МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Ярославский государственный университет им. П.Г. Демидова
Факультет информатики и вычислительной техники

УТВЕРЖДАЮ

Проректор по развитию образования

_____Е.В. Сапир

"_____"_____2012 г.

Рабочая программа дисциплины
послевузовского профессионального образования
(аспирантура)

Верификация программного обеспечения

по специальности научных работников

05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Ярославль 2012

1. Цели освоения дисциплины

Целями освоения дисциплины «Верификация программного обеспечения» в соответствии с общими целями основной профессиональной образовательной программы послевузовского профессионального образования (аспирантура) (далее ОПП– образовательная программа послевузовского профессионального образования) являются:

изучение общих основ моделирования программ, способов спецификации свойств программ, методов и приемов исследования свойств программ, анализа и доказательства корректности программ и их моделей.

2. Место дисциплины в структуре ООП послевузовского профессионального образования (аспирантура)

Данная дисциплина относится к разделу обязательные дисциплины (подраздел дисциплины по выбору аспиранта) образовательной составляющей образовательной программы послевузовского профессионального образования по специальности научных работников 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.

Дисциплина «Верификация программного обеспечения» необходима для решения задач построения корректных программ, анализа корректности и исследования различных свойств программ и их моделей.

Для изучения данной дисциплины необходимы «входные» знания, умения, полученные в процессе обучения по программам специалитета или бакалавриата-магистратуры. Дисциплина «Верификация программного обеспечения» опирается на дисциплины «Математическая логика», «Дискретная математика», «Языки программирования и методы трансляции», «Основы информатики». От аспиранта требуется наличие логического мышления, образованность, организованность и трудолюбие, самостоятельность, настойчивость в достижении цели, а также знания, полученные при изучении указанных выше дисциплин.

3. Требования к результатам освоения содержания дисциплины

В результате освоения дисциплины обучающийся должен:

Знать:

1) способы моделирования программ;
2) способы спецификации и анализа свойств программ;
3) способы дедуктивного доказательства корректности программ;
4) стратегию спецификации и доказательства корректности программ, написанных на процедурном языке высокого уровня;
5) методы автоматической проверки корректности программной модели.

Уметь:

1) проводить спецификацию программ на языке предикатов;
2) применять метод доказательства теорем для доказательства корректности программ, написанных на языках высокого уровня;
3) строить программные модели и проводить спецификацию и верификацию программных свойств на языке темпоральной логики;
4) строить модели систем реального времени с помощью формализма временных автоматов и проводить спецификацию свойств таких систем на языках временных темпоральных логик.

Владеть:

1) формальными методами моделирования и спецификации программ;
2) формальными методами анализа корректности программ (алгоритмов): дедуктивным анализом (метод доказательства теорем) и методом проверки модели (model checking).

4. Структура и содержание дисциплины «Верификация программного обеспечения»
Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 часов.

№ п/п	Раздел Дисциплины	Курс	Неделя	Виды учебной работы, включая самостоятельную работу обучающихся, и трудоемкость (в часах) Форма обуч.: очная/заочная					Формы текущего контроля успеваемости *(по неделям)* Форма промежуточной аттестации
				Лекций	Лабораторных работы	Практических	Сам. работа	Контроль сам.	
1	**Теория семантики и верификации программ** Дедуктивный анализ корректности программ на примере «простого» языка программирования. Спецификация программ с помощью пред- и постусловий. Доказательство корректности программ относительно спецификации, инвариантов и ограничивающей функции. Построение инвариантов и ограничивающих функций.	1	1-3	2/1			34/ 36		Индивидуальное задание № 1 (срок 3 неделя)
2	**Модели вычислительных процессов. Верификация моделей и теория автоматов.** Построение моделей параллельных и распределенных систем .Асинхронные и синхронные процессы. Взаимодействие процессов. Структура Крипке. Метод проверки модели. Верификация моделей и теория автоматов. Автоматы над бесконечными словами. Структура Крипке как автомат Бюхи. Темпоральная логика линейного времени LTL. Формула LTL как обобщенный автомат Бюхи. Редукция автомата Бюхи для формулы LTL. Пересечение языков структуры Крипке и автомата Бюхи. Проверка пустоты автомата Бюхи. Проверка модели «на лету».	1	4-6	1/1			17/ 18		Индивидуальное задание № 2 SPIN (срок 6 неделя)

