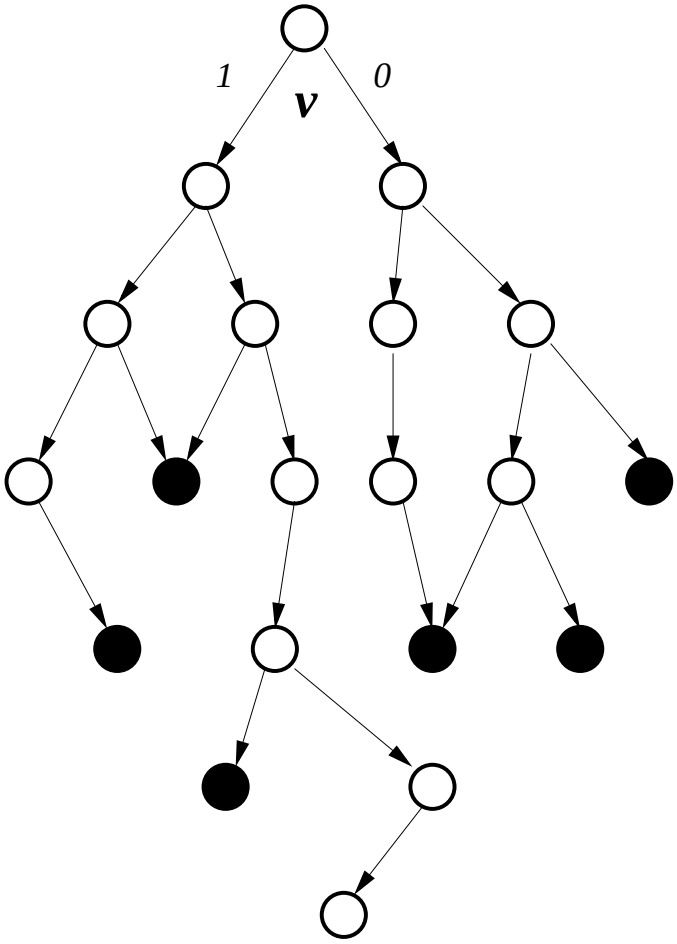
Resolution Tunnels for Improved SAT Solver Performance

A Search Space



Speed up search

Reduce Search Space:

- **Reduce breadth of search space by initially adding constraints that have no business being there**
- **At some search depth, when breadth is decreasing, remove the constraints**
- **Continue to the end**

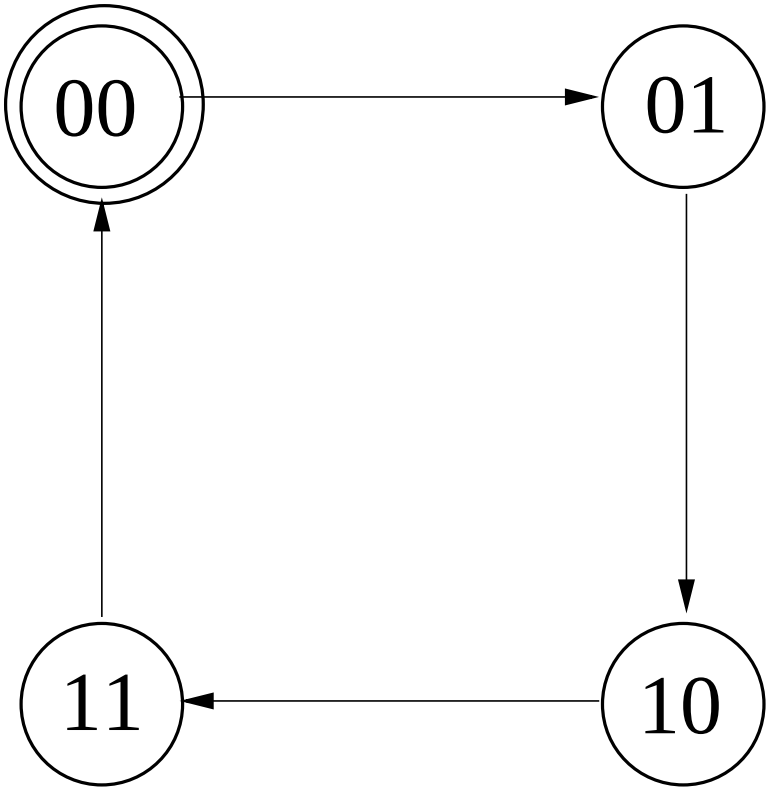
Speed up search

Reduce Search Space:

