Chapter 4: Basic Constraint Reasoning (SEND+MORE=MONEY)

Helmut Simonis

Cork Constraint Computation Centre
Computer Science Department
University College Cork
Ireland

ECLiPSe ELearning Overview

Licence

Outline

What we want to introduce

- Finite Domain Solver in ECLiPSe
- Models and Programs
- Constraint Propagation and Search
- Basic constraints: linear arithmetic, alldifferent, disequality
- Built-in search: Labeling
- Visualizers for variables, constraints and search

Problem Definition

A Crypt-Arithmetic Puzzle

We begin with the definition of the SEND+MORE=MONEY puzzle. It is often shown in the form of a hand-written addition:

	S	E	N	D
+	M	O	R	E
M	O	N	E	Y

Rules

- Each character stands for a digit from 0 to 9.
- Numbers are built from digits in the usual, positional notation.
- Repeated occurrence of the same character denote the same digit.
- Different characters denote different digits.
- Numbers do not start with a zero.
- The equation must hold.

	S	E	N	D
+	M	O	R	E
M	O	N	E	Y

Model

- Each character is a variable, which ranges over the values 0 to 9.
- An *alldifferent* constraint between all variables, which states that two different variables must have different values. This is a very common constraint, which we will encounter in many other problems later on.
- Two *disequality constraints* (variable *X* must be different from value *V*) stating that the variables at the beginning of a number can not take the value 0.
- An arithmetic *equality constraint* linking all variables with the proper coefficients and stating that the equation must hold.

General Program Structure

```
:- module(sendmory).
:- export(sendmory/1).
:- lib(ic).
sendmory(L):-
    L = [S,E,N,D,M,O,R,Y],⇨ Variables
    L :: 0..9,
    alldifferent(L),⇨ Constraints
    S #\= 0, M #\= 0,
    1000*S + 100*E + 10*N + D +
    1000*M + 100*O + 10*R + E #=
    10000*M + 1000*O + 100*N + 10*E + Y,
    labeling(L).⇨ Search
```

Choice of Model

- This is *one* model, not *the* model of the problem
- Many possible alternatives
- Choice often depends on your constraint system
  - Constraints available
  - Reasoning attached to constraints
- Not always clear which is the *best* model
- Often: Not clear what is the *problem*

Alternative 1 Alternative 2

Running the program

- To run the program, we have to enter the query
  - `sendmory:sendmory(L).`
- Result
  - `L = [9, 5, 6, 7, 1, 0, 8, 2]`
  - `yes (0.00s cpu, solution 1, maybe more)`

Question

- But how did the program come up with this solution?

Domain Definition

```
L = [S,E,N,D,M,O,R,Y],
L :: 0..9,
```

$$[S, E, N, D, M, O, R, Y] \in \{0..9\}$$

Domain Visualization

Columns = Values

Rows = Variables

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M			Cells= State							
O										
R										
Y										

Alldifferent Constraint

```
alldifferent(L),
```

- Built-in of `ic` library
- No initial propagation possible
- *Suspends*, waits until variables are changed
- When variable is fixed, remove value from domain of other variables
- *Forward checking*

Alldifferent Visualization

Uses the same representation as the domain visualizer

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Disequality Constraints

```
S #\= 0, M#\= 0,
```

Remove value from domain

$$S \in \{1..9\}, M \in \{1..9\}$$

Constraints solved, can be removed

Domains after Disequality

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Equality Constraint

- Normalization of linear terms
    - Single occurence of variable
    - Positive coefficients
- Propagation

Normalization

	1000*S+	100*E+	10*N+	D
	+1000*M+	100*O+	10*R+	E
10000*M+	1000*O+	100*N+	10*E+	Y

is transformed into

	1000*S+	91*E+		D
		+	10*R	
9000*M+	900*O+	90*N+		Y

Simplified Equation

$$1000*S+91*E+10*R+D=9000*M+900*O+90*N+Y$$

Propagation

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{1000..9918} =$$
$$\underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..89919}$$

Deduction:

$$M = 1, S = 9, O \in \{0..1\}$$

Why? Skip

Cork Constraint Computation Centre

Consider lower bound for S

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} = \underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