3	**<u>Верификация моделей для логики CTL. Двоичные диаграммы решений.</u>** Верификация моделей для логики CTL. Темпоральная логика CTL. Верификация моделей для CTL. Верификация моделей и неподвижные точки. Символьная верификация моделей для CTL. Двоичные диаграммы решений. Диаграммы ROBDD. Построение и манипуляция ROBDD.	1	7-9	1/1			17/ 18		Индивидуальное задание № 3 SMV (срок 9 неделя)
4	**<u>Теория временных автоматов.</u>** Временные автоматы Бюхи и Мюллера. Моделирование, спецификация и верификация систем реального времени с помощью временных автоматов	1	10-12	2/1			34/ 34		Индивидуальное задание № 4 Uppaal (срок 12 неделя)
	Всего	**1**	**12**	**6/4**			**102/ 104**		**зачет**

5. Образовательные технологии

В основу образовательной технологии по дисциплине «Верификация программного обеспечения» помимо традиционных форм занятий в виде лекций положена также форма, состоящая в выполнении обучающимся индивидуальных заданий по темам дисциплины. Имеются четыре индивидуальных задания по дисциплине. Задания должны быть решены письменно или в электронном виде с последующей устной защитой. Первое задание закрывает тематику дедуктивного анализа корректности последовательных программ. Представляет собой небольшую программу, написанную на «простом» языке высокого уровня, корректность которой необходимо доказать относительно спецификации, составляемой аспирантом по словесному описанию требований к программе. При безуспешных со стороны аспиранта попытках построения инварианта, ограничивающей функции и постусловия эти необходимые для выполнения задания компоненты могут быть предоставлены аспиранту. Выполнение второго, третьего и четвертого заданий предполагает моделирование (представление в виде структуры Крипке), спецификацию (запись свойств на языке темпоральной логики) и автоматическую верификацию некоторой параллельной и распределенной программы с помощью свободно распространяемых для учебных целей средств верификации SPIN, SMV и Uppaal. Ошибки, допущенные при выполнении задания, отмечаются подробно преподавателем, ведущим дисциплину. После исправления ошибок задание сдается вновь преподавателю на проверку. Аспиранты, сдавшие все индивидуальные задания в установленные сроки, после успешного ответа на ряд дополнительных вопросов, закрывающих оставшиеся темы, получают отметку о сдаче зачета досрочно. Такой подход стимулирует постоянную работу аспиранта и активизирует усвоение материала. Эта технология позволяет проводить индивидуальное обучение аспирантов и дает хорошие результаты. Она дополняется обсуждением общих (типичных) ошибок на лекционных занятиях.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы обучающихся

Текущий контроль успеваемости аспирантов организован в виде четырех индивидуальных заданий, которые должен выполнить каждый обучающийся. В предыдущем разделе описана технология индивидуального обучения аспирантов при помощи таких заданий. Промежуточная аттестация по итогам освоения дисциплины проводится в виде зачета.

Примеры **индивидуальных заданий**:

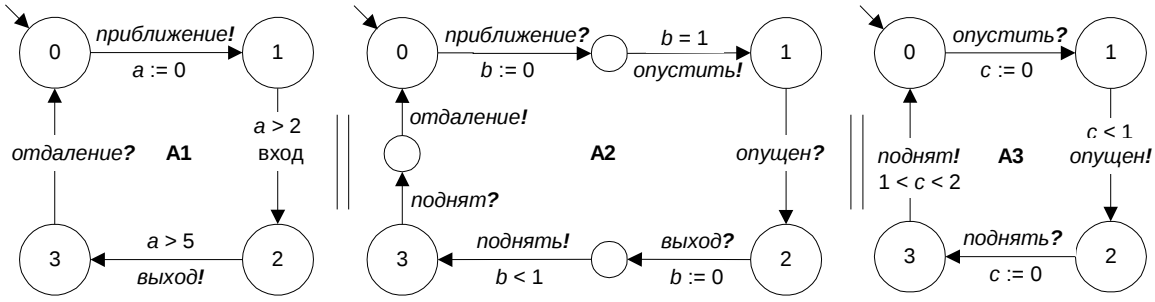
I. Докажите формально, что следующий алгоритм предназначен для записи в переменную *z* произведения чисел *a* и *b* при $b \geq 0$ без использования операции умножения.