- **Reduce breadth of search space by initially adding constraints that have no business being there**
- **At some search depth, when breadth is decreasing, remove the constraints**
- **Continue to the end**

Notes:

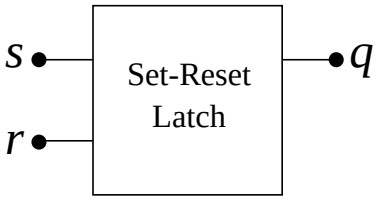
- **Obtain constaints by looking for patterns in *Solutions* to smaller instances of the same family (downplay formula structure)**
- **May fail to find an existing solution, cannot be applied to unsat formulas, but**

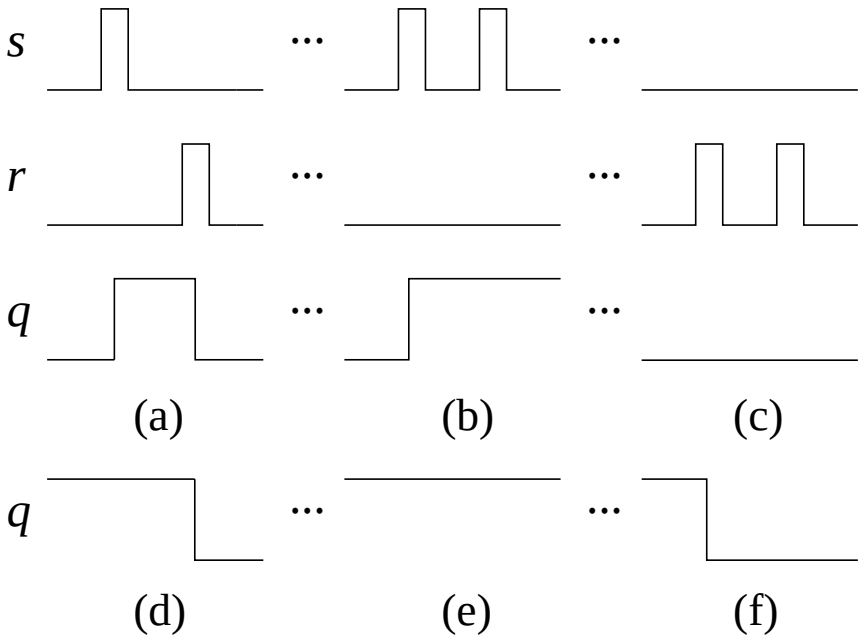


A state machine representing a sequential circuit

Circuits Have Interesting Properties

<u>Example of a Property</u>: For a latch, it is possible to have an output q value of `True` if both inputs `S` and `R` have value `True`.





Some Interesting Things to Prove

- For every path property P is *true* at the next time step.
- For every path property P is *true* at some future time step.
- For every path property P is *true* at every future time step.
- For every path property P is *true* until property Q is *true*.
- There exists a path such that property P is *true* at the next time step.
- There exists a path such that property P is *true* at some future time step.
- There exists a path such that property P is *true* at every future time step.
- There exists a path such that property P is *true* until property Q is *true*.

Can Do This With Boolean Expressions

How: The Boolean expression must have components which:

1. **force the property or properties of the time dependent expression to hold,**
2. **establish the starting state,**
3. **force legal transitions to occur.**

Can Do This With Boolean Expressions

How: The Boolean expression must have components which:

1. **force the property or properties of the time dependent expression to hold,**
2. **establish the starting state,**
3. **force legal transitions to occur.**

For reasonable size of expression need to bound number of time steps in which the time-dependent expression is to be verified; hence, *bounded model checking*.

Example

A two-bit counter (variables v_1^i, v_2^i):

Current Output		Next Output
00	**:**	**01**
01	**:**	**10**
10	**:**	**11**
11	**:**	**00**

Initial state: $v_1^0 = 0$, $v_2^0 = 0$

Example

A two-bit counter (variables v_1^i, v_2^i):

Current Output		Next Output
00	**:**	**01**
01	**:**	**10**
10	**:**	**11**
11	**:**	**00**

Initial state: $v_1^0 = 0$, $v_2^0 = 0$

Boolean description of state machine:

$$(v_2^{i+1} \equiv \neg v_2^i) \wedge (v_1^{i+1} \equiv v_1^i \oplus v_2^i).$$

Example

A two-bit counter (variables v_1^i, v_2^i):

Current Output		Next Output
00	**:**	**01**
01	**:**	**10**
10	**:**	**11**
11	**:**	**00**

Initial state: $v_1^0 = 0$, $v_2^0 = 0$

Boolean description of state machine:

$$(v_2^{i+1} \equiv \neg v_2^i) \wedge (v_1^{i+1} \equiv v_1^i \oplus v_2^i).$$

Property to prove:

Can the two-bit counter reach a count of 11 in exactly three time steps?

