Azucar: A SAT-based CSP solver using compact order encoding (Tool Presentation)

Tomoya Tanjo[1] Naoyuki Tamura[2] Mutsunori Banbara[2]

[1]Transdisciplinary Research Integration Center, Japan

[2]Kobe University, Japan

SAT2012
June 17th, 2012

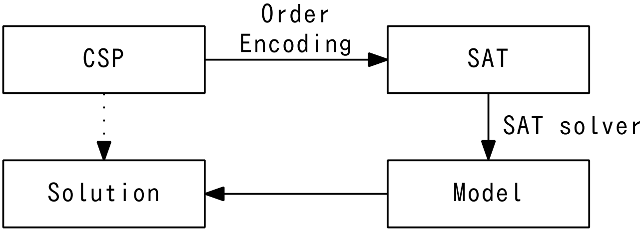
SAT-based system

Recently, SAT-based systems become applicable for solving hard and practical problems.

- Verification, planning, scheduling, test case generation, and Constraint Satisfaction Problems (CSP) [Biere 99, Kautz 06, Crawford *et al.* 94, Larrabee 92]
- A number of SAT-based CSP solver have been developed.

Solver	SAT encoding method
FznTini	Log encoding
SAT4J CSP	Direct and support encodings
Bumblebee	Direct and order encodings
Sugar	Order encoding

Sugar: a SAT-based CSP solver



- Sugar is a award-winning system of GLOBAL categories of the 2008 and 2009 International CSP Solver Competitions.
- It can handle finite linear CSP, COP[1], and Max-CSP over integers.
- The order encoding [Tamura *et al.* 09] used in Sugar shows a good performance for various applications [Tamura *et al.* 09, Soh *et al.* 10, Banbara *et al.* 10].

[1] Constraint Optimization Problem

Order Encoding

A comparison $x \le a$ is encoded by a different propositional variable $p(x \le a)$ for each integer variable x and integer value a.

Advantage

- The Unit Propagation in SAT solvers corresponds to the Bounds Propagation in CSP solvers.
- Tractable CSP can be encoded into tractable SAT [Petke *et al.* 11].

Disadvantage

- It generates too large SAT instances when the domain size of a CSP is large.
    - Each addition $z = x + y$ is encoded into $O(d^2)$ clauses[2].

[2] d: domain size of integer variables

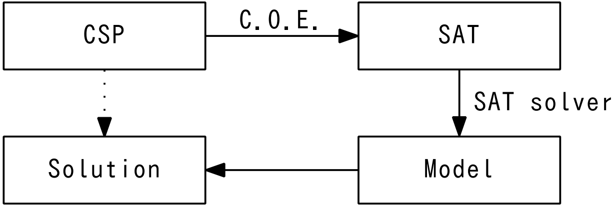
Compact Order Encoding (C.O.E.)

Compact order encoding (C.O.E.)

- **Using a numeral system of base $B \geq 2$.**
  - Each integer variable is represented by $m = \lceil \log_B d \rceil$ digits.
- **Encoding each digit by using the order encoding.**

- The Unit Propagation in SAT solvers corresponds to the Bounds Propagation in the most significant digit in CSP solvers.
- It requires much less clauses than the order encoding.
  - Each addition $z = x + y$ is encoded into $O(B^2 \log_B d)$ clauses.
- It is a generalization of the order and log encodings.
  - It is equivalent to the order encoding when $B \geq d$.
  - It is equivalent to the log encoding when $B = 2$.

Azucar: an enhancement version of Sugar



- It can handle finite non-linear CSP, COP[1], and Max-CSP over integers.
- It uses the compact order encoding (C.O.E.).

[1] Constraint Optimization Problem

Usage of Azucar

Example: Solving sample.csp ($B = \lceil \sqrt{d} \rceil$ is used by default.)

$ azucar sample.csp

- It can handle CSP written in Lisp-like format or XCSP format used in the CSP solver competitions.

Example: Using a numeral system of base $B = 10$.

$ azucar -b 10 sample.csp

- When choosing -b 2, Azucar uses the log encoding.

Example: Using a numeral system of base $B = \lceil \sqrt[3]{d} \rceil$

$ azucar -m 3 sample.csp

- When choosing -m 1, Azucar uses the order encoding.