x, *y*, *z* := *a*, *b*, 0;
do *y* > 0 & even(*y*) -> *y*, *x* := *y*/2, *x*+*x* []
odd(*y*) -> *y*, *z* := *y*-1, *z*+*x*
od

II. – III. Используя определения синтаксиса и семантики формул темпоральной логики LTL (или CTL) произведите спецификацию с последующей верификацией с помощью средства верификации SPIN и SMV указанных ниже свойств для системы асинхронных параллельных процессов со взаимным исключением, представленной на рис. 1.1. в пособии [4]:

1. «Взаимное исключение». Процессы никогда не окажутся в своих критических участках одновременно. Другими словами, система ни при каких обстоятельствах не попадет в состояние, в котором процессы Пр1 и Пр2 будут находиться в своих локальных состояниях с номерами 7.
2. «Отсутствие взаимной блокировки». Не существует ситуации, при которой ни процесс Пр1, ни процесс Пр2 не могут перейти в другое локальное состояние.
3. «Отсутствие бесконечного откладывания процессов». Если один из процессов пожелает войти в свою критическую секцию, он обязательно в нее войдет. Другими словами, исключается возможность, при которой один из процессов бесконечно часто заходит в свой критический участок, а второй процесс вынужден постоянно откладывать свой вход в этот участок, бесконечно долго, таким образом, ожидая своей очереди.
4. «Справедливость». Если оба процесса одновременно вышли из своих некритических участков, т. е. процессы Пр1 и Пр2 находятся в локальных состояниях с номерами 1, то в свой критический участок первым обязательно войдет тот процесс, приоритет которого в данным момент выше (приоритет определяется исходя из значения переменной *trn*).
5. «Отсутствие чередования». Если первый процесс только что посетил свой критический участок и хочет вновь в него войти, а второй процесс не выражает такого желания, то первому нет необходимости дожидаться, пока второй процесс проявит себя, войдет в критическую область, а затем покинет ее. Другими словами, посещения процессами своих критических участков не обязательно должны чередоваться. Если один из процессов навсегда остается в своем некритическом участке (например, закончил работу), то это не оказывает никакого влияния на возможность входа другого процесса в свой критический участок.

IV. Проведите проверку свойств модели (реального времени) железнодорожного переезда с помощью программного средства верификации Uppaal, базирующемся на теории временных автоматов.



Модель автоматной системы в виде параллельной композиции синхронизирующихся временны*х* автоматов

Свойства:
1) шлагбаум не может быть опущен (переезд не может быть закрыт) более чем на 10 мин., т.е. состояние автомата А3 «0. шлагбаум поднят» достигается как максимум через 10 мин. после попадания им в состояние «2. Шлагбаум опущен».
2) когда поезд войдёт в переезд, шлагбаум уже должен быть опущен.

Кроме того, при сдаче задания могут быть заданы вопросы по теории. Материал по теме задания должен быть подробно и полно освещен и проиллюстрирован на примере решения этого задания.

Вопросы к зачету.

Теория семантики и верификации программ

1. Корректность программ. Спецификация и верификация. Верификация и тестирование.
2. Спецификация программ. Предусловие. Постусловие. Примеры спецификаций программ. Представление начальных и конечных значений переменных. Наброски доказательств.
3. Семантика простого языка программирования. Преобразователь предикатов wp. Спецификация программ через преобразователь предикатов wp. Свойства wp.
4. Семантика простого языка программирования. Команды skip, abort и композиция команд. Команда присваивания. Кратное присваивание. Присваивание элементу массива.
5. Семантика простого языка программирования. Команда выбора. Примеры. Теорема о команде выбора. Доказательство корректности программ, не содержащих команд повторения. Примеры доказательств.
6. Семантика простого языка программирования. Команда повторения. Инвариант. Ограничивающая функция. Теорема о цикле, инварианте и ограничивающей функции. Доказательство программ, содержащих циклы. Список условий для проверки цикла. Примеры доказательств корректности цикла.
7. Построение программ. Стратегия построения команд выбора.
8. Построение программ. Построение циклов исходя из инвариантов и ограничений.
9. Построение инвариантов цикла. Теория воздушного шарика. Основная идея и стратегии построения инвариантов.
10. Построение инвариантов цикла методом устранения конъюнктивного члена. Примеры.
11. Построение инвариантов цикла методом замены константы переменной. Примеры.
12. Построение инвариантов цикла методом расширения области значений переменной. Примеры.
13. Построение инвариантов цикла методом комбинирования пред- и постусловий. Примеры.

