一般化将棋・チェス・囲碁問題の定数時間アルゴリズム

ITO Hiro (伊藤大雄)
The University of Electro-Communications (UEC), Tokyo, Japan.

Joint work with
NAGAO Atsuki (長尾篤樹) and PARK Teagun (朴 台根).

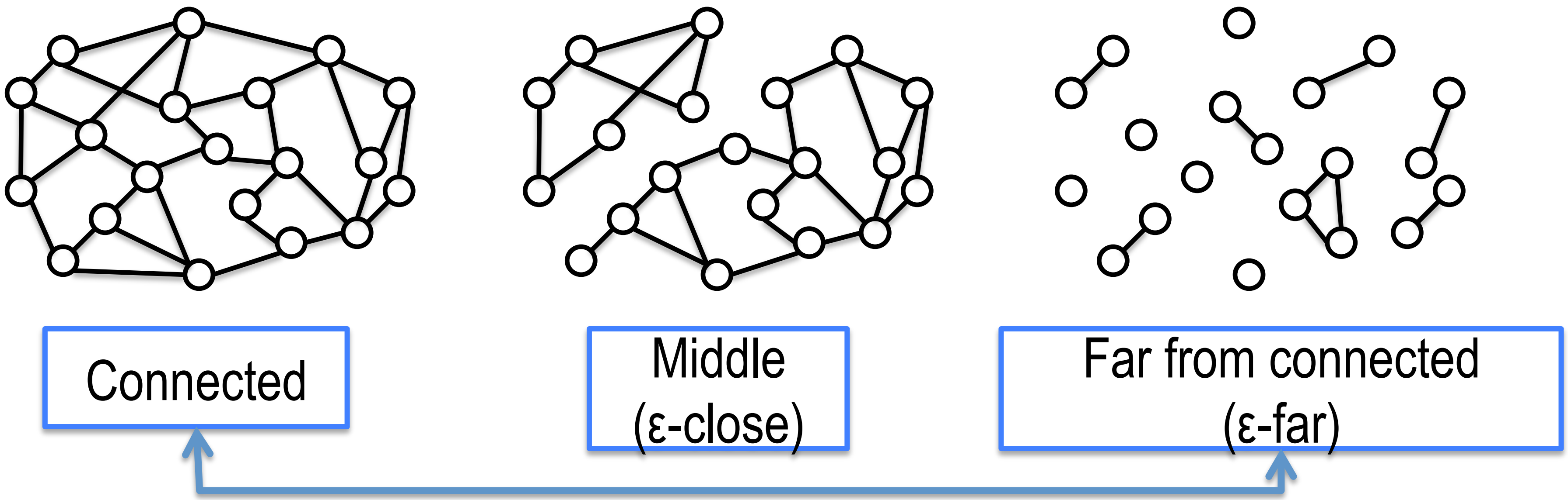
What is a constant-time algorithm?

- Common sense (to date): Every algorithm must read whole of the input, i.e., computation time = $\Omega(n)$.
- However, can we do it by reading a very small part (constant-size) of the input?

- Reading only O(1) part of the input:
  - **Theoretical** assurance
  - Nice to treat **big data**, e.g. web-graphs, genom, etc.

Property Testing (the most well-studied framework in the area of const.-time algs.)

- Concept 1: Approximation



Distinguish dist. =0 and > ε （0<∀ε<1）

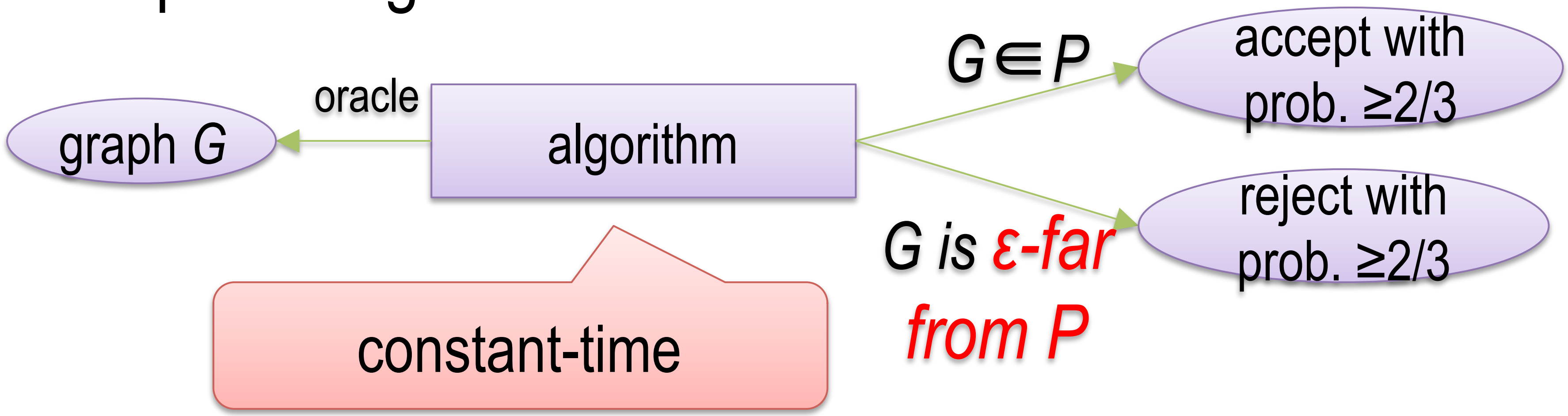
- Concept 2: Probabilistic
  - (For any input) output the correct answer with prob. ≥ 2/3.

Decision Alg. and Property Testing

$\Omega(n)$ -time

- (Traditional) Decision Alg.

- Prop. Testing



Important Previous Results on Const.-Time Algs.

- Dense-graph model:
  - Every hereditary property is testable. [Alon et al. FOCS05]
  - A necessary and sufficient condition of testability. [Alon et al., STOC06]

- Bounded-degree model:
  - Every minor-closed property is testable. [Benjamini et al. STOC08]
  - For hyperfinite graphs, every property is testable. [Newman & Sohler, STOC11]

- General graph model:
  - For a a class of hierarchically scale-free multi-graphs, every property is testable. [Ito, 15]

Important Previous Results on Const.-Time Algs.

- Dense-graph model:
  - Every hereditary property is testable. [Alon et al. FOCS05]
  - A necessary and sufficient condition of testability. [Alon et al., STOC06]
- Bounded-degree model:
  - Every minor-closed property is testable. [Benjamini et al. STOC08]
  - For hyperfinite graphs, every property is testable. [Newman & Sohler, STOC11]
- General graph model:
  - For a a class of hierarchically scale-free multi-graphs, every property is testable. [Ito, 15]

If G has the property, then any its subgraph also has the property.

If G has the property, then any its minor also has the property.

A graph property ⇔ a (possibly infinite) subset of graphs

By removing small # of edges, sizes of every connected component is bounded by a constant.

1. Power-law degree distribution
2. Including isolated cliques
3. If these cliques are contacted, the resulting multigraph has the same property.

Generalized Chess-Type Games

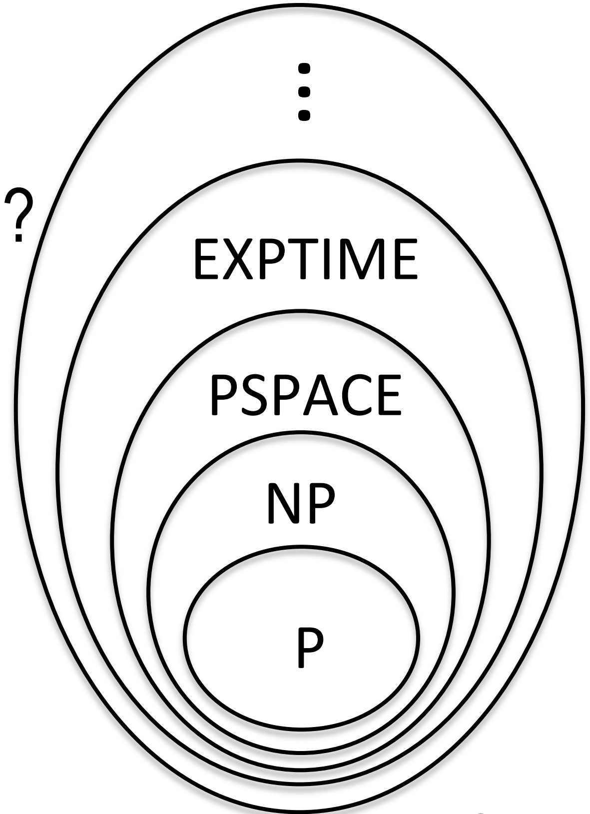
- Use a $\sqrt{n}$ x $\sqrt{n}$ board (# of cells is n) and O(n) pieces (only one king for each player).
- **Input (Instance):** a position: defined by fixing for each piece, the owner of it and the place (cell) of it.
- **Objective:** For a given position, we decide whether Alice (the player who moves first) win or not.

Known Results on Computational Complexity of Generalized Chess-Type Games

- The generalized chess [Fraenkel and Lichtenstein 81] and shogi (Japanese chess) [Adachi, et al. 87] are **EXPTIME-complete**.

- How about constant-time testability of them?

