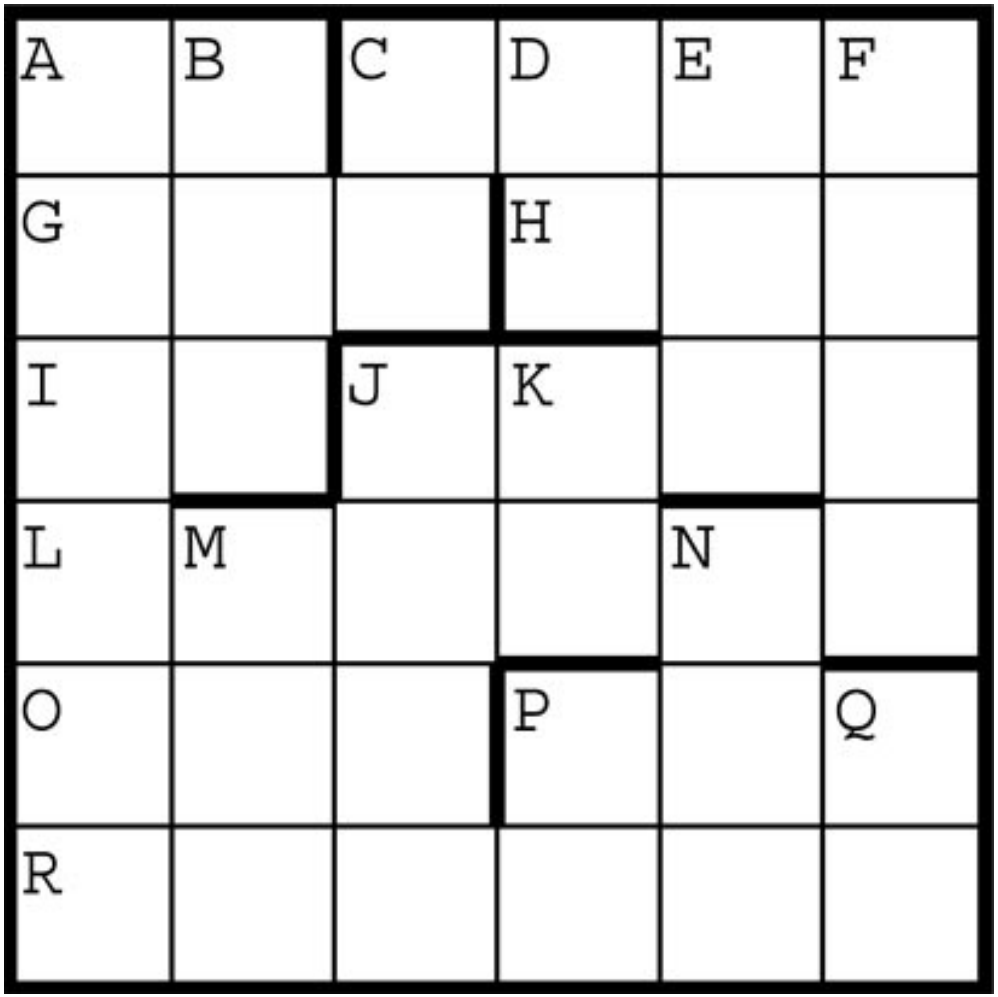
7down – ein Kreuzzahlrätsellöser

Joachim Breitner, Manuel Holtgrewe, Lucas Lürich, Mathias Ziebarth

7down@lists.nomeata.de

14. März 2008

Ein Kreuzzahlenrätsel



Regeln:

Waagerecht:

A Quersumme von L waagerecht
C Mal M senkrecht ergibt R waagerecht
G Vielfaches des Rückwerts von C senkrecht
H C senkrecht plus Rückwert von O waagerecht
I Rückwert von G waagerecht ist Vielfaches J
P waagerecht plus Rückwert von F senkrecht
L A senkrecht plus R waagerecht
O E senkrecht plus K senkrecht
P Vielfaches des Rückwerts von Q senkrecht
R Quadrat von B senkrecht

Senkrecht:

A Vielfaches von A waagerecht
B Quadrat von A waagerecht
C L waagerecht ist Vielfaches
D Die Quersumme von H waagerecht
E Der Rückwert von L waagerecht ist Vielfaches des Rückwerts
F C senkrecht plus der Rückwert dieser Zahl ergibt J senkrecht
J Quadrat von Q senkrecht
K Primzahl
M Mal A waagerecht ergibt J senkrecht **N** Die Quersumme dieser Zahl ist D senkrecht
P Der Rückwert ist eine Primzahl
Q Mal J senkrecht ergibt R waagerecht
(Quelle: Die Zeit 2007/35)

Funktionaler Prototyp

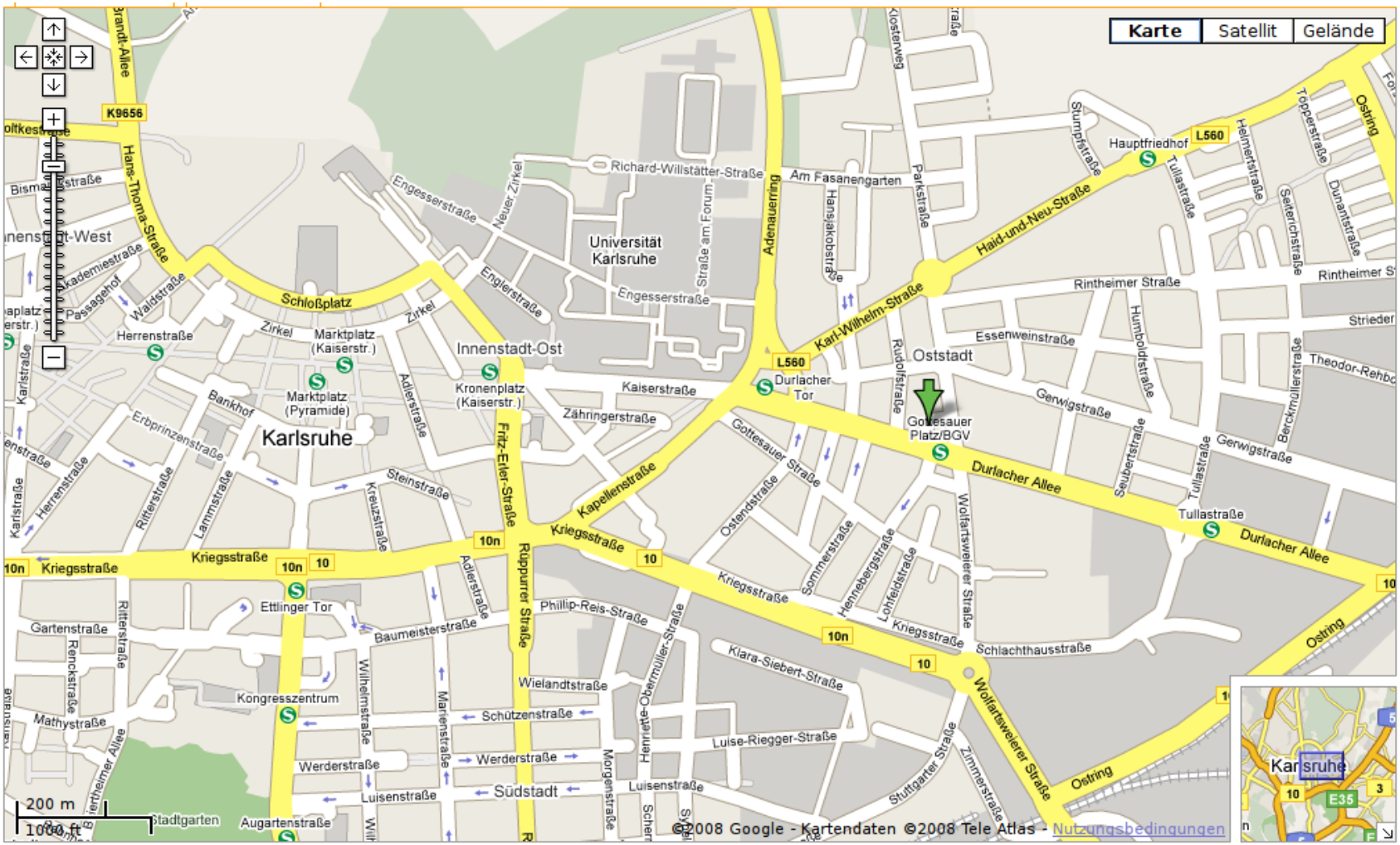
```
data Tightness = TotallyTight | MightConstrain | JustTestable

class Slot a where
        choices      :: a → Possibilities
        splitable    :: a → Bool
        splitPoss    :: a → (a,a)

data Function s =Function {
        tightness    :: Tightness,
        test         :: [s] → s → Maybe Bool,
        run          :: [s] → s → Maybe s
        }

data Edge i s =Edge {
        sources      :: [i],
        target       :: i,
        func         :: Function s
        }
```

