Positional Games and QBF

The Corrective Encoding

Abdallah Saffidine [1]
Valentin Mayer-Eichberger [2]

[1] The University of New South Wales, Australia

[2] Technische Universitaet Berlin, Germany

The rise of SAT Solvers

> *Satisfiability solvers can now be effectively deployed in practical applications.*

S. Malik and L. Zhang. “Boolean satisfiability from theoretical hardness to practical success”. In: *Communications of the ACM* 52.8 (2009), pp. 76–82

The rise of SAT Solvers

> *Satisfiability solvers can now be effectively deployed in practical applications.*

S. Malik and L. Zhang. “Boolean satisfiability from theoretical hardness to practical success”. In: *Communications of the ACM* 52.8 (2009), pp. 76–82

Double-exponential is the new Polynomial

That was a joke but. . .

- . . . maybe PSPACE is the new NP?
- . . . maybe QBF is the new SAT?

Quantified Boolean Formulas

The prototypical PSPACE-complete problem.

Example

$$\exists x_1 \forall y_2 \exists x_3 \forall y_4 \exists x_5 (x_1 \vee y_2 \vee x_3) \wedge (\neg x_3 \vee y_4 \vee x_5) \wedge (\neg y_2 \vee x_5)$$

Game Semantics

- Two players *Existential* $\exists$ and *Universal* $\forall$ choose variables.
- $\exists$ tries to satisfy and $\forall$ tries to falsify.
- Formula is true iff $\exists$ has a winning strategy.

More natural than SAT for modeling Multi-Agent systems.

Toby Walsh's invited talk at SAT2003

Push QBF solvers research via a Connect 4 challenge

Motivating related work

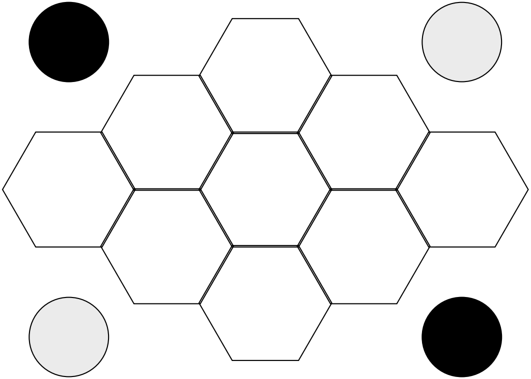
I. P. Gent and A. G. D. Rowley. "Encoding Connect-4 Using Quantified Boolean Formulae". In: *Modelling and Reformulating Constraint Satisfaction Problems*. 2003, pp. 78–93

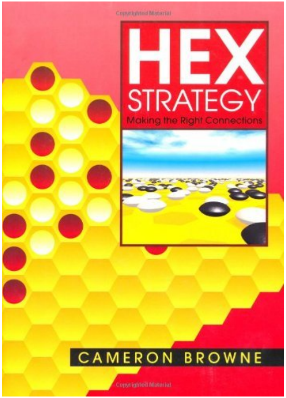
- *A key part of our encoding are variables capturing the notion of a player "cheating"*

Diptarama et al. "QBF Encoding of Generalized Tic-Tac-Toe". In: *4th International Workshop on Quantified Boolean Formulas*. 2016, pp. 14–26

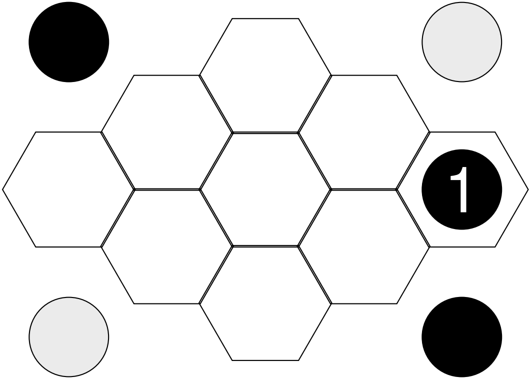
- Also uses "cheat variables"

