Improving Depth-first PN-Search: $1 + \varepsilon$ Trick

Jakub Pawlewicz Łukasz Lew

Institute of Informatics
Warsaw University

Computer and Games 2006

Outline

Relevant Work.

1994 Allis et al: Proof-Number Search.

1998 Nagai: PDS – Proof Disproof Search.

2002 Nagai: DF-PN – Depth-first PN-Search.

2004 Winands et al: PDS-PN.

2005 Kishimoto et al: DF-PN with heuristic threshold increments.

Outline

PN-Search.

- PN-Search is AND/OR tree search algorithm.
- Uses proof and disproof numbers to find MPN.
- Iteratively expands Most Proving Node.

Recursive formula for the proof and disproof numbers

PN-Search.

- PN-Search is AND/OR tree search algorithm.
- Uses proof and disproof numbers to find MPN.
- Iteratively expands Most Proving Node.

Recursive formula for the proof and disproof numbers

PN-Search.

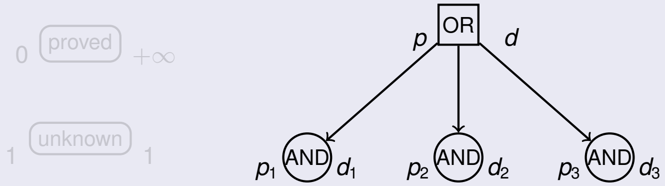
- PN-Search is AND/OR tree search algorithm.
- Uses proof and disproof numbers to find MPN.
- Iteratively expands Most Proving Node.

Recursive formula for the proof and disproof numbers

PN-Search.

- PN-Search is AND/OR tree search algorithm.
- Uses proof and disproof numbers to find MPN.
- Iteratively expands Most Proving Node.

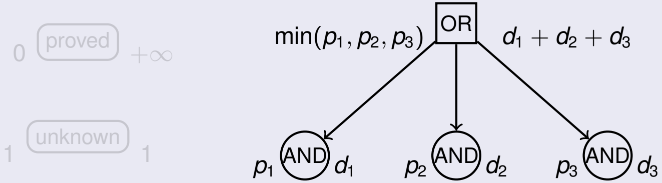
Recursive formula for the proof and disproof numbers



PN-Search.

- PN-Search is AND/OR tree search algorithm.
- Uses proof and disproof numbers to find MPN.
- Iteratively expands Most Proving Node.

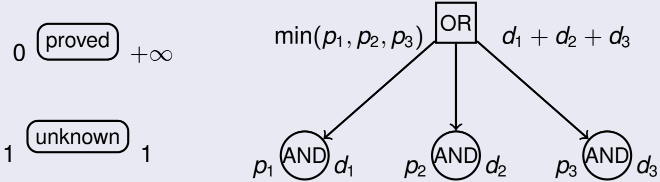
Recursive formula for the proof and disproof numbers



PN-Search.

- PN-Search is AND/OR tree search algorithm.
- Uses proof and disproof numbers to find MPN.
- Iteratively expands Most Proving Node.

Recursive formula for the proof and disproof numbers



DF-PN

- DF-PN is a PN-Search transformation to depth-first algorithm.
- Suspend updates as long as MPN is in current node's subtree.
- Uses proof and disproof number thresholds.
- Return condition: $p \geq pt \vee d \geq dt$.

Calculating the thresholds

$p_1 \leq p_2 \leq p_3$

DF-PN

- DF-PN is a PN-Search transformation to depth-first algorithm.
- Suspend updates as long as MPN is in current node's subtree.
- Uses proof and disproof number thresholds.
- Return condition: $p \geq pt \vee d \geq dt$.

Calculating the thresholds

$p_1 \leq p_2 \leq p_3$

DF-PN

- DF-PN is a PN-Search transformation to depth-first algorithm.
- Suspend updates as long as MPN is in current node's subtree.
- Uses proof and disproof number thresholds.
- Return condition: $p \geq pt \vee d \geq dt$.

Calculating the thresholds

$p_1 \leq p_2 \leq p_3$

DF-PN

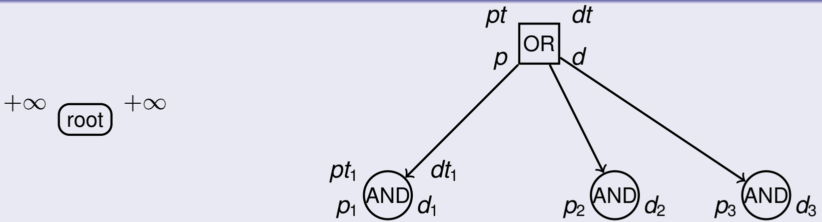
- DF-PN is a PN-Search transformation to depth-first algorithm.
- Suspend updates as long as MPN is in current node's subtree.
- Uses proof and disproof number thresholds.
- Return condition: $p \geq pt \vee d \geq dt$.

Calculating the thresholds

DF-PN

- DF-PN is a PN-Search transformation to depth-first algorithm.
- Suspend updates as long as MPN is in current node's subtree.
- Uses proof and disproof number thresholds.
- Return condition: $p \geq pt \vee d \geq dt$.

