List Theorem Solving:

A segmentation approach based on length and congruence closure

by *Jeroen Kool*

Outline

Outline

Problem

- Lists show up in often in programming
- We want a way to check program correctness
- There are several program verification tools, such as Viper, VeriFast, Iris, and VST that produce side conditions
- Therefore we need a good solver for lists

What did we do?

We create a system to solve theorems about lists

- Theoretical inference system
- Practical implementation in Coq

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

List with numbers 1 until 4

$$[1; 2; 3; 4]$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

List with numbers 1 until 4

$$[1; 2; 3; 4]$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

Empty list

$$[\ \ ]$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

List containing one element

$$[[]1]$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

Concatenation of two lists

$$[1;2]\ ++\ [3;4] = [1;2;3;4]$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

The reverse of a list

$$\texttt{rev}\ [1;2;3;4] = [4;3;2;1]$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

Length of a list

$$\texttt{length}\ [1; 2; 3; 4] = 4$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

Give a list with n number of the same elements

$$\texttt{repeat}\ 4\ 2 = [2; 2; 2; 2]$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

Take the first n elements of a list

$$\texttt{take}\ 2\ [1; 2; 3; 4; 5] = [1; 2]$$

Allowed operators

- nil
- singleton
- append
- reverse
- length
- repeat
- take
- drop
- nth
- map
- update
- flip_ends

Example

Example

$$l_1 = l_3 \to$$
$$\text{rev } l_1 \mathbin{++} \text{rev } l_2 = \text{rev } l_3 \mathbin{++} \text{rev } l_4 \to$$
$$l_2 = l_4$$

Which we can also denote, in Coq's judgment style, as

Example

$$\frac{\begin{array}{c}\text{H1: } l_1 = l_3 \\ \text{H2: } \text{rev } l_1 \mathbin{++} \text{rev } l_2 = \text{rev } l_3 \mathbin{++} \text{rev } l_4\end{array}}{l_2 = l_4}$$

Example

Example

$$l_1 = l_3 \rightarrow$$
$$\text{rev } l_1 +\!\!+ \text{ rev } l_2 = \text{rev } l_3 +\!\!+ \text{ rev } l_4 \rightarrow$$
$$l_2 = l_4$$

Which we can also denote, in Coq's judgment style, as

Example

$$\frac{\begin{array}{c}\text{H1: } l_1 = l_3 \\ \text{H2: } \text{rev } l_1 +\!\!+ \text{ rev } l_2 = \text{rev } l_3 +\!\!+ \text{ rev } l_4\end{array}}{l_2 = l_4}$$

Prior work

	Own work	VST	SMT string theory
nil	x	x	x
append	x	x	x
reverse	x	x	
length	x	x	x
take	x	x	x
drop	x	x	x
nth	x	x	x
repeat	x	x	x
map	x	x	
update	x	x	x

Table: Supported operators of prior work

Outline

Overview of key ideas

- Operation rearrangement
- Reverse list assumptions
- List segmentation
- Take and drop substitution (At the end if time left)

Operation rearrangement

- Defined a (conditional) term rewrite system
- With the rewrite system we aim to place the operators in a certain order.
- Obtain a normal form

$$[\,] > \; ++ \; > \texttt{repeat} > \texttt{map} > \texttt{rev} > \texttt{drop} > \texttt{take} > \texttt{nth}$$

Example operation rearrangement

We have, for example, the following rules:

$$\texttt{rev}(l_1 \;++\; l_2) = \texttt{rev}\ l_2 \;++\; \texttt{rev}\ l_1$$
$$\texttt{rev}\ (\texttt{repeat}\ n\ v) = \texttt{repeat}\ n\ v$$

