

## Lab 9 – The Scheme Movie Database

1. Download the files `lab9-startcode.scm` and `scheme_movie_database.scm` from the class web page (under Labs), and open `lab9-startcode.scm` in DrRacket. In class, we wrote the function `find-director-of`, which takes a movie's title and the movie database as input, searches the database for a matching movie record, and returns the name of the movie's director. In our movie database, names and titles are represented as lists of symbols containing only lowercase letters or numbers. For example, to find the director of *Jurassic Park*, you would type:

```
(find-director-of '(jurassic park) movie-database)
```

Try searching the database for other directors. Who directed *Plan 9 from Outer Space*?

2. Write the function (**year-released *title database***), which takes a movie's title and the movie database as input, and returns the year that the movie was released. Use the definition of `find-director-of` as a guide in writing your function. What year was *Plan 9 from Outer Space* released? What about *Star Wars*? Test your function on a few other movie titles as well.
3. Write the function (**cast-of *title database***), which returns the list of actors who appeared in a specific movie. Who appeared in *Plan 9 from Outer Space*? Who appeared in *Star Wars*? Test your function on a few other movie titles as well.
4. We wrote the `find-movie` function in class:

```
(define find-movie
  (lambda (predicate? database)
    (cond
      [(null? database) 'unknown]
      [(predicate? (car database)) (car database)]
      [else (find-movie predicate? (cdr database))])))
```

This function takes a *predicate?* function and the movie database as input, and retrieves the first record in the database that satisfies the predicate. For example, we can retrieve the record for *Star Wars* as follows:

```
(find-movie (lambda (movie) (equal? (get-title movie) '(star wars)))
  movie-database)
```

Type this expression at the Scheme prompt to test it out. Then write and test similar expressions to retrieve (1) the movie *Jurassic Park*, (2) a movie that was directed by Clint Eastwood, (3) a movie that was released in 1999, and (4) a movie featuring the actor Brad Pitt.

5. In each case above, the movie's entire record is returned. If we just need a particular piece of information about the movie, we can call the appropriate selector function after retrieving the record. For example, to retrieve just the year that *Jurassic Park* was released, we could do this:

```
(get-year (find-movie (lambda (movie) (equal? (get-title movie) '(jurassic park)))
  movie-database))
```

Based on this example, write a new version of the function (**find-director-of *title database***) that uses `find-movie` as a helper function. Your new version should take exactly the same input parameters as before, but should not be recursive.

6. Rewrite **year-released** and **cast-of** in a similar way. Your new versions should take exactly the same input parameters as before, but should not be recursive.

7. The `find-movie` function returns the *first* movie in the database that satisfies some predicate function. Suppose we wish to retrieve *all* movies that satisfy the predicate. We can do this easily using the `filter` function. For example, the expression below will return a list of all movies directed by Clint Eastwood:

```
(filter (lambda (movie) (equal? (get-director movie) '(clint eastwood)))
       movie-database)
```

Type this expression at the Scheme prompt. How many movies directed by Clint Eastwood are there in the database? (Hint: use `length` to find out.)

8. Write the function (**movies-directed-by *director database***), which takes the name of a director and the movie database as input, and returns a list of all movies by that director. This function should use `filter` as a helper. How many movies in the database were directed by Steven Spielberg?
9. Write the function (**movies-made-in-year *year database***), which returns a list of all movies made in the specified year, using `filter` as a helper. How many movies were made in 1999?
10. Write the function (**movies-with-actor *actor database***), which returns a list of all movies in which the specified actor appeared. How many movies did Clint Eastwood appear in? What about Brad Pitt?
11. Suppose we want to create a list of the titles of all movies released in 1999. We can easily do this using `map`. Fill in the blank below, and then test your answer at the Scheme prompt:

```
(map _____ (movies-made-in-year 1999 movie-database))
```

12. Write a similar expression to find all of the years in which movies directed by Clint Eastwood were released. When did he direct his first movie (at least according to the information in the database)? What is the oldest movie in the database in which he appears as an actor? What year did Brad Pitt first appear in a movie (according to the database)?
13. Using `map` and `filter` as helpers, write a general function called **find-movies** that takes a *predicate?* function, a *selector* function, and the *database* as input, and retrieves a list of all of the movies satisfying *predicate?*. It should then apply the specified *selector* function to each record using `map`, and return the list of results. For example, we could retrieve the titles of all movies released in 1974 by calling

```
(find-movies (lambda (movie) (= (get-year movie) 1974))
             get-title
             movie-database)
```

14. Use your `find-movies` function to answer the questions below, using expressions similar to the one above.
- \* Which movies in the database were directed by Steven Spielberg?
  - \* Which movies in the database were released before 1940?
  - \* How many movies in the database were released between 1980 and 1989 (inclusive)?
  - \* Which directors were active in the years between 1950 and 1959 (inclusive)?
  - \* In which movies did the director also appear as an actor?