Fair and Adventurous Enumeration of Quantifier Instantiations

Mikoláš Janota[1] Haniel Barbosa[2]
Pascal Fontaine[3] Andrew Reynolds[4]

[1] Czech Technical University in Prague

[2] Universidade Federal de Minas Gerais

[3] University of Liège

[4] University of Iowa

FMCAD 2021

Motivation — SMT instantiation

Disprove:

$$(\forall x\ f(x) \geq x) \wedge (\forall y\ f(y) \leq 5)$$

Motivation — SMT instantiation

Disprove:

$$(\forall x\ f(x) \geq x) \wedge (\forall y\ f(y) \leq 5)$$

$x \mapsto 6$ $\downarrow$ $\quad$ $\downarrow$ $y \mapsto 6$

$$f(6) \geq 6 \wedge f(6) \leq 5$$

Background: Herbrand

- In FOL $(\forall x \phi)$ is unsatisfiable iff there is unsatisfiable finite grounding from the Herbrand universe

Background: Herbrand

- In FOL $(\forall x \phi)$ is unsatisfiable iff there is unsatisfiable finite grounding from the Herbrand universe
- Example

$$
\begin{aligned}
& f(f(c)) \neq c \\
\wedge \quad & (\forall x)(f(x) = x)
\end{aligned}
$$

Background: Herbrand

- In FOL $(\forall x \phi)$ is unsatisfiable iff there is unsatisfiable finite grounding from the Herbrand universe
- Example

$$\begin{array}{ll} & f(f(c)) \neq c \\ \wedge & (\forall x)(f(x) = x) \end{array}$$

Instantiation:

$$\begin{array}{ll} & f(f(c)) \neq c \\ \wedge & f(c) = c \\ \wedge & f(f(c)) = f(c) \end{array}$$

Herbrand universe: $\{f^i(c) \mid c \in \mathbb{N}_0\}$

Background: Herbrand++

- Consider only ground terms in the formula:
  $(\forall x \phi) \wedge G$

Background: Herbrand++

- Consider only ground terms in the formula:
  $(\forall x \phi) \wedge G$

  $(\forall x \phi) \wedge G \wedge \phi[t_1/x]$, where $t_1 \in G$

Background: Herbrand++

- Consider only ground terms in the formula:
  $(\forall x\phi) \wedge G$

  $(\forall x\phi) \wedge G \wedge \phi[t_1/x]$, where $t_1 \in G$

  $(\forall x\phi) \wedge G \wedge \phi[t_1/x] \wedge \phi[t_2/x]$, where $t_2 \in (G \wedge \phi[t_1/x])$

Background: Herbrand++

- Consider only ground terms in the formula:
  $(\forall x \phi) \wedge G$

  $(\forall x \phi) \wedge G \wedge \phi[t_1/x]$, where $t_1 \in G$

  $(\forall x \phi) \wedge G \wedge \phi[t_1/x] \wedge \phi[t_2/x]$, where $t_2 \in (G \wedge \phi[t_1/x])$

  $(\forall x \phi) \wedge G \wedge \phi[t_1/x] \wedge \phi[t_2/x] \wedge \phi[t_3/x]$, where $t_3 \in (G \wedge \phi[t_1/x] \wedge \phi[t_2/x])$

Background: Herbrand++

- Consider only ground terms in the formula:
  $(\forall x \phi) \land G$

  $(\forall x \phi) \land G \land \phi[t_1/x]$, where $t_1 \in G$

  $(\forall x \phi) \land G \land \phi[t_1/x] \land \phi[t_2/x]$, where $t_2 \in (G \land \phi[t_1/x])$

  $(\forall x \phi) \land G \land \phi[t_1/x] \land \phi[t_2/x] \land \phi[t_3/x]$, where
  $t_3 \in (G \land \phi[t_1/x] \land \phi[t_2/x])$
- Still infinite but finite in each step!
  [Ge and de Moura, 2009, Reynolds et al., 2018]

Warning!

(Improved) Herbrand does not hold for theories

$$c = 21 \wedge (\forall m, n)((m > 1 \wedge n > 1) \Rightarrow (mn \neq c))$$

SMT Quantifiers by Enumeration

- Straightforward application of improved Herbrand:

SMT Quantifiers by Enumeration

- Straightforward application of improved Herbrand:
  - Start enumerating all combinations.

SMT Quantifiers by Enumeration

- Straightforward application of improved Herbrand:
  - Start enumerating all combinations.
  - Interleave by satisfiability checks of the ground part.

