# Proof-number search based solver for the Sprouts game

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Bc. Tomáš Čížek

*Supervisor:* Mgr. Martin Schmid, Ph.D.

*Advisor:* doc. RNDr. Martin Balko, Ph.D.

# Sprouts rules

A two-player game with simple rules:

- Starts with $n$ initial spots.
- Players alternate in connecting spots by curves (cycles are allowed).
- Curves cannot cross.
- A new spot is added along a newly drawn curve.
- Each spot can be incident to at most three curves.
- The first player with no move loses.

# Sprouts rules

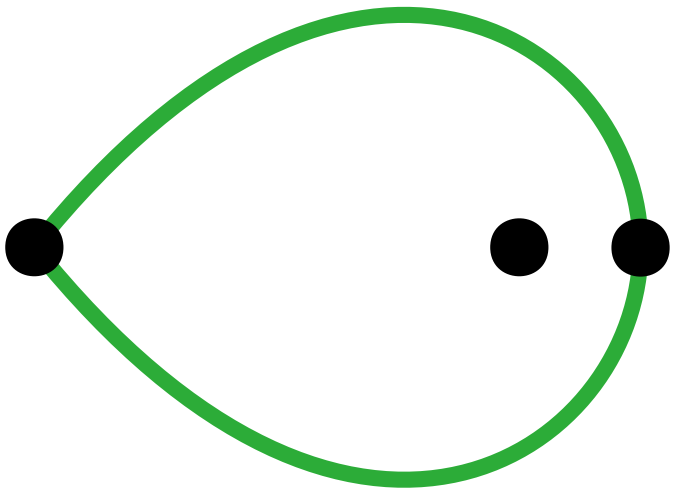
<u>A two-player game with simple rules:</u>

- Starts with *n* initial spots.
- Players alternate in connecting spots by curves (cycles are allowed).
- Curves cannot cross.
- A new spot is added along a newly drawn curve.
- Each spot can be incident to at most three curves.
- The first player with no move loses.

# Sprouts rules

<u>A two-player game with simple rules:</u>

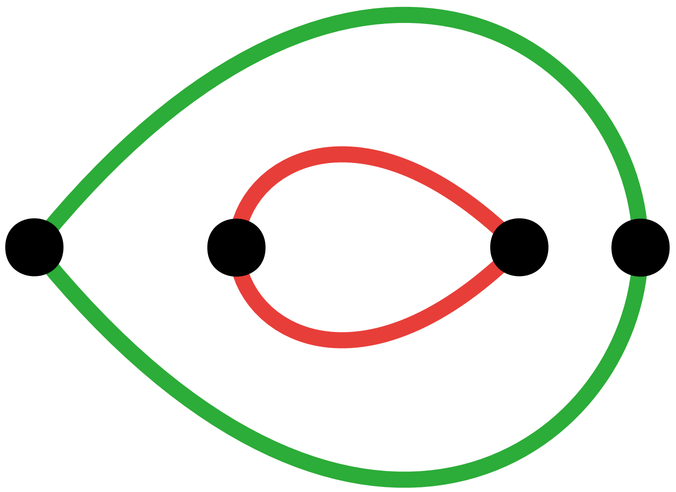
- Starts with $n$ initial spots.
- Players alternate in connecting spots by curves (cycles are allowed).
- Curves cannot cross.
- A new spot is added along a newly drawn curve.
- Each spot can be incident to at most three curves.
- The first player with no move loses.



# Sprouts rules

<u>A two-player game with simple rules:</u>

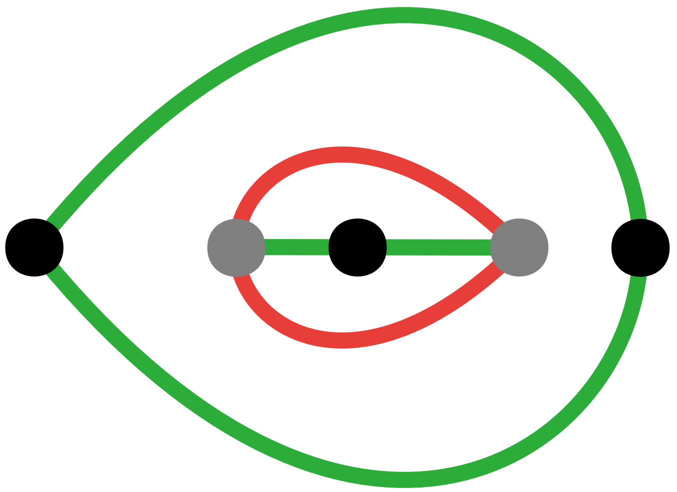
- Starts with *n* initial spots.
- Players alternate in connecting spots by curves (cycles are allowed).
- Curves cannot cross.
- A new spot is added along a newly drawn curve.
- Each spot can be incident to at most three curves.
- The first player with no move loses.



# Sprouts rules

<u>A two-player game with simple rules:</u>

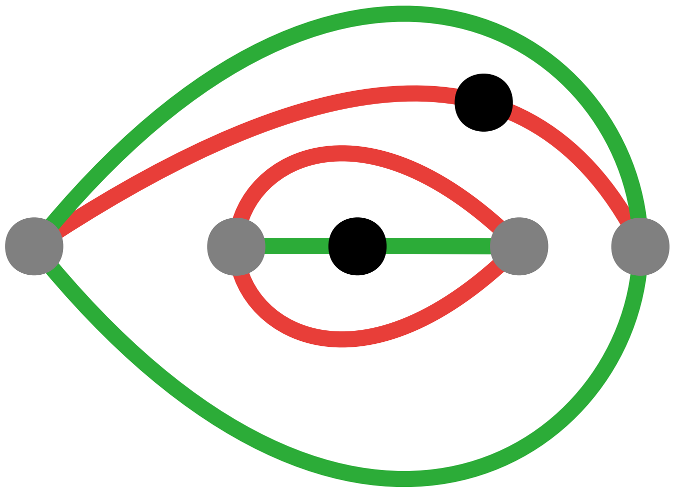
- Starts with *n* initial spots.
- Players alternate in connecting spots by curves (cycles are allowed).
- Curves cannot cross.
- A new spot is added along a newly drawn curve.
- Each spot can be incident to at most three curves.
- The first player with no move loses.



# Sprouts rules

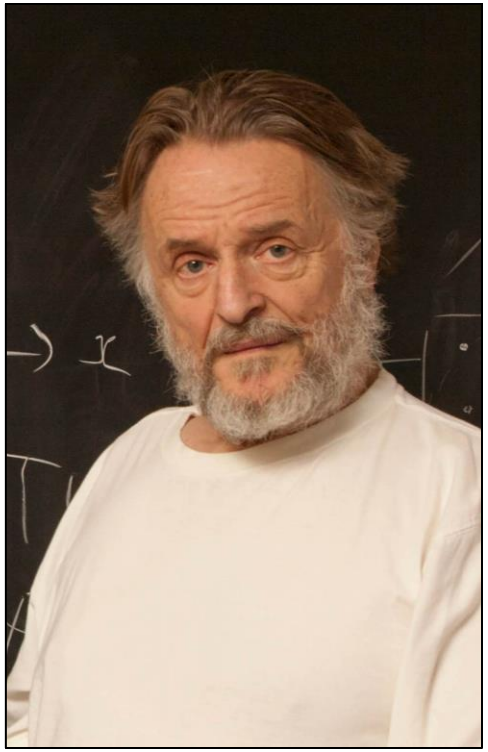
<u>A two-player game with simple rules:</u>

- Starts with *n* initial spots.
- Players alternate in connecting spots by curves (cycles are allowed).
- Curves cannot cross.
- A new spot is added along a newly drawn curve.
- Each spot can be incident to at most three curves.
- The first player with no move loses.

