Centre for
Data Analytics

Insight

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

Helmut Simonis

CRT-AI CP Week 2025

DCU NUI Galway OÉ Gaillimh UCC UCD Dublin sfi

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

Licence

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit `http://creativecommons.org/licenses/by-nc-sa/3.0/` or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 at Insight the SFI Research Centre for Data Analytics at UCC, which is co-funded under the European Regional Development Fund.
A version of this material was developed as part of the ECLiPSe ELearning course: `https://eclipseclp.org/ELearning/index.html`.
Support from Cisco Systems and the Silicon Valley Community Foundation is gratefully acknowledged.

Example 1: SEND+MORE=MONEY

- Example of Finite Domain Constraint Problem
- Models and Programs
- Constraint Propagation and Search
- Some Basic Constraints: linear arithmetic, alldifferent, disequality
- A Built-in search
- 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:

$$\begin{array}{cccccc} & & S & E & N & D \\ + & & M & O & R & E \\ \hline & M & O & N & E & Y \end{array}$$

The puzzle was first proposed by Henry Dudeney in the Strand Magazine from 1924.

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.

$$\begin{array}{ccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

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.

SEND+MORE=MONEY Models

- ECLiPSe ▸ Show
- MiniZinc ▸ Show
- NumberJack ▸ Show
- CPMpy ▸ Show
- Choco-solver ▸ Show

ECLiPSe Model

```
:- lib(ic).

sendmore(Digits) :-
    Digits = [S,E,N,D,M,O,R,Y],
    Digits :: [0..9],
    alldifferent(Digits),
    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(Digits).
```

▸ Continue

MiniZinc Model

```
include "alldifferent.mzn";
var 0..9: S;
var 0..9: E;
var 0..9: N;
var 0..9: D;
var 0..9: M;
var 0..9: O;
var 0..9: R;
var 0..9: Y;
constraint S != 0;
constraint M != 0;
constraint          1000 * S + 100 * E + 10 * N + D
                  + 1000 * M + 100 * O + 10 * R + E
       = 10000 * M + 1000 * O + 100 * N + 10 * E + Y;
constraint alldifferent([S,E,N,D,M,O,R,Y]);

solve satisfy;
```

Continue

NumberJack Model (from https://github.com/eomahony/Numberjack/)

```
from Numberjack import *

def get_model():
    model = Model()
    s, m = VarArray(2, 1, 9)
    e, n, d, o, r, y = VarArray(6, 0, 9)
    model.add(      s*1000 + e*100 + n*10 + d +
                    m*1000 + o*100 + r*10 + e ==
          m*10000 + o*1000 + n*100 + e*10 + y)
    model.add(AllDiff((s, e, n, d, m, o, r, y)))
    return s, e, n, d, m, o, r, y, model

def solve(param):
    s, e, n, d, m, o, r, y, model = get_model()
    solver = model.load(param['solver'])
    solver.setVerbosity(param['verbose'])
    solver.solve()
```

Continue

CPMpy Model (from https://github.com/CPMpy/)

```
from cpmpy import *
import numpy as np

s,e,n,d,m,o,r,y = intvar(0,9, shape=8)
model = Model(
    AllDifferent([s,e,n,d,m,o,r,y]),
    (    sum(   [s,e,n,d] * np.array([       1000, 100, 10, 1]) ) \
       + sum(   [m,o,r,e] * np.array([       1000, 100, 10, 1]) ) \
      == sum( [m,o,n,e,y] * np.array([10000, 1000, 100, 10, 1]) ) ),
    s > 0,
    m > 0,
)

model.solve()
```