Note: Many problems known to be constant-time testable are in NP

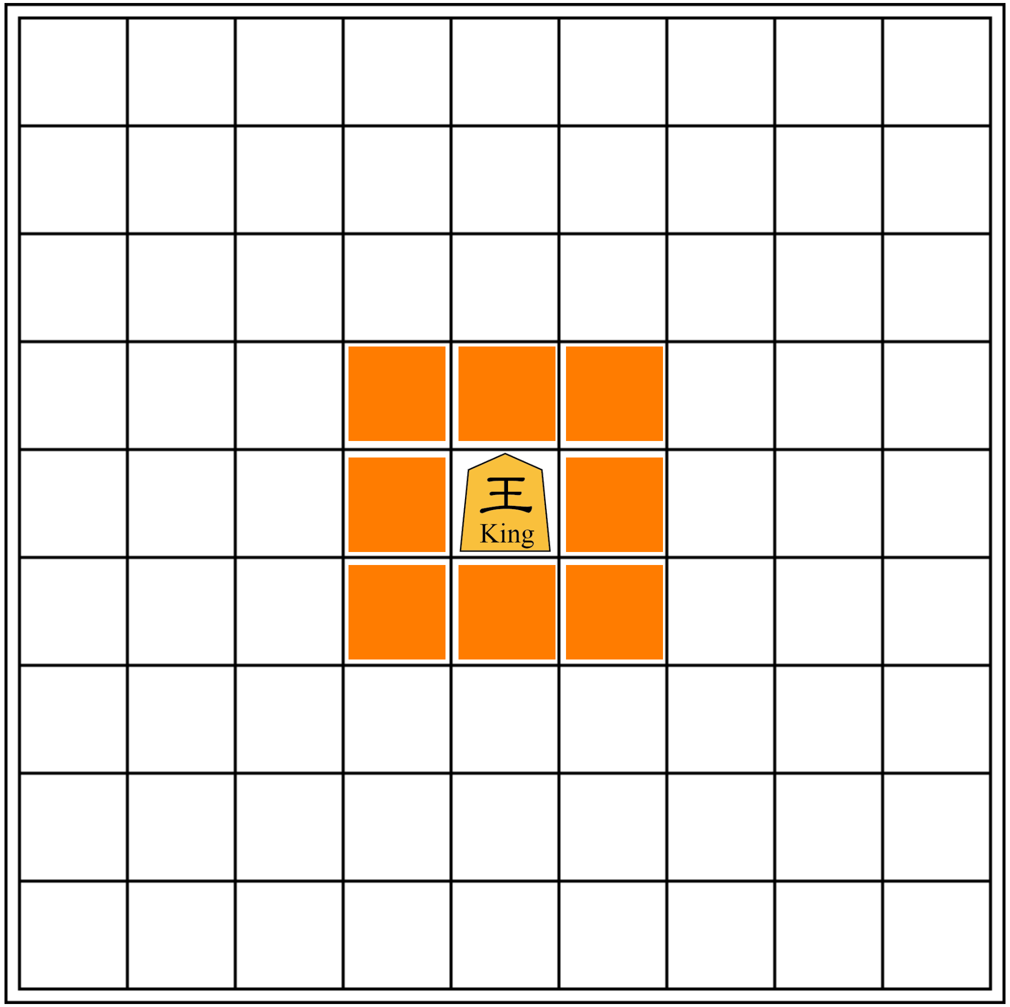


The Rules of Shogi (将棋)

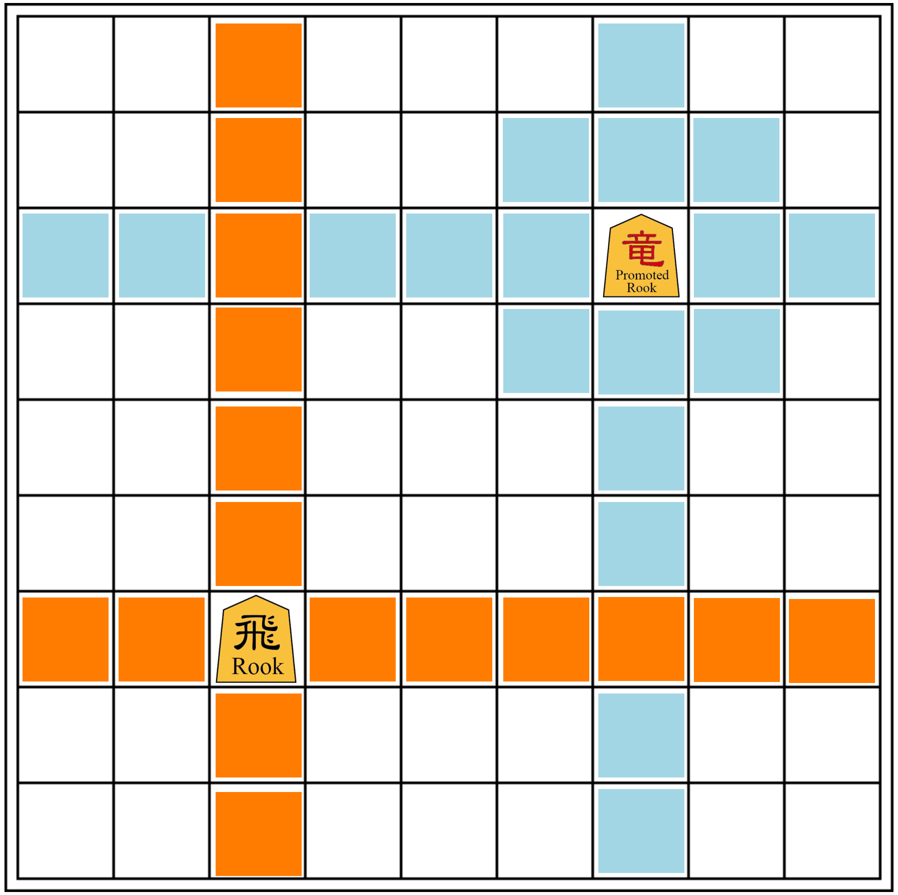
- 9x9 board
- 8 kind of pieces: king (王), rook (飛), bishop (角), gold (金), silver (銀), knight (桂), lance (香), pawn (歩)
- Every piece captured becomes the capturing players' piece, i.e., it can be placed on the board in his/ her turn.
- Each camp (consisting of the first three rows) is opponent's promotion area, i.e., if a piece enters the enemy's camp, it can be promoted

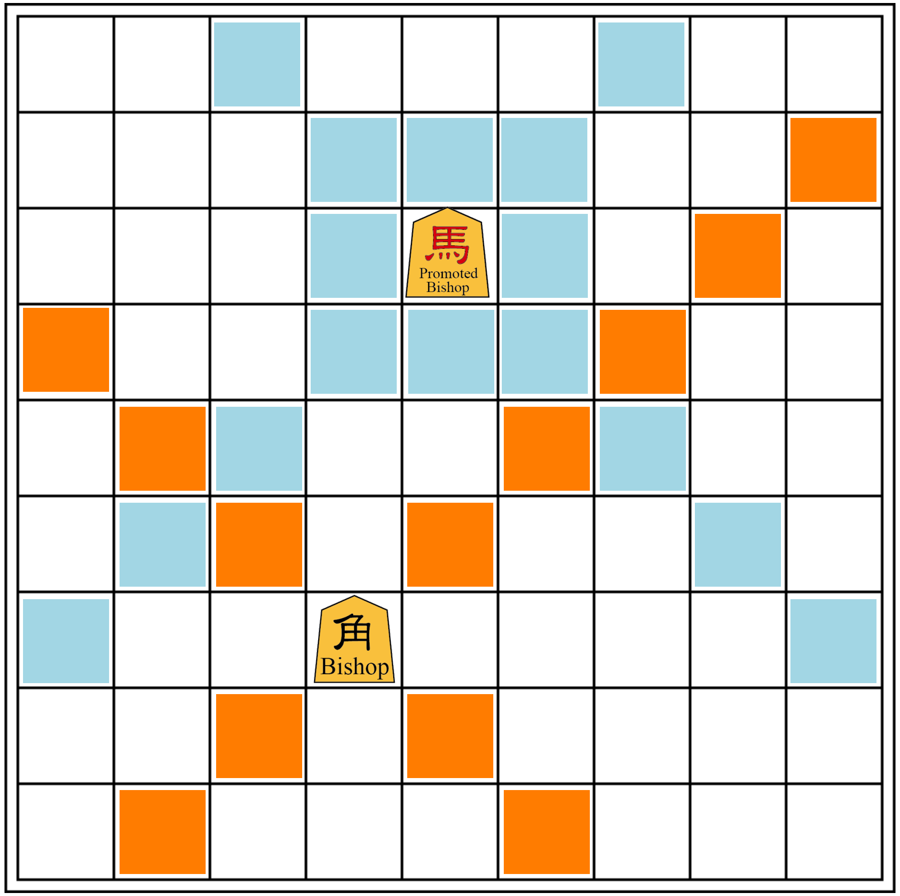
Movement of King (王)

- A player who is captured his/her king loses.
- King can't be promoted.

