**Программа учебной дисциплины «Верификация программного обеспечения»**

Утверждена

Академическим советом ООП

Протокол № от «\_\_»\_\_\_\_\_20\_\_ г.

|  |  |
| --- | --- |
| Автор | Камкин Александр Сергеевич |
| Число кредитов |  |
| Контактная работа (час.) | 64 |
| Самостоятельная работа (час.) | 126 |
| Курс | 1 курс МП «Системное программирование» |
| Формат изучения дисциплины | Без использования онлайн-курса |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

Целями освоения дисциплины «Верификация программного обеспечения» являются получение знаний о базовых принципах и методах верификации программного обеспечения; приобретение навыков, необходимых для практического применения методов верификации программного обеспечения.

В результате освоения дисциплины студент должен:

**знать:**

- базовые принципы формальной верификации программного обеспечения;

- методы формализации семантики языков программирования;

- методы дедуктивной верификации программ;

- методы проверки моделей компьютерных систем.

**уметь:**

- методы проверки моделей компьютерных систем.

- описывать условия корректности программ в форме пред- и постусловий;

- аналитически доказывать частичную и полную корректность программ;

- строить формальные модели компьютерных систем;

- описывать свойства реагирующих систем в виде формул темпоральной логики.

**владеть:**

- навыками спецификации и моделирования программного обеспечения;

- навыками аналитической верификации программ;

- инструментальными средствами дедуктивной верификации программ;

- инструментальными средствами проверки моделей компьютерных систем.

Изучение дисциплины «Верификация программного обеспечения» базируется на следующих дисциплинах:

- дискретная математика;

- программирование;

- информатика, математическая логика и теория алгоритмов.

Для освоения учебной дисциплины студенты должны владеть следующими знаниями и компетенциями:

- обладать навыками программирования;

- обладать навыками работы с ОС Linux;

- знать язык программирования C;

- знать основы теории графов;

- знать основы математической логики.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- параллельное программирование;

- компьютерные сети.

1. **СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Тема 1. Общие принципы формальной верификации**

Общая схема формальной верификации. Примеры методов формальной верификации: дедуктивная верификация, проверка моделей, проверка эквивалентности. Программные контракты (пред- и постусловия). Частичная и полная корректность программ.

**Тема 2. Формализация семантики языков программирования**

Язык программирования while. Операционная семантика языка while. Аксиоматическая семантика языка while. Слабейшее предусловие и сильнейшее постусловие.

**Тема 3. Методы дедуктивной верификации программ**

Метод индуктивных утверждений. Метод фундированных множеств. Автоматизация дедуктивного анализа программ: синтез инвариантов циклов, доказательство условий верификации.

**Тема 4. Инструментальные средства дедуктивной верификации программ**

Язык аннотации C-программ ACSL (ANSI C Specification Language). Платформа статического анализа C-программ Frama-C. Плагин дедуктивной верификации C-программ Jessie. Платформа дедуктивной верификации Why. SMT-решатели.

**Тема 5. Параллельные программы и реагирующие системы**

Параллельные программы над общими переменными. Синхронный и асинхронный параллелизм. Семантика чередований. Реагирующие системы. Справедливость планировщика.

**Тема 6. Темпоральная логика линейного времени**

Синтаксис и семантика темпоральной логики линейного времени (LTL). Основные тождества. Выражение свойств реактивных систем в логике LTL. Свойства безопасности и живости.

**Тема 7. Теоретико-автоматный метод проверки моделей**

Моделирование реактивных систем структурами Крипке. Автоматы Бюхи. Построение автомата Бюхи для формулы LTL. Построение синхронной композиции автоматов Бюхи. Проверка пустоты языка, допускаемого автоматом Бюхи.

**Тема 8. Символический метод проверки моделей**

Символическое представление множеств и отношений. Символические алгоритмы. Двоичные решающие диаграммы (BDD) и операции над ними. Ограниченная проверка моделей.

**Тема 9. Инструментальные средства проверки моделей**

Язык моделирования компьютерных систем и протоколов Promela (Process/Protocol Meta Language). Инструмент проверки моделей Spin. Инструмент символической проверки моделей SMV.

**Тема 10. Использование формальных методов в тестировании**

Тестирование программ (методы черного и белого ящика). Тестирование на основе моделей. Использование техник символического исполнения для генерации тестов.

1. **ОЦЕНИВАНИЕ**

В рамках курса студентам предлагается выполнить одну контрольную работу (оценка: ) и два домашних задания (оценки: и ). Текущая оценка () рассчитывается как среднее арифметическое оценок , и :

Оценка за самостоятельную работу () отражает правильность и полноту выполнения домашних работ по темам практических занятий.

Накопленная оценка рассчитывается следующим образом:

В диплом выставляет результирующая оценка по учебной дисциплине, которая формируется по следующей формуле:

где — оценка за устный экзамен.

По всем видам работ выставляются 10-ти балльные оценки.

Округление при вычислении результирующей оценки осуществляется в пользу студента.