Calculating the thresholds

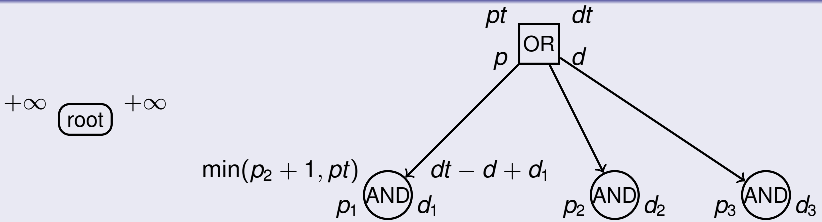


$p_1 \leq p_2 \leq p_3$

DF-PN

- DF-PN is a PN-Search transformation to depth-first algorithm.
- Suspend updates as long as MPN is in current node's subtree.
- Uses proof and disproof number thresholds.
- Return condition: $p \geq pt \vee d \geq dt$.

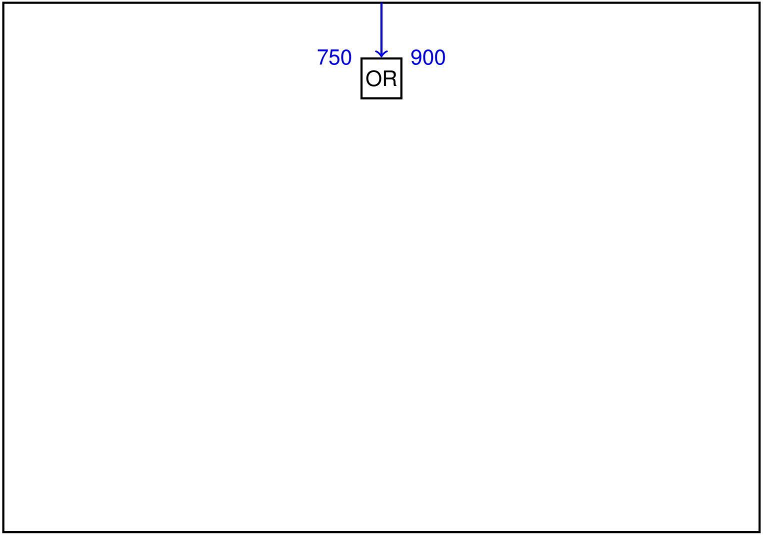
Calculating the thresholds



$p_1 \leq p_2 \leq p_3$

Outline

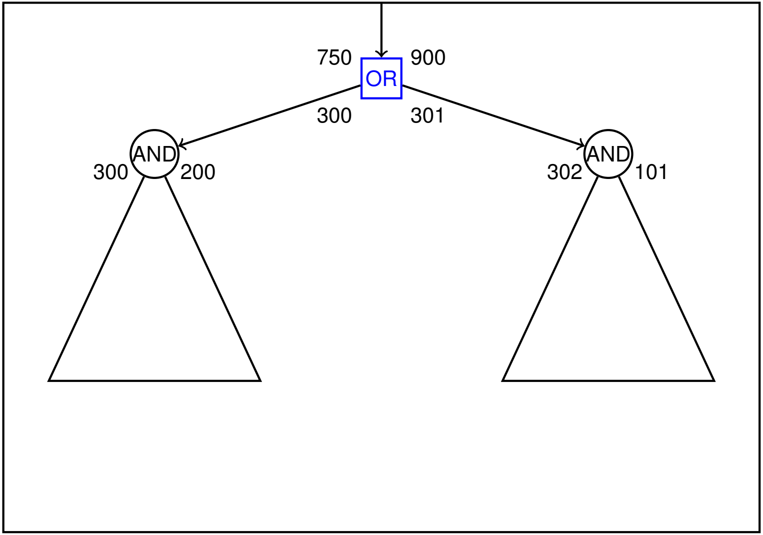
Typical Situation During a Run of DF-PN.



PN History

300
302
305
306
308
311
315
317
⋮
742
746
750
752

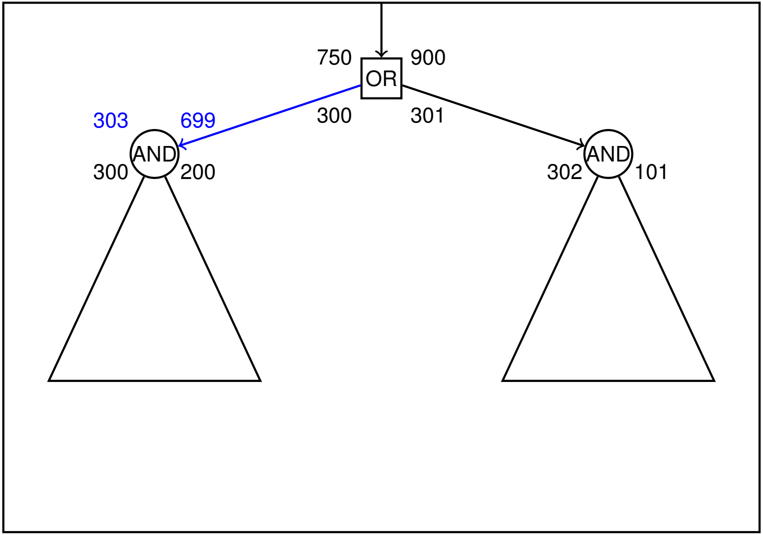
Typical Situation During a Run of DF-PN.