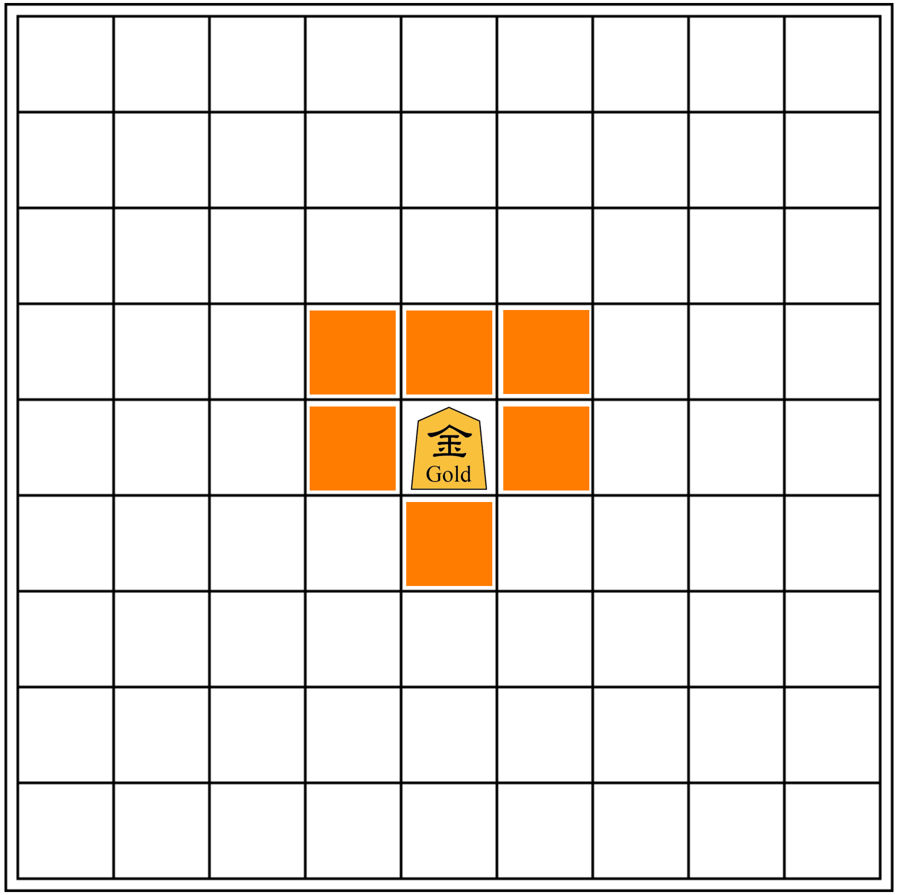


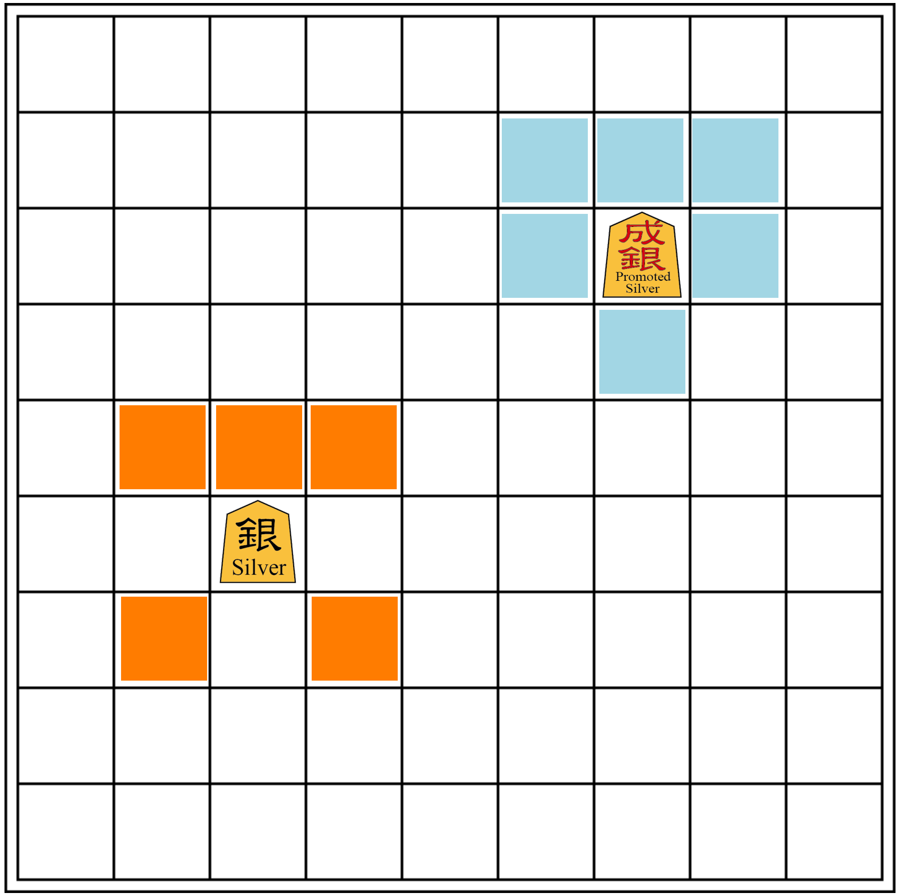
Movement of Rook (飛) and Bishop (角)





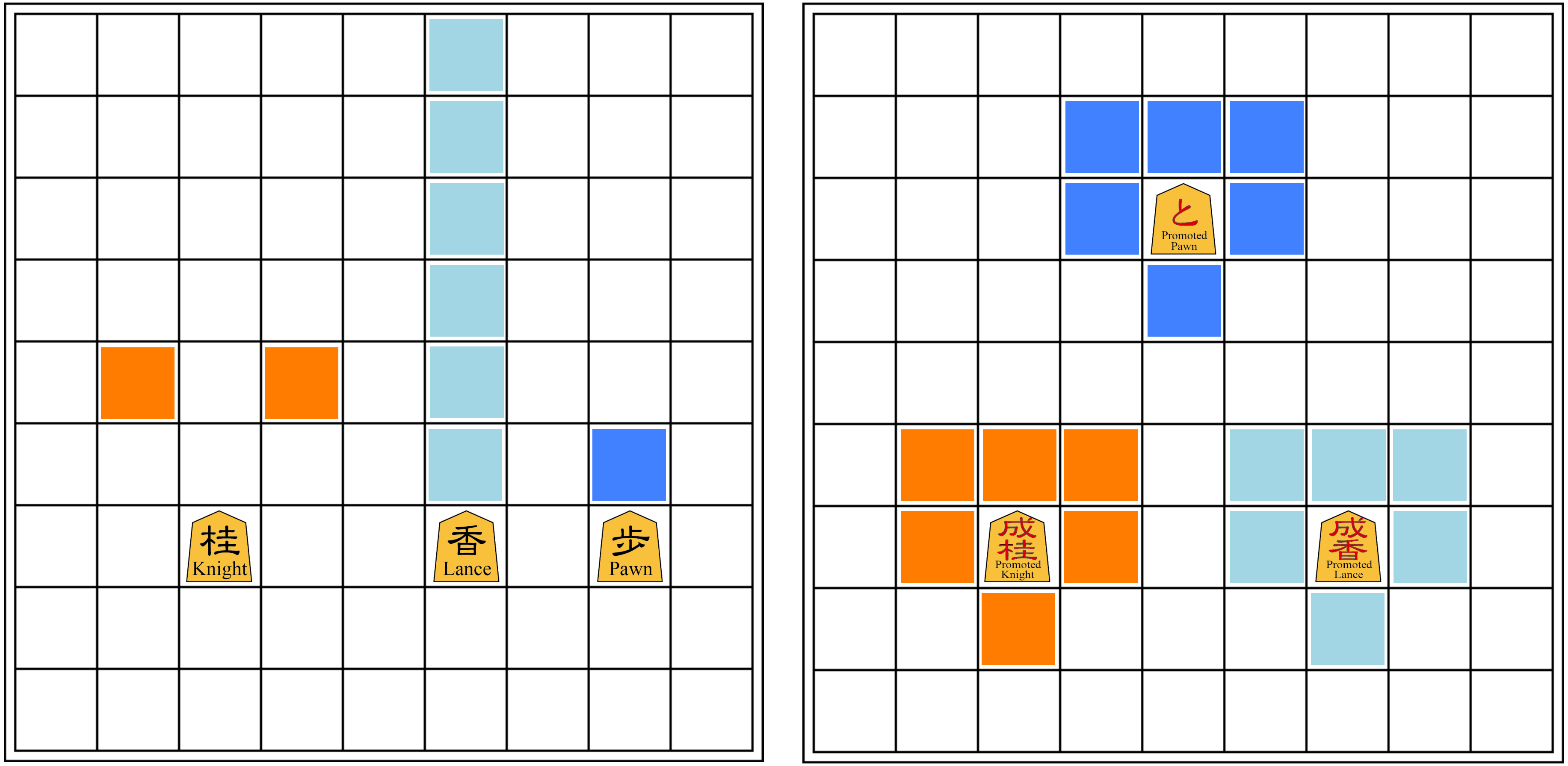
Movement of Gold (金) and Silver (銀)





- Gold is not promoted. Silver is promoted to gold.

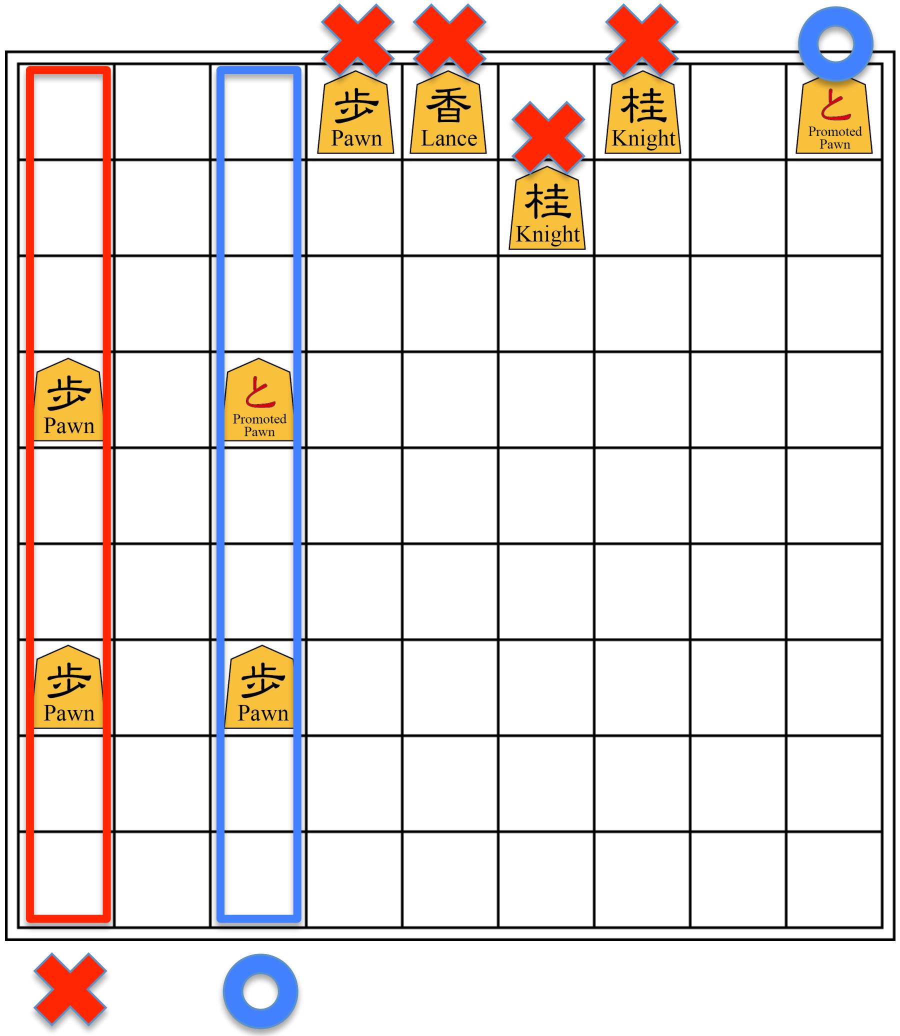
Movement of Knight (桂), Lance (香), and Pawn (歩)



- Knight, lance, and pawn are promoted to gold.