CSC'09 (GLOBAL categories)

We also used large instances used in GLOBAL categories of the 2009 CSP Solver Competition. We selected 41 instances in which domain sizes are larger than 1000.

SAT/UNSAT	#Instances	Sugar	Azucar $m = 2$
SAT	21	21 (5.39)	21(5.99)
UNSAT	20	20 (36.20)	20 (18.30)
Total	41	41 (13.65)	41 (10.33)

- For UNSAT instances, Azucar solved about 2 times faster than Sugar.

Demonstration: SAT $\times$ COE = AZUCAR

$$
\begin{array}{cccccc}
 & & & S & A & T \\
\times & & & C & O & E \\
\hline
A & Z & U & C & A & R
\end{array}
$$

- Each letter corresponds to a unique one digit number.
- The objective is to find a assignment that satisfies SAT $\times$ COE $=$ AZUCAR.

Conclusion

- We introduced a SAT-based CSP solver Azucar which is an enhancement version of Sugar.
- Azucar uses the compact order encoding and can handle large domain CSP.
- It is published in open source license:

URL: http://code.google.com/p/azucar-solver/

Future work

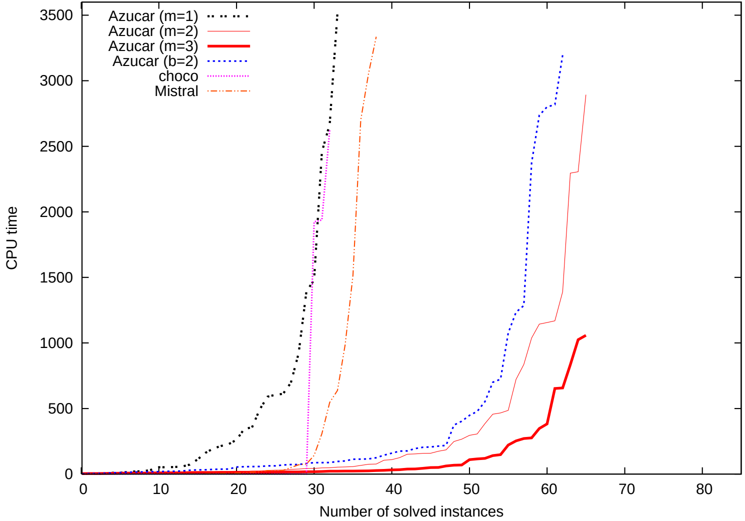
- Choose an appropriate base B (or the number of digit m)
- Support big integers.
- Extend the compact order encoding to fixed point number.

Evaluation (OSS): the number of solved instances

To evaluate the scalability and efficiency of Azucar, we used 85 Open-Shop Scheduling (OSS) problems with large domain sizes.

Domain size d	#Ins	Azucar $m=1$	Azucar $m=2$	Azucar $m=3$	Azucar $b=2$	choco	Mistral
10^3	17	14	14	14	13	6	14
10^4	17	12	13	13	13	6	12
10^5	17	8	13	13	12	7	7
10^6	17	0	14	13	12	7	4
10^7	17	0	12	13	13	7	2
Total	85	34	66	66	63	33	39

Evaluation: comparison of CPU times



CSC09 (GLOBAL categories)

We use 556 instances used in GLOBAL categories of the 2009 CSP Solver Competition.

Domain size	#Instances	Sugar	Azucar $m = 2$
0-99	476	429	382
100-199	25	6	3
200-299	4	0	0
300-399	2	0	0
400-499	2	0	0
500-599	4	0	0
600-799	0	-	-
800-899	2	0	0
900-999	0	-	-
1000-1099	1	1	1
1100-84499	0	-	-
84500-84599	40	40	40
Total	556	476	426

CSC09 (GLOBAL categories): Large instances

We compared the average CPU times of Sugar and Azucar.

SAT/UNSAT	#Instances	Sugar	Azucar ($m = 2$)
SAT	20	5.56	6.30
UNSAT	20	36.20	18.30
Total	40	14.18	10.74

- Azucar solved UNSAT instances about 2 times faster than Sugar on average.

Summary of C.O.E.

