理論計算機科学特論 (2026 年前学期)

計算複雑性の基礎

第 10 回

交代性計算：AP = PSPACE

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2026 年 6 月 16 日

最終更新：2026 年 6 月 17 日　12:39

スケジュール (前半)

1. 計算理論の復習 (4/7)
2. 時間計算量：P, NP, coNP (4/14)
3. 帰着と完全性：NP 完全 (4/21)
4. 領域計算量：L, NL, PSPACE (4/28)

$*$ 休み (祝日) (5/5)

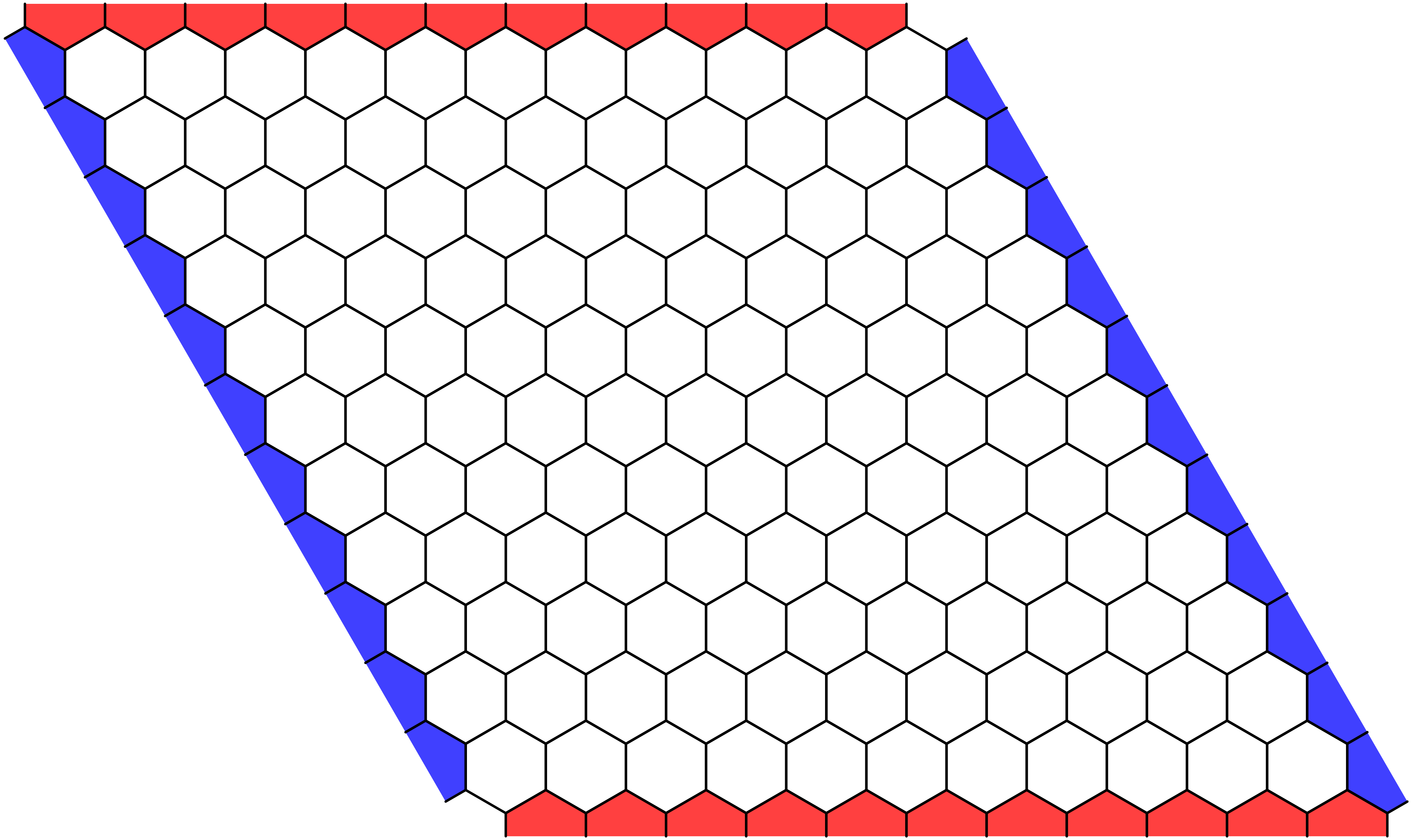
5. 時間と領域の関係：$\mathrm{P} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$ (5/12)
6. 階層定理：$\mathrm{P} \neq \mathrm{EXPTIME}$ (5/19)
7. Ladner の定理：$\mathrm{NP} - \mathrm{P} = \mathrm{NPC} \Rightarrow \mathrm{P} = \mathrm{NP}$ (5/26)

スケジュール (後半)

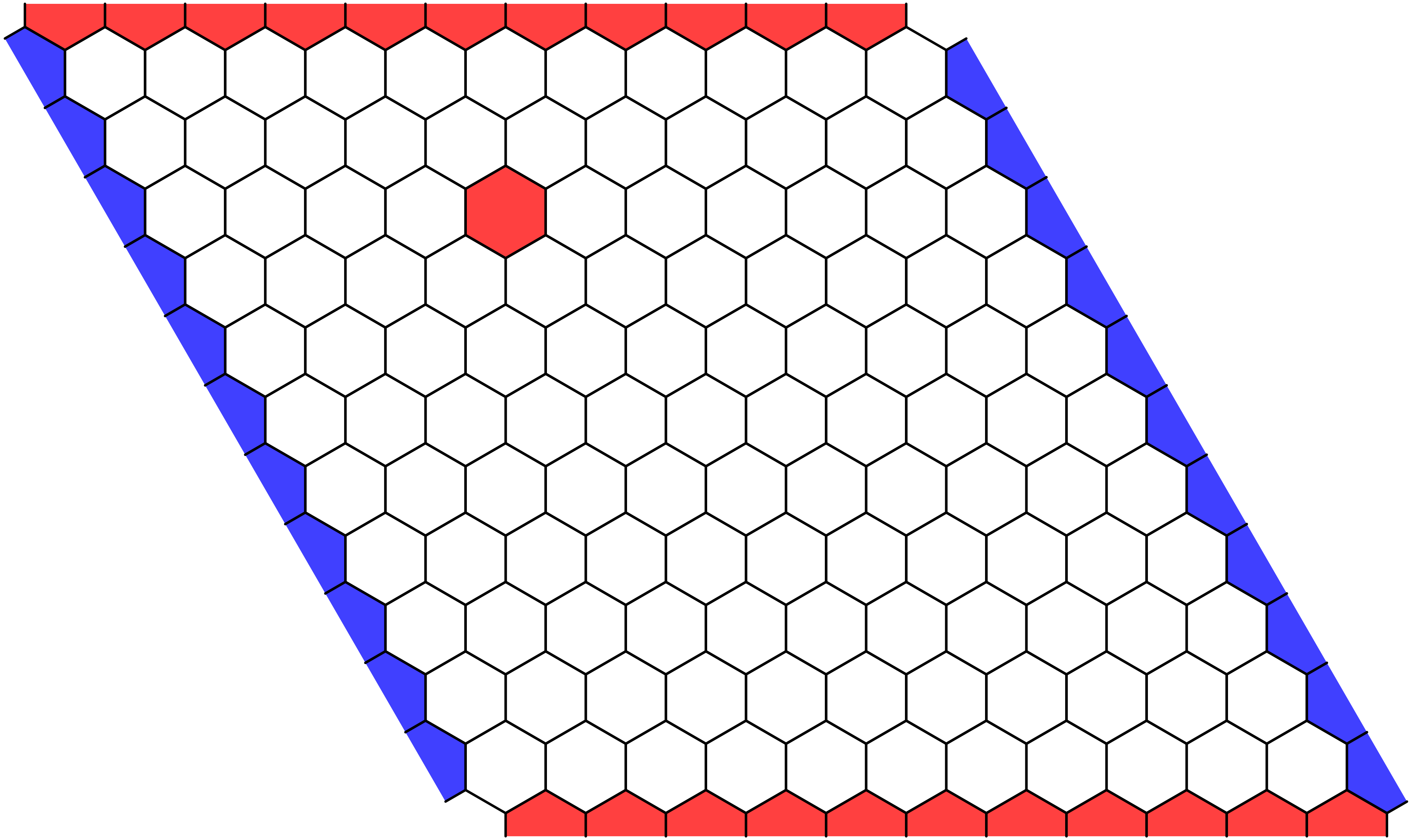
8. Savitch の定理：$\mathrm{PSPACE} = \mathrm{NPSPACE}$ (6/2)
9. Immerman-Szelepcsényi の定理：$\mathrm{NL} = \mathrm{coNL}$ (6/9)
10. **交代性計算：$\mathrm{AP} = \mathrm{PSPACE}$** (6/16)
11. 多項式階層：$\mathrm{P} = \mathrm{NP} \Rightarrow \mathrm{P} = \mathrm{PH}$ (6/23)
12. 確率的計算：$\mathrm{P} \subseteq \mathrm{BPP} \subseteq \mathrm{PP}$, $\mathrm{NP} \subseteq \mathrm{PP}$ (6/30)
13. 対話証明系 (1)：$\mathrm{NP} \subseteq \mathrm{MA} \subseteq \mathrm{AM}$ (7/7)
14. 対話証明系 (2)：$\mathrm{IP} \subseteq \mathrm{PSPACE}$ (7/14)
15. 対話証明系 (3)：$\mathrm{PSPACE} \subseteq \mathrm{IP}$ (7/21)

$*$ 休み (授業のない日) (7/28)

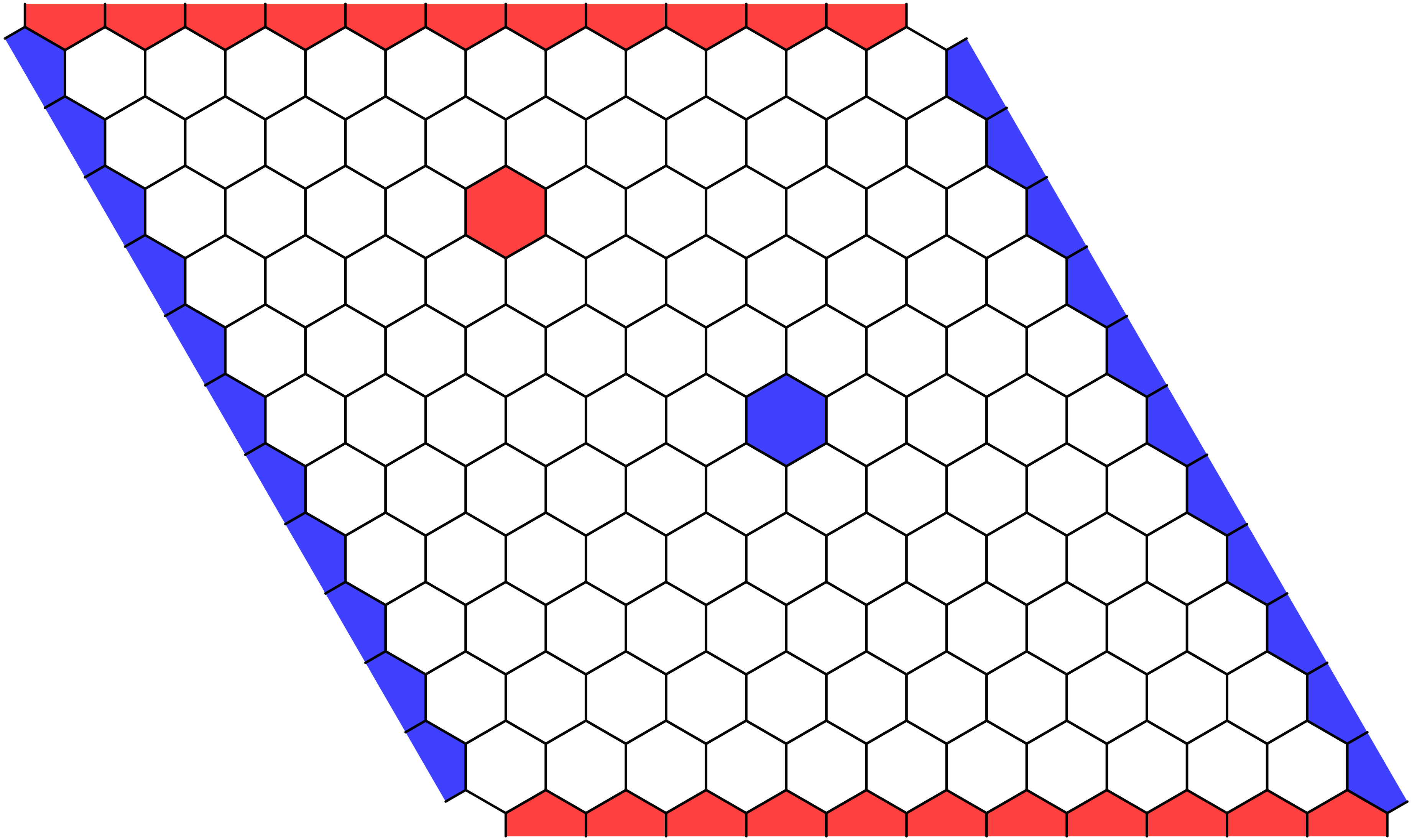
HEX (ヘックス)



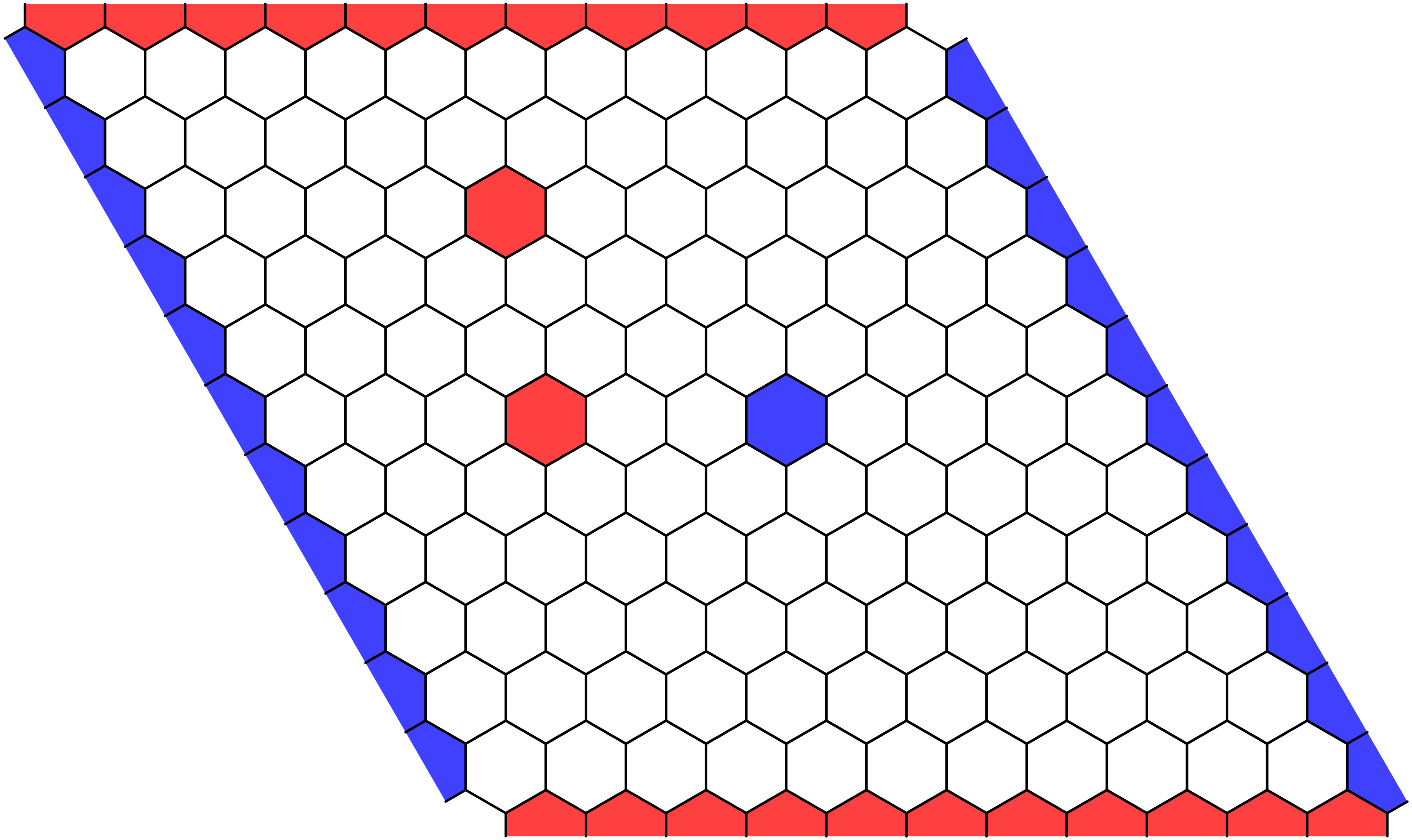
HEX (ヘックス)



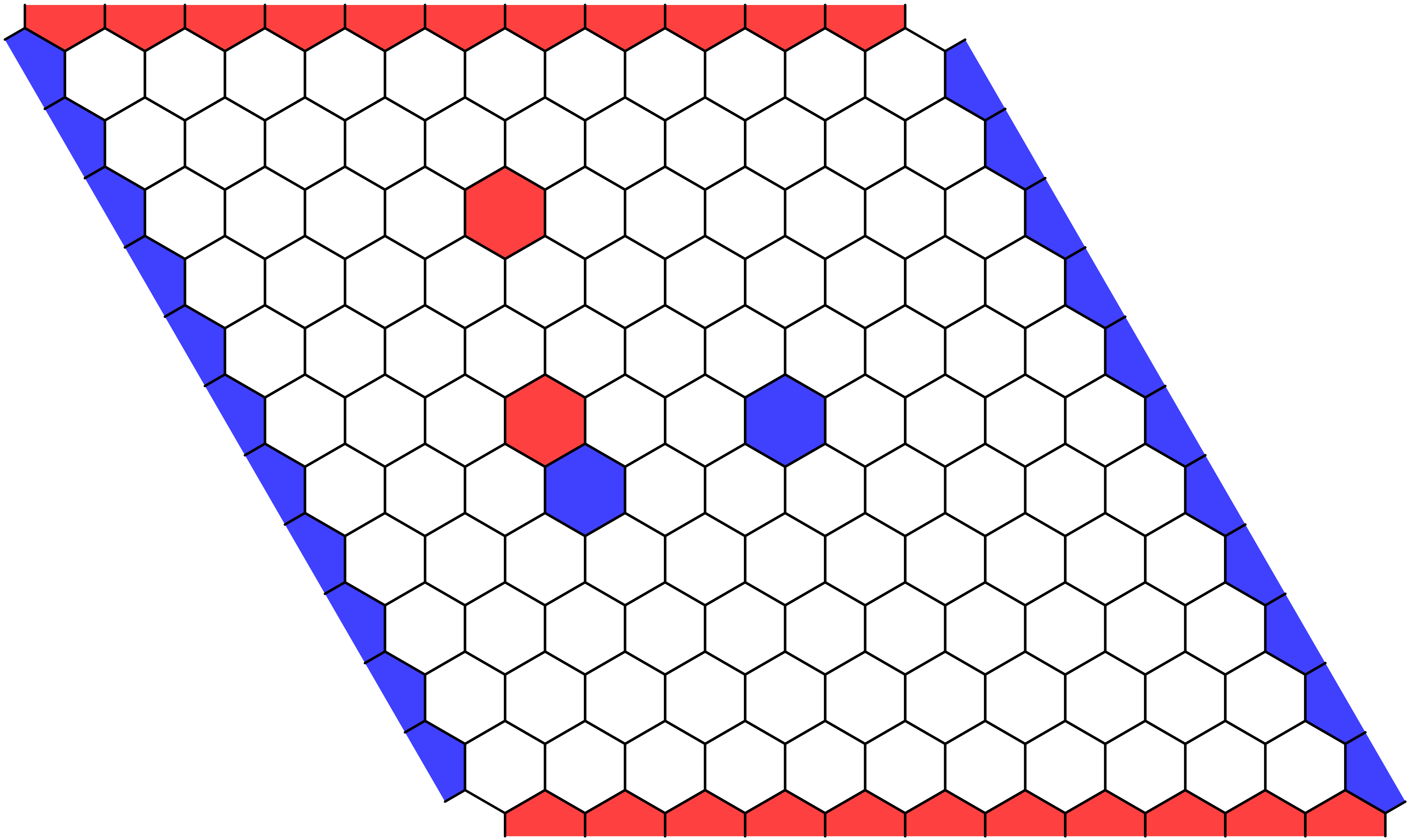
HEX (ヘックス)



