Projects

11. Reversible Logic Lattice
12. Dog's World
13. Fairy Tale Middle Age world with sexual reproduction
14. Robot head - emotions. Language of emotions
15. Tick-tack-toe
16. Robot - Teacher of English
17. Family relations
18. Bridges of Koenigsberg
19. Electricity, gas, water and sewer
20. Theorem Proving in Planar Geometry

Project #11

Reversible Lattice Structure

Reversible Lattice Structure

- Multi-input, multi-output function can be realized in a “binary lattice” structure. Several papers on lattices by Marek Perkowski and others are available on his web-page. There are also papers by Dr. Jeske on her web-page.
- This concept has been generalized to binary lattice using reversible logic. This is explained in ECE 472 class materials under Reversible Logic.
- Reversible lattices are build in two or three dimensions.
- It is sufficient if you will write a Lisp program for two-dimensions. **Writing for three dimensions will give you additional credit.**
- Assume that your initial function to be realized in Lattice is a Lisp expression with operators OR, AND, NOT and EXOR and variables.
- Write a program that will realize arbitrary SINGLE-OUTPUT function in a 2-dimensional reversible lattice structure.
- First analyze the case of symmetric functions only

Project #12

Dog’s world

This is similar to “turtle” and “ants worlds” Dogs and owners are agents living in the housing area with a park

- Dogs like to be walked by their owners.
- Dogs follow path left them by other dogs using their smell sensor.
- Dogs pay attention to other dogs, for instance they want to mate or to fight, or exchange friendly greetings by waging tails.
- In the housing district (draw its map) the owners walk their dogs. Each starts from his house at random time and returns there. Their paths can be random or non-random.
- Simulate what can happen when there are several owners with dogs.
- The dogs are initially on leashes. The length of a leash is a parameter. Many data should be paremetrized and use probabilities.
- I have 20 years old experience with my dogs so I will be able to evaluate the realism of your simulation.
- **You can re-use any code from literature, this class or WWW. The same in any other program assigned in this exam.**

Project #13

Fairy Tale Middle Age world with sexual reproduction

- This program belongs to Turtle-like World simulations.
- Read the article by Marek Perkowski presented at LDL'2000 symposium about Oregon Cyber Theatre
- Create a map of a fairy tale Middle Age land with three countries, rivers and forests.
- In each country there are N couples, each has a male and a female. The male has a sword. The female can get pregnant and takes care of babies. After K "*years*" the child becomes and adult - the female can have children, the male can fight.Genetic algorithm is used to create the genotype of the offspring. You can take it from my WWW page.
- When male and female meet they can mate with probability P and they have a random R= from one to 4 children. Sex of children is also random with probability M of a male.
- If the male mates with not his female, the female's husband fights with him. He learns about her infidelity from other people. In case of fight the person dies with probability U

- Males can create coalitions to fight against other. Their probability of winning is respectively increased.
- If they win, they have more land. If they have more land, more children will survive. Otherwise most children dies in infancy.
- Add more random parameters to this world. It needs some simple graphics for males, females, children, countries, rivers, etc. Each country can have a different color so you see how they change with time as a result of wars.
- Create a very simple language with which they communicate. It should have not more than 10 sentences in which, however, there can be arbitrary numbers.
- Persons can talk only if they are in a small distance.
- Everybody knows the whole map, locations of homes of everybody else.
- Run the simulation with different values of parameters.
- There is an amazingly many emergent situations in this simple world.
- By controlling morality of people using few parameters (like the probability of an unfaithful wife or a fight with no reason) you can create completely different scenario of events, leading to wars (remember the Helena of Troy?) and other behaviors.