Wöchentliche Programmiersitzungen



Jetzt aber richtig: In Python

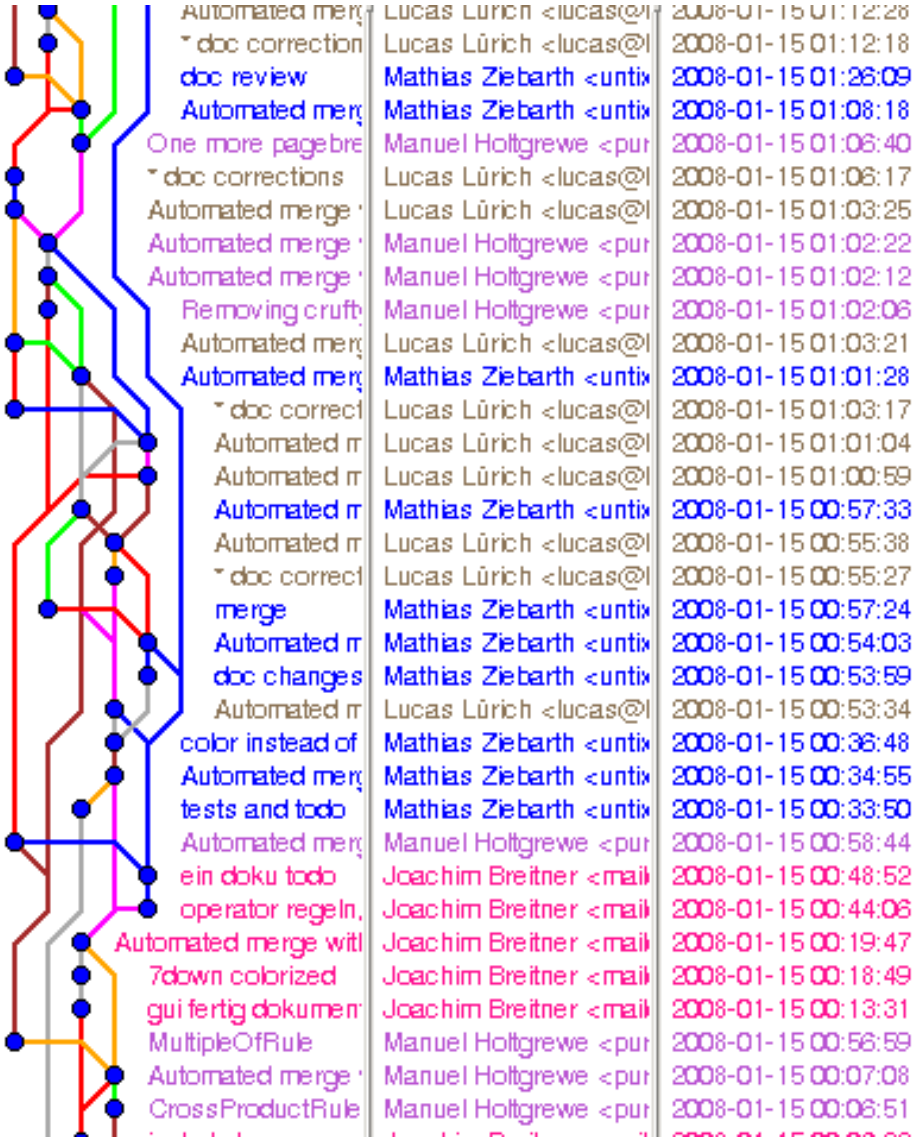
Warum Python?