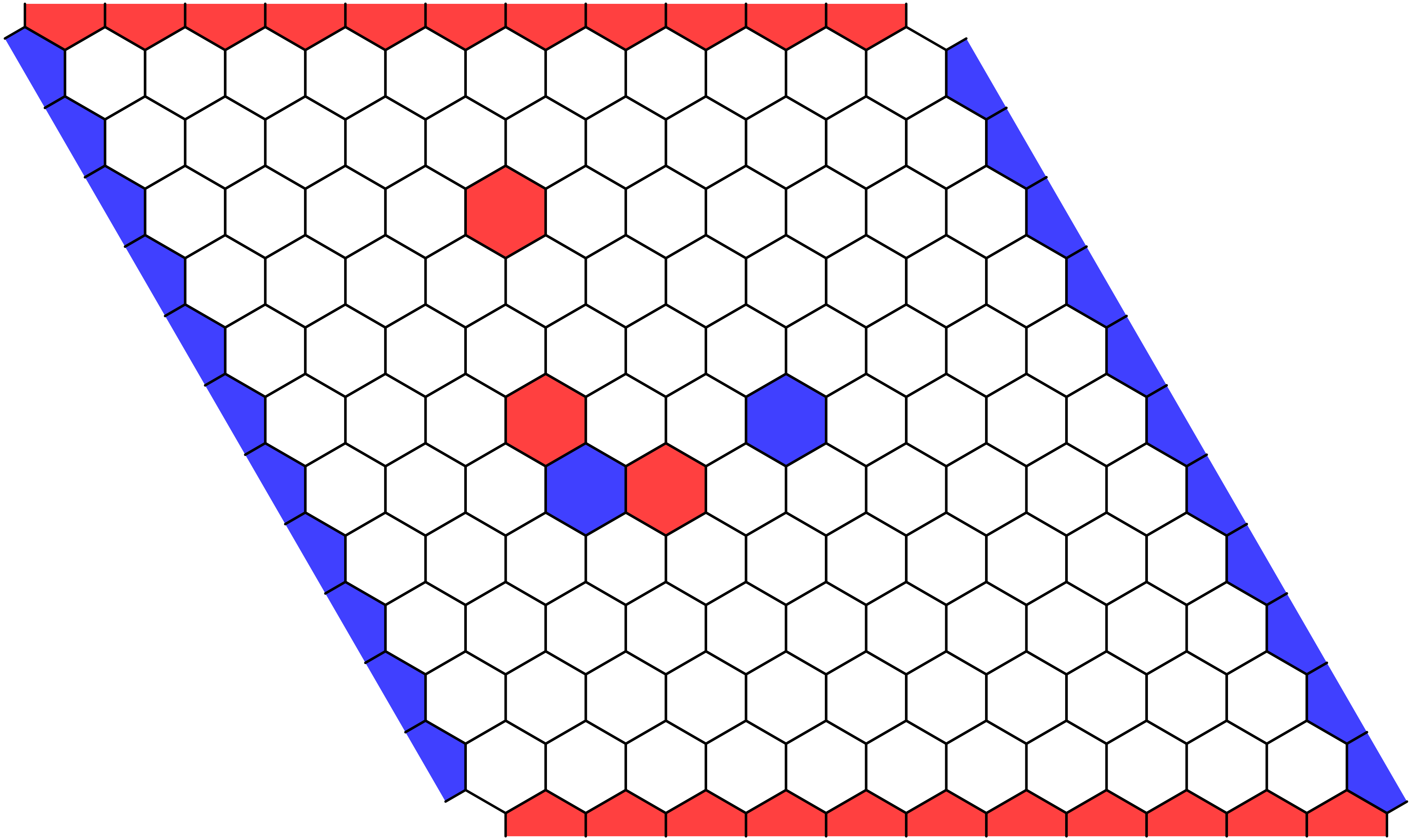
HEX (ヘックス)



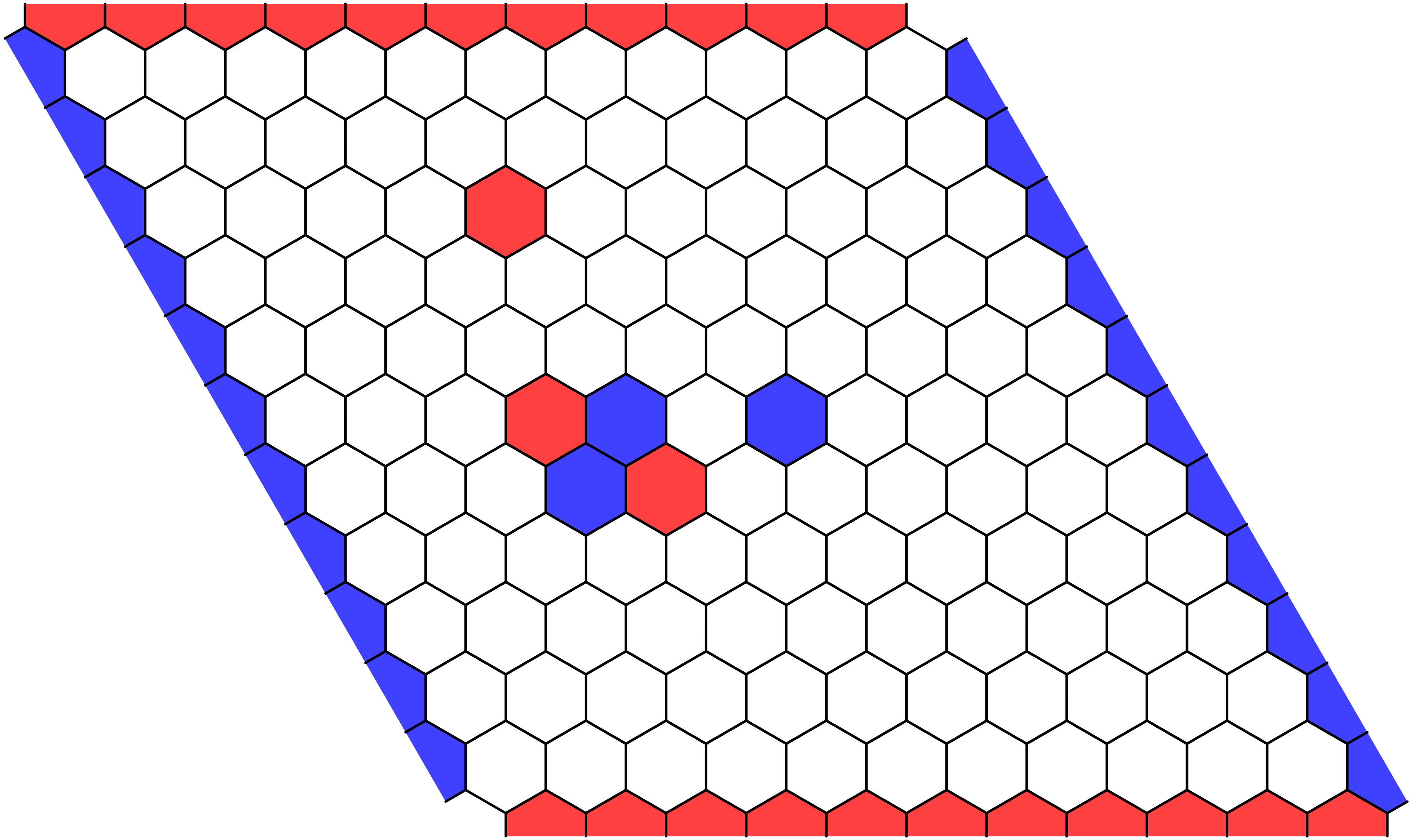
HEX (ヘックス)

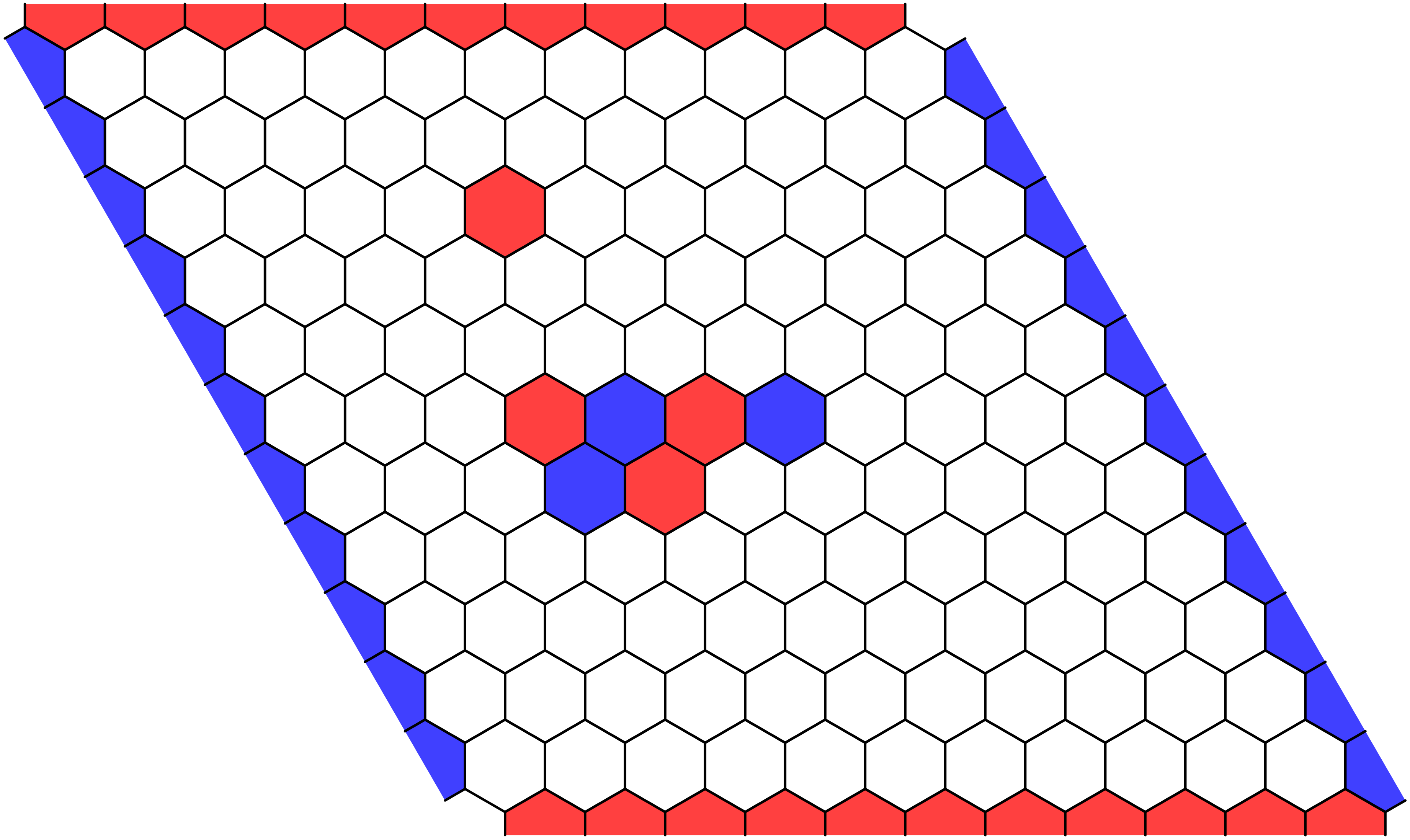


HEX (ヘックス)

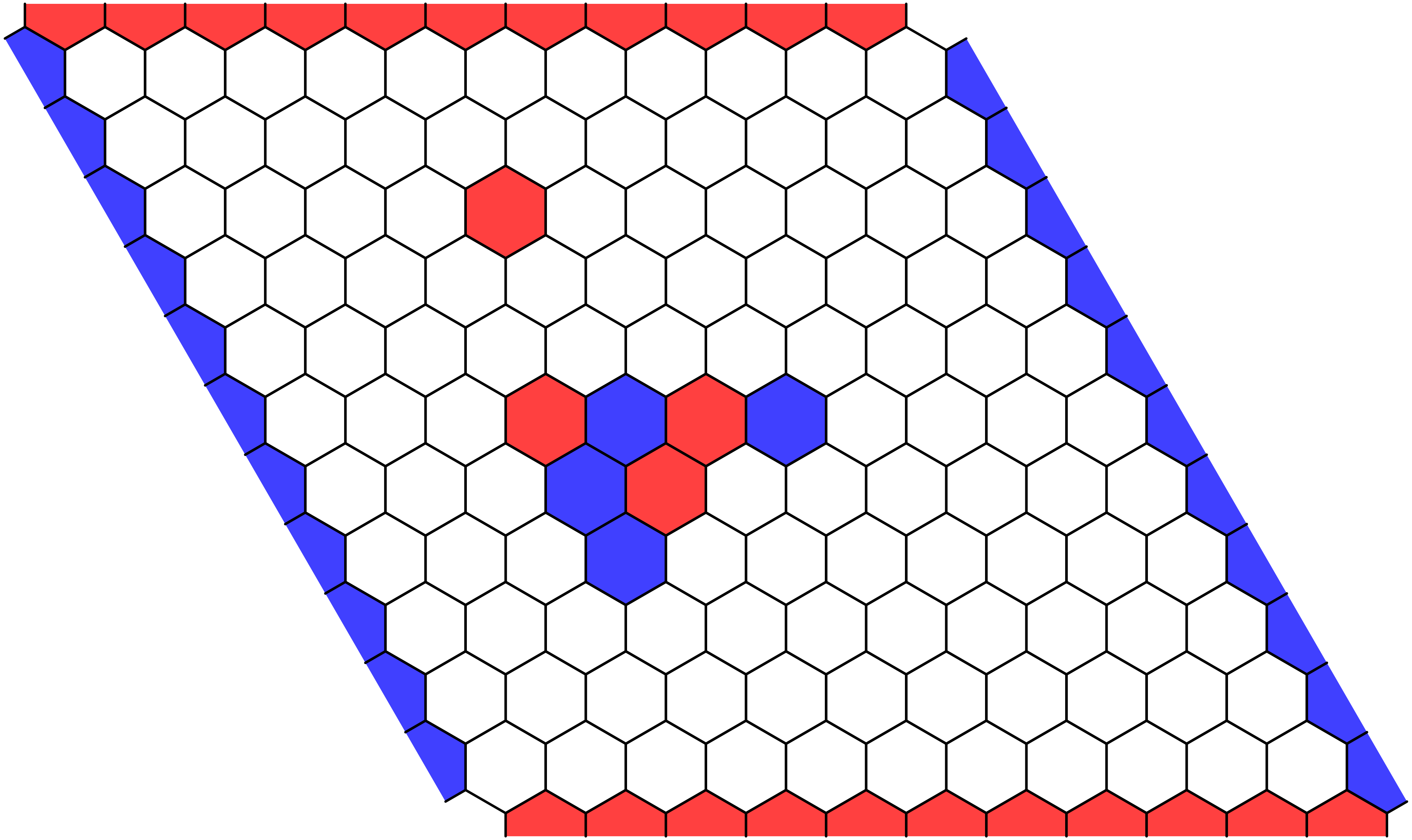


HEX (ヘックス)

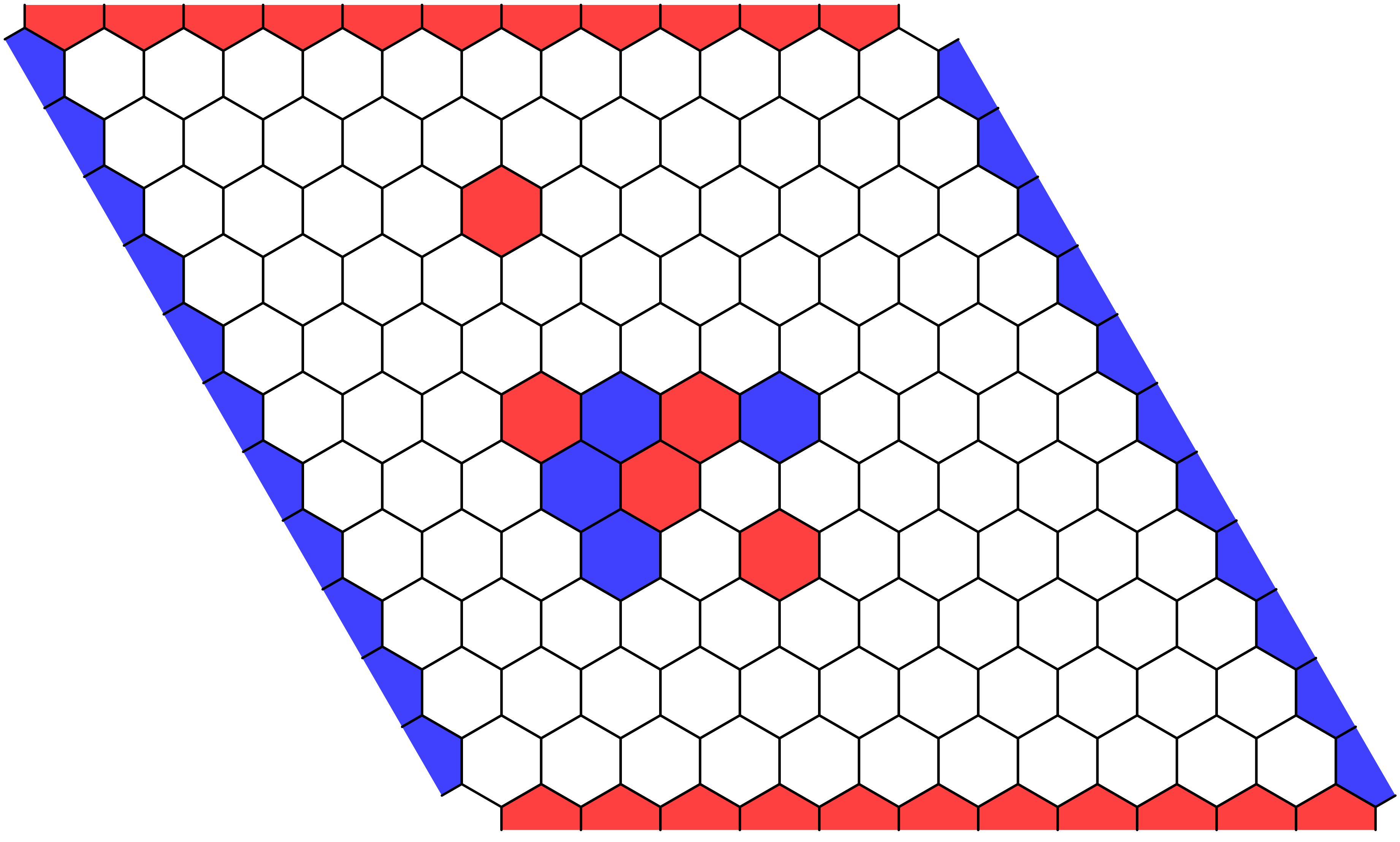




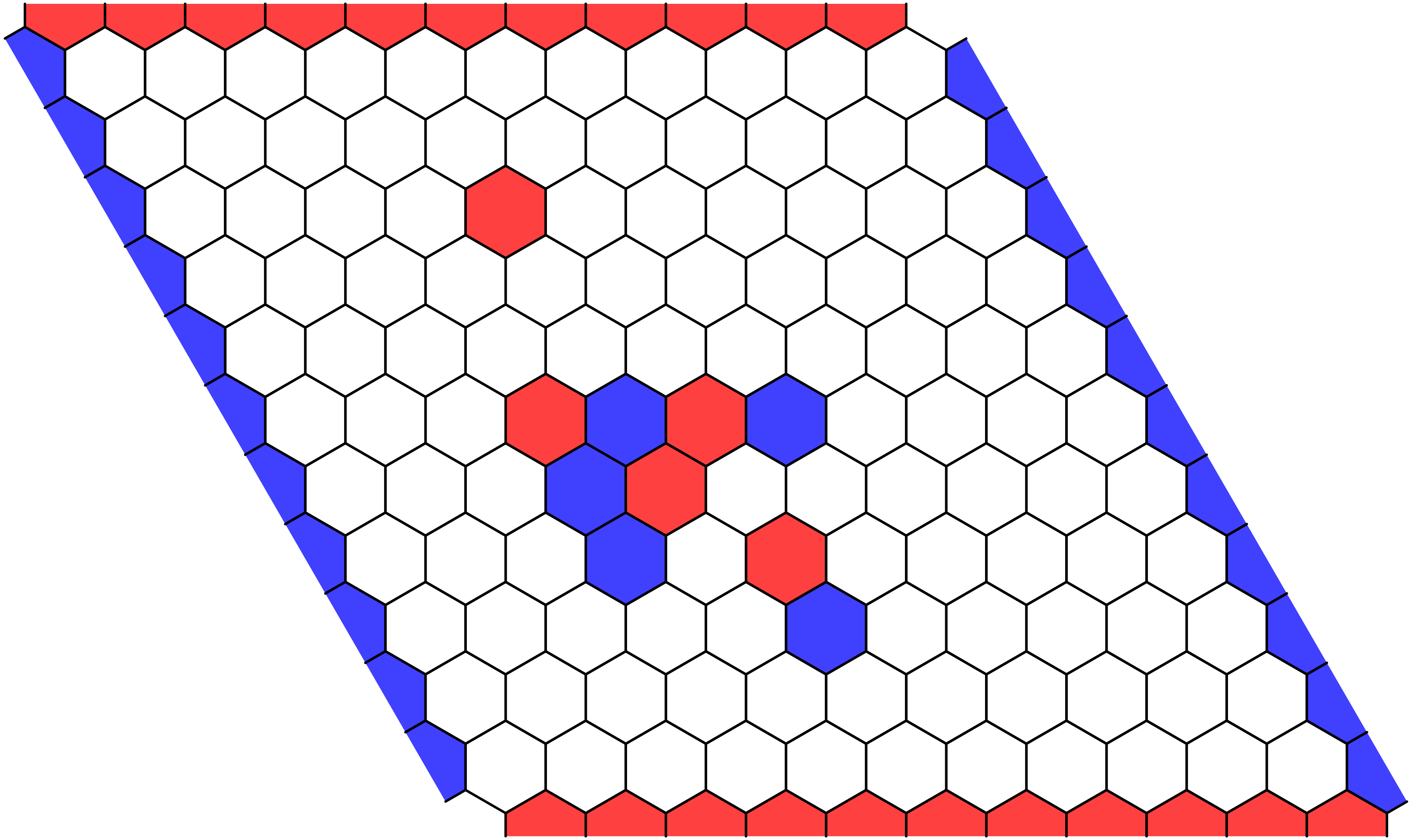
HEX (ヘックス)



