Data Structure:

Consider the following definition for a data structure called *FilmState:*

```
# FilmState is a title, minutes, genre, and star-rating

data FilmState:
    | film(  title         ::  String,
             minutes       ::  Number,
             genre         ::  String,
             star-rating   ::  Number )
end
```

To make instances of this structure, I would write:

Casablanca = film("Casablanca", 102, "Drama", 5)

TheShining = film("The Shining", 146, "Horror", 4)

Choose one of your above instances, and note which dot-accessors you would use to access each of its fields:

Casablanca.title

Casablanca.minutes

Casablanca.genre

Casablanca.star-rating

Which of the following are functions that *could be* written based on the data definition for **`FilmState`**? Check all that apply

☑
```
# is-longer : FilmState, FilmState -> Boolean
# Consumes a two FilmStates, produces true if the
# first film is longer than the second
```

☐
```
# has-role : FilmState, String -> Booolean
# consumes a FilmState and the name of an actor. Produces
# true if that actor has a role in the given film
```

☑
```
# too-scary : FilmState -> Boolean
# consumes a FilmState and produces true if the film is a
# horror movie
```

☑
```
# editor : FilmState, Number -> FilmState
# Consumes a FilmState and number of minutes, produces
# a FilmState which is the same as the given film, but with
# the given number of minutes cut out
```

☐
```
# top-billed : FilmState -> String
# Consumes a FilmState and produces the top-billed
# actor in that film
```