- dynamisch
- interpretiert
- lesbar
- kompakt

Weiter verwendet wurden:

- GUI-Toolkit: gtk
- Grafikbibliothek: cairo

Mercurial


Automated merg Lucas Lürich <lucas@l 2008-01-15 01:12:26
doc correction Lucas Lürich <lucas@l 2008-01-15 01:12:18
doc review Mathias Ziebarth <untix 2008-01-15 01:26:09
Automated merg Mathias Ziebarth <untix 2008-01-15 01:08:18
One more pagebre Manuel Holtgrewe <pur 2008-01-15 01:06:40
doc corrections Lucas Lürich <lucas@l 2008-01-15 01:06:17
Automated merge Lucas Lürich <lucas@l 2008-01-15 01:03:25
Automated merge Manuel Holtgrewe <pur 2008-01-15 01:02:22
Automated merge Manuel Holtgrewe <pur 2008-01-15 01:02:12
Removing cruft Manuel Holtgrewe <pur 2008-01-15 01:02:06
Automated merg Lucas Lürich <lucas@l 2008-01-15 01:03:21
Automated merg Mathias Ziebarth <untix 2008-01-15 01:01:28
doc correct Lucas Lürich <lucas@l 2008-01-15 01:03:17
Automated m Lucas Lürich <lucas@l 2008-01-15 01:01:04
Automated m Lucas Lürich <lucas@l 2008-01-15 01:00:59
Automated m Mathias Ziebarth <untix 2008-01-15 00:57:33
Automated m Lucas Lürich <lucas@l 2008-01-15 00:55:38
doc correct Lucas Lürich <lucas@l 2008-01-15 00:55:27
merge Mathias Ziebarth <untix 2008-01-15 00:57:24
Automated m Mathias Ziebarth <untix 2008-01-15 00:54:03
doc changes Mathias Ziebarth <untix 2008-01-15 00:53:59
Automated m Lucas Lürich <lucas@l 2008-01-15 00:53:34
color instead of Mathias Ziebarth <untix 2008-01-15 00:36:48
Automated merg Mathias Ziebarth <untix 2008-01-15 00:34:55
tests and todo Mathias Ziebarth <untix 2008-01-15 00:33:50
Automated merg Manuel Holtgrewe <pur 2008-01-15 00:58:44
ein doku todo Joachim Breitner <mail 2008-01-15 00:48:52
operator regeln, Joachim Breitner <mail 2008-01-15 00:44:06
Automated merge with Joachim Breitner <mail 2008-01-15 00:19:47
7down colorized Joachim Breitner <mail 2008-01-15 00:18:49
gui fertig dokumen Joachim Breitner <mail 2008-01-15 00:13:31
MultipleOfRule Manuel Holtgrewe <pur 2008-01-15 00:56:59
Automated merge Manuel Holtgrewe <pur 2008-01-15 00:07:08
CrossProductRule Manuel Holtgrewe <pur 2008-01-15 00:06:51

Unittests

```
~/projekte/7down $ ./tests/runtests.py -v
[...]
apply() on a number which does not have an isDone() Digit. ... ok
Test that UniqueDigitsRule objects are properly converted to ... o
testIt (test_utils.TestCubeNumbers) ... ok
testIt (test_utils.TestSquareNumbers) ... ok
testDigits (test_utils.TestUtilityFunctions) ... ok
testFactorsOf (test_utils.TestUtilityFunctions) ... ok
testGcd (test_utils.TestUtilityFunctions) ... ok
testGcdList (test_utils.TestUtilityFunctions) ... ok

----------------------------------------------------------------------
Ran 168 tests in 0.157s

OK
```