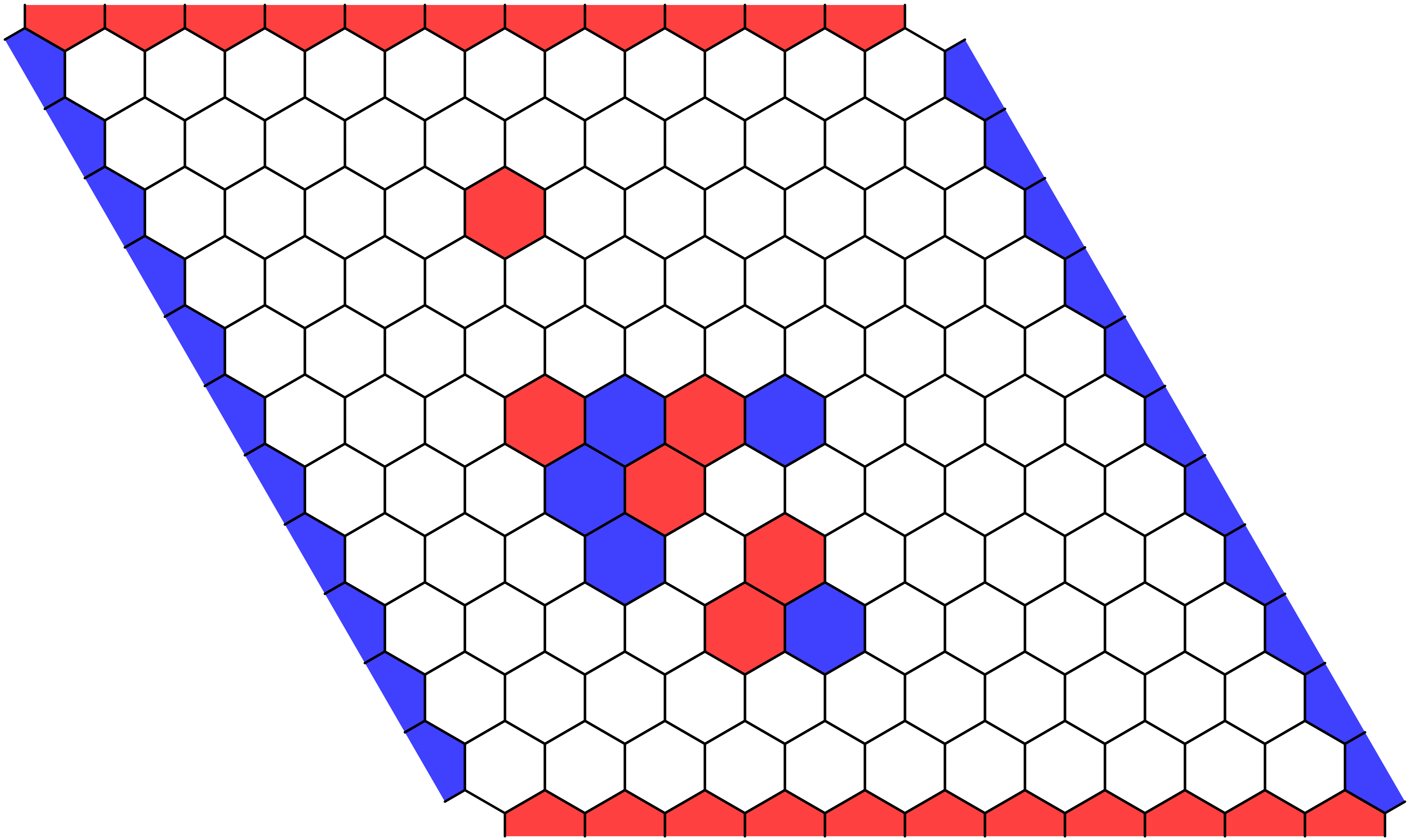
HEX (ヘックス)



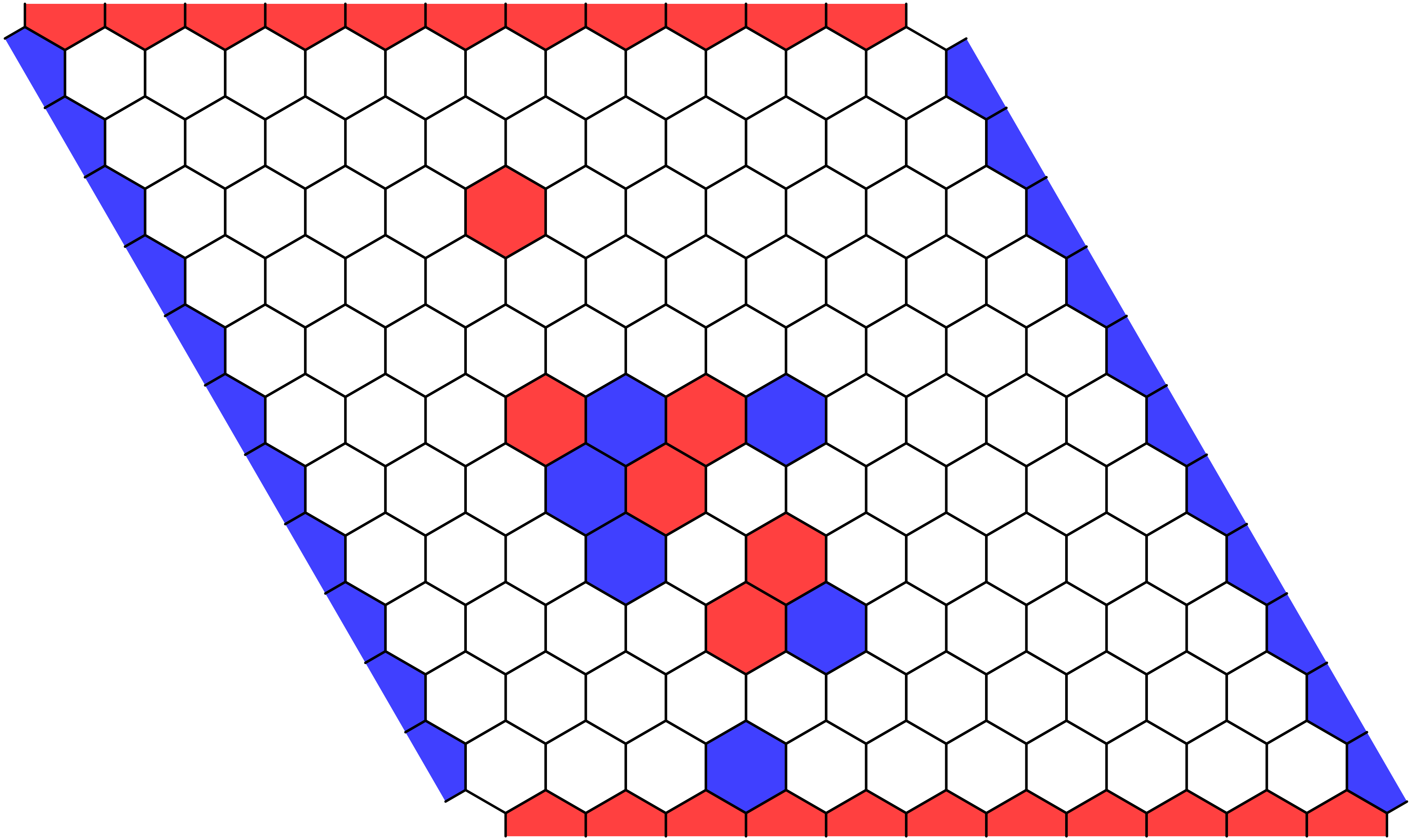
HEX (ヘックス)



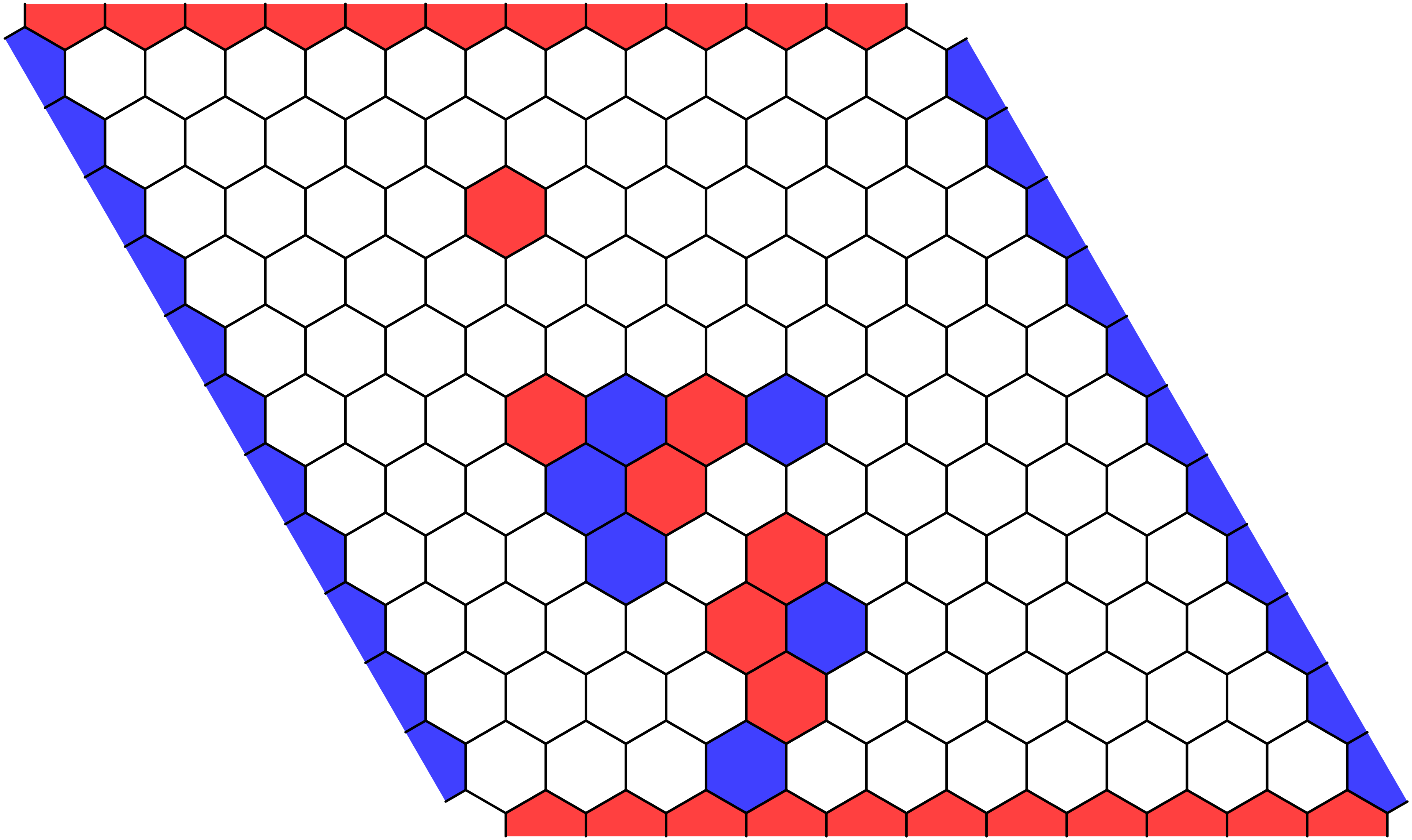
HEX (ヘックス)



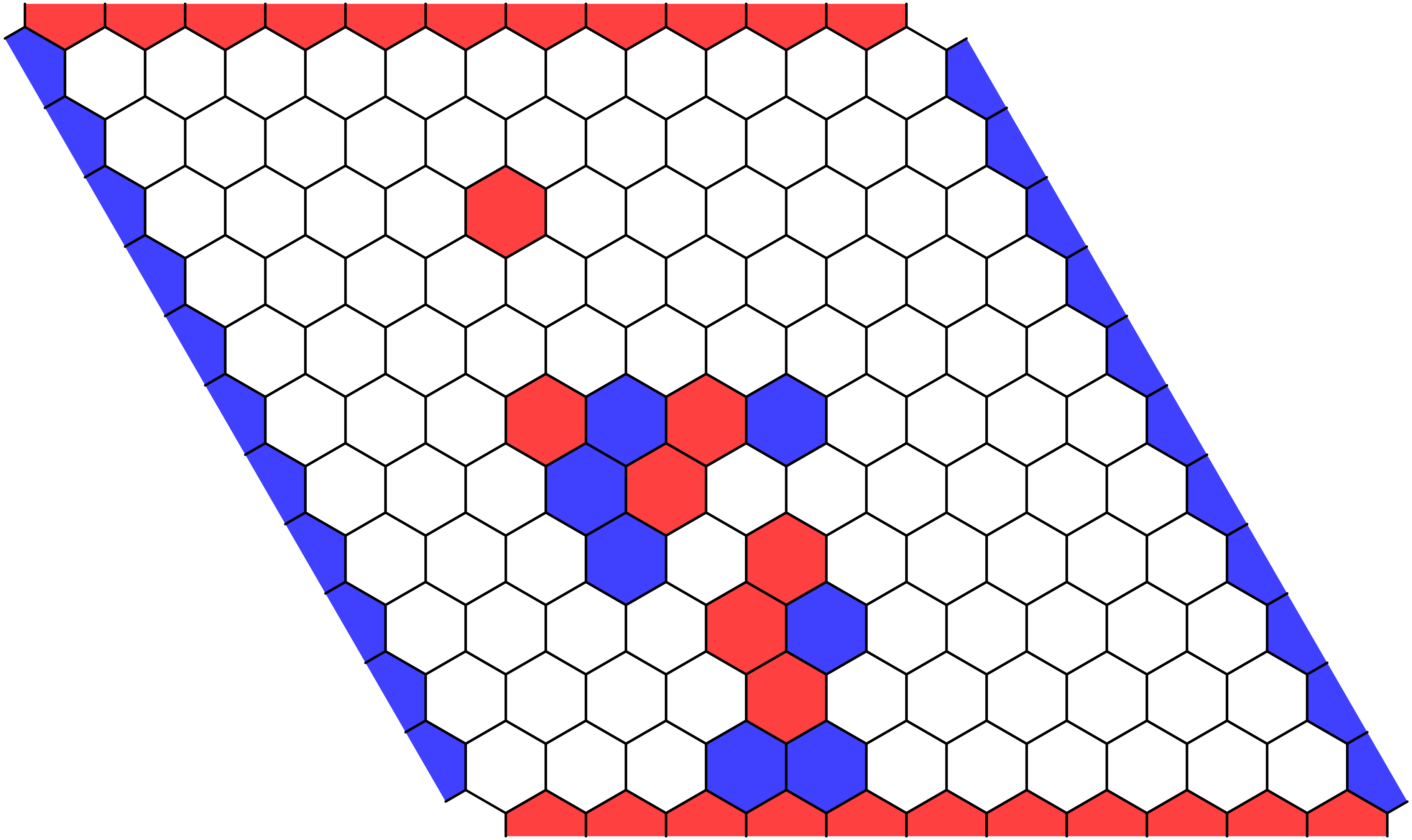
HEX (ヘックス)



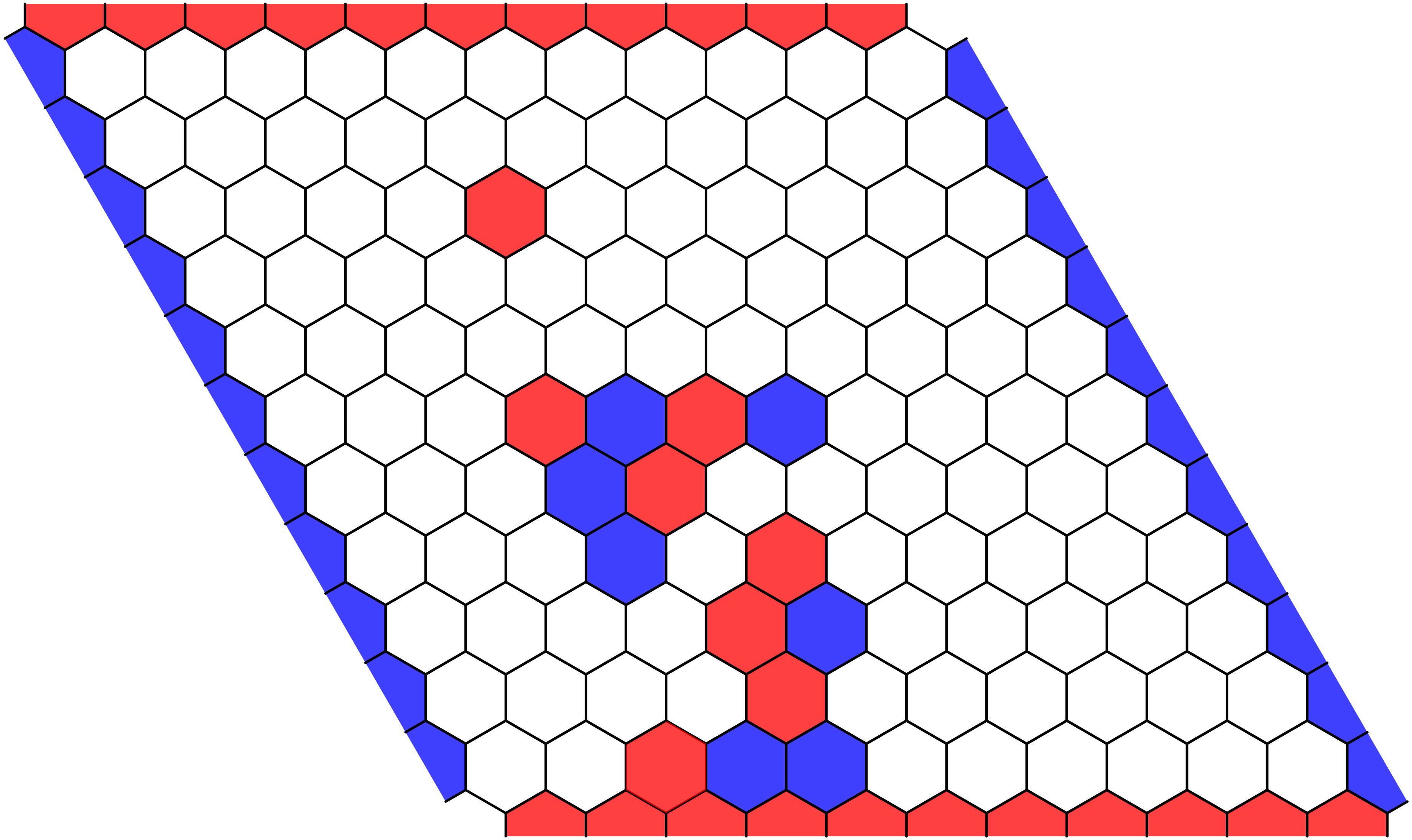
HEX (ヘックス)

