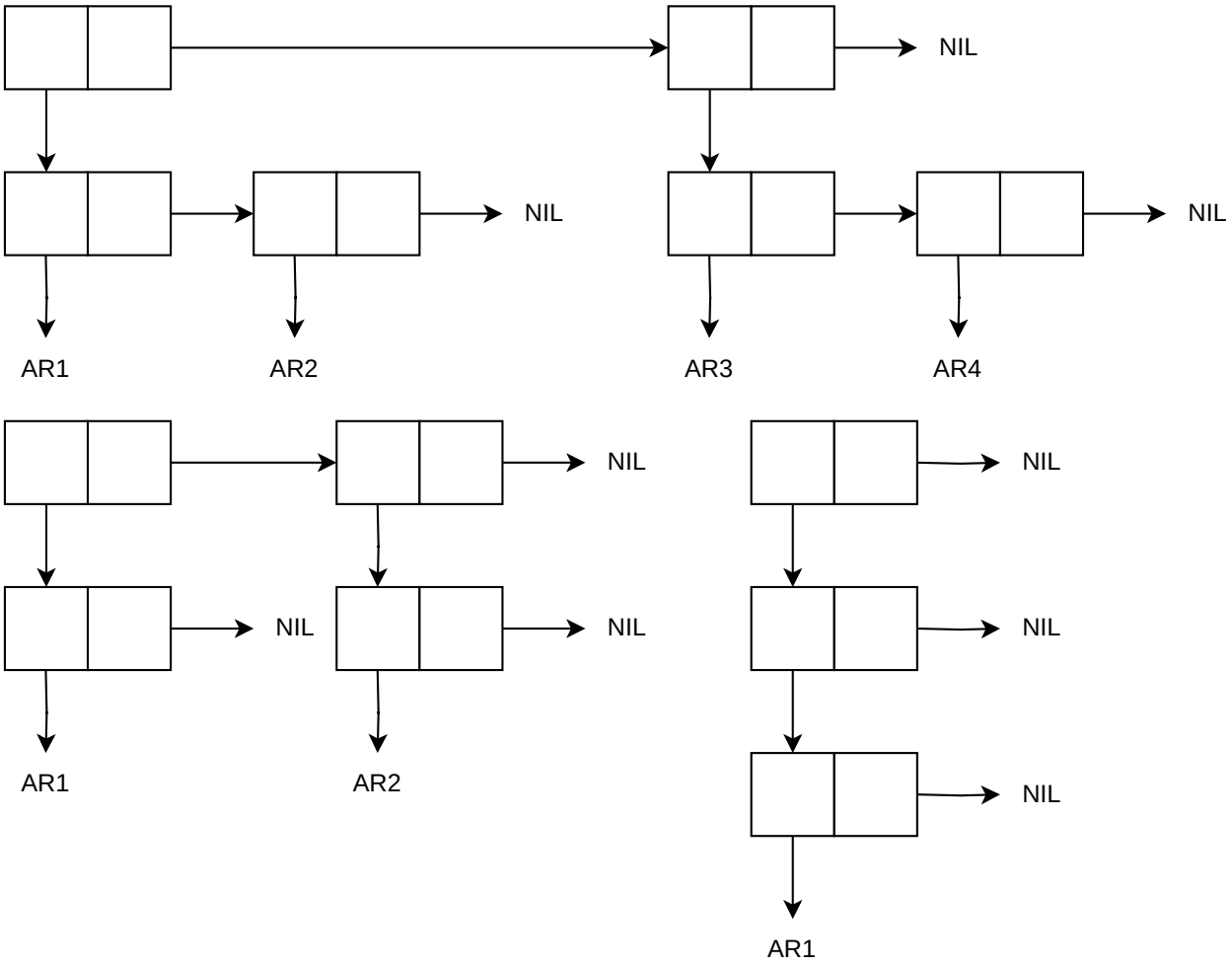
3) Функция:

```
(defun f (ar1) (cons (cons (cons ar1 nil) nil) nil))
```

Лямбда-выражение:

```
(lambda (ar1) (cons (cons (cons ar1 nil) nil) nil))
```

4) Представление результатов в виде списковых ячеек:



2 Теоретические вопросы

2.1 Элементы языка

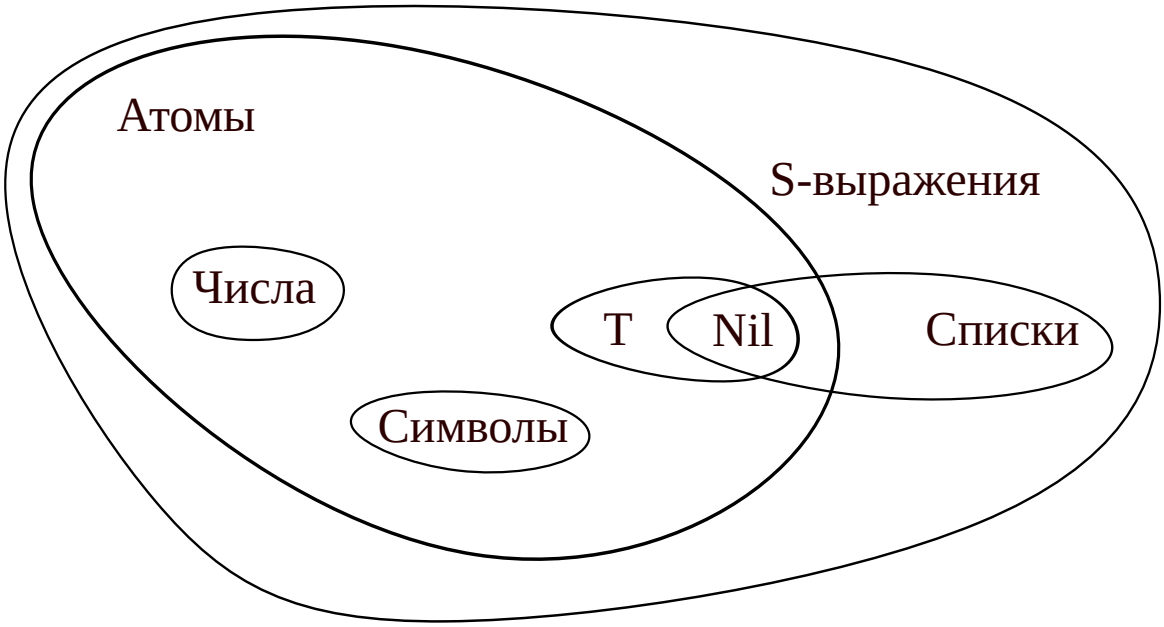


Рисунок 1 – Элементы языка Lisp

Вся информация (данные и программы) в Lisp представляются в виде символьных выражений — S-выражений. По определению

S-выражение ::= <атом> | <точечная пара>.

Элементарные значения структур данных:

Атомы:

— символы (идентификаторы) — синтаксически — набор литер (букв и цифр), начинающихся с буквы;

— специальные символы — {T, Nil} (используются для обозначения логических констант);

— самоопределимые атомы — натуральные числа, дробные числа (например, 2/3), вещественные числа, строки — последовательность символов, заключенных в двойные апострофы (например, “abc”);

Более сложные данные — списки и точечные пары (структуры) строятся из унифицированных структур — блоков памяти — бинарных узлов.

2.1.1 Определения

Точечные пары ::= (<атом>.<атом>) | ((<атом>.<точечная пара>) | (<точечная пара>.<атом>) | (<точечная пара>.<точечная пара>);

Список ::= <пустой список> | <непустой список>, где

<пустой список> ::= () | Nil,

<непустой список> ::= (<первый элемент>.<хвост>),

<первый элемент> ::= <S-выражение>,

<хвост> :: <список>.

2.1.2 Синтаксис

Любая структура (точечная пара или список) заключается в круглые скобки. (A.B) — точечная пара, (A) — список из одного элемента). Пустой список изображается как Nil или (); непустой список по определению может быть изображен: (A.(B.(C.(D())))), допустимо изображение списка последовательностью атомов, разделенных пробелами — (A B C D).

Элементы списка могут, в свою очередь, быть списками (любой список заключается в круглые скобки), например — (A (B C) (D (E))).

Таким образом, синтаксически наличие скобок является признаком структуры — списка или точечной пары.