SMT Quantifiers by Enumeration

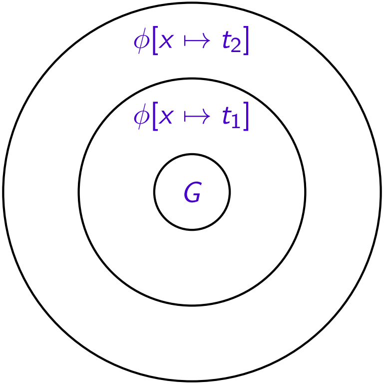
- Straightforward application of improved Herbrand:
  - Start enumerating all combinations.
  - Interleave by satisfiability checks of the ground part.
- Question:
  What order?

Old Age Is Good!

$$G \wedge \forall x.\, \phi$$

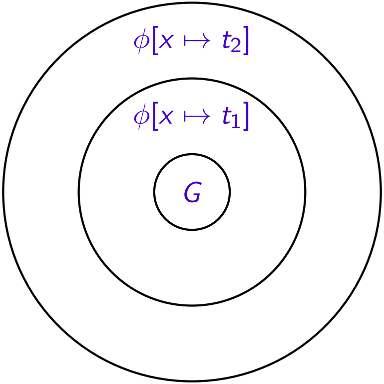
Old Age Is Good!

$$G \wedge \forall x.\, \phi$$



Old Age Is Good!

$$G \land \forall x.\, \phi$$



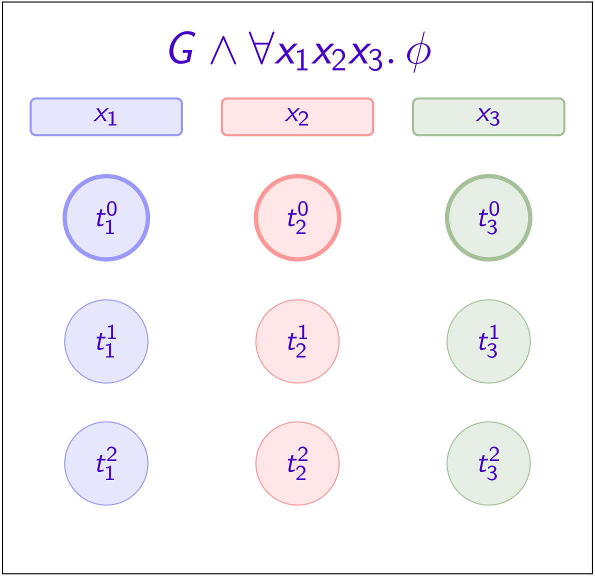
$$G \begin{cases} t_1 \\ t_2 \\ \dots \end{cases}$$

$$\phi[x \mapsto t_1] \begin{cases} \\ \dots \end{cases}$$

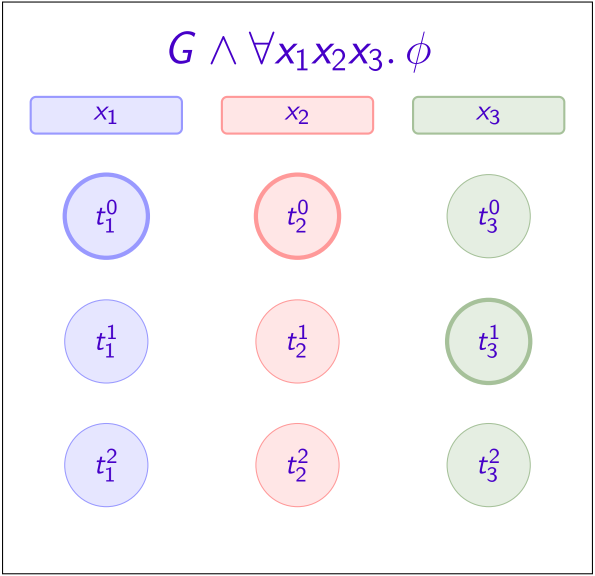
$$\phi[x \mapsto t_2] \begin{cases} \\ \dots \end{cases}$$

$$\dots$$

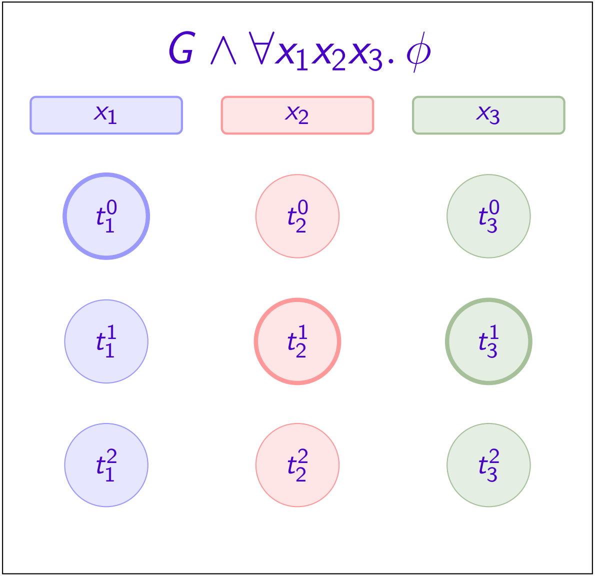
What about Tuples?