Systemtest

```
~/projekte/7down $ ./tests/system_test.py
Running all tests within '~/projekte/7down/tests/examples'...
          OK (0.0378s; simple5.in)
          OK (2.9086s; zeit-07-42.in)
          OK (0.0332s; book.in)
          OK (0.7910s; prim10mal10mitGleichheit.in)
          OK (1.7817s; unsolvable2.in)
          OK (0.0447s; pal_u.in)
          OK (0.5549s; pyramid.in)
          OK (0.0906s; prim10-test.in)
          OK (1.9494s; zeit-07-38.in)
          OK (2.7520s; kzr-test2.in)
          OK (1.1250s; kzr.in)
          OK (0.0086s; reverseAndZero.in)
          OK (2.4786s; prim10mal10.in)
          OK (0.0014s; kzr-gi2.in)
          OK (0.7250s; zeit-07-35.in)
          OK (0.0012s; kzr-gi.in)
[... 11 Zeilen ausgelassen ...]
          OK (1.8311s; prim7mal7.in)
          OK (0.0024s; prim-test.in)
          OK (0.0092s; simple3.in)
```

Statistiken

	NOTSOLVED			65	17	1	206	17	0
	FAILED			231	17	1	239	17	0
	651	480	2	200	17	1	231	17	0
	FAILED			219	17	1	199	17	0
	FAILED			249	17	1	207	17	0
	FAILED			207	17	1	NOTSOLVED		
	FAILED			228	17	1	FAILED		
	FAILED			207	17	1	231	19	0
	FAILED			220	17	1	FAILED		
	FAILED			247	17	1	219	19	0
	FAILED			199	17	1	229	24	0
	FAILED			67	17	1	231	19	0
	FAILED			231	17	1	231	16	0
	FAILED			199	17	1	199	18	0
	FAILED			231	17	1	227	16	0
	FAILED			70	17	1	67	12	0
	FAILED			219	17	1	199	12	0
	438	591	2	68	17	1	251	12	0

	831	575	2	235	17	1	236	13	0
	828	614	2	290	17	1	432	12	0
	831	575	2	235	17	1	235	13	0
)7-35.in				kzr-gi.in					
	Stack			Ordered			Stack		
	ms	a	r	ms	a	r	ms	a	r

Abstraktion

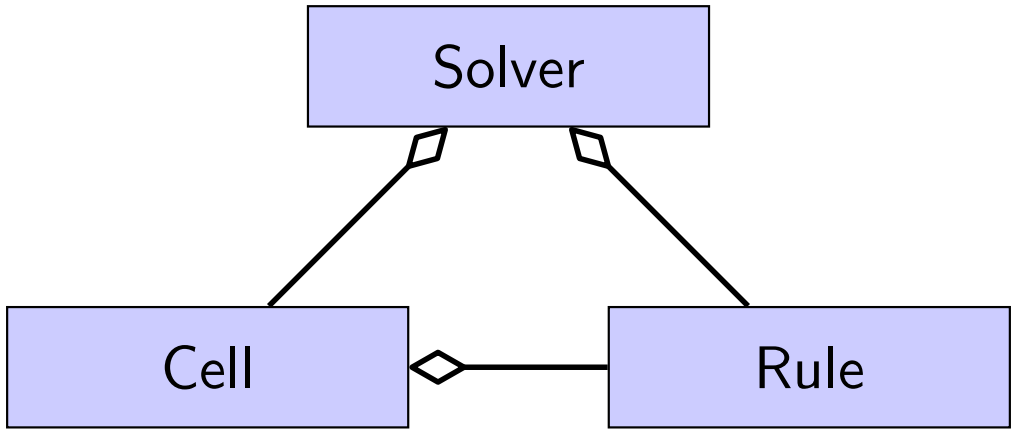
Problemübersetzung

Problemstellung		*Ansicht des Solvers*
Ziffern und Zahlen	→	Mengen („Zellen“)
feste Werte	→	einelementige Mengen
Regeln	→	schränken Mengen ein

