Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:
Директор
А. В. Замятин

Оценочные материалы по дисциплине

Методы компиляции

по направлению подготовки / специальности

02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) подготовки:
DevOps-инженерия в администрировании инфраструктуры ИТ-разработки
Форма обучения
Очная

Квалификация
Бакалавр

Год приема
2024

СОГЛАСОВАНО:
Руководитель ОП
С.П.Сущенко

Председатель УМК
С.П.Сущенко

Томск – 2024

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности.

ОПК-5 Способен инсталлировать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-2.2 Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий

ИОПК-2.3 Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности

ИОПК-5.3 Выполняет работы по настройке, администрированию и проверке работоспособности программного и аппаратного обеспечения при решении задач профессиональной деятельности

2. Оценочные материалы текущего контроля и критерии оценивания

Элементы текущего контроля:

- письменные контрольные работы по лекционному материалу (4 работы в течении семестра);
- результат выполнения лабораторных работ (разработка синтаксиса учебного языка программирования и этапов реализации транслятора).

Билет на каждую контрольную работу содержит одно теоретические и одно практическое задание.

Пример билета на 1-ю контрольную работу (ОПК-2, ИОПК-2.2).

1.Преобразование недетерминированной автоматной грамматики в детерминированную. Трудоемкость преобразования. Привести пример грамматики и её преобразования.

2.Построить грамматику, конечный автомат и семантические программы для анализа следующих входных данных: Строка символов, состоящая из слов русского языка, в конце – символ точка. Слова разделены пробелами (одним или более). На выходе автомата: 1-е слово, число символов в слове, 2-е слово, число символов в слове, и т.д. В конце вывод – общее количество слов. При ошибке на входе – выдать сообщение.

Пример билета на 2-ю контрольную работу (ОПК-2, ИОПК-2.3).

1. Недетерминированный алгоритм анализа сверху-вниз (магазинный автомат). Его построение по заданной КС-грамматике, функционирование. Доказательство корректности. Пример.

2.Рассмотреть по шагам пример анализа, генерации ОПС для примера: c=a+e;

Пример билета на 3-ю контрольную работу (ОПК-5, ИОПК-5.3).

1.Принципы генерации команд на основе ОПС, реализующих операции индексирования элементов массива. Пример.

2.Пример генерации команд для выражения x:=(a+b)*(c-d). Записать по шагам все действия с магазином при генерации.

Пример билета на 4-ю контрольную работу (ОПК-5, ИОПК-5.2).

1.Грамматика нестрогого предшествования, определение. Преобразование такой грамматики к грамматике простого предшествования. Пример.

2. Пример генерации команд для операторов:

a:=L[1]; b:=2; while a>b do begin b:=b*3; a:=a*2 end.

Записать по шагам все действия с магазином при генерации.

Критерии оценивания контрольной работы:

- «отлично», задания выполнены полностью и без ошибок, приведён правильный пример;
- «хорошо», задания выполнены полностью, но с несущественными ошибками, приведён в целом правильный пример, возможно с несущественными ошибками;
- «удовлетворительно», задания выполнены не полностью, с существенными ошибками, но приведён в целом правильный пример, или задания выполнены с несущественными ошибками, но примеры отсутствуют;
- «неудовлетворительно», оба задания отсутствуют, или выполнено только одно задание или выполнены оба задания но оба с грубейшими ошибками.

Требования к разработке синтаксиса учебного языка программирования и реализации для него транслятор-интерпретатора (ОПК-2, ИОПК-2.2, ИОПК-5.3).

Работа выполняется бригадой из двух или трёх студентов.

Общие требования к языку для бригады из двух студентов:

1) типы данных – целые или вещественные числа и переменные (можно без явных описаний типа), одномерные массивы, причём массивы могут быть статическими;

2) операторы присваивания и формулы со скобками и операциями с двумя приоритетами, (+, –) – низшего, (*, /) – высшего;

3) условные операторы и циклы с условиями, включая операции сравнения;

4) операторы ввода и вывода.

Если в бригаде три человека, то в языке должны быть некоторые расширения, например, 2 типа данных (целые и вещественные), или стандартные математические функции в математических выражениях и др.

Замечание: на минимальную положительную оценку достаточно, чтобы в языке были только простые переменные без явных описаний типа, а из операторов - присваивания и формулы со скобками с операциями +, –, *, /, а также операторы ввода и вывода.

Должны быть разработаны следующие описания к транслятору-интерпретатору:

1) список лексем с номерами лексем, таблица переходов автомата;

2) КС-грамматика языка, в которой лексемы суть терминалы;

3) КС-грамматика языка, преобразованная в нестрогую форму Грейбах;

4) семантические действия для генерации ОПС;

5) список операций ОПС;

6) формат ОПС.