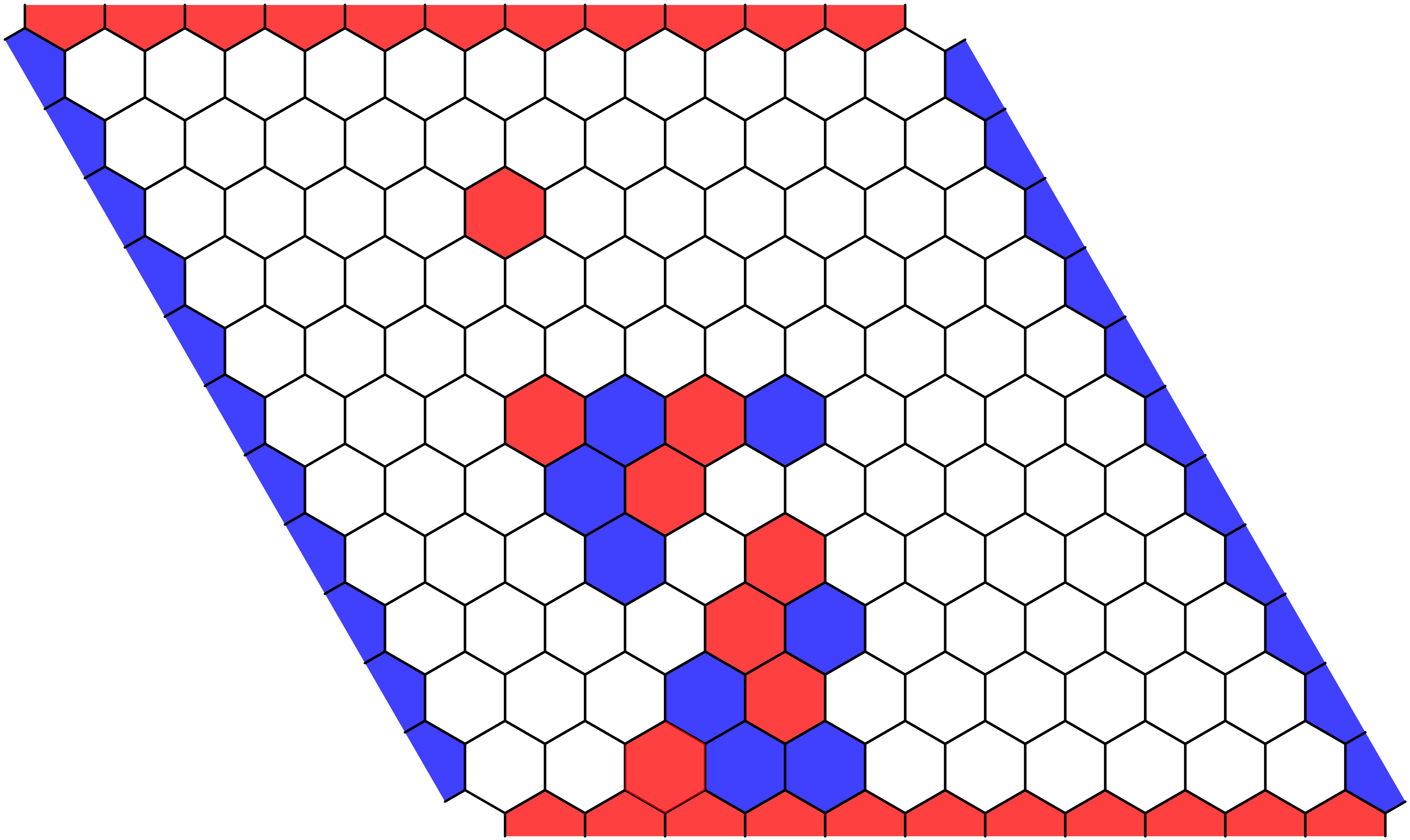


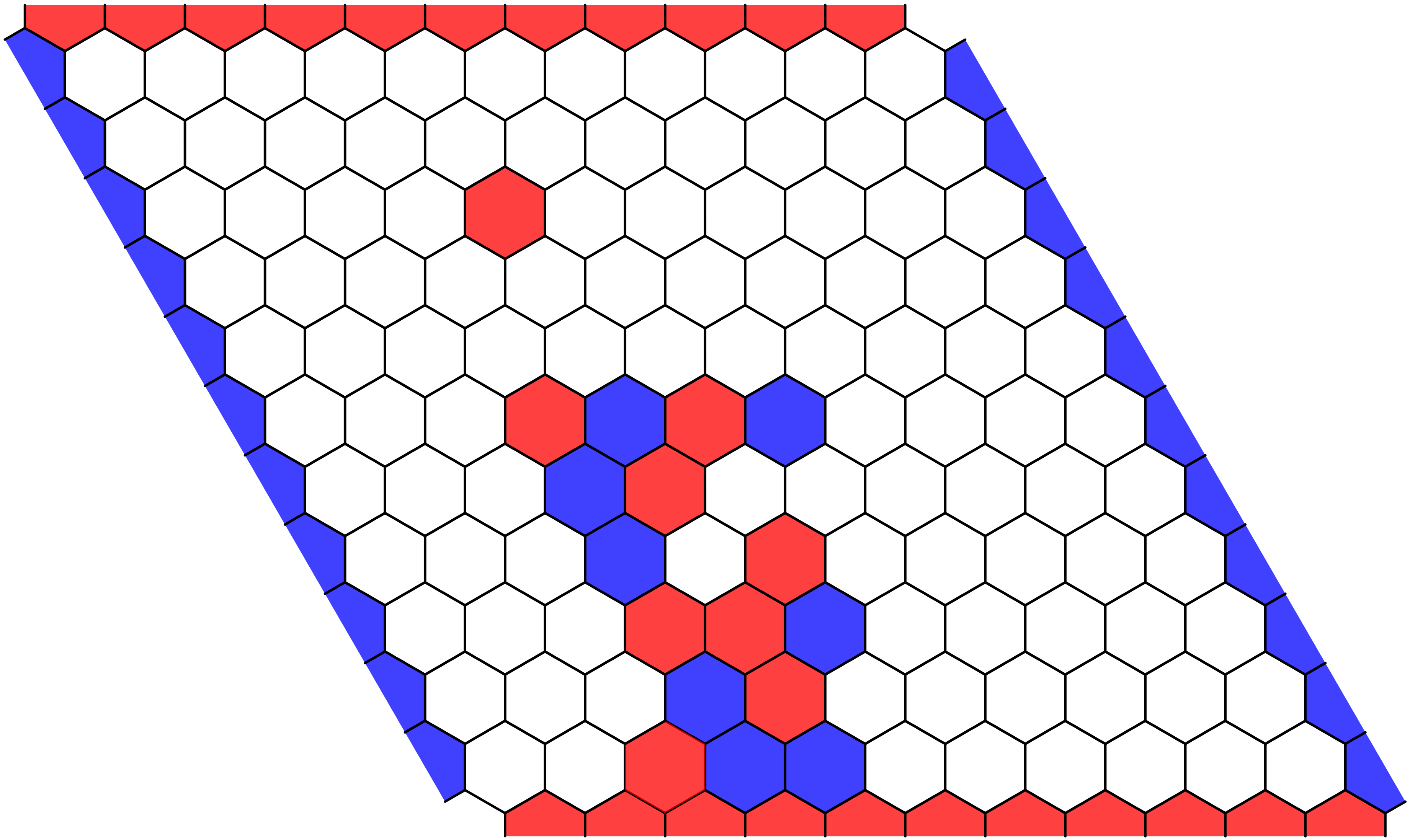
HEX (ヘックス)



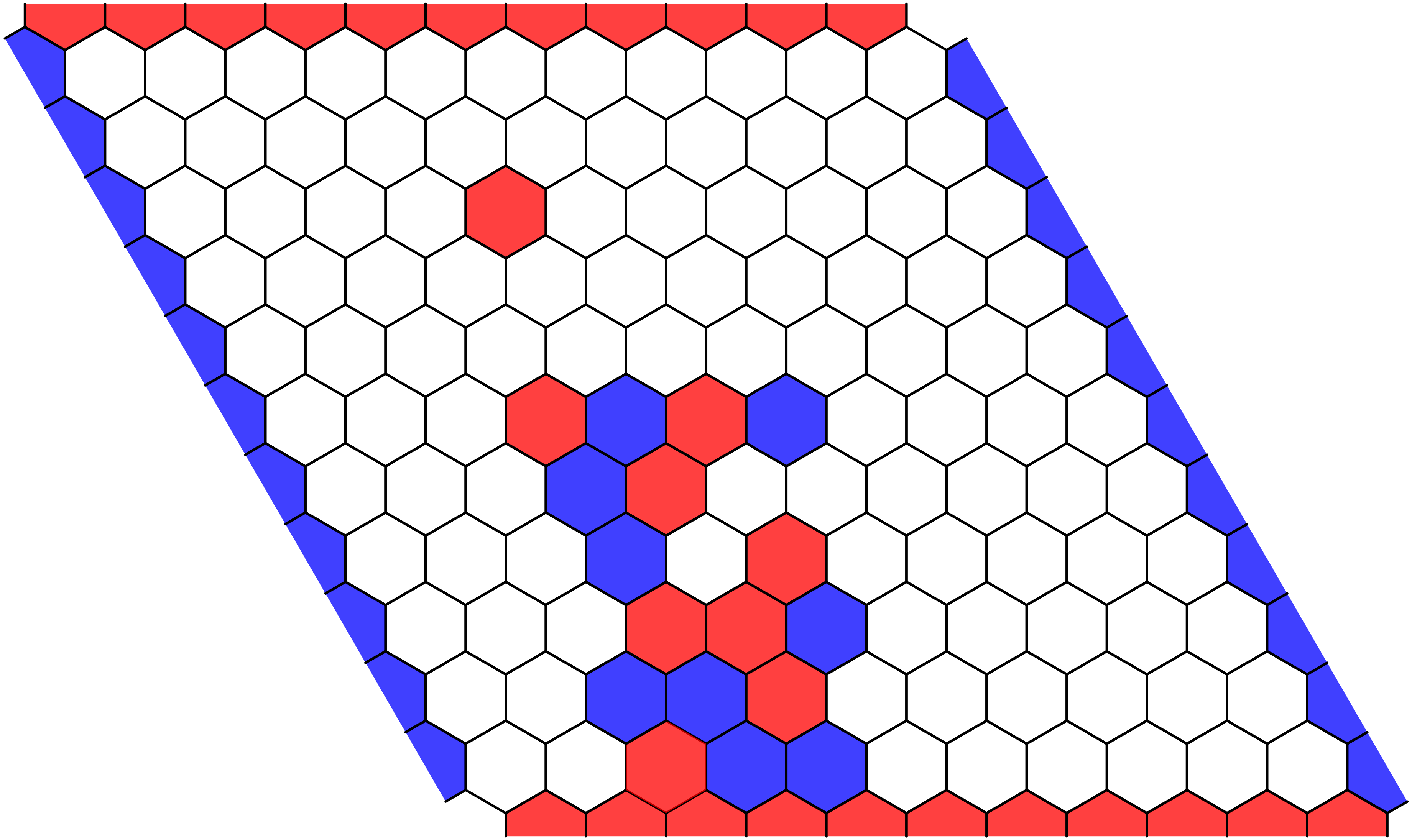
HEX (ヘックス)

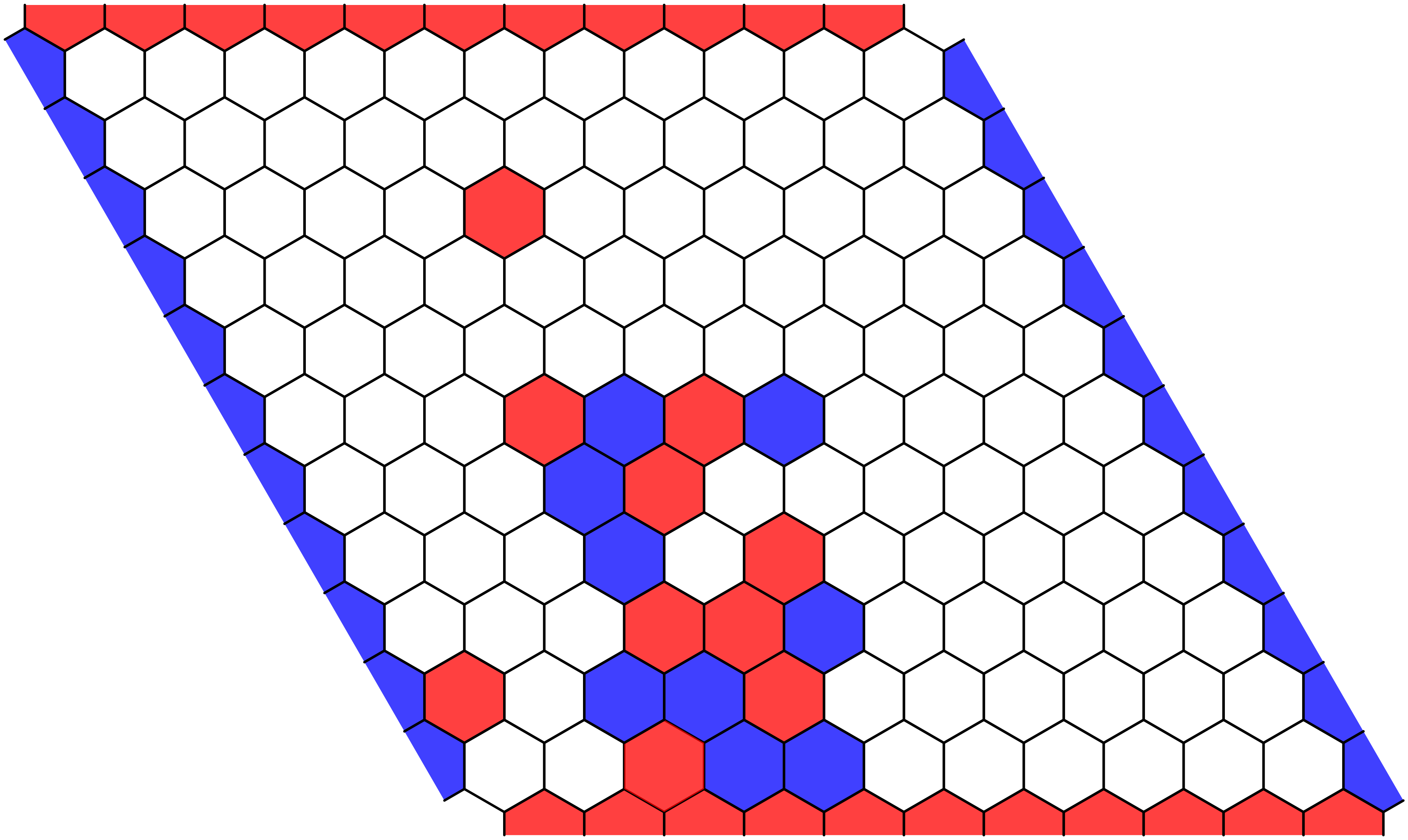




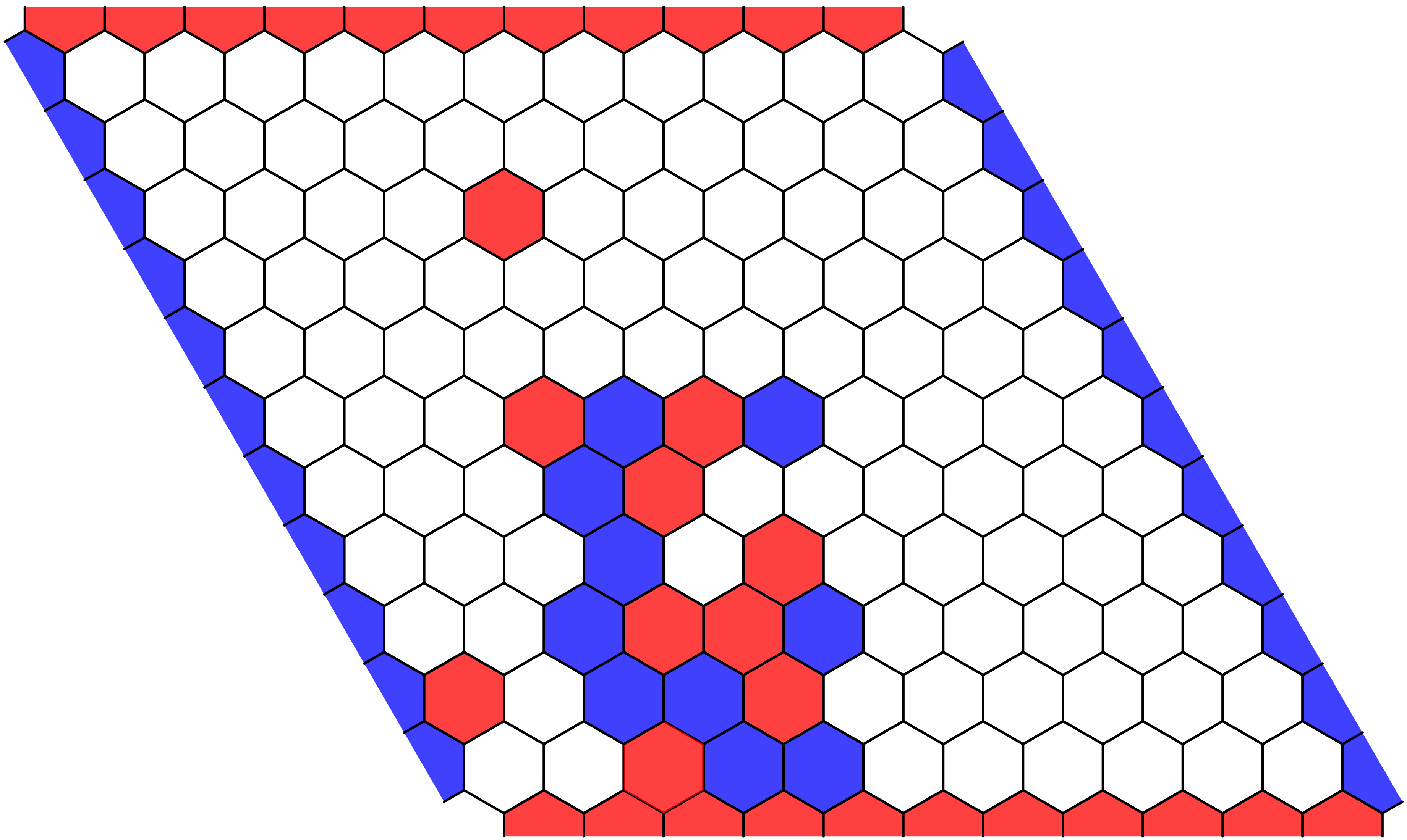


HEX (ヘックス)

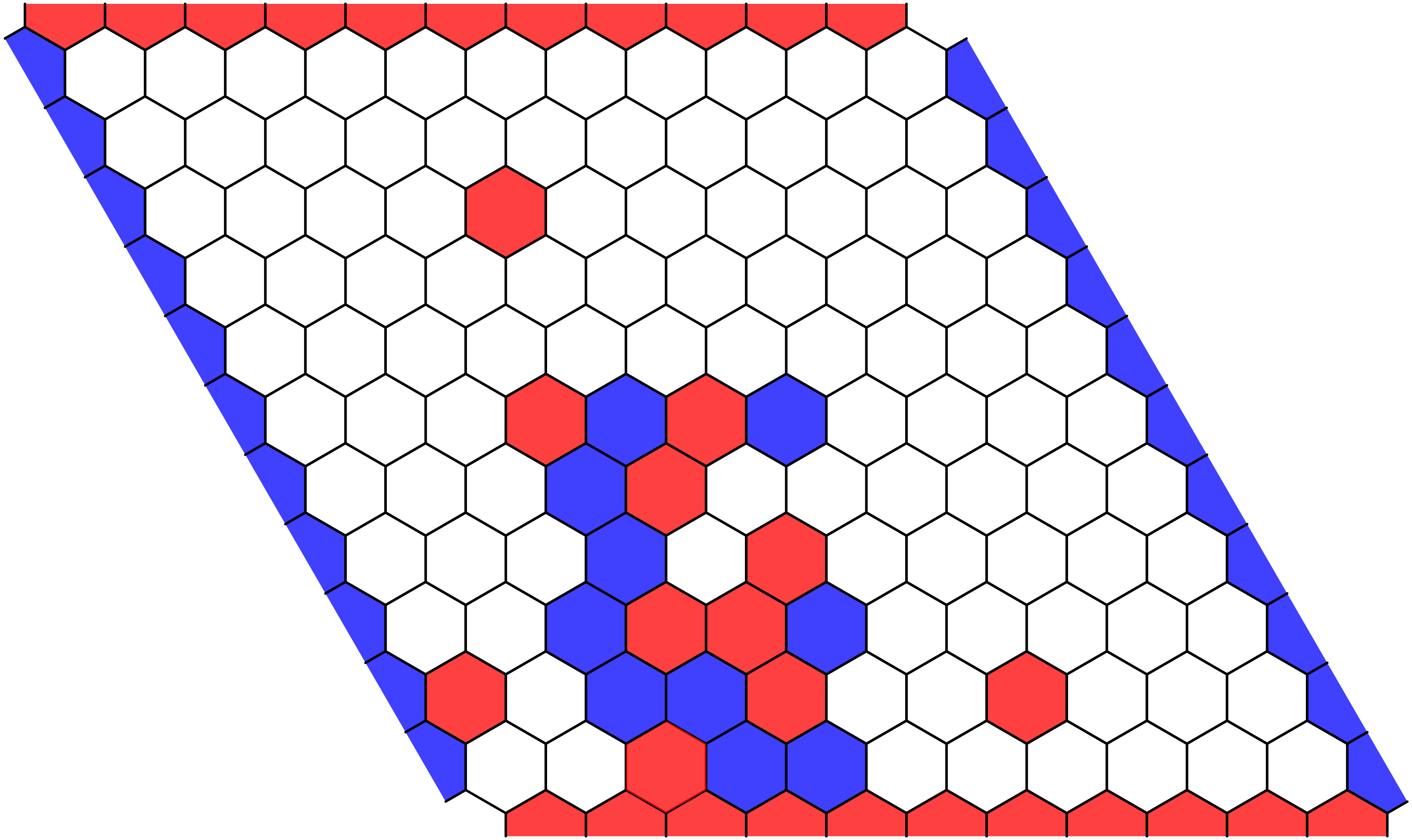




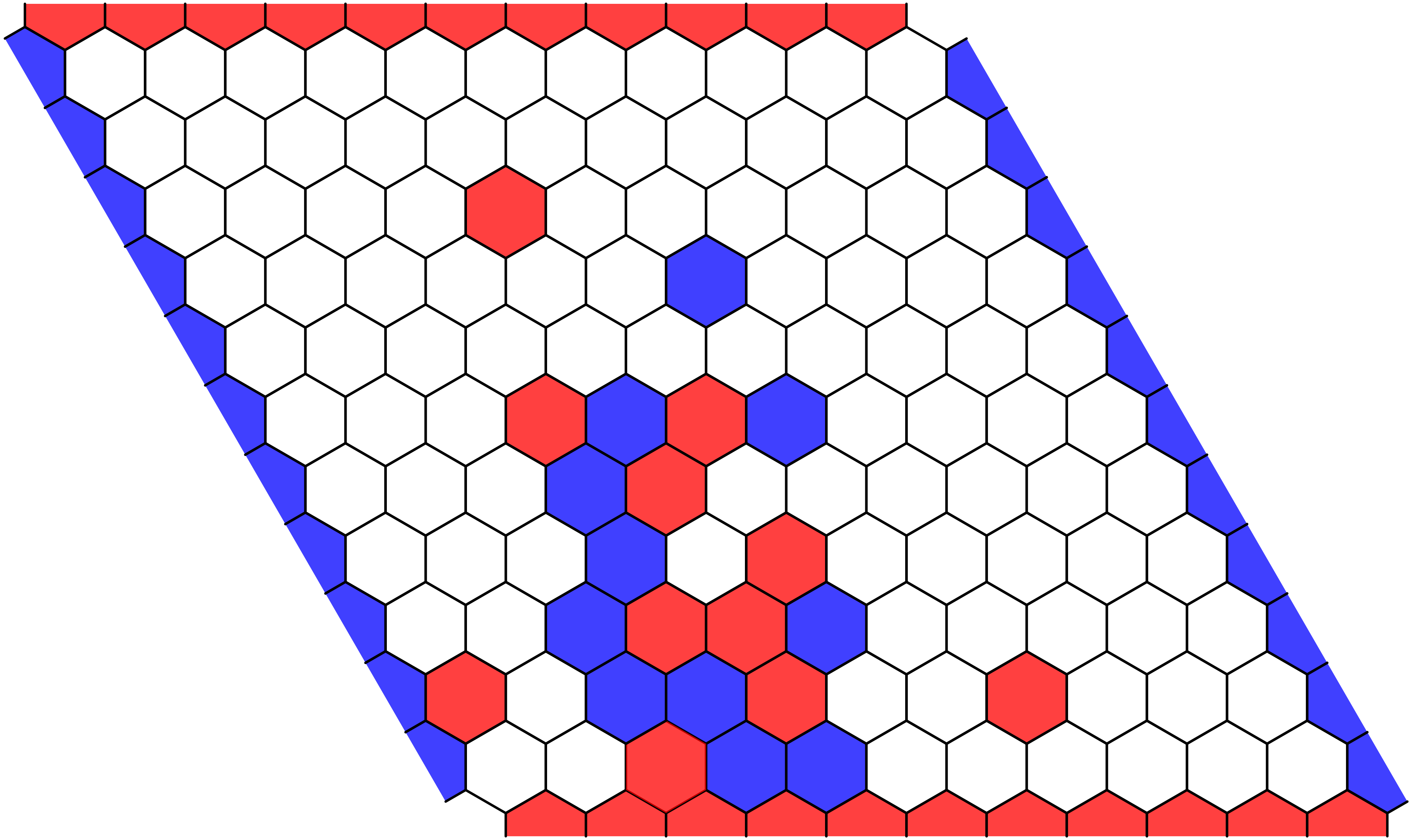
HEX (ヘックス)