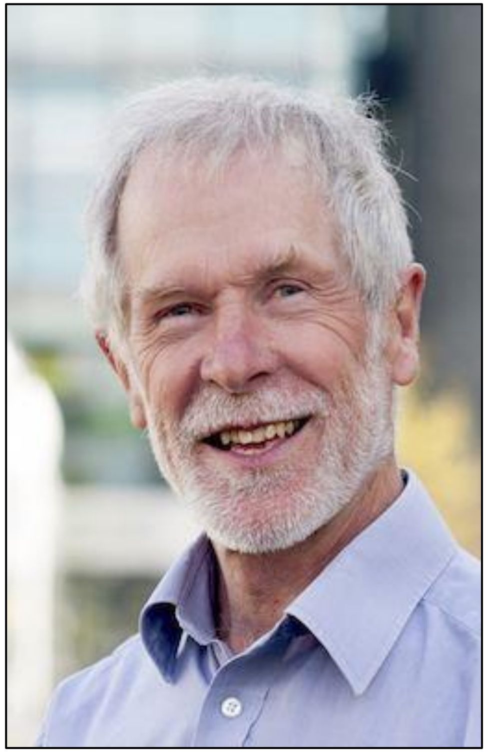


# The early beginnings

- Designed by British mathematicians J. Conway and M. Paterson in 1967.
- Easy to play, difficult to analyze.
- Many attempts to determine outcomes of Sprouts positions with $n$ initial spots under the perfect play (weak solutions).



www.princeton.edu



www.alchetron.com/Mike-Paterson

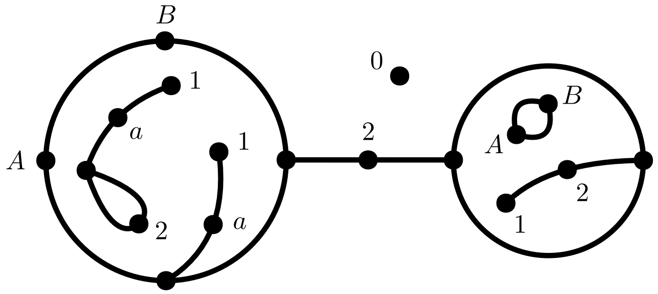
# The early beginnings

- Designed by British mathematicians J. Conway and M. Paterson in 1967.
- Easy to play, difficult to analyze.
- Many attempts to determine outcomes of Sprouts positions with *n* initial spots under the perfect play (weak solutions).

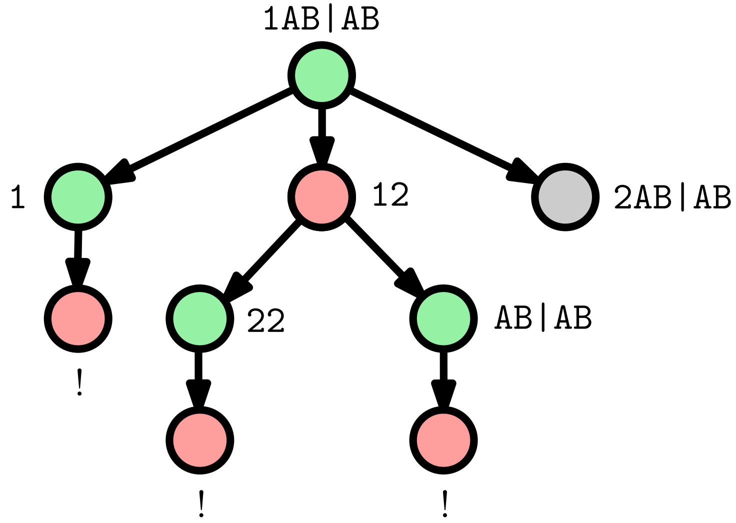
| n | outcome | author |
|---|---|---|
| 1 | L | Conway |
| 2 | L | Conway |
| 3 | W | Conway |
| 4 | W | Mollison |
| 5 | W | Mollison |
| 6 | L | Mollison |
| 7 | ? | ? |
| 8 | ? | ? |
| 9 | ? | ? |
| 10 | ? | ? |

# Searching trees computationally

- Applegate, Jacobson, and Sleator (1991) [1] created the first computer solver for Sprouts.
- A necessity of a state representation (planar embeddings).
- Using a simple Alpha-Beta pruning for search.




0.2AB|1a2a.1aABa+12.AB|AB



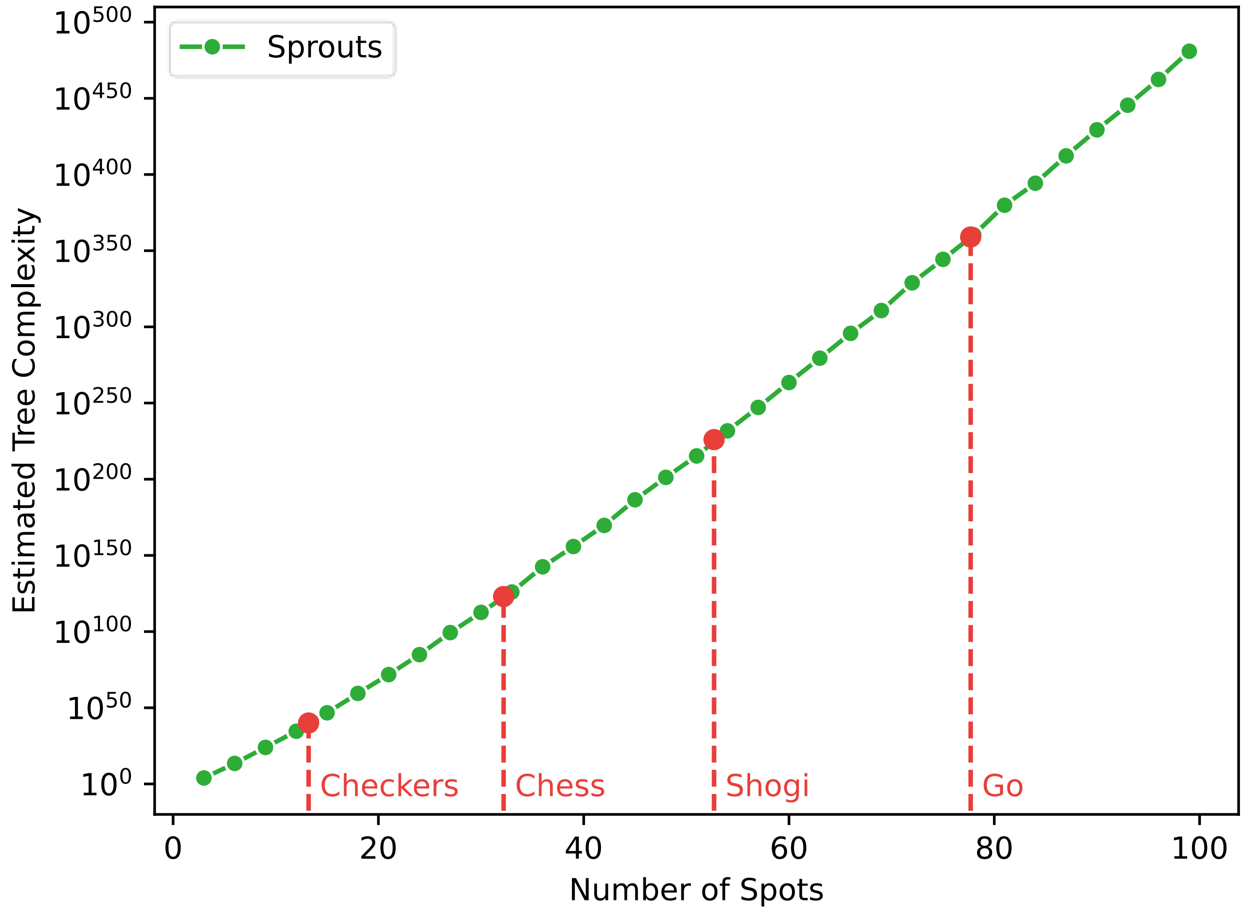
# Searching trees computationally

- Applegate, Jacobson, and Sleator (1991) [1] created the first computer solver for Sprouts.
- A necessity of a state representation (planar embeddings).
- Using a simple Alpha-Beta pruning for search.