Тесты для проверки функционирования транслятора-интерпретатора:

1) проверка сложных формул с вводом и выводом;

2) тест с действиями: ввод n, ввод n элементов массива, упорядочение массива, вывод массива.

Компоненты транслятора-интерпретатора:

1) лексический анализатор в виде функции;

2) синтаксический анализатор – генератор ОПС;

3) интерпретатор ОПС.

Критерии оценивания реализации транслятора:

- «отлично», описания транслятора выполнены полностью и без существенных ошибок, транслятор реализует все необходимые конструкции языка и правильно исполняет все тесты;

- «хорошо», описания транслятора выполнены почти полностью и без существенных ошибок, транслятор реализует все необходимые конструкции языка, возможно с замечаниями, правильно исполняет почти все тесты;

- «удовлетворительно», описания транслятора соответствуют языку программирования, но в языке реализованы не все конструкции, указанные в задании, сами описания выполнены почти полностью и без существенных ошибок, транслятор реализует некоторые конструкции языка, возможно с замечаниями, и правильно исполняет соответствующие тесты;

- «неудовлетворительно», описания и реализация не соответствует требованиям оценки «удовлетворительно».

Транслятор можно реализовать на любом удобном языке программирования, например, на Си.

Так как реализация транслятора осуществляется бригадой, то в описании транслятора и его реализации в программном коде должны разбираться все члены бригады. Если студент, входящий в бригаду, разбирается с трудом и с ошибками, то его личная оценка снижается по сравнению с общей оценкой реализации, а если совсем не разбирается, то его личная оценка = «неудовлетворительно».

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Для получения положительной оценки по зачёту требуется выполнить каждую контрольную работу и реализацию транслятора не хуже, чем на «удовлетворительно». Общая оценки по зачёту вычисляется путём усреднения всех этих оценок, причём вес оценки за реализацию транслятора удваивается.

Примерный перечень теоретических заданий на контрольных работах.

1.Построение недетерминированного конечного автомата по недетерминированной автоматной грамматике и его функционирование. Трудоемкость работы автомата. Привести пример грамматики, автомата и его работы!

2.Преобразование недетерминированной автоматной грамматики в детерминированную. Трудоемкость преобразования. Привести пример грамматики и её преобразования!

3.Преобразование праволинейной грамматики в автоматную. Трудоемкость преобразования. Привести пример!

4.Преобразование автоматной грамматики с удалением бесполезных нетерминалов. Трудоемкость преобразования. Привести пример!

5.Преобразование регулярного выражения в праволинейную грамматику. Трудоемкость преобразования. Привести пример!

6. Недетерминированный алгоритм анализа сверху-вниз (магазинный автомат). Его построение по заданной КС-грамматике, функционирование. Доказательство корректности. Пример!

7. Порождающие правила для грамматики, включающей присваивания, выражения, переменные с индексами, последовательность операторов присваивания. Генерация ОПС для всех этих конструкций языка при работе анализатора LL(1). Пример!

8.Порождающие правила для грамматики, включающей вложенные условные операторы IF-THEN-ELSE, выражения с операциями сравнения. Генерация ОПС для всех этих конструкций языка при работе анализатора LL(1). Пример!

9.Порождающие правила для грамматики, включающей описания переменных и массивов, операторы ввода-вывода и операторы выделения памяти для массивов. Генерация ОПС для всех этих конструкций языка при работе анализатора LL(1). Пример!

10.Архитектура одноадресных команд. Принципы генерации команд на основе ОПС. Правила генерации команд, реализующих арифметические операции и присваивание. Пример!

11.Принципы генерации команд на основе ОПС, реализующих операции индексирования элементов массива. Пример!

12. Правила генерации команд на основе ОПС, реализующих операции сравнения. Пример!

13.Общий алгоритм вычисления адресов для команд перехода при генерации команд на основе ОПС. Пример!

14.Принципы генерации команд на основе ОПС, реализующих операции условного и безусловного перехода. Пример!

15.Недетерминированный алгоритм анализа снизу-вверх (магазинный автомат). Его построение по заданной произвольной КС-грамматике, функционирование. Доказательство корректности. Трудоемкость работы алгоритма. Пример.

16.Отношения простого предшествования, определение, вычисление из элементарных отношений first и last. Построение анализатора и его функционирование. Условия однозначности его работы. Трудоемкость работы анализатора. Пример.

17.Отношения операторного предшествования, определение, вычисление из элементарных отношений first, last, firstterm и lastterm. Построение анализатора и его функционирование. Условия однозначности его работы. Трудоемкость работы анализатора. Пример.

18.Грамматика нестрогого предшествования, определение. Преобразование такой грамматики к грамматике простого предшествования. Пример.

19.Построение анализатора методом рекурсивного спуска, преобразование грамматики для этого. Условия однозначности его работы. Трудоемкость работы анализатора. Пример.