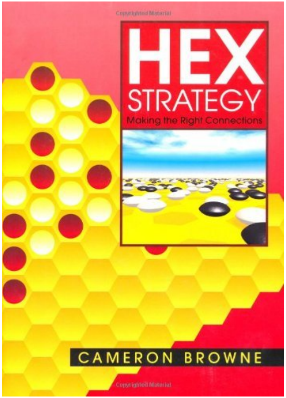
The Game of Hex



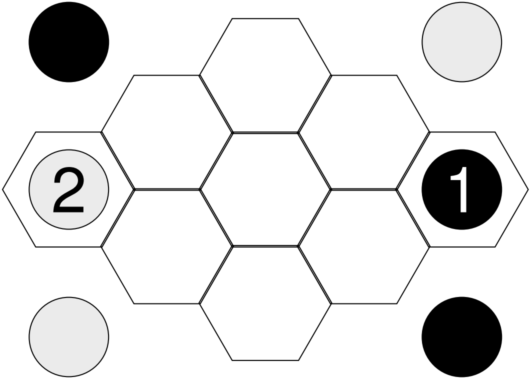


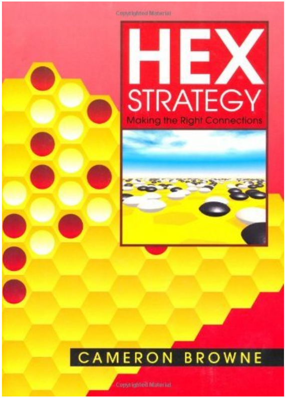
The Game of Hex



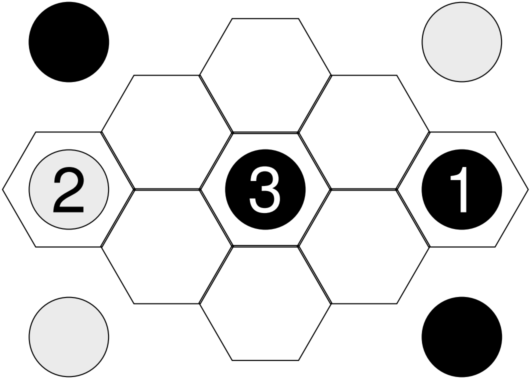


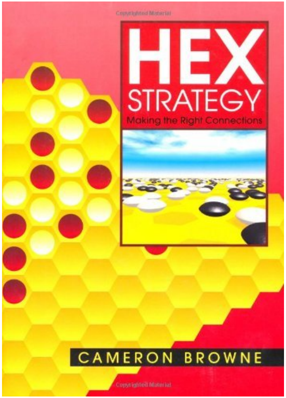
The Game of Hex



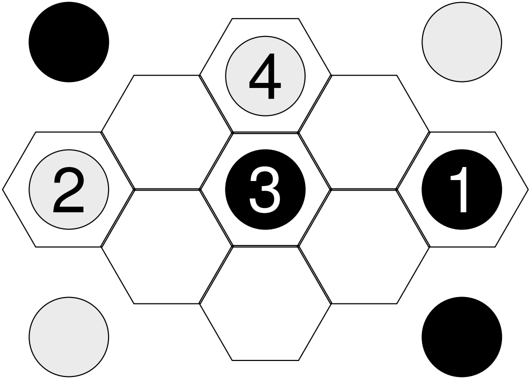


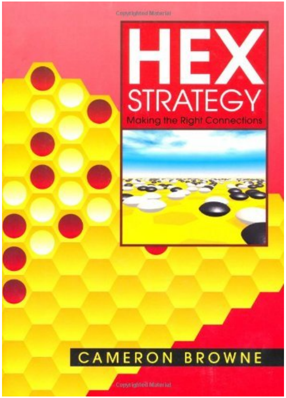
The Game of Hex



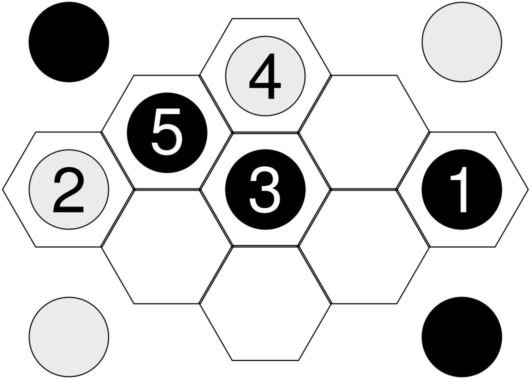


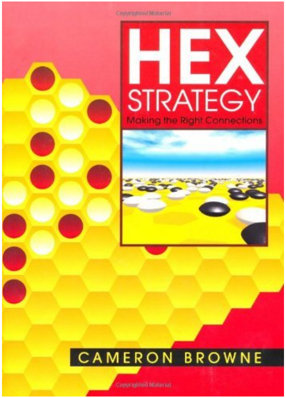
The Game of Hex



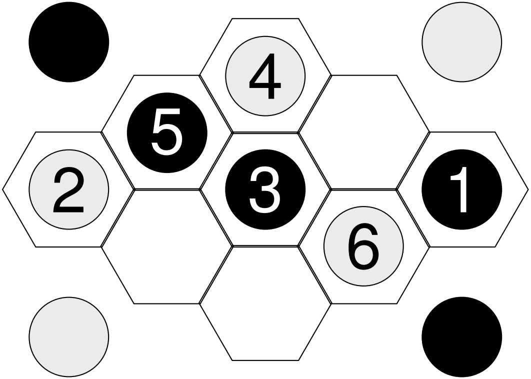


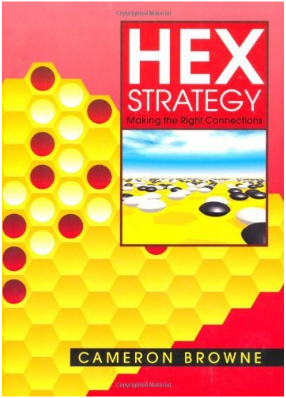
The Game of Hex



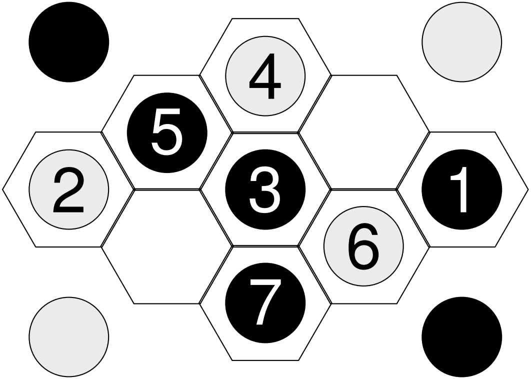


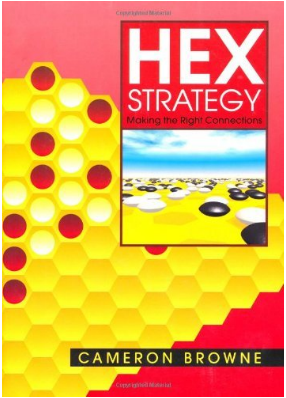
The Game of Hex



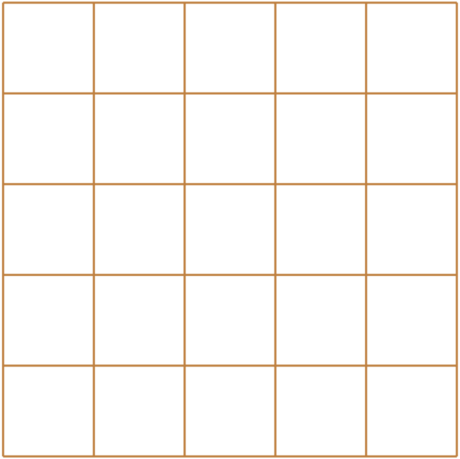


The Game of Hex

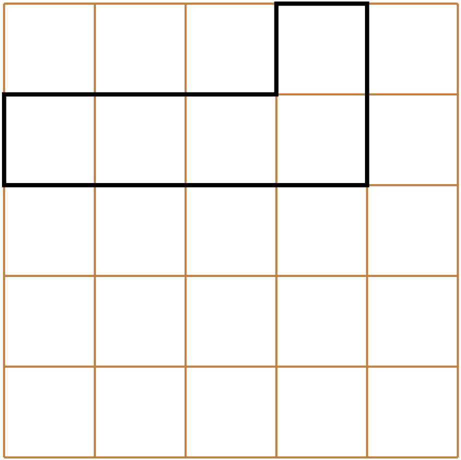




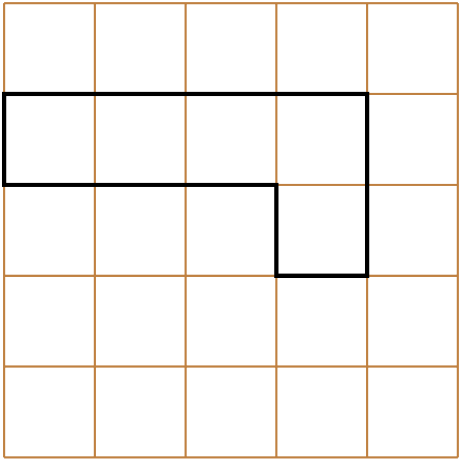
Generalized Tic-Tac-Toe



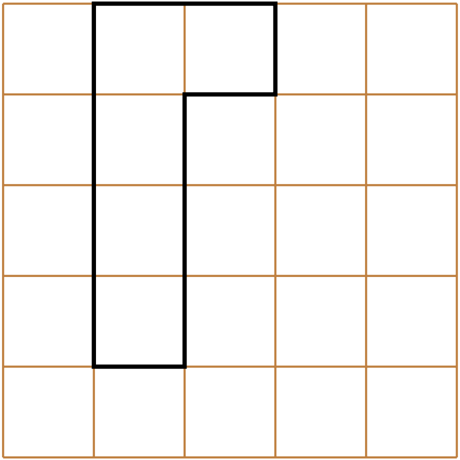
Generalized Tic-Tac-Toe



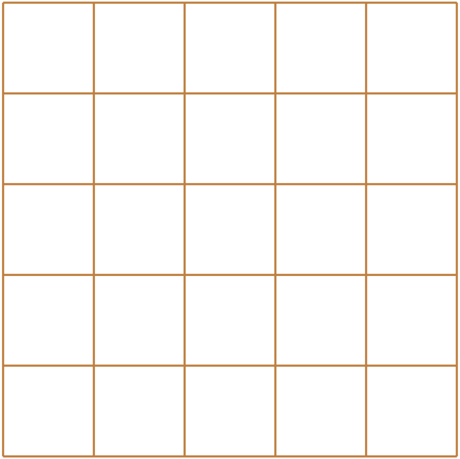
Generalized Tic-Tac-Toe



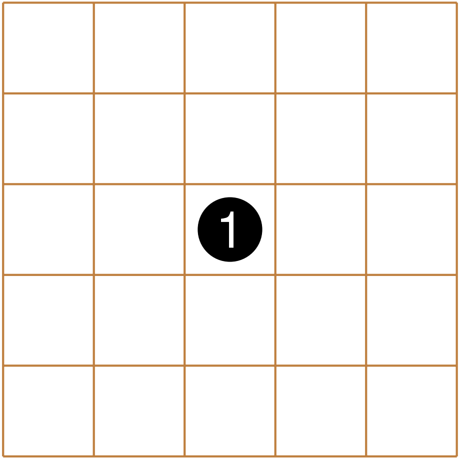
Generalized Tic-Tac-Toe



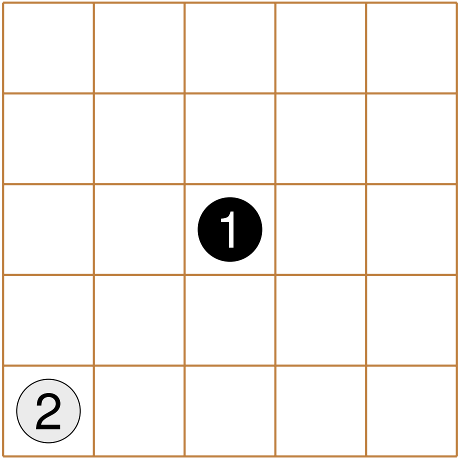
Generalized Tic-Tac-Toe



