

# Grundlagen der Wissensverarbeitung

## Blatt 6

Daniel Speck, Lena Niermeyer

21.11.2015

### Exercise 1.1: (Constraints)

Formalize this riddle in the form of a constraint network.

We have the following relations:

$$r1: Y = (D + E) \bmod 10$$

$$r2: E = (N + R + ((D + E) \bmod 10)) \bmod 10$$

$$r3: N = (E + O + (N + R + ((D + E) \bmod 10)) \bmod 10) \bmod 10$$

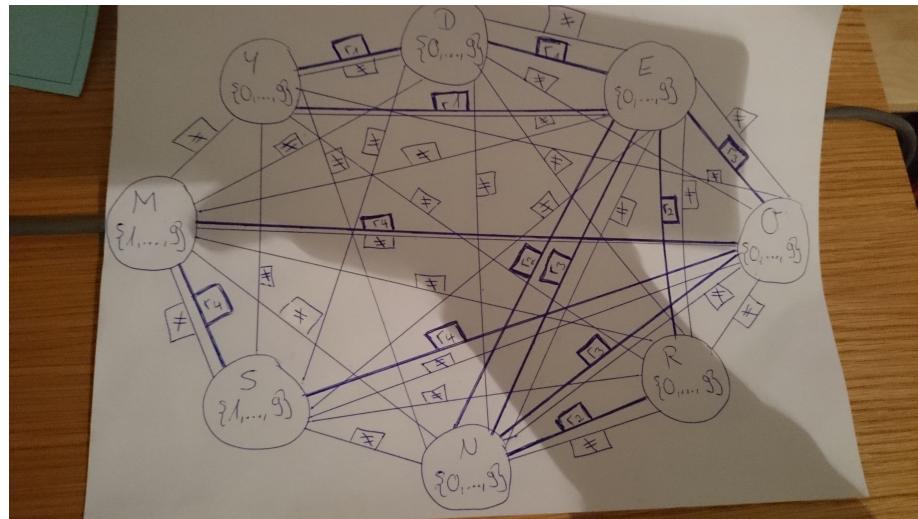
$$r4: O = (S + M + (E + O + (N + R + ((D + E) \bmod 10)) \bmod 10) \bmod 10) \bmod 10$$

Since we have relations between 3 variables (e.g. Y, D, E, because  $Y = (D + E) \bmod 10$ ), 3 arcs in the network are the result.

The mod 10 indicates a possible carry after each sum.

Every variable can be a digit between 0 and 9. S and M can't be zero, because they are the leading digit of "SEND" and "MONEY"/"MORE".

Every variable represents another digit, so no variable can be the same as another.



Manual constraint solving. First, try to solve the problem without any formal methods or tools.

Human solution: Trial and Error with some logic.

Step 1:

Which words can be filled in 1Across (A1xD1, A1xD2, A1xD3) or 1Down (A1xD1, A2xD1, A3xD1)?

**add?** No, because then we need 2 words, which start with d. But we don't have any word starting with d.

(...)

**are?** Yes, because we have other words starting with a or e and one word starting with r.

(...)

**bag?** No because we don't have a word starting with g.

(...)

**lee?** No, because we would need a second word starting with l and we don't have one.

**oaf?** No, because we would need a second word starting with o and we don't have one.

(...)

The following words remain, if we exclude words from the list by the logic above:  
are, art, bat, bee, boa, ear, eel, eft, far, fat, tar.

Step 2:

Enter the remaining words from step 1, which start with the same letter 1Across and 1Down. Test if possible:

**are and art.** Impossible, because then we would need two words starting with r, but we have only one from the origin list (rat).

**far and fat.** Impossible, because if we use far, we also have to enter rat in 3Down. Then we have a word starting and ending with t in 3Across. But we don't have a word of this form in the origin list.

**tar.** Impossible, because no other word starting with t in the list from step 1.

**ear and eel, ear and eft or eel and eft.** Impossible, (...).

**bat and bee or bat and boar.** Impossible. Case bat and bee: If we choose bee as 1Across and bat as 1Down, then we can put the words ear, eel, eft 2Down and 3 Down. But that gives us an r, l or t in A3xD2, which is not possible, because only words from origin list start with t, but none of them has r, l or t as second letter. (...)

Only one solution: **bee and boa** in 1Across and 1Down. Rest see table:

x	D1	D2	D3
A1	b	e	e
A2	o	a	f
A3	a	r	t

**Solve the problem by hand using domain consistency as a first step and as a second step the arc consistency.**

At the beginning every variable's domain consists of the full set of characters of all words. This set is:

a b c d e f g h i k l m n o p r s t u w y

At first we define our domain consistency constraints. Considering that the riddle only allows words with a length of 3 characters and they either have to be written from left to right or from top to bottom we can create constraints for the starting letters of every word.  $A3D1$ ,  $A2D1$ ,  $A1D1$ ,  $A1D2$  and  $A1D3$  can not contain other letters than the first letters of every word. So we get the set of first letters of every word and create the intersection with each of the aforementioned variable's domains.

The next step are the characters in the middle of each word. Again we create a set of these characters and then create the intersection with this set and  $A2D1$ ,  $A2D2$ ,  $A2D3$ ,  $A1D2$  and  $A3D2$  because these fields contain the characters in the middle of each word.

We repeat this procedure with the end of every word's character. After building the set of this characters we create the intersection of this set and  $A3D1$ ,  $A3D2$ ,  $A3D3$ ,  $A2D2$  and  $A1D3$  since these fields represent the ending of each word in row or column.

After this approach our domains look like this:

	D1	D2	D3
A1	a b e f l o r t	a e f o r	a e f l o r t
A2	a e f o r	a d e f g i n o p r s u w y	a d e f g n o r y
A3	a e f l o r t	a d e f g n o r	a c d e f g h k l m n o r t y

Now we define our arc consistency constraints. Each row ( $A1$ ,  $A2$ ,  $A3$ ) and each column ( $D1$ ,  $D2$ ,  $D3$ ) has to be pairwise unequal to every other row / column because no word should appear twice so we define a pairwise unequal arc consistency and apply it on each row and each column.

In the end we define ternary arc constraints for each row and column so that they only can contain a valid word. E.g. such a constraint would be applied on the variables  $A1D1$ ,  $A1D2$ ,  $A1D3$  (row  $A1$ ) so that they can only contain a valid tuple (one word like *add*).

On these constraints the generalized arc consistency algorithm would be applied. In the beginning every arc is marked, after one is processed its mark is removed. Every arc that is possibly affected by the applying of one arc (constraint) is being marked for processing again. Therefore we somehow intelligently visit every arc which possibly could supply an update to our variables.