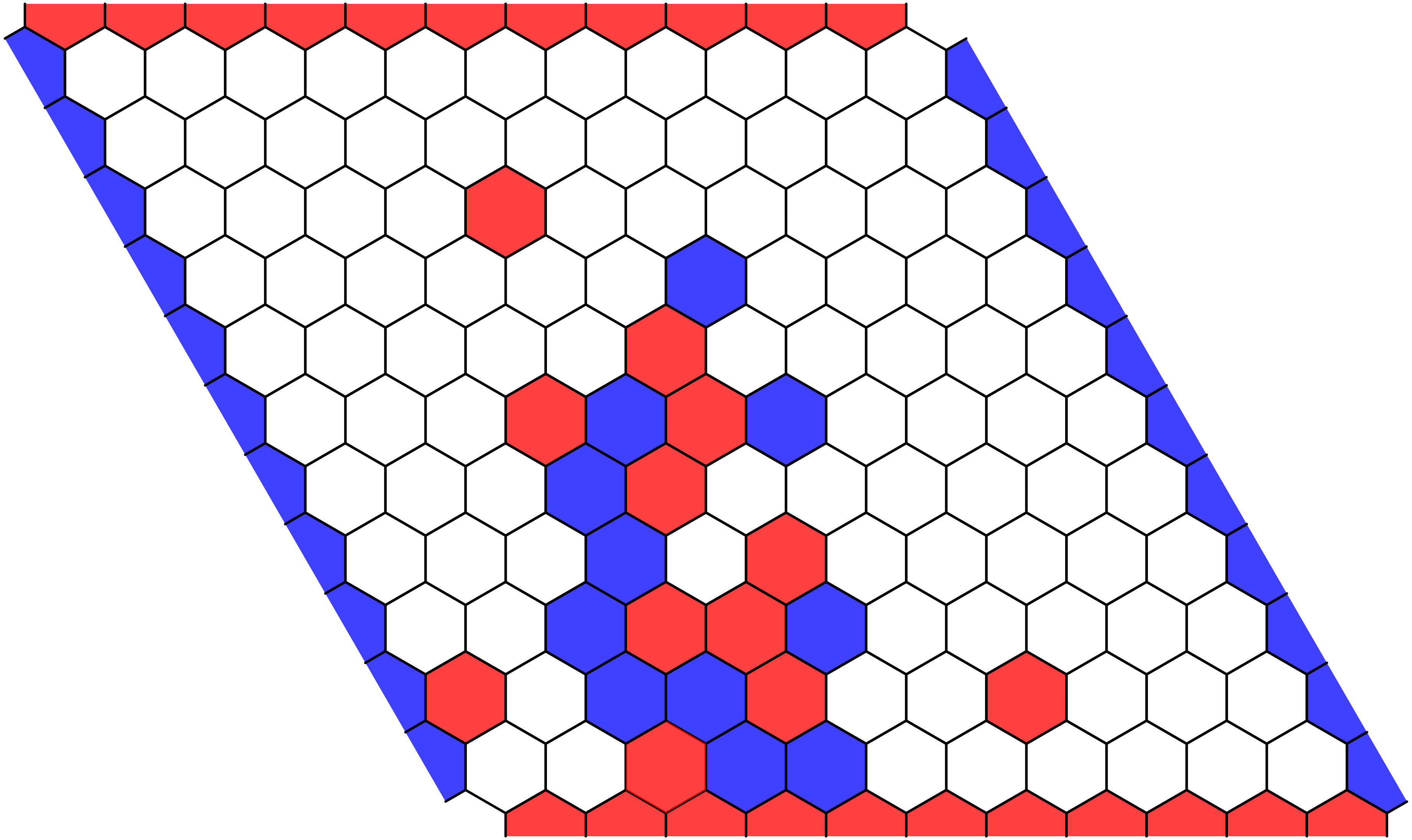
HEX (ヘックス)



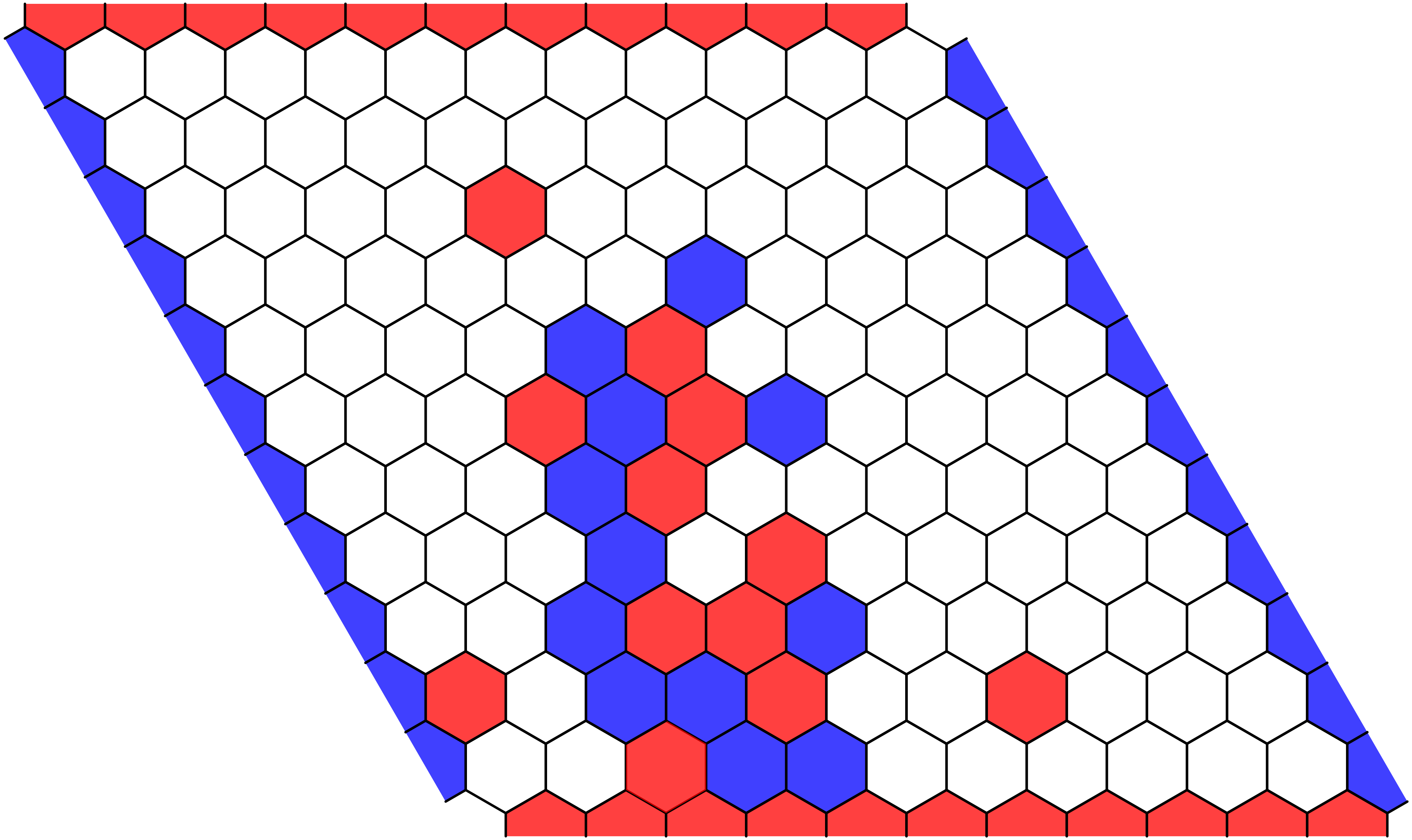
HEX (ヘックス)



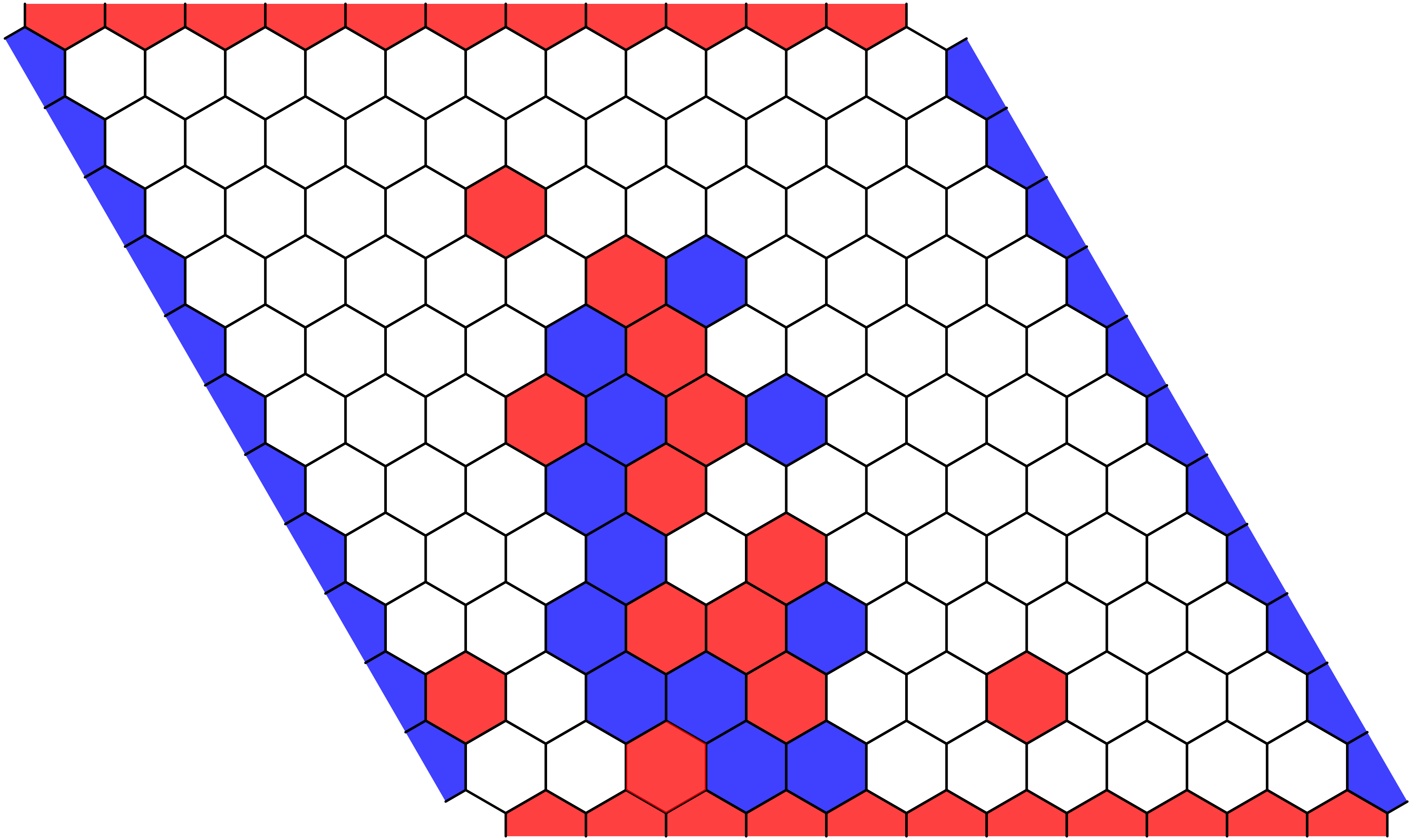
HEX (ヘックス)



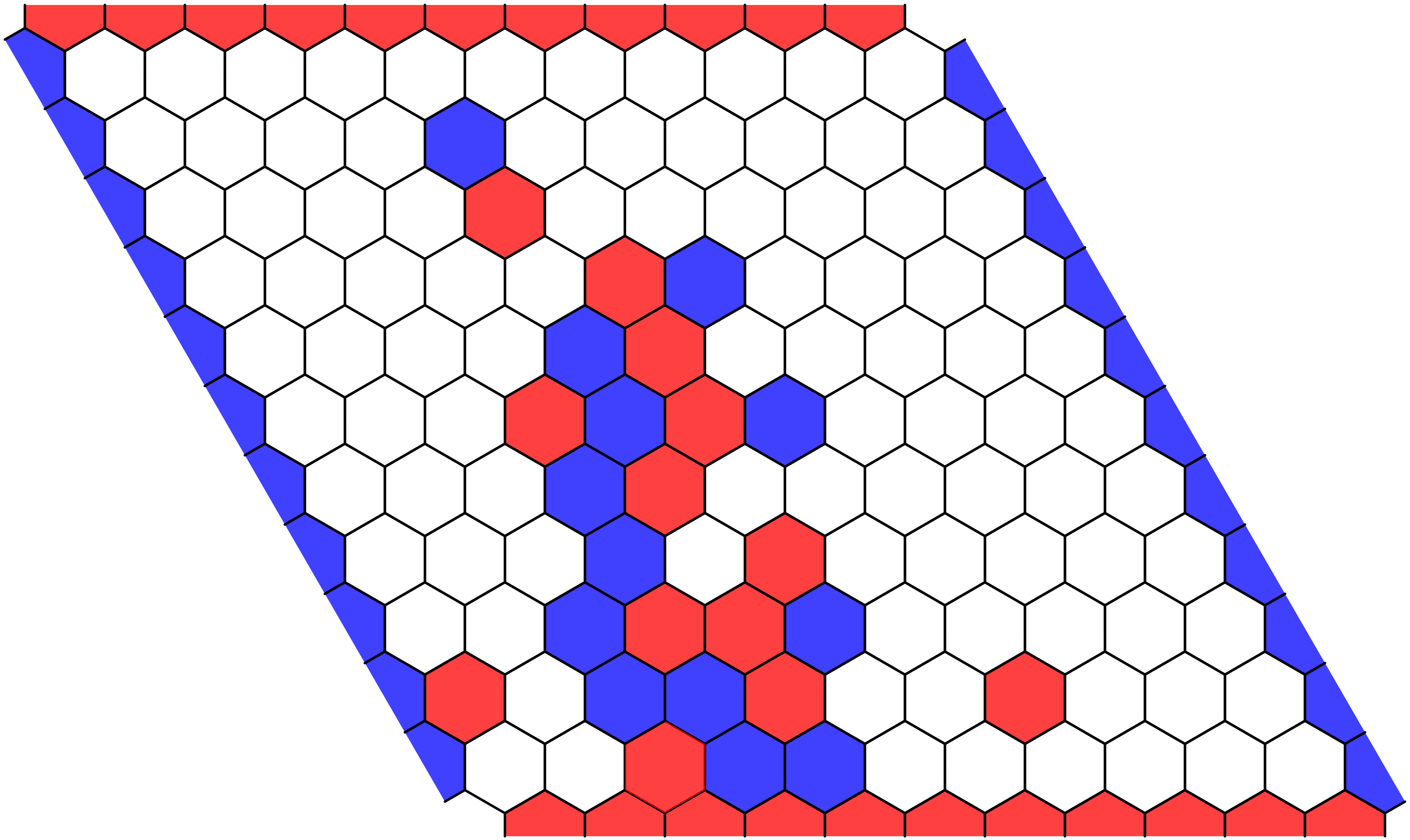
HEX (ヘックス)



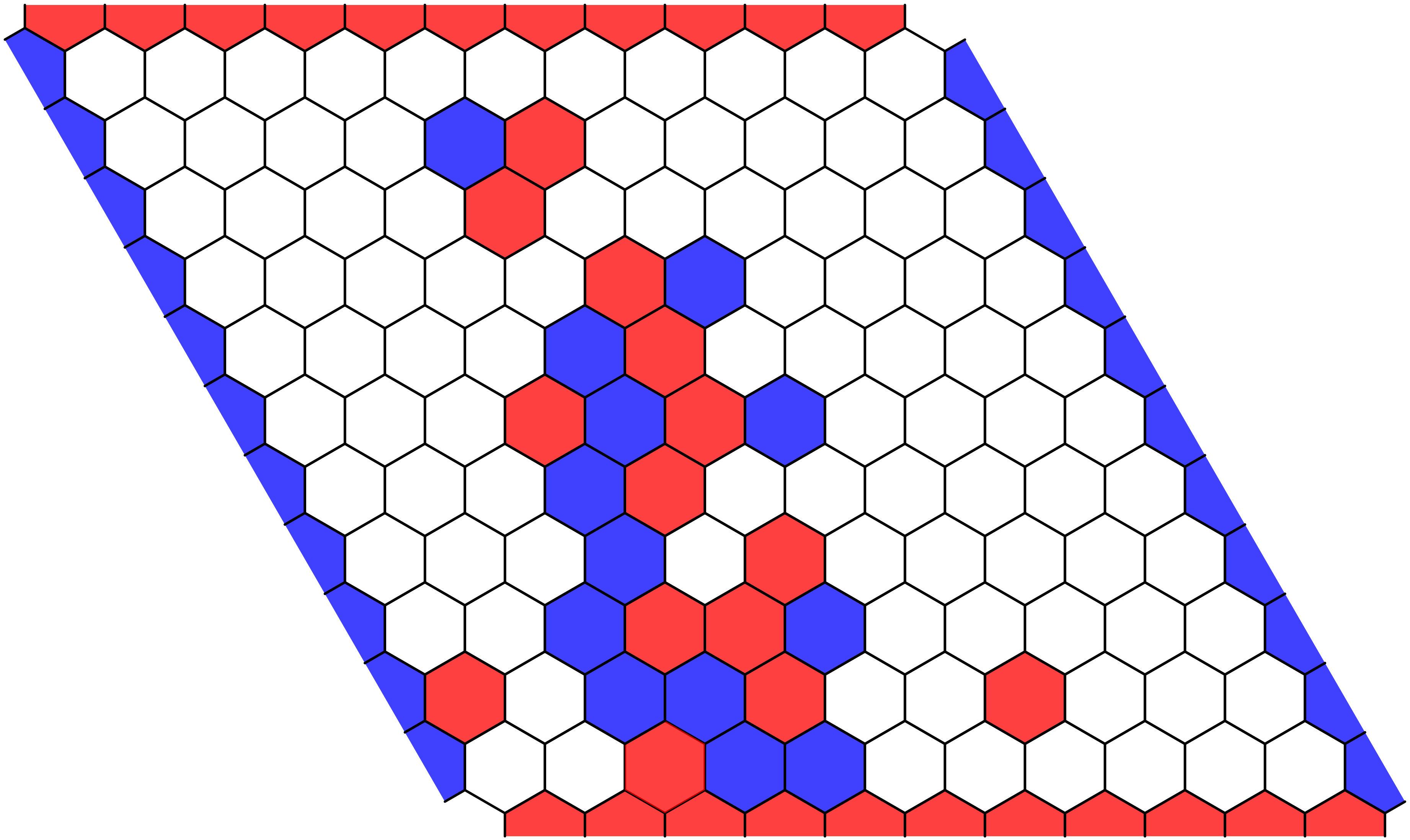
HEX (ヘックス)