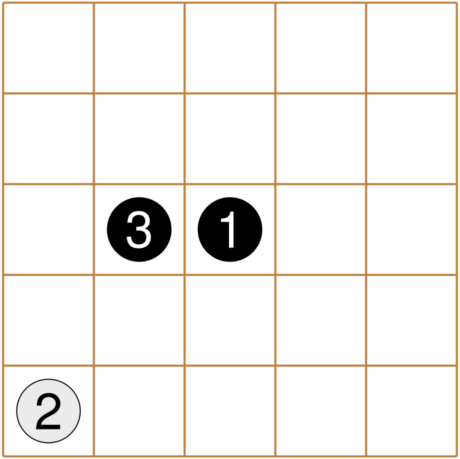
Generalized Tic-Tac-Toe



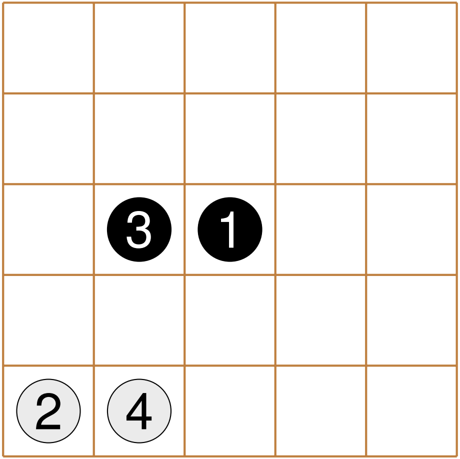
Generalized Tic-Tac-Toe



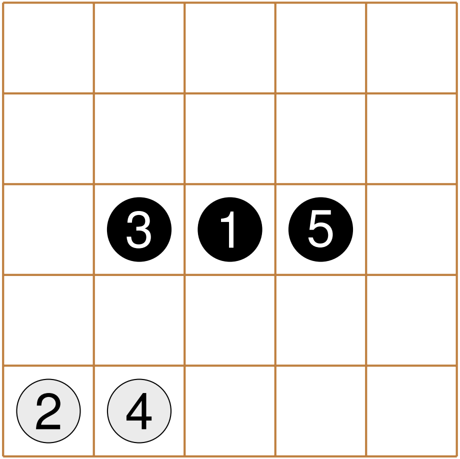
Generalized Tic-Tac-Toe



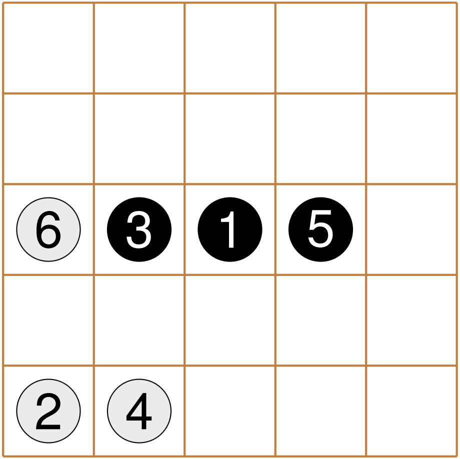
Generalized Tic-Tac-Toe



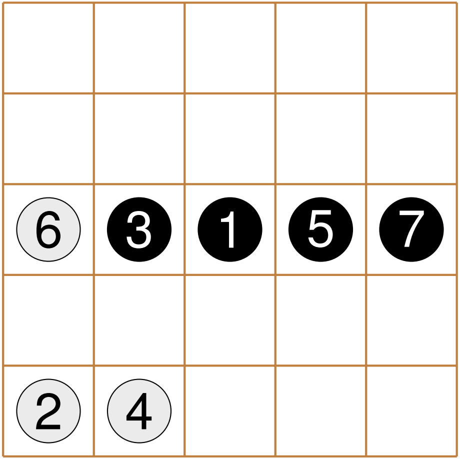
Generalized Tic-Tac-Toe



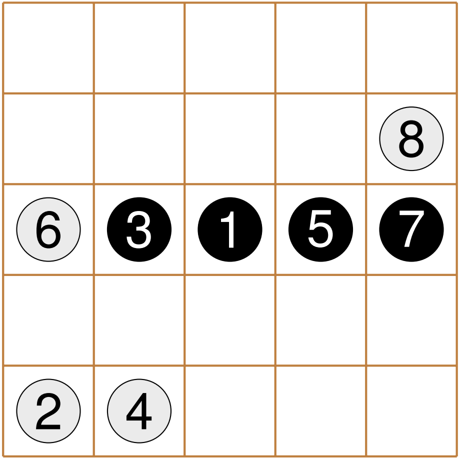
Generalized Tic-Tac-Toe



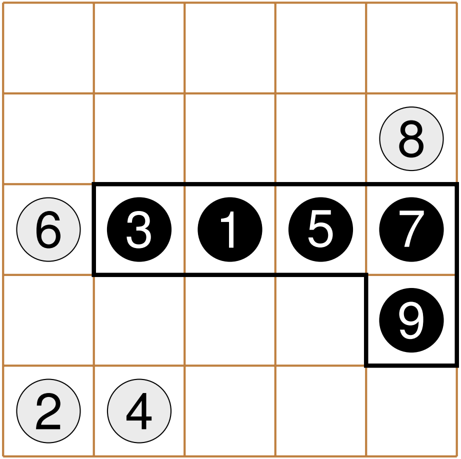
Generalized Tic-Tac-Toe



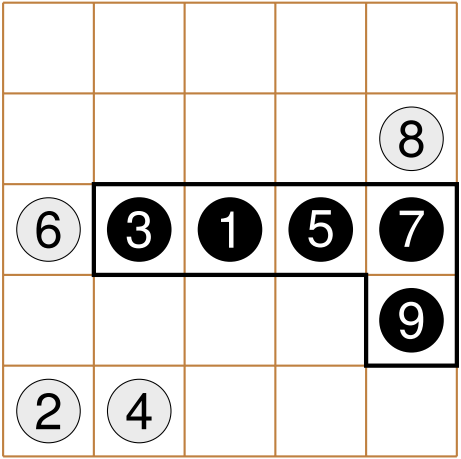
Generalized Tic-Tac-Toe

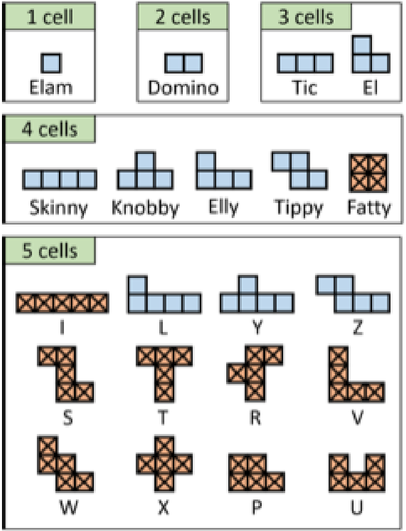


Generalized Tic-Tac-Toe



Generalized Tic-Tac-Toe





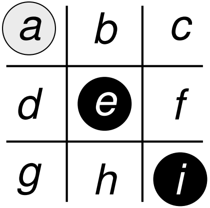
Positional Games

- Two players black (B) and white (W).
- Hypergraph $H = (V, E)$ where hyperedges $E = E_{\mathrm{B}} \cup E_{\mathrm{W}}$ are winning configurations.
- A player wins by claiming all vertices of a winning configuration before the opponent.

Example

- Hex
- Tictactoe
- ~~Chess~~ (Different pieces)
- ~~Go~~ (Captures)
- ~~Connect 4~~ (Gravity)

TIC-TAC-TOE and its winning configurations.