2.1.3 Представление в памяти

Любая непустая структура Lisp в памяти представляется списковой ячейкой, хранящей два указателя: на голову (первый элемент) и хвост — все остальное.

2.2 Особенности языка Lisp. Структура программы

— В лиспе программист описывает то, что он хочет получить;

— В лиспе функция всегда возвращает какое-то значение;

— Лисп опирается на лямбда-исчисление Черча: все действия, которые можно выполнить, можно организовать в виде функций;

— Малое количество констркуций языка дает много возможностей;

— Поддерживается символьная обработка данных, в связи с чем можно писать программы, изменяющие собственный код в процессе выполнения;

— Лисп — безтиповый язык без операторов с автоматическим управлением памятью;

— Память выделяется блоками, всегда одинаковыми; блок памяти — списковая ячейка ячейка (два указателя).

2.3 Символ апостроф

Символ апостроф — сокращенное обозначение функции quote. Можно считать, что апостроф «блокирует» вычисления.

2.4 Базис языка Lisp. Ядро языка

Базис языка Lisp — минимальный набор конструкций языка и структур данных, с помощью которого можно написать любую программу.

Базис языка Lisp включает атомы, бинарные узлы (для представления структур), базовые функции и функционалы.

ПРИЛОЖЕНИЕ А

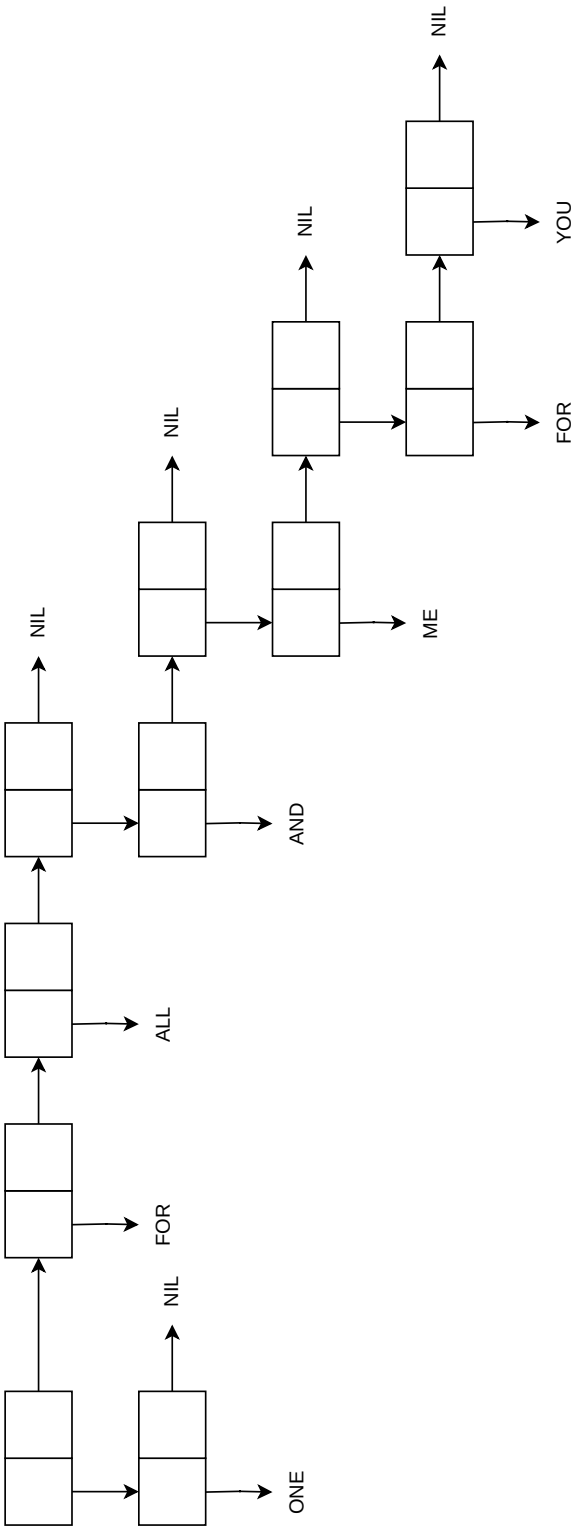


Рисунок 2 – ’((one) for all (and (me (for you))))