20.LR(k)-анализатор. Принципы построения LR(1)-анализатора, его функционирование. Условия однозначности его работы. Трудоемкость работы анализатора. Пример.

Примерный перечень практических заданий на контрольных работах.

1.Построить грамматику, конечный автомат и семантические программы для анализа следующих входных данных: Строка символов, состоящая из слов русского языка, в конце – символ точка. Слова разделены пробелами (одним или более). На выходе автомата: 1-е слово, число символов в слове, 2-е слово, число символов в слове, и т.д. В конце вывод – общее количество слов. При ошибке на входе – выдать сообщение.

2.Построить грамматику, конечный автомат и семантические программы для анализа следующих входных данных: Строка символов, состоящая из целых десятичных чисел со знаком (+ или – обязательно!), в конце – символ ‘#’. Числа разделены пробелами (одним или более). На выходе автомата: 1-е число, число цифр в нем, 2-е число, число цифр в нем, и т.д. В конце вывод – общее количество чисел. При ошибке на входе – выдать сообщение.

3.Построить грамматику, конечный автомат и семантические программы для анализа следующих входных данных: Строка символов, состоящая из слов английского языка, в конце – символ точка. Слова разделены запятыми (после запятой – один или более пробелов) или пробелами (одним или более). На выходе автомата: 1-е слово, число символов в слове, 2-е слово, число символов в слове, и т.д. В конце вывод – общее количество слов. При ошибке на входе – выдать сообщение.

4.Рассмотреть по шагам пример анализа, генерации ОПС для примера: c=a+e;

5.Рассмотреть по шагам вычисление ОПС для примера: b1=1; a[i+1]=b1–5.4*c[i];

6.Рассмотреть по шагам пример анализа, генерации ОПС для примера:
begin c=a+e end

7.Рассмотреть по шагам вычисление ОПС для примера: begin b=1; a=(b–5)*c end

8.Рассмотреть по шагам пример анализа, генерации ОПС для примера: {c=a*5+e;}

9.Рассмотреть по шагам вычисление ОПС для примера: {b=1; a[i+1]=(b–4)*c[i];}

10.Рассмотреть по шагам пример анализа, генерации ОПС для примера:
begin c=a*7 end

11.Рассмотреть по шагам вычисление ОПС для примера:
begin b=9; a=(b+5)/(c+4) end

12.Пример генерации команд для операторов: a:=10; b:=M[a]; if a>b then b:=1 else b:=2. Записать по шагам все действия.

13. Пример генерации команд для выражения x:=(b*c-d)*a. Записать по шагам все действия с магазином при генерации.

14.Пример генерации команд для выражения x:=(a+b)*(c-d). Записать по шагам все действия с магазином при генерации.

15. Пример генерации команд для выражения x[i+1]:=y[i,j-a]*2. Записать по шагам все действия с магазином при генерации.

16. Пример генерации команд для операторов: a:=L[1]; b:=2; while a>b do begin b:=b*3; a:=a*2 end. Записать по шагам все действия с магазином при генерации.

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

Практические задания (ОПК-2, ИОПК-2.2).

1. Построить конечный автомат для анализа следующих входных данных: Строка символов, состоящая из слов русского языка, в конце – символ точка. Слова разделены пробелами. Привести пример работы автомата по шагам при анализе строки символов:

Это всё.

2.Рассмотреть по шагам пример анализа сверху-вниз по грамматике простых арифметических выражений примера: a+e. Привести эту грамматику и действия при анализе.

3. Преобразовать выражение a[i+1]=(b–4)*c[i] в ОПС.

4. Преобразовать выражение (b–4)*c в ОПС и вычислить его, если b=3, c=2.

5. Преобразовать выражение (b–4)*c в ОПС и записать одноадресные команды для его вычисления.

Теоретические вопросы (ОПК-5, ИОПК-5.3).

1. Задачи и этапы трансляции. Типы трансляторов.
2. Формальные грамматики и языки.
3. Порождающие правила.
4. Классификация грамматик и языков по Хомскому.
5. Работа конечного автомата для А-грамматики.
6. Деревья грамматического анализа для КС-грамматики.
7. Нестрогая нормальная форма Грейбах для КС-грамматики.
8. Принципы генерации ОПС при LL-анализе.
9. Принципы генерации ОПС при LR-анализе.
10. Грамматический анализ для грамматики простого предшествования.
11. Грамматический анализ для грамматики операторного предшествования.
12. Вычисление ОПС в интерпретаторе.
13. Принципы распределения памяти при вычислении ОПС в интерпретаторе.
14. Принципы генерации одноадресных команд по ОПС.
15. Принципы распределения памяти при вычислении сгенерированной программы.

Информация о разработчиках

Костюк Юрий Леонидович, д.т.н., профессор кафедры теоретических основ информатики.