| n | outcome | author |
|---|---|---|
| **1** | L | Conway |
| **2** | L | Conway |
| **3** | W | Conway |
| **4** | W | Mollison |
| **5** | W | Mollison |
| **6** | L | Mollison |
| **7** | L | AJS |
| **8** | L | AJS |
| **9** | W | AJS |
| **10** | W | AJS |
| **11** | W | AJS |
| **12** | L | Purinton |
| **13** | L | Purinton |
| **14** | L | Purinton |
| **15** | ? | ? |

- The famous Sprouts conjecture was formed.
  - $n$-spot position is winning $\leftrightarrow$ $n \equiv 3, 4, 5 \pmod 6$.

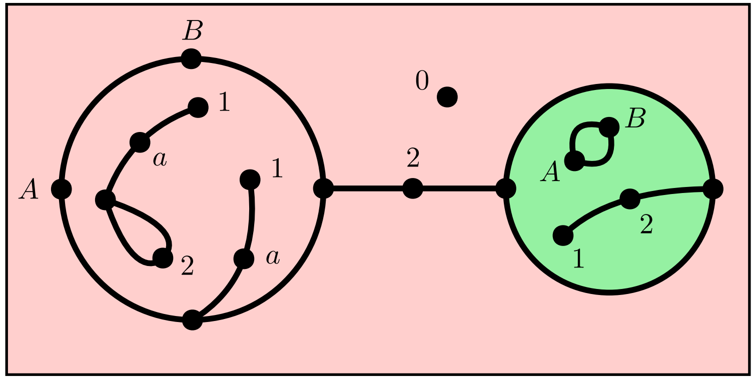
# Game trees grow extremely fast

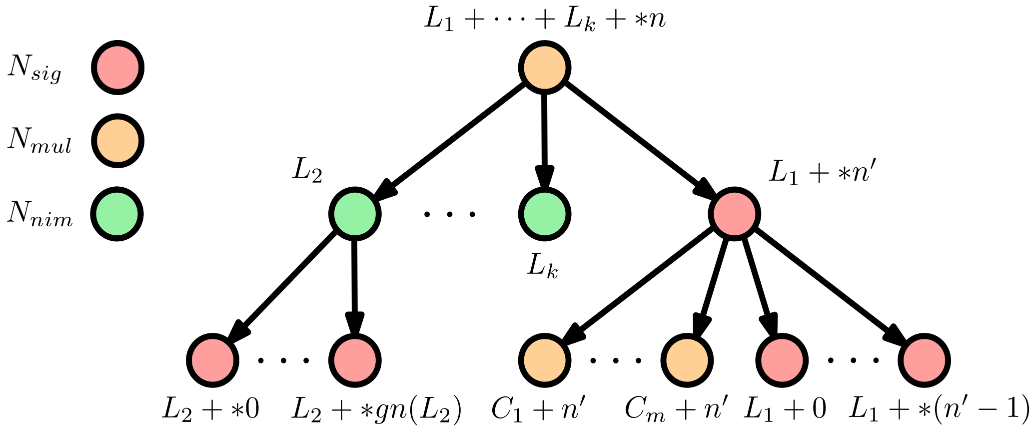


# Grundy numbers

- Lemoine and Viennot (2007) [2] introduced a solver Glop utilizing the Sprague–Grundy theorem.
- Analyze independent parts (lands) separately:
  - Compute Grundy number (nimber) *gn(L)* for each land *L*.
  - $gn(L_1 + L_2) = gn(L_1) \oplus gn(L_2)$.
  - $L_1 + L_2$ is loss $\leftrightarrow$ $gn(L) = 0$.

→ More complicated NAND trees with Grundy numbers.