Abstraktion

Problemübersetzung

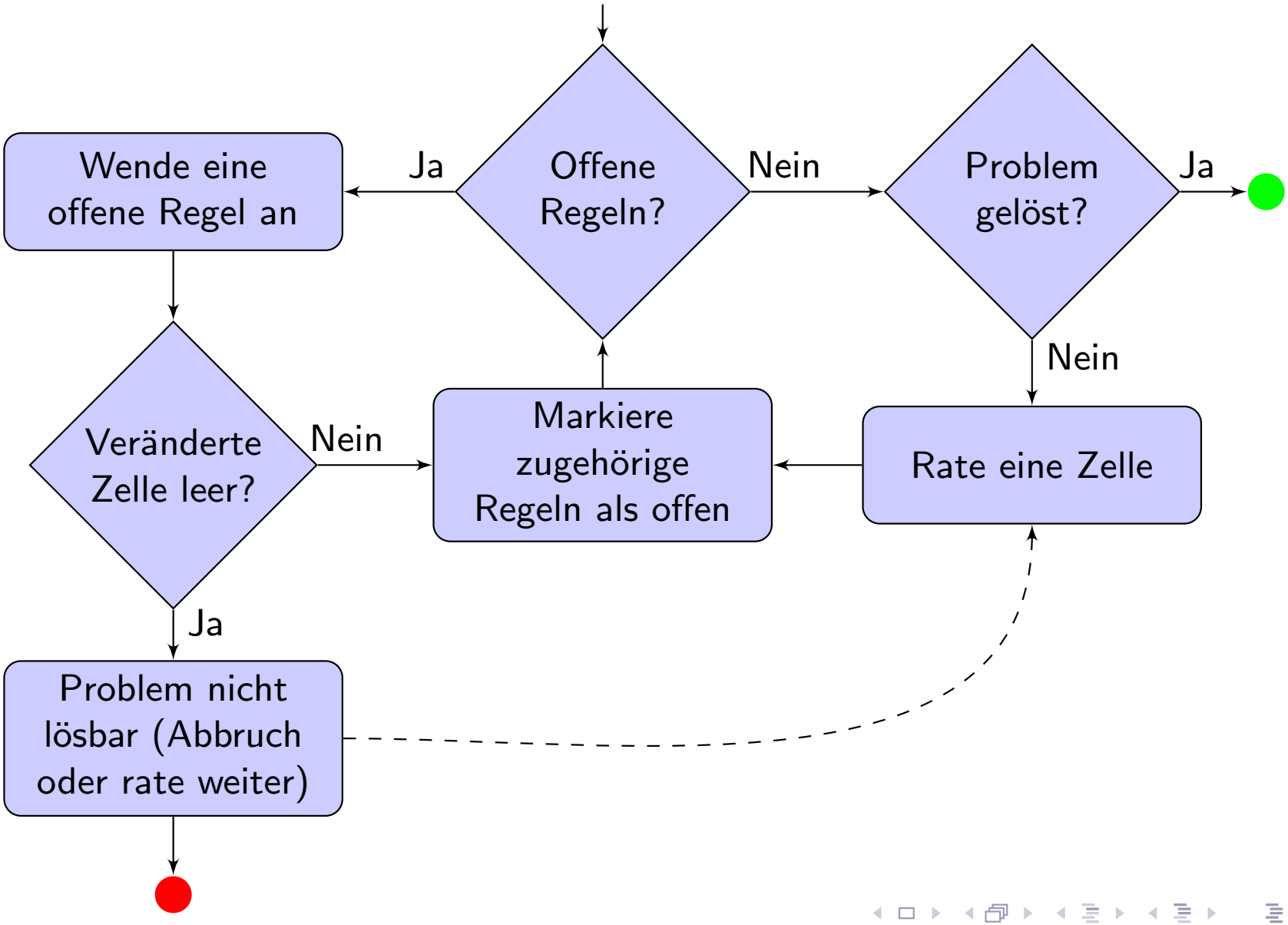
Problemstellung		*Ansicht des Solvers*
Ziffern und Zahlen	→	Mengen („Zellen“)
feste Werte	→	einelementige Mengen
Regeln	→	schränken Mengen ein