Example

We want to rewrite the following expression:

$$\texttt{rev}\ (l \;++\; (\texttt{repeat}\ n\ v))$$

We apply the first rule and get:

$$(\text{rev}\ (\texttt{repeat}\ n\ v)) \;++\; \texttt{rev}\ l$$

We apply the second rule and get:

$$(\texttt{repeat}\ n\ v) \;++\; \texttt{rev}\ l$$

Introduction

Key ideas

Evaluation and conclusion

Example operation rearrangement

We have, for example, the following rules:

$$\texttt{rev}(l_1 \ ++\ l_2) = \texttt{rev}\ l_2 \ ++\ \texttt{rev}\ l_1$$

$$\texttt{rev}\ (\texttt{repeat}\ n\ v) = \texttt{repeat}\ n\ v$$

Example

We want to rewrite the following expression:

$$\texttt{rev}\ (l \ ++\ (\texttt{repeat}\ n\ v))$$

We apply the first rule and get:

$$(\text{rev}\ (\texttt{repeat}\ n\ v)) \ ++\ \texttt{rev}\ l$$

We apply the second rule and get:

$$(\texttt{repeat}\ n\ v) \ ++\ \texttt{rev}\ l$$

Example operation rearrangement

We have, for example, the following rules:

$$\texttt{rev}(l_1 \; ++ \; l_2) = \texttt{rev}\ l_2 \; ++ \; \texttt{rev}\ l_1$$

$$\texttt{rev}\ (\texttt{repeat}\ n\ v) = \texttt{repeat}\ n\ v$$

Example

We want to rewrite the following expression:

$$\texttt{rev}\ (l \; ++ \; (\texttt{repeat}\ n\ v))$$

We apply the first rule and get:

$$(\text{rev}\ (\texttt{repeat}\ n\ v)) \; ++ \; \texttt{rev}\ l$$

We apply the second rule and get:

$$(\texttt{repeat}\ n\ v) \; ++ \; \texttt{rev}\ l$$

Example operation rearrangement

We have, for example, the following rules:

$$\texttt{rev}(l_1 \ ++\ l_2) = \texttt{rev}\ l_2\ ++\ \texttt{rev}\ l_1$$
$$\texttt{rev}\ (\texttt{repeat}\ n\ v) = \texttt{repeat}\ n\ v$$

Example

We want to rewrite the following expression:

$$\texttt{rev}\ (l\ ++\ (\texttt{repeat}\ n\ v))$$

We apply the first rule and get:

$$(\mathrm{rev}\ (\texttt{repeat}\ n\ v))\ ++\ \texttt{rev}\ l$$

We apply the second rule and get:

$$(\texttt{repeat}\ n\ v)\ ++\ \texttt{rev}\ l$$

Reverse list assumptions

Lemma

$$l_1 = l_2 \to \mathit{rev}\ l_1 = \mathit{rev}\ l_2$$

We use this often in combination with the following rewrite rule:

$$\texttt{rev}\ (\texttt{rev}\ l) = l$$

This allows us to prove:

Example

$$\frac{\texttt{rev}\ l_1 = \texttt{rev}\ l_2}{l_1 = l_2} \quad\Longrightarrow\quad \frac{\begin{array}{c}\texttt{rev}\ l_1 = \texttt{rev}\ l_2 \\ \mathrm{rev}\ (\mathrm{rev}\ l_1) = \mathrm{rev}\ (\mathrm{rev}\ l_2)\end{array}}{l_1 = l_2}$$

Reverse list assumptions

Lemma

$$l_1 = l_2 \rightarrow \mathit{rev}\ l_1 = \mathit{rev}\ l_2$$

We use this often in combination with the following rewrite rule:

$$\texttt{rev}\ (\texttt{rev}\ l) = l$$

This allows us to prove:

Example

$$\frac{\texttt{rev}\ l_1 = \texttt{rev}\ l_2}{l_1 = l_2} \quad \Longrightarrow \quad \frac{\begin{array}{c}\texttt{rev}\ l_1 = \texttt{rev}\ l_2 \\ \text{rev}\,(\text{rev}\ l_1) = \text{rev}\,(\text{rev}\ l_2)\end{array}}{l_1 = l_2}$$

List segmentation

Main idea is using

Lemma

$$\text{length } l_1 = \text{length } l_3 \to l_1 ++ l_2 = l_3 ++ l_4 \to (l_1 = l_3 \land l_2 = l_4)$$

We utilize this with

- Asserting hypotheses for the length of lists

$$l_1 = l_2 \to \text{length } l_1 = \text{length } l_2$$

- Using congruence closure algorithm

Example, prove a theorem with the key ideas

We want to prove the following lemma:

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H2: } \mathsf{rev}\ l_1 \mathbin{++} \mathsf{rev}\ l_2 = \mathsf{rev}\ l_3 \mathbin{++} \mathsf{rev}\ l_4}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H2: } \text{rev } l_1 ++ \text{rev } l_2 = \text{rev } l_3 ++ \text{rev } l_4}{l_2 = l_4}$$

We first take the reverse of list hypothesis H2. We will omit hypothesis H2, because of space on our slide.