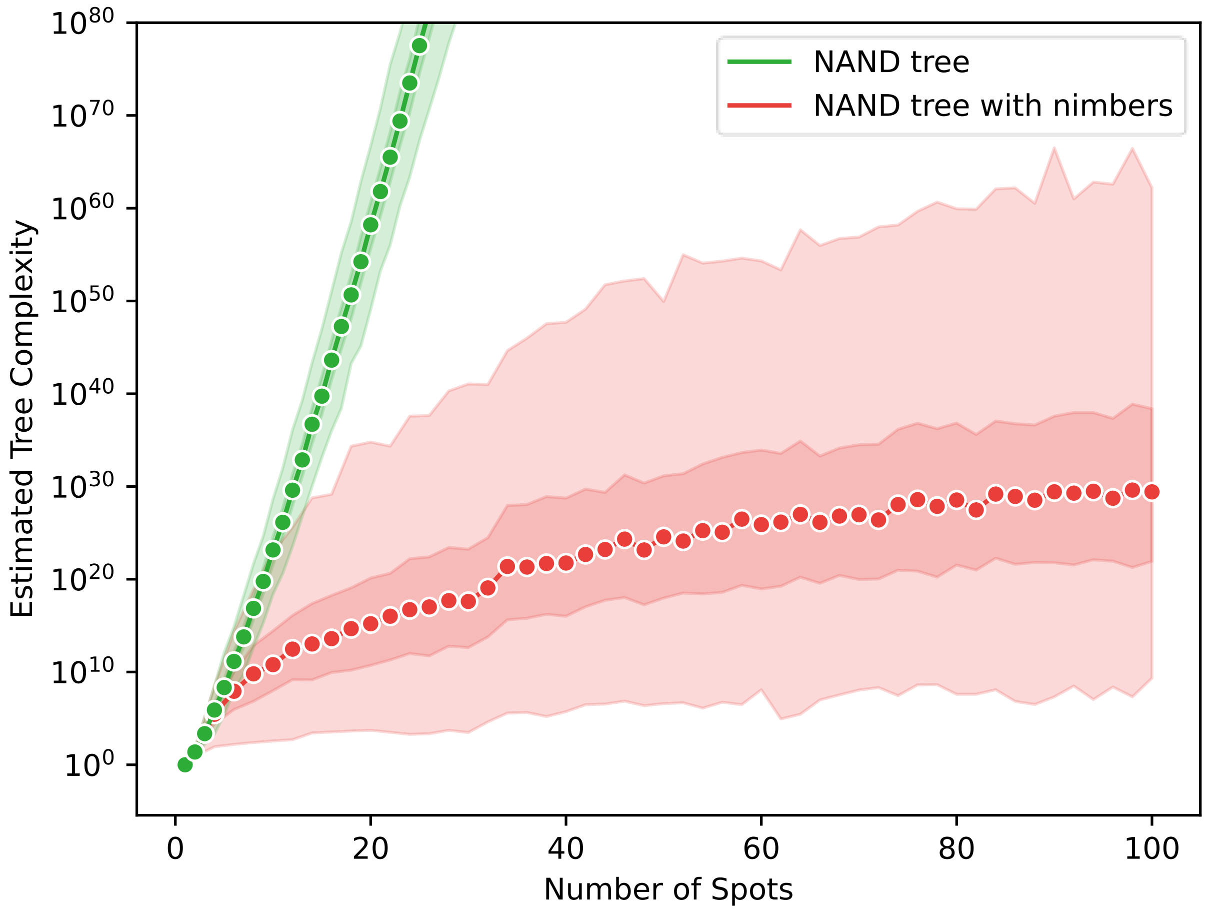
# Grundy numbers

- Lemoine and Viennot (2007) [2] introduced a solver Glop utilizing the Sprague–Grundy theorem.
- Analyze independent parts (lands) separately:
  - Compute Grundy number (nimber) $gn(L)$ for each land $L$.
  - $gn(L_1 + L_2) = gn(L_1) \oplus gn(L_2)$.
  - $L_1 + L_2$ is loss $\leftrightarrow$ $gn(L) = 0$.

→ More complicated NAND trees with Grundy numbers.

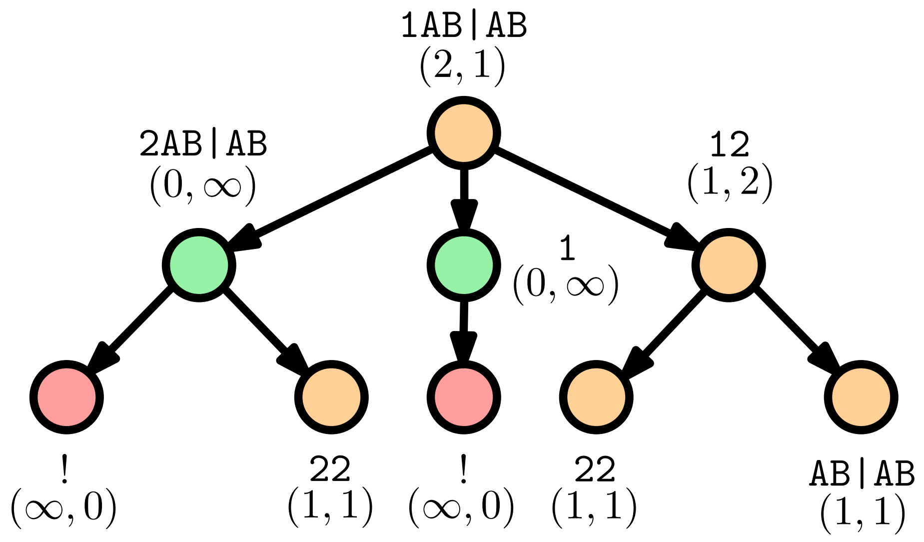
| n | out. | auth. | n | out. | auth. | n | out. | auth. |
|---|---|---|---|---|---|---|---|---|
| 1 | L | Conway | 19 | L | Glop '07 | 37 | ? | ? |
| 2 | L | Conway | 20 | L | Glop '07 | 38 | ? | ? |
| 3 | W | Conway | 21 | W | Glop '07 | 39 | ? | ? |
| 4 | W | Mollison | 22 | W | Glop '07 | 40 | W | Glop '07 |
| 5 | W | Mollison | 23 | W | Glop '07 | 41 | W | Glop '07 |
| 6 | L | Mollison | 24 | L | Glop '07 | 42 | ? | ? |
| 7 | L | AJS | 25 | L | Glop '07 | 43 | ? | ? |
| 8 | L | AJS | 26 | L | Glop '07 | 44 | ? | ? |
| 9 | W | AJS | 27 | W | Glop '07 | 45 | ? | ? |
| 10 | W | AJS | 28 | W | Glop '07 | 46 | ? | ? |
| 11 | W | AJS | 29 | W | Glop '07 | 47 | W | Glop '07 |
| 12 | L | Purinton | 30 | L | Glop '07 | 48 | ? | ? |
| 13 | L | Purinton | 31 | L | Glop '07 | 49 | ? | ? |
| 14 | L | Purinton | 32 | L | Glop '07 | 50 | ? | ? |
| 15 | W | Glop '07 | 33 | ? | ? | 51 | ? | ? |
| 16 | W | Glop '07 | 34 | W | Glop '07 | 52 | ? | ? |
| 17 | W | Glop '07 | 35 | W | Glop '07 | 53 | ? | ? |
| 18 | L | Glop '07 | 36 | ? | ? | 54 | ? | ? |

# Largely imbalanced game trees



# Proof-number search

- Alpha-beta pruning can get stuck in difficult subtrees if the heuristic was wrong.
- Explore subtrees with potentially shortest proof → Proof-number search (PNS) [4]:
  - Each node *N* is associated with a proof number *pn(N)* and a disproof number *dn(N)*.
  - Initialized heuristically in leaves.
  - Aggregated from children in expanded nodes.
  - Expand the most-proving node (MPN) and update.
- Must be adapted for NAND trees with Grundy numbers!
  - A basic variant by Lemoine (and Viennot) in Glop [3].



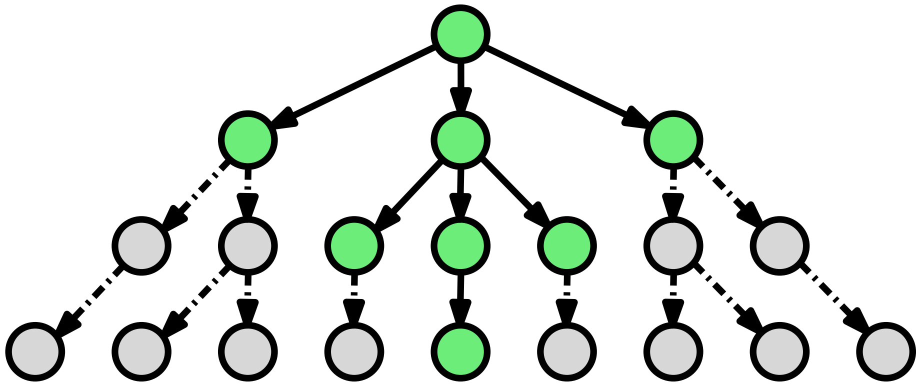
# Proof-number search

- Alpha-beta pruning can get stuck in difficult subtrees if the heuristic was wrong.
- Explore subtrees with potentially shortest proof → Proof-number search (PNS) [4]:
  - Each node *N* is associated with a proof number *pn(N)* and a disproof number *dn(N)*.
  - Initialized heuristically in leaves.
  - Aggregated from children in expanded nodes.
  - Expand the most-proving node (MPN) and update.
- Must be adapted for NAND trees with Grundy numbers!
  - A basic variant by Lemoine (and Viennot) in Glop [3].

