

DIG Bachelor Study
and CS-for-all

Problem Solving and Algorithmic Thinking

2UE

Lecturers:


Sebastian Forster and Ana Sokolova
Department of Computer Sciences

Classes Schedule

- **Classes** Monday 12:45 pm - 2:15 pm (Group D) and Monday 4pm - 5:30pm (Group E) in **SRI.33**
- **Tutors** tba
- **Blackboard** communication, assignments, announcements, and grades
PlusOnline communication (emails)

Classes Schedule

- **Classes** Monday 12:45 pm - 2:15 pm (Group D) and Monday 4pm - 5:30pm (Group E) in **SRI.33**
- **Tutors** tba
- **Blackboard** communication, assignments, announcements, and grades
PlusOnline communication (emails)



starting
next week

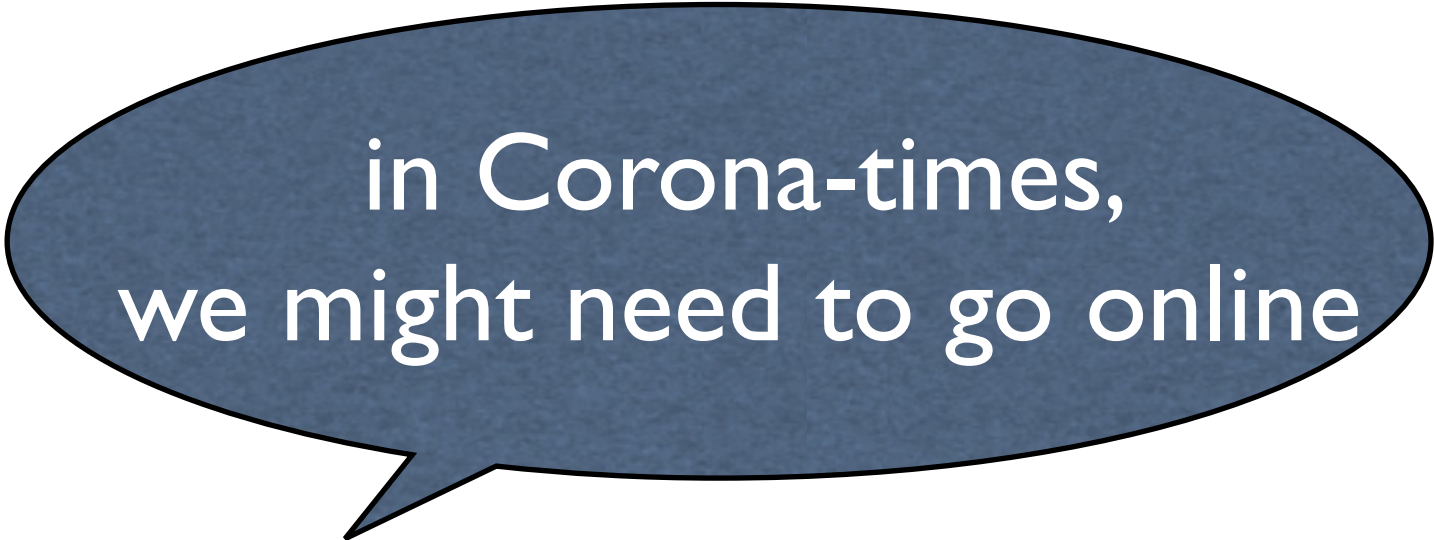
Classes and Grading

- Each week we will cover an important topic in algorithms, complexity, CS in general, by letting you solve simple example problems in class, in larger groups under our (the lecturers') supervision
- 4 assignments will be given and graded throughout the semester
- The assignments are to be solved in groups of 4 students; 2/3 of the grade determined by the points on the assignments
- One small test towards the end of the semester (quiz of simple exercises) will finally determine the remaining 1/3 grade
- Activity in class may increase your grade for one
- 50% of the total points are sufficient for a positive grade

Presence

- Presence is **obligatory**
- You may miss up to **2 weeks** of classes, for anything beyond that a serious justification is needed

Presence



in Corona-times,
we might need to go online

- Presence is **obligatory**
- You may miss up to **2 weeks** of classes, for anything beyond that a serious justification is needed

Teaching and Studying in Corona Times

- Both challenges and opportunities
- Things may change on short notice
- Be adaptive
- Be considerate

We count on normal classes in presence!
All material and information will be given via Blackboard.

Plan B



In case of further
corona-related problems



Everyone participates online in Webex.

ECTS calculation

- 3 ECTS for the course (UE)

ECTS calculation



1 ECTS = 25
hours

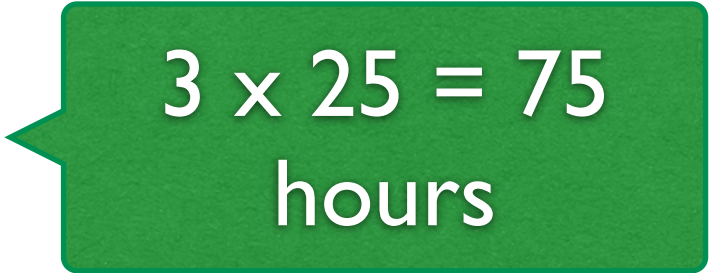
- 3 ECTS for the course (UE)

ECTS calculation



1 ECTS = 25
hours

- 3 ECTS for the course (UE)



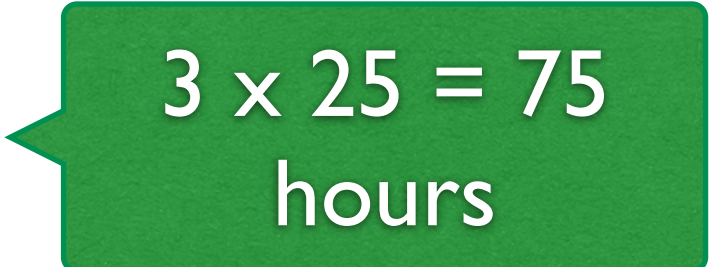
$3 \times 25 = 75$
hours

ECTS calculation



1 ECTS = 25
hours

- 3 ECTS for the course (UE)



