МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

МОО ВО Кыргызско-Российский Славянский университет имени первого Президента Российской Федерации Б.Н. Ельцина

Высшая школа Цифровых технологий

Фонд оценочных средств

по дисциплине «Введение в синтез программ»

Уровень высшего образования

МАГИСТРАТУРА

Направление подготовки

09.04.04 - РФ, 710400 - КР Программная инженерия

(код и наименование направления подготовки)

"Основы анализа и разработки приложений с большими объемами распределенных данных" (профиля) образовательной программы)

Квалификация

магистр

Фонд оценочных средств предназначен для контроля знаний обучающихся по направлению подготовки 09.04.04 – РФ, 710400 - КР «Программная инженерия» по дисциплине « Введение в синтез программ».

Фонд оценочных средств рассмотрен и утвержден
на заседании Высшей школы Цифровых технологий протокол №1 от 02.09.26

Директор Высшей школы Цифровых технологий:

к.т.н., доцент Хмелева И.В.

Руководитель образовательной программы:

д.т.н., профессор Лыченко Н.М.

Исполнители (разработчики):

Доцент, к.т.н., Шошмина И.В. (СПбПУ)

СОГЛАСОВАНО:
Зам. директора по учебной работе
Политехнического института КРСУ Комарцов Н.М.

Раздел 1. Перечень компетенций, с указанием этапов их формирования в процессе освоения дисциплины/практики

Формируемые компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций	Виды оценочных средств/ шифр раздела в данном документе
ПК-6: Способен разработать программное обеспечение на основе встроенной аналитики больших данных (ИД-3 ПК-6: осуществляет разработку систем обработки больших данных с использованием суперкомпьютерных технологий).	Знать: – принципы построения систем обработки данных и синтеза программ по формальным спецификациям; – синтаксические грамматики, перечислительный поиск, SAT/SMT и ограничения; – основы типизации, логики Хоара и нейросимвольного синтеза.	Блок А – задания репродуктивного уровня – Устный опрос – Тестовые задания
	Уметь: – задавать грамматики, формальные требования и ограничения к синтезируемой программе; – применять перечислительный, стохастический поиск и SAT/SMT-решатели; – сравнивать альтернативные программы, проверять корректность и оценивать вычислительную сложность.	Блок В – задания реконструктивного уровня – Практические задания № 1–13
	Владеть: – методами синтеза и формальной проверки программ; – инструментами SAT/SMT и средствами поиска в пространстве решений; – навыками документирования, воспроизводимого тестирования и анализа синтезированных алгоритмов обработки данных.	Блок С – задания практико-ориентированного и/или исследовательского уровня – Самостоятельные задания № 1–13 – Защита отчётов

Раздел 2. Технологическая карта дисциплины
Введение в синтез программ

Курс 1, семестр 2, количество ЗЕ – 3, отчётность – зачёт с оценкой

Название модулей дисциплины согласно РПД	**Контроль**	**Форма контроля**	**зачетный минимум**	**зачетный максимум**	**график контроля**
Модуль 1					
Синтаксический синтез программ: методы перечисления и стратегии поиска	Текущий контроль	Практические работы № 1–5	4	8	25
	Рубежный контроль	Самостоятельные задания № 1–5	9	15	
Модуль 2					
Синтез по ограничениям и спецификациям: SAT/SMT, формальные требования и наброски	Текущий контроль	Практические работы № 6–9	4	8	31
	Рубежный контроль	Самостоятельные задания № 6–9	10	16	
Модуль 3					
Типизированный, дедуктивный и нейросимвольный синтез программ	Текущий контроль	Практические работы № 10–13	4	8	37
	Рубежный контроль	Самостоятельные задания № 10–13	9	15	
ВСЕГО за семестр			40	70	
Промежуточный контроль (зачёт с оценкой)		Письменная работа: вопросы и практическое задание	20	30	
Семестровый рейтинг по дисциплине			60	100	

Раздел 3. Типовые контрольные задания и иные материалы, необходимые для оценки планируемых результатов обучения по дисциплине (оценочные средства)