| n | out. | auth. | n | out. | auth. | n | out. | auth. |
|---|---|---|---|---|---|---|---|---|
| **1** | L | Conway | **19** | L | Glop '07 | **37** | L | Glop '10 |
| **2** | L | Conway | **20** | L | Glop '07 | **38** | L | Glop '10 |
| **3** | W | Conway | **21** | W | Glop '07 | **39** | W | Glop '11 |
| **4** | W | Mollison | **22** | W | Glop '07 | **40** | W | Glop '07 |
| **5** | W | Mollison | **23** | W | Glop '07 | **41** | W | Glop '07 |
| **6** | L | Mollison | **24** | L | Glop '07 | **42** | L | Glop '11 |
| **7** | L | AJS | **25** | L | Glop '07 | **43** | L | Glop '11 |
| **8** | L | AJS | **26** | L | Glop '07 | **44** | L | Glop '11 |
| **9** | W | AJS | **27** | W | Glop '07 | **45** | ? | ? |
| **10** | W | AJS | **28** | W | Glop '07 | **46** | W | Glop '10 |
| **11** | W | AJS | **29** | W | Glop '07 | **47** | W | Glop '07 |
| **12** | L | Purinton | **30** | L | Glop '07 | **48** | ? | ? |
| **13** | L | Purinton | **31** | L | Glop '07 | **49** | ? | ? |
| **14** | L | Purinton | **32** | L | Glop '07 | **50** | ? | ? |
| **15** | W | Glop '07 | **33** | W | Glop '10 | **51** | ? | ? |
| **16** | W | Glop '07 | **34** | W | Glop '07 | **52** | ? | ? |
| **17** | W | Glop '07 | **35** | W | Glop '07 | **53** | W | Glop '10 |
| **18** | L | Glop '07 | **36** | L | Glop '10 | **54** | ? | ? |

# Depth-first Proof-number search

- The space complexity of PNS is too large (the whole tree is stored in memory).
- Depth-first Proof-number search (df-pn) [5]:
  - A recursive and memory-efficient variant of PNS (logarithmic space complexity).
  - Two thresholds $pt(N_{curr})$ and $dt(N_{curr})$ to guarantee MPN in the subtree of $N_{curr}$.
  - Combined with a transposition table (replacing policy) and a nimber database.



# Depth-first Proof-number search

- The space complexity of PNS is too large (the whole tree is stored in memory).
- Depth-first Proof-number search (df-pn) [5]:
  - A recursive and memory-efficient variant of PNS (logarithmic space complexity).
  - Two thresholds $pt(N_{curr})$ and $dt(N_{curr})$ to guarantee MPN in the subtree of $N_{curr}$.
  - Combined with a transposition table (replacing policy) and a nimber database.

- The update rules for $N_{next}$:

$$pt(N_{next}) = dt(N_{curr}) - dn_0(N_{curr}) + pn_0(N_{next}),$$
$$dt(N_{next}) = \min\{pt(N_{curr}), dn(N_{next_2}) + 1\}.$$

# Depth-first Proof-number search

- The space complexity of PNS is too large (the whole tree is stored in memory).
- Depth-first Proof-number search (df-pn) [5]:
  - A recursive and memory-efficient variant of PNS (logarithmic space complexity).
  - Two thresholds $pt(N_{curr})$ and $dt(N_{curr})$ to guarantee MPN in the subtree of $N_{curr}$.
  - Combined with a transposition table (replacing policy) and a nimber database.

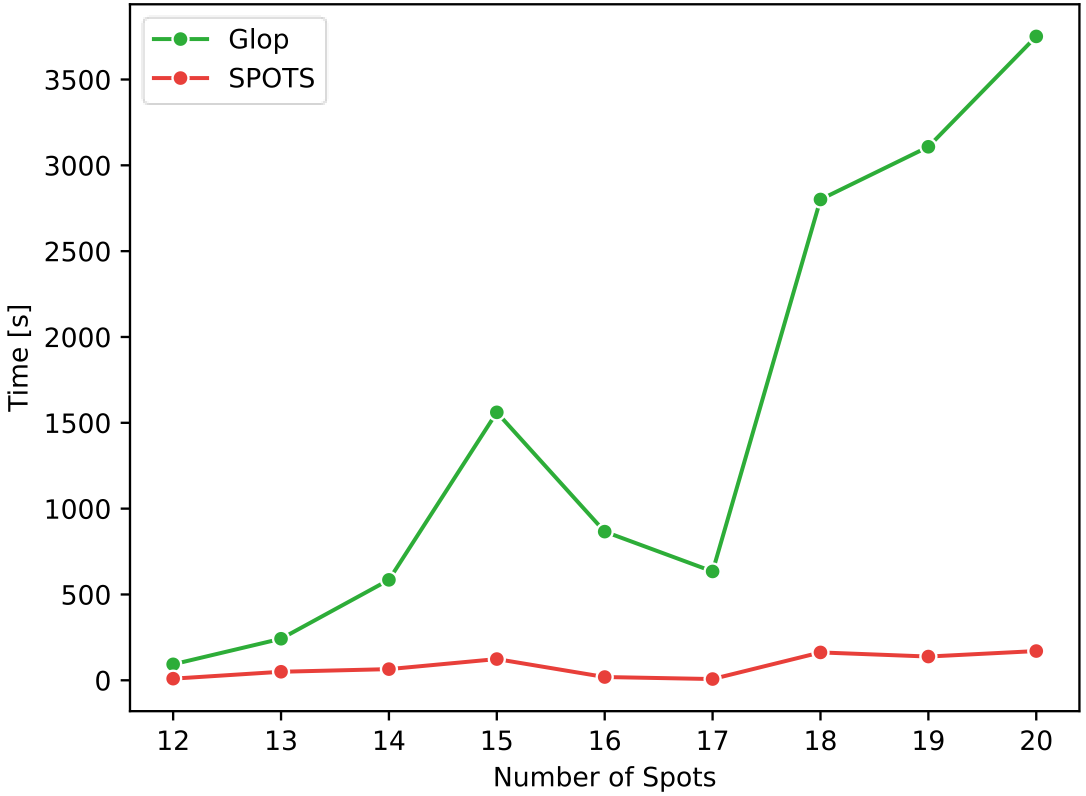
- Extended update rules for $N_{next}$ with Grundy numbers:

$$\begin{aligned}
pt(N_{next}) &= dt(N_{curr}) - dn_0(N_{curr}) + pn_0(N_{next}),\\
dt(N_{next}) &= \min\{pt(N_{curr}), dn(N_{next_2}) + 1\},\\
mint(N_{next}) &= mint(N_{curr}),\\
pS(N_{next}) &= dS(N_{curr}) + dn_0(N_{curr}) - pn_0(N_{next}),\\
dS(N_{next}) &= pS(N_{curr}).
\end{aligned}$$

$$\begin{aligned}
pt(N_{next}) = dt(N_{next}) &= \infty,\\
mint(N_{next}) &= \mathrm{thMins}_{curr} - \mathrm{otherMins}_{curr,next},\\
pS(N_{next}) = dS(N_{next}) &= 0,
\end{aligned}$$