$3 \times 25 = 75$
hours



in class app. 18 hours

ECTS calculation

1 ECTS = 25
hours

- 3 ECTS for the course (UE)

$3 \times 25 = 75$
hours

in class app. 18 hours

remaining 57
hours

ECTS calculation

1 ECTS = 25
hours

- 3 ECTS for the course (UE)

$3 \times 25 = 75$
hours

in class app. 18 hours

remaining 57
hours

app. 4-5 hours
a week :-)

ECTS calculation

1 ECTS = 25
hours

- 3 ECTS for the course (UE)

$3 \times 25 = 75$
hours

in class app. 18 hours

remaining 57
hours

or : 8 hours per assignment +
25 hours test preparation

app. 4-5 hours
a week :-)

Goals:

Goals:



Goals:



Image source: Springer



Goals:

Goals:

Throughout the course,
you will learn to:

Goals:

Throughout the course,
you will learn to:

- Think algorithmically with the help of suitably chosen examples
- Intuitively understand fundamental computation costs as well as the limits of computability
- Participate in goal-oriented team work

Goals:

Throughout the course,
you will learn to:

and solve
problems

- Think algorithmically with the help of suitably chosen examples
- Intuitively understand fundamental computation costs as well as the limits of computability
- Participate in goal-oriented team work

Topics and Schedule:

11.10. Representing Information

18.10. Rewriting

Ana Sokolova

8.11. Normal Algorithms

15.11. Sorting I

Sebastian Forster

22.11. Sorting II

29.11. Graph Algorithms I

Sebastian Forster

6.12. Graph Algorithms II

13.12. Finite Automata

20.12. Test

Ana Sokolova

10.1. Turing Machines

17.1. Backtracking

Sebastian Forster

24.1. Reductions / Computability

Ana Sokolova

Topics and Schedule:

11.10. Representing Information

18.10. Rewriting

Assignment 1

Ana Sokolova

8.11. Normal Algorithms

15.11. Sorting I

Assignment 2

Sebastian Forster

22.11. Sorting II

29.11. Graph Algorithms I

Assignment 3

Sebastian Forster

6.12. Graph Algorithms II

13.12. Finite Automata

20.12. Test

Ana Sokolova

10.1. Turing Machines

Assignment 4

17.1. Backtracking

Sebastian Forster

24.1. Reductions / Computability

Ana Sokolova

Let's start with a
puzzle...

The river-crossing puzzle

[Hopcroft et al, Kastens et al]

The river-crossing puzzle

- A man stands with a wolf, a goat, and a cabbage at the left bank of a river, that he wants to cross.
- The man has a boat that is large enough to carry him and another object to the other side.
- If the man leaves the wolf and the goat, or the goat and the cabbage on one side without supervision, one of them will get eaten :-)
- Is it possible to cross the river so that neither the goat nor the cabbage is eaten?

The river-crossing puzzle

- A man stands with a wolf, a goat, and a cabbage at the left bank of a river, that he wants to cross.
- The man has a boat that is large enough to carry him and another object to the other side.
- If the man leaves the wolf and the goat, or the goat and the cabbage on one side without supervision, one of them will get eaten :-)
- Is it possible to cross the river so that neither the goat nor the cabbage is eaten?

The river-crossing puzzle

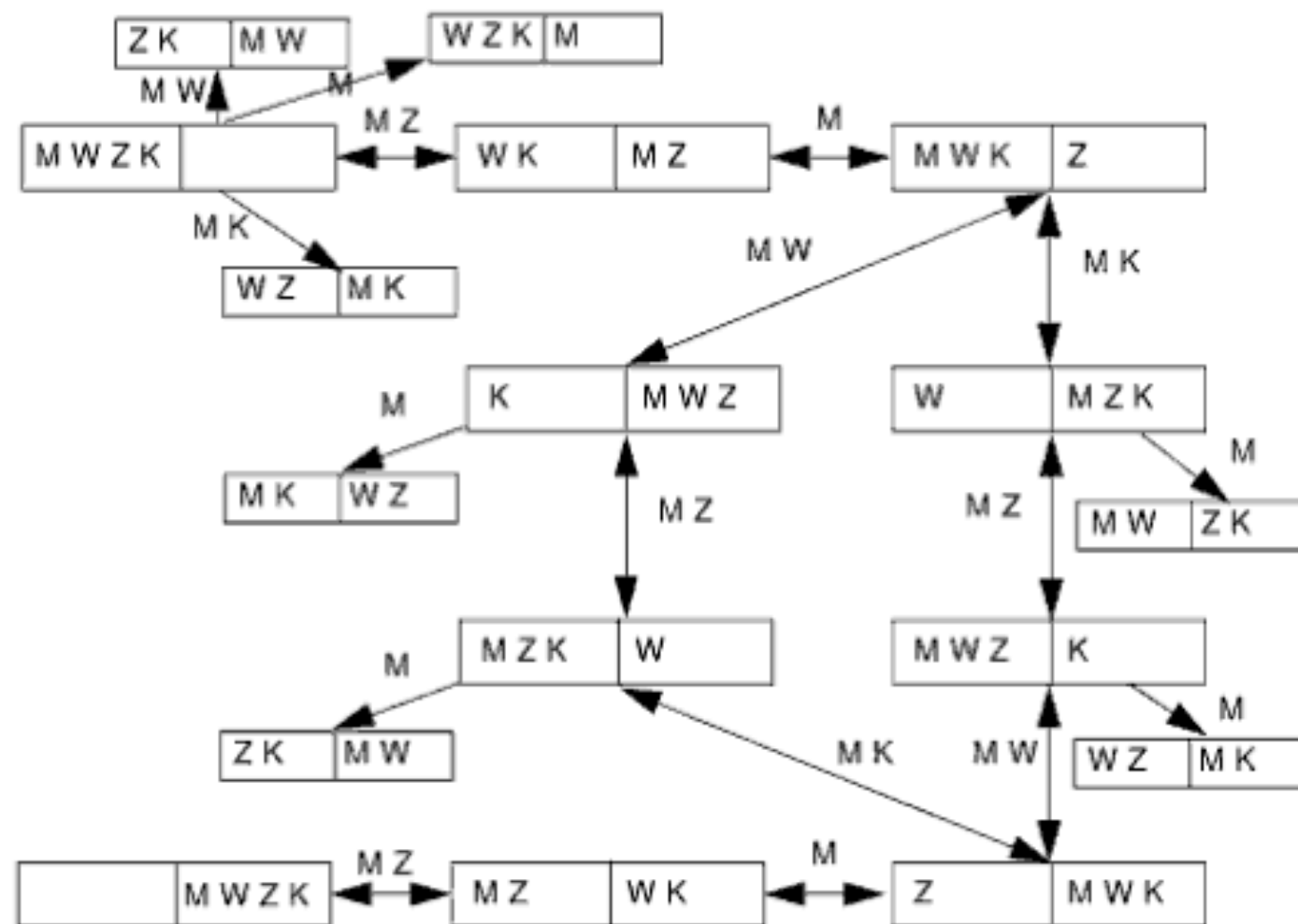
- A man stands with a wolf, a goat, and a cabbage at the left bank of a river, that he wants to cross.
- The man has a boat that is large enough to carry him and another object to the other side.
- If the man leaves the wolf and the goat, or the goat and the cabbage on one side without supervision, one of them will get eaten :-)
- Is it possible to cross the river so that neither the goat nor the cabbage is eaten?