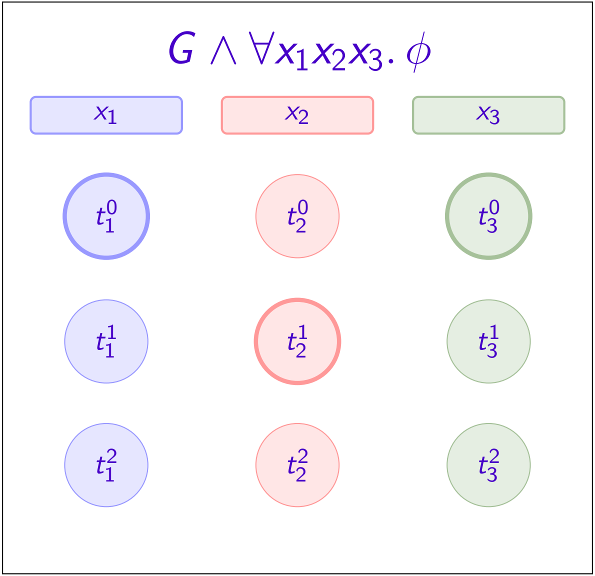
What about Tuples?



What about Tuples?

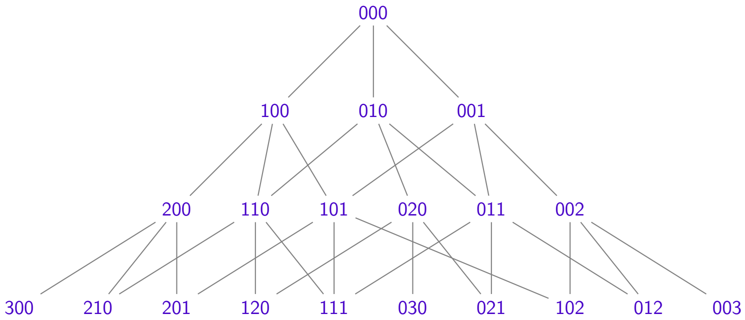


What about Tuples?



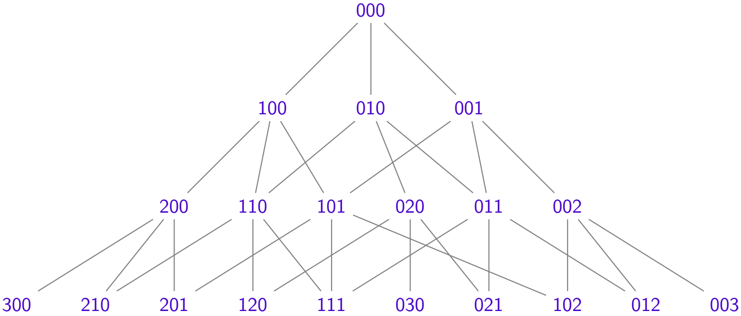
General Idea

1. Pick best term for each variable.



General Idea

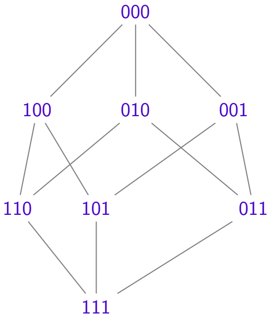
1. Pick best term for each variable.
2. Then go worsening as little as possible.