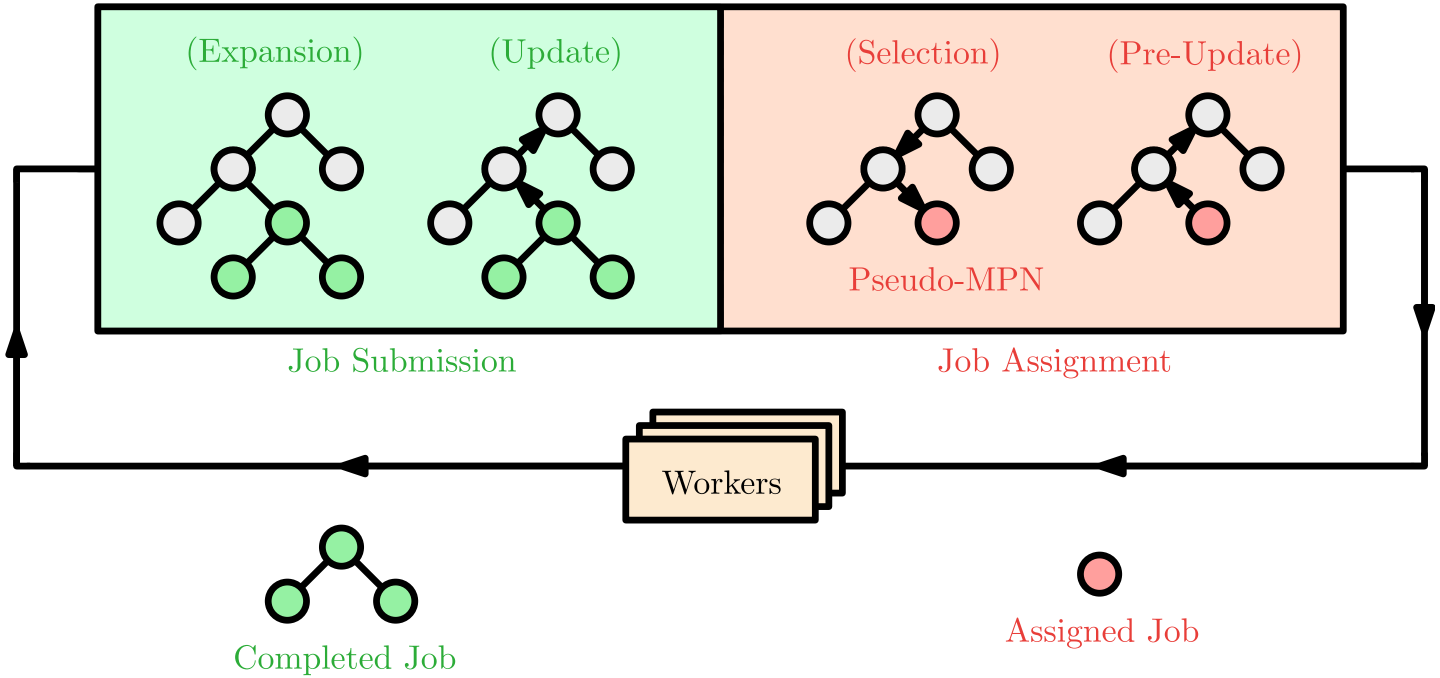
# Our sequential solver SPOTS

- Approximately 18 times faster than Glop + memory advantageous df-pn.



# Parallel Proof-number search

- Adapting PNS for large computational cluster with distributed memory.
- Master-workers scheme:



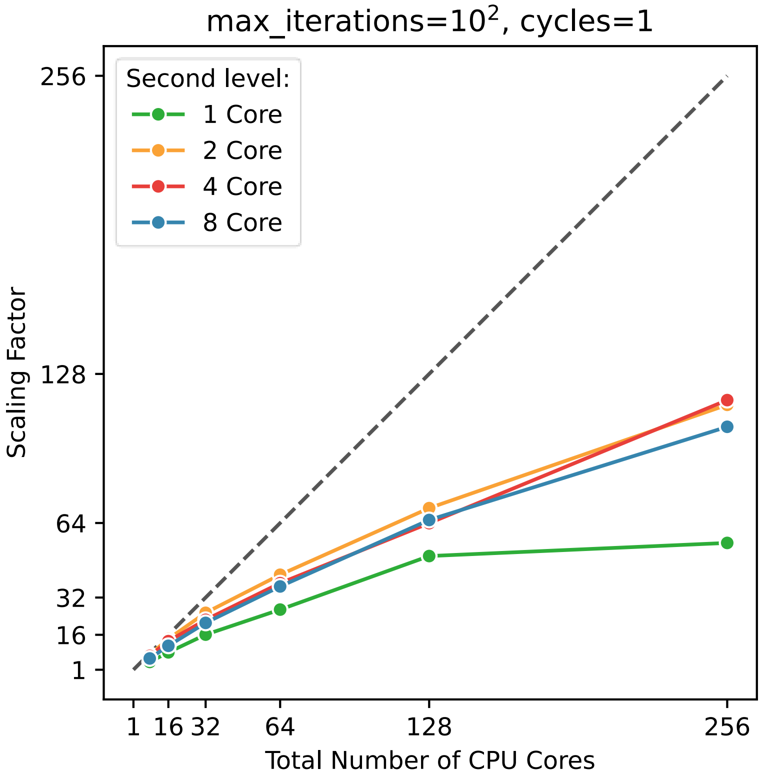
# $1^{st}$ improvement — share results

- Share Grundy numbers with other workers to prevent search overhead:
  - <u>Small number</u> – no additional communication overhead.
  - <u>Highly reusable</u> – a land may occur in many positions.

| | Sharing | 1 | 2 | 4 | 8 | Eff. |
|---|---|---|---|---|---|---|
| **Iterations**<br>**Time** | ✗ | 393k<br>374 s | 843k<br>371 s | 1,470k<br>281 s | 2,180k<br>190 s | **36%** |
| **Iterations**<br>**Time** | ✓ | 398k<br>374 s | 572k<br>259 s | 818k<br>174 s | 847k<br>81 s | **61%** |

# $2^{nd}$ improvement — $P^2PNS$

- Parallelize the workers themselves rather than adding more of them:
  - Reduced communication overhead, more relevant work, local memory utilization.



# Scaling efficiency of $P^2$PNS

- Reaching roughly 100x speedup (480 cores) compared to the sequential df-pn.
- Much better scaling efficiency than other parallel PNS:

**Current PPN$^2$ search [6]**
35% on 32 CPU cores

✖

**Our $P^2$PNS**
34% on 480 CPU cores

→ Addresses the scaling problem of PNS posed by Kishimoto et al. [7].

| $P^2$PNS | Iterations | Time | Scaling | Efficiency | Speedup |
|---|---|---|---|---|---|
| **df-pn** | 3,150k | 47.5 min | — | — | **1.00x** |
| **1 core** | 3,650k | 78.0 min | 1.00x | 100% | **0.61x** |
| **2 cores** | 3,680k | 39.3 min | 1.98x | 99% | **1.21x** |
| **4 cores** | 3,260k | 18.8 min | 4.14x | 104% | **2.53x** |
| **8 cores** | 4,190k | 11.6 min | 6.72x | 84% | **4.09x** |
| **16 cores** | 3,760k | 294 s | 15.9x | 99% | **9.70x** |
| **32 cores** | 4,492k | 178 s | 26.3x | 81% | **16.0x** |
| **64 cores** | 4,120k | 88.4 s | 52.9x | 83% | **32.2x** |
| **128 cores** | 5,710k | 54.1 s | 86.5x | 68% | **52.7x** |
| **256 cores** | 7,290k | 36.6 s | 128x | 50% | **78.3x** |
| **480 cores** | 8,710k | 28.1 s | 167x | 34% | **101x** |
| **960 cores** | 13,500k | 21.4 s | 219x | 23% | **133x** |

# Resulting solver

- SPOTS roughly 2800x faster (480 cores) than Glop.

→ **1 day of SPOTS ≈ 8 years of Glop**

# Resulting solver

- SPOTS roughly 2800x faster (480 cores) than Glop.