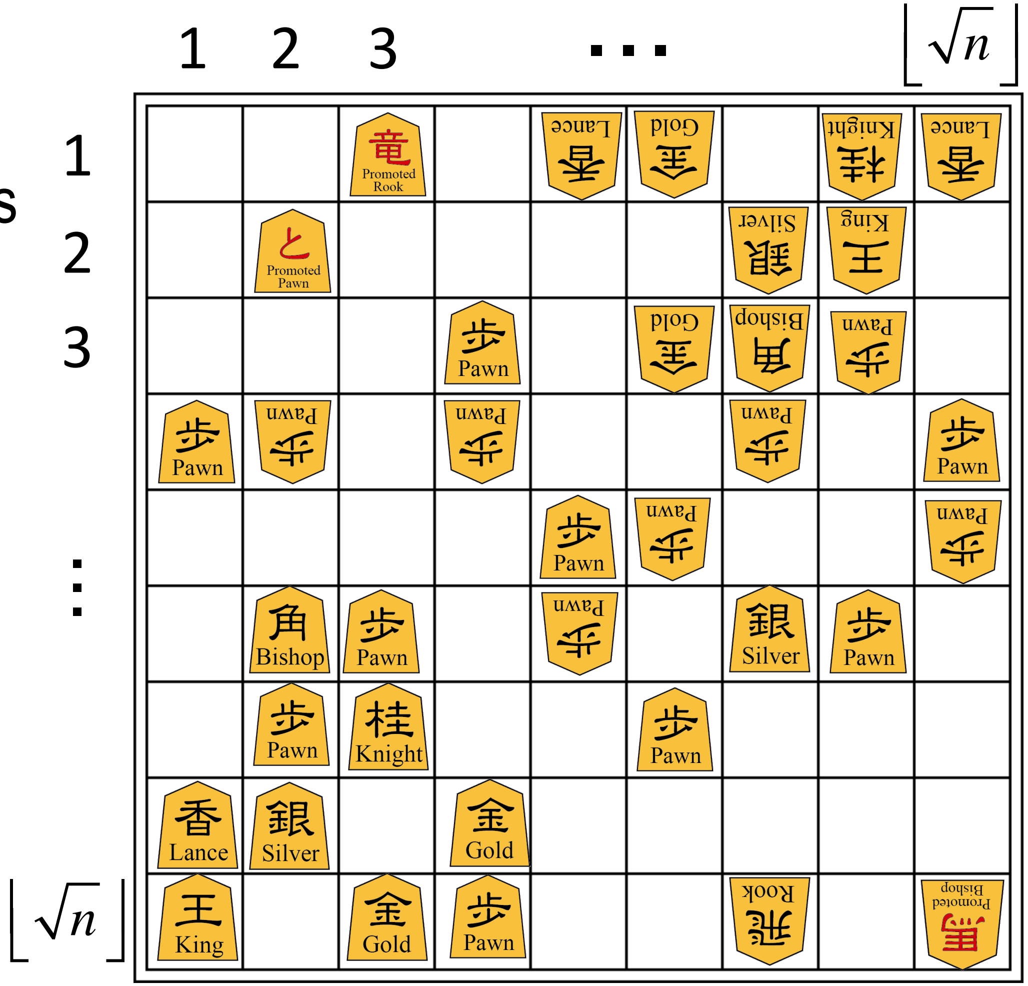
Fouls

- Nifu (二歩, double pawn): Two or more unpromoted pawns of a same player never be the same column simultaneously.
- Dead end: Pawns, lances, and nights never be to moved to, or dropped onto cells from which they have no next moves.



Generalized Shogi

- Use $\lfloor \sqrt{n} \rfloor \times \lfloor \sqrt{n} \rfloor$ board.
- Use two kings and $\lfloor cn \rfloor$ pieces for any other piece-kind (i.e., rook, bishop, ... , pawn).
- A position is called a winner if Alice has a winning strategy.
- **Generalized Shogi Problem: For any given position (input), we decide whether it is a winner or not.**



Piece ID, etc.

- Piece-kind-numbers:
  - King: 0, Rook: 1, Bishop: 2, Gold: 3, Silver: 4, Knight: 5, Lance: 6, Pawn: 7.
- Each piece is identified by its own ID (k,L),
  - $k \in \{0,\ldots,7\}$: piece-kind-number,
  - $L \in \{1,\ldots,\lfloor cn \rfloor\}$: piece-number.

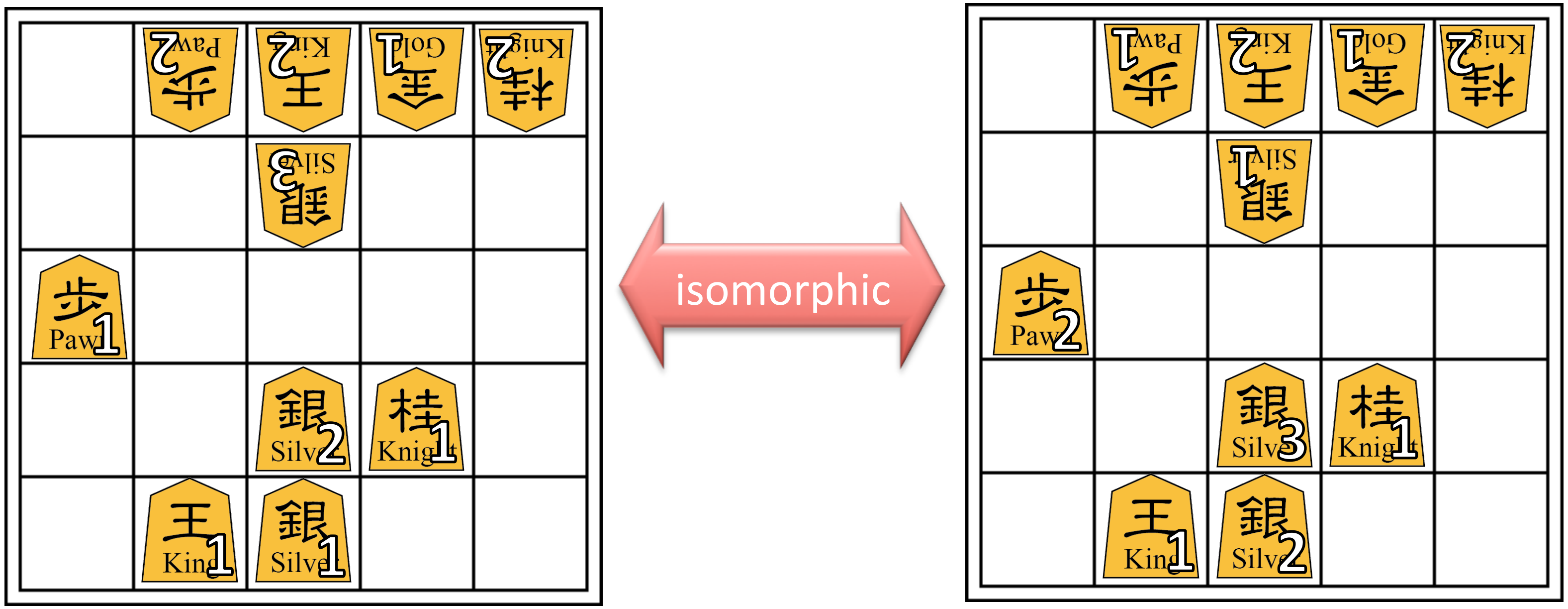
- Assume that $c \leq 1/8$, since $7cn+2 \leq n$ and thus all pieces can be arranged on the board simultaneously.

Oracle: An algorithm knows the given position S through oracles

- Piece oracle: $q_1(k,L;S)=(p,i,j,r)$:
  - A given piece ID (k,L), the oracle answers (p,i,j,r):
    - $p \in \{0,1,2\}$ means the owner (1: Alice, 2: Bob, 0: not used),
    - (i,j) means the coordinate of the cell (if i=0, it's a captured piece),
    - $r \in \{0,1\}$ means promoted or not (1: promoted, 0: otherwise).
- Position oracle A: $q_2(i,j;S)=(p,k,L,r)$:
  - A given coordinate (i,j), the oracle answers (p,k,L,r), which shows the information of the piece being in the cell:
    - p means the owner, (k,L) means the piece ID, and r means promoted or not.
- Position oracle B: $q'_2(p,k;S)=L$:
  - Player p is capturing L pieces of piece-number k,
  - e.g., q'(1,5)=48 represents that Alice is capturing 48 knights.
- S can be omitted if it is clear.
- A position is fixed by fixing piece oracle for all k and L (or position oracle for all (i,j) and (p,k)).

Position Isomorphism

- Σ : the set of all positions.
- Two positions S, S' ∈ Σ are isomorphic if we can make S the same with S' by changing only piece-numbers. (Note: changing piece-kind-numbers is not allowed.)



What are Properties?

- A set of positions that is closed under isomorphism is called a **property**.
  - (i.e., If Π is a property and S∈Π, then any S' that is isomorphic to S is also in Π.)
- Let $\mathcal{W} \subset \Sigma$ be the set of winners.
- Note: $\mathcal{W}$ is a property.

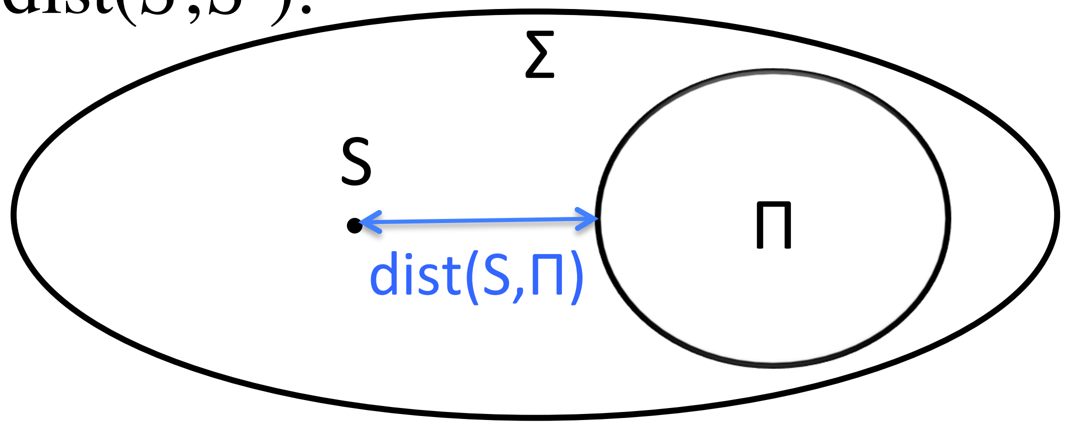
Distance

- The distance between two positions S and S' is defined as the number of pieces (k,L) such that the answers for $q_1(k,L;S)$ and $q_1(k,L;S')$ are different, i.e.,

$$\mathrm{dist}(S,S') := \frac{\left|\left\{(k,L) \mid q_1(k,L;S) \neq q_1(k,L;S')\right\}\right|}{n}.$$