Assemble Boolean Expression

Force the property to hold:

$$(\neg(v_1^0 \wedge v_2^0) \wedge \neg(v_1^1 \wedge v_2^1) \wedge \neg(v_1^2 \wedge v_2^2) \wedge (v_1^3 \wedge v_2^3))$$

Assemble Boolean Expression

Force the property to hold:

$$(\neg(v_1^0 \wedge v_2^0) \wedge \neg(v_1^1 \wedge v_2^1) \wedge \neg(v_1^2 \wedge v_2^2) \wedge (v_1^3 \wedge v_2^3))$$

Express the starting state:

$$(\neg v_1^0 \wedge \neg v_2^0)$$

Assemble Boolean Expression

Force the property to hold:

$$(\neg(v_1^0 \wedge v_2^0) \wedge \neg(v_1^1 \wedge v_2^1) \wedge \neg(v_1^2 \wedge v_2^2) \wedge (v_1^3 \wedge v_2^3))$$

Express the starting state:

$$(\neg v_1^0 \wedge \neg v_2^0)$$

Force legal transitions (repetitions of the transition relation):

$$(v_2^1 \equiv \neg v_2^0) \wedge (v_1^1 \equiv v_1^0 \oplus v_2^0) \wedge (v_2^2 \equiv \neg v_2^1) \wedge (v_1^2 \equiv v_1^1 \oplus v_2^1) \wedge$$
$$(v_2^3 \equiv \neg v_2^2) \wedge (v_1^3 \equiv v_1^2 \oplus v_2^2)$$

Assemble Boolean Expression

Force the property to hold:

$$(\neg(v_1^0 \wedge v_2^0) \wedge \neg(v_1^1 \wedge v_2^1) \wedge \neg(v_1^2 \wedge v_2^2) \wedge (v_1^3 \wedge v_2^3))$$

Express the starting state:

$$(\neg v_1^0 \wedge \neg v_2^0)$$

Force legal transitions (repetitions of the transition relation):

$$(v_2^1 \equiv \neg v_2^0) \wedge (v_1^1 \equiv v_1^0 \oplus v_2^0) \wedge (v_2^2 \equiv \neg v_2^1) \wedge (v_1^2 \equiv v_1^1 \oplus v_2^1) \wedge$$
$$(v_3^3 \equiv \neg v_2^2) \wedge (v_1^3 \equiv v_1^2 \oplus v_2^2)$$

Satisfied *only* by

$$v_1^0 = 0, v_2^0 = 0, v_1^1 = 1, v_2^1 = 0, v_1^2 = 0, v_2^2 = 1, v_1^3 = 1, v_2^3 = 1$$

How to Solve It

Translate into CNF

$$(\bar{v}_0 \vee v_1 \vee \bar{v}_2) \wedge (\bar{v}_1 \vee \bar{v}_3) \wedge (v_0 \vee v_3 \vee \bar{v}_4 \vee \bar{v}_5) \wedge (v_3)$$

How to Solve It

Translate into CNF

$$(\bar{v}_0 \vee v_1 \vee \bar{v}_2) \wedge (\bar{v}_1 \vee \bar{v}_3) \wedge (v_0 \vee v_3 \vee \bar{v}_4 \vee \bar{v}_5) \wedge (v_3)$$

