Market Lane Simulation Report

# sim-utils.rkt

## sim-secs2hour

The sim-secs2hour function converts a total number of seconds into a list representing the equivalent time in hours, minutes, and seconds.

**Immutability:** The function uses let\* to bind values to hours, minutes, and secs, ensuring that these values are immutable within the scope of the let\* block.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Functional Composition:** The function uses nested quotient and remainder functions to calculate hours, minutes, and secs.

**Data Transformation:** The function transforms a single input (total seconds) into a structured output (list of hours, minutes, and seconds).

# sim-lane.rkt

## lane?

The lane? function checks whether a given value represents a valid lane.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Immutability:** The function does not modify any data; it only checks the properties of the input value.

**Data Transformation:** It transforms a single input (the value val) into a boolean output, indicating whether val represents a valid lane or not.

**Function Composition:** The function uses and to compose multiple conditions, ensuring that all conditions must be true for the overall expression to be true.

**Predicate Function:** The function acts as a predicate, returning true or false based on whether the input value meets certain criteria.

## lane-length

The lane-length function calculates the length of a lane, which is represented as a list.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Transformation:** It transforms a single input (the lane ln) into an integer output representing the length of the lane.

**Function Composition:** The function uses cdr to access the tail of the lane, and then length to calculate the length of that tail.

**Immutable Data:** The function does not modify the original lane; it only accesses its elements to calculate the length.

## lane-bussy

The lane-bussy function checks if a lane is busy or not.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Predicate Function:** It acts as a predicate, returning true (#t) if the lane is busy (i.e., if the first element of the second element of lane is not null) and false (#f) otherwise.

**Conditional Expression:** It uses an if expression to check if the lane is busy and return the appropriate result.

**Immutable Data:** The function does not modify the original lane; it only accesses its elements to determine if it is busy or not.

## lane-id

The lane-id function retrieves the ID of a lane, which is represented as a pair.

**Pure Function**: The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts a specific piece of data (the car of lane, which is the ID of the lane) without modifying the original data structure.

**Immutable Data:** The function does not modify the original lane; it only accesses its elements to retrieve the ID.

## lane-user

The lane-user function retrieves the user from a lane, which is represented as a pair.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts a specific piece of data (the car of the cdr of val, which is the user of the lane) without modifying the original data structure.

**Immutable Data:** The function does not modify the original val; it only accesses its elements to retrieve the user.

## lane-queue

The lane-queue function retrieves the queue from a lane, which is represented as a pair.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts a specific piece of data (the cdr of the cdr of val, which is the queue of the lane) without modifying the original data structure.

**Immutable Data:** The function does not modify the original val; it only accesses its elements to retrieve the queue.

# sim-event.rkt

## event

The event function constructs a new event based on the given time, id, and params if they form a valid event.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Transformation:** It constructs a new event by combining the time, id, and params into a list, which represents the event, without modifying the original data.

**Error Handling:** If the input parameters do not form a valid event (as determined by the event? function), it raises an error (using raise) instead of returning an invalid result.

**Function Composition:** It uses the event? function to validate the input parameters and construct the event only if they are valid.

## event?

The event? function checks whether a given value val represents a valid event.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts specific pieces of data from the input val, such as the time, event type, user, and lane ID, without modifying the original data structure.

**Conditional Expressions:** It uses nested if expressions to check various conditions and return true (#t) if val represents a valid event and false (#f) otherwise.

**Function Calls:** It calls other functions (event-time, event-type) to extract the time and event type from the event.

## event-time

The event-time function retrieves the time component of an event ev.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts the time component of the event ev using the car function, which accesses the first element of the event list.

**Immutable Data:** The function does not modify the original event ev; it only accesses its elements to retrieve the time.

## event-type

The event-type function retrieves the type component of an event ev.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts the type component of the event ev using the car function on the cdr of ev, which accesses the second element of the event list.

**Immutable Data:** The function does not modify the original event ev; it only accesses its elements to retrieve the type.

## event-params

The event-params function retrieves the parameters component of an event ev.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts the parameters component of the event ev using the cdr function twice, which accesses the third element of the event list.

**Immutable Data:** The function does not modify the original event ev; it only accesses its elements to retrieve the parameters.

## event-user

The event-user function retrieves the user component of an event ev.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts the user component of the event ev using the car function on the cdr of the cdr of ev, which accesses the third element of the event list.

**Immutable Data:** The function does not modify the original event ev; it only accesses its elements to retrieve the user.

## event-lane

The event-lane function retrieves the lane component of an event ev.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Data Extraction:** It extracts the lane component of the event ev using the cdr function three times, which accesses the fourth element of the event list.

**Immutable Data:** The function does not modify the original event ev; it only accesses its elements to retrieve the lane.

# sim-event-queue.rkt

## add-new-event

The sim-add-event function adds an event ev to a simulated event queue sim-event-queue while maintaining the queue sorted by time and user ID.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Function Composition:** It uses function composition to construct the event-less? function, which compares events based on their time and user ID.

**Higher-Order Function:** It uses sort as a higher-order function to sort the sim-event-queue based on the event-less? comparison function.

**Recursion:** It uses recursion in the insert-event function to insert the new event into the sorted queue.

**Immutable Data:** The function does not modify the original sim-event-queue; it creates a new sorted queue with the new event added.

# sim-lane-list.rkt

## lane-list?

The lane-list? function checks if a given list lst is a list of lanes.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Recursion:** It uses recursion to iterate over the elements of the list lst and check if each element is a lane using the lane? function.

**Conditional Expressions:** It uses nested if expressions to handle the base case (an empty list) and the recursive case (checking each element of the list).

**Immutable Data:** The function does not modify the original list lst; it only iterates over its elements to check if they are lanes.

## less-crowded

The less-crowded function returns the least crowded lane from a list of lanes sim-lanes.

**Pure Function:** The function does not have side effects and always returns the same output for the same input.

**Function Composition:** It uses the sort function as a higher-order function to sort the sim-lanes based on the lane-less comparison function.

**Recursion:** Although not explicitly shown in this function, the sort function likely uses recursion internally to sort the list.

**Immutable Data:** The function does not modify the original sim-lanes; it creates a new sorted list and returns the least crowded lane.