- The distance between a position S and a property Π is

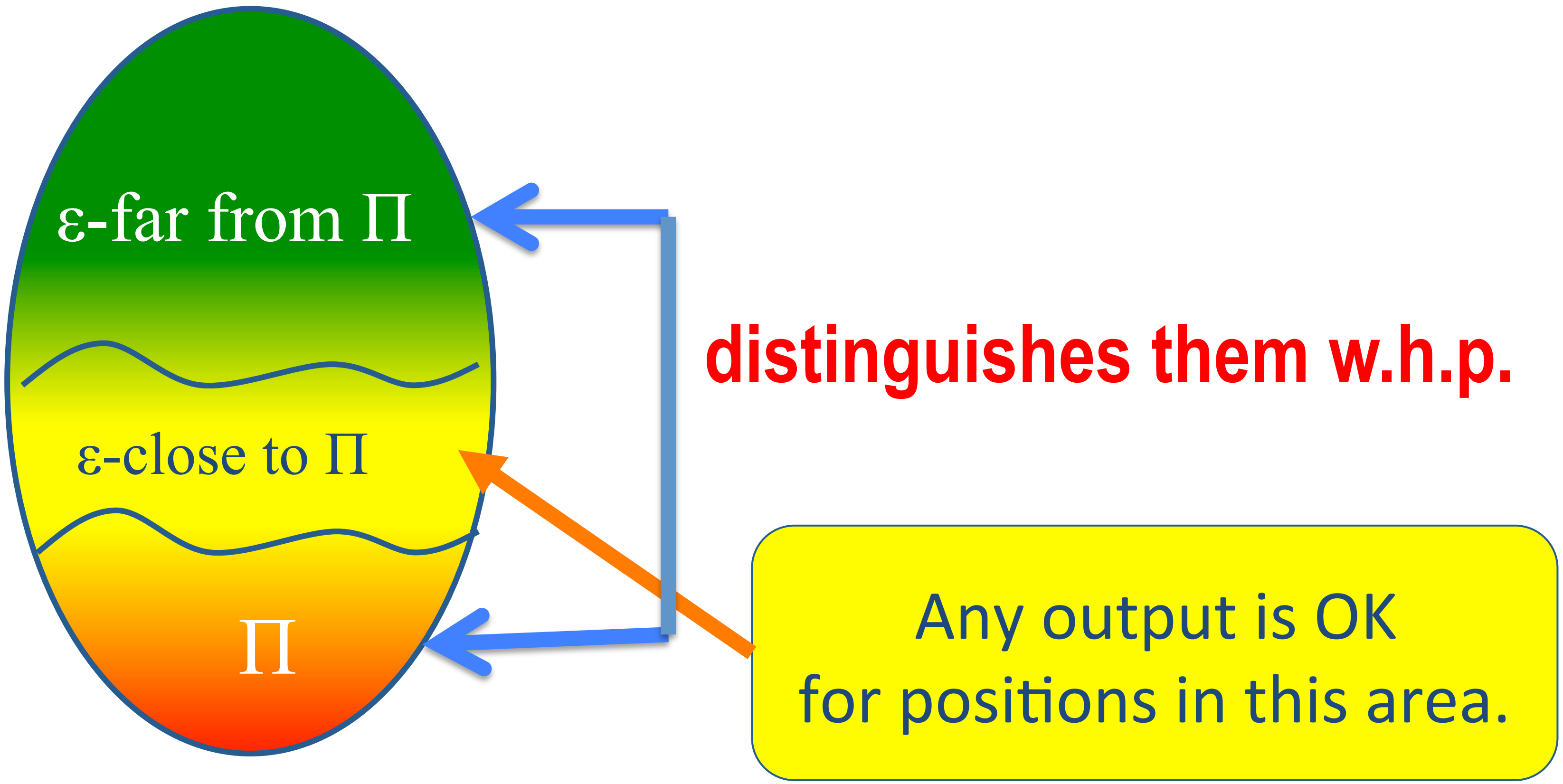
$$\mathrm{dist}(S,\Pi) := \min_{S' \in \Pi} \mathrm{dist}(S,S').$$



Tester for a property Π

- If dist(S,Π)>ε, then S is ε-far from Π.
- If dist(S,Π)≤ε, then S is ε-close to Π.
- The number of calling oracles of an algorithm A is the query complexity of A.
- A tester for a property Π is an algorithms (for any constant 0<ε≤1) that
  - accepts ∀S∈Π with probability ≥2/3 (*), and
  - rejects ∀S that is ε-far from Π with probability ≥2/3 (**)
  - with a constant (that can depend on ε) query complexity.
- If * is 1, it is called "one-sided-error."
- If * and ** are 1, it is called "no-error."

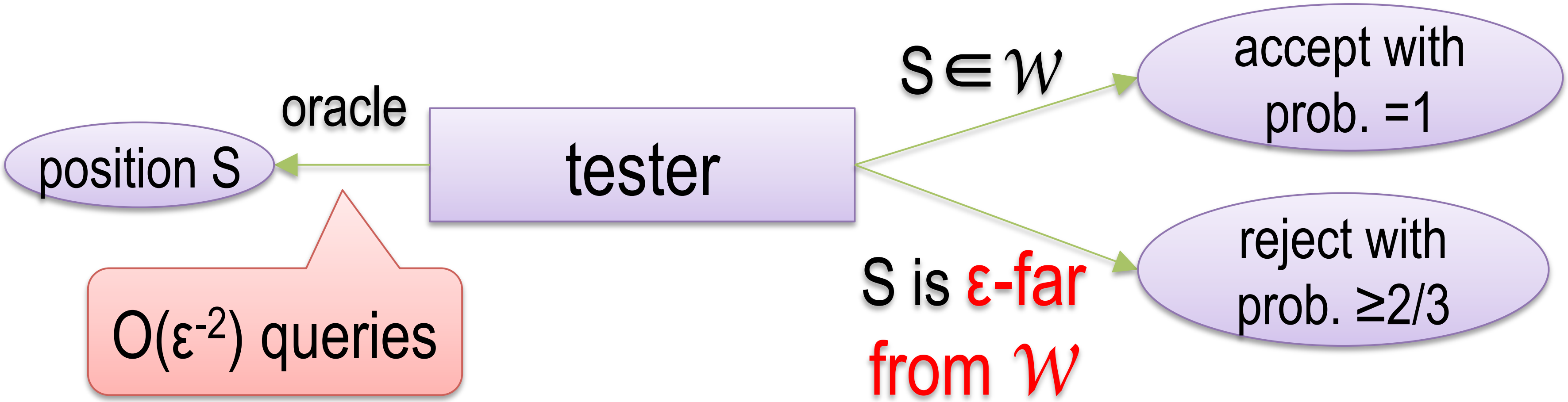
The Role of a Tester



Σ: the set of all positions

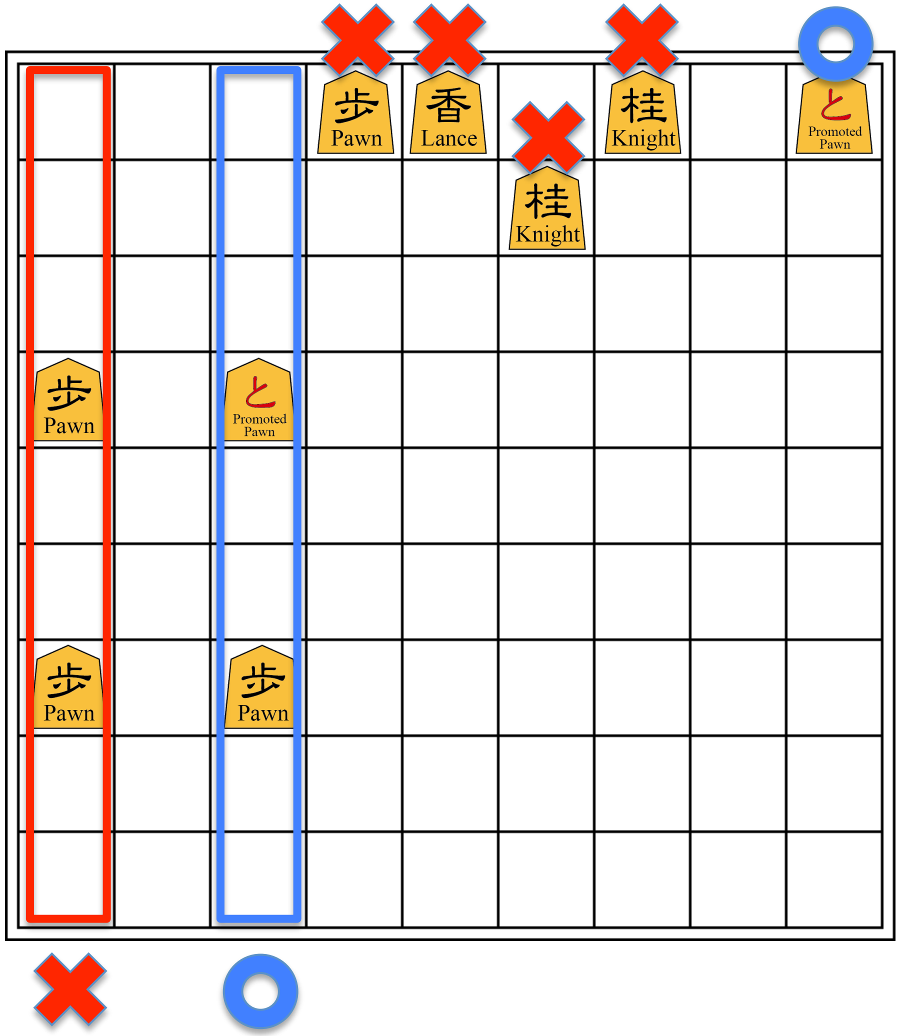
Main Theorem

- **Theorem 1**: There is a one-sided–error tester whose query complexity is $O(\varepsilon^{-2})$ for the generalized shogi problem.



Fouls

- Nifu (二歩, double pawn): Two or more unpromoted pawns of a same player never be the same column simultaneously.
- Dead end: Pawns, lances, and nights never be to moved to, or dropped onto cells from which they have no next moves.
- If only one player does such fouls in a given position S, the player loses.
- If both players do, S ends in a draw, i.e., S∉$\mathcal{W}$.