Easy to do. For example,

$$(v_1 \equiv v_2)$$

translates to

$$(v_1 \vee \bar{v}_2) \wedge (\bar{v}_1 \vee v_2)$$

Apply a SAT Solver

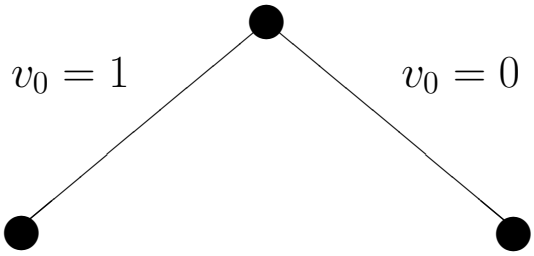
Sets:

$$\psi = \{\{\bar{v}_0, v_1, \bar{v}_2\}, \{\bar{v}_1, \bar{v}_3\}, \{v_0, v_3, \bar{v}_4, \bar{v}_5\}, \{v_3\}\}$$

Resolution:

$$\psi_v = \{c - \{\bar{v}\} : c \in \psi, v \notin c\}$$
$$\psi_{\bar{v}} = \{c - \{v\} : c \in \psi, \bar{v} \notin c\}$$

$$\{\{\bar{v}_0, v_1, \bar{v}_2\}, \{\bar{v}_1, \bar{v}_3\}, \{v_0, v_3, \bar{v}_4, \bar{v}_5\}, \{v_3\}\}$$



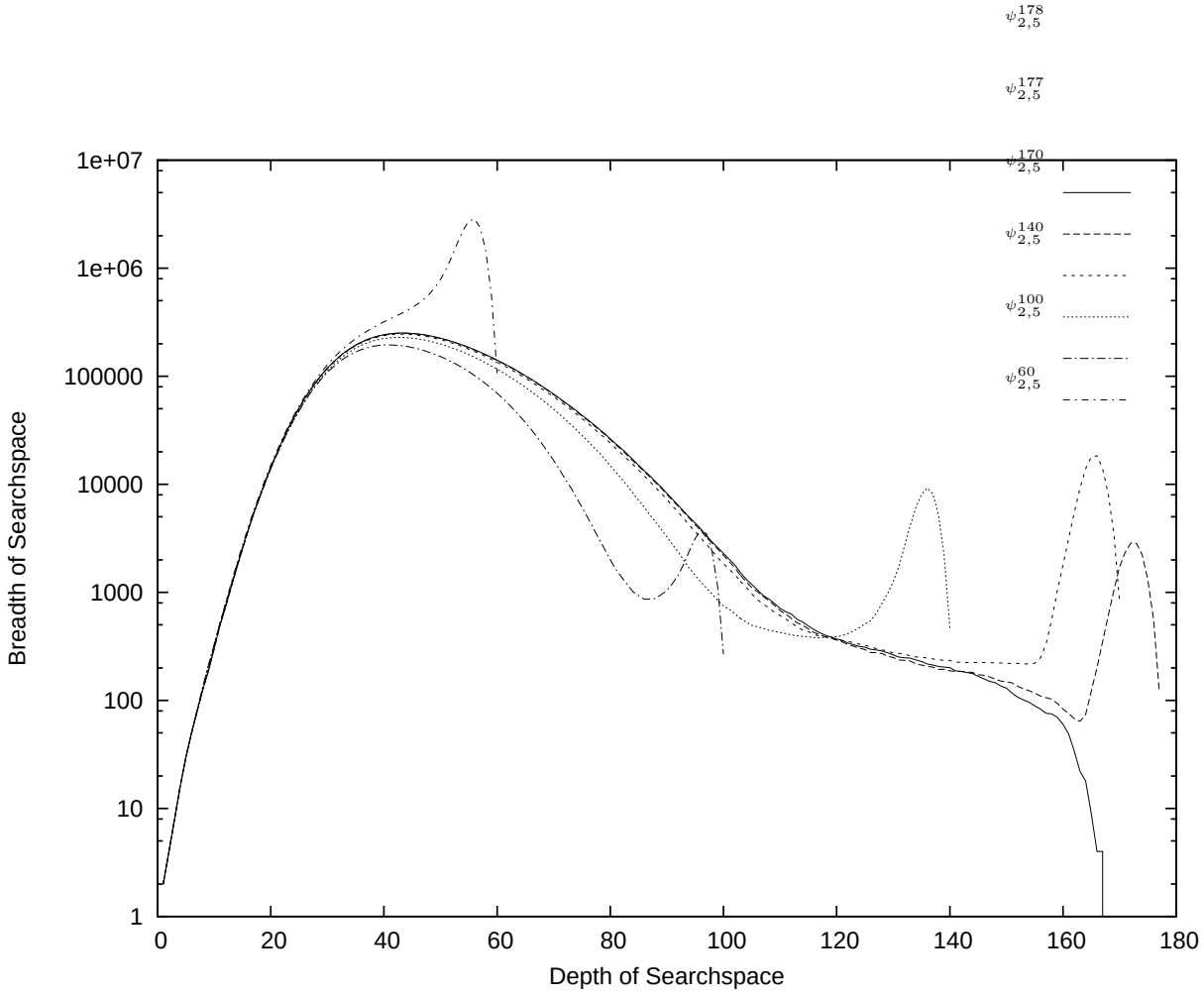
$\{\{\bar{v}_2\}, \{\bar{v}_1, \bar{v}_3\}, \{v_3\}\}$ $\{\{\bar{v}_1, \bar{v}_3\}, \{v_3, \bar{v}_4, \bar{v}_5\}, \{v_3\}\}$