The river-crossing puzzle

- A man stands with a wolf, a goat, and a cabbage at the left bank of a river, that he wants to cross.
- The man has a boat that is large enough to carry him and another object to the other side.
- If the man leaves the wolf and the goat, or the goat and the cabbage on one side without supervision, one of them will get eaten :-)
- Is it possible to cross the river so that neither the goat nor the cabbage is eaten?

The river-crossing puzzle

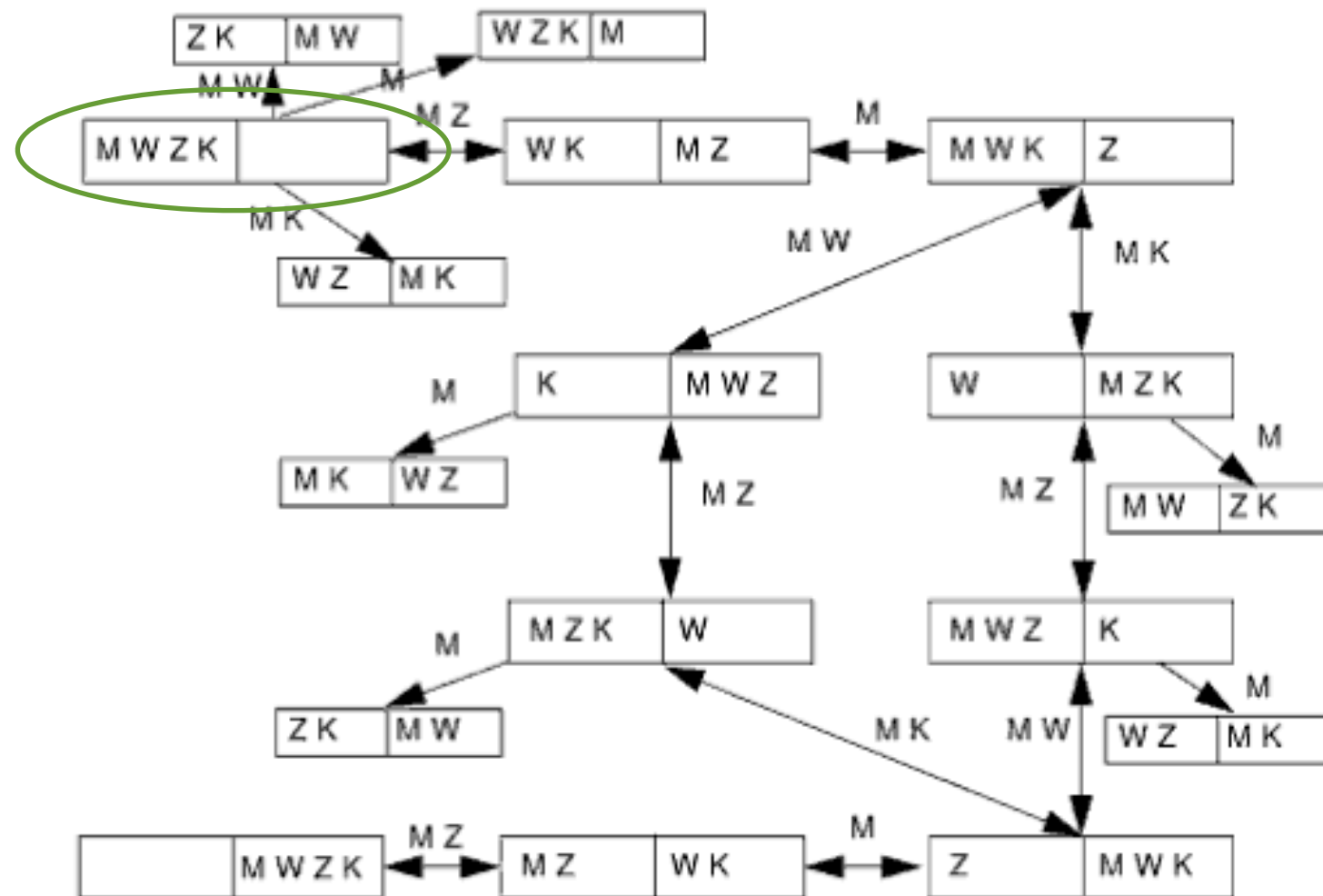
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