→ **1 day of SPOTS ≈ 8 years of Glop**

- Outcome of 47 *n*-spot positions known until now.

| n | out. | auth. | n | out. | auth. | n | out. | auth. |
|---|---|---|---|---|---|---|---|---|
| **1** | L | Conway | **36** | L | Glop '10 | 71 | ? | ? |
| **2** | L | Conway | **37** | L | Glop '10 | 72 | ? | ? |
| **3** | W | Conway | **38** | L | Glop '10 | 73 | ? | ? |
| **4** | W | Mollison | **39** | W | Glop '11 | 74 | ? | ? |
| **5** | W | Mollison | **40** | W | Glop '07 | 75 | ? | ? |
| **6** | L | Mollison | **41** | W | Glop '07 | 76 | ? | ? |
| **7** | L | AJS | **42** | L | Glop '11 | 77 | ? | ? |
| **8** | L | AJS | **43** | L | Glop '11 | 78 | ? | ? |
| **9** | W | AJS | **44** | L | Glop '11 | 79 | ? | ? |
| **10** | W | AJS | 45 | ? | ? | 80 | ? | ? |
| **11** | W | AJS | **46** | W | Glop '10 | 81 | ? | ? |
| **12** | L | Purinton | **47** | w | Glop '07 | 82 | ? | ? |
| **13** | L | Purinton | 48 | ? | ? | 83 | ? | ? |
| **14** | L | Purinton | 49 | ? | ? | 84 | ? | ? |
| **15** | W | Glop '07 | 50 | ? | ? | 85 | ? | ? |
| **16** | W | Glop '07 | 51 | ? | ? | 86 | ? | ? |
| **17** | W | Glop '07 | 52 | ? | ? | 87 | ? | ? |
| **18** | L | Glop '07 | **53** | W | Glop '10 | 88 | ? | ? |
| **19** | L | Glop '07 | 54 | ? | ? | 89 | ? | ? |
| **20** | L | Glop '07 | 55 | ? | ? | 90 | ? | ? |
| **21** | W | Glop '07 | 56 | ? | ? | 91 | ? | ? |
| **22** | W | Glop '07 | 57 | ? | ? | 92 | ? | ? |
| **23** | W | Glop '07 | 58 | ? | ? | 93 | ? | ? |
| **24** | L | Glop '07 | 59 | ? | ? | 94 | ? | ? |
| **25** | L | Glop '07 | 60 | ? | ? | 95 | ? | ? |
| **26** | L | Glop '07 | 61 | ? | ? | 96 | ? | ? |
| **27** | W | Glop '07 | 62 | ? | ? | 97 | ? | ? |
| **28** | W | Glop '07 | 63 | ? | ? | 98 | ? | ? |
| **29** | W | Glop '07 | 64 | ? | ? | 99 | ? | ? |
| **30** | L | Glop '07 | 65 | ? | ? | 100 | ? | ? |
| **31** | L | Glop '07 | 66 | ? | ? | 101 | ? | ? |
| **32** | L | Glop '07 | 67 | ? | ? | 102 | ? | ? |
| **33** | W | Glop '10 | 68 | ? | ? | 103 | ? | ? |
| **34** | W | Glop '07 | 69 | ? | ? | 104 | ? | ? |
| **35** | W | Glop '07 | 70 | ? | ? | 105 | ? | ? |

# Resulting solver

- SPOTS roughly 2800x faster (480 cores) than Glop.

→ **1 day of SPOTS ≈ 8 years of Glop**

- Outcome of 47 *n*-spot positions known until now.

→ **We compute 42 new outcomes!**

  - The largest proof is 1000x larger than the largest so far (took 24 days to compute ≈ 280,000 CPU hours).
  - The Sprouts conjecture remains open.

| n | out. | auth. | n | out. | auth. | n | out. | auth. |
|---|---|---|---|---|---|---|---|---|
| 1 | L | Conway | 36 | L | Glop '10 | 71 | W | SPOTS |
| 2 | L | Conway | 37 | L | Glop '10 | 72 | ? | ? |
| 3 | W | Conway | 38 | L | Glop '10 | 73 | L | SPOTS |
| 4 | W | Mollison | 39 | W | Glop '11 | 74 | L | SPOTS |
| 5 | W | Mollison | 40 | W | Glop '07 | 75 | ? | ? |
| 6 | L | Mollison | 41 | W | Glop '07 | 76 | W | SPOTS |
| 7 | L | AJS | 42 | L | Glop '11 | 77 | W | SPOTS |
| 8 | L | AJS | 43 | L | Glop '11 | 78 | ? | ? |
| 9 | W | AJS | 44 | L | Glop '11 | 79 | L | SPOTS |
| 10 | W | AJS | 45 | W | SPOTS | 80 | L | SPOTS |
| 11 | W | AJS | 46 | W | Glop '10 | 81 | ? | ? |
| 12 | L | Purinton | 47 | W | Glop '07 | 82 | W | SPOTS |
| 13 | L | Purinton | 48 | L | SPOTS | 83 | W | SPOTS |
| 14 | L | Purinton | 49 | L | SPOTS | 84 | ? | ? |
| 15 | W | Glop '07 | 50 | L | SPOTS | 85 | L | SPOTS |
| 16 | W | Glop '07 | 51 | W | SPOTS | 86 | L | SPOTS |
| 17 | W | Glop '07 | 52 | W | SPOTS | 87 | ? | ? |
| 18 | L | Glop '07 | 53 | W | Glop '10 | 88 | W | SPOTS |
| 19 | L | Glop '07 | 54 | L | SPOTS | 89 | W | P-SPOTS |
| 20 | L | Glop '07 | 55 | L | SPOTS | 90 | ? | ? |
| 21 | W | Glop '07 | 56 | L | SPOTS | 91 | L | SPOTS |
| 22 | W | Glop '07 | 57 | W | P-SPOTS | 92 | L | SPOTS |
| 23 | W | Glop '07 | 58 | W | SPOTS | 93 | ? | ? |
| 24 | L | Glop '07 | 59 | W | SPOTS | 94 | W | P-SPOTS |
| 25 | L | Glop '07 | 60 | L | P-SPOTS | 95 | W | P-SPOTS |
| 26 | L | Glop '07 | 61 | L | SPOTS | 96 | ? | ? |
| 27 | W | Glop '07 | 62 | L | SPOTS | 97 | L | P-SPOTS |
| 28 | W | Glop '07 | 63 | ? | ? | 98 | L | P-SPOTS |
| 29 | W | Glop '07 | 64 | W | SPOTS | 99 | ? | ? |
| 30 | L | Glop '07 | 65 | W | SPOTS | 100 | W | P-SPOTS |
| 31 | L | Glop '07 | 66 | ? | ? | 101 | ? | ? |
| 32 | L | Glop '07 | 67 | L | SPOTS | 102 | ? | ? |
| 33 | W | Glop '10 | 68 | L | SPOTS | 103 | L | P-SPOTS |
| 34 | W | Glop '07 | 69 | ? | ? | 104 | L | P-SPOTS |
| 35 | W | Glop '07 | 70 | W | SPOTS | 105 | ? | ? |

# Conclusion

- Formalized the NAND trees with Grundy numbers.
- **We extended df-pn for NAND  trees with Grundy numbers.**
- New observations about learning heuristics for PNS.
- **The new well-scaling parallel variant $P^2$PNS of PNS.**
- **We almost doubled the verified number of spots for the Sprouts conjecture.**

# Conclusion

- Formalized the NAND trees with Grundy numbers.
- **We extended df-pn for NAND  trees with Grundy numbers.**
- New observations about learning heuristics for PNS.
- **The new well-scaling parallel variant $P^2$PNS of PNS.**
- **We almost doubled the verified number of spots for the Sprouts conjecture.**

Thank you for your attention.