ψ_{v_0} $\psi_{\bar{v}_0}$

The Sound Barrier

- **To reduce search: need to learn forced (inferenced) constraints (clauses and values)**
- **If formula is unsatisfiable, and is sparse, then must infer exponentially many large constraints before smaller constraints and values can be inferred**

The Sound Barrier



Typical SAT solver performance on a "hard" CNF formula.

Breaking The Sound Barrier

Introduce constraints that have no business being there because they are not inferred.

After "tunneling" through the mountain, retract the constraints and continue.

Example - Van der Waerden Numbers

Partition the set $S_n = \{1, ...n\}$ of the first n positive consecutive integers into k classes. Let $P_{n,k}(l)$ be a proposition that is $True$ if and only if all partitions of S_n into k classes contain at least one arithmetic progression of length l in at least one class. The k, l Van der Waerden number, denoted $W(k, l)$, is the minimum n for which $P_{n,k}(l)$ is $True$.

Example: $k = 2$, $l = 3$, $n = 9$.

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

Example: $k = 2$, $l = 3$, $n = 8$.

$\{\{1, 2, 5, 6\}\{3, 4, 7, 8\}\}$

Example - Van der Waerden Numbers

There is no known closed form expression for $W(k, l)$ and all but five of the first few numbers are unknown.

Table below shows all the known Van der Waerden numbers. In 1979 $W(4, 3)$ became the most recent addition to this table.

$k \setminus l$	**3**	**4**	**5**
2	**9**	**35**	**178**
3	**27**		
4	**76**		

Previous Bounds, Van der Waerden Numbers

$k \setminus l$	**3**	**4**	**5**	**6**	**7**	**8**
2	**9**	**35**	**178**	> 695	> 3702	> 7483
3	**27**	> 291	> 1209	> 8885	> 43854	> 161371
4	**76**	> 1047	> 10436	> 90306	> 262326	
5	> 125	> 2253	> 24044	> 177955		
6	> 206	> 3693	> 56692			

Example - Van der Waerden Formulas

Variables	Subscript Range	Meaning
$v_{i,j}$	$1 \leq i \leq n, 1 \leq j \leq k$	$v_{i,j} \equiv 1$ **iff** $i \in C_j$

Clauses	Subscript Range	Meaning
$\{\bar{v}_{i,r}, \bar{v}_{i,s}\}$	$1 \leq i \leq n, 1 \leq r < s \leq k$	i **is in at most one class**
$\{v_{i,1}, \ldots, v_{i,k}\}$	$1 \leq i \leq n$	i **is in at least one class**
$\{\bar{v}_{r,j}, \bar{v}_{r+1,j}, \ldots, \bar{v}_{r+l-1,j}\}$ $\{\bar{v}_{r,j}, \bar{v}_{r+2,j} \ldots, \bar{v}_{r+2(l-1),j}\}$ **...** $\{\bar{v}_{r,j}, \bar{v}_{r+t,j} \ldots, \bar{v}_{r+t(l-1),j}\}$	$1 \leq r \leq n-l+1$ $1 \leq j \leq k$ **...** $t = \lfloor (n-r)/(l-1) \rfloor$	**no arithmetic progression of length** l **in** C_j

Previous Bounds, Van der Waerden Formulas

$k \setminus l$	**3**	**4**	**5**	**6**	**7**	**8**
2	**9**	**35**	**178**	> 341	> 614	> 1322
3	**27**	> 193	> 676	> 2236		
4	**76**	> 416				
5	> 125	> 880				
6	> 194					