Let d be the domain size of integer variables, B be the base, and $m = \lceil \log_B d \rceil$ be the number of digits.

	Order ($B \geq d$)	C.O.E.	Log ($B = 2$)
Representation of integers	Unary	Base B	Binary
Size of SAT instances	Large	⟷	Small
$z = x + y$	$O(d^2)$	$O(mB^2)$	$O(\log_2 d)$
$z = xy$	$O(d^2)$	$O(mB^3 + m^2B^2)$	$O(\log_2^2 d)$
Propagation	Fast	⟷	Slow
#carry ripples	0	$O(m)$	$O(\log_2 d)$

- Columns "$z = x + y$", "$z = xy$" means the number of clauses to encode each constraint.
- "#carry ripples" means the number of carry ripples to detect a inconsistency.

Summary of C.O.E.

Let d be the domain size of integer variables, B be the base, and $m = 2$ be the number of digits.

	Order ($B \geq d$)	C.O.E. ($B = \lceil\sqrt{d}\rceil$)	Log ($B = 2$)
Representation of integers	Unary	Base B	Binary
Size of SAT instances	Large	⟷	Small
$z = x + y$	$O(d^2)$	$O(d)$	$O(\log_2 d)$
$z = xy$	$O(d^2)$	$O(\sqrt{d^3})$	$O(\log_2^2 d)$
Propagation	Fast	⟷	Slow
#carry ripples	0	$O(1)$	$O(\log_2 d)$

- Columns "$z = x + y$", "$z = xy$" means the number of clauses to encode each constraint.
- "#carry ripples" means the number of carry ripples to detect a inconsistency.

Order encoding

- A comparison $x \le a$ is encoded by a different propositional variable $p(x \le a)$ for each integer variable x and integer value a.

$$p(x \le a) \iff x \le a$$

- For example, the following four propositional variables are used to encode an integer variable $x \in \{2..6\}$,

Propositional variables for $x \in \{2..6\}$

$$p(x \le 2) \quad p(x \le 3) \quad p(x \le 4) \quad p(x \le 5)$$

- Propositional variable $p(x \le 6)$ is unnecessary since $x \le 6$ is always true.

Order encoding

- The following clauses are required to a make $p(x \leq a)$ be true iff $x \leq a$.

3 clauses for $x \in \{2,3,4,5,6\}$

$$\neg p(x \leq 2) \vee p(x \leq 3)$$
$$\neg p(x \leq 3) \vee p(x \leq 4)$$
$$\neg p(x \leq 4) \vee p(x \leq 5)$$

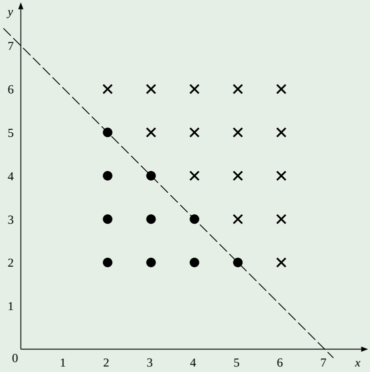
- The following table shows possible satisfiable assignments for the given clauses.

$p(x \leq 2)$	$p(x \leq 3)$	$p(x \leq 4)$	$p(x \leq 5)$	Interpretation
T	T	T	T	$x = 2$
F	T	T	T	$x = 3$
F	F	T	T	$x = 4$
F	F	F	T	$x = 5$
F	F	F	F	$x = 6$

Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \leq 7$ $(x, y \in \{2..6\})$

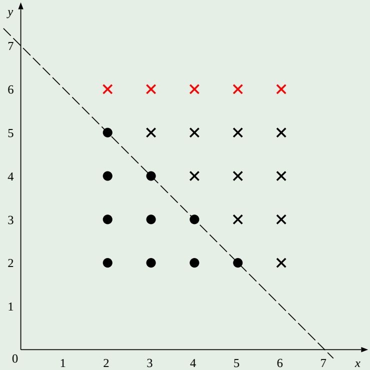


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \leq 7$ $(x, y \in \{2..6\})$