Блок А

Вопросы для опроса

Модуль 1. Синтаксический синтез программ: методы перечисления и стратегии поиска

1. Что такое программный синтез и чем он отличается от обычной компиляции?
2. Что представляет собой синтаксически управляемый синтез SyGuS?
3. Из каких элементов состоит грамматика пространства допустимых программ?
4. Как формулируется спецификация по входным и выходным примерам?
5. В чём различие синтеза по примерам и синтеза по формальной спецификации?
6. Как устроен перечислительный поиск синтаксических деревьев?
7. Что понимают под пространством поиска и его отсечением?
8. Как корректно удалять эквивалентные кандидаты?
9. Что такое взвешенный перечислительный поиск и как задаются веса правил?
10. Чем поиск снизу вверх отличается от поиска сверху вниз?
11. Как работает стохастический поиск, и что является функцией качества?
12. Какие метрики позволяют сравнить эффективность двух стратегий синтеза?

Модуль 2. Синтез по ограничениям и спецификациям: SAT/SMT, формальные требования и наброски

1. Что означает выполнимость булевой формулы и как устроена задача SAT?
2. Чем SMT отличается от SAT и что называют теорией ограничений?

3. Как кодируется ограничение на допустимые значения переменных?
4. Что означает результат sat, unsat или unknown?
5. Чем отличается модель решения от доказательства корректности?
6. Как обосновать отсутствие решения у системы ограничений?
7. Из каких частей состоит формальная спецификация функции?
8. Что такое синтез с ограничениями и как устанавливают допустимую грамматику?
9. Как преобразовать набор пар вход–выход в ограничения на искомую функцию?
10. Что такое набросок программы (program sketch) и «дырка» в шаблоне?
11. Как устроен цикл «генерация кандидата — проверка — уточнение» (CEGIS)?
12. Какие ограничения и риски есть у SAT/SMT-подходов при большом пространстве поиска?

Модуль 3. Типизированный, дедуктивный и нейросимвольный синтез программ

1. Какую роль играет система типов в синтезе программ?
2. Что такое типизированный терм и проверка корректности типов?
3. Что означает тройка Хоара {P} C {Q}?
4. Что такое инвариант цикла и как его использовать для доказательства корректности?
5. Как формулируется слабейшее предусловие?
6. Чем спецификация поведения отличается от тестового примера?
7. Что такое сепарационная логика и почему она полезна при работе с памятью?
8. Что означает фрейм-правило и локальность рассуждений?
9. Чем дедуктивный синтез отличается от перечислительного?
10. Как совместить обучение модели и символическую проверку результата?
11. Какие ограничения возникают при нейросимвольном синтезе?
12. Как проверять воспроизводимость и достоверность результата синтеза?

Тест (один из вариантов)

1. Синтез программ по грамматике и спецификации называется:
   А. SyGuS
   Б. Линковка
   В. Профилирование
   Г. Архивация
2. Какое свойство отличает корректное отсечение кандидатов?
   А. Удаляет все программы
   Б. Не исключает все допустимые решения
   В. Всегда сокращает память вдвое
   Г. Не требует спецификации
3. Если SMT-решатель вернул unsat, это означает:
   А. Найдена модель
   Б. Доказана невыполнимость заданных ограничений в принятой постановке
   В. Программа откомпилирована
   Г. Данные зашифрованы
4. Какой оператор чаще описывает условный выбор в грамматиках SyGuS?
   А. ite
   Б. import
   В. include
   Г. goto
5. Набросок (sketch) программы содержит:
   А. Только комментарии

Б. Неизвестные компоненты, подлежащие синтезу

В. Только двоичный код

Г. Полный тестовый отчёт

6. Для тройки Хоара {P} C {Q} условие P означает:

А. Постусловие

Б. Предусловие

В. Тип результата

Г. Длительность

7. Инвариант цикла необходим прежде всего для:

А. Ускорения компилятора

Б. Формального обоснования поведения цикла

В. Сжатия файла

Г. Визуализации

8. При нейросимвольном синтезе символическая часть часто нужна для:

А. Цветового оформления

Б. Проверки ограничений и корректности

В. Шифрования сети

Г. Обязательной замены тестов

Блоки B и C

Практические задания

Практическая работа № 1. Синтаксический синтез программ.

Задание. Задать грамматику для арифметических выражений из x, констант и операций +, −, ×. По трём парам вход–выход синтезировать выражение. Проверить найденное решение на дополнительных значениях.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 2. Отсечение пространства поиска программ.