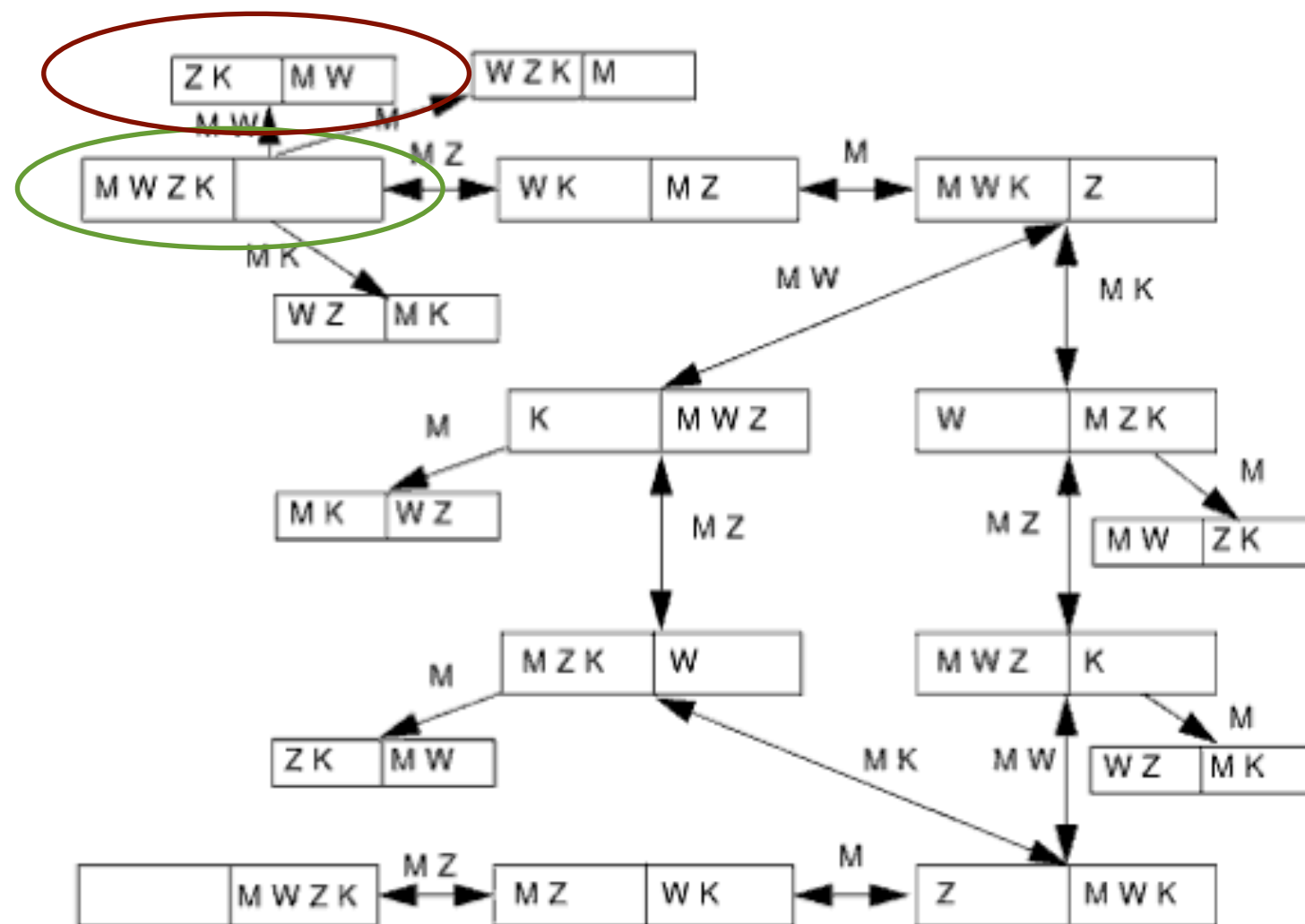
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

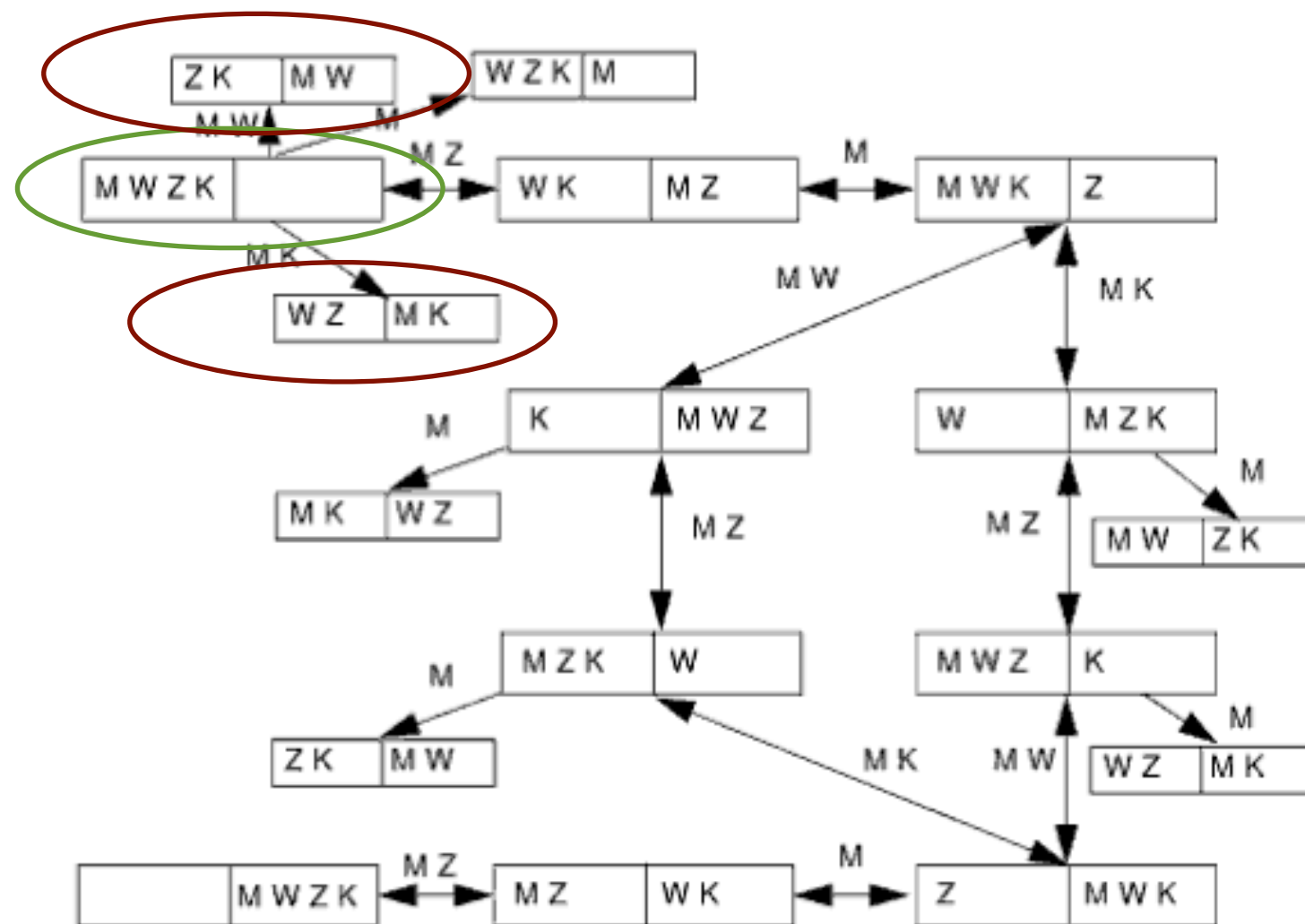
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