$\neg(y \geq 6)$

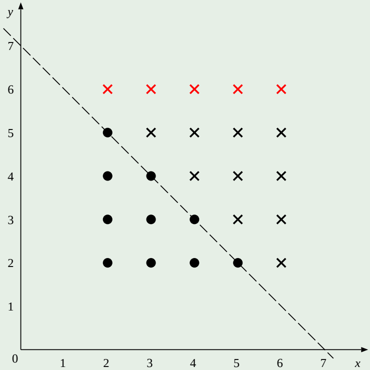


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \le 7$ ($x, y \in \{2..6\}$)

$p(y \le 5)$

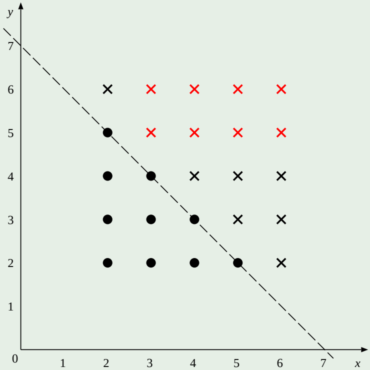


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \leq 7$ $(x, y \in \{2..6\})$

$p(y \leq 5)$
$\neg(x \geq 3 \wedge y \geq 5)$

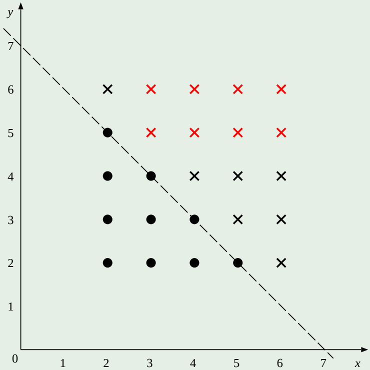


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \leq 7$ $(x, y \in \{2..6\})$

$p(y \leq 5)$
$p(x \leq 2) \vee p(y \leq 4)$

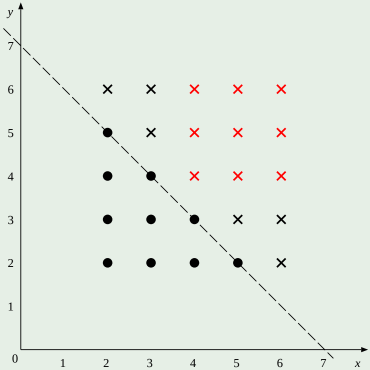


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \leq 7$ ($x, y \in \{2..6\}$)

$p(y \leq 5)$
$p(x \leq 2) \vee p(y \leq 4)$
$\neg(x \geq 4 \wedge y \geq 4)$

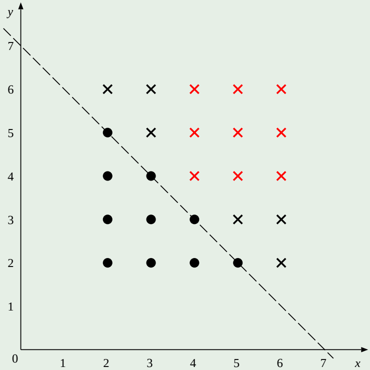


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \le 7$ $(x, y \in \{2..6\})$

$p(y \le 5)$
$p(x \le 2) \vee p(y \le 4)$
$p(x \le 3) \vee p(y \le 3)$

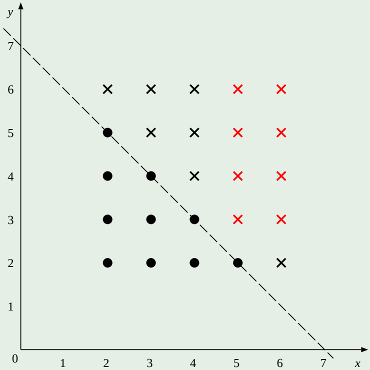


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \leq 7$ $(x, y \in \{2..6\})$

$p(y \leq 5)$
$p(x \leq 2) \vee p(y \leq 4)$
$p(x \leq 3) \vee p(y \leq 3)$
$\neg(x \geq 5 \wedge y \geq 3)$