Formulas

$k \setminus l$	**3**	**4**	**5**	**6**	**7**	**8**
2	**9**	**35**	**178**	> 695	> 3702	> 7483
3	**27**	> 291	> 1209	> 8885	> 43854	> 161371
4	**76**	> 1047	> 10436	> 90306	> 262326	
5	> 125	> 2253	> 24044	> 177955		
6	> 206	> 3693	> 56692			

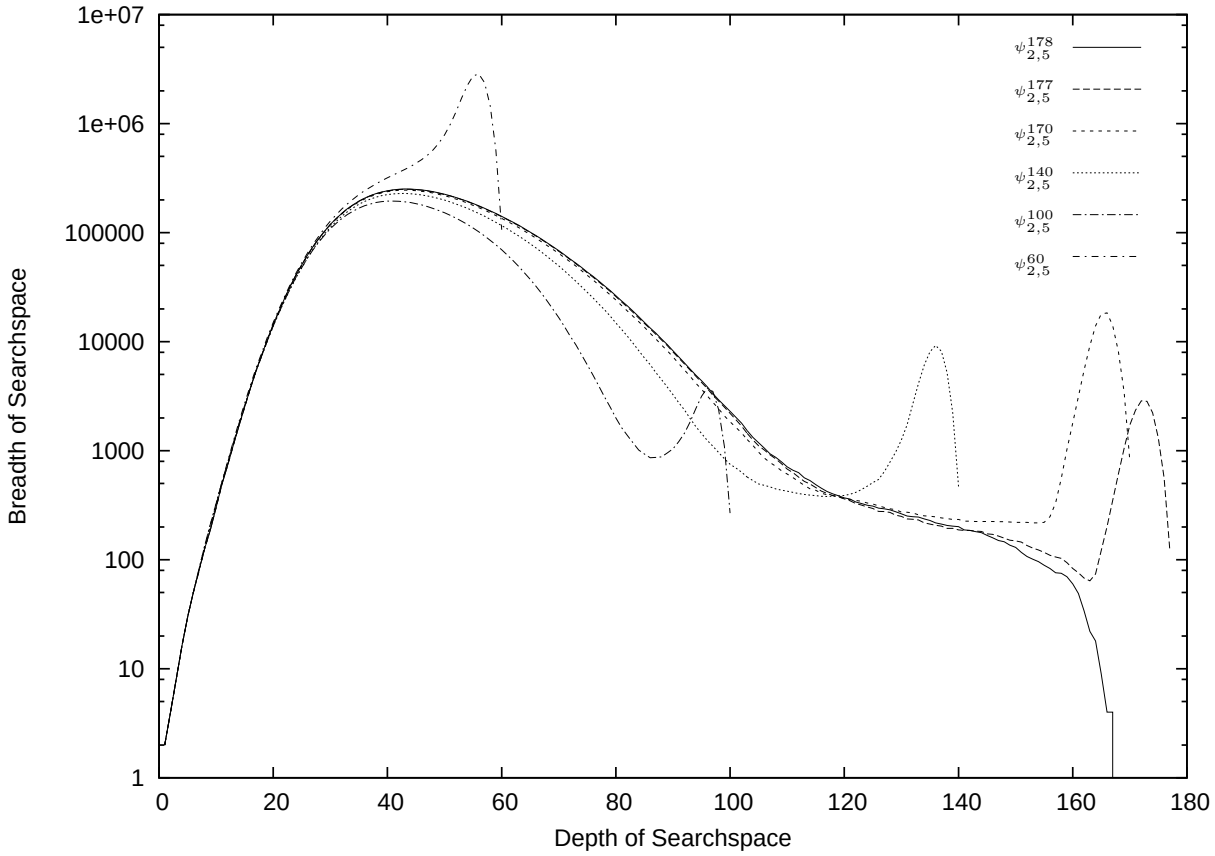
Analysis

Target W(2,6)

Variables	Subscript Range	Meaning
v_i	$-n/2 < i \leq n/2$	$v_i \equiv 1$ **if** $i + n/2 \in C_1$ $v_i \equiv 0$ **if** $i + n/2 \in C_2$