PN History

300
302
305
306
308
311
315
317
⋮
742
746
750
752

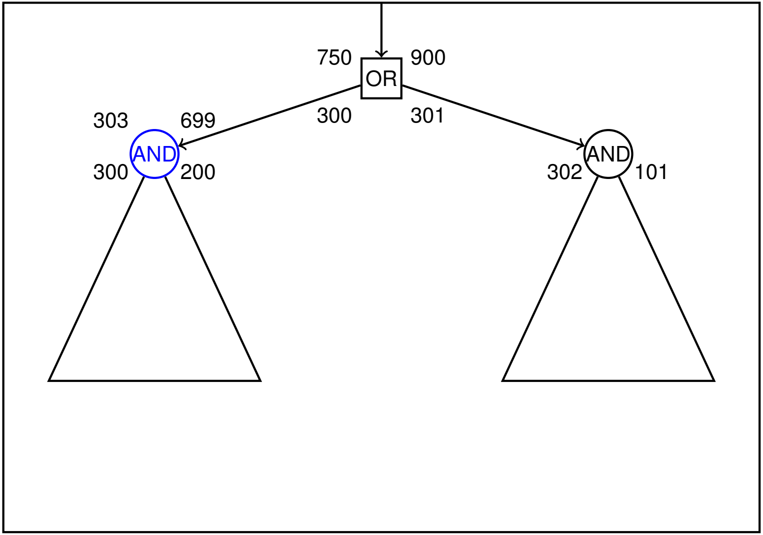
Typical Situation During a Run of DF-PN.



PN History

300
302
305
306
308
311
315
317
⋮
742
746
750
752

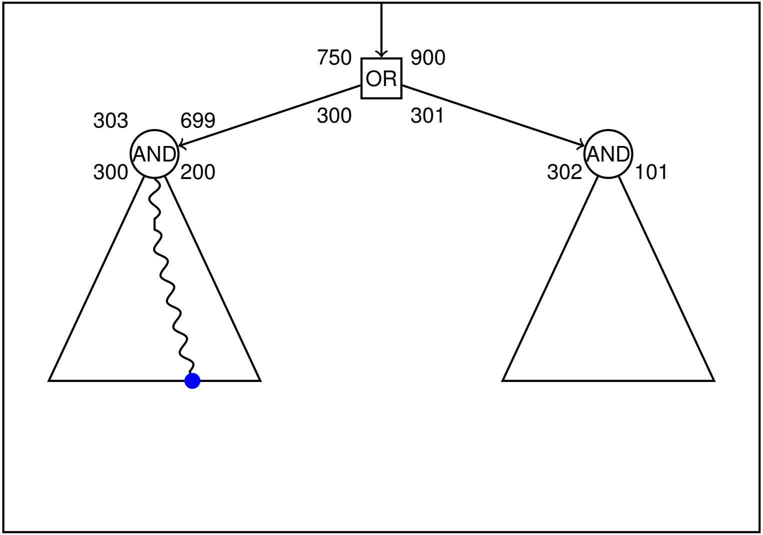
Typical Situation During a Run of DF-PN.



PN History

300
302
305
306
308
311
315
317
⋮
742
746
750
752

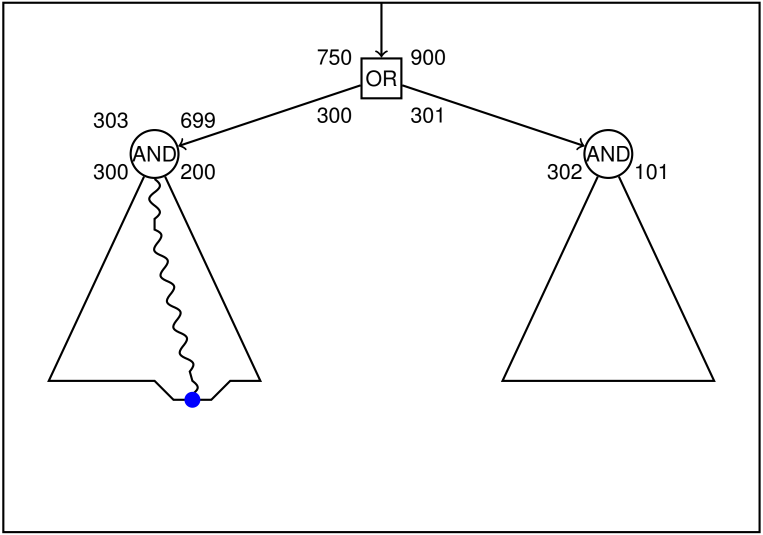
Typical Situation During a Run of DF-PN.



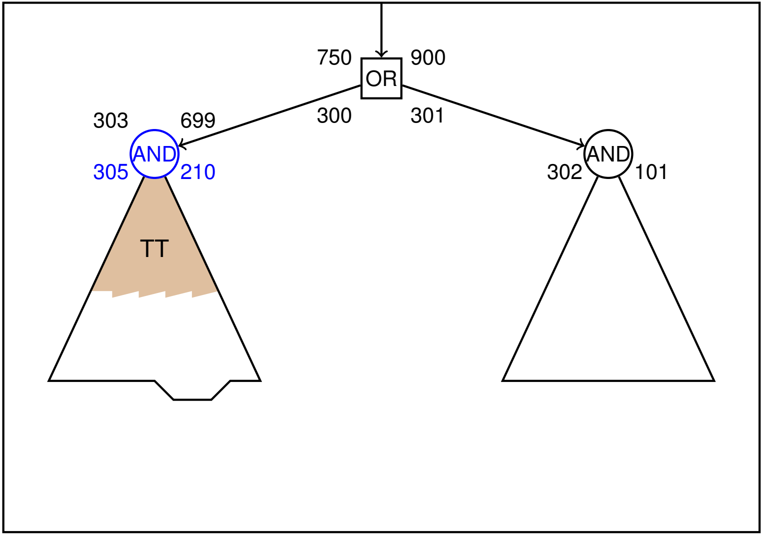
PN History

300, 302, 305, 306, 308, 311, 315, 317, ⋮, 742, 746, 750, 752

Typical Situation During a Run of DF-PN.