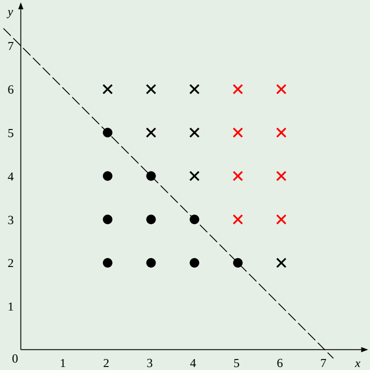


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \leq 7$ ($x, y \in \{2..6\}$)

$$
\begin{aligned}
&p(y \leq 5) \\
&p(x \leq 2) \vee p(y \leq 4) \\
&p(x \leq 3) \vee p(y \leq 3) \\
&p(x \leq 4) \vee p(y \leq 2)
\end{aligned}
$$

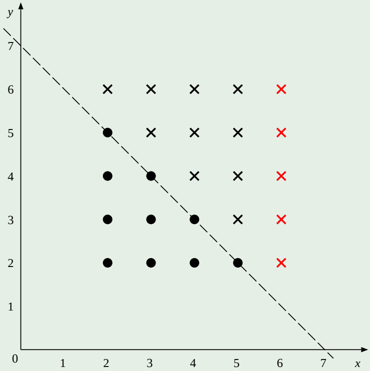


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \leq 7$ ($x, y \in \{2..6\}$)

$p(y \leq 5)$
$p(x \leq 2) \vee p(y \leq 4)$
$p(x \leq 3) \vee p(y \leq 3)$
$p(x \leq 4) \vee p(y \leq 2)$
$\neg(x \geq 6)$

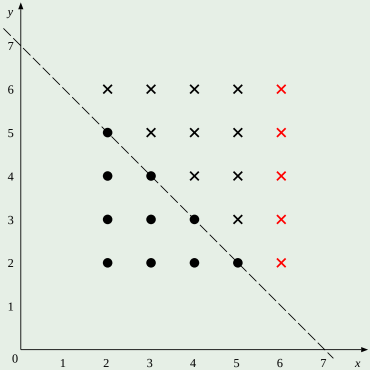


Order encoding

A constraint is encoded by enumerating its conflict regions.

Encoding a constraint $x + y \le 7$ ($x, y \in \{2..6\}$)

$p(y \le 5)$
$p(x \le 2) \vee p(y \le 4)$
$p(x \le 3) \vee p(y \le 3)$
$p(x \le 4) \vee p(y \le 2)$
$p(x \le 5)$



Bounds Propagation in the order encoding

Encoding a constraint $x + y \le 7$ ($x, y \in \{2..6\}$)

$$
\begin{array}{rc}
C_1 : & p(y \le 5) \\
C_2 : & p(x \le 2) \lor p(y \le 4) \\
C_3 : & p(x \le 3) \lor p(y \le 3) \\
C_4 : & p(x \le 4) \lor p(y \le 2) \\
C_5 : & p(x \le 5)
\end{array}
$$

- When $p(x \le 3)$ becomes false (i.e. $x \ge 4$), $p(y \le 3)$ becomes true (i.e. $y \le 3$) by Unit Propagation for C_3.
- This corresponds to the Bounds Propagation in CSP solvers.

Example: $x \in \{0..999\}$

$$x \in \{0..999\}$$

When we choose $B = 10$, the compact order encoding uses three digits to represent x (x_2 is a most significant digit).

Three digits to represent $x \in \{0..999\}$

$$x_2 \in \{0..9\} \qquad x_1 \in \{0..9\} \qquad x_0 \in \{0..9\}$$

Each variable x_i is encoded into SAT by using the order encoding.

Example: $z = x + 123\ (x, z \in \{0..999\})$

$$z = x + 123\ (x, z \in \{0..999\})$$

The compact order encoding decomposes $z = x + 123$ into digit-wise constraints.

Decomposition of $z = x + 123$

$$\begin{array}{ll} & (z_2 = x_2 + 1 + c_1) \\ \wedge & (10c_1 + z_1 = x_1 + 2 + c_0) \\ \wedge & (10c_0 + z_0 = x_0 + 3) \end{array}$$

$$\begin{array}{cccc} & x_2 & x_1 & x_0 \\ + & 1 & 2 & 3 \\ \hline & z_2 & z_1 & z_0 \end{array}$$

Each $c_i \in \{0, 1\}$ $(i \in \{0, 1\})$ represents the carry digit from the i-th additions.

