Compositional Synthesis using Admissible Strategies

Under Prof. Purandar Bhaduri
Aman Raj (170101006)
Mayank Wadhwani (170101038)

AGENDA

01 INTRODUCTION AND BACKGROUND

02 OUR IDEA

03 EXAMPLES

04 RESULTS AND FUTURE WORK

01

INTRODUCTION AND BACKGROUND

Introduction and Motivation

1. Compositional Synthesis

- Helps in realizing controllers by breaking a component into multiple smaller components

2. Winning Strategies and Admissible Strategies

- Why do we need admissible strategies?

3. Graph Games and Safety Games

- Approach to attempt to solve controller synthesis problem.

Assume Admissible Synthesis

- For each player, search for an admissible strategy that is winning against all admissible strategies.
- So in a way, the players will assume that the other players will play admissibly and then play accordingly.
- AA-winning strategy profile
- **RULE AA:** Each player has a strategy winning against strategies of all other players.

02

OUR IDEA

Our Idea

- Working on Cooperative Components playing via admissible strategies.
- Objectives achieved cooperatively (we try to find a strategy profile)
- We are using Admissible strategies to allow the players to meet their objectives unlike what Damm and Finkbeiner have proposed.
- In our examples, we have only assumed safety objectives for the players.

Advantages

- A winning strategy may not always exist but the chances of admissible strategies existing is high (logically also, because if the other player will help, I am more probable to achieve my goals)

Limitations

- The components will now have to synchronise their strategies (even if some strategy is better, they may take the other one maybe because this strategy may be an obstacle for some other player)

03

EXAMPLES

Example 1: Scheduler

- Real life application of synthesis of a scheduler
- Three players in the system: User, Controller and Scheduler
- Each round involves three steps:
  User -> Controller -> Scheduler

Example 1: Scheduler

- User can send two different actions a1 and a2
- Controller must issue a corresponding request r1 and r2
- Scheduler has to schedule event q1 and q2

Example 1: Scheduler

Objectives

- For User, it is trivial.
- Controller must send request ri within k rounds, after receiving any action ai
- Scheduler must schedule event qi within t rounds on getting request ri

Example 1: Scheduler

Solution

- No winning strategy for both Controller and Scheduler
- Scheduler can constantly play false without scheduling any event
- Controller can keep on sending requests and Scheduler won’t be able to satisfy the requirements

Example 1: Scheduler

Solution

- Converting into 3-player safety game
- Each state having parameters (a1, a2, r1, r2, k1, k2, q1, q2, t1, t2, p)
- Each player can change the state such that he changes his own boolean variables
- Added bad state and found admissible strategies

Example 1: Scheduler

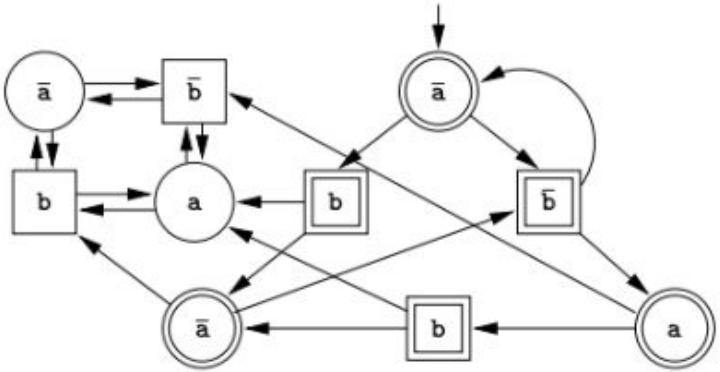
Results

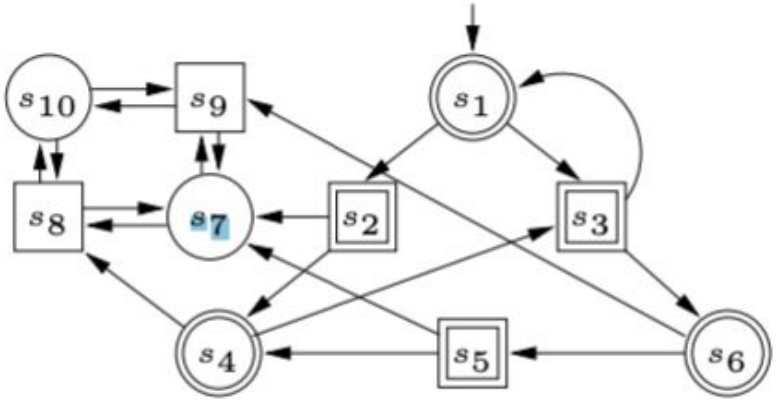
- Nodes in graph: $2^6 * k^2 * t^2 * 3$
- High run time associated with finding the admissible strategies (k=4, t=2 -> 4 hours)
- In example of k=4, t=2, we are at node (1, 1, 0, 0, 4, 4, 0, 0, 2, 2, 0), we can go to (0, 1, 1, 0, 3, 3, 0, 0, 2, 2, 0) and (1, 1, 0, 0, 3, 3, 0, 0, 2, 2, 0).
- Second transition will get dominated by the first one

