Game Probabilistic Lossy Channel Systems (GPLCS)

Sufficient:
- One channel
- One process
- Each state controlled by a player

Properties:
- Infinite state space
- Perfect channel

State: \( s = (I, n) \), where \( I \) is the set of inputs and \( n \) is the number of messages.

Transitions:
- Send: \( I \mapsto n \) for each \( i \in I \)
- Receive: \( n \mapsto I \) for each \( i_n \in I \)
- No-op: \( n \mapsto n \)

Probabilistic message loss:
- Each message is lost with prob \( A \) independently.

Stochastic Games

- Every GPLCS induces a stochastic game
- 3 types of states:
  - Player Good
  - Player Bad
  - Player Random

Overview

- Background
- Models
- Algorithm
- Conclusion

Repeated Reachability for GPLCS

- Input:
  - GPLCS
  - Set \( F \) of final states
- Output: Partition of states:
  - Winning for Good
  - Winning for Bad
Algorithm Correctness

Convergence means:
- No more water \( \implies \) Good can stay outside \( L, W \)
- No more island \( \implies \) Good can force \( \text{Prob}(\text{reach } F) > 0 \)

Correct?
- Convergence means:
  - No more water \( \implies \) Good can stay outside \( L, W \)
  - No more island \( \implies \) Good can force \( \text{Prob}(\text{reach } F) > 0 \)

So Good can force:
- Always \( \text{Prob}(\text{reach } F) > 0 \)
- For GPLCS, this implies \( \text{Prob}(\text{reach } F \text{ Inf. often}) = 1 \) (using attractors, difficult)

Overview

- Background
- Models
- Algorithm
- Conclusion

The Problem We Studied

- GPLCS: Protocol vs. Cracker
- Force-set:
  - Prob(reach G) > 0
- Backward search
- Winning sets for BAS:
  - Prob(reach G Inf. often) < 1
  - Search for islands and water

Algorithm

Convergence:
- Good can force:
  - Always \( \text{Prob}(\text{reach } F) > 0 \)
- For GPLCS, this implies \( \text{Prob}(\text{reach } F \text{ Inf. often}) = 1 \) (using attractors, difficult)

Extensions

- Construct winning strategies
- Concurrent games
- Simultaneous moves
- Repeated games
- Generalization of repeated reachability

The End

Questions?

Recall:
- \( s = L, \text{reps} \)
- Subword ordering:
  - \( \text{reps} \preceq \text{rups} \)
- Upward closure: all “bigger” states
- Message loss \( \implies \) subword

Convergence:
- Good can force:
  - Always \( \text{Prob}(\text{reach } F) > 0 \)
- For GPLCS, this implies \( \text{Prob}(\text{reach } F \text{ Inf. often}) = 1 \) (using attractors, difficult)

Future

Convergence:
- Good can force:
  - Always \( \text{Prob}(\text{reach } F) > 0 \)
- For GPLCS, this implies \( \text{Prob}(\text{reach } F \text{ Inf. often}) = 1 \) (using attractors, difficult)
Convergence:
Well Quasi Orders

If we can lose some messages... then we can lose more!

Conclusion: Random states one step before are upward closed

Theorem (Higman 1952):
Any sequence of increasing upward closed sets converges.

We apply it:
Player states
Random states

Convergence of Force-sets:
Well Quasi Orders

Theorem (Higman 1952):
Any sequence of increasing upward closed sets converges.

We apply it:
Upward closed

Every 2nd step is a loss
terminates!

The Fine Print:
Finite-Memory Strategies

We assume B is restricted to finite-memory strategies.

B cannot remember any amount of history.

Intuition: Evil cracker has a finite hard disk.

Need to prove correctness.

Repeated Reachability

F - set of "target states"

Is F reached infinitely often?

Example:
- Will it always return to the green state?

Leftover