Finally, each constraint is encoded into SAT by using the order encoding.

Example: $x \le 123$ $(x \in \{0..999\})$

The compact order encoding decomposes $x \le 123$ into digit-wise comparisons.

Decomposition of $x \le 123$ with $B = 10$

$$x_2 \le 1 \land \left(x_2 = 1 \Rightarrow \sum_{i=0}^{1} 10^i x_i \le 23 \right)$$

Each digit-wise comparison is encoded into SAT by using order encoding.

Example: $x \le 123$ $(x \in \{0..999\})$

The compact order encoding decomposes $x \le 123$ into digit-wise comparisons.

Decomposition of $x \le 123$ with $B = 10$

$$x_2 \le 1 \land (x_2 = 1 \Rightarrow (x_1 \le 2 \land (x_1 = 2 \Rightarrow x_0 \le 3)))$$

Each digit-wise comparison is encoded into SAT by using order encoding.

Example of the Unit Propagation

- We will show the Unit Propagation in Order encodingand Compact order encoding($m = 10$) by using the following example.
- This example is unsatisfiable.

Example of CSP ($x, y \in \{0..999\}$)

$$x \geq 643$$
$$y \leq 802$$
$$x + 160 \leq y$$

Order encoding: Encoding constraints

The constraints $x \geq 643$ and $y \leq 802$ are encoded as follows:

Encoding a constraint $x \geq 643$

$$\neg p(x \leq 642)$$

Encoding a constraint $y \leq 802$

$$p(y \leq 802)$$

Order encoding: Encoding constraints

Encoding a constraint $x + 160 \le y$

$$\neg p(y \le 159)$$
$$p(x \le 0) \lor \neg p(y \le 160)$$
$$p(x \le 1) \lor \neg p(y \le 161)$$
$$\dots$$
$$p(x \le 838) \lor \neg p(y \le 998)$$
$$p(x \le 839)$$

Unit Propagation in Order encoding

Encoding $x + 160 \le y$, $x \ge 643$, and $y \le 802$

$$\neg p(x \le 642)$$
$$p(y \le 802)$$
$$\neg p(y \le 159)$$
$$p(x \le 0) \lor \neg p(y \le 160)$$
$$p(x \le 1) \lor \neg p(y \le 161)$$
$$\dots$$
$$p(x \le 642) \lor \neg p(y \le 802)$$
$$\dots$$
$$p(x \le 838) \lor \neg p(y \le 998)$$
$$p(x \le 839)$$

Unit Propagation in Order encoding

Encoding $x + 160 \leq y$, $x \geq 643$, and $y \leq 802$

$$\neg p(x \leq 642)$$
$$p(y \leq 802)$$
$$\neg p(y \leq 159)$$
$$p(x \leq 0) \vee \neg p(y \leq 160)$$
$$p(x \leq 1) \vee \neg p(y \leq 161)$$
$$\ldots$$
$$p(x \leq 642) \vee \neg p(y \leq 802)$$
$$\ldots$$
$$p(x \leq 838) \vee \neg p(y \leq 998)$$
$$p(x \leq 839)$$

Unit Propagation in Order encoding

Encoding $x + 160 \leq y$, $x \geq 643$, and $y \leq 802$

$$\neg p(x \leq 642)$$
$$p(y \leq 802)$$
$$\neg p(y \leq 159)$$
$$p(x \leq 0) \vee \neg p(y \leq 160)$$
$$p(x \leq 1) \vee \neg p(y \leq 161)$$
$$\ldots$$
$$p(x \leq 642) \vee \neg p(y \leq 802)$$
$$\ldots$$
$$p(x \leq 838) \vee \neg p(y \leq 998)$$
$$p(x \leq 839)$$

- The order encoding can detect a conflict without decisions.

