

COMP424/524-06A Topics in Software Engineering

Part I – Finite State Machines

6. Controller Synthesis

Robi Malik

THE UNIVERSITY OF
WAIKATO DEPARTMENT OF COMPUTER SCIENCE
TARI ROROHIKO

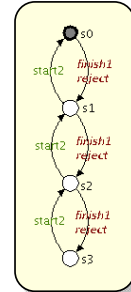
Application-Specific Properties

Question:

Does the transfer line model ensure that **Buffer 1** never has an overflow or underflow?

Answer:

Use **property automata** and **Language Inclusion Check**.



© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 4

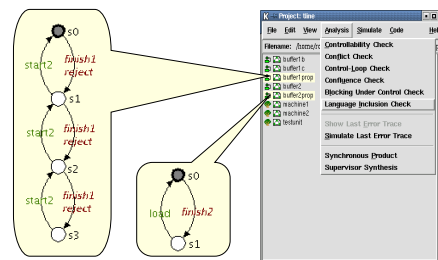
Universal Properties

Controllability and Nonblocking are

- Universal properties
- General consistency checks
- Application-independent

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 2

Language Inclusion Check



© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 5

Does it Really Work?

How to check that the system really does what we want it to do?

- Simulation
- Application-specific properties

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 3

Language Inclusion Check

Definition:

Let A and B be automata. The behaviour of A is **included** in the behaviour of B if

$$\mathcal{L}(A) \subseteq \mathcal{L}(B) .$$

VALID checks whether:

$$\mathcal{L}(A_1 \parallel \dots \parallel A_n) \subseteq \mathcal{L}(B_1 \parallel \dots \parallel B_m) .$$

Selection

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 6

Language Inclusion Check

Does the system behaviour always remain within the constraints given by a property automaton?

Can start2 happen in this state?

Can finish1 or reject happen in this state?

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 7

Plants for Cat and Mouse

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 10

A Maze for Cat and Mouse

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 8

Controlling Cat and Mouse

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 11

Specification for Cat and Mouse

Environment

- 5 rooms, 13 doors
- Each door passed only by cat or mouse
- Some doors can be shut by controller

Desired behaviour

- Cat and mouse never in the same room
- Can always return to their starting point

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 9

The Easy Way ...

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 12

Supervisor Synthesis

Automatically finds a new specification which

- ... is controllable,
- ... is nonconflicting,
- ... restricts all given specifications,
- ... is as general as possible.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 13

Events for Tic-Tac-Toe

- ✕ **black.x.y**
Black moves on field (x,y).
(controllable)
- **white.x.y**
White moves on field (x,y).
(uncontrollable)

- black.0.0
- white.0.0
- black.0.1
- white.0.1
- black.0.2
- white.0.2
- ...

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 16

Playing Tic-Tac-Toe

○	○	
✕	○	
		✕

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 14

Modelling the Game

white: 0.0
white: 0.1
white: 0.2
white: 1.0
white: 1.1
white: 1.2
white: 2.0
white: 2.1
white: 2.2

white — black

move

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 17

Model of Tic-Tac-Toe

Model

- Two players
 - white (○) to move first
 - black (✕) to move second
- Nine fields indexed by $x = 0..2, y = 0..2$

Control Objective

- Play for black so that you will never lose.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 15

Making a Row

white: 0.0 white: 0.0 white: 0.0
white: 1.1 white: 1.1 white: 1.1
white: 2.2 white: 2.2 white: 2.2

s0 → s1 → s2 → s3 winner_white_d1

white_dwin1

New event:
"White wins by
completing diagonal 1"

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 18

When the Game Ends

```

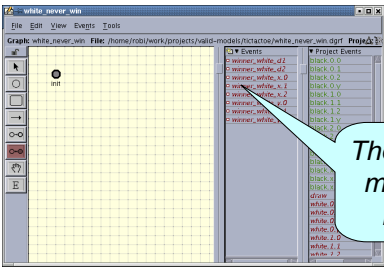
winner_black_d1
winner_black_d2
winner_black_x.0
winner_black_x.1
winner_black_x.2
winner_black_y.0
winner_black_y.1
winner_black_y.2
winner_white_d1
winner_white_d2
winner_white_x.0
winner_white_x.1
winner_white_x.2
winner_white_y.0
winner_white_y.1
winner_white_y.2
draw
game --> over
game_over

```

Marked state:
To be nonblocking, the controller must allow the game to end in some way.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 19

And Finally the Control Objective



These events must never happen.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 20


Solving Games

We have used supervisor synthesis to ...

- generate a strategy for Tic-Tac-Toe,
- show that Tic-Tac-Toe is a “fair” game.

Challenge

- Can you do it for more complex games?



© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-6 Slide 21