▸ Continue

Choco-solver Model (from https://choco-solver.org/)

```
Model model = new Model("SEND+MORE=MONEY");
IntVar S = model.intVar("S", 1, 9, false);
IntVar E = model.intVar("E", 0, 9, false);
IntVar N = model.intVar("N", 0, 9, false);
IntVar D = model.intVar("D", 0, 9, false);
IntVar M = model.intVar("M", 1, 9, false);
IntVar O = model.intVar("O", 0, 9, false);
IntVar R = model.intVar("R", 0, 9, false);
IntVar Y = model.intVar("Y", 0, 9, false);

model.allDifferent(new IntVar[]{S, E, N, D, M, O, R, Y}).post();

IntVar[] ALL = new IntVar[]{
    S, E, N, D,
    M, O, R, E,
    M, O, N, E, Y};
int[] COEFFS = new int[]{
    1000, 100, 10, 1,
    1000, 100, 10, 1,
    -10000, -1000, -100, -10, -1};
model.scalar(ALL, COEFFS, "=", 0).post();

Solver solver = model.getSolver();
solver.showStatistics();
solver.showSolutions();
solver.findSolution();
```

▸ Continue

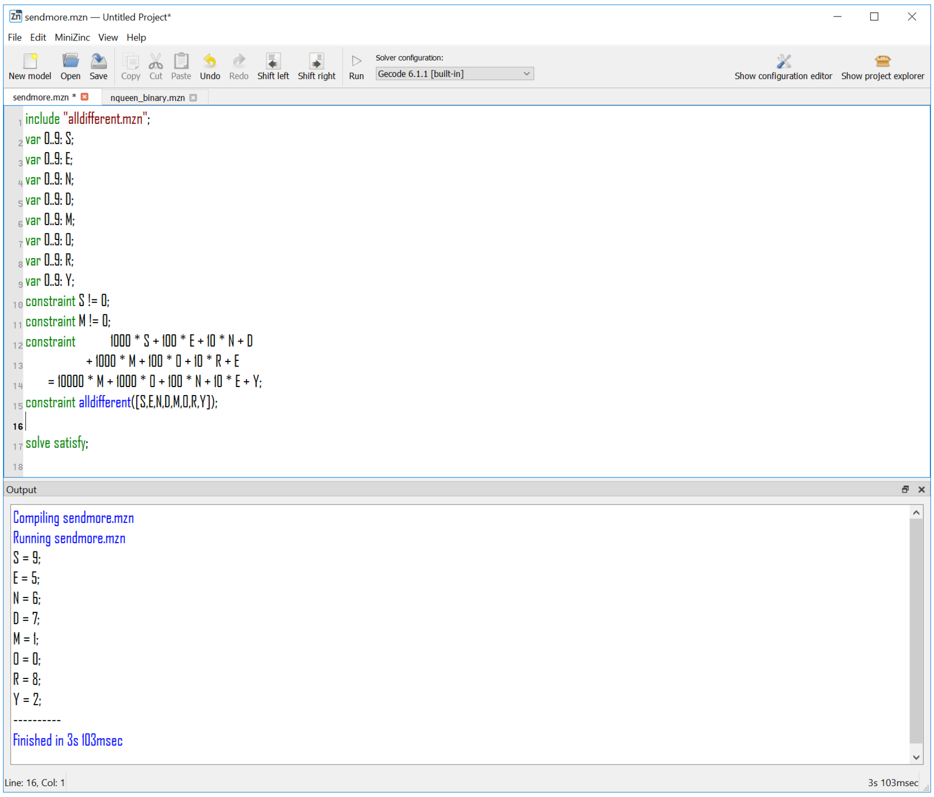
A Note on Syntax

- Some formulations may seem simpler than others
- This largely is an artifact of a very simple problem
- In most models, you do not write down constraints one by one
- You create constraints based on data
- Ease of integration becomes more important than syntax
- Debugging tools for those who need a debugger :-)

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*

Running the Program (MiniZinc IDE)



Question

- But how did the program come up with this solution?
- We show solution with ECLiPse, other solvers vary slightly

Domain Definition

```
var 0..9: S;
var 0..9: E;
var 0..9: N;
var 0..9: D;
var 0..9: M;
var 0..9: O;
var 0..9: R;
var 0..9: Y;
```


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

```
include "alldifferent.mzn";

constraint alldifferent([S,E,N,D,M,O,R,Y]);
```