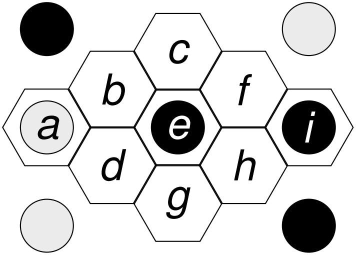
$\{a, b, c\}, \{d, e, f\},$

$\{g, h, i\}, \ldots,$

$\{a, e, i\}, \{c, e, g\}$

Winning sets: aligned triples

HEX and its winning configurations for Black.



$\{a, d, g\}, \{a, d, e, h\},$

$\{a, d, e, f, i\}, \{b, d, g\},$

$\{b, e, g\}, \ldots, \{c, f, i\}$

Winning sets: NW-SE paths

Main idea

One of the difficulties of the encoding
How to map game moves to QBF moves? (ie, select a move → assign some variables to $\top, \bot$)

Simple Solution

- For each move create a corresponding variable "this move was selected"
- Add clauses to prevent Black from chosing many moves per round.
- Add "cheating variables" to prevent White from chosing many moves per round.

Main idea

One of the difficulties of the encoding
How to map game moves to QBF moves? (ie, select a move $\rightarrow$ assign some variables to $\top, \bot$)

Logarithmic idea

- Encode White moves logarithmically
- Avoid “cheating variables”!

The New Encoding is smaller

Preprocessing on 5×5 instance `gttt_1_1_00101121_5x5_b`

		None	Q	H	B	QB	BQ	HQ	QH
DYS	#qb	25	25	25	25	25	25	25	25
	#∀	300	300	300	299	299	299	300	300
