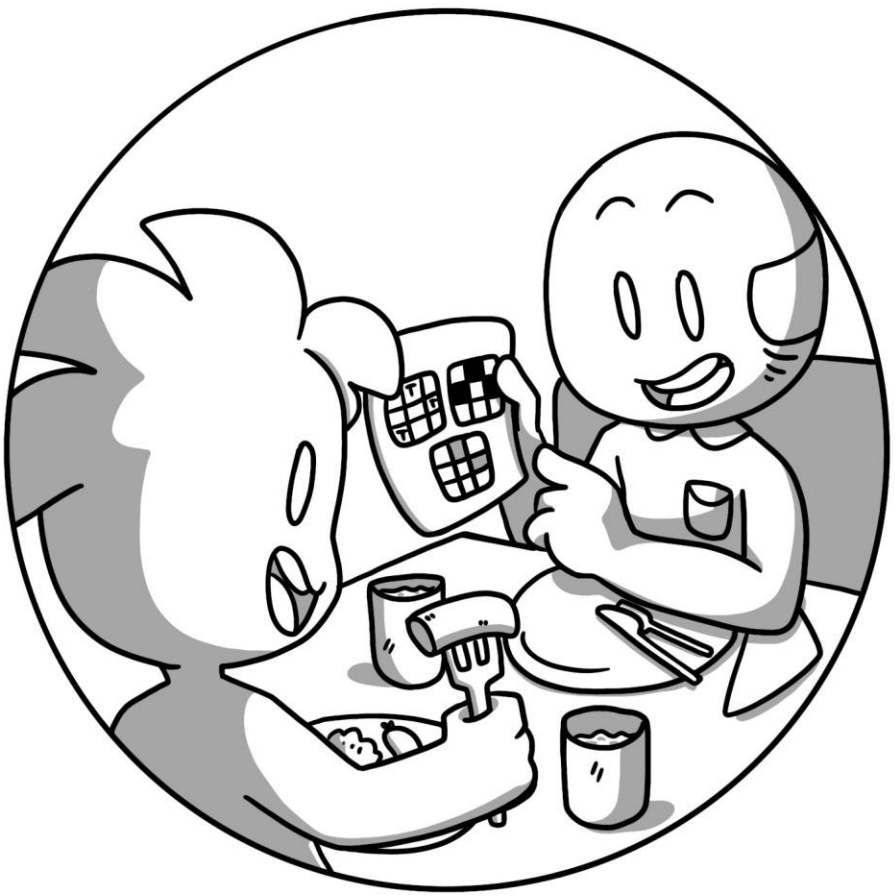
Mathematical Thinking



COMIXPLAIN

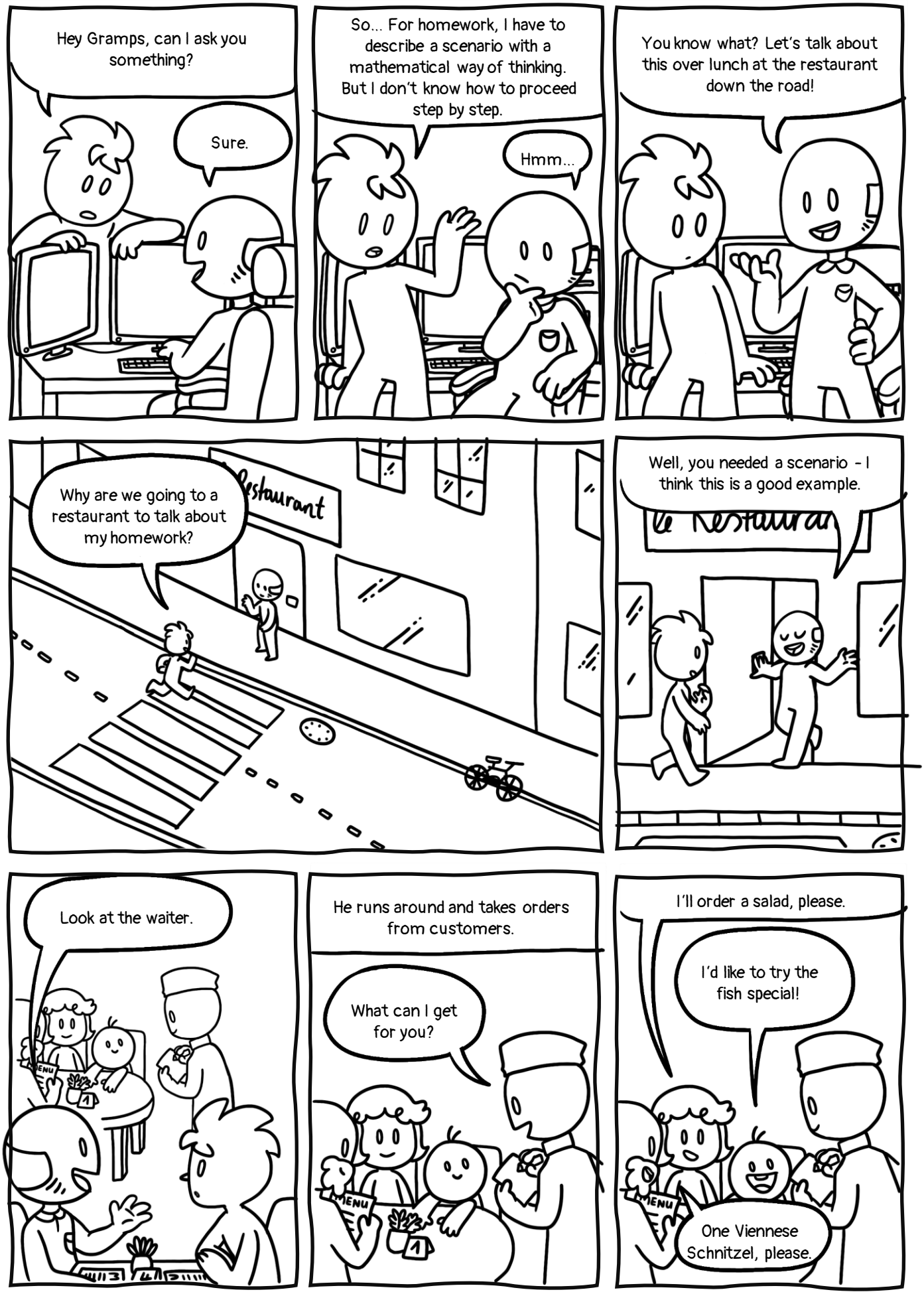
This comic was created in the course of the research project Comixplain, funded by St. Pölten UAS in the course of the Innovation Call 2022.

Project Team:
Victor-Adriel De-Jesus-Oliveira
Hsiang-Yun Wu
Christina Stoiber
Magdalena Boucher
Alena Ertl