Задание. Построить перечислитель арифметических выражений до глубины 3. Реализовать отсечение кандидатов по эквивалентности значений на контрольных примерах; сравнить число проверенных кандидатов до/после.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 3. Взвешенный перечислительный поиск.

Задание. Назначить вес правилам грамматики. Организовать приоритетный перебор кандидатов, сравнить время нахождения программы с невзвешенным поиском на одной спецификации.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 4. Поиск на основе представления пространства решений.

Задание. Представить поиск как граф состояний/дерево термов. Сравнить стратегии поиска в ширину и в глубину на ограниченной грамматике; измерить число посещённых состояний и используемую память.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 5. Стохастический поиск решений.

Задание. Реализовать случайный/эволюционный поиск формул для заданных пар вход–выход. Определить функцию качества, фиксировать генератор случайных чисел; сопоставить качество с перебором.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 6. Решение SAT/SMT-задач.

Задание. С помощью Z3Py сформулировать систему целочисленных ограничений, получить sat/unsat и интерпретировать модель. Изменить одно ограничение так, чтобы задача стала невыполнимой.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 7. Синтез программ на основе ограничений.

Задание. Задать неизвестные коэффициенты линейной функции f(x)=ax+b. По примерам сформулировать ограничения, найти a и b решателем и проверить формулу для всех значений в заданном диапазоне.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 8. Формальные спецификации для синтеза программ.

Задание. Составить предусловия, постусловия и примеры для функции max(x,y). Записать формальную спецификацию в терминах логических ограничений; проверить разные реализации.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 9. Синтез по наброскам программы (sketches).

Задание. Подготовить набросок программы с двумя неизвестными параметрами («дырками»). Определить ограничения на них, синтезировать параметры и выполнить тесты на корректность результата.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 10. Синтез программ на основе типов.

Задание. Для набора функций с типами Int→Int, Bool→Bool и Int×Int→Bool построить дерево возможных композиций. Исключить неверно типизированные комбинации, обосновать корректность.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 11. Логика Хоара и корректность синтезируемых программ.

Задание. Записать тройки Хоара для присваивания и простого цикла суммирования. Выделить инвариант и проверить на выбранном наборе тестов, соблюдаются ли пред- и постусловия.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 12. Сепарационная логика и дедуктивный синтез.

Задание. Для двух участков программы, изменяющих непересекающиеся структуры данных, составить локальные утверждения. Пояснить применение фрейм-правила и границы допустимого изменения состояния.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Практическая работа № 13. Нейросимвольный поиск.

Задание. Получить предложение программы с помощью нейросетевой/эвристической модели или подготовленного набора кандидатов. Формализовать ограничения и проверить решения тестами и SMT-решателем.

Форма представления результата: программный код или формальная модель, исходная спецификация, результаты проверки, краткий отчёт с выводами.

Задания на самостоятельную работу

Самостоятельное задание № 1. Синтаксический синтез программ.

Подготовить обзор языков/систем синтаксически управляемого синтеза и две грамматики для одной функции; пояснить ограничения каждой. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 2. Отсечение пространства поиска программ.

Сравнить два способа отсечения и привести пример некорректного отсечения, теряющего допустимое решение. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 3. Взвешенный перечислительный поиск.

Исследовать влияние изменения весов трёх правил грамматики на порядок перебора и число рассмотренных кандидатов. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 4. Поиск на основе представления пространства решений.

Составить сравнительную таблицу поиска в ширину, в глубину и с эвристической оценкой для одного пространства программ. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 5. Стохастический поиск решений.

Провести не менее пяти запусков стохастического поиска с различными начальными состояниями генератора; представить распределение качества. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 6. Решение SAT/SMT-задач.

Описать два SAT/SMT-примера: выполнимый и невыполнимый, приложить входную постановку и объяснить вывод решателя. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 7. Синтез программ на основе ограничений.

Разработать задачу синтеза арифметической функции по 4–6 примерам с ограничением на допустимые операции. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 8. Формальные спецификации для синтеза программ.

Сравнить формальную спецификацию функции со спецификацией по примерам; привести контрпример к недостаточному набору тестов. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 9. Синтез по наброскам программы (sketches).

Подготовить схему цикла CEGIS с контрпримером для выбранной функции; объяснить условия завершения. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 10. Синтез программ на основе типов.

Построить набор типовых правил вывода и пример отклонения некорректной композиции по типам. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 11. Логика Хоара и корректность синтезируемых программ.