	#∃	21056	12058	7553	2750	2605	2750	7553	7545
	#cl	54k	36k	33k	21k	20k	19k	30k	30k
	#lits	191k	127k	145k	120k	107k	107k	103k	135k
	time(s)	0	46	1210	9	55	22	1233	2030
COR	#qb	25	25	25	25	25	25	25	25
	#∀	60	60	58	58	58	58	58	58
	#∃	4649	3127	3433	1396	1360	1396	3432	2981
	#cl	12k	8k	29k	8k	8k	7k	15k	20k
	#lits	29k	20k	118k	35k	34k	31k	52k	89k
	time(s)	0	0	275	2	2	2	277	20

Preprocessors: QratPre+ 2.0 (Q), HQSPRE 1.4 (H), Bloqqer v37 (B)

The New Encoding is smaller

Preprocessing on 5×5 instance `gttt_1_1_00101121_5x5_b`

		None	Q	H	B	QB	BQ	HQ	QH
DYS	#qb	25	25	25	25	25	25	25	25
	#∀	300	300	300	299	299	299	300	300
	#∃	21056	12058	7553	2750	2605	2750	7553	7545
	#cl	54k	36k	33k	21k	20k	19k	30k	30k
	#lits	191k	127k	145k	120k	107k	107k	103k	135k
	time(s)	0	46	1210	9	55	22	1233	2030
COR	#qb	25	25	25	25	25	25	25	25
	#∀	60	60	58	58	58	58	58	58
	#∃	4649	3127	3433	1396	1360	1396	3432	2981
	#cl	12k	8k	29k	8k	8k	7k	15k	20k
	#lits	29k	20k	118k	35k	34k	31k	52k	89k
	time(s)	0	0	275	2	2	2	277	20

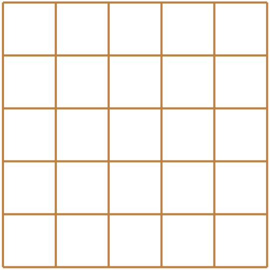
Preprocessors: QratPre+ 2.0 (Q), HQSPRE 1.4 (H), Bloqqer v37 (B)