- Lower bound of equation is 9000
- Rest of lhs (left hand side) $(91 * E^{0..9} + 10 * R^{0..9} + D^{0..9})$ is atmost 918
- S must be greater or equal to $\frac{9000-918}{1000} = 8.082$
  - otherwise lower bound of equation not reached by lhs
- S is integer, therefore $S \geq \lceil \frac{9000-918}{1000} \rceil = 9$
- S has upper bound of 9, so $S = 9$

Cork Constraint Computation Centre

Consider upper bound of M

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} = \underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

- Upper bound of equation is 9918
- Rest of rhs (right hand side) $900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}$ is at least 0
- M must be smaller or equal to $\frac{9918-0}{9000} = 1.102$
- M must be integer, therefore $M \leq \lfloor \frac{9918-0}{9000} \rfloor = 1$
- M has lower bound of 1, so $M = 1$

Problem
Program
Constraint Setup
Search
Lessons Learned
Domain Definition
Alldifferent Constraint
Disequality Constraints
Equality Constraint

Consider upper bound of O

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} = \underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

- Upper bound of equation is 9918
- Rest of rhs (right hand side) $9000 * 1 + 90 * N^{0..9} + Y^{0..9}$ is at least 9000
- O must be smaller or equal to $\frac{9918-9000}{900} = 1.02$
- O must be integer, therefore $O \leq \lfloor \frac{9918-9000}{900} \rfloor = 1$
- O has lower bound of 0, so $O \in \{0..1\}$

Propagation of equality: Result

	0	1	2	3	4	5	6	7	8	9
S		-	-	-	-	-	-	-	-	✹
E										
N										
D										
M		✹	-	-	-	-	-	-	-	-
O			✖	✖	✖	✖	✖	✖	✖	✖
R										
Y										

Propagation of alldifferent

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

$$O = 0, [E, R, D, N, Y] \in \{2..8\}$$

Waking the equality constraint

- Triggered by assignment of variables
- *or* update of lower or upper bound

Removal of constants

$$1000 * 9 + 91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} = 9000 * 1 + 900 * 0 + 90 * N^{2..8} + Y^{2..8}$$

$$\mathbf{1000} * \mathbf{9} + 91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} = \mathbf{9000} * \mathbf{1} + \mathbf{900} * \mathbf{0} + 90 * N^{2..8} + Y^{2..8}$$

$$91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} = 90 * N^{2..8} + Y^{2..8}$$

Propagation of equality (Iteration 1)

$$\underbrace{91 * E^{2..8} + 10 * R^{2..8} + D^{2..8}}_{204..816} = \underbrace{90 * N^{2..8} + Y^{2..8}}_{182..728}$$

$$\underbrace{91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} = 90 * N^{2..8} + Y^{2..8}}_{204..728}$$

$$N \geq 3 = \lceil \frac{204-8}{90} \rceil, E \leq 7 = \lfloor \frac{728-22}{91} \rfloor$$

Problem
Program
Constraint Setup
Search
Lessons Learned

Domain Definition
Alldifferent Constraint
Disequality Constraints
Equality Constraint

Propagation of equality (Iteration 2)

$$91 * E^{2..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}$$

$$\underbrace{91 * E^{2..7} + 10 * R^{2..8} + D^{2..8}}_{204..725} = \underbrace{90 * N^{3..8} + Y^{2..8}}_{272..728}$$

$$\underbrace{91 * E^{2..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}}_{272..725}$$

$$E \geq 3 = \lceil \frac{272-88}{91} \rceil$$

Propagation of equality (Iteration 3)

$$91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}$$

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8}}_{295..725} = \underbrace{90 * N^{3..8} + Y^{2..8}}_{272..728}$$

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}}_{295..725}$$

$$N \geq 4 = \lceil \frac{295 - 8}{90} \rceil$$

Cork Constraint Computation Centre

Propagation of equality (Iteration 4)

$$91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}$$

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8}}_{295..725} = \underbrace{90 * N^{4..8} + Y^{2..8}}_{362..728}$$