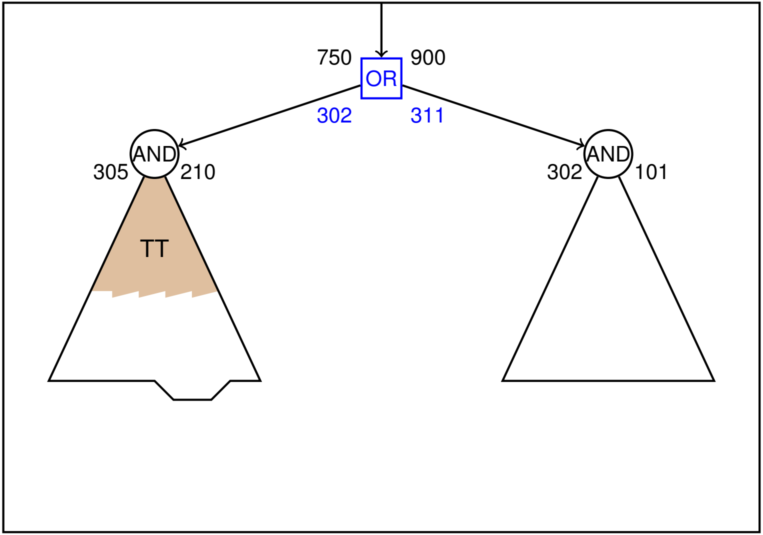
Typical Situation During a Run of DF-PN.



PN History

300 302 305 306 308 311 315 317 ⋮ 742 746 750 752

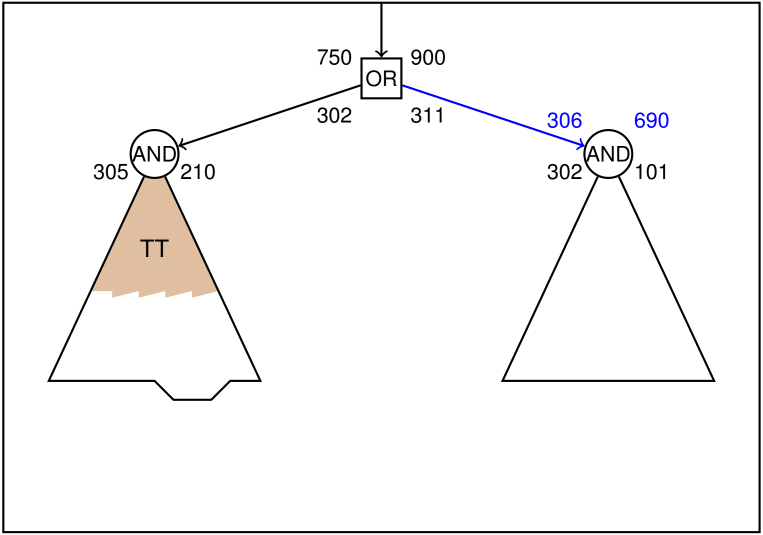
Typical Situation During a Run of DF-PN.



PN History

300	302
305	306
308	311
315	317
⋮	
742	746
750	752

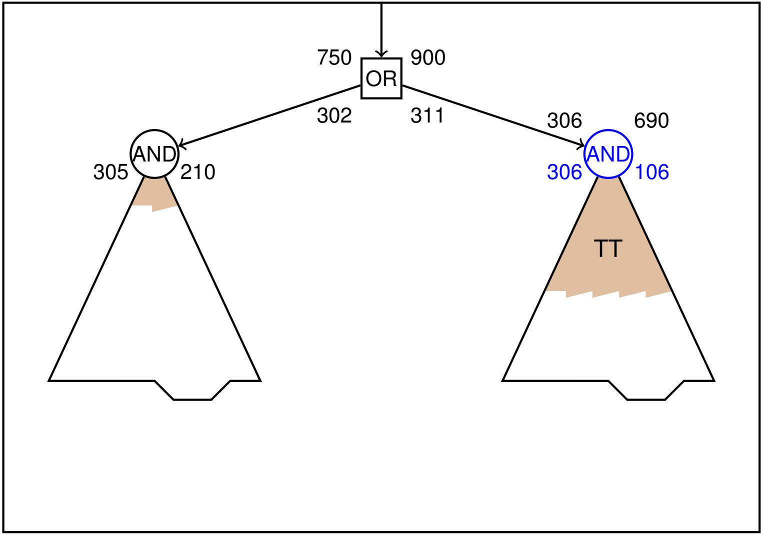
Typical Situation During a Run of DF-PN.



PN History

300
302
305
306
308
311
315
317
⋮
742
746
750
752

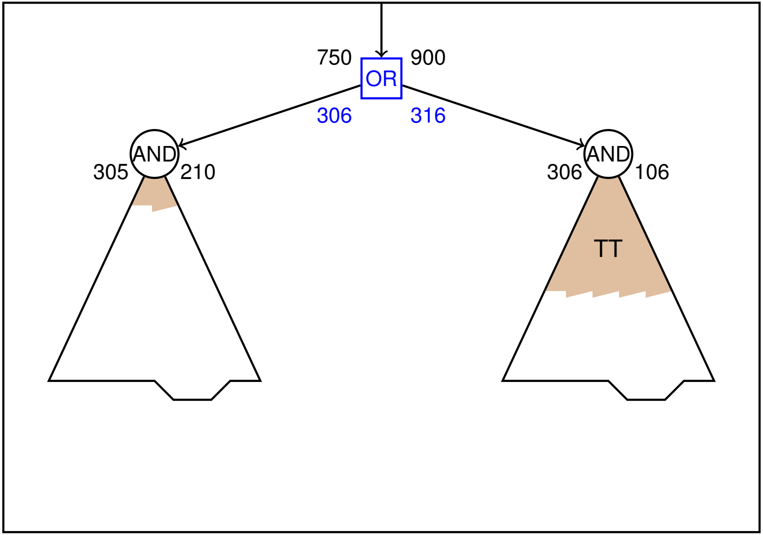
Typical Situation During a Run of DF-PN.



