PROPERTIES FOR ALL

BY IVY ROGATKO

WHY PROPERTY TESTING?

SIMPLE CACHE EXAMPLE

```
property "get for the empty cache returns nil", numtests: 5 do
   forall {key} <- {term()} do
     equals(nil, SimplerCache.get(key))
   end
 end
```

COMPONENTS OF A PROPERTY TESTING LIBRARY:

GENERATORS

SHRINKING

GENERATORS

1. A programatic way to create a set of values to be fed into your tests
2. Propcheck example: "pos_integer()" aka an integer from 1 to :infinity

SOME VERY SIMPLE EXAMPLES OF COMPOSITION

```
defp key(), do: term()
defp val(), do: term()
defp update_function(), do: function(1, term())
defp fallback_function(), do: function(0, term())
```

SHRINKING

1. Simplifying a counter-example (something that caused a test failure) through X cycles until a minimum example or maximum cycle count is reached
2. Usually can be derived from other generators as generators tend to be composed
3. Attempts to bring generators back to zero point

SHRINKING EXAMPLE: (I REMOVED SETTING A TTL)

```
property "Set ttl guarantees key dies after x time", numtests: 120 do
    forall {key, val, timer_ttl_ms} <- {term(), term(), integer(1, 100_000)} do
      {:ok, :inserted} = SimplerCache.insert_new(key, val)
      {:ok, :updated} = SimplerCache.set_ttl_ms(key, timer_ttl_ms)
      :timer.sleep(timer_ttl_ms + 10)
      equals(SimplerCache.get(key), nil)
    end
  end
```

```
!
Failed: After 1 test(s).
{{},{},#Fun}
0.8754956844752398 != {}

Shrinking

  1) property Set ttl guarantees key dies after x time (SimplerCacheTest)
     test/simple_cache_test.exs:15
     Property Elixir.SimplerCacheTest.property Set ttl guarantees key dies after x time() failed. Counter-Example is:
     [{0.3503550953096605, [], 2}]

     code: nil
     stacktrace:
       (propcheck) lib/properties.ex:126: PropCheck.Properties.handle_check_results/3
       test/simple_cache_test.exs:15: (test)
```

PROBLEMS WITH PROPERTY TESTING

WHAT IS MODEL CHECKING?

WHAT IS A MODEL?

1. A simple representation of the state of your system
2. For a simple cache an example model could just be a map

```
def initial_state(), do: %{}
```

ENTER TLA+

USE OF FORMAL METHODS AT AMAZON WEB SERVICES

http://lamport.azurewebsites.net/tla/formal-methods-amazon.pdf

WHY PROPERTY TESTING, THEN?

PROPERTY TESTING STATEFUL SYSTEMS

INSERT_NEW MACRO COMMAND EXAMPLE FOR SIMPLE CACHE

```
defcommand :insert_new do
    def impl(key, val), do: SimplerCache.insert_new(key, val)
    def args(_state), do: [key(), val()]
    def next(old_state, _args, {:error, _any}), do: old_state
    def next(old_state, [key, val], _any) do
        Map.put_new(old_state, key, val)
    end

    def post(entries, [key, _new_val], call_result) do
      case Map.has_key?(entries, key) do
        true ->
          call_result == {:error, :item_is_in_cache}
        false ->
          call_result == {:ok, :inserted}
      end
    end
  end
```

EXAMPLE FAILURE (FAULTY GET)

```
Shrinking ..........(10 time(s))
[{#{},{set,{var,1},{call,'Elixir.PropCheck.Test.CacheModel',get,[0]}}}]
History: [
  {%{}, {:call, PropCheck.Test.CacheModel, :get, [0]},
   {{:var, 1}, {:post_condition, false}}}
]
State: %{}
Env: %{{:var, 1} => {:post_condition, false}}
Result: {:post_condition, false}

  1) property run the cache commands (PropCheck.Test.CacheModel)
     test/cache_model_test.exs:16
     Property Elixir.PropCheck.Test.CacheModel.property run the cache commands() failed. Counter-Example is:
     [[{%{}, {:set, {:var, 1}, {:call, PropCheck.Test.CacheModel, :get, [0]}}}]]

     code: nil
     stacktrace:
       (propcheck) lib/properties.ex:126: PropCheck.Properties.handle_check_results/3
       test/cache_model_test.exs:16: (test)
```

EXAMPLE MACRO-DSL TEST OUTPUT

```
OK: Passed 100 test(s).

length of commands
minimum: 1
average: 26.5
maximum: 60

23% {'Elixir.PropCheck.Test.CacheModel',get_or_store,2}
16% {'Elixir.PropCheck.Test.CacheModel',insert_new,2}
15% {'Elixir.PropCheck.Test.CacheModel',get,1}
15% {'Elixir.PropCheck.Test.CacheModel',put,2}
14% {'Elixir.PropCheck.Test.CacheModel',delete,1}
8% {'Elixir.PropCheck.Test.CacheModel',update_existing,2}
7% {'Elixir.PropCheck.Test.CacheModel',size,0}
```

WHY USE STATEM VARIANT?

```
result == :ok || result == :no_possible_interleaving
```

HTTPS://GITHUB.COM/IROG/SIMPLER_CACHE

HTTPS://WWW.THEREALREAL.COM/CAREERS