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}}_{362..725}$$

$$E \geq 4 = \lceil \frac{362 - 88}{91} \rceil$$

Cork Constraint Computation Centre

Propagation of equality (Iteration 5)

$$91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}$$

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8}}_{386..725} = \underbrace{90 * N^{4..8} + Y^{2..8}}_{362..728}$$

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}}_{386..725}$$

$$N \geq 5 = \lceil \frac{386 - 8}{90} \rceil$$

Propagation of equality (Iteration 6)

$$91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}$$

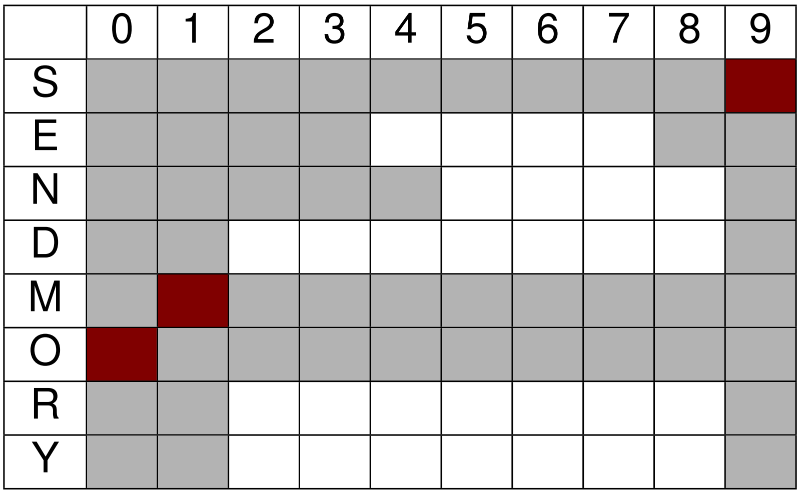
$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8}}_{386..725} = \underbrace{90 * N^{5..8} + Y^{2..8}}_{452..728}$$

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}}_{452..725}$$

$$N \geq 5 = \lceil \frac{452 - 8}{90} \rceil, E \geq 4 = \lceil \frac{452 - 88}{91} \rceil$$

No further propagation at this point

Domains after setup



`labeling` built-in

```
labeling([S,E,N,D,M,O,R,Y])
```

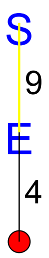
- Try variable is order given
- Try values starting from smallest value in domain
- When failing, backtrack to last open choice
- *Chronological Backtracking*
- *Depth First search*

Search Tree Step 1

Variable S already fixed

Step 2, Alternative $E = 4$

Variable $E \in \{4..7\}$, first value tested is 4



Assignment $E = 4$

	0	1	2	3	4	5	6	7	8	9
S										
E					✸	-	-	-		
N										
D										
M										
O										
R										
Y										

Cork Constraint Computation Centre

Propagation of $E = 4$, equality constraint

$$91 * 4 + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}$$

$$\underbrace{91 * 4 + 10 * R^{2..8} + D^{2..8}}_{386..452} = \underbrace{90 * N^{5..8} + Y^{2..8}}_{452..728}$$

$$\underbrace{91 * 4 + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}}_{452}$$

$$N = 5, Y = 2, R = 8, D = 8$$

Cork Constraint Computation Centre

Result of equality propagation

	0	1	2	3	4	5	6	7	8	9
S										
E										
N						✸	-	-	-	
D			-	-	-	-	-	-	✸	
M										
O										
R			-	-	-	-	-	-	✸	
Y			✸	-	-	-	-	-	-	

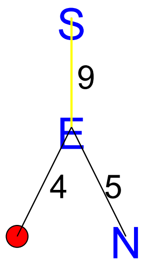
Propagation of alldifferent

	0	1	2	3	4	5	6	7	8	9
S									|	
E									|	
N						✸	-	-	|	
D			-	-	-	-	-	-	✸	
M									|	
O									|	
R			-	-	-	-	-	-	✸	
Y			✸	-	-	-	-	-	|	

Alldifferent fails!