- Built-in alldifferent predicate included
- 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

```
constraint S != 0;
constraint 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

$$
\begin{array}{rrrrr}
 & 1000*S+ & 100*E+ & 10*N+ & D \\
 & +1000*M+ & 100*O+ & 10*R+ & E \\
\hline
10000*M+ & 1000*O+ & 100*N+ & 10*E+ & Y
\end{array}
$$

is transformed into

$$
\begin{array}{rrrrr}
 & 1000*S+ & 91*E+ & & D \\
 & & + & 10*R & \\
\hline
9000*M+ & 900*O+ & 90*N+ & & Y
\end{array}
$$

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

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 $S = 9$

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 $M = 1$

Consider upper bound of O

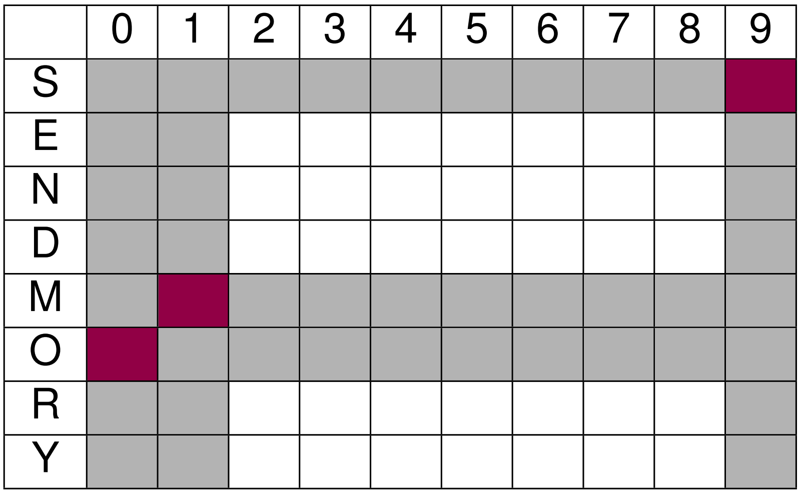
$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} = \underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

- 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



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

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)

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

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

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

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)

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

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

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

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

Propagation of equality (Iteration 4)

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

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

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

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

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)

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

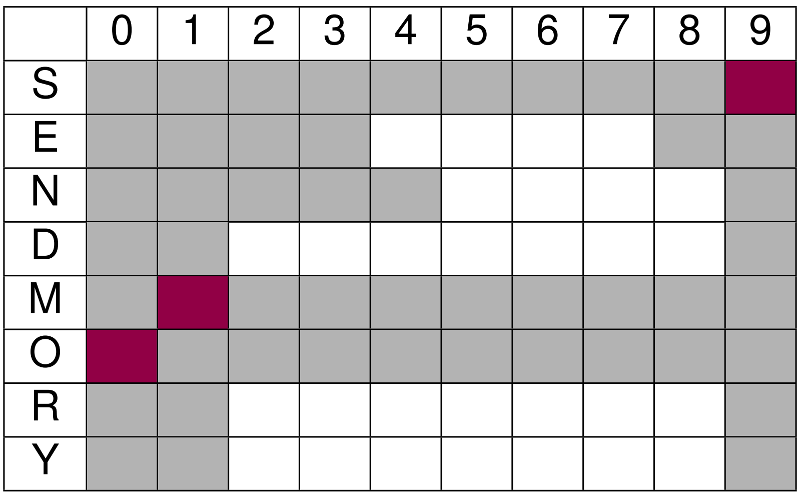
$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8}}_{386..725} = \underbrace{90 * N^{5..8} + Y^{2..8}}_{452..728}$$