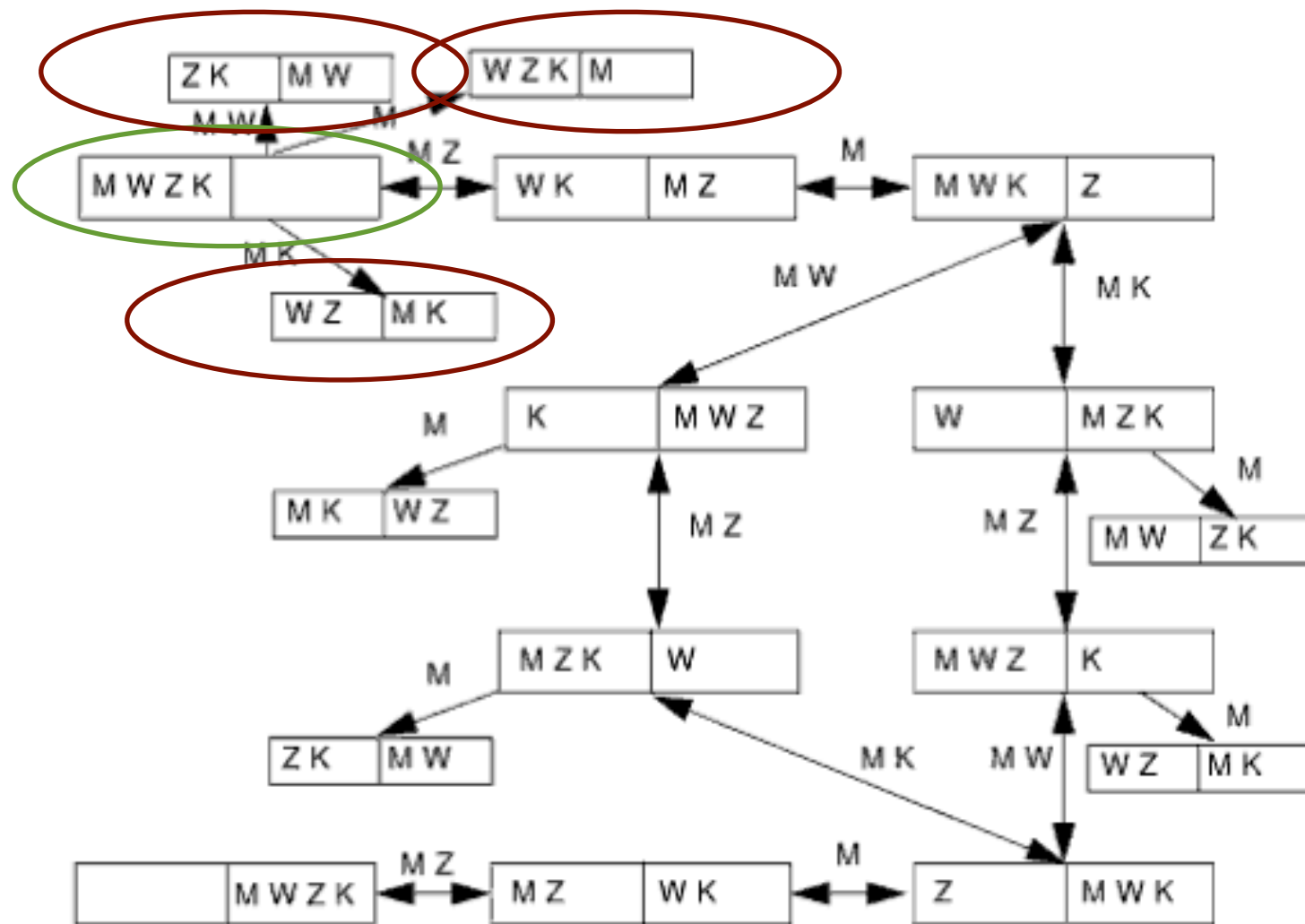
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