Модели вычислительных процессов. Верификация моделей и теория автоматов

1. Модели вычислительных процессов: сети Петри, взаимодействующие процессы и т.д.
2. Взаимодействие процессов. Асинхронные и синхронные процессы. Синхронизация параллельных процессов. Проблема критических участков. Анализ подходов к решению проблемы. Алгоритм Деккера. Программная реализация взаимоисключений.
3. Структура Крипке и метод автоматической верификации моделей.
4. Автоматы над бесконечными словами.
5. Структура Крипке как автомат Бюхи.
6. Темпоральная логика линейного времени LTL.
7. Формула LTL как обобщенный автомат Бюхи. Замыкание формулы. Правила разметки последовательностей. Построение обобщенного автомата Бюхи по формуле LTL.

8. Редукция обобщенного автомата Бюхи для формулы логики LTL. Исключение избыточных переходов. Построение автомата исходя из необходимости. Определение эквивалентных состояний.
9. Пересечение языков структуры Крипке и автомата Бюхи.
10. Проверка пустоты автомата Бюхи.
11. Проверка модели «на лету»

Верификация моделей для логики CTL. Двоичные диаграммы решений

1. Темпоральная логика CTL.
2. Верификация моделей для CTL.
3. Верификация моделей и неподвижные точки.
4. Символьная верификация моделей для CTL.
5. Двоичные диаграммы решений ROBDD.
6. Построение и манипуляция ROBDD. Процедура Mk.
7. Построение и манипуляция ROBDD. Процедура Build.
8. Построение и манипуляция ROBDD. Процедура Apply.
9. Построение и манипуляция ROBDD. Процедура Restrict.
10. Построение и манипуляция ROBDD. Кванторы существования и всеобщности.

Теория временных автоматов

1. Детерминированные и недетерминированные временные автоматы Бюхи и Мюллера.
2. Разрешимые свойства временных автоматов.
3. Верификация систем реального времени с помощью временных автоматов.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Кузьмин Е.В. Верификация моделей программ. – Учебное пособие, Ярославль, ЯрГУ, 2008. – 176 с.
2. Кузьмин Е.В. Введение в теорию вычислительных процессов и структур. – Учебное пособие, Ярославль, ЯрГУ, 2006. – 140 с.

б) дополнительная литература:

3. Грис Д. Наука программирования. – М.: Мир, 1984. – 416 с.
4. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем. – СПб.: БХВ-Петербург, 2010. – 560 с.
5. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ: Model Checking. Пер. с англ. – М.: МЦНМО, 2002. – 416 с.
6. Карпов Ю.Г. Теория автоматов. – СПб.: Питер, 2003. – 208 с.
7. Математическая логика в программировании: сб. статей 1980—1988 гг.: Пер. с англ. – М.: Мир, 1991. – 408 с.
8. Минский М. Вычисления и автоматы – М.: Мир, 1971. – 268 с.
9. Непомнящий В.А., Рякин М.О. Прикладные методы верификации программ – М.: Радио и связь, 1988. – 256 с.
10. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 263 с.
11. Хоар Ч. Взаимодействующие последовательные процессы. – М.: Мир, 1989. – 264 с.
12. Хопкрофт Д., Мотвани Р., Ульман Д. Введение в теорию автоматов, языков и вычислений. – М.: Вильямс, 2002. – 528 с.

в) программное обеспечение и Интернет-ресурсы:

13. SMV. Symbolic Model Verifier. Carnegie Mellon University. http://www.cs.cmu.edu/~modelcheck/smv.html
14. SPIN. http://spinroot.com/spin/whatispin.html
15. UPPAAL. http://www.uppaal.com

8. Материально-техническое обеспечение дисциплины

Компьютер, мультимедийный проектор,
набор электронных презентаций и схем.

Программа составлена в соответствии с федеральными государственными требованиями к структуре основной профессиональной образовательной программы послевузовского профессионального образования (аспирантура) (приказ Минобрнауки от 16.03.2011 г. № 1365) с учетом рекомендаций, изложенных в письме Минобрнауки от 22.06.2011 г. № ИБ – 733/12.

Программа одобрена на заседании кафедры теоретической информатики.

17.10.2012 г. (протокол № 11)

Заведующий кафедрой д. ф.-м. н., проф. В.А. Соколов

Автор д. ф.-м. н., доцент Е.В. Кузьмин