Самостоятельно обосновать инвариант цикла и вывести слабейшее предусловие для простой последовательности присваиваний. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 12. Сепарационная логика и дедуктивный синтез.

Проанализировать пример манипулирования двумя независимыми структурами памяти и сформулировать фрейм-условие. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Самостоятельное задание № 13. Нейросимвольный поиск.

Сопоставить результаты нейросимвольной и чисто символической стратегии на одном примере; оценить корректность и воспроизводимость. Представить отчёт с постановкой задачи, экспериментальными данными и выводами.

Блок D (промежуточный контроль)

Вопросы для проверки уровня обученности ЗНАТЬ

1. Постановка задачи синтеза программ: синтаксис, семантика, спецификация.
2. Перечислительный синтез и стратегии поиска: сверху вниз, снизу вверх.
3. Эквивалентность программ и корректное отсечение пространства поиска.

4. Взвешенный перебор и стохастические методы поиска.

5. Задача выполнимости SAT; основные особенности SMT.

6. Формальное кодирование ограничений для синтеза функции.

7. Синтез по примерам и синтез по полным спецификациям.

8. Наброски программ и заполнение неизвестных частей.

9. Алгоритм CEGIS: кандидат, верификация, контрпример.

10. Системы типов как ограничения на синтезируемые выражения.

11. Логика Хоара: правила присваивания, последовательности, ветвления и цикла.

12. Инварианты циклов и слабейшие предусловия.

13. Сепарационная логика и фрейм-правило.

14. Проверка корректности синтезированных программ.

15. Нейросимвольный синтез: комбинация обучения и формальных ограничений.

16. Оценка качества и вычислительной эффективности синтеза.

Задачи/задания для проверки уровня обученности УМЕТЬ и ВЛАДЕТЬ

1. Задать грамматику арифметических выражений и синтезировать функцию по примерам; проверить результат.

2. Предложить способ сокращения пространства кандидатов и оценить его корректность.

3. Сформулировать и решить SMT-задачу на выбор неизвестных параметров функции.

4. Построить SAT-кодирование небольшого ограничения и интерпретировать результат.

5. Составить спецификацию функции max(x,y) и проверить два кандидата.

6. Завершить набросок программы, реализующей линейное преобразование данных.

7. Проверить тройку Хоара для фрагмента кода с циклом и сформулировать инвариант.

8. Проанализировать типовую корректность композиции функций.

9. Провести сравнение перечислительного и стохастического поиска по двум измеримым критериям.

10. Проверить предложенную нейросетевой моделью программу и построить контрпример при нарушении спецификации.

Пример

БИЛЕТ ДЛЯ ЗАЧЁТА С ОЦЕНКОЙ № ___

1. Теоретический вопрос: формулировка задачи SyGuS и роль грамматики в ограничении пространства поиска.

2. Теоретический вопрос: SAT/SMT и проверка корректности синтезированной программы.

3. Практическое задание: сформулировать ограничения на функцию $f(x)=a\cdot x+b$ по заданным вход–выход примерам, найти a и b, проверить два дополнительных значения x и представить вывод.

Раздел 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Применяемые оценочные средства:

- Защита практических работ № 1–13 (текущий контроль).
- Самостоятельные задания № 1–13, включая отчёты и воспроизводимые вычислительные эксперименты (рубежный контроль).
- Устный опрос и короткие тесты по вопросам блока A.
- Письменная работа на зачёте с оценкой, содержащая вопросы и практическое задание блока D.

Результаты оцениваются по технологической карте: текущий и рубежный контроль — до 70 баллов (минимум по карте — 40), промежуточная аттестация — до 30 баллов (минимум — 20); общая сумма — до 100 баллов. Для допуска к промежуточной аттестации необходимо представить все практические и самостоятельные работы. Итоговая оценка определяется общей суммой баллов и фактическим выполнением обязательных работ.

ШКАЛА ОЦЕНИВАНИЯ ПРАКТИЧЕСКИХ (ЛАБОРАТОРНЫХ) РАБОТ (текущий/рубежный контроль)

• 85–100 % от максимума: спецификация корректна, программа синтезирована, проверка воспроизводима, отчёт обоснован; замечаний нет.

• 70–84 %: задача решена, имеются небольшие неточности в постановке, измерениях или оформлении.