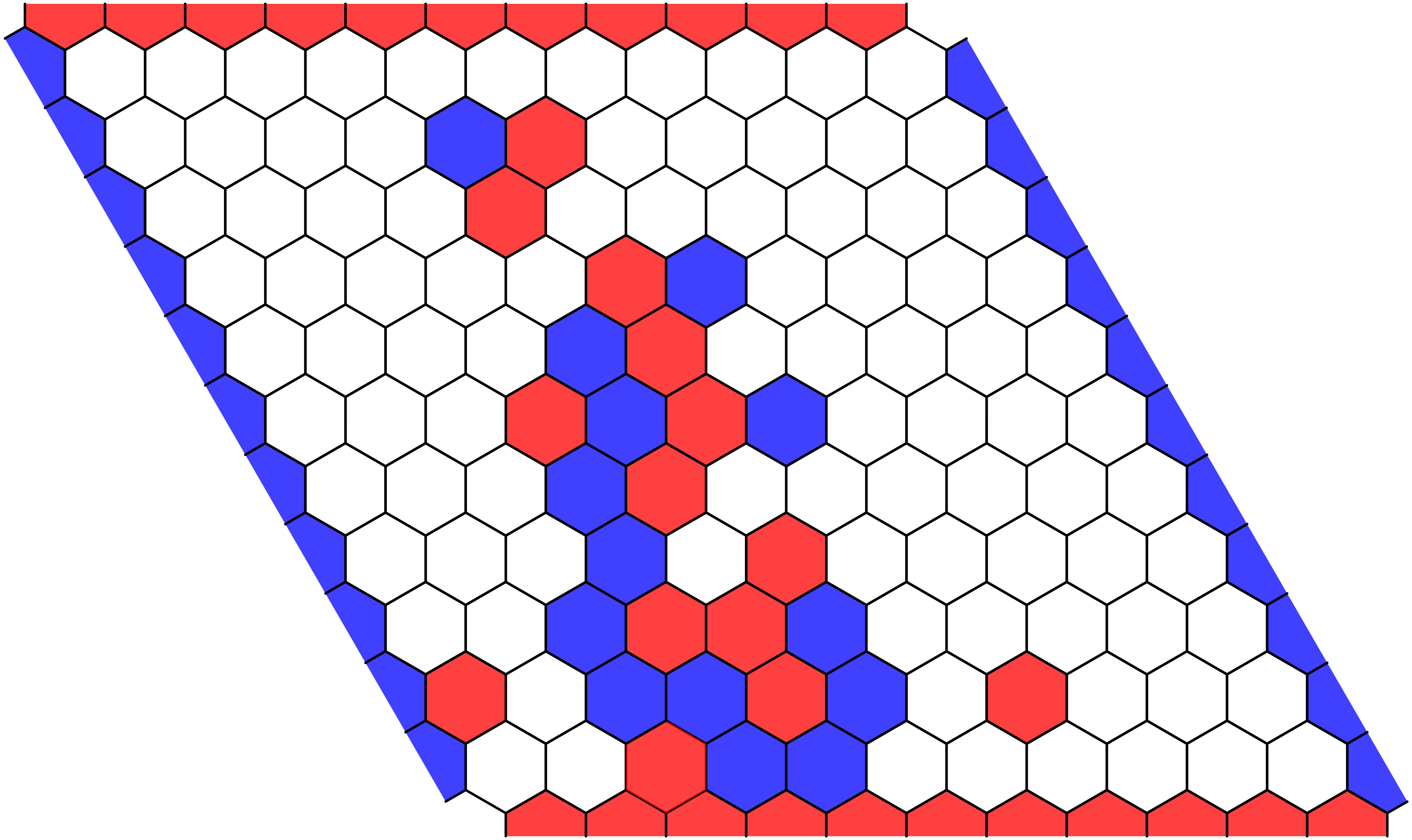
HEX (ヘックス)



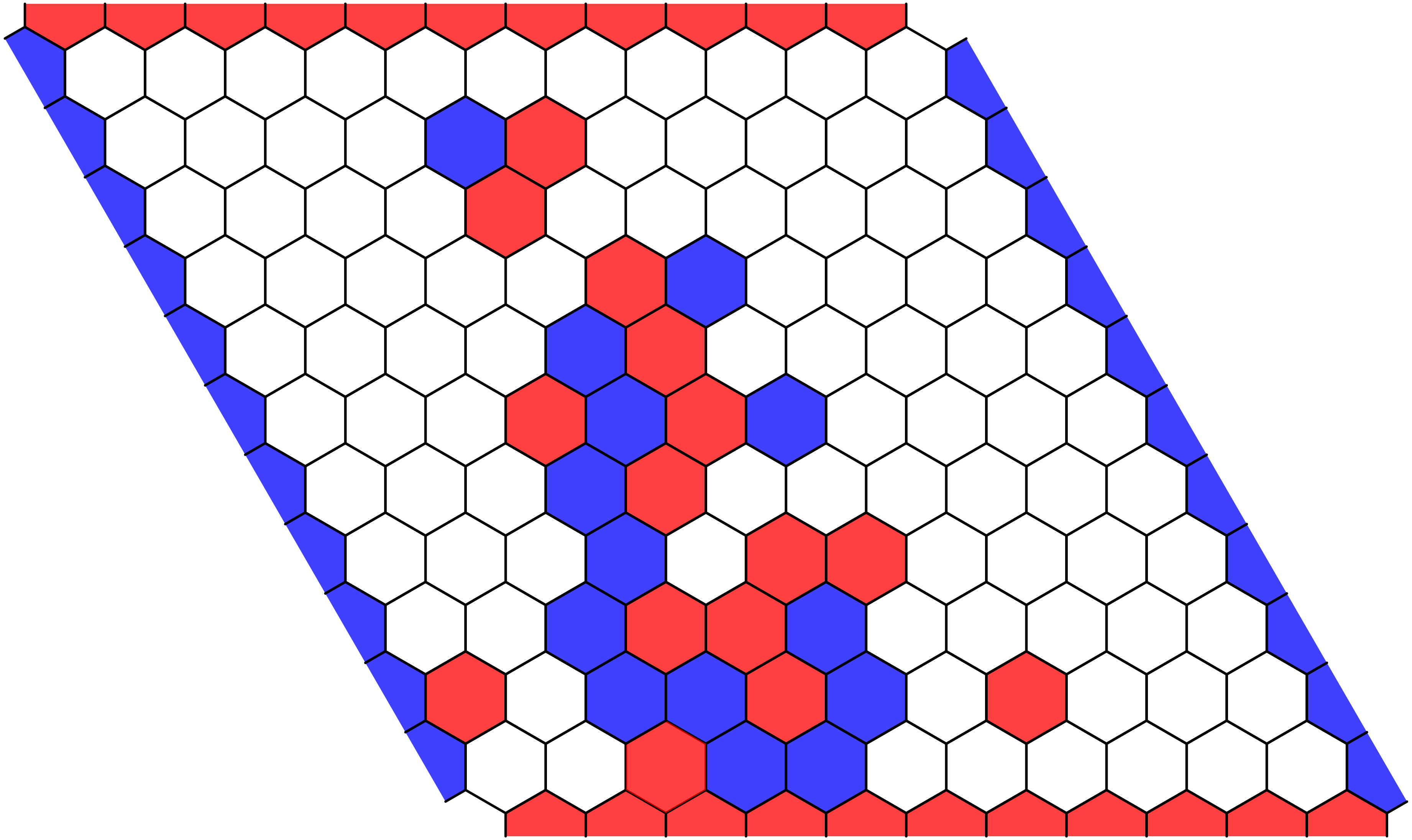
HEX (ヘックス)



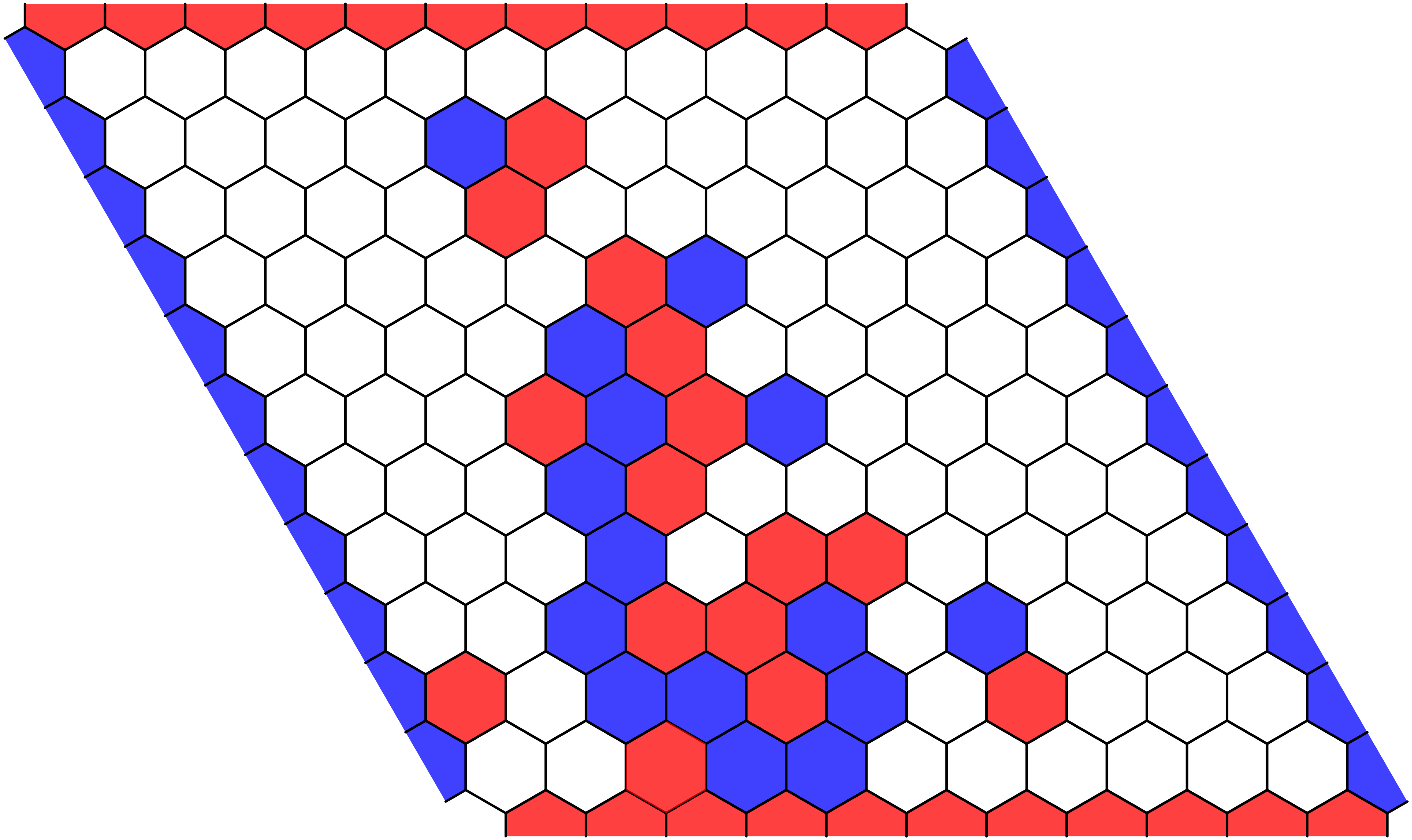
HEX (ヘックス)



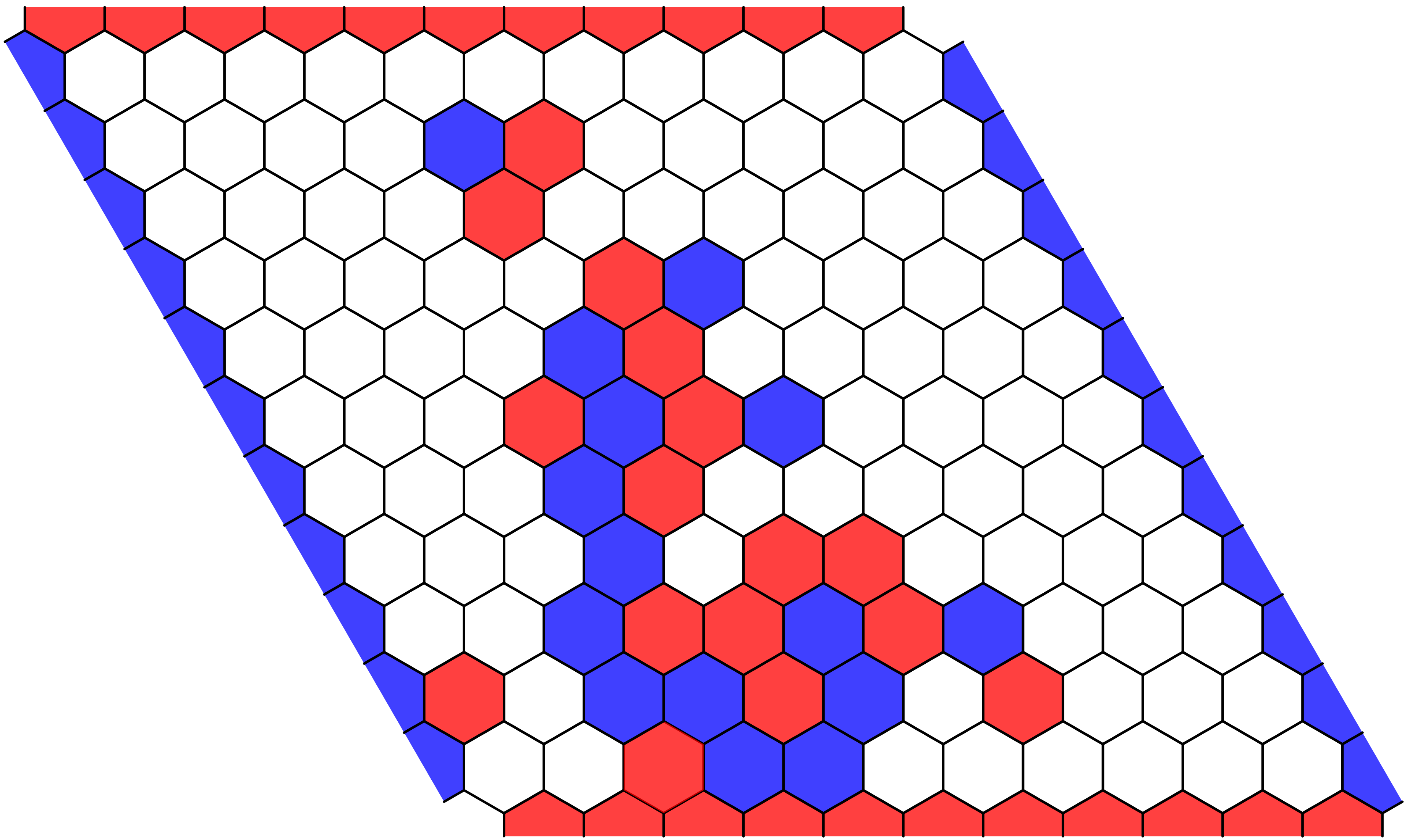
HEX (ヘックス)



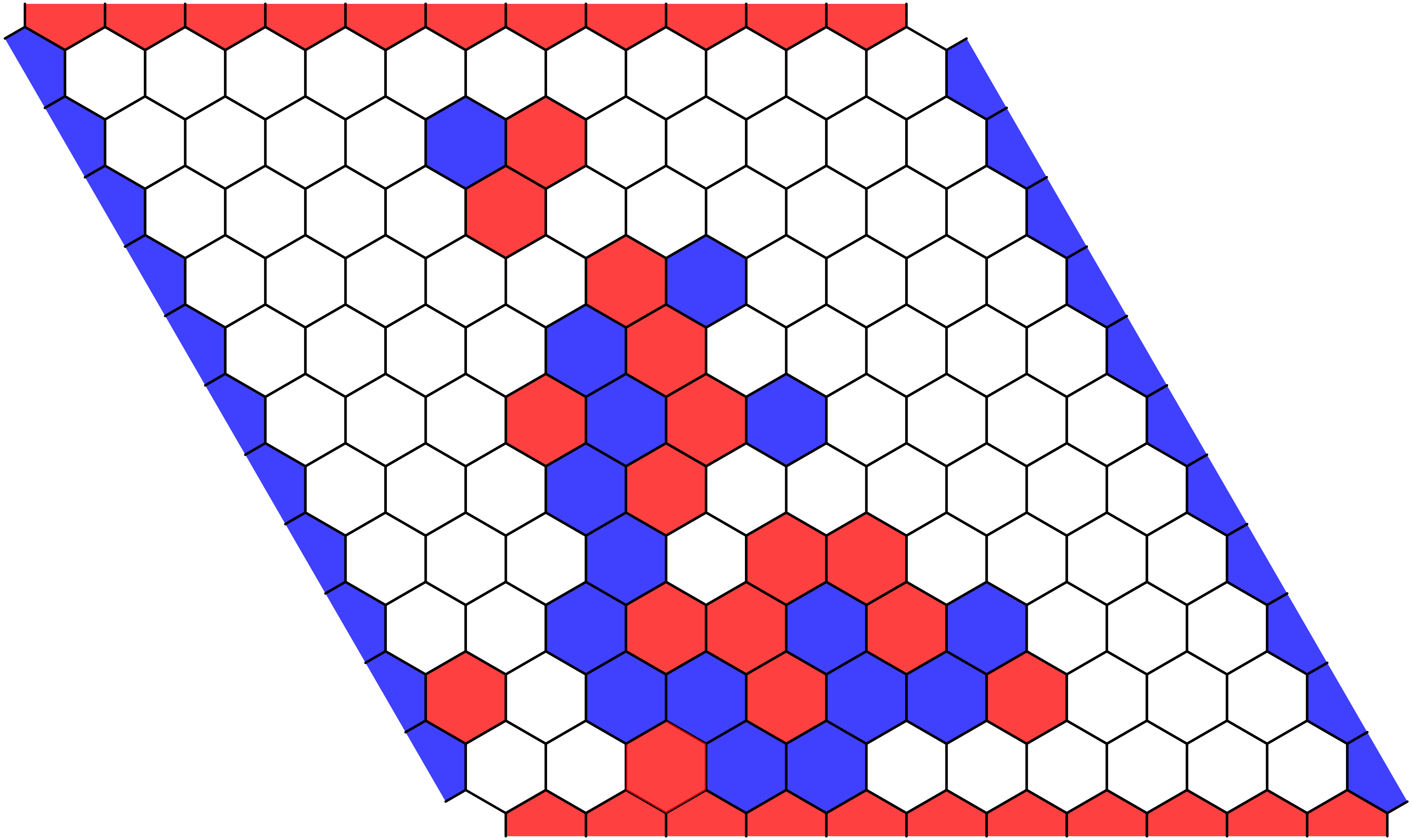
HEX (ヘックス)



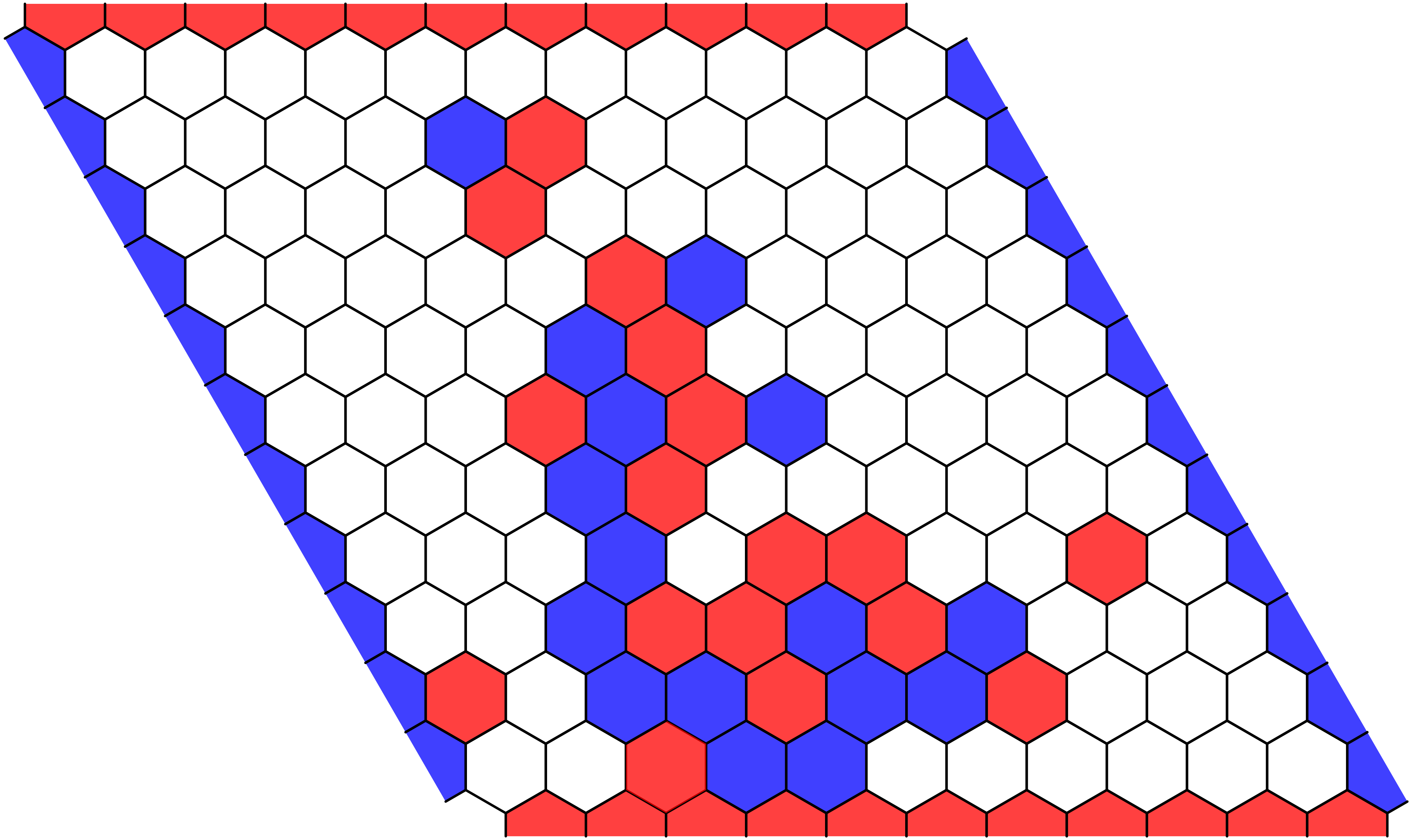
HEX (ヘックス)



