

Введение в семантический анализ

четверг, 6 января 2022 г. 14:23

- Лексический анализ
 - Выявляет недопустимые токены (лексемы)
- Синтаксический анализ
 - Выявляет недопустимые комбинации токенов
- Семантический анализ
 - Выявляет все остальные ошибки
 - Несоответствие типов
 - Отсутствие объявлений
 - Дублирование переменных
 - Уточнение соответствия переменных с блоками данных
- Некоторые ошибки не могут быть выявлены парсером
- Некоторые языковые конструкции не являются контекстно-свободными

Проверки

1. Все идентификаторы определены
 2. Соответствие типов
 3. Отношения наследования
 4. Классы объявляются только 1 раз
 5. Методы в пределах класса объявляются по одному разу
 6. Зарезервированные идентификаторы используются корректно
- Зависят от языка
 - Выявление соответствия между объявлениями и использованиями идентификаторов
 - Важный шаг статистического анализа для большинства языков
 - Решение этой задачи связано с понятием области видимости

Область видимости - участок программы, где с конкретным идентификатором связан конкретный объект

Область видимости

- Статическое определение области видимости
 - Область видимости идентификатора определяется в тексте программы
 - Можно понять просто посмотрев на исходник
- Динамическое определение области видимости
 - Область видимости зависит от хода выполнения программы
 - Нельзя точно определить область видимости, глядя на текст программы

Таблицы символов

- Большая часть семантического анализа сводится в обходу AST сверху вниз
 - Сначала: обработать узел n
 - Рекурсия: обработать дочерние элементы n
 - После: завершить обработку узла n
- Для обработки AST на этапе семантического анализа необходимо иметь информацию об идентификаторах
- Таблица символов - структура данных, хранящая информацию об идентификаторах
- Простейшая таблица символов может быть реализована на основе стека
- Операции
 - AddSymbol(x) - добавление символа
 - FindSymbol(x) - поиск символа
 - RemoveSymbol() - удаление символа
- Более сложная таблица символов
 - EnterScope() начало области видимости
 - FindSymbol(x) поиск символа x

- AddSymbol(x) добавление символа x
- CheckScope(x) проверка принадлежности
- ExitScope() окончание области видимости

Типы

- Тип данных
 - Множество значений
 - Множество операций
 - Свойства

Add dx, bx

- Какой тип имеют операнды?
- В языке Си
 - Сложение указателя на функцию с целым числом - недопустимая операция
 - Сложение целого числа с целым числом - допустимая операция
- В ASM (и машинном коде)
 - Обеим операциям соответствует одна и та же команда
- Языки программирования делятся на 3 группы
 - **Статически типизированный**: все или почти все проверки соответствия типов происходят при компиляции программы
 - Быстрая
 - **Динамически типизированные**: все или почти все проверки соответствия типов происходят во время выполнения
 - Стоит дороже во время выполнения
 - Прототипирование
 - **Нетипизированные**: машинные языки
- Строгая типизация
 - Pascal, Java, Delphi
- Нестрогая типизация
 - C
- **Статическая типизация**
 - Выявляет многие ошибки при компиляции
 - Не требует дополнительных затрат на проверки во время выполнения
- **Динамическая типизация**
 - В меньшей степени ограничивает программиста
 - Упрощает быстрое прототипирование
- Большая часть кода написана на языках со статической типизацией, имеющих "лазейки"
 - Пример: небезопасное приведение типов