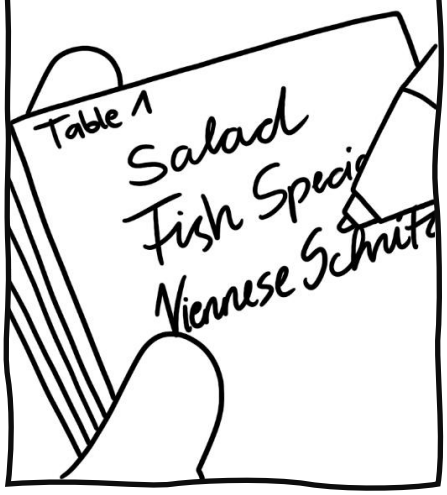
Contact:
victor.oliveira@fhstp.ac.at

Illustrations:
Magdalena Boucher & Alena Ertl

https://fhstp.github.io/comixplain

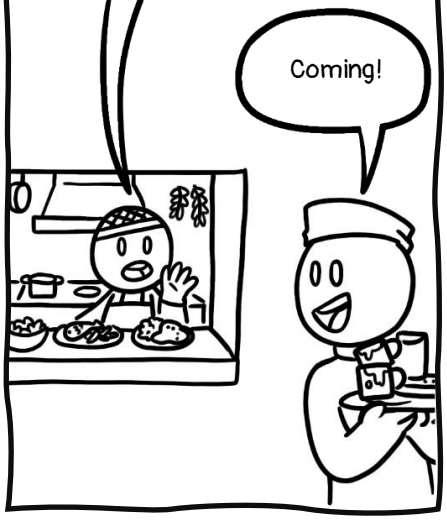

Hey Gramps, can I ask you something?
Sure.
So... For homework, I have to describe a scenario with a mathematical way of thinking. But I don't know how to proceed step by step.
Hmm...
You know what? Let's talk about this over lunch at the restaurant down the road!
Why are we going to a restaurant to talk about my homework?
Restaurant
Well, you needed a scenario - I think this is a good example.
Look at the waiter.
MENU
He runs around and takes orders from customers.
What can I get for you?
MENU
I'll order a salad, please.
I'd like to try the fish special!
MENU
One Viennese Schnitzel, please.

The waiter writes down the table's order...



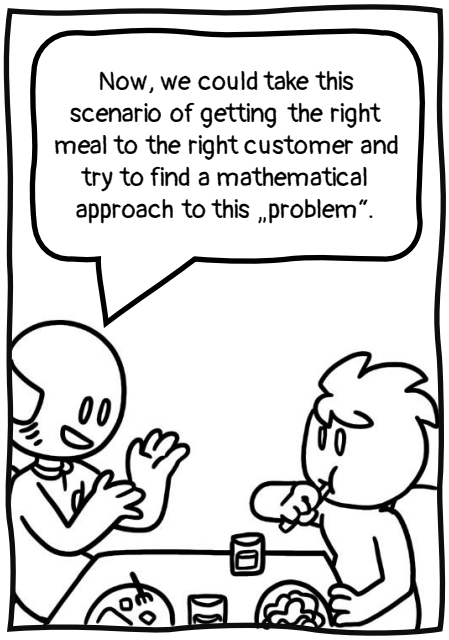
...and continues his work, until...

Food for table one is ready!

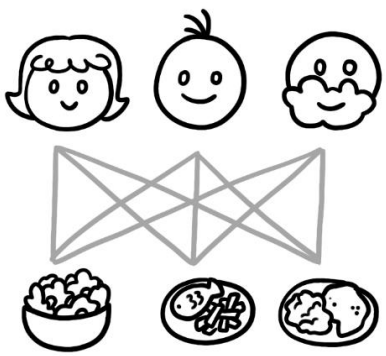


Now the waiter must give each customer the correct meal. Usually, he will ask who gets what.

For the last meal, he doesn't have to ask, because there is only one food and one customer left.

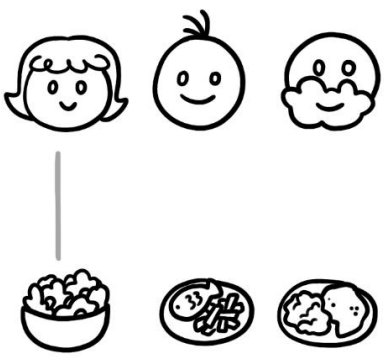


We have three customers and three meals.

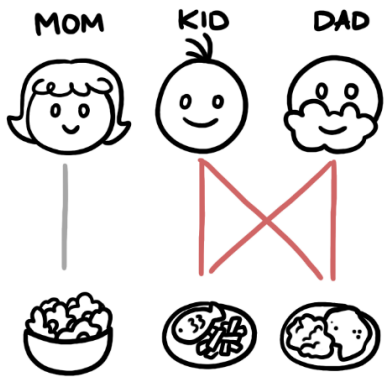


This means that there are nine possible combinations.

What happens as soon as one match is found?