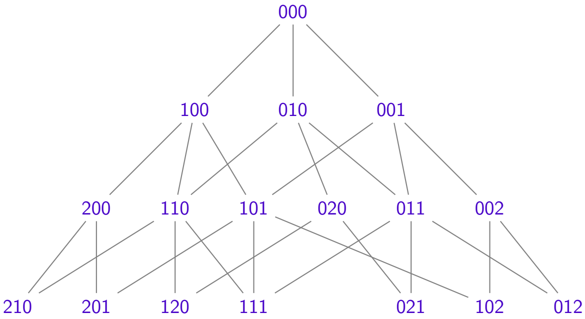
Different Strategies: Maximal Digit

000

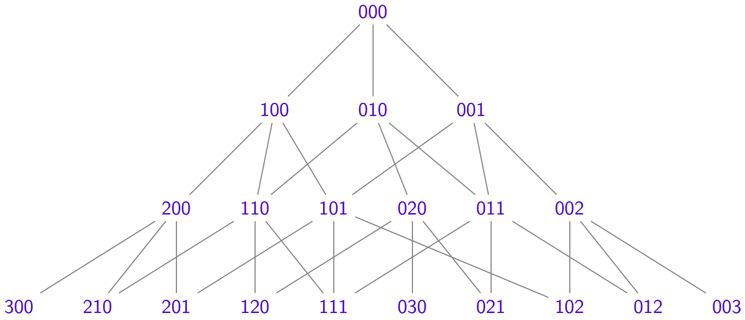
Different Strategies: Maximal Digit



Different Strategies: Maximal Digit



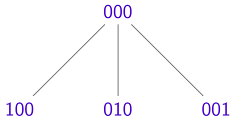
Different Strategies: Maximal Digit



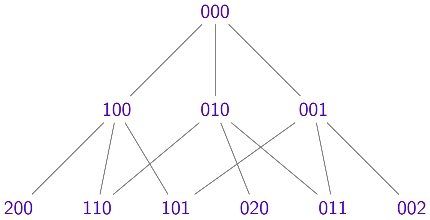
Different Strategies: Sum of Digits

000

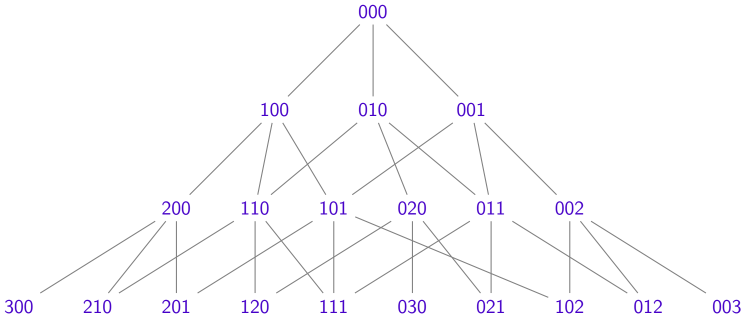
Different Strategies: Sum of Digits



Different Strategies: Sum of Digits



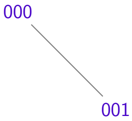
Different Strategies: Sum of Digits



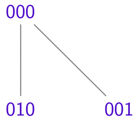
Different Strategies: Leximax

000

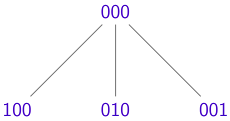
Different Strategies: Leximax



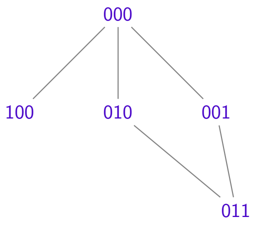
Different Strategies: Leximax



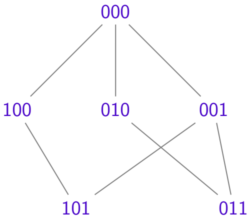
Different Strategies: Leximax



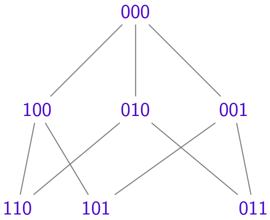
Different Strategies: Leximax



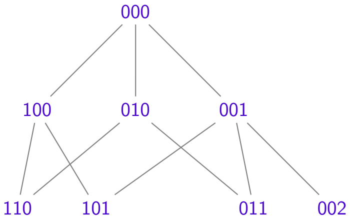
Different Strategies: Leximax



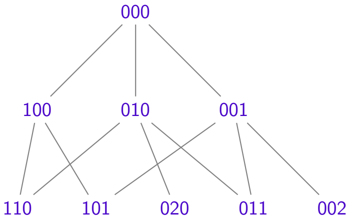
Different Strategies: Leximax