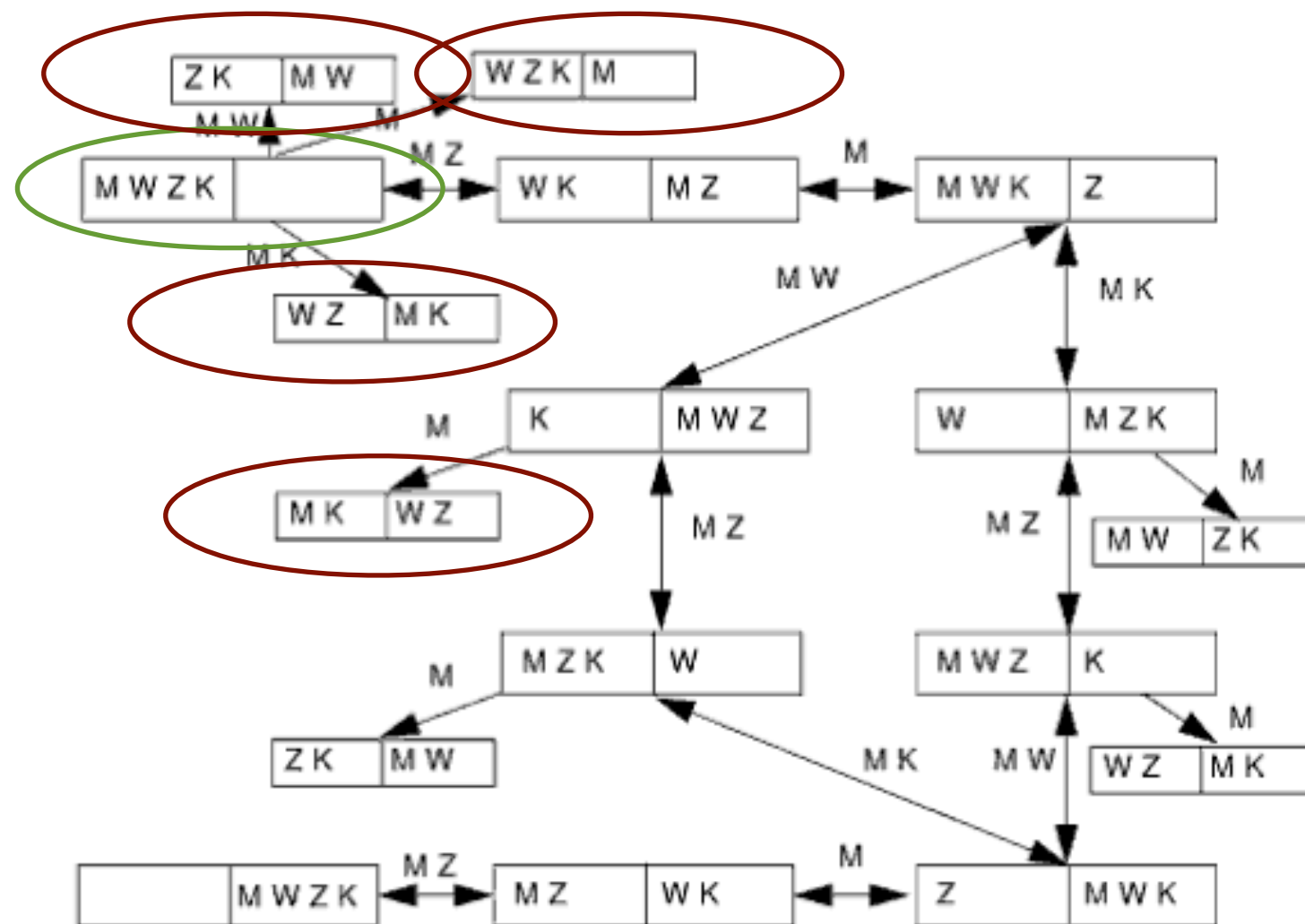
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

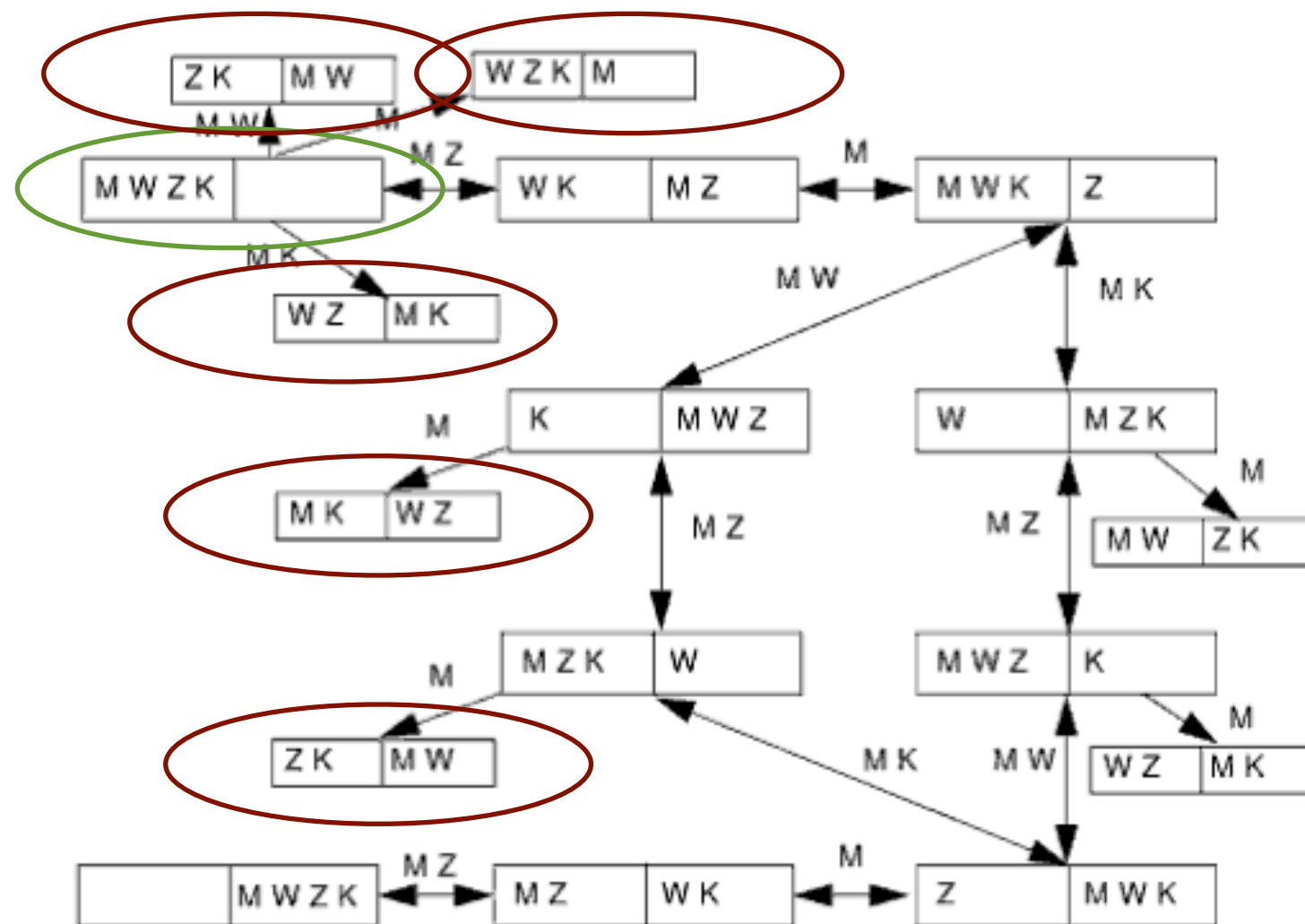
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