Step 2, Alternative $E = 5$

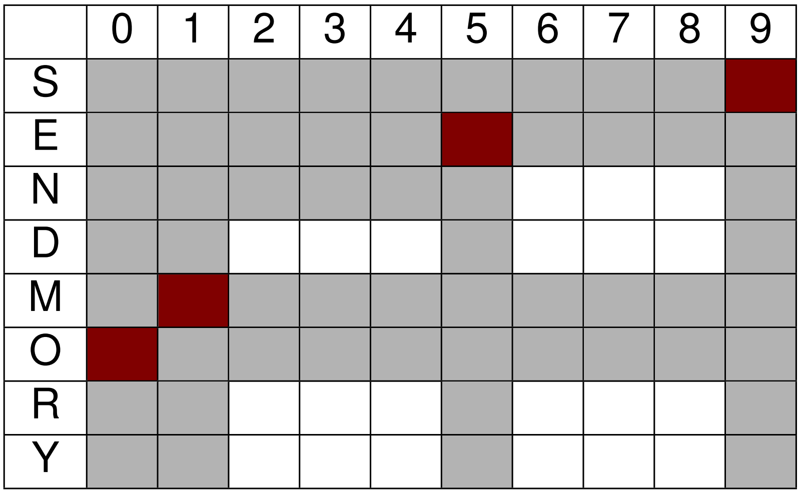
Return to last open choice, *E*, and test next value



Assignment $E = 5$

	0	1	2	3	4	5	6	7	8	9
S										
E					-	✸	-	-		
N										
D										
M										
O										
R										
Y										

Propagation of alldifferent



$$N \neq 5, N \geq 6$$

Cork Constraint Computation Centre

Propagation of equality

$$91 * 5 + 10 * R^{2..8} + D^{2..8} = 90 * N^{6..8} + Y^{2..8}$$

$$\underbrace{91 * 5 + 10 * R^{2..8} + D^{2..8}}_{477..543} = \underbrace{90 * N^{6..8} + Y^{2..8}}_{542..728}$$

$$\underbrace{91 * 5 + 10 * R^{2..8} + D^{2..8} = 90 * N^{6..8} + Y^{2..8}}_{542..543}$$

$$N = 6, Y \in \{2, 3\}, R = 8, D \in \{7..8\}$$

Cork Constraint Computation Centre

Result of equality propagation

	0	1	2	3	4	5	6	7	8	9
S										
E										
N							✹	-	-	
D			✖	✖	✖		✖			
M										
O										
R			-	-	-		-	-	✹	
Y					✖		✖	✖	✖	

Propagation of `alldifferent`

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

$$D = 7$$

Propagation of equality

$$91 * 5 + 10 * 8 + 7 = 90 * 6 + Y^{2..3}$$

$$\underbrace{91 * 5 + 10 * 8 + 7}_{542} = \underbrace{90 * 6 + Y^{2..3}}_{542..543}$$

$$\underbrace{91 * 5 + 10 * 8 + 7 = 90 * 6 + Y^{2..3}}_{542}$$

$$Y = 2$$

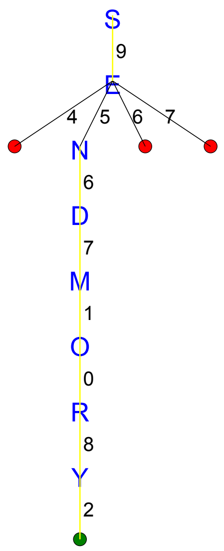
Cork Constraint Computation Centre

Last propagation step

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y			✹	-						

Cork Constraint Computation Centre

Complete Search Tree



Solution

$$
\begin{array}{cccccc}
 & & 9 & 5 & 6 & 7 \\
 & + & 1 & 0 & 8 & 5 \\
\hline
 & 1 & 0 & 6 & 5 & 2
\end{array}
$$

Topics introduced

- Finite Domain Solver in ECLiPSe, `ic` library
- Models and Programs
- Constraint Propagation and Search
- Basic constraints: linear arithmetic, `alldifferent`, disequality
- Built-in search: `labeling`
- Visualizers for variables, constraints and search

Lessons Learned

- Constraint models are expressed by variables and constraints.
- Problems can have many different models, which can behave quite differently. Choosing the best model is an art.
- Constraints can take many different forms.
- Propagation deals with the interaction of variables and constraints.
- It removes some values that are inconsistent with a constraint from the domain of a variable.
- Constraints only communicate via shared variables.