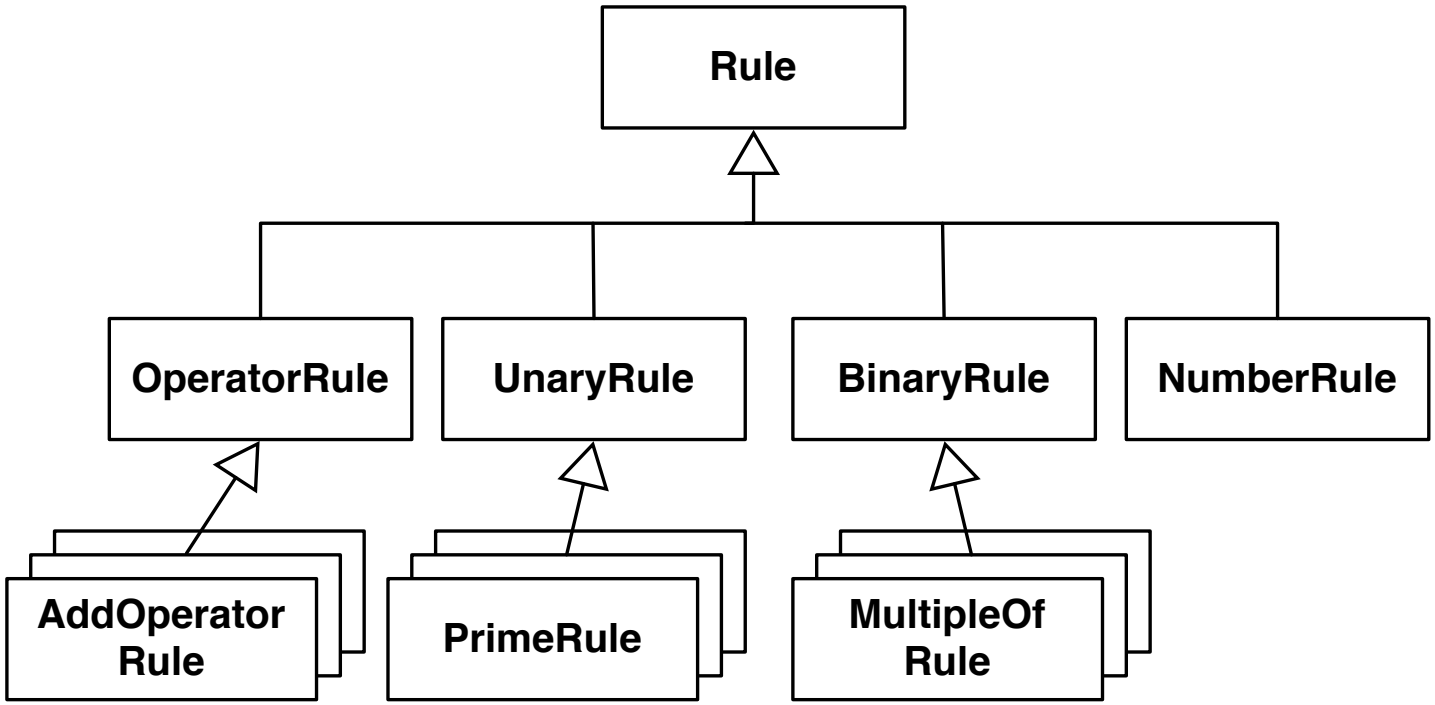
Eine Evolution von Solvern

- **StupidSolver**
  Noch ohne Backtracking
- **SimpleSolver**
  Vollständig, aber unperformant
- **OrderedSolver**
  Weiß, welche Regeln überhaupt sinnvoll sind
- **StackSolver**
  Ermöglicht Interaktion mit der GUI

Arbeitsweise des Solvers



Klassenarbeiten



Stolperstein

Das Problem:

```
~/projekte/7down $ python -c 'print hash(object())'
-1210219448
~/projekte/7down $ python -c 'print hash(object())'
-1209981880
```

Unsere Lösung:

```
class Rule(object):

# Mehr Methoden hier...

    def __hash__(self):
        return hash(self.name)
```

Stolperstein

Das Problem:

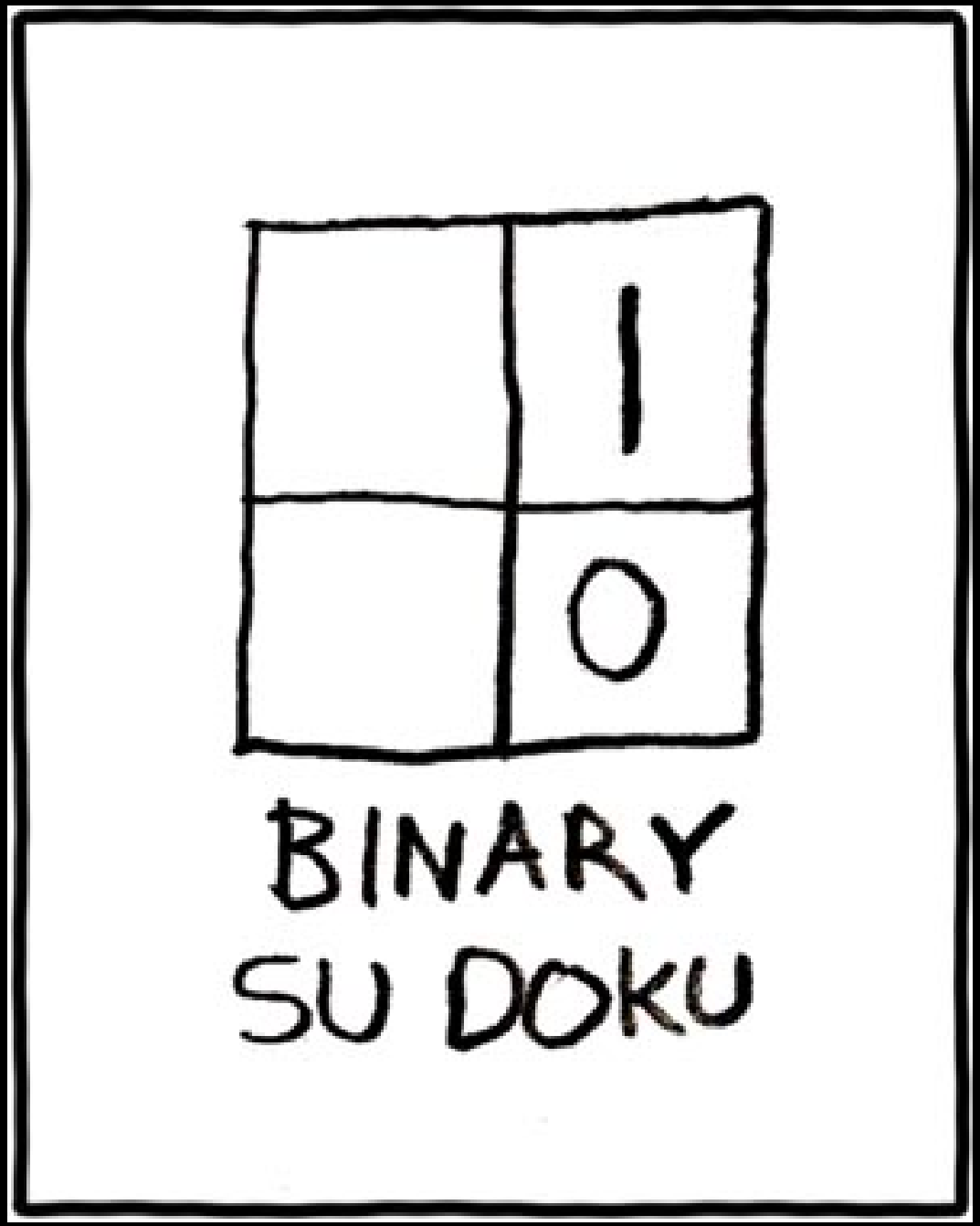
```
~/projekte/7down $ python -c 'print hash(object())'
-1210219448
~/projekte/7down $ python -c 'print hash(object())'
-1209981880
```

Unsere Lösung:

```
class Rule(object):

# Mehr Methoden hier...

    def __hash__(self):
        return hash(self.name)
```


1
0
BINARY
SU DOKU