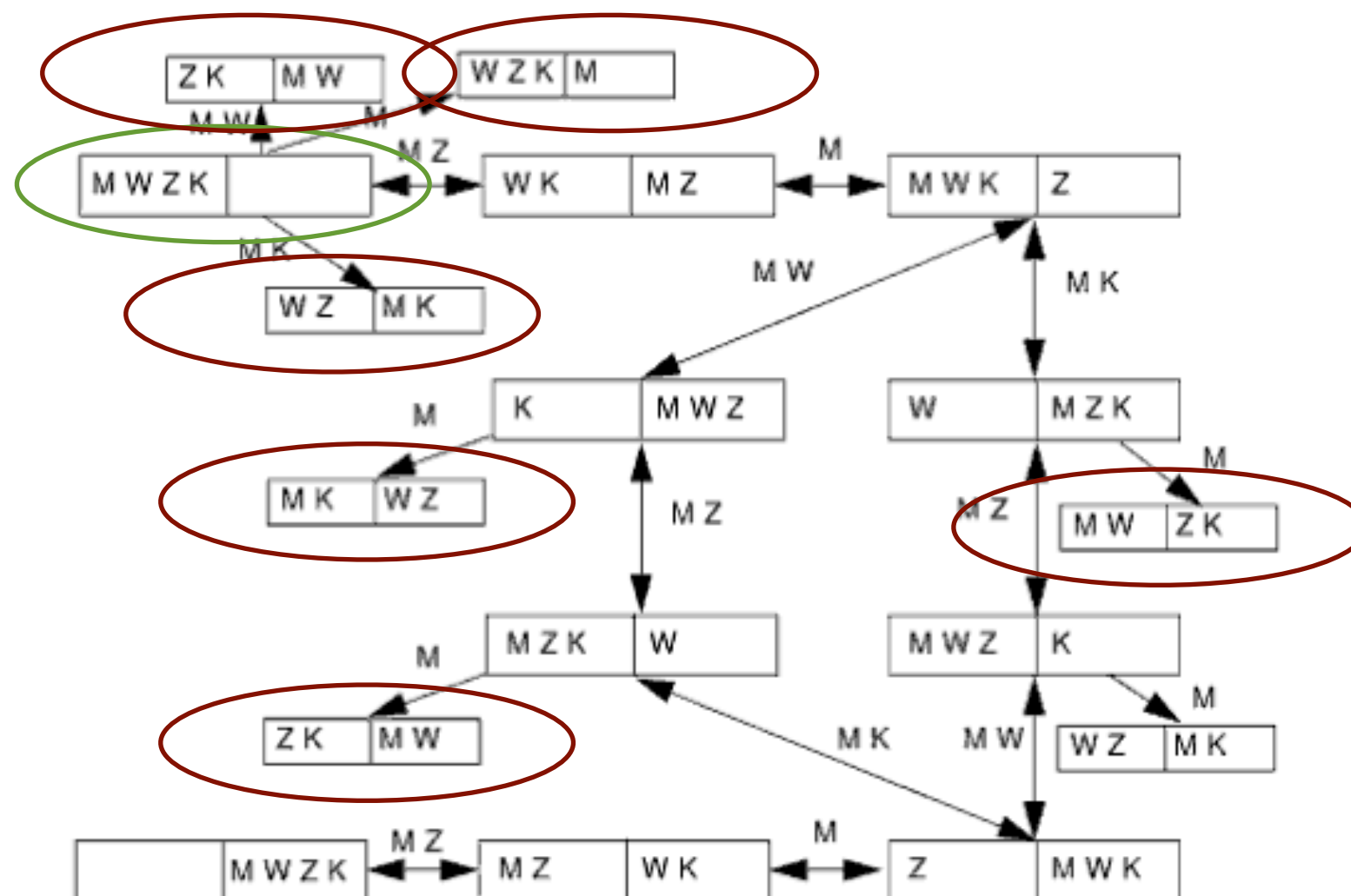
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