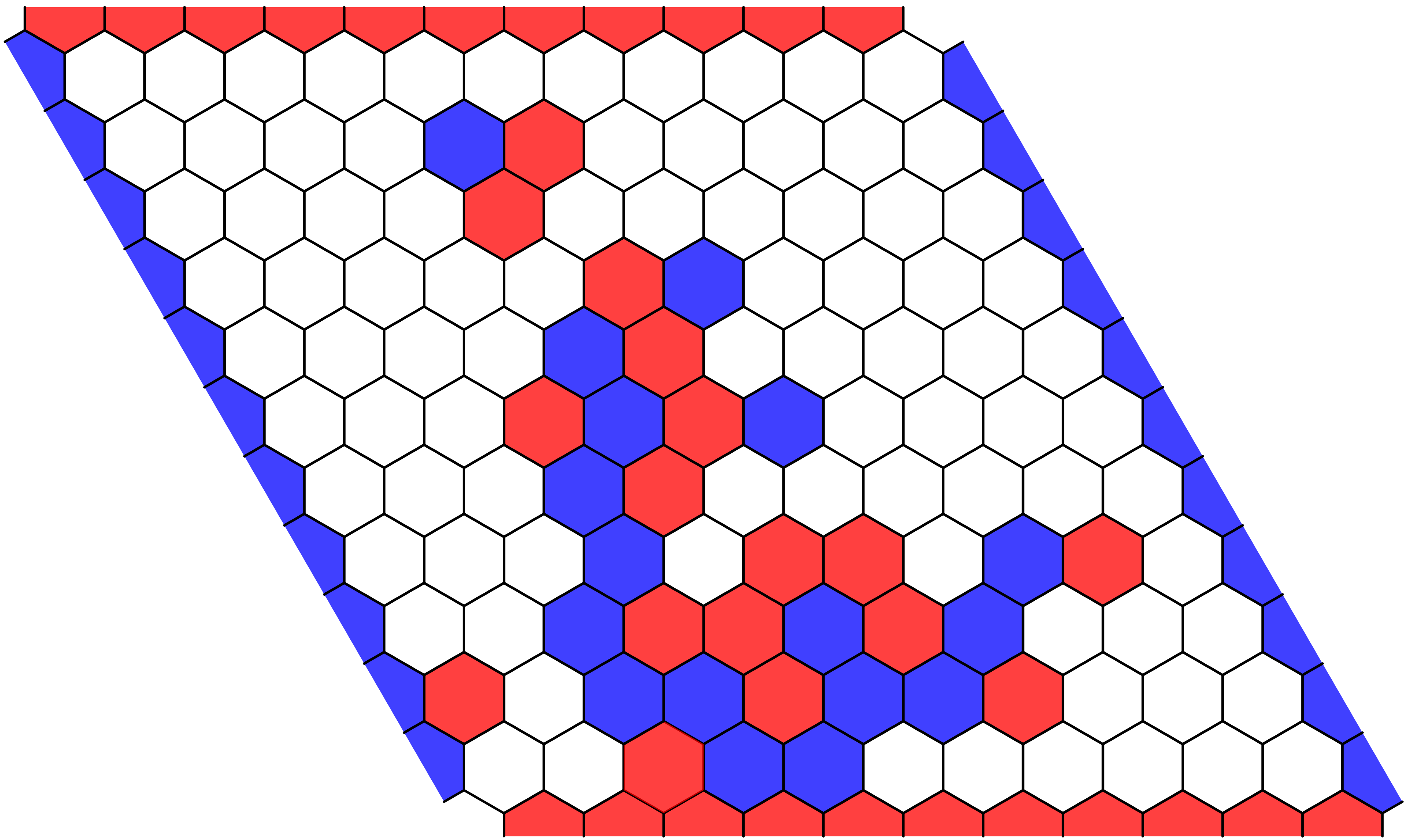
HEX (ヘックス)



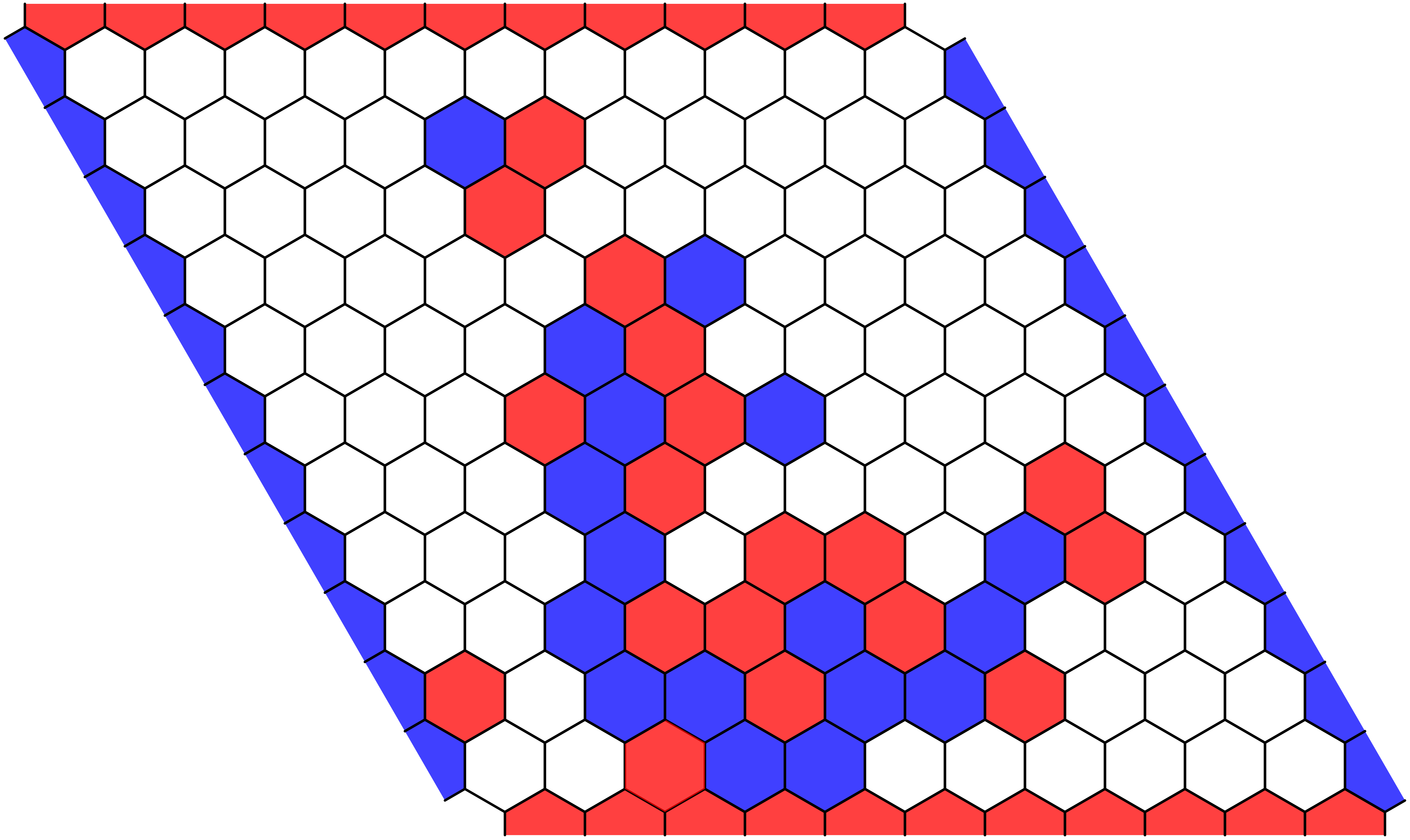
HEX (ヘックス)



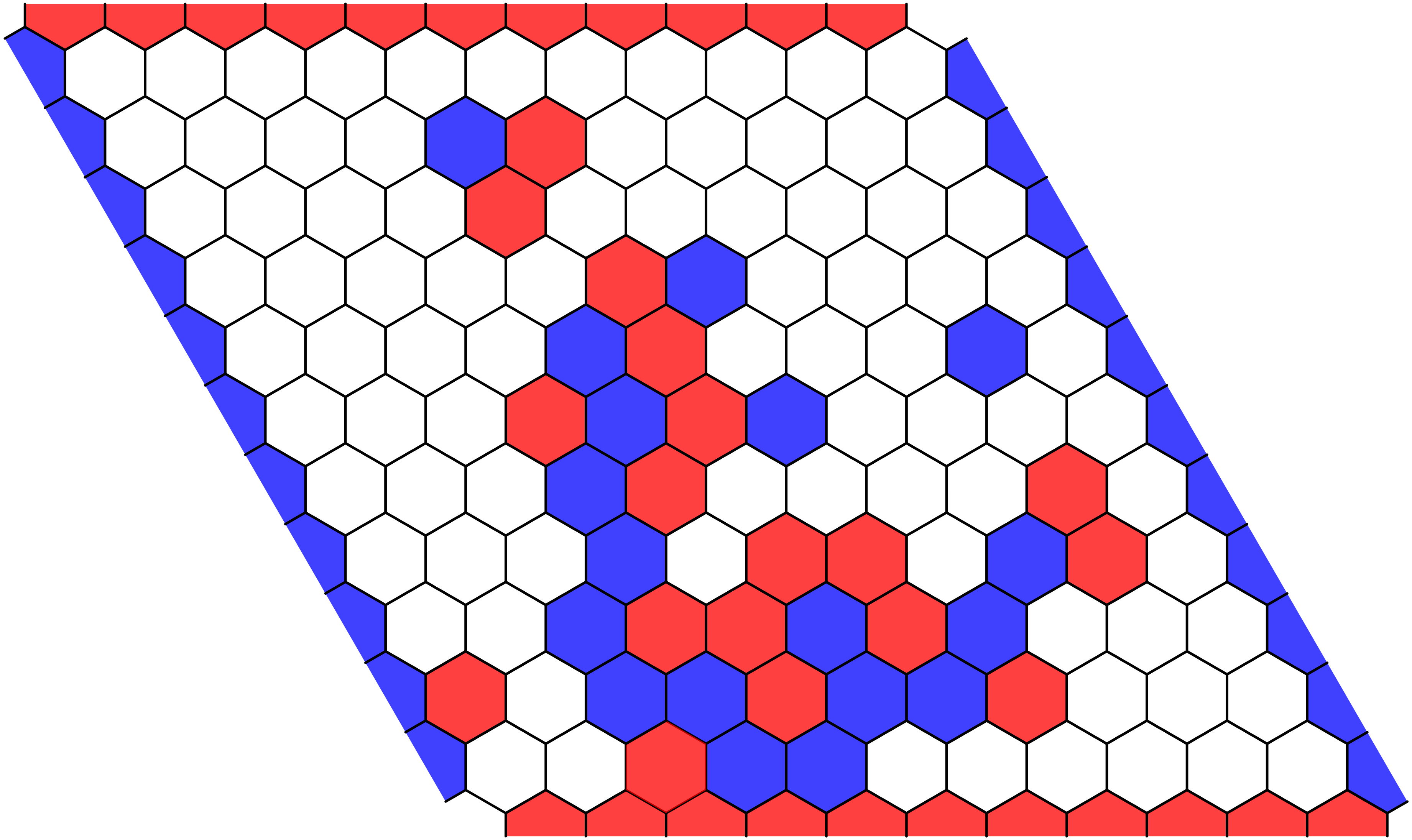
HEX (ヘックス)



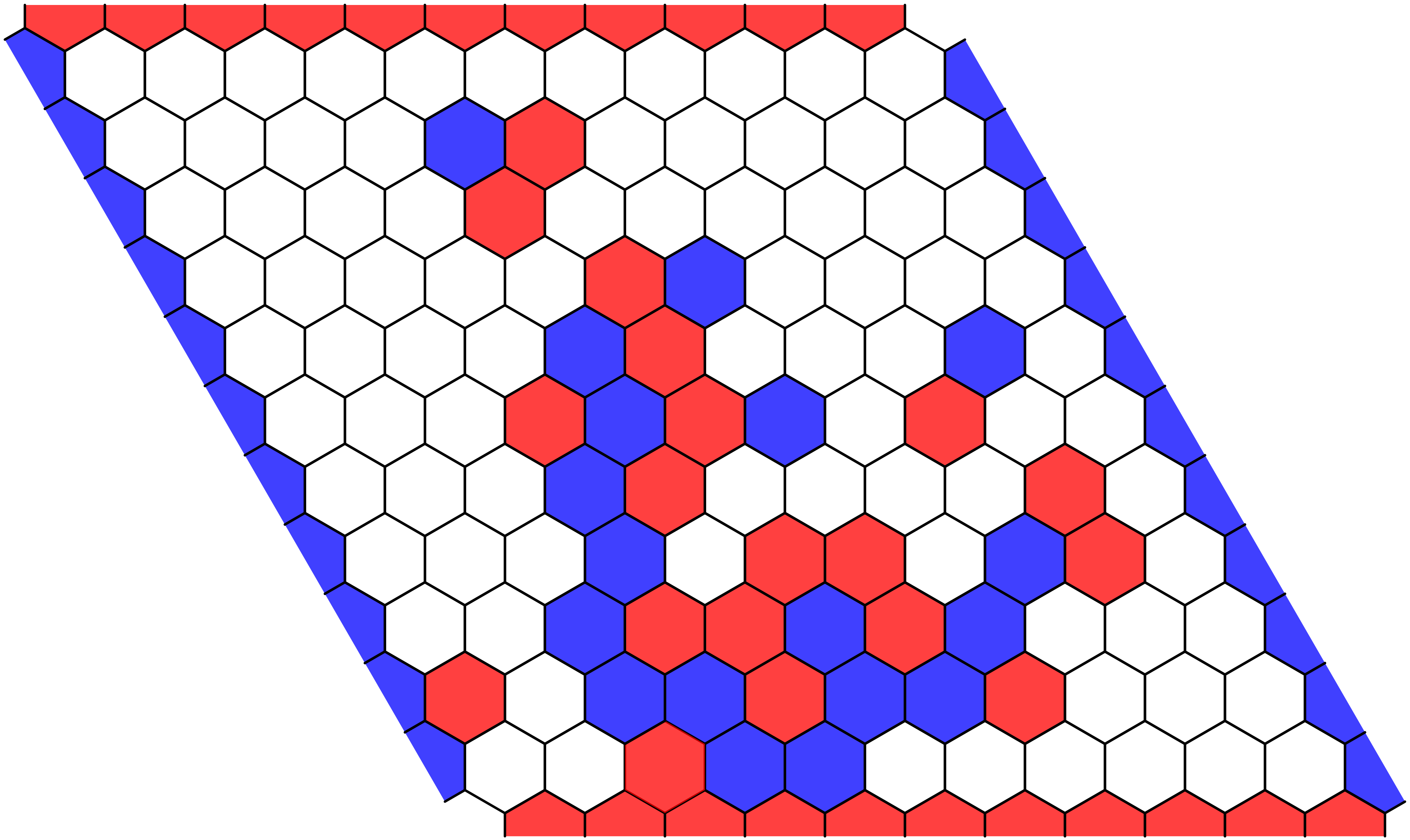
HEX (ヘックス)



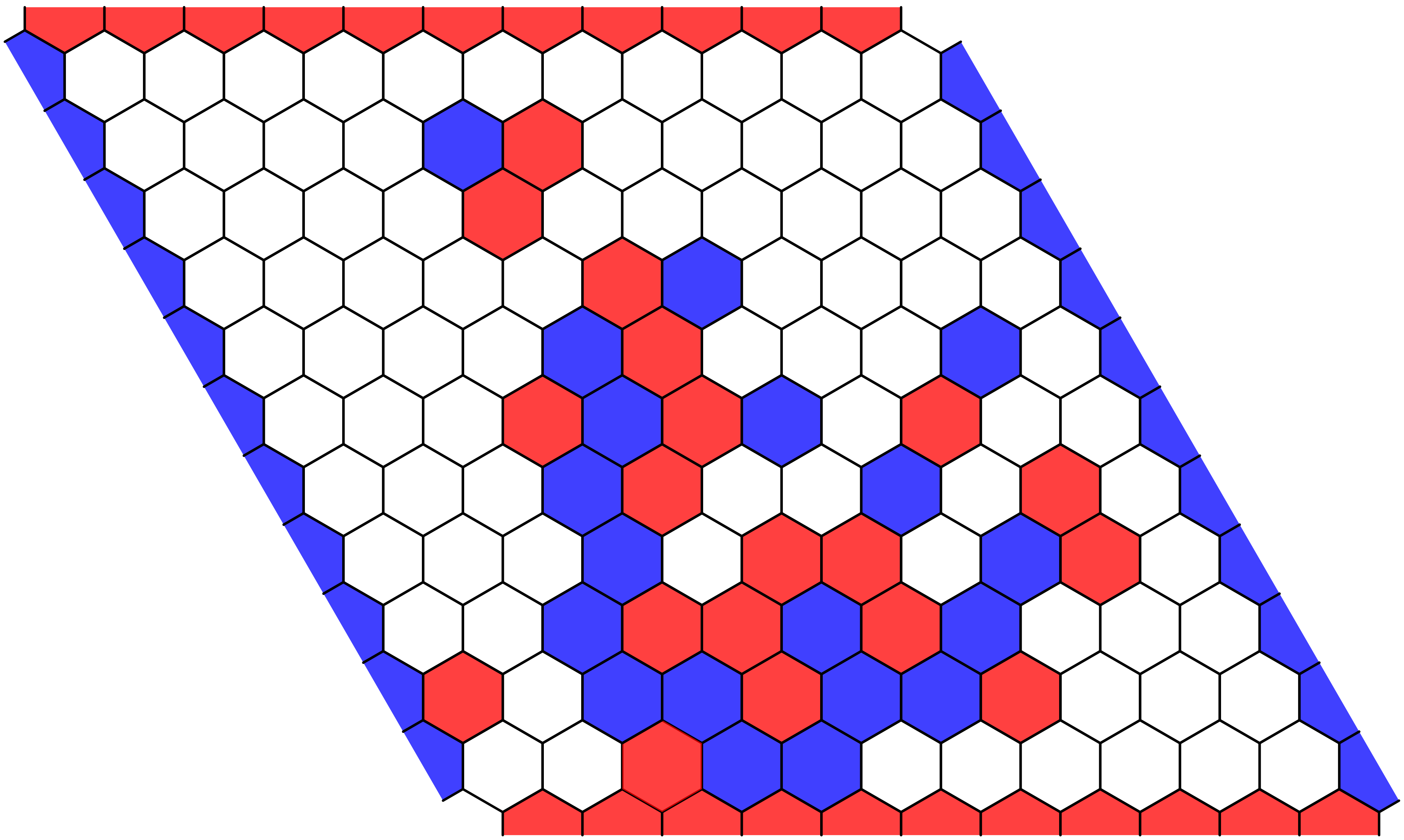
HEX (ヘックス)