PN History

300
302
305
306
308
311
315
317
⋮
742
746
750
752

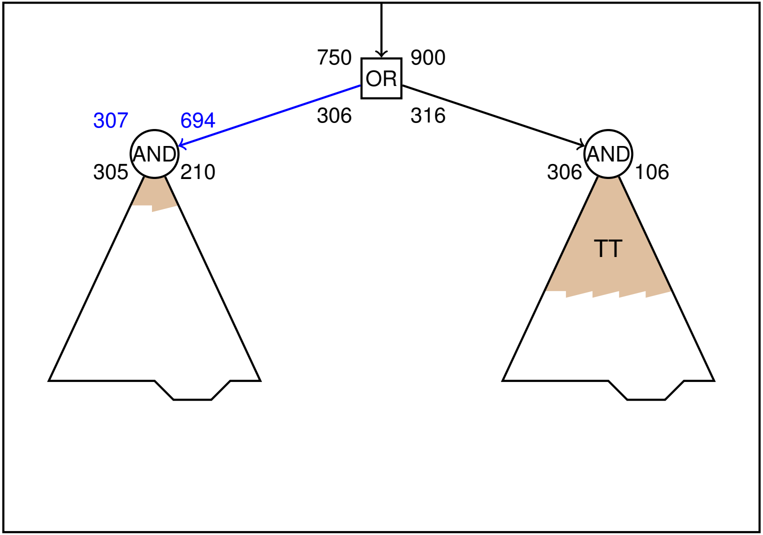
Typical Situation During a Run of DF-PN.



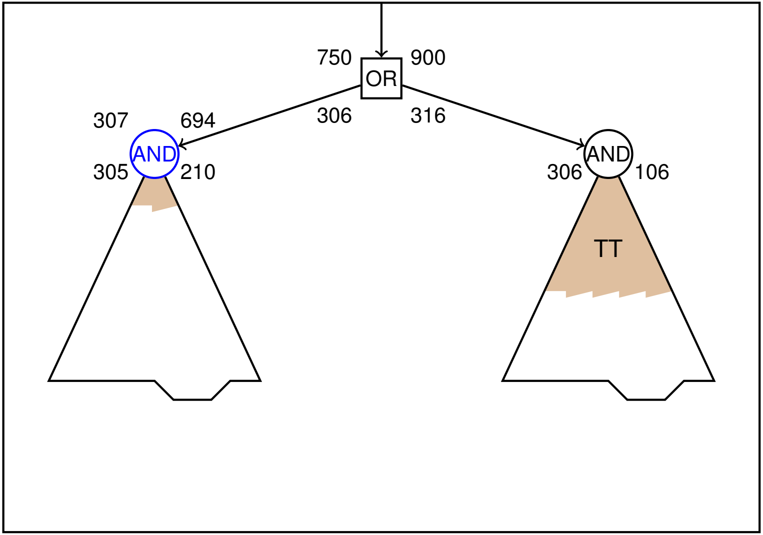
PN History

300 302 305 306 308 311 315 317 ⋮ 742 746 750 752

Typical Situation During a Run of DF-PN.



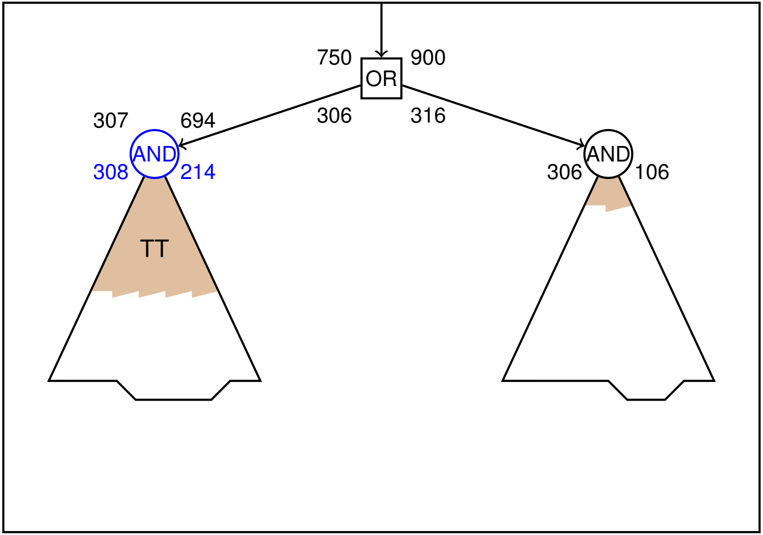
Typical Situation During a Run of DF-PN.