Clauses	Subscript Range	Meaning
$\{\bar{v}_i, \bar{v}_{i+1}, \ldots, \bar{v}_{i+5}\}$ $\{\bar{v}_i, \bar{v}_{i+2} \ldots, \bar{v}_{i+10}\}$ **...** $\{\bar{v}_i, \bar{v}_{i+t} \ldots, \bar{v}_{i+5t}\}$	$-n/2 < i \leq n/2 - 5$ **...** $t = \lfloor (n/2 - i)/5 \rfloor$	**no arithmetic progression of length 6 in** C_1
$\{v_i, v_{i+1} \ldots, v_{i+5}\}$ $\{v_i, v_{i+2} \ldots, v_{i+10}\}$ **...** $\{v_i, v_{i+t} \ldots, v_{i+5t}\}$	$-n/2 < i \leq n/2 - 5$ **...** $t = \lfloor (n/2 - i)/5 \rfloor$	**no arithmetic progression of length 6 in** C_2

Motivation - Analysis



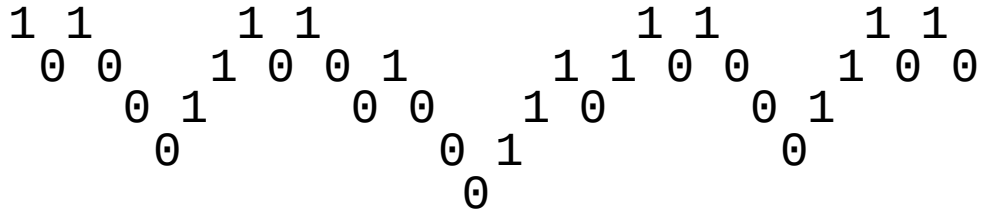
Observation 0.1. ***Consider a performance plot of search breadth vs. depth for any common SAT solver applied to*** $\psi_{2,l}^{n}$***. Such a plot has two maxima, the greatest of which occurs at approximately the same depth, say*** $W(2,l)/(2(l-1))$***, for*** $n > W(2,l)/(l-1)$***. The value of the greatest maximum is approximately the same for*** $n > W(2,l)/(l-1)$ ***and several orders of magnitude greater than the search breadth at depth*** $W(2,l)/(l-1)$***.***

Observation 0.2. $W(2,l) \approx l * W(2,l-1)$***, at least for small*** l***.***

Solver Outline

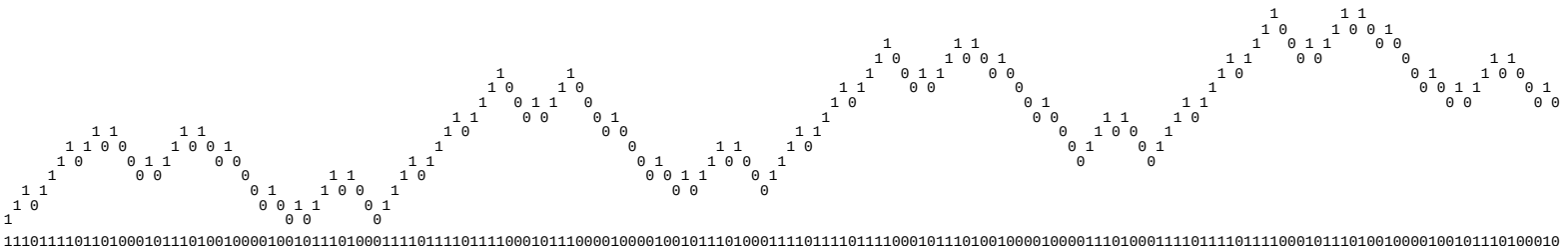
- **Choose small n formula (say around 210 variables)**
- **Add crazy constraints having no business there**
- **Solve until get through the mountain (check search breadth)**
- **Hopefully there is a solution left. Remove the crazy constraints.**
- **Finish solving. Hopefully there is a solution left.**
- **Increase n (add some "normal" clauses)**
- **Finish solving. Repeat until cannot get a solution.**

Tunnel 1



10100011101001000111011010001110 10

Analysis of solutions to $\psi_{2,4}$



Analysis of solutions to $\psi_{2,5}$

Limited length patterns of reverse symmetry

Tunnel 1

Conjecture 0.3 ***For every*** $\psi_{2,l}^{W(2,l)-1}$ ***there exists a solution that contains at least one reflected pattern of length*** $W(2,l)/((l-1)*2)$ ***with the middle positioned somewhere between*** $W(2,l)/(l-1)$ ***and*** $W(2,l)*(l-2)/(l-1)$**.**

Tunnel 1 is designed as a filter for consecutive variable assignment patterns that are not reverse symmetric.

