

*Пошаговое выведение свойств в Isabelle/HOL**Когтенков Александр Валентинович¹*¹ «Лаборатория Касперского»

2025-11-05

Требования к решению задания

Решение задания должно удовлетворять следующим требованиям:

- список импортируемых теорий должен остаться неизменным;
- допустимыми методами доказательства являются только *blast*, *cases*, *induction*, *simp*;
- доказательство должно быть прямым и не использовать *apply*-скрипты.

Термин «прямое» рассмотрен в подготовительных материалах.

Задание

Файл `Task.thy` содержит шаблон с незавершёнными фрагментами задач, которые необходимо решить. Задачи разбиты на три не связанные между собой части. Поэтому каждую из частей можно решать независимо друг от друга.

Запуск: `Isabelle2025 Task.thy`*I. Свойства свёртки значений истинности*

Специализация функции объединения элементов, используемой при свёртке, позволяет вывести дополнительные свойства элементов свёртки.

1. Доказать лемму *foldr_conj_init*, утверждающую, что если правая свёртка конъюнкцией по всем элементам списка некоего булева свойства истинна, то начальный элемент этой свёртки истинен.
2. Доказать лемму *foldr_conj_prop*, утверждающую, что если правая свёртка конъюнкцией по всем элементам списка некоего булева свойства истинна, то все элементы списка обладают указанным свойством.

Очки: 4

Очки: 8

II. Эквивалентность функций взятия префикса

Одну и ту же функцию можно задать разными способами. Иногда эта вынужденная мера. При этом доказательства для одного варианта могут быть проще, чем для другого. В этом разделе моделируется случай, когда самостоятельно заданная функция эквивалентна библиотечной, для которой уже доказано много утверждений.

1. Завершить декларацию примитивно-рекурсивной функции *pick*, которая возвращает список из заданного числа первых элементов заданного списка или весь список, если заданное число элементов больше его длины, используя рекурсию по первому аргументу.

Очки: 6

- Доказать лемму *pick_length*, что длина списка, возвращаемого функцией *pick*, является минимумом из значения первого аргумента и длины списка, переданного во втором аргументе.
- Доказать лемму *pick_items*, что все элементы списка, возвращаемые функцией *pick*, являются элементами исходного списка на тех же местах в списке.
- Посмотреть, что лемма *pick_take_eq* доказывает, что функции *pick*, заданная индукцией по первому аргументу, и библиотечная *take*, заданная индукцией по второму аргументу, эквивалентны. В доказательстве использованы утверждения, доказанные выше.

Очки: 6

Очки: 8

Очки: 0

В этом пункте ничего доказывать не требуется, просто насладитесь результатом.

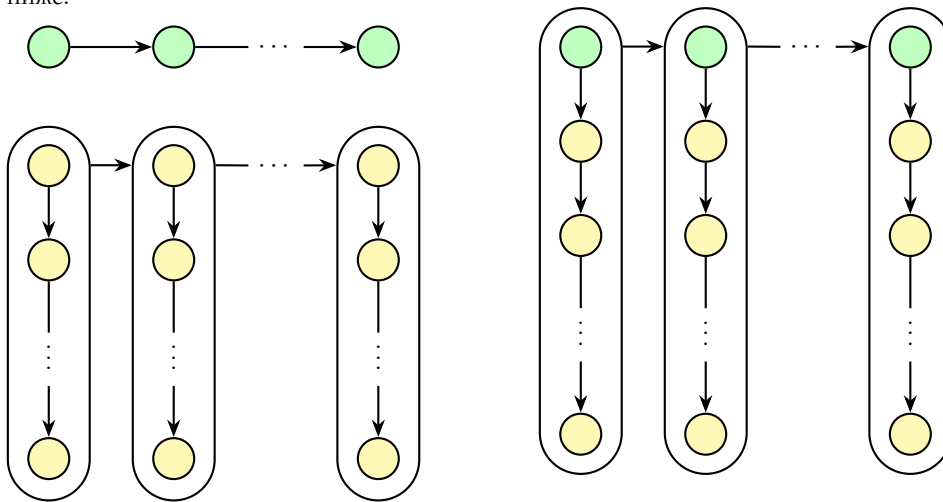
III. Свойства функций над списком списков

Сложные структуры данных можно моделировать, используя стандартные списки. Функции работы с такими структурами могут содержать дополнительные проверки для сохранения свойств структуры.

Подобные задачи возникли при моделировании хранилища значений переменных специализированного языка программирования. Оно похоже на таблицу, к которой можно добавлять строки/столбцы.

- Задать примитивно-рекурсивную функцию *conss*, которая последовательно помещает элементы списка из первого аргумента в начало списков из второго аргумента, как изображено на рисунке ниже:

Очки: 8



Для проверки, сразу после декларации функции *conss* указаны леммы *conss_test_N*, которые при правильном задании функции будут истинны.

Если длины списков первого и второго аргумента отличаются, «лишние» хвосты отбрасываются.

- Доказать лемму *conss_length*, что длина списка, возвращаемого функцией *conss*, равна минимуму длин списков в первом и втором аргументе.
- Доказать лемму *conss_Cons_Cons*, что применение функции *cons* к непустым спискам эквивалентно списку, головой которого является список с головой из головы первого аргумента и хвостом из головы второго аргумента, а хвостом — применение этой функции к хвостам первоначальных списков.

Очки: 8

Очки: 2

- | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| <p>4. Доказать лемму <i>hd_conss_tl</i>, что если все списки в списке списков непустые, то применение функции <i>conss</i> к спискам из голов и хвостов этих списков даёт исходный список.</p> | <p>Очки: 4</p> |
| <p>5. Доказать лемму <i>hd_conss_tl_filter</i>, что применение функции <i>conss</i> к головам и хвостам списков, отфильтрованных по непустоте и некоторому свойству, эквивалентно фильтрации этих списков по тем же критериям.</p> | <p>Очки: 4</p> |
| <p>6. Доказать лемму <i>tl_filter_hd</i>, что непустой список входит в множество списков тогда и только тогда, когда его хвост входит в множество хвостов списков, головой которого является голова указанного списка.</p> | <p>Очки: 24</p> <p>Возможно, для доказательства потребуются вспомогательные леммы.</p> |

Оценка решения

Решение задания оценивается по сумме очков за все задачи. Максимальное число очков за каждую задачу указано на полях справа от её формулировки.

Нарушение требований, указанных в разделе [Требования к решению задания](#), повлечёт снижение оценки соответствующего пункта согласно следующей таблице:

Нарушение	Коэффициент
Изменение списка импорта	0
Метод доказательства не из списка	0.5
Непрямое доказательство	0.75

Коэффициент за комбинацию нарушений рассчитывается как произведение коэффициентов за каждое нарушение.