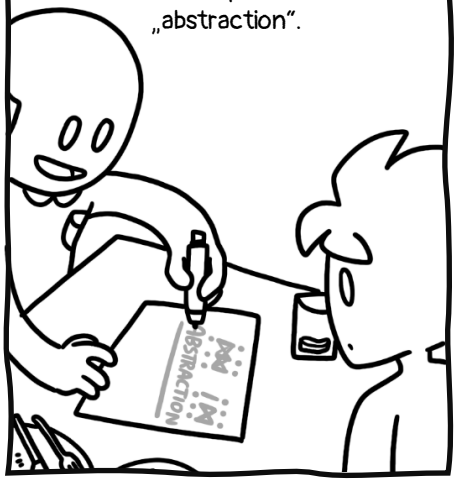


How many possibilities are left?

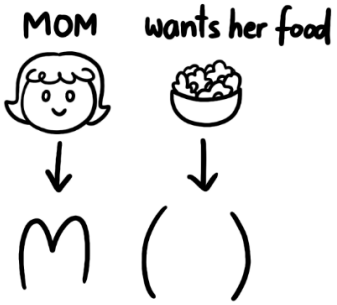


So, as soon as one combination is fixed, the problem becomes easier - there are only four possibilities left.

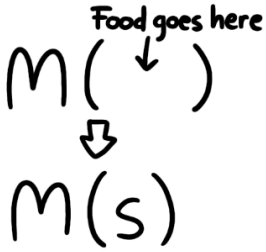
We can also describe this meal-to-customer-problem with mathematical notation. This transformation process is called „abstraction".



Through the abstraction, the problem is described in a way that it can easily – mathematically – be applied to other cases. For example, the mom, who gets the salad, can be abstracted in the following way:

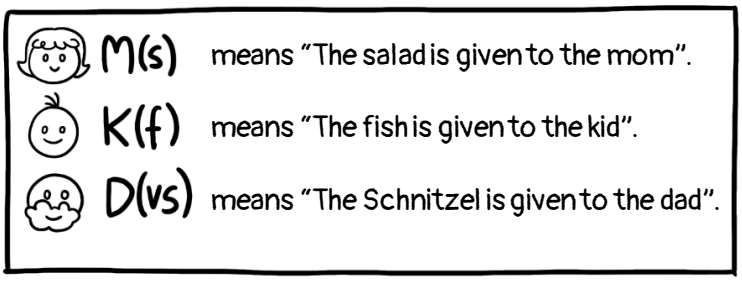


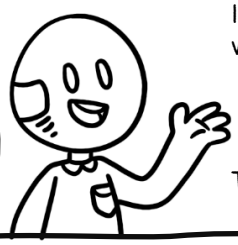
The food for the mom has to be put into the brackets. She gets the salad, so we'll abbreviate that with "s".



M(s) is our mathematical notation for the sentence "The salad is given to the mom".

The same can be done with the other customer-meal-relationships as well:





If the salad is the correct food for the mom, then we can say:

$$M(s) = \text{true}$$

This describes *a fact*.

Mathematicians want to write as little letters as possible, so technically, instead of

$$M(s) = \text{true}$$

we could also write

$$m = \text{true}$$

Or x=true. Or a=true. Anything, really; you could even use an emoji. It's just a variable that represents a value.

We can describe everything the waiter knows at first in a mathematical way.

If each customer can get one meal, the mom can get either:

salad fish schnitzel

In a mathematical notation, this would look like this:

$$s \vee f \vee vs$$

↑ this is the "math symbol" for OR

After having delivered two correct meals, he knows that the dad can't get the salad or the fish, since the following is already on the table:

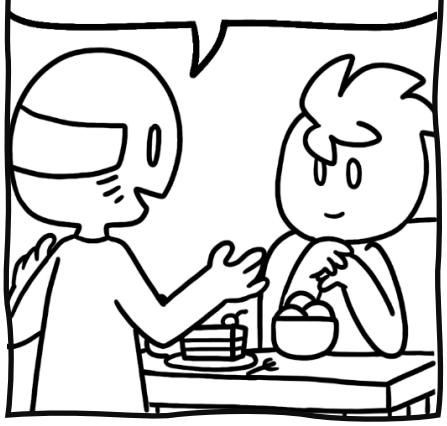
This means that the dad does NOT get the salad AND NOT the fish, which leaves only the schnitzel:

this is the "math symbol" for NOT

$$\neg s \wedge \neg f = vs$$

↑ and this stands for AND

As the waiter, I could evaluate my performance mathematically, too. Basically, I want to check how many people got their food delivered correctly.