• 60–69 %: результат в основном получен, но проверка ограничена либо есть существенные замечания к анализу.

• 31–59 %: реализована лишь часть решения, корректность недостаточно подтверждена.

• 0–30 %: решение отсутствует или не соответствует требованиям.

ШКАЛА ОЦЕНИВАНИЯ КОНТРОЛЬНЫХ РАБОТ (рубежный контроль)

• 85–100 %: все этапы исследования и формальная постановка выполнены, вывод обоснован сравнимыми данными.

• 70–84 %: основная задача решена правильно, имеются несущественные ошибки.

• 60–69 %: выполнена большая часть задания, выводы неполные.

• 31–59 %: частичное решение с существенными ошибками в методе или результатах.

• 0–30 %: решение не представлено или выполнено неверно.

ШКАЛА ОЦЕНИВАНИЯ ПИСЬМЕННОГО ОПРОСА (промежуточный контроль – «ЗНАТЬ»)

• 9–10 баллов: точное объяснение понятий и методов, корректные ограничения, формулы и примеры.

• 7–8 баллов: хорошее знание материала при одной–двух небольших неточностях.

• 6 баллов: понимание основных понятий при недостаточной полноте объяснения.

• 3–5 баллов: фрагментарные знания с заметными ошибками.

• 0–2 балла: существенное непонимание или отсутствие ответа.

ШКАЛА ОЦЕНИВАНИЯ ПРАКТИЧЕСКИХ ЗАДАНИЙ (промежуточный контроль – «УМЕТЬ и ВЛАДЕТЬ»)

• 18–20 баллов: задача полностью решена, программа/модель проверена, ограничения и выводы обоснованы.

• 14–17 баллов: верное решение с небольшими недостатками в проверке или оформлении.

• 10–13 баллов: существенная часть решения верна, но проверка или выводы неполны.

• 4–9 баллов: имеется частичное решение, присутствуют методические ошибки.

• 0–3 балла: решение отсутствует либо не соответствует постановке.

На зачёте с оценкой: два теоретических вопроса оцениваются по 0–5 баллов каждый (в сумме до 10), одно практическое задание — до 20; всего 30 баллов. Время выполнения — 80 минут. За семестр максимально начисляется 70 баллов; итоговая сумма не превышает 100 баллов.

Перевод баллов в оценку: 85–100 — «отлично»; 70–84 — «хорошо»; 60–69 — «удовлетворительно»; менее 60 — «неудовлетворительно».

Раздел 5. Методические указания для обучающегося по освоению дисциплины и выполнению контрольных заданий

Текущий контроль осуществляется в ходе защиты практических работ. Отчёт должен содержать: постановку задачи, грамматику/спецификацию, ограничения, исходный код, результаты запуска, тесты или формальную проверку, анализ корректности и выводы. Рубежный контроль проводится посредством самостоятельных заданий и защиты отчётов по ним.

Для допуска к зачёту с оценкой студент представляет результаты практических и самостоятельных работ. Если задание предполагает инструментальный синтез, в отчёте указываются версия решателя, параметры запуска, исходные ограничения и воспроизводимые тесты.

5.1. Общие рекомендации к организации самостоятельной работы

Изучать разделы РПД последовательно: от синтаксического поиска к спецификациям и SAT/SMT, затем к типизированному, дедуктивному и нейросимвольному синтезу. При выполнении заданий сначала формулировать свойства требуемого результата, затем выбирать стратегию поиска и проверять корректность. Учитывать различие между успешным тестированием на примерах и доказательством выполнения полной спецификации.

5.2. Подготовка к практическим занятиям

До занятия изучить соответствующую лекцию и задания ФОС; подготовить среду Python, SMT-решатель или иной рекомендованный преподавателем инструмент, сформулировать критерии успешности. После занятия оформить отчёт с входными данными, описанием метода, программой, логами, тестовыми и экспериментальными результатами. К защите подготовить объяснение выбранных ограничений, корректности решения и возможных причин неуспеха синтеза.

Справочные ресурсы для самостоятельного освоения инструментов (дополняют, но не заменяют литературу РПД): спецификация SyGuS — https://sygus-org.github.io/language/ ; руководство Z3Py — https://microsoft.github.io/z3guide/programming/Z3%20Python/Introduction/ ; руководство Rosette для solver-aided programming — https://docs.racket-lang.org/rosette-guide/ .