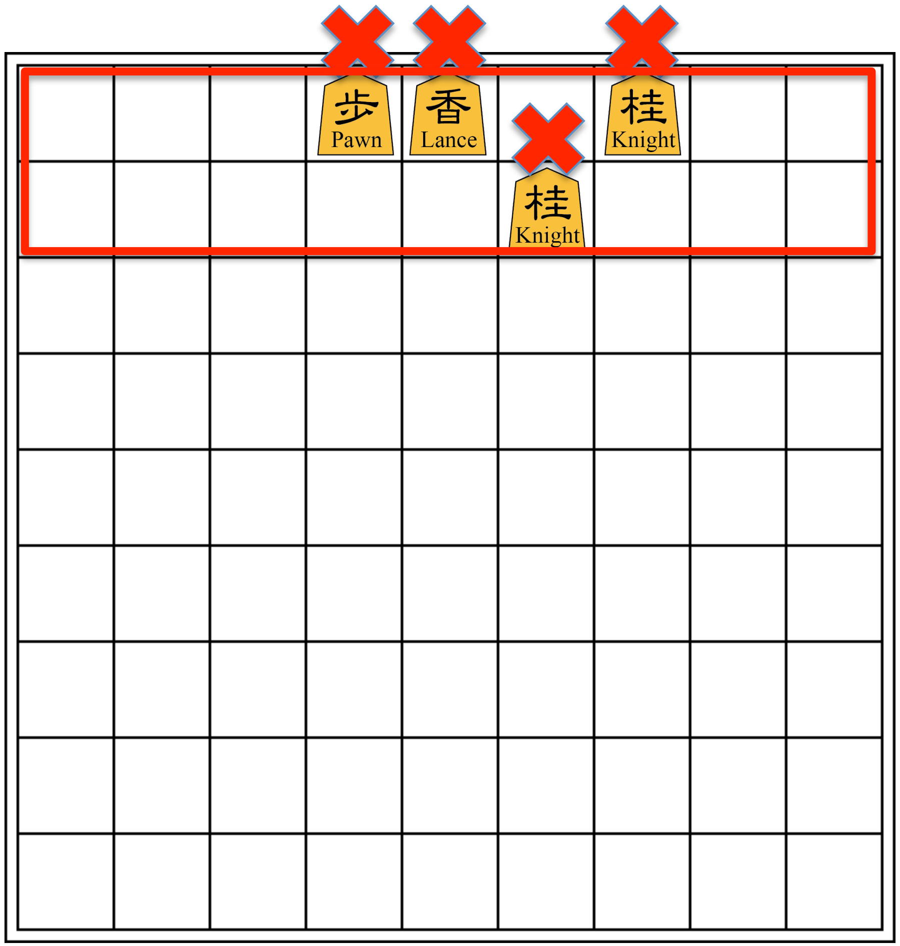


Nifu-free

- Let $\mathcal{N} \subset \Pi$ be the set of positions including no nifu-foul of Alice.
- Nifu-free generalized shogi problem: a promise problem of the generalized shogi problem such that every given position is from $\mathcal{N}$.
- **Lemma 1**: For any nifu-free position $S \in \mathcal{N}$ and any $0<\varepsilon\leq 1$, if $n>\max\{1/c, 36/\varepsilon^2\}$, then S is ε-close to $\mathcal{W}$, where n is the size of S.

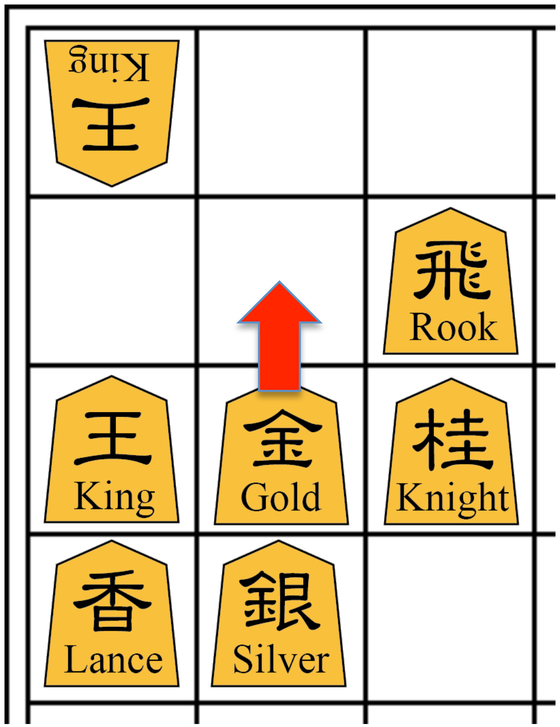
Proof of Lemma 1

- S is nifu-free.
- Then if Alice plays foul in S, it is a dead end.
- Alice's pieces that play this foul is in the first or the second rows.
- Thus, # of such pieces is at most 2√n.



Proof of Lemma 1

- S -> removing such foul pieces -> S'. dist(S,S') ≤ 2√n.
- Make S'' from S' by replacing pieces in cells (i,j), 1≤i≤4, 1≤j≤3 as:
- dist(S',S'')≤19 (=12+7)
- By the next Alice's move (red arrow) from S'', Bob's king is checkmated.
- S'' includes no Alice's foul.
- Thus, S''∈$\mathcal{W}$.
- dist(S,$\mathcal{W}$) ≤ dist(S,S'')
- ≤ dist(S,S') + dist(S',S'')
- ≤ 2√n+19.



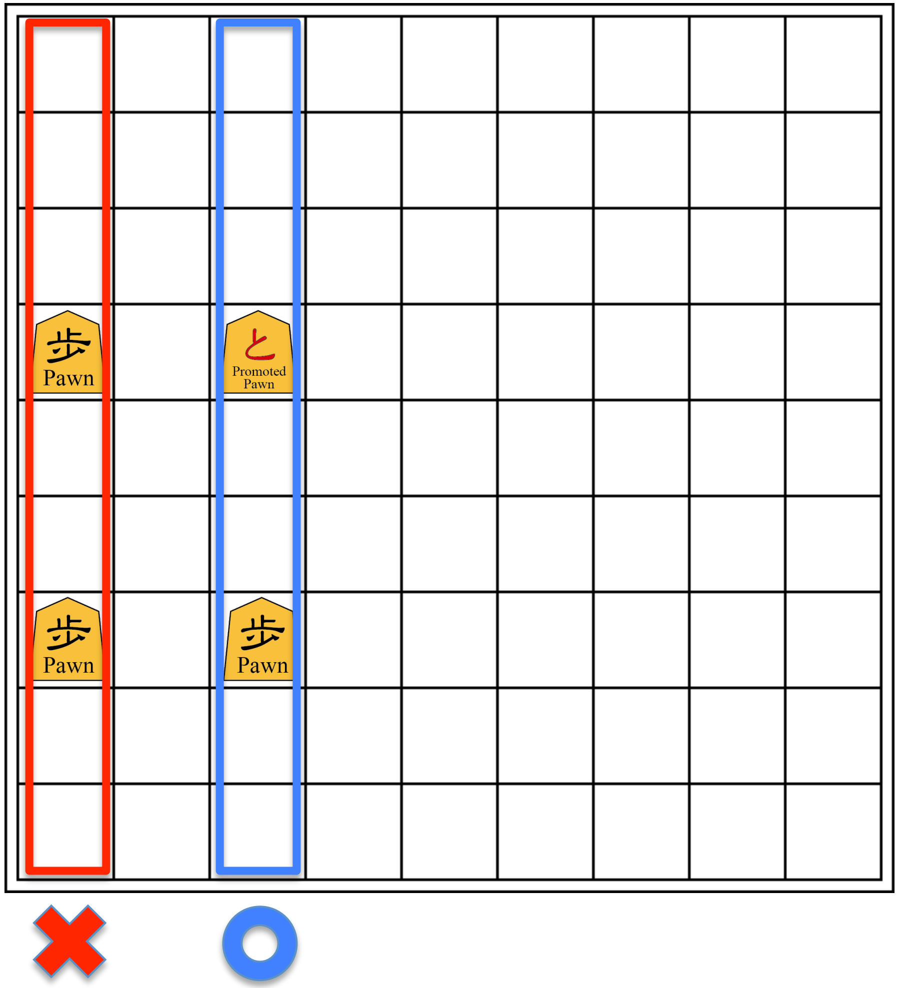
Proof of Lemma 1

- $\mathrm{dist}(S,\mathcal{W}) \le 2\sqrt{n}+19$.
- If $n \ge 25$, then $2\sqrt{n}+16 < 6\sqrt{n}$, and thus from $n > 36/\varepsilon^2$ (>25),
- $\mathrm{dist}(S,\mathcal{W}) \le 6\sqrt{n} = (6/\sqrt{n})n < \varepsilon n$. Q.E.D.

- **Lemma 1**: For any nifu-free position $S \in \mathcal{N}$ and any $0 < \varepsilon \le 1$, if $n > \max\{1/c, 36/\varepsilon^2\}$, then S is ε-close to $\mathcal{W}$, where n is the size of S.

Testing Nifu-freeness

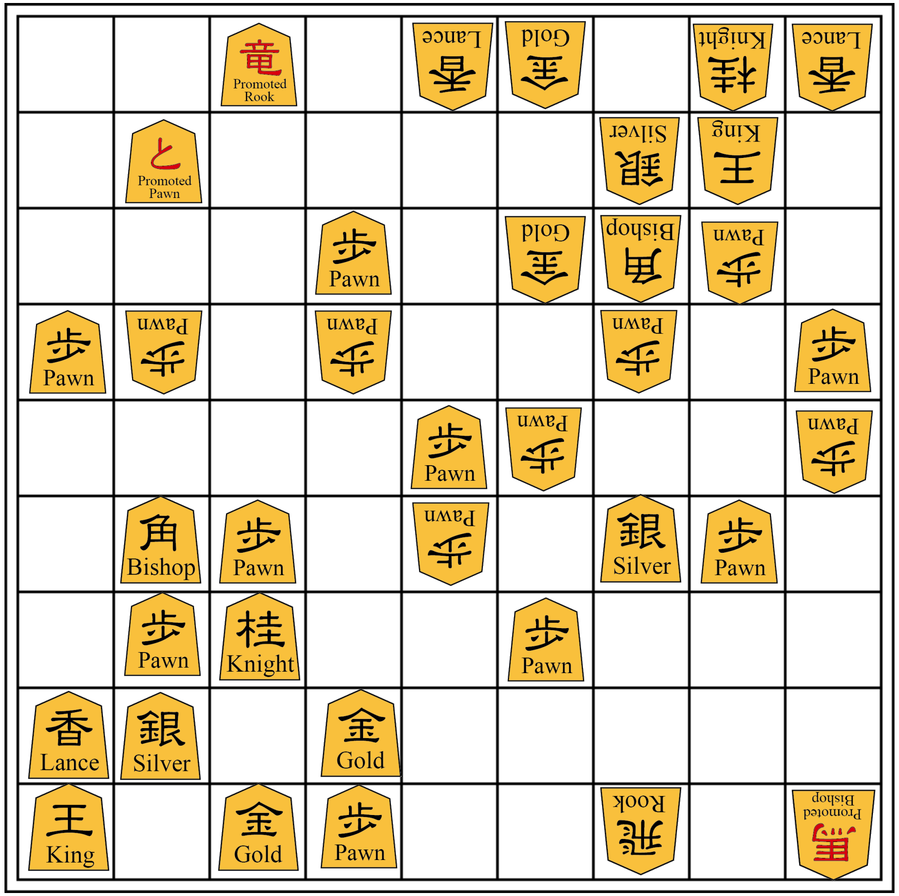
- **Lemma 2**: There is a one-sided-error tester for $\mathcal{N}$ (nifu-freeness) with query complexity $O(\varepsilon^{-2})$.