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

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

No further propagation at this point

Domains after setup



Search

solve satisfy;

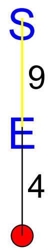
- Try to find a feasible solution, choice left to solver
- Naive search strategy shown here
  - Try variable in order given
  - Try values starting from smallest value in domain
  - When failing, backtrack to last open choice
  - *Chronological Backtracking*
  - *Depth First search*

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										

Propagation of $E = 4$, equality constraint

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

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

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

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

Result of equality propagation

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

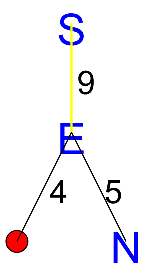
Propagation of alldifferent

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

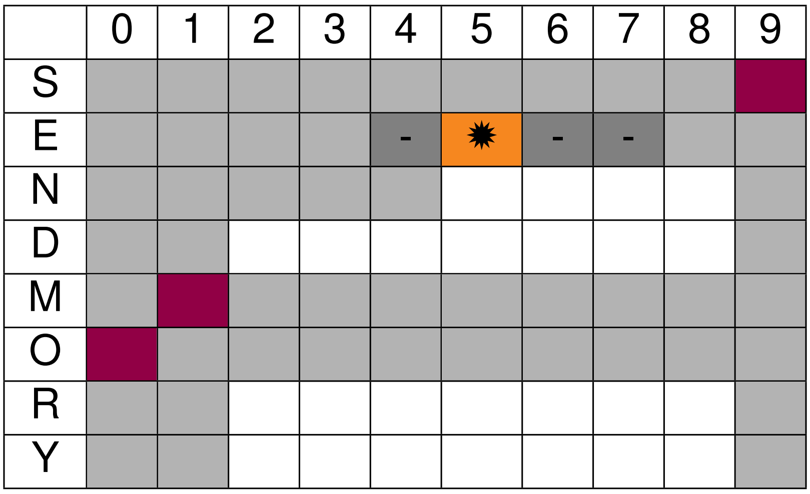
Alldifferent fails!

Step 2, Alternative $E = 5$

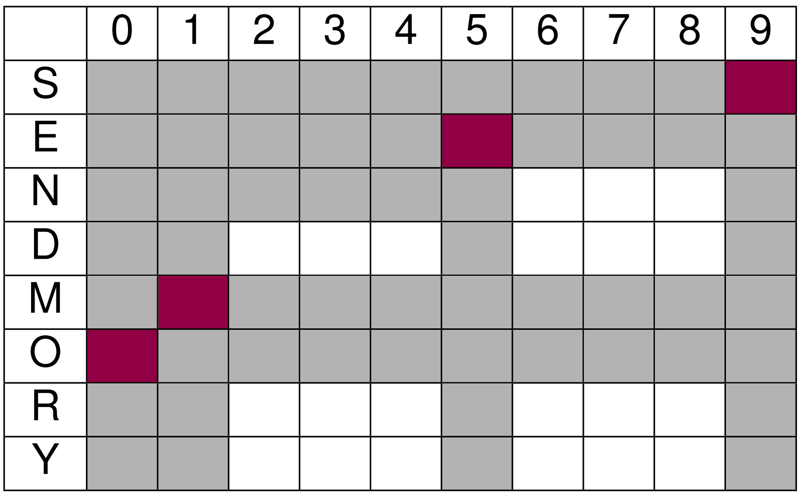
Return to last open choice, E, and test next value



Assignment $E = 5$



Propagation of alldifferent



$N \neq 5, N \geq 6$

Propagation of equality

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

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

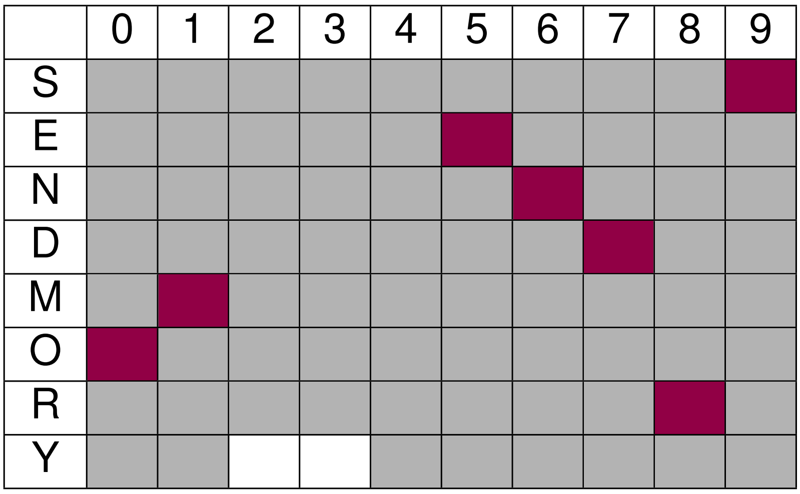
$$\underbrace{91 * 5 + 10 * R^{2..8} + D^{2..8} = 90 * N^{6..8} + Y^{2..8}}_{542..543}$$

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

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`



$$D = 7$$

Propagation of equality

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

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

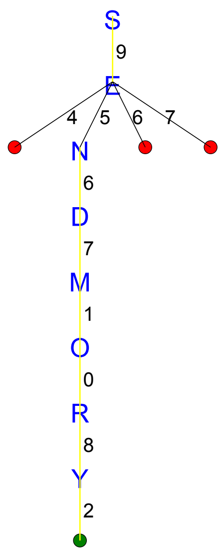
$$\underbrace{91 * 5 + 10 * 8 + 7 = 90 * 6 + Y^{2..3}}_{542}$$