Project #14

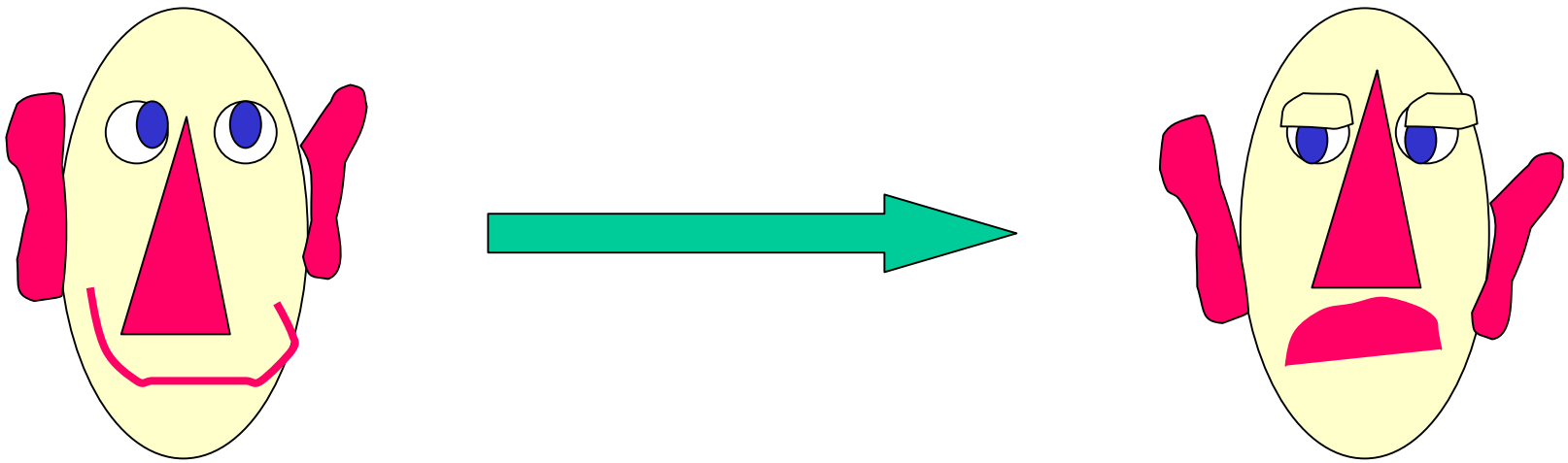
Robot head - emotions.
Language of emotions

For somebody who likes computer graphics

- Assume a schematic human head with eyes, ears, nose, mouth, hair, beard, glasses, etc.
- Each of the above components is an object that has (some or all) of the following properties
  - ***visibility/no-visibility***
  - ***permanent movement (vibration?)***
  - ***shape***
  - ***size***
  - ***angle***
  - ***location***
  - ***color***
- Write a program that will display some emotions on the face by changing gradually the properties of the objects.
- For instance, if the face in in the state HAPPY-SMILE and it goes to the state UNHAPPY-CRIES, the sequence of gradual changes in parameters/properties must be seen, not immediate jump from emotion to emotion.
- Describe the person mood as a stochastic Finite State Machine

For somebody who likes computer graphics

- the results of this program will be useful for head animations.



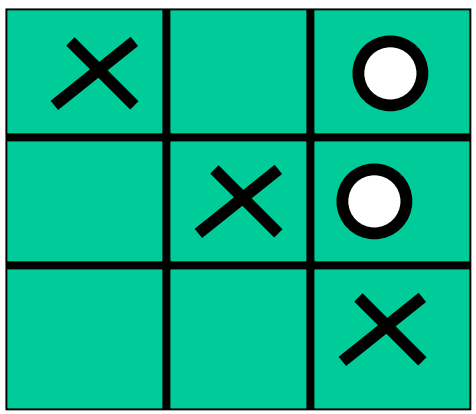
A change from HAPPY-SURPRISED TO UNHAPPY-CONFUSED.

Project #15

Tic-Tac-Toe

Tic-Tac-Toe

- Everybody knows Tic-Tac-Toe
- Write a lisp program to play tic-tac-toe
- Your own choice of data structures, graphics and algorithm.
- The program should always win or draw, newer fail. The program (X) starts first.



Project #16

Robot - Teacher of English

As you remember, there is a program in Lisp that can generate correct (syntactically) sentences.