Algorithm

Procedure DetectingNifu(S,ε)

- Choose a column j and two rows i and i' (i≠i') u.a.r.
- If pieces in (i,j) and (i',j) are both Alice's unpromoted pawn, then "reject."
- Iterate the above $4/\varepsilon^2$ times.
- Output "accept," and stop.



Proof of Lemma 2

- If $n \leq \max\{1/c, 4/\varepsilon^2\}$, then we get the perfect information of S by calling oracle for all piece ID (k,L) and we can determine whether S is a winner or not.
- Then we assume $n > \max\{1/c, 4/\varepsilon^2\}$.
- We prove DetectingNifu works correctly.
- Since it rejects only when it finds nifu, it always accepts $S \in \mathcal{N}$, i.e., it is one-sided-error.

Proof of Lemma 2

- Assume S is ε-far from $\mathcal{N}$, i.e., there are more than εn Alice's pawn on the board. The algorithm selects (i,j) and (i',j) with i≠j u.a.r.
- P := Pr[the algorithm rejects S in one comparison]
- p := # of pairs of Alice's pawn in the same column.
- x: Total # of Alice's pawn on the board.
- x_j: # of Alice's pawn on row j.
- $x_1+x_2+\cdots+x_r = x > \varepsilon n$, where $r := \lfloor \sqrt{n} \rfloor$.
- Let $f(x) := x(x-1)/2$.
- Then $p = f(x_1) + \cdots + f(x_r)$.

Proof of Lemma 2

- Since f(x) is convex, from Jensen's inequality,

$$p = f(x_1) + \cdots + f(x_r) \geq rf\left(x/r\right) \geq rf\left(\varepsilon\sqrt{n}\right).$$

- Therefore

$$P = \frac{p}{rf(r)} \geq \frac{f\left(\varepsilon\sqrt{n}\right)}{f\left(\sqrt{n}\right)} = \frac{\varepsilon\sqrt{n}\left(\varepsilon\sqrt{n}-1\right)}{\sqrt{n}\left(\sqrt{n}-1\right)} = \varepsilon^2 \frac{\sqrt{n}-1/\varepsilon}{\sqrt{n}-1}.$$

- From $n > 4/\varepsilon^2, \sqrt{n} - 1/\varepsilon > \sqrt{n}/2$ holds and

$$P > \frac{\varepsilon^2}{2} \cdot \frac{\sqrt{n}}{\sqrt{n}-1} > \frac{\varepsilon^2}{2}$$

- follows.

Proof of Lemma 2

- Thus the algorithm rejects prob. ≥ $\varepsilon^2/2$ in one iteration.
- # of iterations is $4/\varepsilon^2$, the prob. of S is not rejected through all iterations is less than

$$\left(1-\frac{\varepsilon^2}{2}\right)^{\frac{4}{\varepsilon^2}} = \left(\left(1-\frac{\varepsilon^2}{2}\right)^{\frac{2}{\varepsilon^2}}\right)^2 \le \left(\left(e^{-\frac{2}{\varepsilon^2}}\right)^{\frac{2}{\varepsilon^2}}\right) = \left(e^{-1}\right)^2 < \frac{1}{3}.$$

$$1+x \le e^x, \forall x$$

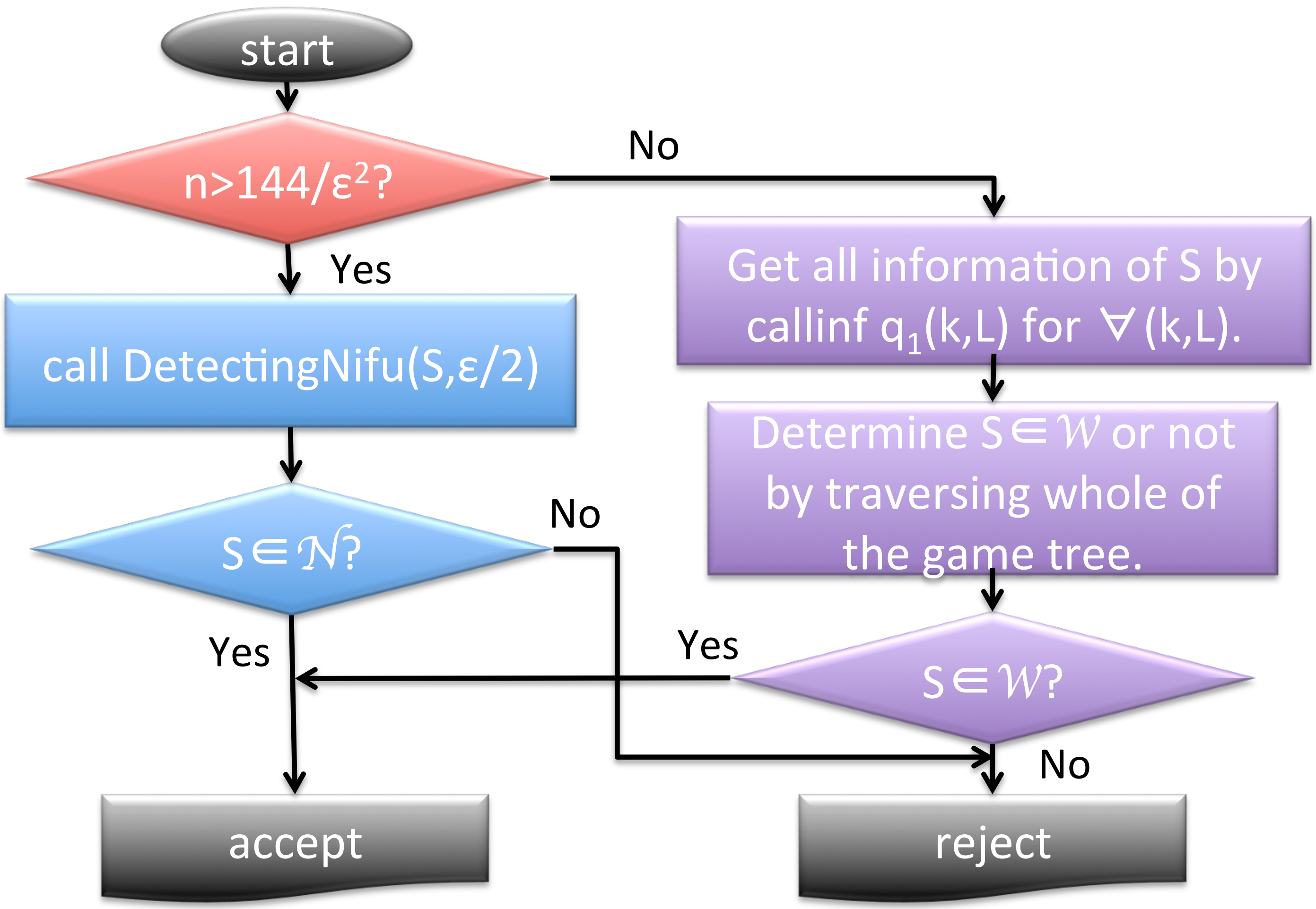
- Therefore, S, which is ε-far from $\mathcal{N}$ is rejected with prob.≥2/3.
- Q.E.D.

Main Theorem

- **Theorem 1**: There is a one-sided–error tester whose query complexity is $O(\varepsilon^{-2})$ for the generalized shogi problem.

- **Lemma 1**: For any nifu-free position $S \in \mathcal{N}$ and any $0<\varepsilon\leq 1$, if $n>\max\{1/c, 16/\varepsilon^2\}$, then S is ε-close to $\mathcal{W}$, where n is the size of S.
- **Lemma 2**: There is a one-sided-error tester for $\mathcal{N}$ (nifu-freeness) with query complexity $O(\varepsilon^{-2})$.

Algorithm (Tester) for Theorem 1

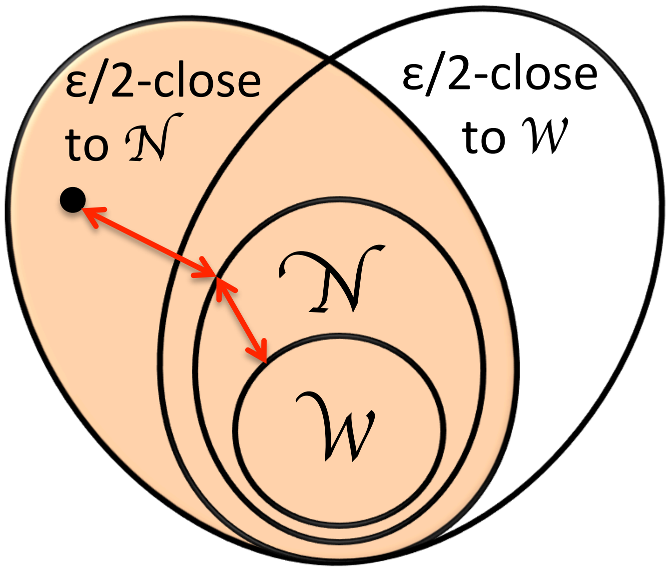


Correctness of the Tester

- Obtained from Lemmas 1 and 2.

If $n \leq 144/\varepsilon^2$ -> Clear.

If $n > 144/\varepsilon^2$



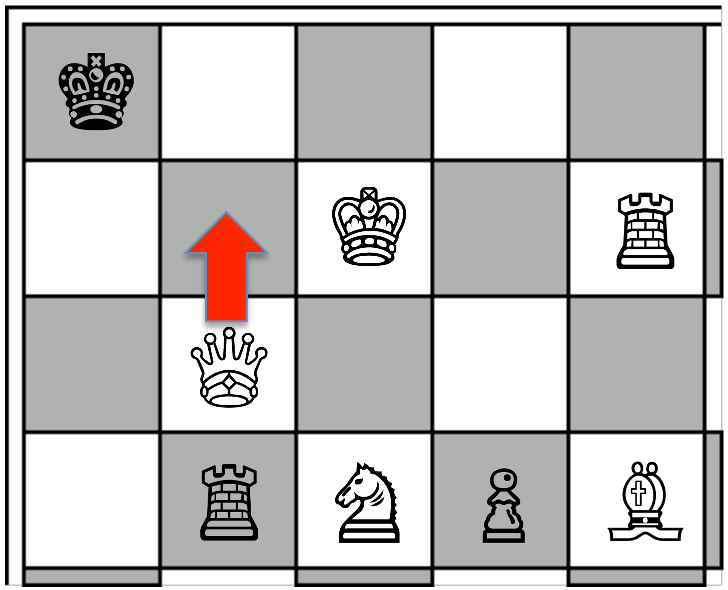
Q.E.D.

Chess

- **Theorem 2**: There is a no-error tester whose query complexity is $O(\varepsilon^{-1})$ for the generalized chess problem.

