

Генерация внутренних представлений

Гладков Артемий Николаевич

06.11.2024



После синтаксического анализа необходимо получить некоторое внутренне представление программы для дальнейшей трансляции в объектный код или интерпретации. Этот код универсален для всех машин, на которых происходит исполнение.



Способы внутреннего представления программы

- Многоадресный код с явно именуемыми результатами.
- Многоадресный код с неявно именуемыми результатами.
- Связанные списочные структуры, представляющие синтаксические деревья
- Префиксная запись
- Постфиксная запись (польская инверсная запись)



Способы внутреннего представления программы

- Многоадресный код с явно именуемыми результатами.
- Многоадресный код с неявно именуемыми результатами.
- Связанные списочные структуры, представляющие синтаксические деревья
- Префиксная запись
- Постфиксная запись (польская инверсная запись), сокращённо **ПОЛИЗ**



В инфиксной записи:

$$a * (b + c) - (d - e) / f$$

в постфиксной:

$$abc + *de - f / -$$

Порядок операндов сохранился, а для операций учтён приоритет, и исчезли скобки.

Выражения, записанные в ПОЛИЗ-е удобно вычислять с помощью стека.



Генерация ПОЛИЗ-а по выражению:

- Если E — операнд, то T — его ПОЛИЗ;
- ПОЛИЗом $E_1\theta E_2$, где θ — бинарная операция, а E_1, E_2 — выражения, является запись $E'_1E'_2\theta$, где E'_1 и E'_2 — ПОЛИЗ выражений E_1 и E_2 ;
- ПОЛИЗом выражения θE , где θ — знак унарной операции, а E — операнд, является запись $E'\theta$, где E' — ПОЛИЗ выражения E ;
- ПОЛИЗом выражения (E) является ПОЛИЗ выражения E .



Выражение прочитывается один раз слева направо и:

- если очередной элемент ПОЛИЗа – это операнд, то его значение заносится в стек;
- если очередной элемент ПОЛИЗа – это операция, то на верхушке стека сейчас находятся ее операнды; они извлекаются из стека, над ними выполняется операция, результат снова заносится в стек;
- когда выражение, записанное в ПОЛИЗе, прочитано, в стеке останется один элемент – это значение всего выражения.



Присваивание

$$I := E$$

будет записано в ПОЛИЗе как

$$\tilde{I}E :=$$

где $:=$ — это двухместная операция, а \tilde{I} и E — операнды. \tilde{I} — означает, что операндом является адрес I , а не её значение.



Оператор перехода означает, что процесс интерпретации надо продолжить с того элемента ПОЛИЗа, который указан как операнд операции перехода.

Чтобы можно было ссылаться на элементы ПОЛИЗа, будем считать, что все они перенумерованы, начиная с 1 (допустим, занесены в массив). Пусть ПОЛИЗ оператора, помеченного меткой L , начинается с номера p , тогда оператор перехода `goto L` в ПОЛИЗе можно записать как

$p!$

где $!$ — операция выбора элемента ПОЛИЗа, номер которого равен p .



Оператор перехода означает, что процесс интерпретации надо продолжить с того элемента ПОЛИЗа, который указан как операнд операции перехода.

Чтобы можно было ссылаться на элементы ПОЛИЗа, будем считать, что все они перенумерованы, начиная с 1 (допустим, занесены в массив).

Пусть ПОЛИЗ оператора, помеченного меткой L , начинается с номера p , тогда оператор перехода `goto L` в ПОЛИЗе можно записать как

$p!$

где $!$ — операция выбора элемента ПОЛИЗа, номер которого равен p .



Оператор перехода означает, что процесс интерпретации надо продолжить с того элемента ПОЛИЗа, который указан как операнд операции перехода.

Чтобы можно было ссылаться на элементы ПОЛИЗа, будем считать, что все они перенумерованы, начиная с 1 (допустим, занесены в массив). Пусть ПОЛИЗ оператора, помеченного меткой L , начинается с номера p , тогда оператор перехода `goto L` в ПОЛИЗе можно записать как

$$p!$$

где $!$ — операция выбора элемента ПОЛИЗа, номер которого равен p .



Условный оператор перехода

Введем вспомогательную операцию – условный переход «по лжи» с семантикой:

if (not B) then goto L

Это двухместная операция с операндами B и L . Обозначим ее $!F$, тогда в ПОЛИЗе она будет записана как:

$B \text{ } p \text{ } !F$

аналогично p – номер элемента, с которого начинается ПОЛИЗ оператора, помеченного меткой L .



а в ПОЛИЗе:

if B then $S1$ else $S2$

if(not B) then goto $L2$; $S1$; goto $L3$; **L2:** $S2$; **L3:** ...,

$$B \ p2 \ !F \ S1 \ p3 \ ! \ S2 \ \dots$$

$\begin{matrix} & & & \uparrow & \uparrow \\ & & & p2 & p3 \end{matrix}$

Тогда **оператор цикла**

while B do S

будет записан, как

if(not B)then goto $L2$; $S1$; goto $L3$; **L2: $S2$; **L3:** ... ,**

а в ПОЛИЗе:

$$\begin{array}{ccccc} B & p1 & !F & S & p0 & !... \\ \uparrow & & & & \uparrow & \\ p0 & & & & p1 & \end{array}$$


И, наконец, оператор ввода **read** (I) в ПОЛИЗе будет записан как $I R$;
оператор вывода **write** (E) – как $E W$.



Генерацию ПОЛИЗа удобно совместить с синтаксическим анализатором.



Рассмотрим грамматику:

$$E \rightarrow T\{+T\}$$

$$T \rightarrow F\{*F\}$$

$$F \rightarrow a \mid b \mid (E)$$

Тогда грамматика с действиями по переводу этого выражения в ПОЛИЗ будет такой:

$$E \rightarrow T\{+T < put(" + ") >\}$$

$$T \rightarrow F\{*F < put(" * ") >\}$$

$$F \rightarrow a < put(" a ") > \mid b < put(" b ") > \mid (E)$$



```
...  
while a < b do begin  
  {if beleberda just commentary  
  write(a);}  
  a := a + 1;  
  c := c * a  
end &  
...
```

Преобразовав, получим:

$$\dots \underset{\uparrow 11}{a} b < 26 ! F \tilde{a} a 1 + := \tilde{c} s a * := 11 ! \dots \underset{\uparrow 26}{}{}$$


- Посмотреть видео с моей реализацией Генерации внутренних представлений: <ссылка>.



Что дальше?

- Посмотреть видео с моей реализацией Генерации внутренних представлений: <ссылка>.
- Доработать текущее решение лабораторной работы №3. Добавить в него генерацию ПОЛИЗа.



- Занятие 5: первая половина пары — исполнение кода.
- Занятие 6 — приём Лабораторных. Или конец материала, если что-то не успеем.