Tunnel Clauses	Subscript Range	Meaning
$\{v_{-i}, v_{i+1}\}, \{\bar{v}_{-i}, \bar{v}_{i+1}\}$	$0 \leq i < s/2$	**force** $v_{-i} \equiv \bar{v}_{i+1}$**.**

Tunnel 2

- **Some small assignment patterns *do not* occur in solutions.**
- **The second tunnel filters those patterns.**
- **This action is opposite to that of *forcing* patterns to occur which is the objective of the first tunnel.**

Tunnel Clauses	Subscript Range	Filters
$\{v_i, \bar{v}_{i+t}, v_{i+2t}, \bar{v}_{i+3t}, v_{i+4t}, \bar{v}_{i+5t}\}$	$-n/2 < i \leq n/2 - 5t$	**010101**
$\{\bar{v}_i, v_{i+t}, \bar{v}_{i+2t}, v_{i+3t}, \bar{v}_{i+4t}, v_{i+5t}\}$	$1 \leq t \leq 20$	**101010**
$\{v_i, v_{i+t}, \bar{v}_{i+2t}, \bar{v}_{i+3t}, v_{i+4t}, \bar{v}_{i+5t}, \bar{v}_{i+6t}, v_{i+7t}\}$		**00110110**
$\{\bar{v}_i, \bar{v}_{i+t}, v_{i+2t}, v_{i+3t}, \bar{v}_{i+4t}, v_{i+5t}, v_{i+6t}, \bar{v}_{i+7t}\}$		**11001001**
$\{v_i, \bar{v}_{i+t}, \bar{v}_{i+2t}, v_{i+3t}, \bar{v}_{i+4t}, \bar{v}_{i+5t}, v_{i+6t}, v_{i+7t}\}$		**01101100**
$\{\bar{v}_i, v_{i+t}, v_{i+2t}, \bar{v}_{i+3t}, v_{i+4t}, v_{i+5t}, \bar{v}_{i+6t}, \bar{v}_{i+7t}\}$	$-n/2 < i \leq n/2 - 7t$	**10010011**
$\{v_i, v_{i+t}, \bar{v}_{i+2t}, \bar{v}_{i+3t}, \bar{v}_{i+4t}, v_{i+5t}, v_{i+6t}, \bar{v}_{i+7t}\}$	$1 \leq t \leq 20$	**00111001**
$\{\bar{v}_i, \bar{v}_{i+t}, v_{i+2t}, v_{i+3t}, v_{i+4t}, \bar{v}_{i+5t}, \bar{v}_{i+6t}, v_{i+7t}\}$		**11000110**
$\{\bar{v}_i, v_{i+t}, v_{i+2t}, \bar{v}_{i+3t}, \bar{v}_{i+4t}, \bar{v}_{i+5t}, v_{i+6t}, v_{i+7t}\}$		**10011100**
$\{v_i, \bar{v}_{i+t}, \bar{v}_{i+2t}, v_{i+3t}, v_{i+4t}, v_{i+5t}, \bar{v}_{i+6t}, \bar{v}_{i+7t}\}$		**01100011**

Tunnel 3

- **Analytic solutions to $\psi_{2,6}^{n}$ have been found for various values of n including 565 and 695. For solution to $\psi_{2,6}^{565}$, re-index the assigned variables**

  $$v_{-282}, ..., v_0, v_1, ...v_{282}$$

  to

  $$v_{-564}, v_{-562}, ..., v_0, v_2, ..., v_{562}, v_{564}$$

  and add unassigned variables

  $$v_{-565}, v_{-563}, ..., v_1, ..., v_{563}, v_{565}.$$

- **This operation *will not introduce any arithmetic progression* among the even indexed variables.**
- **The assignment to the even indexed variables is the third tunnel.**

Results

- **Any of the tunnels works - improves performance of off-the-shelf solvers (modified) by orders of magnitude**
- **All tunnels give the same result - bound of 1132 on $W(2, 6)$ (compare with 696 and 342)**
- **Probably broke the sound barrier on this particular instance**

Future

- Want to extend the results to Bounded Model Checking and other circuit problems.
- Similarities in formula construction
- But generally we want to show BMC formulas are not satisfiable. Maybe we can actually introduce "bugs," then see if they are found. If all bugs are found and no solution is found for the original formula, we have a pretty good feeling that the formula was not satisfiable in the first place.