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } \text{rev}\,(\text{rev } l_1 ++ \text{rev } l_2) = \text{rev}\,(\text{rev } l_3 ++ \text{rev } l_4)}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H2: } \mathrm{rev}\ l_1 \mathbin{++} \mathrm{rev}\ l_2 = \mathrm{rev}\ l_3 \mathbin{++} \mathrm{rev}\ l_4}{l_2 = l_4}$$

We first take the reverse of list hypothesis H2. We will omit hypothesis H2, because of space on our slide.

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } \mathrm{rev}\,(\mathrm{rev}\ l_1 \mathbin{++} \mathrm{rev}\ l_2) = \mathrm{rev}\,(\mathrm{rev}\ l_3 \mathbin{++} \mathrm{rev}\ l_4)}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } \mathrm{rev}\,(\mathrm{rev}\ l_1 \mathbin{++} \mathrm{rev}\ l_2) = \mathrm{rev}\,(\mathrm{rev}\ l_3 \mathbin{++} \mathrm{rev}\ l_4)}{l_2 = l_4}$$

We then will apply operation rearrangement to obtain the normal forms in hypothesis H3. We first rewrite with the rule

$$\mathrm{rev}\,(l \mathbin{++} l') = \mathrm{rev}\ l' \mathbin{++} \mathrm{rev}\ l.$$

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } \mathrm{rev}\,(\mathrm{rev}\ l_2) \mathbin{++} \mathrm{rev}\,(\mathrm{rev}\ l_1) = \mathrm{rev}\,(\mathrm{rev}\ l_4) \mathbin{++} \mathrm{rev}\,(\mathrm{rev}\ l_3)}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\begin{array}{c}\text{H1: } l_1 = l_3 \\ \text{H3: } \text{rev}\,(\text{rev}\ l_1 \mathbin{++} \text{rev}\ l_2) = \text{rev}\,(\text{rev}\ l_3 \mathbin{++} \text{rev}\ l_4)\end{array}}{l_2 = l_4}$$

We then will apply operation rearrangement to obtain the normal forms in hypothesis H3. We first rewrite with the rule

$$\text{rev}\,(l \mathbin{++} l') = \text{rev}\ l' \mathbin{++} \text{rev}\ l.$$

Example

$$\frac{\begin{array}{c}\text{H1: } l_1 = l_3 \\ \text{H3: } \text{rev}\,(\text{rev}\ l_2) \mathbin{++} \text{rev}\,(\text{rev}\ l_1) = \text{rev}\,(\text{rev}\ l_4) \mathbin{++} \text{rev}\,(\text{rev}\ l_3)\end{array}}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } \mathrm{rev}\,(\mathrm{rev}\ l_2) \mathbin{++} \mathrm{rev}\,(\mathrm{rev}\ l_1) = \mathrm{rev}\,(\mathrm{rev}\ l_4) \mathbin{++} \mathrm{rev}\,(\mathrm{rev}\ l_3)}{l_2 = l_4}$$

We then rewrite with the rule

$$\mathrm{rev}\,(\mathrm{rev}\ l) = l.$$

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } l_2 \mathbin{++} l_1 = l_4 \mathbin{++} l_3}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } \text{rev}\,(\text{rev}\; l_2) \mathbin{++} \text{rev}\,(\text{rev}\; l_1) = \text{rev}\,(\text{rev}\; l_4) \mathbin{++} \text{rev}\,(\text{rev}\; l_3)}{l_2 = l_4}$$

We then rewrite with the rule

$$\text{rev}\,(\text{rev}\; l) = l.$$

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } l_2 \mathbin{++} l_1 = l_4 \mathbin{++} l_3}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } l_2 \mathbin{++} l_1 = l_4 \mathbin{++} l_3}{l_2 = l_4}$$

To be able to apply the segmentation idea, we first have to obtain information about the length of the lists from hypothesis H1.

