



Системы программирования.

лекции 11.04.2011

*Семантический анализ. Грамматики с действиями.
Синтаксически управляемый перевод.*





Этапы трансляции

- лексический анализ
- синтаксический анализ
- семантический анализ
- генерация внутреннего представления программы
- оптимизация
- генерация объектной программы



Семантический анализ

Задача :

– Контроль контекстных условий, имеющих в языке программирования

Используемые грамматики: контекстно-свободные с действиями

Какие ошибки в программе могут быть обнаружены на этапе семантического анализа?

Описание синтаксиса модельного языка

$P \rightarrow \text{program } D1; B \perp$

$D1 \rightarrow \text{var } D \{,D\}$

$D \rightarrow I \{,I\}: [\text{int} \mid \text{bool}]$

$B \rightarrow \text{begin } S \{;S\} \text{end}$

$S \rightarrow I := E \mid \text{if } E \text{ then } S \text{ else } S \mid \text{while } E \text{ do } S \mid B \mid \text{read } (I) \mid \text{write } (E)$

$E \rightarrow E1 [= \mid < \mid > \mid \leq \mid \geq \mid \neq] E1 \mid E1$

$E1 \rightarrow T \{ [+ \mid - \mid \text{or}] T \}$

$T \rightarrow F \{ [* \mid / \mid \text{and}] F \}$

$F \rightarrow I \mid N \mid L \mid \text{not } F \mid (E)$

$L \rightarrow \text{true} \mid \text{false}$



Грамматика с действиями

Пример 1.

Описать с помощью грамматики язык

$L = \{\text{любые цепочки из 0 и 1 с неравным количеством 0 и 1}\}$

Написать по грамматике анализатор.



Примеры контекстных условий

1. Контроль повторных описаний

Каждый используемый в программе идентификатор должен быть описан, но не более одного раза в одной зоне описания;

2. Контроль количества параметров при вызове функций

При вызове функции число фактических параметров и их типы должны соответствовать числу и типам формальных параметров;

3. Ограничения, накладываемые на типы операндов

- типы левой и правой частей в операторе присваивания,
- на тип параметра цикла,
- на тип условия в операторах цикла и условном операторе

и т.д.

Обработка описаний

раздел описаний модельного языка

$D1 \rightarrow \text{var } D \{,D\}$

$D \rightarrow I \{,I\}: [\text{int} \mid \text{bool}]$

$D1 \rightarrow \text{var } D \{,D\}$

$D \rightarrow \langle \text{Si.push}(-1); \rangle I \langle \text{Si.push}(c.\text{value}) \rangle$

$\{,I \langle \text{Si.push}(c.\text{value}) \rangle\}:$

$[\text{int} \langle \text{check_types}(\text{"int"}); \rangle \mid \text{bool} \langle \text{check_types}(\text{"bool"}); \rangle]$

Контроль типов операндов в выражении

```
const char* gettype (const char * op, const char * t1, const char* t2);
```

```
gettype("+","int","int") ----> "int"    gettype("+","bool","int")-----> NULL
```

...

```
E1 → T { [ + | - | or ] T }
```

```
T → F { [ * | / | and ] F }
```

```
F → I | N | L | not F | (E)
```

```
E1 → T
```

```
{ [ +<Scc.push("+");> | - <Scc.push("-");> | or<Scc.push("or");> ]  
  T<checkop();> }
```

```
T → F
```

```
{ [ *<Scc.push("*");> | /<Scc.push("/");> | and <Scc.push("and");> ]  
  F<checkop();> }
```

```
F → I<checkid();> |
```

```
    N<Scc.push("int");> |
```

```
    L<Scc.push("bool");> |
```

```
    not F<checknot();> | (E)
```




Контроль типов операндов в операторах

S → I := E | if E then S else S

S → I <checkid();> :=
E <if (strcmp(Scc.pop(), Scc.pop())) throw "Incompatible types" ;>
|
if E <if (strcmp(Scc.pop(), "bool")) throw "Incompatible types" ;>
then S else S

Синтаксически управляемый перевод

Пусть T_1 и T_2 — алфавиты.

Формальный перевод t — это подмножество множества всевозможных пар цепочек в алфавитах T_1 и T_2 : $t \subseteq (T_1^* \times T_2^*)$.

Назовем входным языком перевода t язык

$L_{вх}(t) = \{ \alpha \mid \exists \beta : (\alpha, \beta) \in t \}$.

Назовем целевым (или выходным) языком перевода t язык

$L_{ц}(t) = \{ \beta \mid \exists \alpha : (\alpha, \beta) \in t \}$.

Перевод t неоднозначен, если для некоторых

$\alpha \in T_1^*$, $\beta, \gamma \in T_2^*$, $\beta \neq \gamma$

справедливы соотношения: $(\alpha, \beta) \in t$ и $(\alpha, \gamma) \in t$.

Синтаксически управляемый перевод

Пример 1.

Входной язык $L1 = \{ \text{тексты, переданные азбукой Морзе} \}$

Целевой язык $L2 = \{ \text{тексты на анл. яз.} \}$

Пример 2.

Входной язык $L1 = \{ \text{десятичные числа} \}$

Целевой язык $L2 =$


$\{ \text{текстовое описание десятичных чисел на анл. яз.} \}$

Пример 3.

Входной язык $L1 = \{ \text{любые слова из букв a,b} \}$

Целевой язык $L2 = \{ \text{реверс слов исходного алфавита} \}$

Замечание. Грамматика $L1$ должна быть такой, чтобы по ней можно было написать корректный анализатор



Генерация внутреннего представления программы

Построение внутреннего представления будет сделано с помощью синтаксически управляемого перевода

Входной язык $L1 = \{ \text{программы} \}$

Целевой язык $L2 = \{ \text{внутренние представления программ} \}$

Построение внутреннего представления делается процедурами РС (одновременно с синтаксическим анализом)