Lexical Scope
software engineering

Opaque functions in Dafny (performance/proveablity,readbility)

Posted on **May 15, 2014**, Modified on May 22, 2014

I am using Opaque functions in Dafny quite heavily, and for three main reasons:

OPAQUE FUNCTIONS CAN IMPROVE PERFORMANCE

I have a very large predicate which judges partial isomorphism between two program states. This predicate appears in the requires, ensures and bodies of a large number of lemmas and functions in my proof. The number of facts that are introduce by Dafny automatically revealing these function definitions seems to substantially slow down the proof step, sometimes to the point where it no longer goes through in an amount of time I am willing to wait.

For example take a predicate like this

```
prediate I(a:T,b:T)
{
   P(a,b) && Q(a,b) && R(a,b) && S(a,b)
}
predicate P(a:T,b:T) { ... }
predicate Q(a:T,b:T) { ... }
...
static lemma IThenSomethingThatReliesOnQ(a:T,b:T)
  requires I(a,b);
  ensures SomethingThatReliesOnQ(a,b);
{
   // Q automatically revealed, but so is P,R and S
}
```

Instead we can write

```
prediate I(a:T,b:T)
{
   P(a,b) && Q(a,b) && R(a,b) && S(a,b)
```

```
}
predicate {:opaque true} P(a:T,b:T) { ... }
predicate {:opaque true} Q(a:T,b:T) { ... }
...
static lemma IThenSomethingThatReliesOnQ(a:T,b:T)
  requires I(a,b);
  ensures SomethingThatReliesOnQ(a,b);
{
   reveal_Q(); // reveals Q only, not P,R and S
}
```

OPAQUE FUNCTIONS CAN IMPROVE READABILITY

I initially really liked the automatic function definition revealing feature, but as my proof got larger I liked it less. Particularly in the presence of big predicates (like my isomorphism predicate), I found it can get quite hard to understand which parts of the proof rely on which facts. This made it harder for me to work out what I need to establish in other parts of the proof (in this case, that the isomorphism has certain properties and that those can be used to show it is preserved by some particular program execution steps).

The example above shows how using opaque and reveal makes the proof more self documenting, allows us to automatically check that documentation and provides (I think) more insight into the proof.

OPAQUE FUNCTIONS HELP YOU FIND PLACES THAT NEED RE-VERIFYING

If you are using {:verify false} to restrict re-verification only to the elements you are working on, then having functions set to opaque helps you find all the places that depend on their definition. So, if you change the definition then you can easily find all the places that will need to be re-verified with the new definition.

Related posts:

1. **Inverting Maps in Dafny**
2. **Naming quantifiers in Dafny**
3. **Alternative distance function for Stable Abstractions Principle**
4. **Building Dafny Visual Studio Extension**

This entry was posted in **dafny** by **Tim Wood**. Bookmark the **permalink [http://www.lexicalscope.com/blog/2014/05/15/opaque-functions-in-dafny-performanceproveablityreadbility/]** .