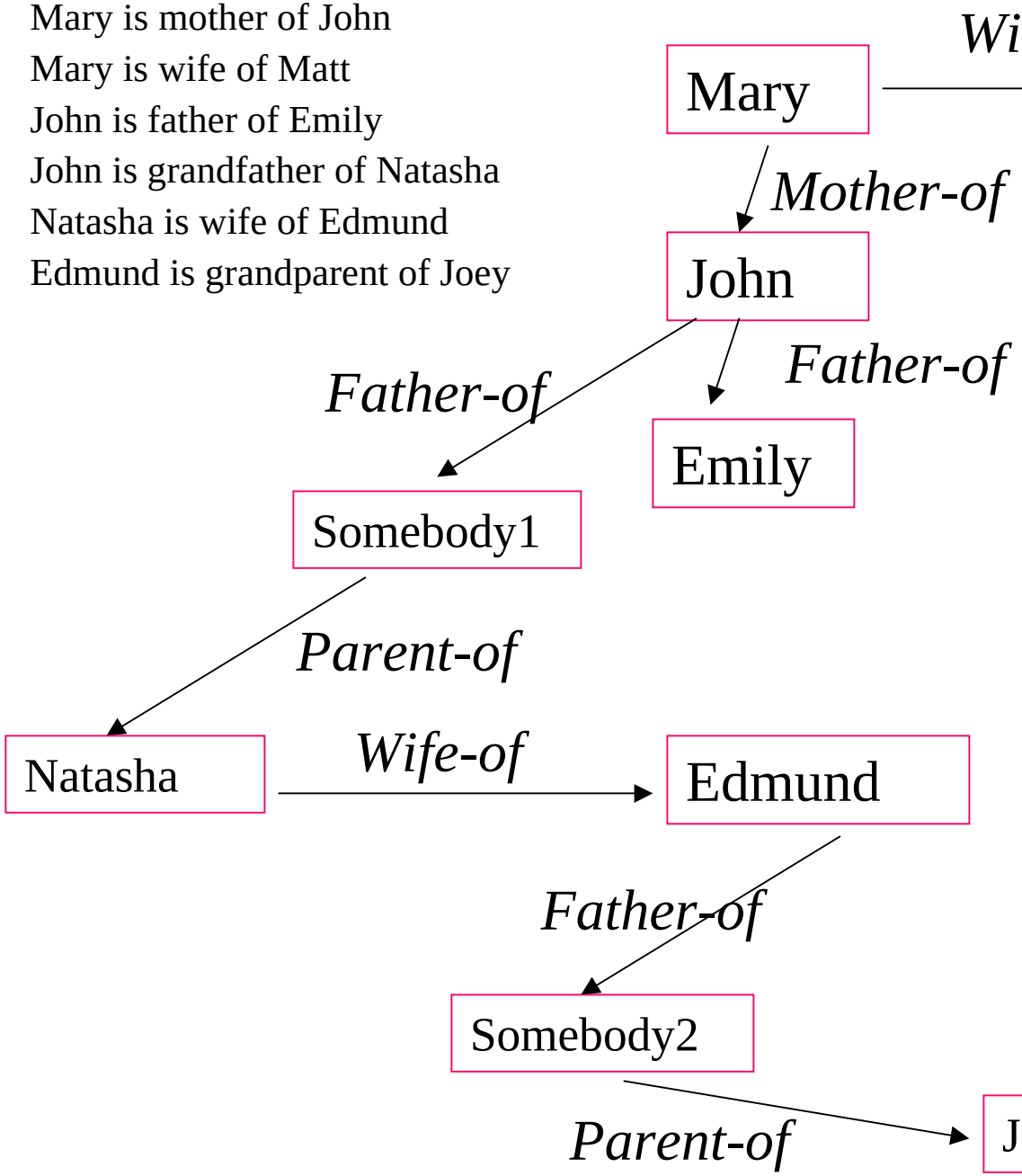
- Write a program (using anything available in literature and WWW and my web-page) that reads a sentence.
- If the sentence is correct, the program shows its grammatical structure like a teacher of English explaining grammar to a child.
- If the sentence is not correct, the program tries to explain what is not correct (like a Pascal compiler). It is left up to you what part of your incorrect sentence the program will actually print and how it will comment it.
- This program will be useful for our robots.
- Do not use the Eliza or Rule-based approaches. You have to use the syntax analyzer from a book or WWW.