PN History

300 302 305 306 308 311 315 317 ⋮ 742 746 750 752

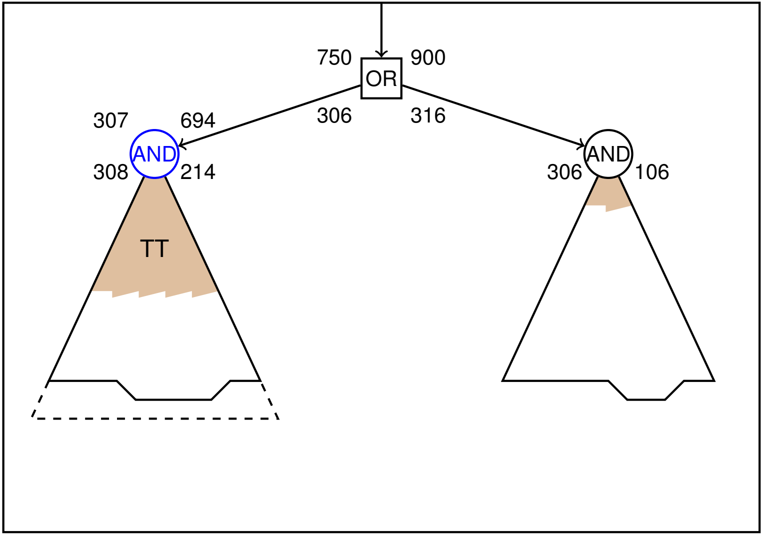
Typical Situation During a Run of DF-PN.



PN History

300
302
305
306
308
311
315
317
⋮
742
746
750
752

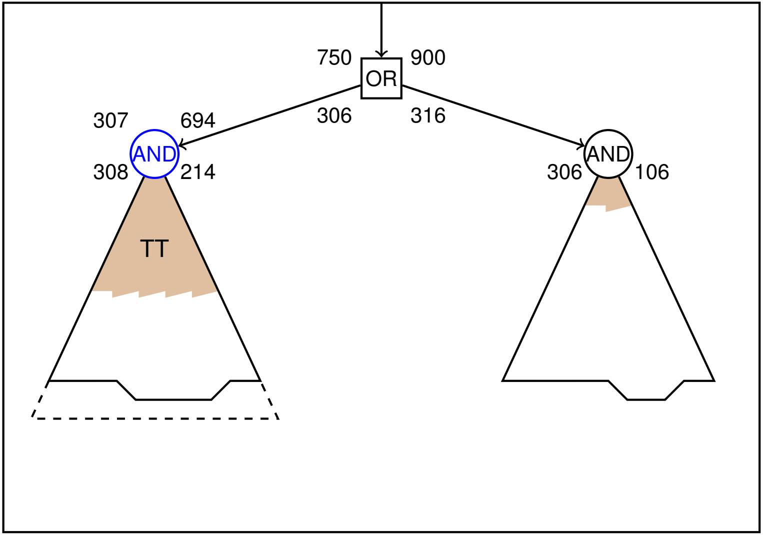
Typical Situation During a Run of DF-PN.



PN History

300
302
305
306
308
311
315
317
⋮
742
746
750
752

Typical Situation During a Run of DF-PN.

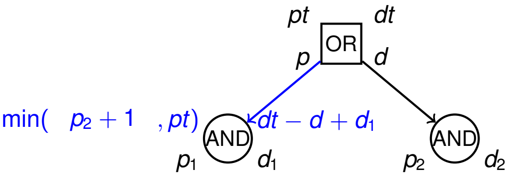


PN History

300 302
305 306
308 311
315 317
⋮
742 746
750 752

Outline

$1 + \varepsilon$ Trick.



PN History

300
302
379
476
597
750
752

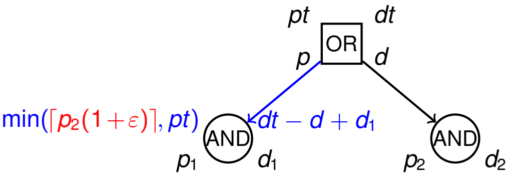
Advantages

- Reduced number of recursive calls and transposition table refills.
- Less tree traversion.

Disadvantages

- Leaves are not expanded in the same order as in PN-Search.
- For bigger ε the algorithm may spent too much time in inessential part of a tree.

$1 + \varepsilon$ Trick.



PN History

300
302
379
476
597
750
752

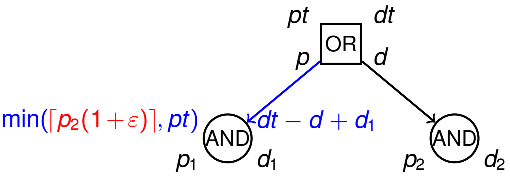
Advantages

- Reduced number of recursive calls and transposition table refills.
- Less tree traversion.

Disadvantages

- Leaves are not expanded in the same order as in PN-Search.
- For bigger ε the algorithm may spent too much time in inessential part of a tree.

$1 + \varepsilon$ Trick.



PN History

300
302
379
476
597
750
752

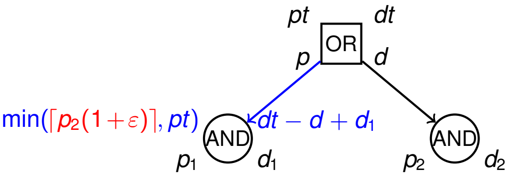
Advantages

- Reduced number of recursive calls and transposition table refills.
- Less tree traversion.

Disadvantages

- Leaves are not expanded in the same order as in PN-Search.
- For bigger ε the algorithm may spent too much time in inessential part of a tree.

$1 + \varepsilon$ Trick.



PN History

300	302
379	476
597	750
752	

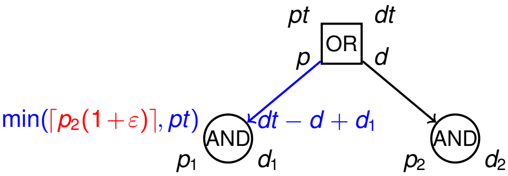
Advantages

- Reduced number of recursive calls and transposition table refills.
- Less tree traversion.