$$Y = 2$$

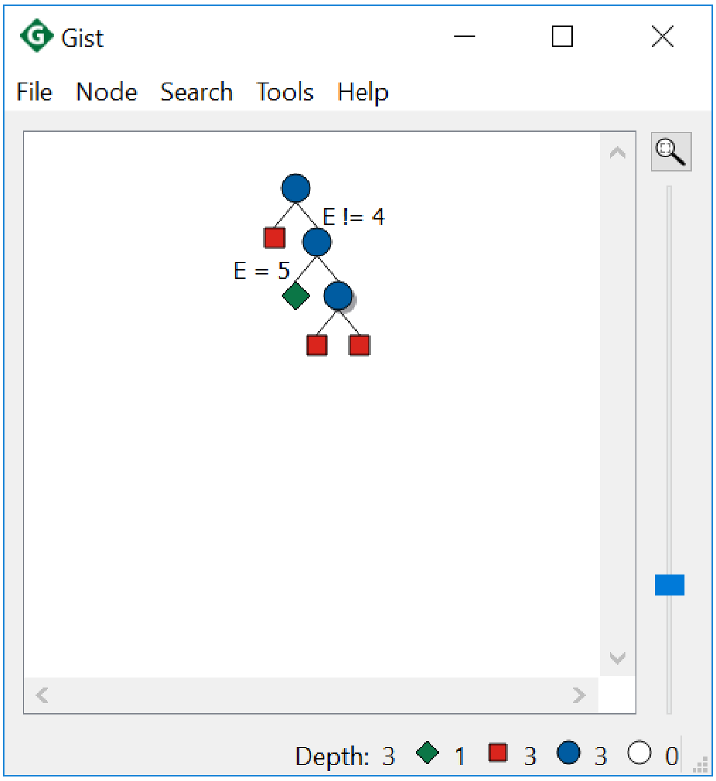
Last propagation step

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

Complete Search Tree



Search Tree with Gecode/GIST



Some Differences

- Uses binary branching
  - var equal value, var not equal value
- Solutions in green, failure leafs in red, internal nodes in blue
- By default, shows all failed sub trees collapsed
- By default, uses different search strategy

Solution

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

Points to Remember

- 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.

Points to Remember

- 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.

For Puzzle Purists Only

- We did not follow the puzzler ethos!
- We should solve the puzzle without making choices
- Even a case analysis should be avoided
- The puzzle has a single solution, we should be able to deduce the solution
- In the process shown, we are limited by the underlying assumptions
  - Treat each constraint on its own, they interact only by domains of variables
  - We only use the constraints that we stated in the model
- Can we do better than that?

Skip

Better Reasoning

- Three possible approaches (possibly many more)
  - Full domain reasoning for arithmetic, not just bound reasoning
  - Interaction of *sum* and *alldifferent* constraints
  - Deduced implied constraint

Looking at more than just bounds

- We only considered the smallest and largest values that can be achieved in the sum constraint
- We can do more
  - Can any of the values between be expressed as the sum of the terms
  - Consider holes in the domains, and in the range of possible values for LHS and RHS
- Usually not done in actual solvers for arithmetic constraints
- Easy to do with Dynamic Programming

Consider the interaction of multiple constraints

- Usually ignored, as only interaction is via domains of shared variables
- Here: Sum and *alldifferent* interact