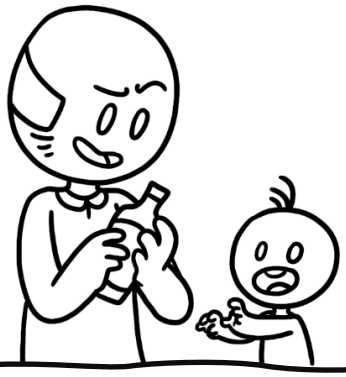


We can do this with a "matching game" of sorts. This table shows all possible combinations of customer and food:

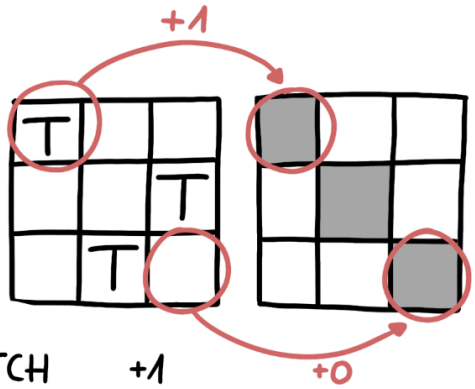
	M	K	D
S			
f			
VS			

...and we have three tags [T] to put in a cell. But a [T] can only appear in one row or column, since every customer can only get one of the three meals.

In different cases, there might even be some restrictions - so-called constraints. For example, a kid can't get any alcohol. In that case, one of the cells would be "blocked".



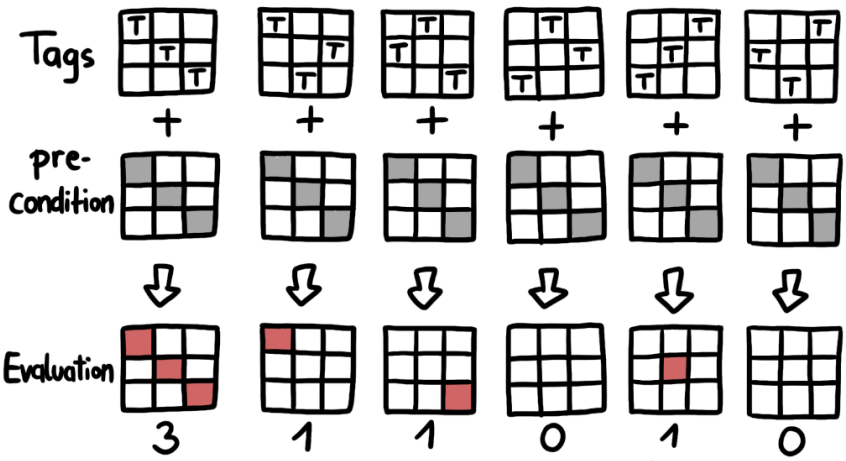
For each meal and customer, we look whether the placement of our [T] matches the precondition. If it does, we get one point, if it doesn't, we don't. And if we get all 3 points, we win!



IF MATCH +1

IF NO MATCH +0

There could be many solutions to our "problem", but only one is correct. The correct solution would be the precondition, and we have to evaluate how our guess fits this precondition.



Oof, I'm full now... but hey, I finished my homework!

Sources:

Ben-Ari, M. (2012). Mathematical logic for computer science. Springer Science & Business Media.

Devlin, K. J. (2012). Introduction to mathematical thinking (Vol. 331). Palo Alto, CA: Keith Devlin.