Lessons Learned

- Propagation usually is not sufficient, search may be required to find a solution.
- Propagation is data driven, and can be quite complex even for small examples.
- The default search uses chronological depth-first backtracking, systematically exploring the complete search space.
- The search choices and propagation are interleaved, after every choice some more propagation may further reduce the problem.

Cork Constraint Computation Centre

Alternative 1

- Do we need the constraint “Numbers do not begin with a zero”?
- This is not given explicitely in the problem statement
- Remove disequality constraints from program
- Previous solution is still a solution
- Does it change propagation?
- Does it have more solutions?

Cork Constraint Computation Centre

Program without Disequality

Listing 1: Alternative 1

```
:-module(alternative1).
:-export(sendmory/1).
:-lib(ic).

sendmory(L):-
        L = [S,E,N,D,M,O,R,Y],
        L :: 0..9,
        alldifferent(L),
        1000*S + 100*E + 10*N + D +
        1000*M + 100*O + 10*R + E #=
        10000*M + 1000*O + 100*N + 10*E + Y,
        labeling(L).
```


After Setup without Disequality

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Setup Comparison

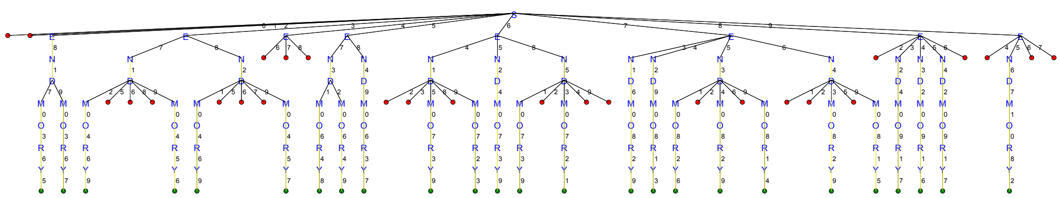
original

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

alternative 1

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Search Tree: Many Solutions



Note:

- Not just a different model, solving a different problem!
- Often we can choose which problem we want to solve
  - Which constraints to include
  - What to ignore
- In this case not acceptable

◂ Choice of Model

Cork Constraint Computation Centre

Alternative 2

- Large equality difficult to understand by humans
- Replace with multiple, simpler equations
- Linked by carry variables (0/1)
- Should produce same solutions
- Does it give same propagation?