Compact order encoding: Decompose constraints

Decomposition of $x \geq 643$

$$x^{(2)} \geq 6$$
$$x^{(2)} \leq 6 \Rightarrow x^{(1)} \geq 4$$
$$(x^{(2)} \leq 6 \wedge x^{(1)} \leq 4) \Rightarrow x^{(0)} \geq 3$$

Decomposition of $y \leq 802$

$$y^{(2)} \leq 8$$
$$y^{(2)} \leq 7 \vee y^{(1)} \leq 0$$
$$y^{(2)} \leq 7 \vee y^{(0)} \leq 2$$

Compact order encoding: Decompose constraints

Decomposition of $x \geq 643$

$$x^{(2)} \geq 6$$
$$x^{(2)} \geq 7 \lor x^{(1)} \geq 4$$
$$(x^{(2)} \leq 6 \land x^{(1)} \leq 4) \Rightarrow x^{(0)} \geq 3$$

Decomposition of $y \leq 802$

$$y^{(2)} \leq 8$$
$$y^{(2)} \leq 7 \lor y^{(1)} \leq 0$$
$$y^{(2)} \leq 7 \lor y^{(0)} \leq 2$$

Compact order encoding: Decompose constraints

Decomposition of $x \geq 643$

$$x^{(2)} \geq 6$$
$$x^{(2)} \geq 7 \lor x^{(1)} \geq 4$$
$$x^{(2)} \geq 7 \lor x^{(1)} \geq 5 \lor x^{(0)} \geq 3$$

Decomposition of $y \leq 802$

$$y^{(2)} \leq 8$$
$$y^{(2)} \leq 7 \lor y^{(1)} \leq 0$$
$$y^{(2)} \leq 7 \lor y^{(0)} \leq 2$$

Compact order encoding: Decompose constraints

A constraint $x + 160 \le y$ is decomposed by considering digit-wise additions and comparisons. (c is a carry bit from $x^{(1)} + 6$).

Decomposition of $x + 160 \le y$ (1)

$$(x^{(1)} + 6 \ge 10) \Leftrightarrow (c = 1)$$
$$x^{(2)} + 1 + c \le y^{(2)}$$
$$x^{(2)} + 1 + c \le y^{(2)} - 1 \lor x^{(1)} + 6 \le y^{(1)} + 10c$$
$$x^{(2)} + 1 + c \le y^{(2)} - 1 \lor x^{(1)} + 6 \le y^{(1)} + 10c - 1 \lor x^{(0)} \le y^{(0)}$$

Compact order encoding: Decompose constraints

Decomposition of $x + 160 \le y$ (1)

$$(x^{(1)} + 6 \ge 10) \Leftrightarrow (c = 1)$$
$$x^{(2)} + 1 + c \le y^{(2)}$$
$$p \vee q$$
$$p \vee r \vee s$$
$$\neg p \vee x^{(2)} + 1 + c \le y^{(2)} - 1$$
$$\neg q \vee x^{(1)} + 6 \le y^{(1)} + 10c$$
$$\neg r \vee x^{(1)} + 6 \le y^{(1)} + 10c - 1$$
$$\neg s \vee x^{(0)} \le y^{(0)}$$

Unit Propagation in Compact order encoding

Decomposition of $x + 160 \le y$, $x \ge 643$, and $y \le 802$

$$(x^{(1)} + 6 \ge 10) \Leftrightarrow (c = 1)$$

$$x^{(2)} \ge 6$$
$$x^{(2)} \ge 7 \vee x^{(1)} \ge 4$$
$$x^{(2)} \ge 7 \vee x^{(1)} \ge 5 \vee x^{(0)} \ge 3$$
$$y^{(2)} \le 8$$
$$y^{(2)} \le 7 \vee y^{(1)} \le 0$$
$$y^{(2)} \le 7 \vee y^{(0)} \le 2$$

$$x^{(2)} + 1 + c \le y^{(2)}$$
$$p \vee q$$
$$p \vee r \vee s$$
$$\neg p \vee x^{(2)} + 1 + c \le y^{(2)} - 1$$
$$\neg q \vee x^{(1)} + 6 \le y^{(1)} + 10c$$
$$\neg r \vee x^{(1)} + 6 \le y^{(1)} + 10c - 1$$
$$\neg s \vee x^{(0)} \le y^{(0)}$$

p	q	r	s	$x^{(2)}$	$x^{(1)}$	$x^{(0)}$	$y^{(2)}$	$y^{(1)}$	$y^{(0)}$
?	?	?	?	0..9	0..9	0..9	0..9	0..9	0..9

