

1 Announcements

Reading assignment for this part of the course: Chapters 8 and 9 of the book.

Informal homework: use ACL2 to prove all of the theorems we've seen in class.

2 Review

Last time we discussed how to use ACL2 to prove the version of the `rev-rev`. This required that we understand how ACL2 proves theorems using rewriting. For example, theorems give rise to rewrite rules and these rules have a lhs (left-hand side) and a rhs (a right-hand side). So, it is important when we state theorems that we take this into account and we *orient* the rules so that lhs is more complex than the rhs. Every time you state a theorem you *have to* think about the rewrite rule you get. At a higher level, you should aim to use rewrite rules to drive expressions into a “canonical” form.

We also got experience in determining what to do when proof attempts fail. The rough idea is to look at checkpoints and to use our model of how ACL2 works to come up with a plan that will allow ACL2 to make progress.

We also discussed the idea of step-wise refinement and how to prove theorems about your programs when you do that. The point was that we can replace simple functions

3 Using ACL2

3.1 Rewriting

We spent a good amount of time going over the rewriter and looking at what the rewriter will do on specific examples.

3.2 Pitfalls

Termination of functions is proven by ACL2, but there is no guarantee that the rewrite rules you put into the logical world are terminating, *e.g.*, last time the question was asked as to what happens if you prove both the following theorems.

```
(defthm app-associative1
  (equal (app (app a b) c)
         (app a (app b c))))
```

```
(defthm app-associative2
```

```
(equal (app a (app b c))
      (app (app a b) c))
```

Well, that will send the rewriter on a wild goose chase as it rewrites

```
(app (app a b) c)
```

to

```
(app a (app b c)))
```

and then back to

```
(app (app a b) c)
```

and then it repeats. ACL2s will stop after it does this a few times and will print out a note about what to do. Try it and you will see what the ACL2 rewriter was doing.