  - When considering the bounds, we cannot assume that each variable takes its smallest/largest value independently
  - Find feasible assignment that minimizes/maximizes the total weight
  - To do this properly, we need some non-trivial reasoning
- Do we do this combined reasoning automatically, or only when prompted by the modeler?

Deduced Implied Constraints

- Look at the partially solved puzzle
```
 9END
+10RE
  ---
```
- `10NEY`
- In the hundreds position, we have $E + 0 + C_{10} = N + 10 * C_{100}$, with C_{10} the 0/1 carry from the tens position
- NB: No carry C_{100} into the thousands, $C_{100} = 0$
- N must be equal to $E + 1$ with $C_{10} = 1$
- If $C_{10} = 0$, then $N = E$, not possible
- We can substitute $N = E + 1$ into our main equation, but keep $N = E + 1$ as well

Expert Mode Reasoning

Starting with

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

we get

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

Eliminating duplicate occurrences of E

$$\underbrace{E^{4..7} + 10 * R^{2..8} + D^{2..8}}_{26..95} = \underbrace{90 + Y^{2..8}}_{92..98}$$

shared range 92..95 To reach 92, R must be equal to 8, therefore N, E, D, Y must be less than 8
As $N = E + 1$, E must be less than 7

$$E^{4..6} + 10 * 8 + D^{2..7} = 90 + Y^{2..7}$$

Simplification yields

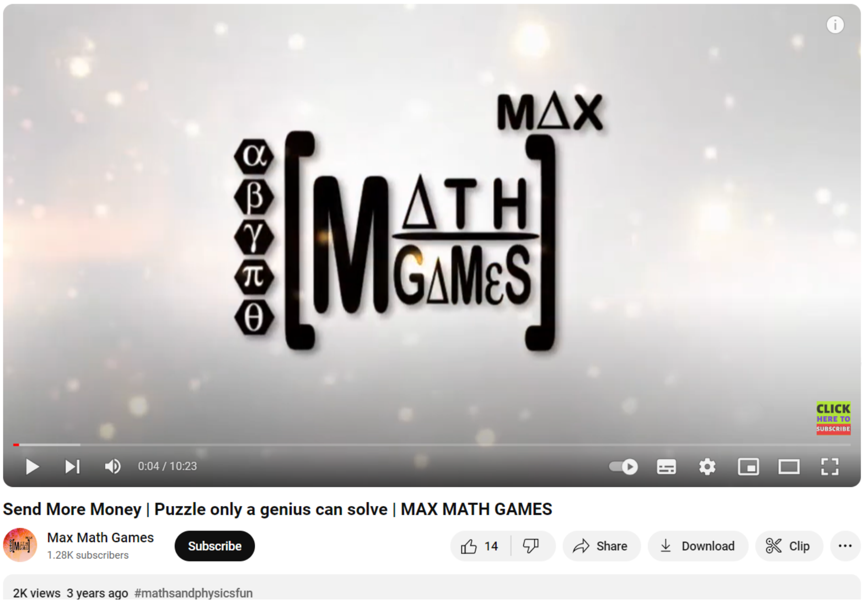
$$\underbrace{E^{4..6} + D^{2..7}}_{6..13} = \underbrace{10 + Y^{2..7}}_{12..17}$$

shared range 12..13 To reach 12 on LHS, E must be greater than 4, D must be greater than 5

Expert Mode Summary

- Often there is more propagation that can be done
- Can be difficult/expensive to do
- Balancing
  - How much work it done at each step of search?
  - How many steps of search you need?
- For hard problems, doing all possible propagation may be exponential
- Not aware that any CP system does the full reasoning shown here

This is how people solve the puzzle by hand



- When writing the first version of this puzzle for CHIP (in 1986), we wanted to mimic the way we solve the puzzle by hand