Solving GTTT 4×4

	Solver	Pre-proc.	Solve ⊤	Solve ⊥	Fail[1]	Total time h:mm:ss
DYS	Caqe 4.0.1	B	31	61	4	**3:11:08**
	DepQbf 6.03	Q	28	54	14	7:33:31
	Qesto 1.0	BQ	27	47	22	9:41:27
	Qute 1.1	BQ	27	42	27	9:57:17
COR	Caqe 4.0.1	Q	34	62	0	1:08:19
	DepQbf 6.03	N	34	62	0	**0:10:02**
	Qesto 1.0	BQ	34	62	0	0:56:44
	Qute 1.1	B	30	52	14	6:27:46

[1] Timeout: 1000 seconds per instance

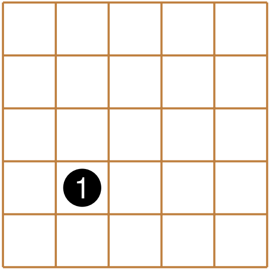
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$

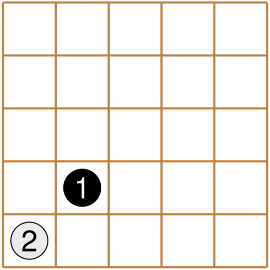
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$

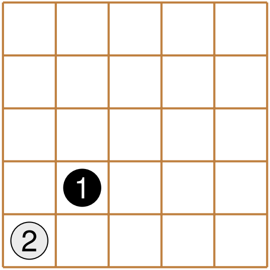
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$

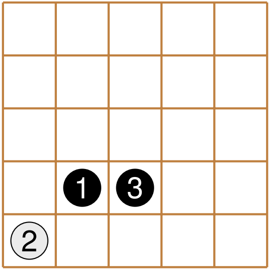
Case study: GTTT 5 × 5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours

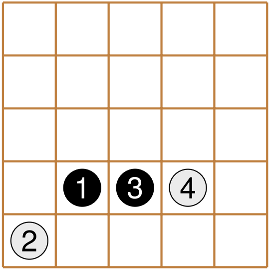
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours

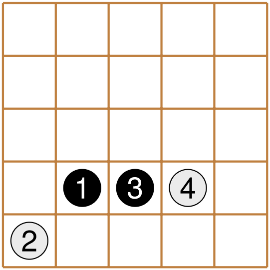
Case study: GTTT 5 × 5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours

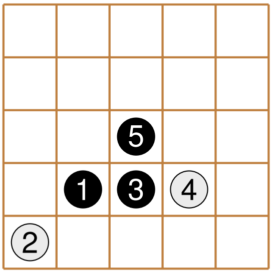
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.

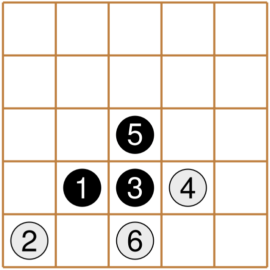
Case study: GTTT 5 × 5, L-shape



Time (s) needed to establish if Black can win within depth ≤ d after k stones are played.

k	d	$\not\models \phi_{d-2}^{k}$	$\models \phi_{d}^{k}$
2	13	1942.	> 8 hours
4	13	239.	1674.

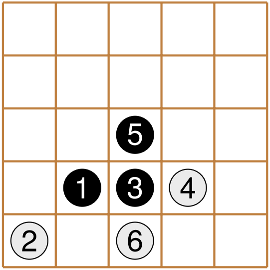
Case study: GTTT 5 × 5, L-shape



Time (s) needed to establish if Black can win within depth ≤ d after k stones are played.

k	d	$\not\models \phi_{d-2}^{k}$	$\models \phi_{d}^{k}$
2	13	1942.	> 8 hours
4	13	239.	1674.

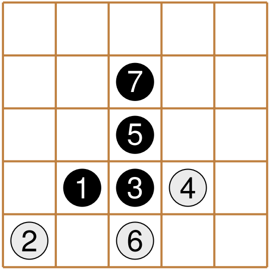
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.

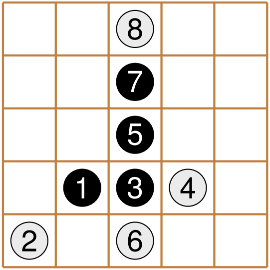
Case study: GTTT 5 × 5, L-shape



Time (s) needed to establish if Black can win within depth ≤ *d* after *k* stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.

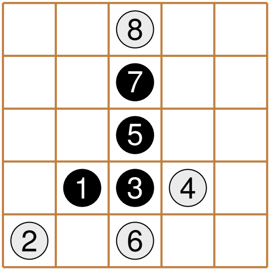
Case study: GTTT 5 × 5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.

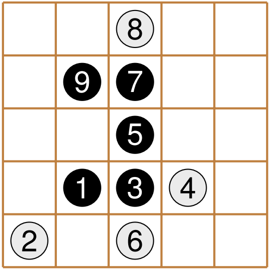
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0

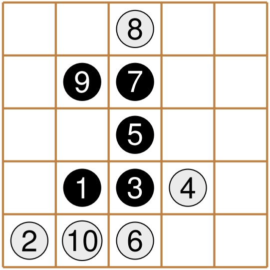
Case study: GTTT 5 × 5, L-shape



Time (s) needed to establish if Black can win within depth ≤ *d* after *k* stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0

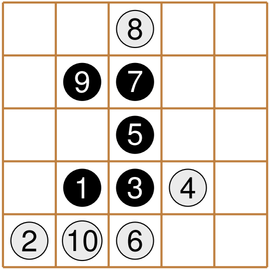
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0

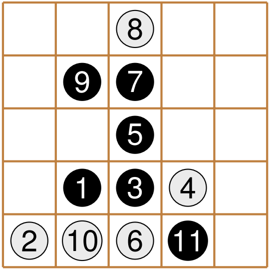
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52