Project #17

Family Relations

- Given are relations in a family, such as this
  - Mary is mother of John
  - Mary is wife of Matt
  - John is father of Emily
  - John is grandfather of Natasha
  - Natasha is wife of Edmund
  - Edmund is grandparent of Joey
- Assume only heterosexual families, no extramarital children and no divorces and remarriages (student's suggestion)
- For arbitrary family given as data and any two members of this family, find their family relationship.

- Mary is mother of John
- Mary is wife of Matt
- John is father of Emily
- John is grandfather of Natasha
- Natasha is wife of Edmund
- Edmund is grandparent of Joey



What is a family relation of Matt and Joey?

What is a family relation of Emily and Edmund?

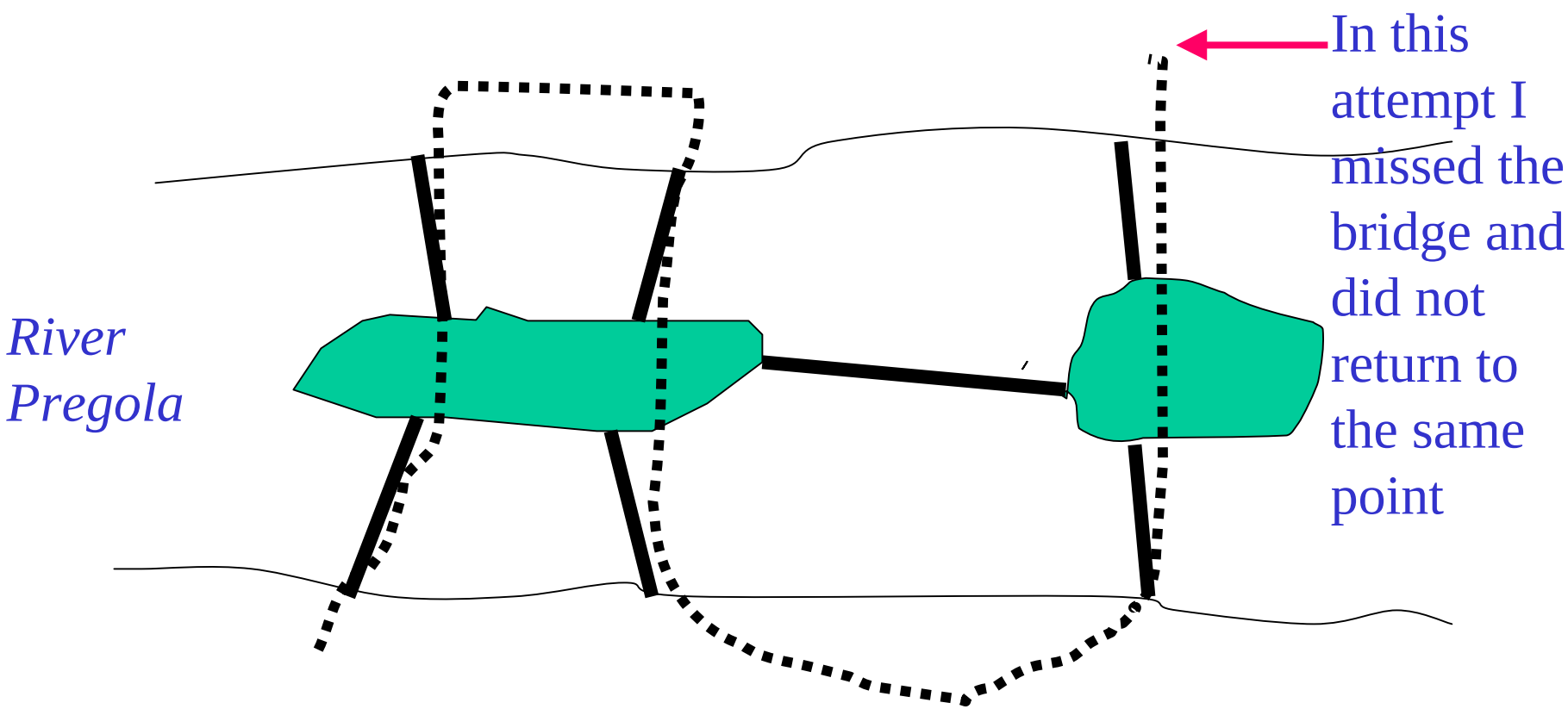
There are at least two corect answers !!