Unit Propagation in Compact order encoding

Decomposition of $x + 160 \le y$, $x \ge 643$, and $y \le 802$

$$(x^{(1)} + 6 \ge 10) \Leftrightarrow (c = 1)$$

$x^{(2)} \ge 6$	$x^{(2)} + 1 + c \le y^{(2)}$
$x^{(2)} \ge 7 \vee x^{(1)} \ge 4$	$p \vee q$
$x^{(2)} \ge 7 \vee x^{(1)} \ge 5 \vee x^{(0)} \ge 3$	$p \vee r \vee s$
$y^{(2)} \le 8$	$\neg p \vee x^{(2)} + 1 + c \le y^{(2)} - 1$
$y^{(2)} \le 7 \vee y^{(1)} \le 0$	$\neg q \vee x^{(1)} + 6 \le y^{(1)} + 10c$
$y^{(2)} \le 7 \vee y^{(0)} \le 2$	$\neg r \vee x^{(1)} + 6 \le y^{(1)} + 10c - 1$
	$\neg s \vee x^{(0)} \le y^{(0)}$

p	q	r	s	$x^{(2)}$	$x^{(1)}$	$x^{(0)}$	$y^{(2)}$	$y^{(1)}$	$y^{(0)}$
?	?	?	?	6..9	0..9	0..9	0..8	0..9	0..9

Unit Propagation in Compact order encoding

Decomposition of $x + 160 \le y$, $x \ge 643$, and $y \le 802$

$(x^{(1)} + 6 \ge 10) \Leftrightarrow (c = 1)$

$x^{(2)} \ge 6$

$x^{(2)} + 1 + c \le y^{(2)}$

$x^{(2)} \ge 7 \vee x^{(1)} \ge 4$

$p \vee q$

$x^{(2)} \ge 7 \vee x^{(1)} \ge 5 \vee x^{(0)} \ge 3$

$p \vee r \vee s$

$y^{(2)} \le 8$

$\neg p \vee x^{(2)} + 1 + c \le y^{(2)} - 1$

$y^{(2)} \le 7 \vee y^{(1)} \le 0$

$\neg q \vee x^{(1)} + 6 \le y^{(1)} + 10c$

$y^{(2)} \le 7 \vee y^{(0)} \le 2$

$\neg r \vee x^{(1)} + 6 \le y^{(1)} + 10c - 1$

$\neg s \vee x^{(0)} \le y^{(0)}$

p	q	r	s	$x^{(2)}$	$x^{(1)}$	$x^{(0)}$	$y^{(2)}$	$y^{(1)}$	$y^{(0)}$
?	?	?	?	6..7	0..9	0..9	7..8	0..9	0..9

Unit Propagation in Compact order encoding

Decomposition of $x + 160 \le y$, $x \ge 643$, and $y \le 802$

$(x^{(1)} + 6 \ge 10) \Leftrightarrow (c = 1)$

$x^{(2)} \ge 6$ $\quad$ $x^{(2)} + 1 + c \le y^{(2)}$

$x^{(2)} \ge 7 \lor x^{(1)} \ge 4$ $\quad$ $p \lor q$

$x^{(2)} \ge 7 \lor x^{(1)} \ge 5 \lor x^{(0)} \ge 3$ $\quad$ $p \lor r \lor s$

$y^{(2)} \le 8$ $\quad$ $\neg p \lor x^{(2)} + 1 + c \le y^{(2)} - 1$

$y^{(2)} \le 7 \lor y^{(1)} \le 0$ $\quad$ $\neg q \lor x^{(1)} + 6 \le y^{(1)} + 10c$

$y^{(2)} \le 7 \lor y^{(0)} \le 2$ $\quad$ $\neg r \lor x^{(1)} + 6 \le y^{(1)} + 10c - 1$

$\neg s \lor x^{(0)} \le y^{(0)}$

p	q	r	s	$x^{(2)}$	$x^{(1)}$	$x^{(0)}$	$y^{(2)}$	$y^{(1)}$	$y^{(0)}$
?	?	?	?	6..7	0..9	0..9	7..8	0..9	0..9

- In this example, we need to assign a value to other variables to detect a conflict.