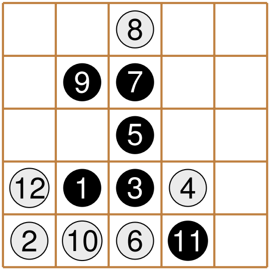
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52

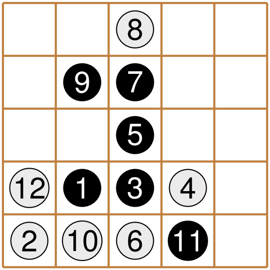
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52

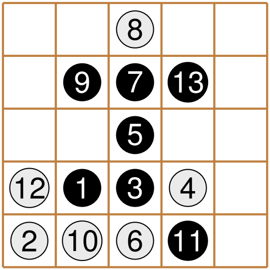
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52
12	7	0.03	0.09

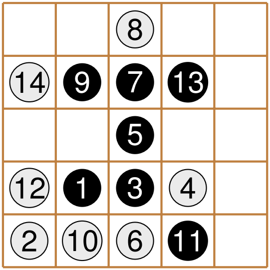
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi_{d-2}^{k}$	$\models \phi_{d}^{k}$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52
12	7	0.03	0.09

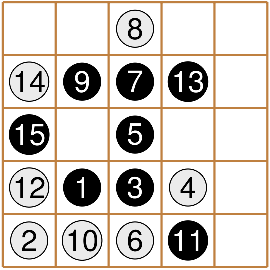
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52
12	7	0.03	0.09
...	...	...	...

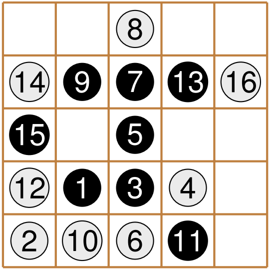
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52
12	7	0.03	0.09
…	…	…	…

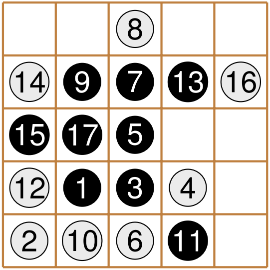
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52
12	7	0.03	0.09
...	...	...	...

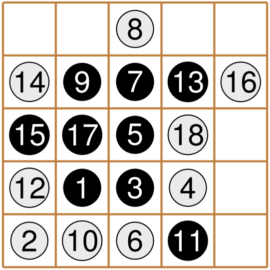
Case study: GTTT 5×5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^{k}_{d-2}$	$\models \phi^{k}_{d}$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52
12	7	0.03	0.09
…	…	…	…

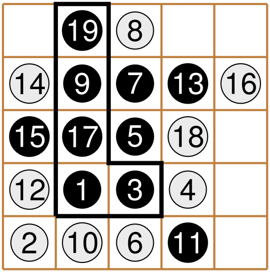
Case study: GTTT 5 × 5, L-shape



Time (s) needed to establish if Black can win within depth $\leq d$ after k stones are played.

k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52
12	7	0.03	0.09
...	...	...	...

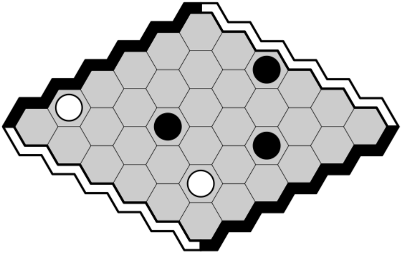
Case study: GTTT 5 × 5, L-shape

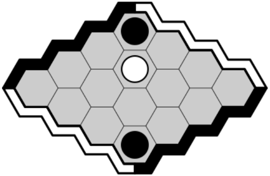


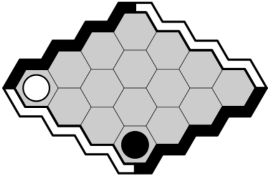
Time (s) needed to establish if Black can win within depth ≤ d after k stones are played.

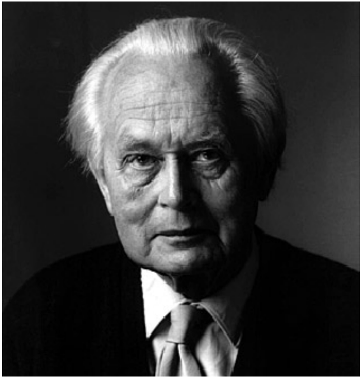
k	d	$\not\models \phi^k_{d-2}$	$\models \phi^k_d$
2	13	1942.	> 8 hours
4	13	239.	1674.
6	13	80.3	927.
8	11	2.16	18.0
10	9	0.05	1.52
12	7	0.03	0.09
...	...	...	...

HEX Puzzles from the 1940s





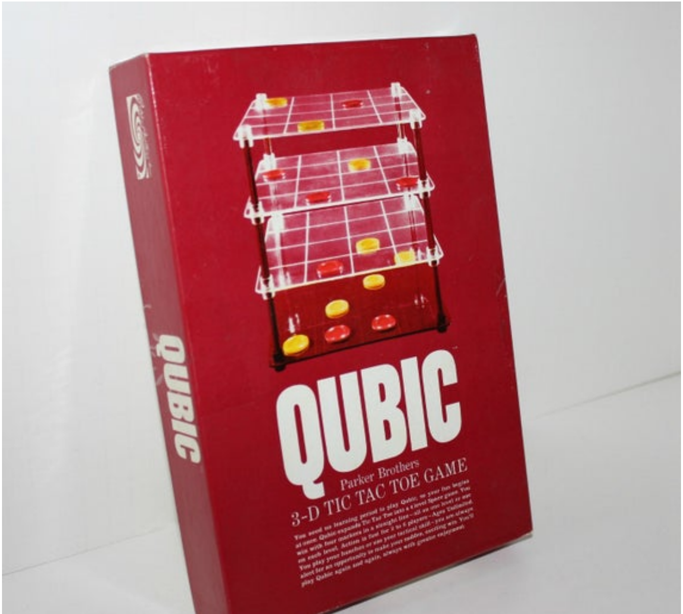




Solving classic HEX puzzles by encoding them through *COR*.

