Assignment 2

Automated Reasoning in AI 2011

Armon Toubman, Torec Luik

May 31, 2011

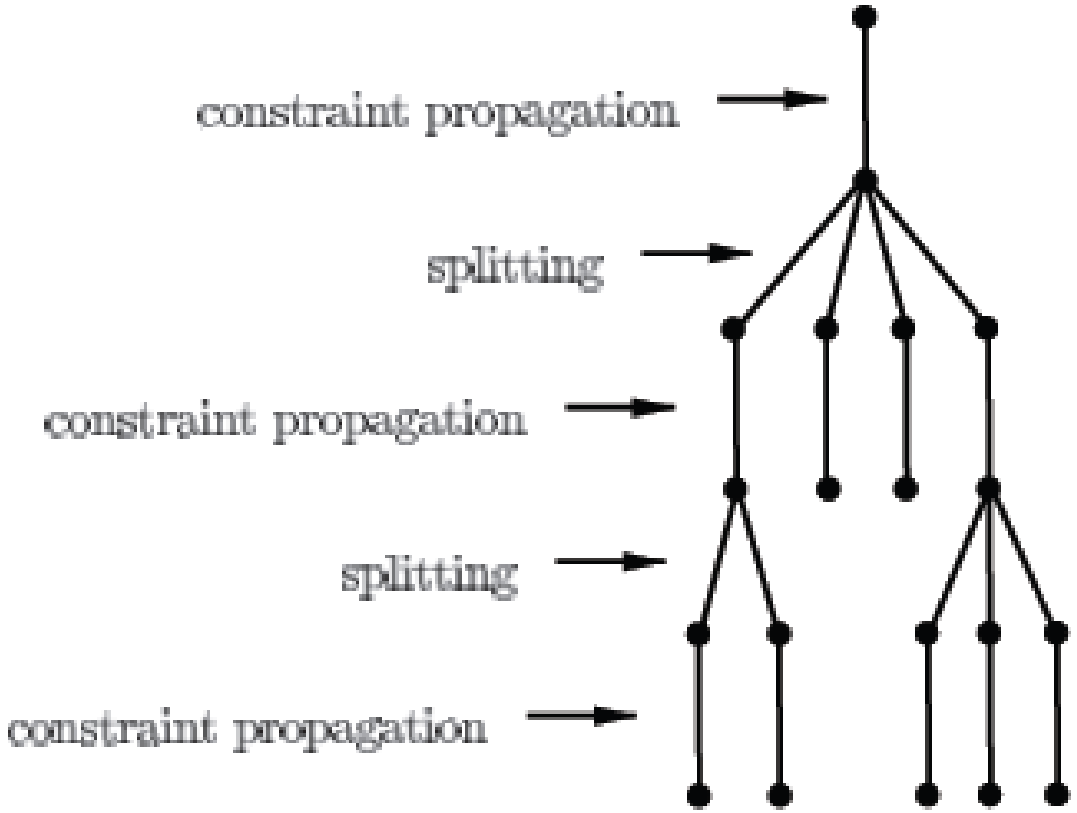
Sudoku

	9	4				1	3	
				7	6			2
	8			1				
	3	2						
			2				6	
				5		4		
					8			7
		6	3		4			8

Sudoku as CSP

- Variables : Unassigned cells.
- Assignment : Assigned cells.
- Domains : Values 1 to 9.
- Constraints : Values 1 to 9 once in every row, column, region.
- Consistency : Constraints not violated.
- Termination : No variables left.

CSP as tree


constraint propagation
splitting
constraint propagation
splitting
constraint propagation

Techniques

Tree search

We start with the basic Depth-First Backtracking algorithm.
We add:

- Heuristics
- Constraint Propagation

Heuristics

- From Depth-First to Best-First!
- Heuristics
  - H1: Smallest domain
  - H3: Most constrained
  - H13: Combine H1 & H3

Constraint propagation

- General
  - Revise
- Specific
  - Hidden Singles
  - Naked Pairs
  - Hidden Pairs

Revise

- Arc consistency
  - Check constraints between multiple variables
- Revise
  - Remove incompatible values from domains
- AC-3
  - Repeat revise (intelligently) until there is no more domain reduction

Revise in action

{1, 2, 3, 4, 5, 6, 7, 8, 9}	9	4				1	3	

{2, 5, 6, 7, 8}	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 8}

Revise in action

{1, 2, 3, 4, 5, 6, 7, 8, 9}	9	4				1	3	

{2, 5, 6, 7, 8}	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 8}

Hidden Singles

{2, 5, 6, 7, 8}	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 8}

7	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 8}

Hidden Singles

{2, 5, 6, 7, 8}	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 8}

7	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 8}

Naked Pairs

7	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 8}

7	9	4	{2, 5}	{2, 5}	6	1	3	8

Naked Pairs

7	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 8}

7	9	4	{2, 5}	{2, 5}	6	1	3	8

Hidden Pairs

{2, 5, 6, 7, 8}	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 6, 7, 8}

{7, 8}	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{7, 8}

Hidden Pairs

{2, 5, 6, 7, 8}	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{2, 5, 6, 7, 8}

{7, 8}	9	4	{2, 5}	{2, 5}	{2, 5, 6}	1	3	{7, 8}

Results

Results: Optimizations

Revise	Hidden Singles	Hidden Pairs	Naked Pairs	sudoku_training	top95
				9m49s	*
x				8m56s	1h55m25s
x	x	x	x	25s	10s
x	x			29s	47s
x	x		x	23s	21s
x	x	x		28s	16s
x		x		8m9s	26m37s
x		x	x	1m21s	3m51s
x			x	1m17s	19m32s

Results: Heuristics

Revise	Heuristic 1	Heuristic 3	sudoku_training	top95
x			8m56s	1h55m25s
x	x		1m45s	5m37s
x		x	2m18s	9m11s
x	x	x	1m45s	9m9s

Results: Optimization + Heuristics

Revise	Hidden Singles	Heuristic 1	Heuristic 3	sudoku_training	top95
x	x			29s	47s
x	x	x		25s	30s
x	x		x	27s	37s
x	x	x	x	26s	26s

Best results

Everything	sudoku_training	top95
x	20s	7s
x	8s $*$	3s $*$

Conclusions

- We created a CSP solver for the Sudoku game in Java
- We optimized it using well-known Sudoku specific techniques
  - Hidden Singles
  - Naked Pairs
  - Hidden Pairs
- & general heuristics
  - H1 : Smallest domain
  - H3 : Most constrained
  - H13 : H1 & H3 combined
- And we got great results with all of these!