Disadvantages

- Leaves are not expanded in the same order as in PN-Search.
- For bigger ε the algorithm may spent too much time in inessential part of a tree.

$1 + \varepsilon$ Trick.



PN History

300
302
379
476
597
750
752

Advantages

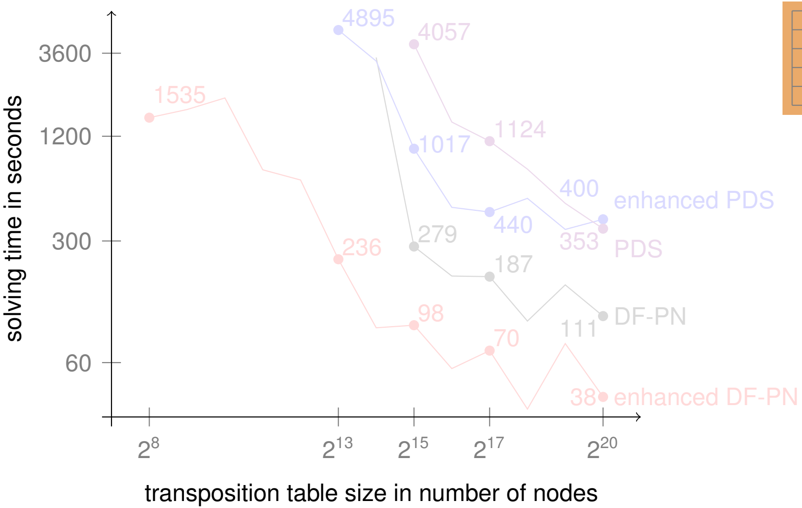
- Reduced number of recursive calls and transposition table refills.
- Less tree traversion.

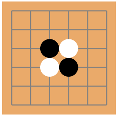
Disadvantages

- Leaves are not expanded in the same order as in PN-Search.
- For bigger ε the algorithm may spent too much time in inessential part of a tree.

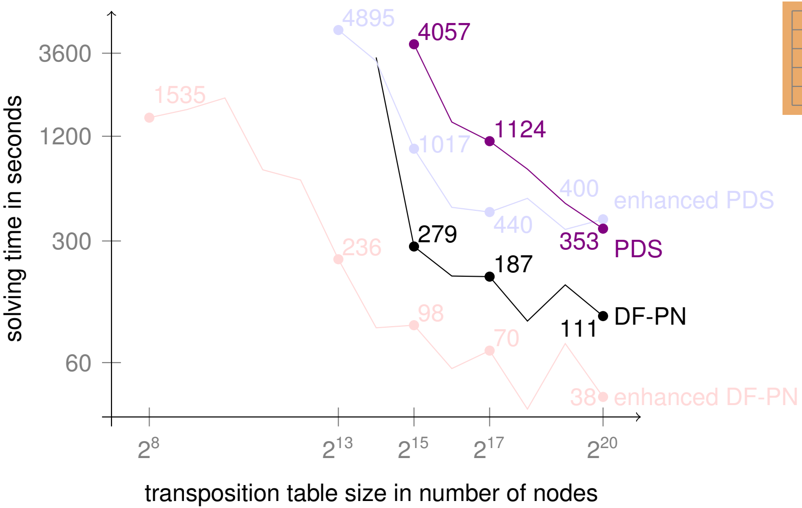
Outline

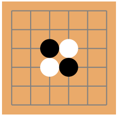
Atari Go: The Size of a Transposition Table



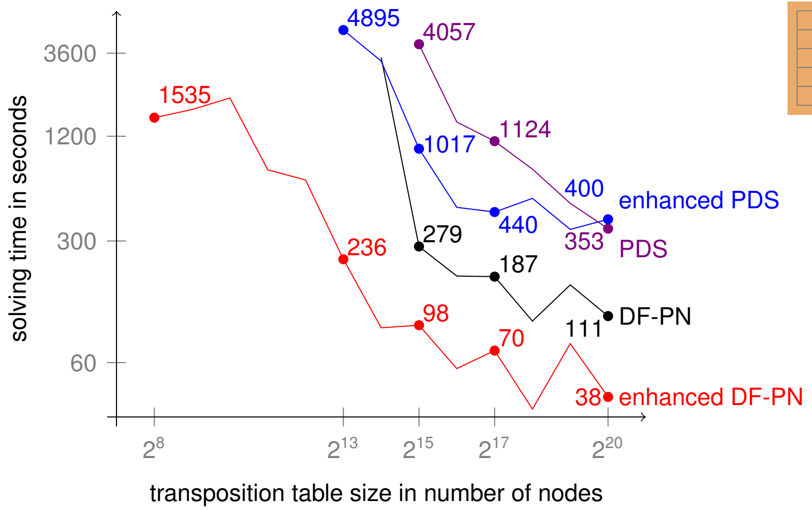


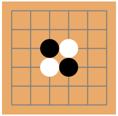
Atari Go: The Size of a Transposition Table



