Лекция 5. Спецификация и верификация программ с указателями

Цель лекции

Узнать особенности спецификации и верификации функций, оперирующих с указателями и массивами.

Содержание

Основные конструкции

- Валидность диапазона указателей лучше специфицировать без квантора всеобщности: `\valid(array + (0 .. size - 1));`
- Постусловие имеет дело с двумя состояниями памяти: до вызова функции и после вызова функции: *метки памяти* - `Pre` и `Post`
- Чтобы разыменовать указатель, надо указать состояние памяти: `\at(expression, label)`
- У предиката метку памяти можно указать явно: `p{L}(n)`

Пример: сортировка выбором

- `sort_1.c` - спецификация сортировки
- `sort_2.c` - реализация сортировки выбором
- `sort_3.c` - доказательство safety
- `sort_4.c` - доказательство упорядоченности
- `sort_5.c` - доказательство перестановочности

Выводы из примера

- Солверам надо подсказывать, как нужно инстанцировать аксиомы
- Полезна бывает аксиома о том, что значение лоджика или предиката не изменится, если такая-то часть памяти между двумя метками не менялась
- Предикаты, аксиомы, леммы, лоджики могут иметь несколько меток памяти
- Можно задавать имя дополнительной метке памяти при помощи ghost
- В начале итерации цикла содержимое памяти надо вручную связывать с содержимым памяти до цикла, если в цикле есть присваивание в эту память

Содержание

Модель памяти

Модель памяти состоит из бесконечного множества блоков.
Блоки бывают:

- выделенными с ненулевым размером
- выделенными с нулевым размером
- невыделенными (свободными)

Блоки и указатели

- Блок представляется тройкой («идентификатор», «адрес начала», «размер»). Все эти атрибуты есть у **всех** блоков (даже свободных). Вид блока можно закодировать в атрибуте «размер»: он неотрицательный для выделенных блоков и отрицательный для свободных.
- Выделение и освобождение памяти - это смена размера блока. Блоки не появляются и не исчезают.
- Указатель - это блок и смещение от его начала. Смещение может быть любым числом (даже выходящим за границы, задаваемые размером, но такие указатели нельзя разыменовать). Блок может быть любого вида (даже свободный).

Внутренние структуры

Модель памяти состоит из нескольких таблиц:

- таблицы блоков (`alloc_table`)
- таблицы тегов (`tag_table`)
- таблицы значений (`map from pointer to value type`)

Валидность проверяется по таблице блоков и тегов, разыменование делается при помощи таблицы значений.

Нормализация кода

Все типы приводятся к указателям на структуру с единственным полем (отсюда такие длинные имена типов в теории Why3, которую генерирует AstraVer).
Для упрощения верификации делаются предположения:

- невыровненных указателей нет
- указатели разных типов не указывают в одну область памяти (или внутрь нее)
- указатели не преобразуются между типами указателей
- память неограничена

Предопределенные предикаты и лоджики

- `\freeable{L}(p)` - в метке памяти L указатель p указывает на начало выделенного блока (любого размера)
- `\allocable{L}(p)` - в метке памяти L указатель p указывает на начало невыделенного блока
- `\offset_min{L}(p)` - в метке памяти L значение смещения от p до начала блока указателя p
- `\offset_max{L}(p)` - в метке памяти L значение смещения от p до конца блока указателя p, если блок выделен

Пример

- `graph_1.c` — спецификация функций создания и удаления графа
- `graph_2.c` — определения функций
- `graph_3.c` — доказательство полной корректности (обратите внимание, как свойства массива `g->vertices` после первого цикла в функции создания сохранились по окончании второго цикла)