It is possible that Emily = Somebody1

Project #18

The Bridges of Koenigsberg - Euler's Problem

- **Your task is to find a path through all bridges, starting from arbitrary point, go through every bridge only once, do not miss any bridge and go back to the same point in town.**
- **Or prove that there is no solution.**
- **Write the program to prove, do not use Euler's theorem.**



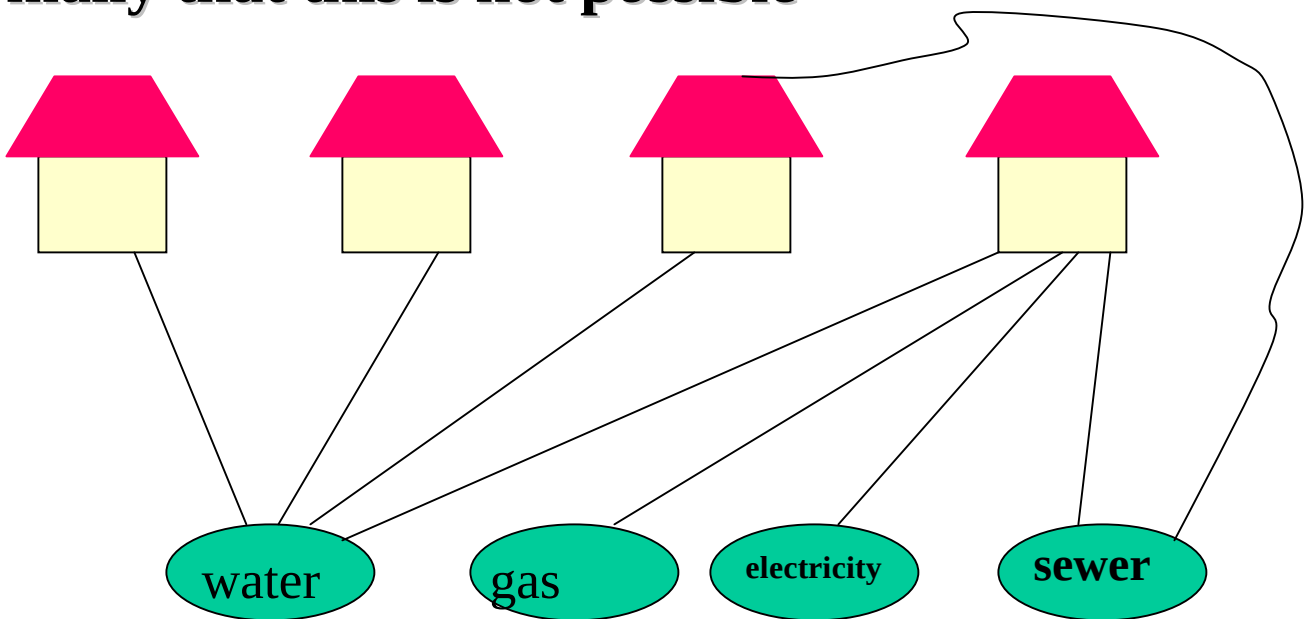
Project #19

Electricity, gas, water and sewer

Electricity, gas, water and sewer

•On the street, given are four sources: electricity, gas, water and sewer.
There are four houses. Each has a sink of electricity , gas, water and sewer.
Try to connect each source on the street with any of the sinks in houses such that no any two lines intersect.
Or prove formally that this is not possible