Lemma 3: If $n>28/\varepsilon$, any position is ε-close to "Winner."
Proof: →

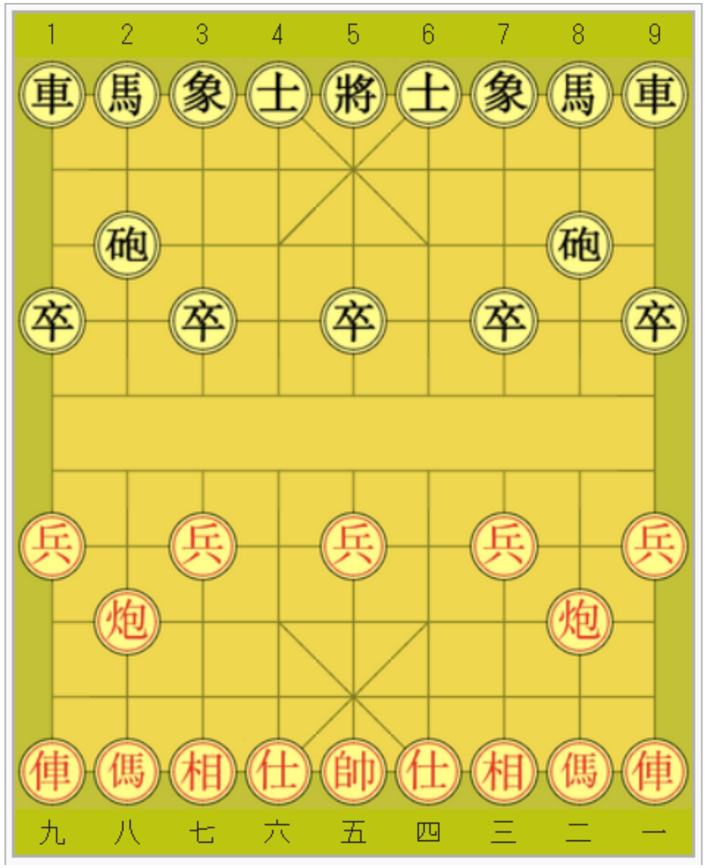
Note: In chess, there is no foul like as in nifu, then we don't care about such a foul.



一般化シャンチー問題の検査可能性

- シャンチーとは中国とベトナムで盛んな将棋型のゲーム。
- 将(または師)を取れば勝ちとなる。

- 基本的なルールはチェスに似ており、取られた駒は再利用できない。



一般化将棋型問題の検査可能性

- 定理3　一般化シャンチー問題は質問計算量が$O(\varepsilon^{-1})$である無誤りの検査アルゴリズムが存在する。

- 方法は将棋、チェスと同様。

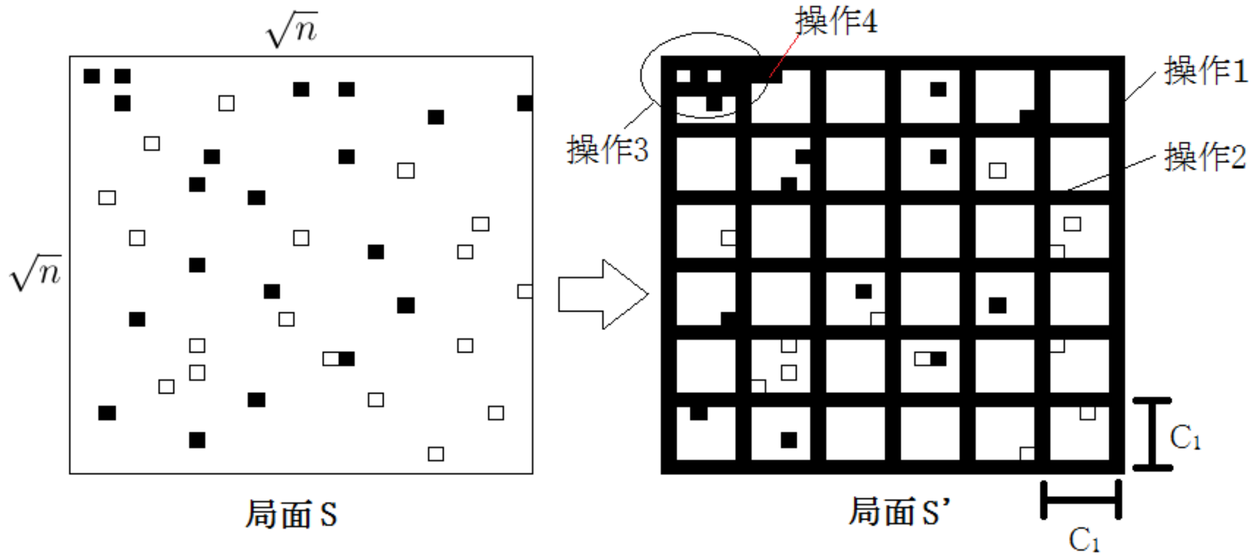
一般化囲碁問題

- 一般化囲碁問題とは盤面をに拡張し、
  石を個に増やした上で、
  任意の局面を入力として与えて、そこから
  先手と後手が最善を尽くしたときに先手が勝つか、
  そうでないかを判定する問題とする。

- 定理4?　一般化囲碁問題には質問計算量が$O(\varepsilon^{-2})$である検査アルゴリズムが存在する。

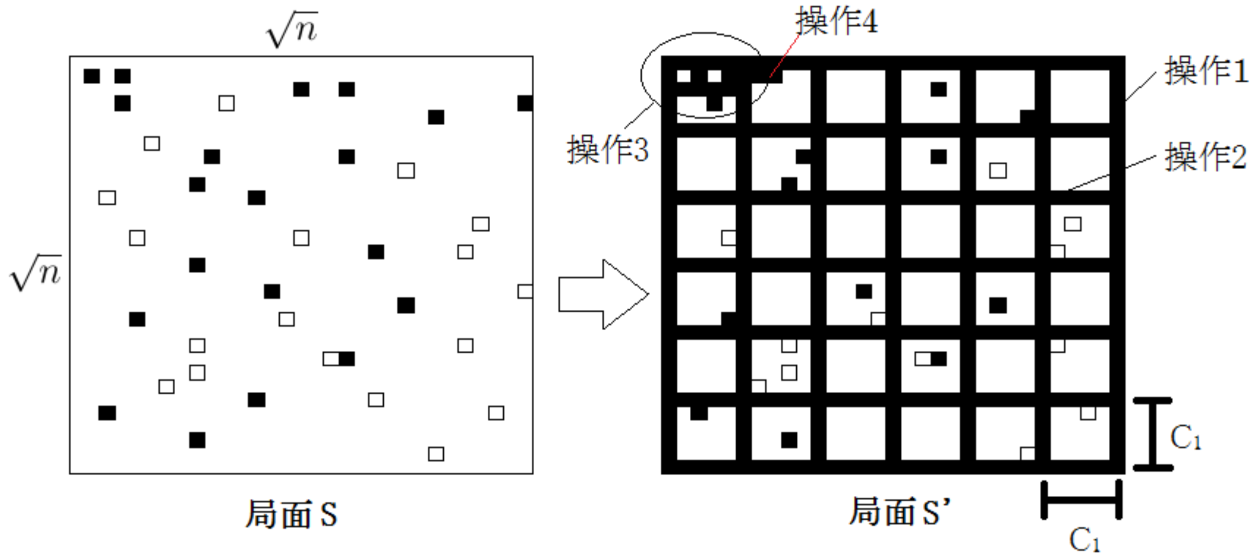
一般化囲碁問題の検査可能性

- 一般化囲碁問題の考え方
- 入力として与えられた任意の局面を図のようなな局面に変更する。
  に変更した際に先手は不利にならない。
- ただし点の数え方は中国式とする（盤面上の石の数＋地の数）。



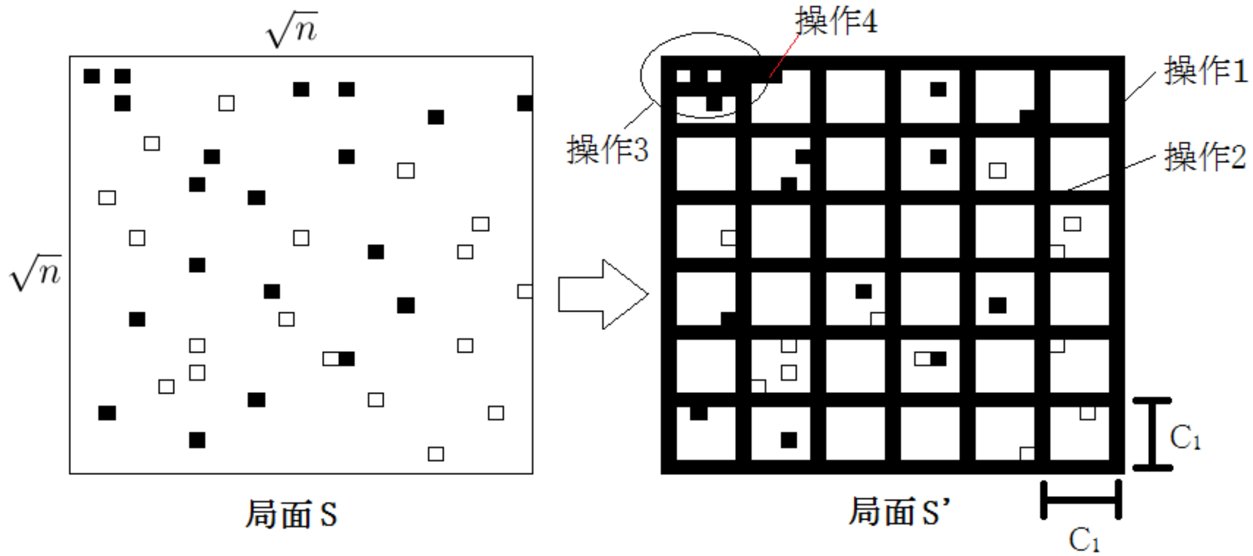
一般化囲碁問題

- 分割された各ブロックは独立に勝敗が決まる。
- アルゴリズムはこれらから定数個のブロックを一様ランダムに選び、ブロックの得点の平均値xを出す。



一般化囲碁問題

- その平均値がある定数tに対して
  X
  を満たすならば受理し、そうでないならば拒否する。



まとめ

- 一般化将棋問題は$O(\varepsilon^{-2})$で片側誤り検査可能。
- 一般化チェスとシャンチーは$O(\varepsilon^{-1})$で無誤り検査可能。
- 一般化囲碁は$O(\varepsilon^{-2})$で（多分）検査可能。