1. **ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ**

# **Оценочные средства для текущего контроля студента**

Примерный вариант контрольной работы

1. Найдите инвариант следующего цикла:

{a ≥ 0}  
x := a;  
n := 1;  
y := 0;  
**while** x ≠ 0 **do** y := y + n;  
 n := n + 2;  
 x := x – 1  
**end**{y = a2}

1. Докажите частичную корректность следующей программы:

{*a* ≥ 0}  
*x* := 0; *y* := 1;  
**while** *y* ≤ *a* **do** *x* := *x* + 1;  
 *y* := *y* + 2\**x* + 1  
**end**{0 ≤ *x*2 ≤ *a <* (*x* + 1)2}

1. Представьте в виде формулы LTL следующее высказывание: «между каждой парой состояний, удовлетворяющих свойству *p*, обязательно встретится состояние, удовлетворяющее свойству *q*».
2. Постройте автомат Бюхи для следующей формулы LTL: **GF**(**¬***p* ∧ **X***q*).

Домашнее задание 1 состоит в дедуктивной верификации небольшой C-программы с помощью инструментов Frama-C (Jessie) и Why.

Домашнее задание 2 состоит в разработке Promela-модели некоторого протокола и ее проверки с помощью инструмента Spin.

**Оценочные средства для промежуточной аттестации**

Вопросы к устной части экзамена

1. Формальная верификация программ. Общая схема и основные подходы. Частичная и полная корректность программ.
2. Формальная семантика языков программирования. Операционная семантика языка программирования. Структурная операционная семантика языка while.
3. Формальная семантика языков программирования. Аксиоматическая семантика языка программирования. Аксиоматическая семантика языка while.
4. Аксиоматическая система Хоара. Инвариант цикла. Слабейшее предусловие программы и его свойства. Сильнейшее постусловие.
5. Метод индуктивных утверждений Флойда. Точки сечения. Индуктивные утверждения. Условия верификации. Схема доказательства частичной корректности блок-схемы.
6. Метод фундированных множеств Флойда. Точки сечения. Индуктивные утверждения. Условия завершимости. Схема доказательства завершимости блок-схемы.
7. Автоматизация дедуктивной верификации программ. Синтез инвариантов циклов. Эвристические методы. Методы на основе абстрактной интерпретации.
8. Автоматизация дедуктивной верификации программ. Доказательство условий верификации. Понятие разрешающей процедуры. SAT- и SMT-решатели.
9. Параллельные программы. Семантика асинхронных чередований. Справедливость планировщика. Построение асинхронной композиции систем переходов.
10. Темпоральная логика линейного времени (LTL). Синтаксис и семантика LTL. Выражение свойств реагирующих систем в логике LTL. Свойства безопасности и живучести.
11. Структуры Крипке. Вычисления и траектории структуры Крипке. Выбор уровня абстракции при моделировании компьютерных систем. Ложные сообщения и пропуски ошибок.
12. Описание ω-языков автоматами Бюхи. Проверка языка на пустоту. Синхронная композиция автоматов Бюхи. Построение автомата Бюхи по расширенному автомату Бюхи.
13. Теоретико-автоматный подход к проверке модели для LTL. Общая схема метода. Построение автомата Бюхи для LTL-формулы.
14. Использование формальных методов в тестировании. Тестирование на основе моделей. Использование техник символического исполнения для генерации тестов.
15. **РЕСУРСЫ**
    1. **Основная литература**
16. А.С. Камкин. Введение в формальные методы верификации программ. М.: МАКС Пресс, 2018.
17. В.А. Непомнящий, О.М. Рякин. Прикладные методы верификации программ. М.: Радио и связь, 1988.
18. Ю.Г. Карпов. Model Checking. Верификация параллельных и распределенных программных систем. БХВ-Петербург, 2010.
19. Э.М. Кларк, О. Грамберг, Д. Пелед. Верификация моделей программ. Model Checking. М.: МЦНМО, 2002.
    1. **Дополнительная литература**
20. K.R. Apt, F.S. de Boer, E.-R. Olderog. Verification of Sequential and Concurrent Programs, Springer, 2009.
21. C. Baier, J.-P. Katoen. Principles of Model Checking. The MIT Press, 2008.
22. P. Baudin, P. Cuoq, J.-C. Filliâtre, C. Marché, B. Monate, Y. Moy, V. Prevosto. ACSL: ANSI/ISO C Specification Language. Version 1.4. 2010.
23. G.J. Holzmann. The SPIN Model Checker. Primer and Reference Manual. Addison-Wesley, 2003.
24. Э. Дейкстра. Дисциплина программирования. М.: Мир, 1978.
25. Д. Грис. Наука программирования. М.: Мир, 1984.
26. Р. Андерсон. Доказательство правильности программ. М.: Мир, 1982.
    1. **Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1. | Frama-C | *Свободно распространяемое ПО* |
| 2. | Why3 | *Свободно распространяемое ПО* |
| 3. | Spin | *Свободно распространяемое ПО* |

* 1. **Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);

мультимедийный проектор с дистанционным управлением.