Summary of Changes to ADT-IND

Lucas Zavalía

March 3, 2022

I decided to test a new approach for testing common lemmas. The knowledge scheme will now add and test lemmas incrementally. First I observed that there are two kinds of common lemmas. Homogenous lemmas are those that express a particular property of one inductive function. Heterogenous lemmas are those that express a relationship between inductive functions. The idea here is to test homogenous lemmas before heterogenous lemmas. Here are some examples:

Homogenous	Heterogenous
$(\forall x : Lst)\ len(x) \geq 0$	$(\forall x : Lst)\ len(rev(x)) = len(x)$
$(\forall x : Lst)\ rev(rev(x)) = x$	$(\forall x_1, x_2 : Lst)\ len(app(x_1, x_2)) = len(x_1) + len(x_2)$

Below is the summary of the changes I made:

1. Changed type of knowledge scheme from `ExprVector` to `map<Expr, ExprSet>` for better organization.
2. Made `buildKnowledgeScheme` more concise.
3. Changed the function `buildKnowledgeScheme` to add homogenous lemmas ad-hoc by analyzing function declarations in the current subgoal instead of building all possible lemmas at once.
4. Consolidated unecessary functions (such as `getFDECLs`).
5. Cleaned up the `printKnowledgeScheme` function.
6. Added (but haven't implemented) a function, `expandKnowledgeScheme`, to add heterogenous lemmas at a later point.
7. Added counter to keep track of backtracks and prevent the knowledge scheme from being used too many times.

Below is a list of known bugs:

1. The `useKnowledgeScheme` function does not find certain valid rewrites
2. The backtrack counter does not seem to prevent the knowledge scheme from being used.