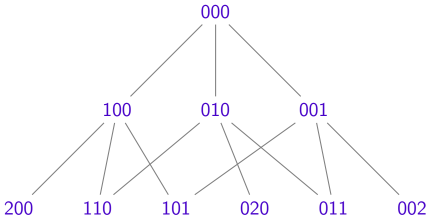
Different Strategies: Leximax



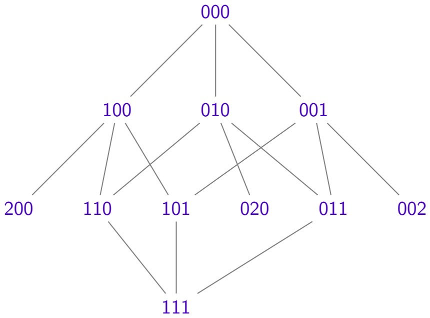
Different Strategies: Leximax



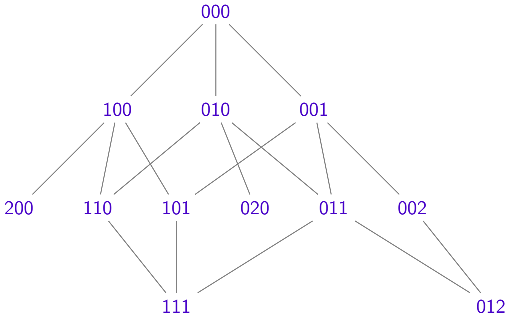
Different Strategies: Leximax



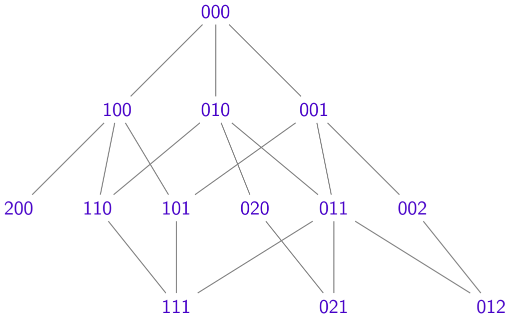
Different Strategies: Leximax



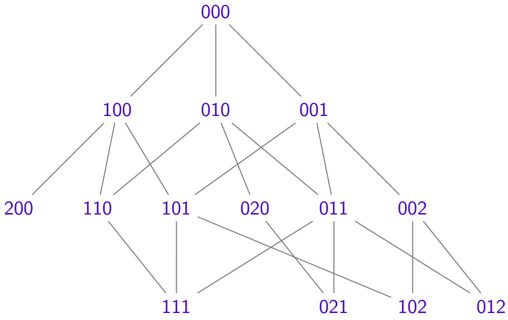
Different Strategies: Leximax



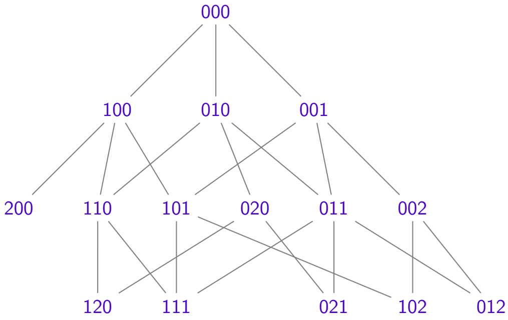
Different Strategies: Leximax



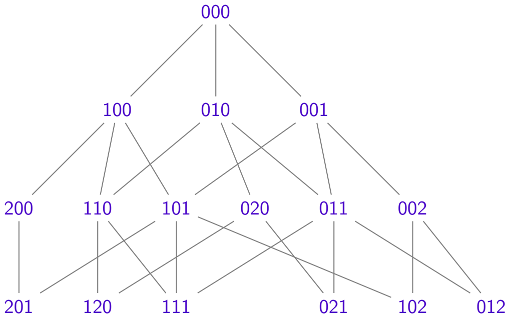
Different Strategies: Leximax



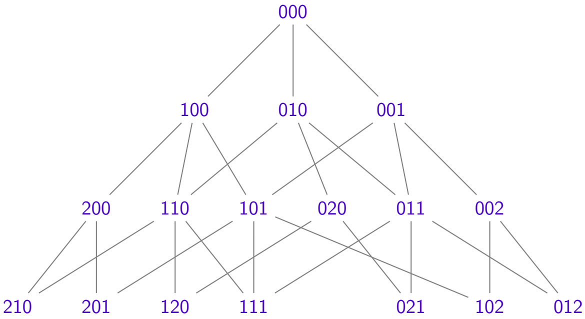
Different Strategies: Leximax



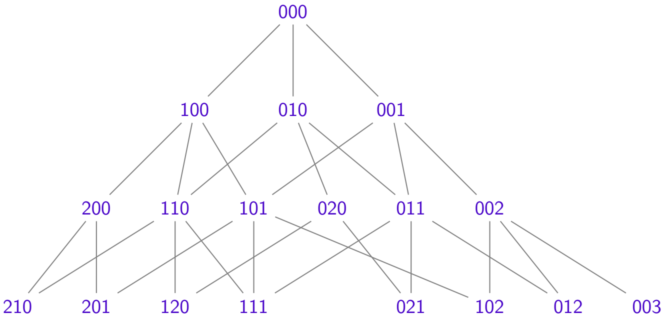
Different Strategies: Leximax



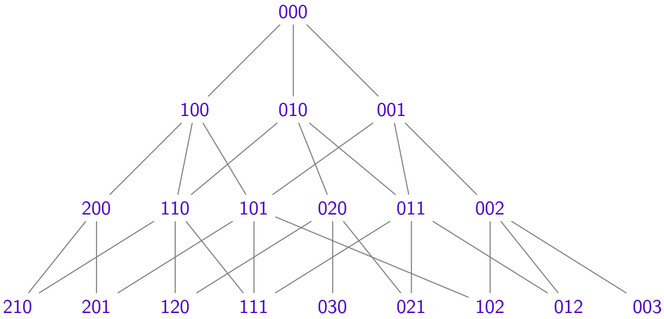
Different Strategies: Leximax



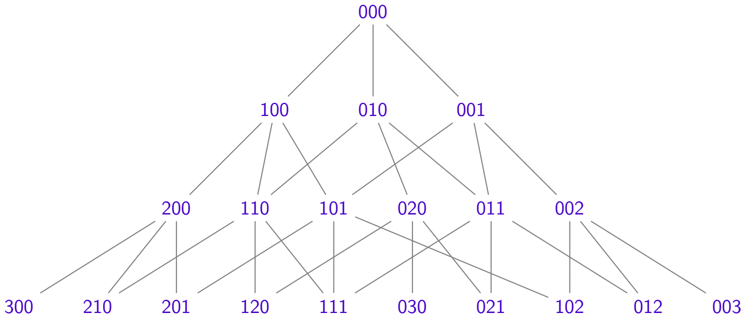
Different Strategies: Leximax



Different Strategies: Leximax



Different Strategies: Leximax

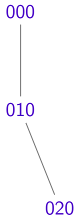


Different Strategies: Random Walk

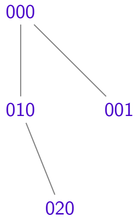
000

Different Strategies: Random Walk

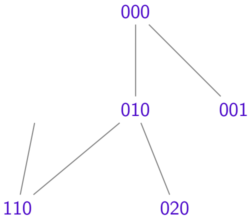