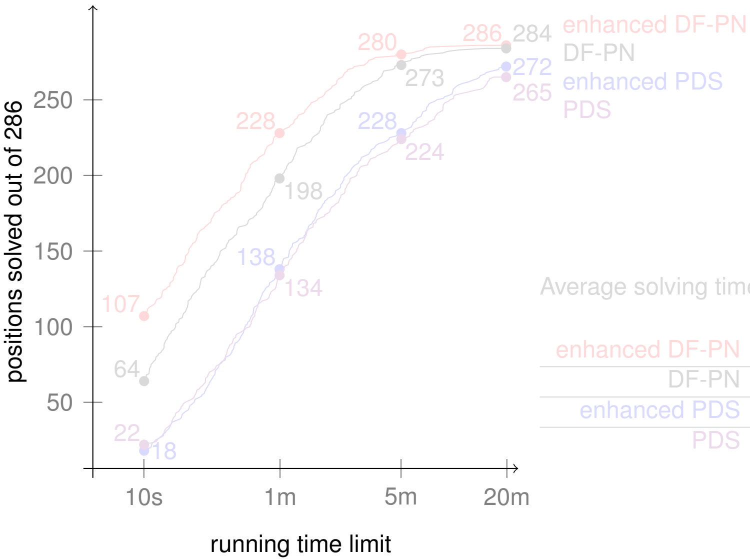


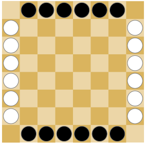
Atari Go: The Size of a Transposition Table





LOA: Efficiency of Solving Hard Problems

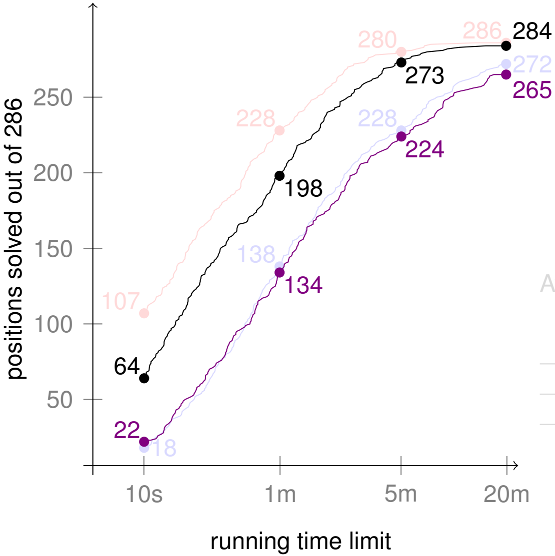


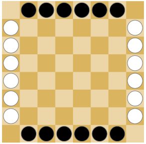


Average solving times in seconds

enhanced DF-PN	45
DF-PN	80
enhanced PDS	241
PDS	328

LOA: Efficiency of Solving Hard Problems

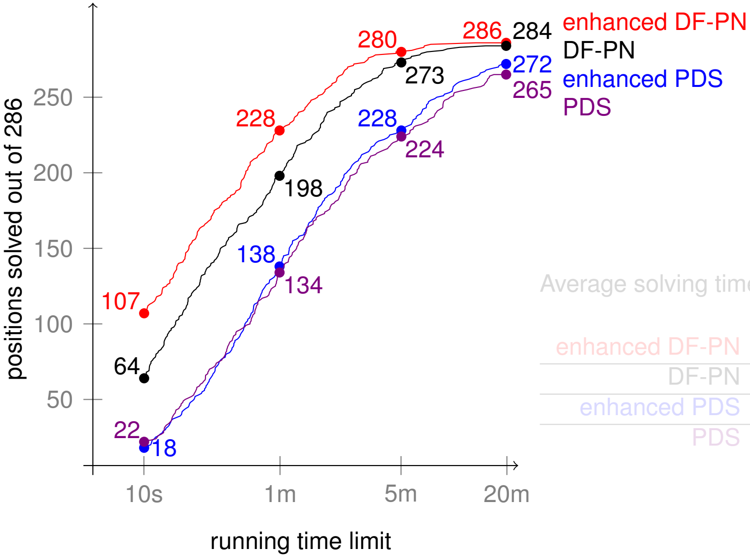


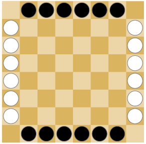


Average solving times in seconds

enhanced DF-PN	45
DF-PN	80
enhanced PDS	241
PDS	328

LOA: Efficiency of Solving Hard Problems

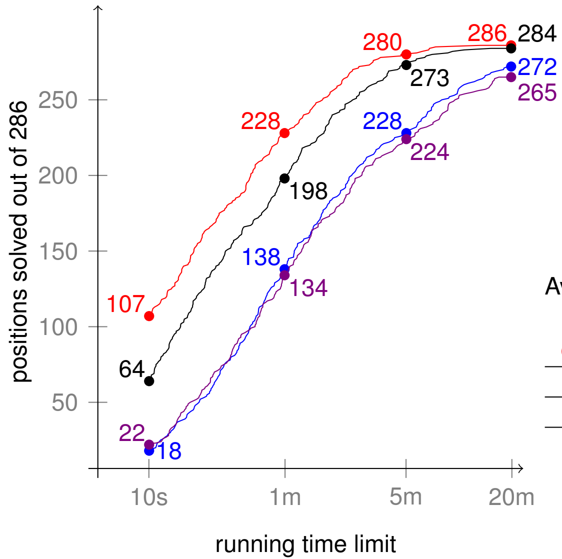


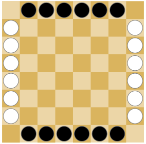


Average solving times in seconds

enhanced DF-PN	45
DF-PN	80
enhanced PDS	241
PDS	328

LOA: Efficiency of Solving Hard Problems





Average solving times in seconds

enhanced DF-PN	45
DF-PN	80
enhanced PDS	241
PDS	328

Summary

- We have pointed out the problems in DF-PN.
- We have introduced $1 + \varepsilon$ trick to enhance DF-PN.
- Atari Go experiment has shown that enhanced methods outperform their plain variants in low memory conditions.
- Experiment on LOA has shown that the trick is also valuable for solving hard problems.

Thank You

Questions?