Example

$$\text{H4: } \text{length } l_1 = \text{length } l_3$$

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } l_2 \mathbin{++} l_1 = l_4 \mathbin{++} l_3}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\text{H1: } l_1 = l_3 \qquad \text{H3: } l_2 \mathbin{++} l_1 = l_4 \mathbin{++} l_3}{l_2 = l_4}$$

To be able to apply the segmentation idea, we first have to obtain information about the length of the lists from hypothesis H1.

Example

$$\frac{\text{H4: } \text{length } l_1 = \text{length } l_3 \qquad \text{H1: } l_1 = l_3 \qquad \text{H3: } l_2 \mathbin{++} l_1 = l_4 \mathbin{++} l_3}{l_2 = l_4}$$

Example, prove a theorem with the key ideas

Example

$$\frac{\begin{array}{c}\text{H4: length } l_1 = \text{length } l_3 \\ \text{H1: } l_1 = l_3 \\ \text{H3: } l_2 \mathbin{++} l_1 = l_4 \mathbin{++} l_3\end{array}}{l_2 = l_4}$$

We now use our idea of segmentation:

$$\text{length } l_1 = \text{length } l_3 \to l_1 \mathbin{++} l_2 = l_3 \mathbin{++} l_4 \to (l_1 = l_3 \wedge l_2 = l_4)$$

With this idea we see that from the hypotheses follows that $l_2 = l_4$.

Outline

1. Introduction
2. Key ideas
3. **Evaluation and conclusion**

Benchmarks

We obtained three sorts of benchmarks

- Own creations (23)
- Coq's standard library and Iris extended standard library (10)
- VST (18)

A total of 51 benchmarks.

Results

	# of solved benchmarks
Own work	50
VST	37
SMT String Theory	21

Table: Number of solved benchmarks

Own work	60.10 sec
VST	1.02 sec

Table: Average time to solve one of the 37 overlapping lemmas

Conclusion

- Developed a solver with a wide solvability which is slow
- Provided a formal (inference) system that can be used to implement solvers for list theorems

Question?

Take and drop substitution

$$l = \text{take}\ n\ l \mathbin{++} \text{drop}\ n\ l$$

Lemma

We can chose new variables lt and ld, then the following will always hold:

$$\begin{gathered} l = lt \mathbin{++} ld \rightarrow \\ \text{length}\ lt = \min(n, \text{length}\ l) \rightarrow \\ \text{length}\ ld = \text{length}\ l - n \rightarrow \\ lt = \text{take}\ n\ l \land ld = \text{drop}\ n\ l \end{gathered}$$

The goal is to substitute all occurrences of take and drop.

Example, take and drop substitution

Example

$$\frac{\text{H1: length } l_1 = n}{\text{drop } n\,(l_1 \mathbin{++} l_2) = l_2}$$

Example, take and drop substitution

Example

$$\frac{\text{H1: length } l_1 = n}{\text{drop } n\,(l_1 \mathbin{++} l_2) = l_2}$$

We apply substitution

Example

$$\begin{array}{c} \text{H4: } l_1 \mathbin{++} l_2 = lt \mathbin{++} ld \\ \text{H3: length } lt = n \\ \text{H2: length } ld = \text{length}\,(l_1 \mathbin{++} l_2) - n \\ \dfrac{\text{H1: length } l_1 = n}{ld = l_2} \end{array}$$

Example, take and drop substitution

Example

$$\frac{\begin{array}{c}\text{H4: } l_1 \mathbin{++} l_2 = lt \mathbin{++} ld \\ \text{H3: length } lt = n \\ \text{H2: length } ld = \text{length } (l_1 \mathbin{++} l_2) - n \\ \text{H1: length } l_1 = n\end{array}}{ld = l_2}$$

We can now apply segmentation, because length $l_1 =$ length lt and conclude the proof.

Question?