Different Strategies: Random Walk



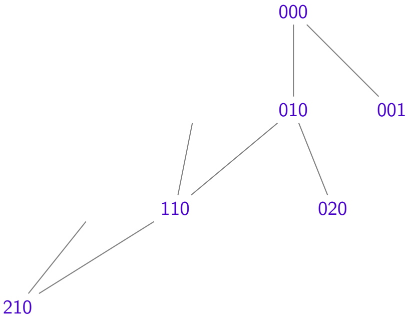
Different Strategies: Random Walk



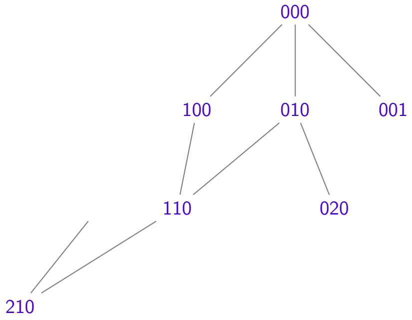
Different Strategies: Random Walk



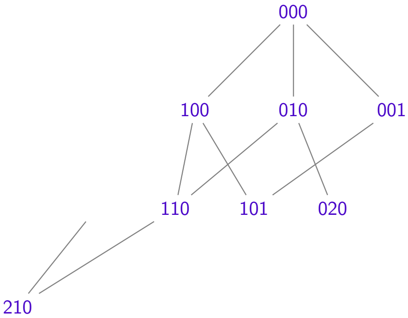
Different Strategies: Random Walk



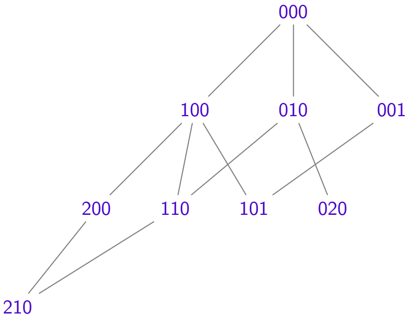
Different Strategies: Random Walk



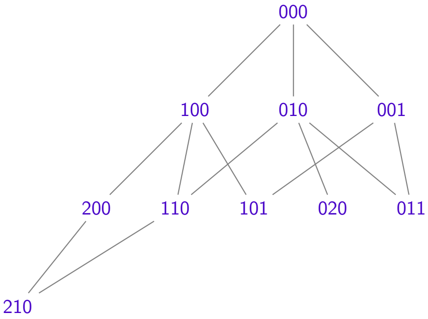
Different Strategies: Random Walk



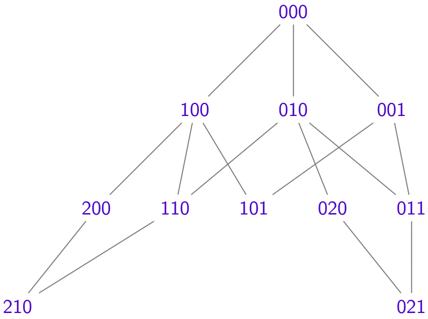
Different Strategies: Random Walk



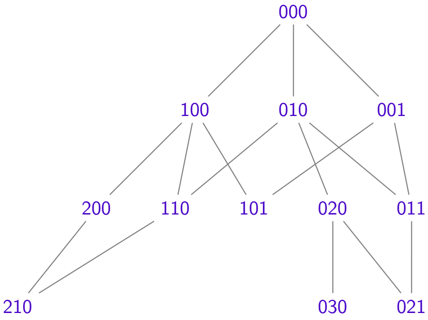
Different Strategies: Random Walk



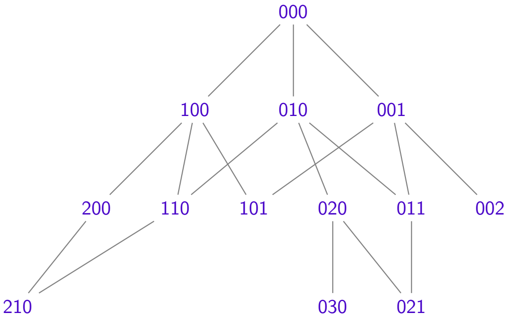
Different Strategies: Random Walk



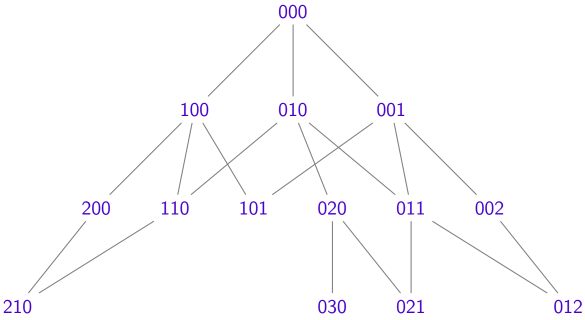
Different Strategies: Random Walk



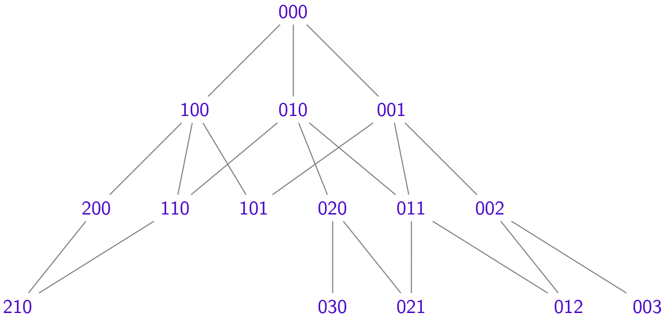
Different Strategies: Random Walk



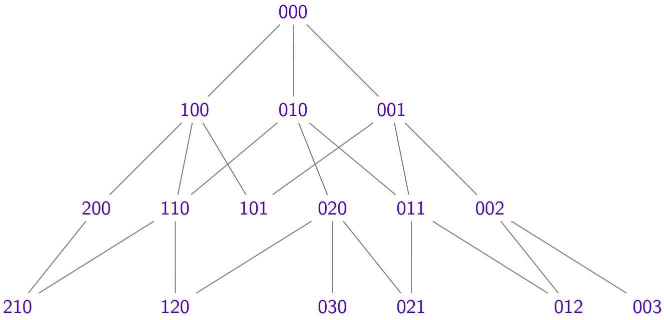
Different Strategies: Random Walk



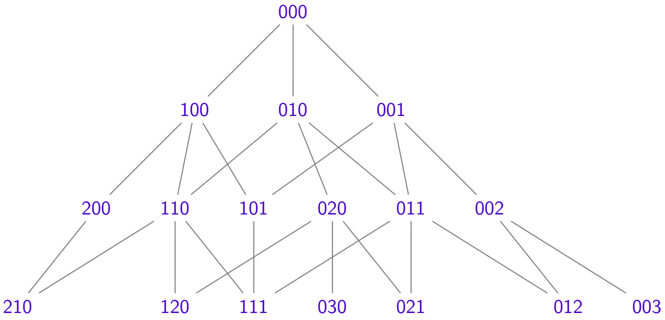
Different Strategies: Random Walk



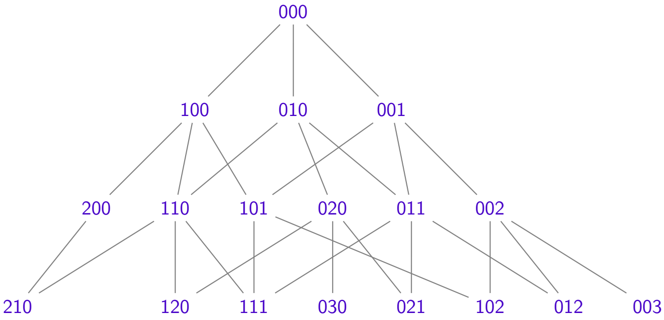
Different Strategies: Random Walk



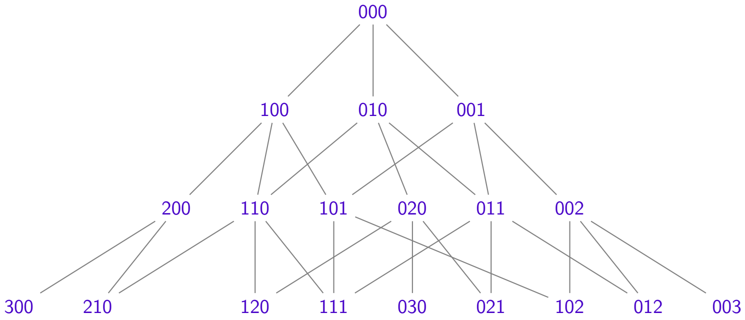