# References


- [1] D. Applegate, G. Jacobson, and D. Sleator. Computer Analysis of Sprouts. Carnegie Mellon University Computer Science technical report CMU-CS-91-144. 1991.
- [2] J. Lemoine and S. Viennot. “Computer Analysis of Sprouts with Nimbers”. In: Games of no chance 4. Vol. 63. Math. Sci. Res. Inst. Publ. Cambridge Univ. Press, New York, 2015, pp. 161–181.
- [3] J. Lemoine. “Méthodes Algorithmiques pour la Résolution des Jeux Combinatoires”. In French. Ph.D. thesis. Université des Sciences et Technologie de Lille - Lille I, 2011.
- [4] L. V. Allis. “Searching for Solutions in Games and Artificial Intelligence”. Ph.D. thesis. Rijksuniversiteit Limburg, 1994.
- [5] A. Nagai. “A New Depth-First-Search Algorithm for AND/OR trees and its Applications”. M.Sc. thesis. The University of Tokyo, 1999.
- [6] A. Saffidine, N. Jouandeau, and T. Cazenave. “Solving Breakthrough with Race Patterns and Job-Level Proof Number Search”. In: Advances in Computer Games. Ed. by H. J. van den Herik and A. Plaat. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 196–207.
- [7] A. Kishimoto, M. H. M. Winands, M. Müller, and J. T. Saito. “Game-Tree Search Using Proof Numbers: The First Twenty Years”. In: ICGA Journal 35.3 (2012), pp. 131–156.

# Learning initialization rules

| Heuristics | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| LR `0*6` | 0.68 | 0.82 | 0.93 | 0.84 | 0.47 | 0.70 | 0.76 | 0.58 | 0.54 | **0.69** |
| LR `0*7` | 0.43 | 0.84 | 0.50 | 0.75 | 0.52 | 0.67 | 0.85 | 0.38 | 0.71 | **0.60** |
| LR `0*8` | 0.64 | 0.56 | 0.54 | 0.70 | 0.50 | 0.65 | 0.70 | 0.56 | 0.65 | **0.61** |
| LR `0*12` | 1.01 | 0.59 | 0.52 | 0.94 | 0.65 | 1.04 | 0.53 | 0.41 | 0.81 | **0.69** |
| LR `0*18` | 0.76 | 0.81 | 0.37 | 1.13 | 0.50 | 1.04 | 0.64 | 0.43 | 0.71 | **0.67** |
| Lives | 0.97 | 0.72 | 0.41 | 0.89 | 0.51 | 0.75 | 0.73 | 0.50 | 1.03 | **0.69** |
| Exp `0*6` | 2.48 | 24.0 | — | — | — | — | — | — | — | — |
| Exp Lives | 1.31 | 3.34 | 6.38 | 12.4 | 37.9 | — | — | — | — | — |

# The resulting solver performance

| **Solver** | **0∗27** | **0∗33** | **0∗39** | **0∗45** | **0∗51** | **Speedup** |
|---|---|---|---|---|---|---|
| **Glop** [31, 32] | $\approx$ 3.7 d | $\approx$ 21 d | $\approx$ 153 d | $\approx$ 180 d | $\approx$ 2.4 y | — |
| **Seq-SPOTS** | 4.9 h | 28 h | 8.5 d | 10 d | $\approx$ 7 w | **18x** |
| **Par-SPOTS** | 2.8 min | 13 min | 35 min | 2.5 h | 12 h | **2,800x** |

# Solutions complexities

| n | auth. | size | n | auth. | size | n | auth. | size |
|---|---|---|---|---|---|---|---|---|
| 1 | Conway | 2e0 | 36 | Glop '10 | 1e6 | 71 | SPOTS | 1e7 |
| 2 | Conway | 4e0 | 37 | Glop '10 | 1e5 | 72 | ? | ? |
| 3 | Conway | 7e0 | 38 | Glop '10 | 1e5 | 73 | SPOTS | 1e7 |
| 4 | Mollison | 2e1 | 39 | Glop '11 | 1e6 | 74 | SPOTS | 1e7 |
| 5 | Mollison | 3e1 | 40 | Glop '07 | 1e5 | 75 | ? | ? |
| 6 | Mollison | 9e1 | 41 | Glop '07 | 2e5 | 76 | SPOTS | 1e7 |
| 7 | AJS | 2e2 | 42 | Glop '11 | 1e6 | 77 | SPOTS | 1e7 |
| 8 | AJS | 3e2 | 43 | Glop '11 | 1e6 | 78 | ? | ? |
| 9 | AJS | 1e1 | 44 | Glop '11 | 1e6 | 79 | SPOTS | 1e7 |
| 10 | AJS | 3e2 | 45 | SPOTS | 3e6 | 80 | SPOTS | 1e7 |
| 11 | AJS | 2e2 | 46 | Glop '10 | 2e5 | 81 | ? | ? |
| 12 | Purinton | 1e3 | 47 | Glop '07 | 2e5 | 82 | SPOTS | 1e7 |
| 13 | Purinton | 1e3 | 48 | SPOTS | 3e6 | 83 | SPOTS | 1e7 |
| 14 | Purinton | 1e4 | 49 | SPOTS | 3e6 | 84 | ? | ? |
| 15 | Glop '07 | 9e4 | 50 | SPOTS | 3e6 | 85 | SPOTS | 2e7 |
| 16 | Glop '07 | 1e3 | 51 | SPOTS | 1e7 | 86 | SPOTS | 2e7 |
| 17 | Glop '07 | 6e2 | 52 | SPOTS | 3e6 | 87 | ? | ? |
| 18 | Glop '07 | 7e3 | 53 | Glop '10 | 8e5 | 88 | SPOTS | 2e7 |
| 19 | Glop '07 | 9e3 | 54 | SPOTS | 1e7 | 89 | P-SPOTS | 1e9 |
| 20 | Glop '07 | 1e4 | 55 | SPOTS | 3e6 | 90 | ? | ? |
| 21 | Glop '07 | 8e4 | 56 | SPOTS | 3e6 | 91 | SPOTS | 2e7 |
| 22 | Glop '07 | 6e3 | 57 | P-SPOTS | 1e9 | 92 | SPOTS | 2e7 |
| 23 | Glop '07 | 4e3 | 58 | SPOTS | 3e6 | 93 | ? | ? |
| 24 | Glop '07 | 5e4 | 59 | SPOTS | 7e5 | 94 | P-SPOTS | 1e9 |
| 25 | Glop '07 | 2e4 | 60 | P-SPOTS | 1e9 | 95 | P-SPOTS | 1e9 |
| 26 | Glop '07 | 4e4 | 61 | SPOTS | 3e6 | 96 | ? | ? |
| 27 | Glop '07 | 3e5 | 62 | SPOTS | 3e6 | 97 | P-SPOTS | 1e9 |
| 28 | Glop '07 | 1e4 | 63 | ? | ? | 98 | P-SPOTS | 1e9 |
| 29 | Glop '07 | 1e4 | 64 | SPOTS | 3e6 | 99 | ? | ? |
| 30 | Glop '07 | 2e5 | 65 | SPOTS | 1e7 | 1e9 | P-SPOTS | 1e9 |
| 31 | Glop '07 | 5e4 | 66 | ? | ? | 101 | ? | ? |
| 32 | Glop '07 | 7e4 | 67 | SPOTS | 1e7 | 102 | ? | ? |
| 33 | Glop '10 | 1e6 | 68 | SPOTS | 1e7 | 103 | P-SPOTS | 1e9 |
| 34 | Glop '07 | 3e4 | 69 | ? | ? | 104 | P-SPOTS | 1e9 |
| 35 | Glop '07 | 3e4 | 70 | SPOTS | 3e6 | 105 | ? | ? |