Post-Mortem

Assignment 03

October 17, 2018

We normally publish the post-mortem for an assignment after it has been marked and released. Here is a list of common errors provided by the graders for assignment 3.

Style and Spacing

• Constants, helper functions, data definition for a data type and template functions should be defined above the design recipe for the function they are used in. Some students are still defining their constants and helper functions between their examples and function definition, or after the main functions specified in the assignment.

General

• Many students did not make use of helper functions to reduce large chunks of repetitive code throughout the assignment.

• For now, equal? should only be used to compare two values of unknown types, or values that can take on more than one type. When the types of the arguments are known, use the most appropriate comparison function (e.g., symbol=?).

• To determine if a boolean value is true or false, use that value directly. Expressions like (boolean=? param true) or (equal? param false) are overly complex.

• Starting a new cond expression immediately in an else clause is unneeded. Instead, directly check for the next condition in the original cond expression. (e.g. use [... ...] directly instead of else (cond [... ...] ))

• Symbol values should not be defined as constants. As stated in the course notes, they are self-documenting, and defining them as constants does not provide any additional meaning.

Design Recipe

• Types in contracts should always be capitalized. This includes any user-defined structures like a Square or Piece.

• Helper functions require their own purpose, contract, and examples.

• Any requirements that are present in a data definition do not have to be repeated in a contract for a function that consumes that type. (e.g. use Square in contract instead of (make-square Nat Sym))

• Purposes should begin with a function header (e.g., (func-name param1 param2 param3)). These headers should include each of the parameter names used in the function.
• Purposes should meaningfully use each parameter name in the description of the function, and these references to the parameter names should be written exactly as they appear in the function header.

Question 2

• In parts (b) and (c), many students did not write the correct requirement specifying that provided Posn does not result in division by zero.

Question 3

• Many students either did not make use of helper functions in their solution, or had helper functions which were nearly identical.

• If your solution needs to check the boundaries of the board, you need to define constants for them. Many students with this kind of solution did not define constants for 1 and 8 or 2 and 7.

• Most of the students did not include selectors for square structure in their template functions. (e.g. (square-row (piece-pos piece)) and (square-column (piece-pos piece)))

• In part (c), some students did not compare vertical distance and horizontal distance for bishop, and row positions and column positions for rook, which make the code complex.

• In part (d), many students did not have requirement specifying that piece can only be ’knight.