



# WORKSHOP 10

---

Missionaries and Cannibals in Prolog

CSX3004 Programming Languages  
Kwankamol Nongpong



# ACTIVITY: LOGIC PROBLEMS WITH PROLOG

**Time Limit:** 90 minutes

**Individual work**

# MISSIONARIES AND CANNIBALS

Three missionaries and three cannibals are on the left bank of the river. All of them need to cross to the right bank using a boat. However, the boat can only carry at most two people.

The cannibals can't outnumber the missionaries on either bank as the missionaries will get attacked.

Write a Prolog program that find the sequence of valid moves to get everyone safely across the river.

# STEP 1: REPRESENTING THE STATE

You can represent a state as:

`state(ML, CL, MR, CR, BoatSide)`

where

- ML = Number of missionaries on the left
- CL = Number of cannibals on the left
- MR = Number of missionaries on the right
- CR = Number of cannibals on the right
- BoatSide = left or right

**Task 1:** Write the initial state and the goal state.

## STEP 2: DEFINE VALID STATES

A state is valid if:

1. Numbers are not negative
2. Cannibals do not outnumber missionaries on either bank of the river (unless no missionaries are there).

**Task 2:** Write a rule `safe(State)` to check these conditions.

## STEP 3: BOAT MOVES

The boat can carry:

- 1 missionary
- 2 missionaries
- 1 cannibal
- 2 cannibals
- 1 missionary and 1 cannibal

**Task 3:** Write all possible moves as facts using the predicate `move(M, C)`.

# STEP 4: STATE TRANSITION

**Task 3:** Define the rule on how moves change the state.

transition(CurrentState, NextState)

# STEP 5: FIND THE SOLUTION

**Task 5:** Define the rule to explore moves.

`path(State, Goal, [State | Path])`



# STEP 6: RUN THE PROGRAM

Run the program using the following query.

?- initial(S), goal(G), path(S, G, Path).

# STEP 7: EXTENSION

Note the following changes.

- There are 4 missionaries and 4 cannibals.
- The boat can carry 3 people.

**Task 6:** Modify the program accordingly.

**Task 7:** Explain why the program sometimes loops infinitely. How to avoid revisiting states?