		S	E	N	D
+		M	O	R	E
+	$C5$	$C4$	$C3$	$C2$	
	M	O	N	E	Y

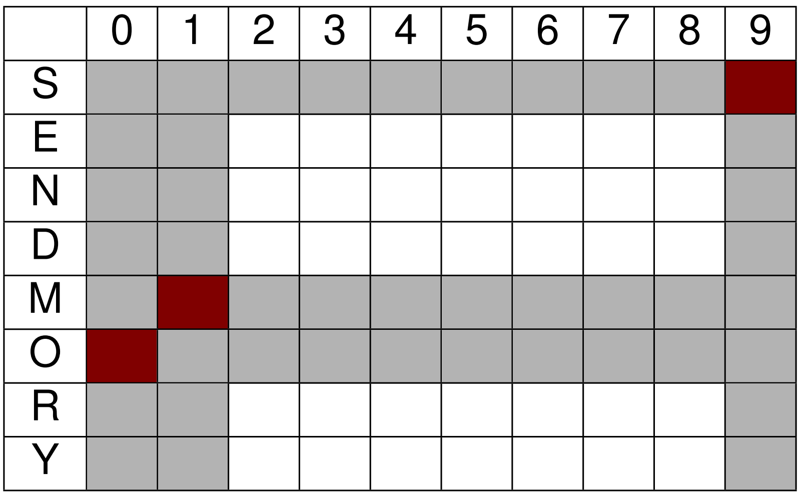
Cork Constraint Computation Centre

Carry Variables with Multiple Equations

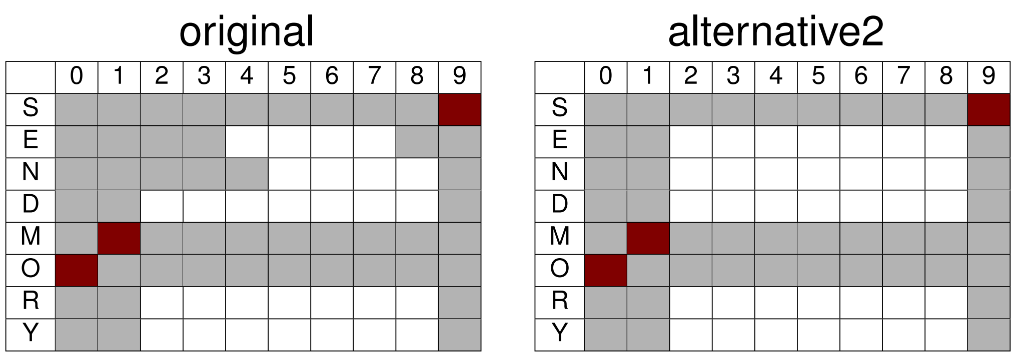
```
:-module(alternative2),export(sendmory/1),lib(ic).
sendmory(L):-➪ same as before
    L=[S,E,N,D,M,O,R,Y],L :: 0..9,
    [C2,C3,C4,C5] :: 0..1, ➪ new
    alldifferent(L),
    S #\= 0,M #\= 0,
    M #= C5,
    S+M+C4 #= 10*C5+O,
    E+O+C3 #= 10*C4+N,
    N+R+C2 #= 10*C3+E,
    D+E    #= 10*C2+Y,
    labeling(L).
```

```
       S   E   N   D
+      M   O   R   E
+C5   C4  C3  C2
   M   O   N   E   Y
```

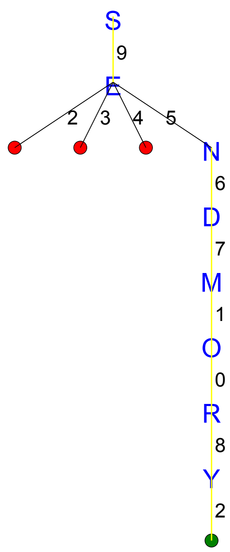

With Carry Variables: After Setup



Setup Comparison

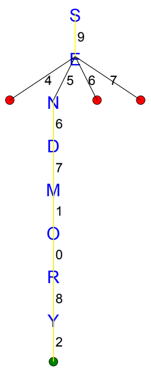


Search Tree: First Solution

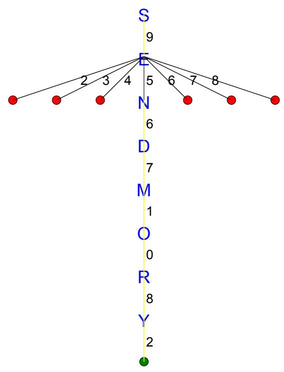


Comparison

Single Equation



Multiple Equations



Observations

- This is solving the original problem
- Search tree slightly bigger
- Caused here by missing interaction of equations
- And repeated variables
- But: Introducing auxiliary variables not always bad!

Choice of Model

More Information

Henry Dudeney.
Send+More=Money.
Strand Magazine, Volume 68:pages 97 and 214, July 1924.

Henry Dudeney.
Amusements in Mathematics.
Project Gutenberg, 1917.
http://www.gutenberg.org/etext/16713.

Exercises

1. Does the reasoning for the equality constraints that we have presented remove all inconsistent values? Consider the constraint Y=2*X.
2. Why is it important to remove multiple occurences of the same variable from an equality constraint? Give an example!
3. Solve the puzzle DONALD+GERALD=ROBERT. What is the state of the variables before the search, after the initial constraint propagation?
4. Solve the puzzle Y*WORRY = DOOOOD. What is different?
5. (extra credit) How would you design a program that finds new crypt-arithmetic puzzles? What makes a good puzzle?