Try to complete without intersections



street

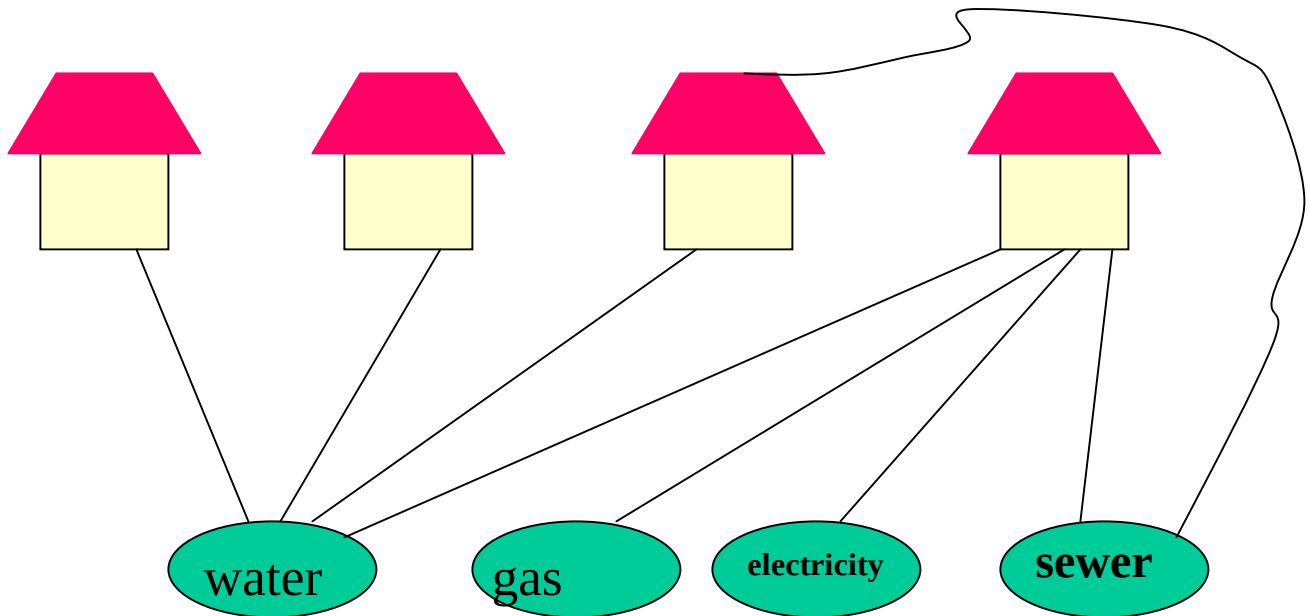
Electricity, gas, water and sewer

I suggest to convert it to a graph coloring problem and run the program exhaustively to prove that there is no solution.

Thanks Jamie Hudson for this problem.

It results directly from Kuratowski's theorem in graph theory that this problem has no solution.

Try to complete without intersections



street

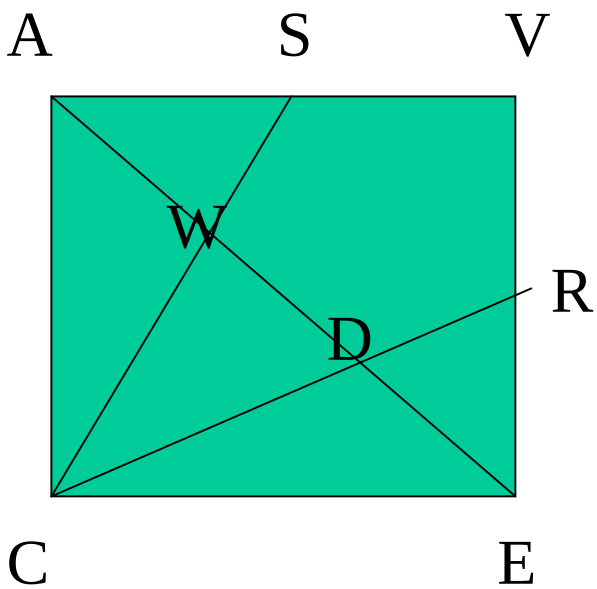
Project #20

Theorem Proving in Planar Geometry

Theorem Proving in Planar Geometry

- Take a high school textbook in planar geometry and find axioms of this system
- Add few more theorems that you can find in the textbook, like Pitagoras or Tales.
- Using resolution mechanism (you can take any theorem-prover that you find on the WWW) prove some theorem that you did not enter and that requires at least three resolution steps to be proven.
- Here is an example of interesting problem that my theorem prover was able to prove.

Theorem Proving in Planar Geometry



Given:
AS=SV=VR=RE=1/2CE

Prove: AW=WD=DE