Puzzle	size	depth d	caqe-Q $\not\models \phi_{d-2}$	caqe-Q $\models \phi_d$	depqbf-N $\not\models \phi_{d-2}$	depqbf-N $\models \phi_d$	questo-BQ $\not\models \phi_{d-2}$	questo-BQ $\models \phi_d$
Hein 04	3x3	05	0.01	0.01	0.02	0.02	0.01	0.00
Hein 09	4x4	07	0.01	0.11	0.03	0.15	0.01	0.06
Hein 12	4x4	07	0.02	0.10	0.05	0.22	0.00	0.02
Hein 07	4x4	09	0.30	4.31	0.33	5.69	0.09	1.66
Hein 06	4x4	13	10.2	15.5	2.95	17.7	3.92	9.79
Hein 13	5x5	09	0.24	15.6	0.72	17.1	0.06	4.61
Hein 14	5x5	09	0.38	19.0	1.24	42.4	0.18	4.40
Hein 11	5x5	11	5.17	240.	21.0	457.	1.84	23.6
Hein 19	5x5	11	2.29	44.4	3.60	80.8	0.91	13.1
Hein 08	5x5	11	4.13	104.	6.84	247.0	1.98	34.4
Hein 10	5x5	13	367.	4906.	443.	10259.	74.3	1543.
Hein 16	5x5	13	651.	8964.	1794.	8506.	278.	4406.
Hein 02	5x5	13	719.	22526.	1258.	10876.	317.	2957.
Hein 15	5x5	15	3247.	26938.	2928.	19469.	767.	MO
Browne	5x5	09	0.87	57.45	0.91	21.2	0.25	2.89

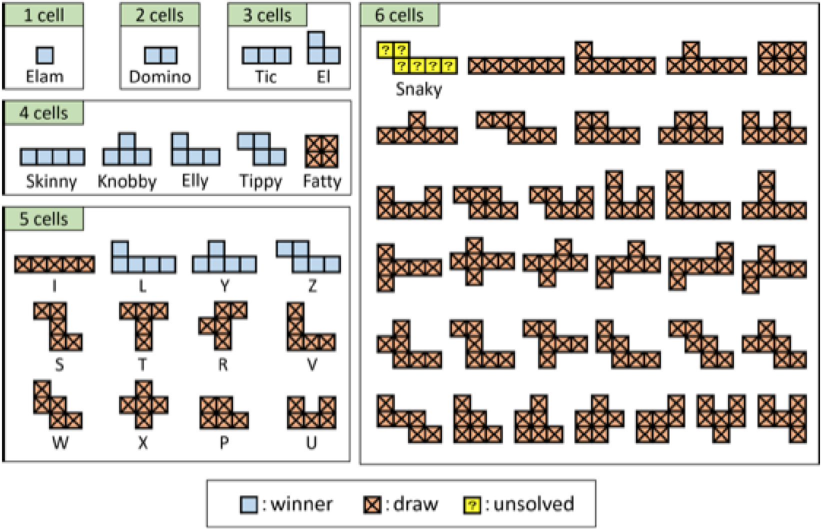
Towards full-scale: Qubic



Towards full-scale: Gomoku (Renju)

Aldis Reims vs Arnis Veidemanis
World Championship in Tallinn, Estonia 1995

Towards full-scale: GTTT with 6 cells



Milestone problems

Challenge problem		First systematic solution	Size in QBF				
Domain	Variant		#qb	# ∀	# ∃	#cl	#lits
QUBIC	$4 \times 4 \times 4$	1980	65	192	29k	80k	246k
SNAKY	9×9	open	81	280	47k	131k	404k
GOMOKU	15×15[2]	1993	225	896	357k	991k	3078k
CONNECT6	19×19[3]	2010	179	1602	511k	1527k	5031k

[2] Freestyle

[3] Mickey Mouse opening

Contributions

- QBF encoding from positional games.
- Improved upon previous encodings by a simpler and more compact encoding.
- Implementation available.
- We can demonstrate this improvement by experiments.

Improvements over DYS GTTT encoding

- $5\times$ more compact encoding of tiny problems
- $30\times$ faster solving of tiny problems
- Can solve “small” problems
- More general encoding (positional games)

> *For the first time, automated Hex solvers have surpassed humans [...]: they can now solve many* 9×9 *Hex openings.*

B. Arneson et al. "Solving hex: beyond humans". In: *Computers and Games*. 2010, pp. 1–10

4

> *For the first time, automated Hex solvers have surpassed humans [...]: they can now solve many 9×9 Hex openings.*

B. Arneson et al. “Solving hex: beyond humans”. In: *Computers and Games*. 2010, pp. 1–10

> *For the first time, QBF-based solvers have surpassed total beginners: they can now solve 5×5 Hex and Tic-Tac-Toe puzzles.*

V. Mayer-Eichberger and A. Saffidine. “QBF solving of positional games: beyond trivial”. In: *SAT*. 2020[4]

---

[4] Not the published title, but that's the spirit!