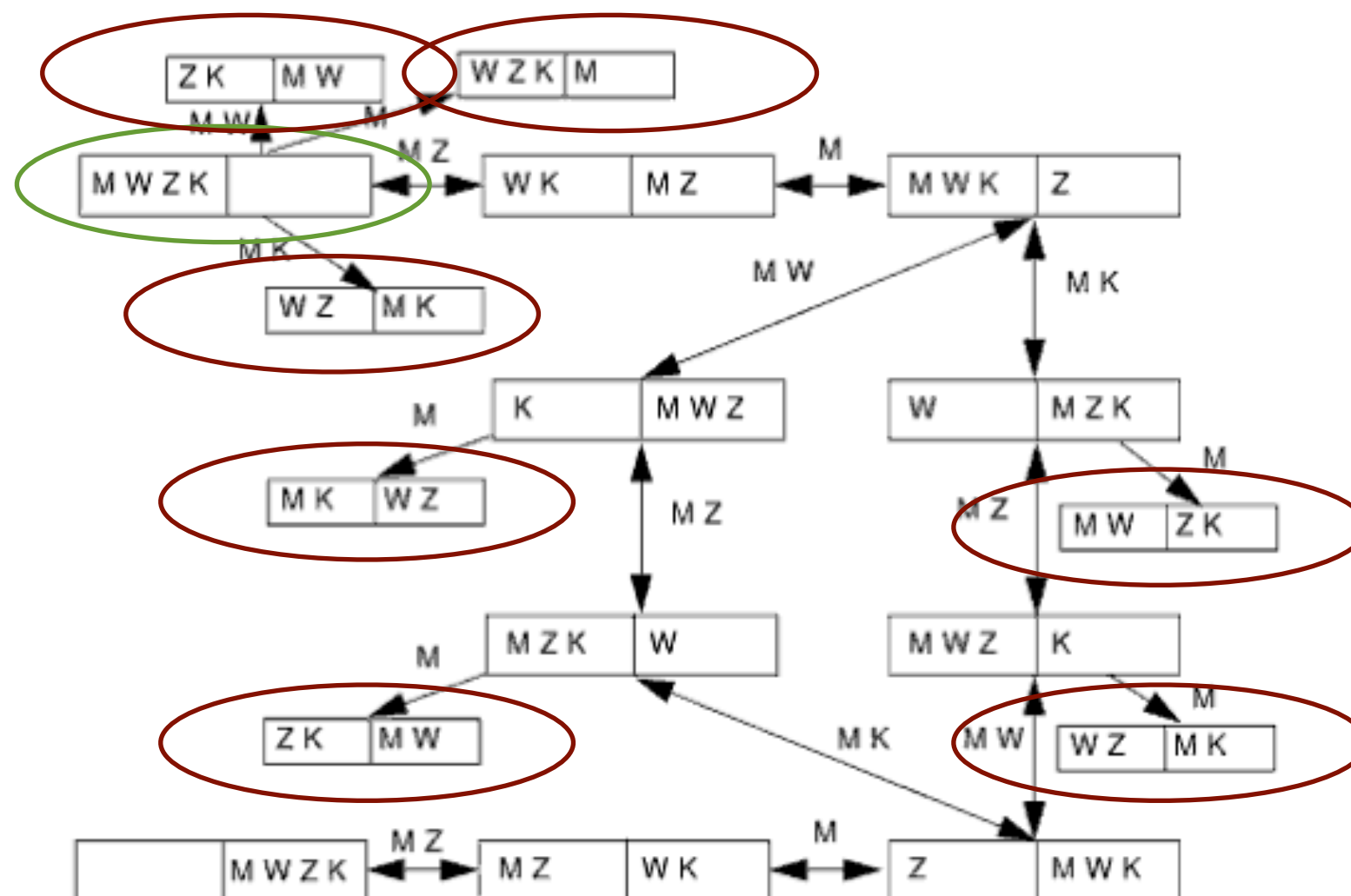
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

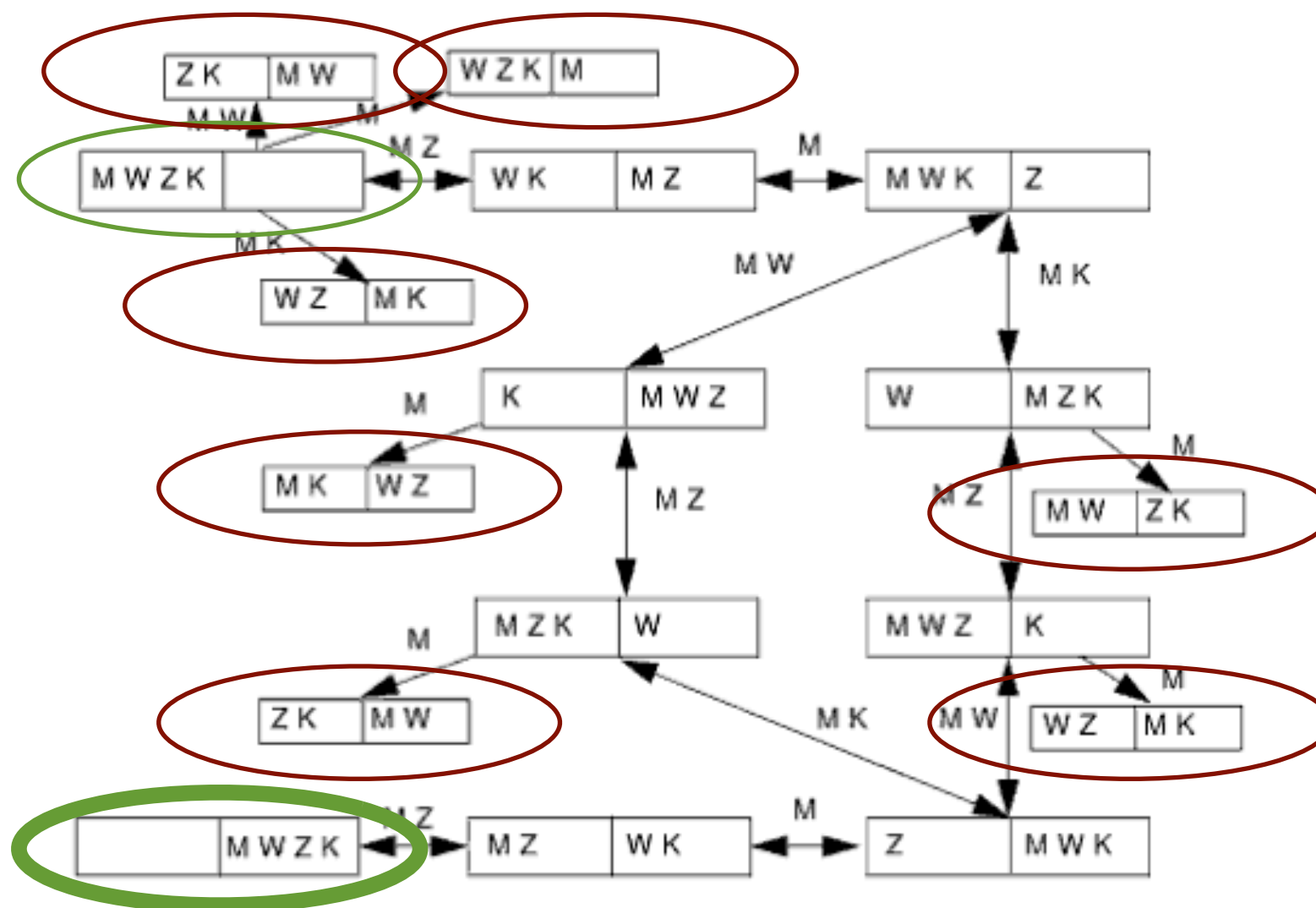
Formalization with a finite automaton [Kastens et al.] :



states and transitions

The river-crossing puzzle

Formalization with a finite automaton [Kastens et al.] :



states and transitions