Different Strategies: Random Walk



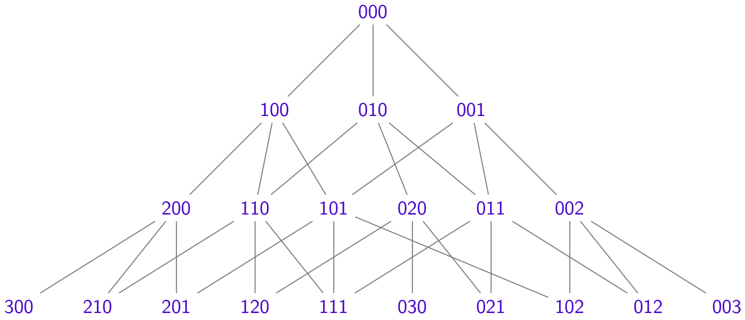
Different Strategies: Random Walk



Different Strategies: Random Walk



Different Strategies: Random Walk



Experiments

Library	#	e	u	id2	id4	lmax	sum	rwlk
TPTP	18627	**7765**	6989	6801	6834	6832	6922	6839
UF	7668	**3243**	3016	2975	2963	2959	3009	2992
UFLIA	10137	**7424**	6024	6018	5897	6001	5980	5994
UFNIA	13509	5715	**7458**	7396	7384	7426	7437	7430

Experiments

Library	#	e	u	id2	id4	lmax	sum	rwlk
TPTP	18627	**7765**	6989	6801	6834	6832	6922	6839
UF	7668	**3243**	3016	2975	2963	2959	3009	2992
UFLIA	10137	**7424**	6024	6018	5897	6001	5980	5994
UFNIA	13509	5715	**7458**	7396	7384	7426	7437	7430

Library	#	e	enu-vbs	z3
TPTP	18627	**7765**	7330	-
UF	7668	**3243**	3120	2905
UFLIA	10137	**7424**	6188	6912
UFNIA	13509	5715	**7620**	6491

Conclusion and Future Work

- Given order on a variable, what's the order on tuples?

Conclusion and Future Work

- Given order on a variable, what's the order on tuples?
- We define a number of diverse strategies.

Conclusion and Future Work

- Given order on a variable, what's the order on tuples?
- We define a number of diverse strategies.
- based on Pareto ordering

Conclusion and Future Work

- Given order on a variable, what's the order on tuples?
- We define a number of diverse strategies.
- based on Pareto ordering
- The new strategies diversify the solver.

Conclusion and Future Work

- Given order on a variable, what's the order on tuples?
- We define a number of diverse strategies.
- based on Pareto ordering
- The new strategies diversify the solver.
- Could we somehow learn what is better?

Conclusion and Future Work

- Given order on a variable, what's the order on tuples?
- We define a number of diverse strategies.
- based on Pareto ordering
- The new strategies diversify the solver.
- Could we somehow learn what is better?
- How to include dependence between variables in the order?

Ge, Y. and de Moura, L. M. (2009).
Complete instantiation for quantified formulas in satisfiabiliby modulo theories.

Reynolds, A., Barbosa, H., and Fontaine, P. (2018).
Revisiting enumerative instantiation.