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