Example 2: Delivery System

- An example of a delivery robot.
- Let a and b be 2 signals.
- a → Burger has to be kept on a table, b → Robot out of delivery
- $\square(a \rightarrow \bigcirc b) \bigwedge \square(b \rightarrow \bigcirc \neg b)$

Example 2: Delivery System



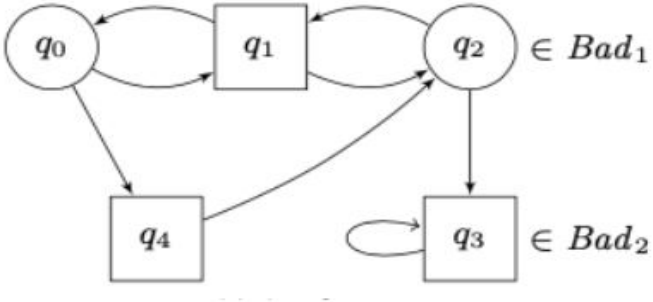


Example 2: Delivery System

Solution

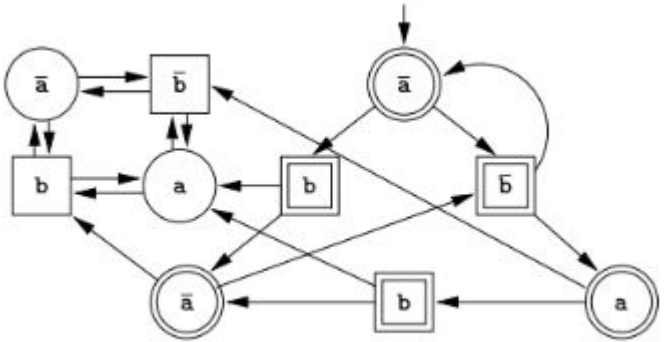
- Represented the problem as a 2 player graph game.
- The controller at desk or the robot can individually lose. Edges like s4 → s8 or s5 → s7 may lead to a loss.

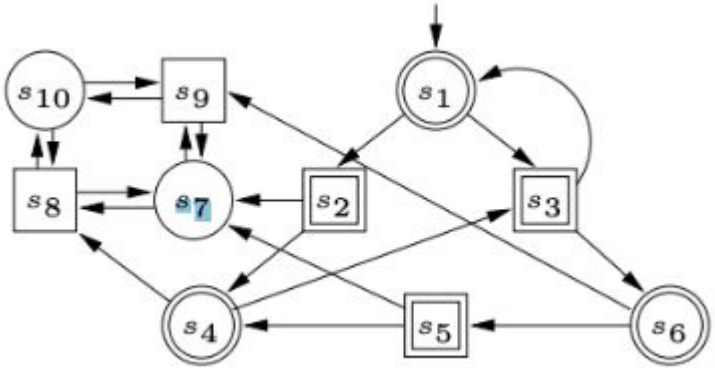
Code Demonstration to find admissible strategies



Example 2: Delivery System

```
src — -bash — 80×24
[users-MacBook-Air:src mayankwadhwani$ python3 robot_delivery.py
The set of dominated strategies:
4 -> 8
6 -> 9
2 -> 7
5 -> 7
users-MacBook-Air:src mayankwadhwani$
```





04

CONCLUSION AND FUTURE WORK

CONCLUSION AND FUTURE WORK

- Make use of admissible strategies in real life applications such as the Scheduler and the Robot Delivery System.
- Extend our work to support more objectives (like Buchi or Muller)
- Apply this on more complex examples!

ACKNOWLEDGEMENT

“

We would like to express our gratitude to our supervisor Dr. Purandar Bhaduri for giving us the opportunity to explore this new field of Compositional Synthesis.

THANK YOU

We are open to questions, if any.