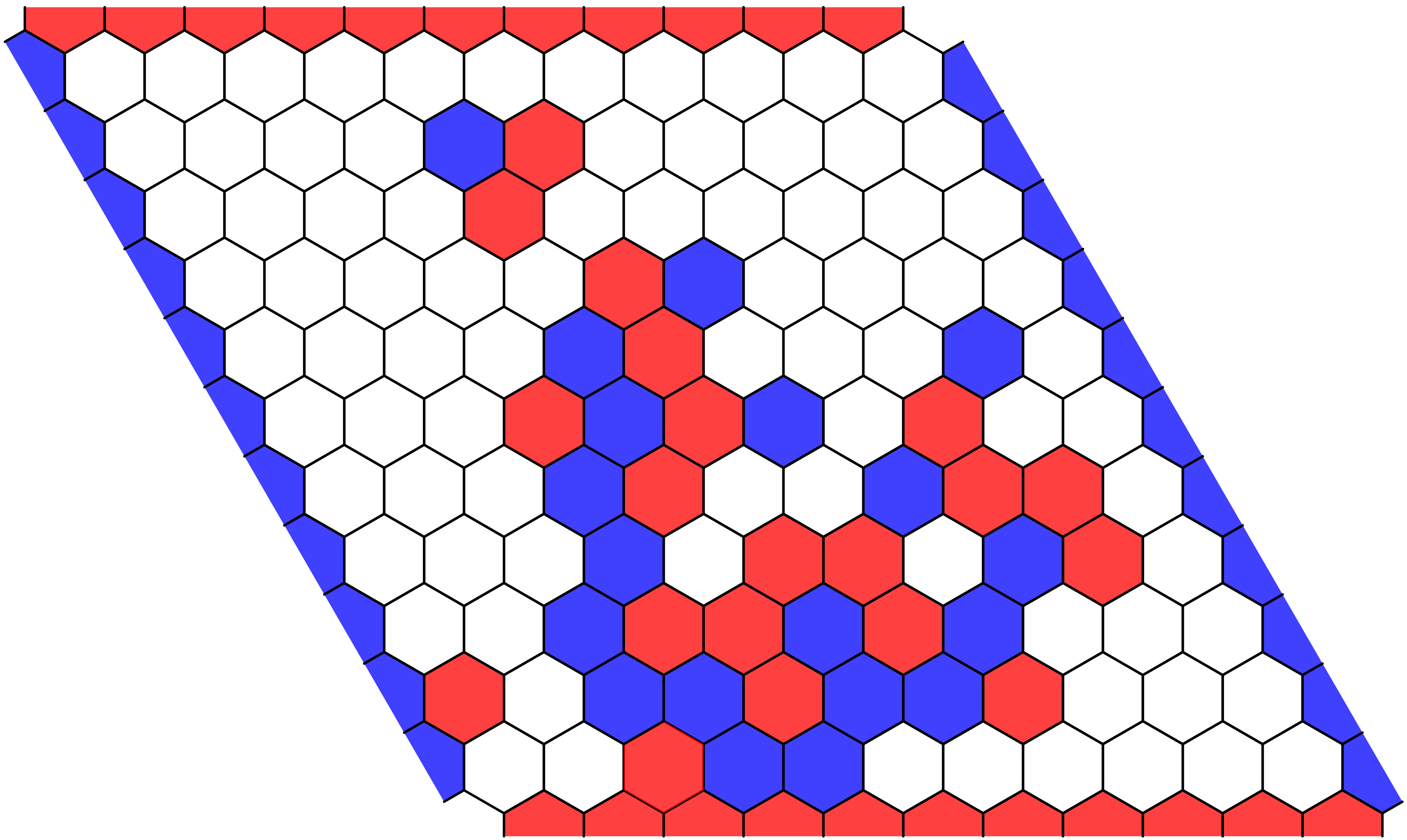
HEX (ヘックス)



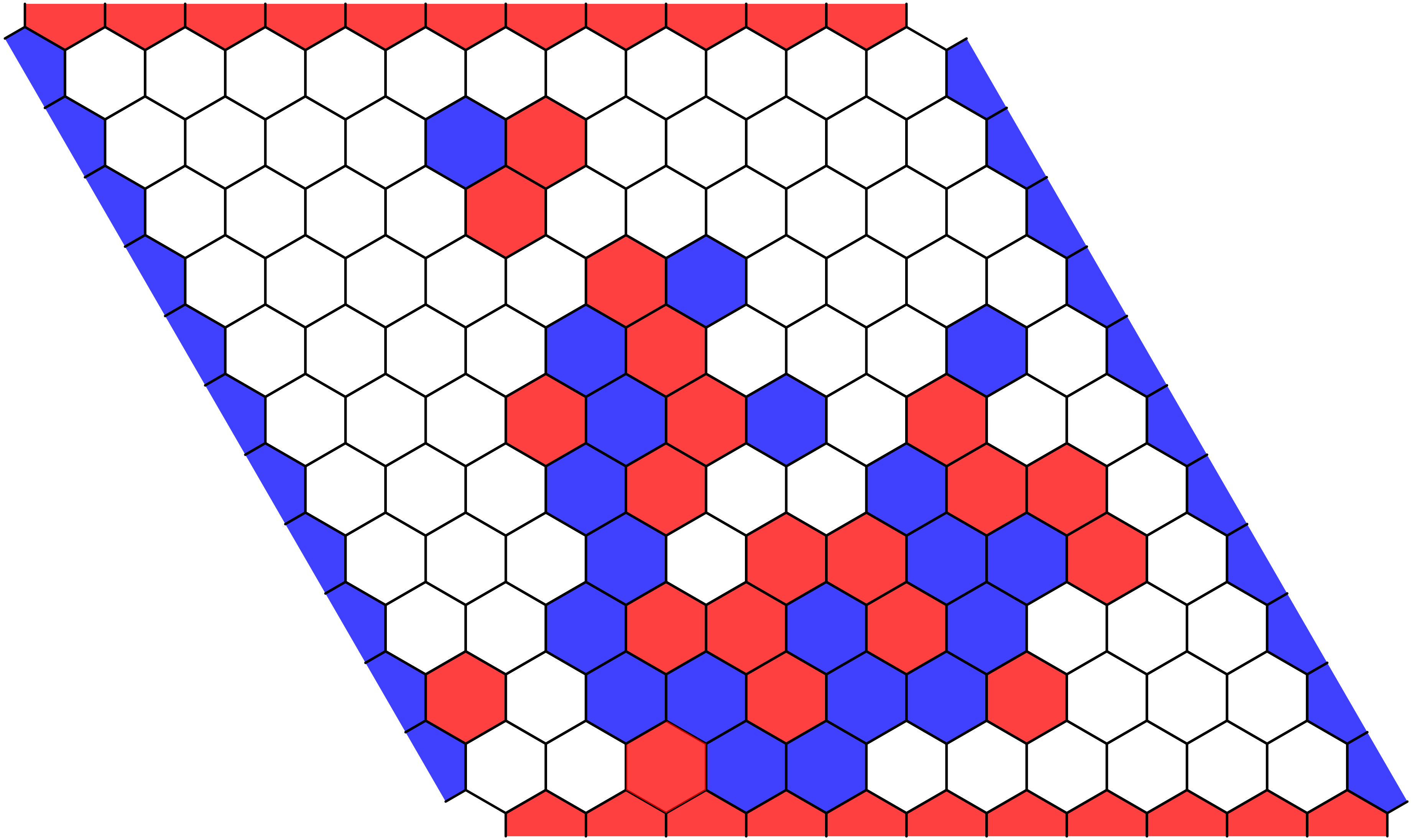
HEX (ヘックス)



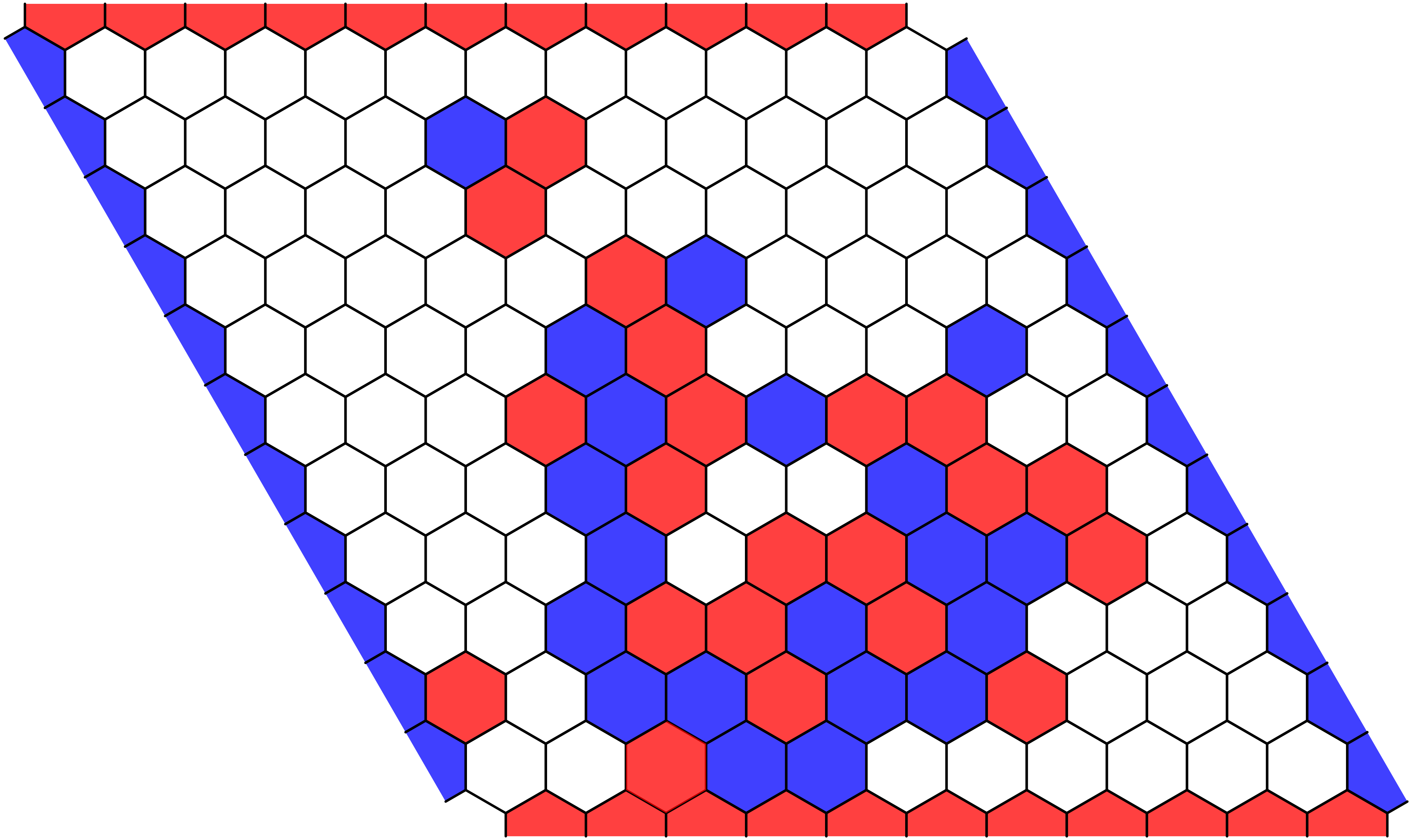
HEX (ヘックス)



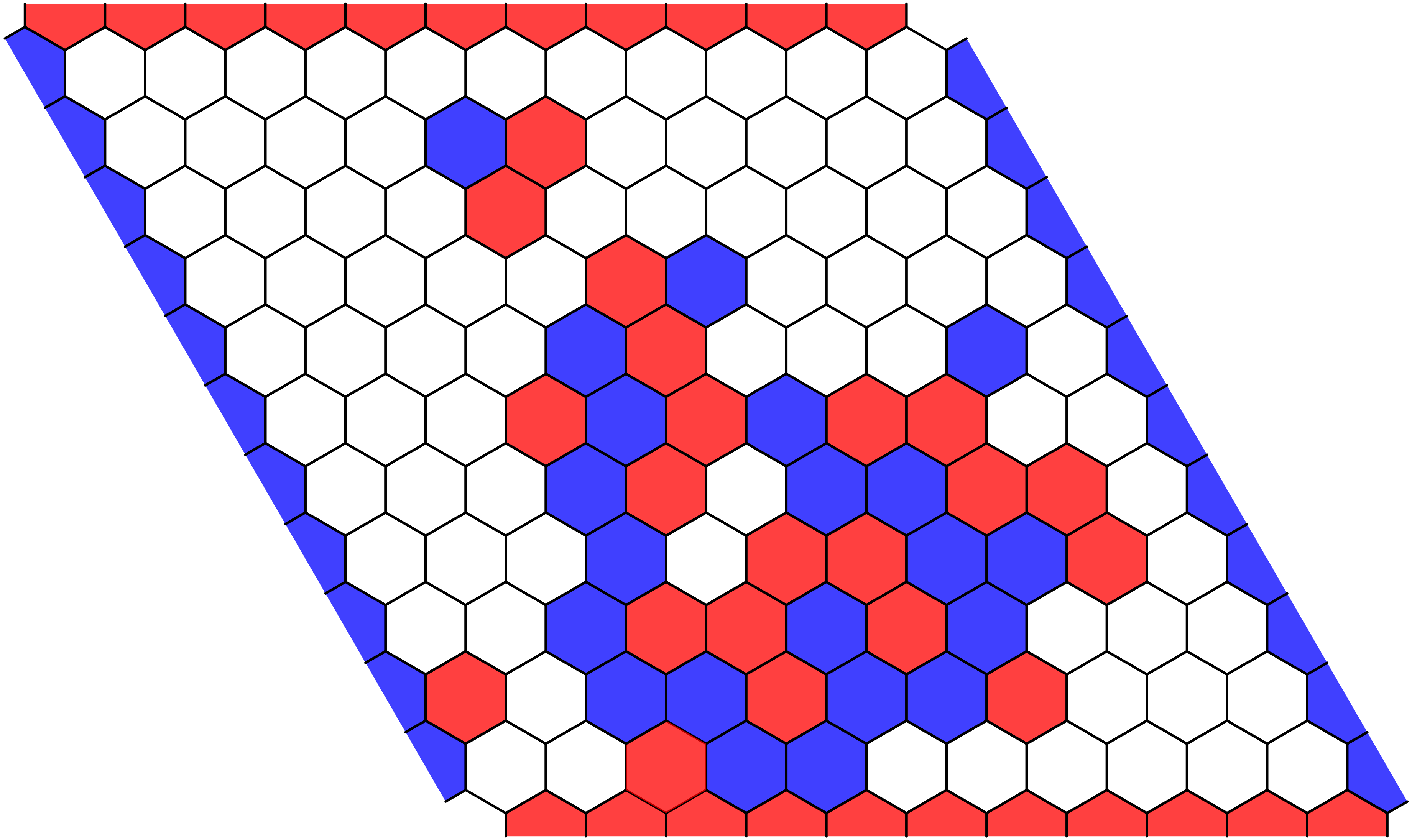
HEX (ヘックス)



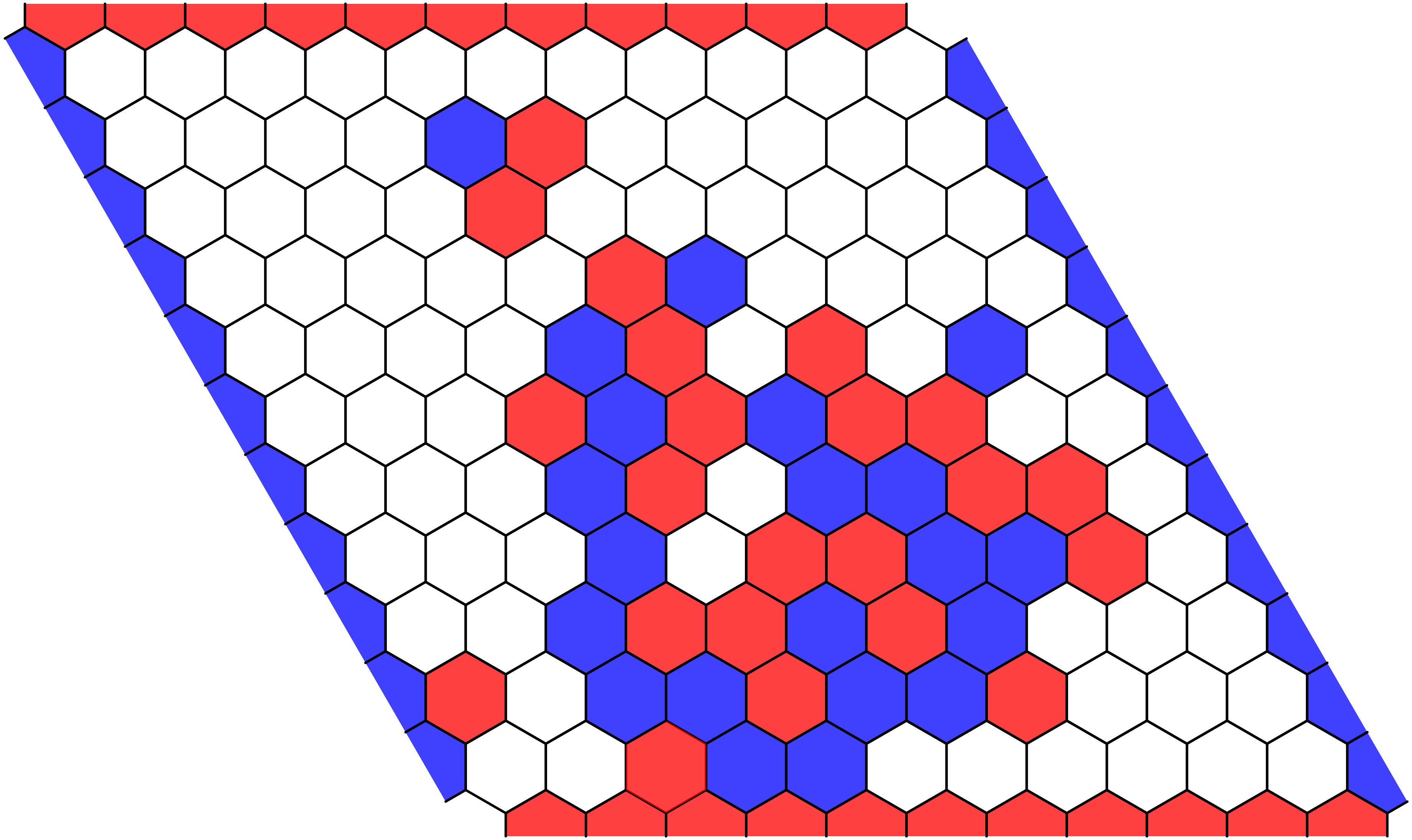
HEX (ヘックス)



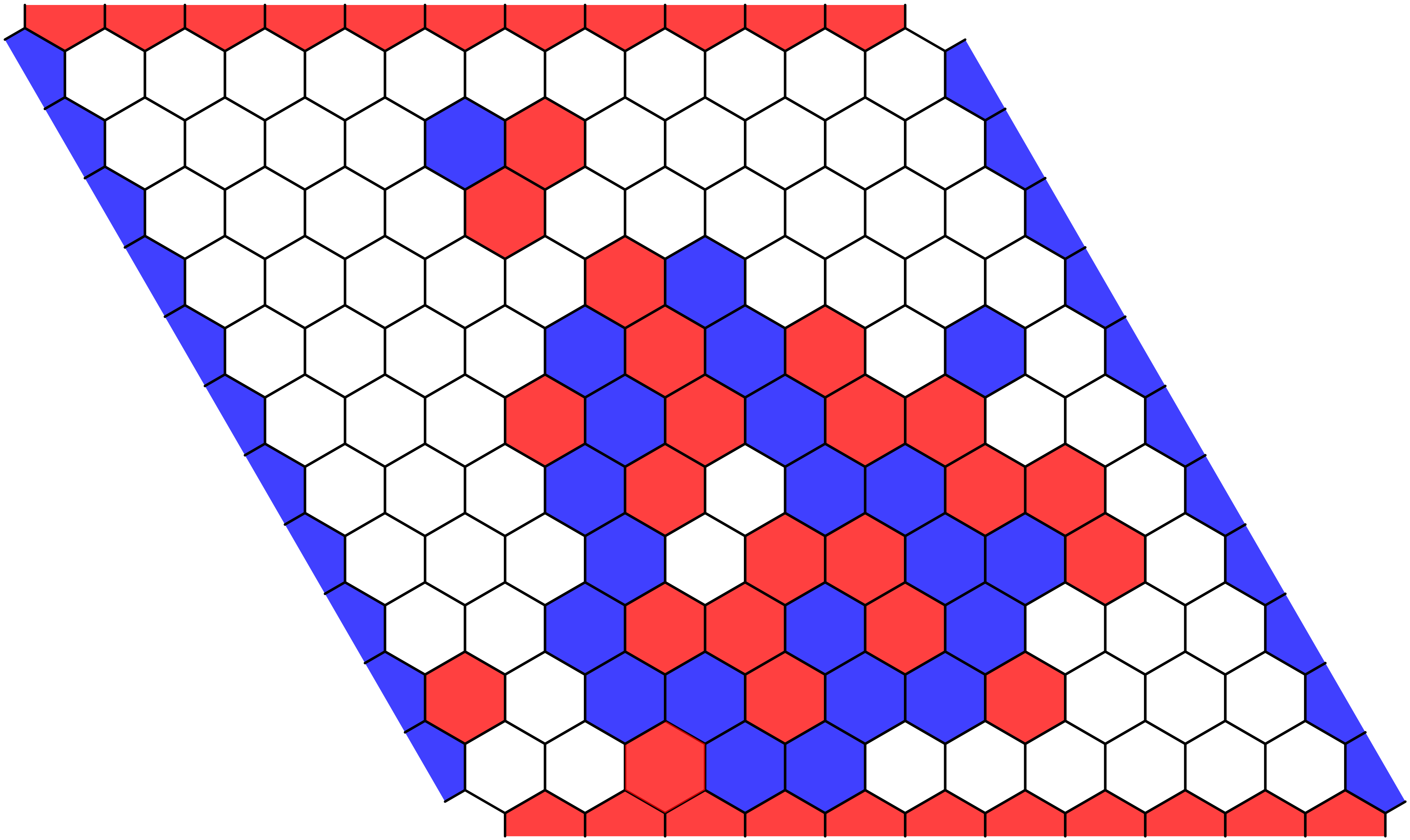
HEX (ヘックス)



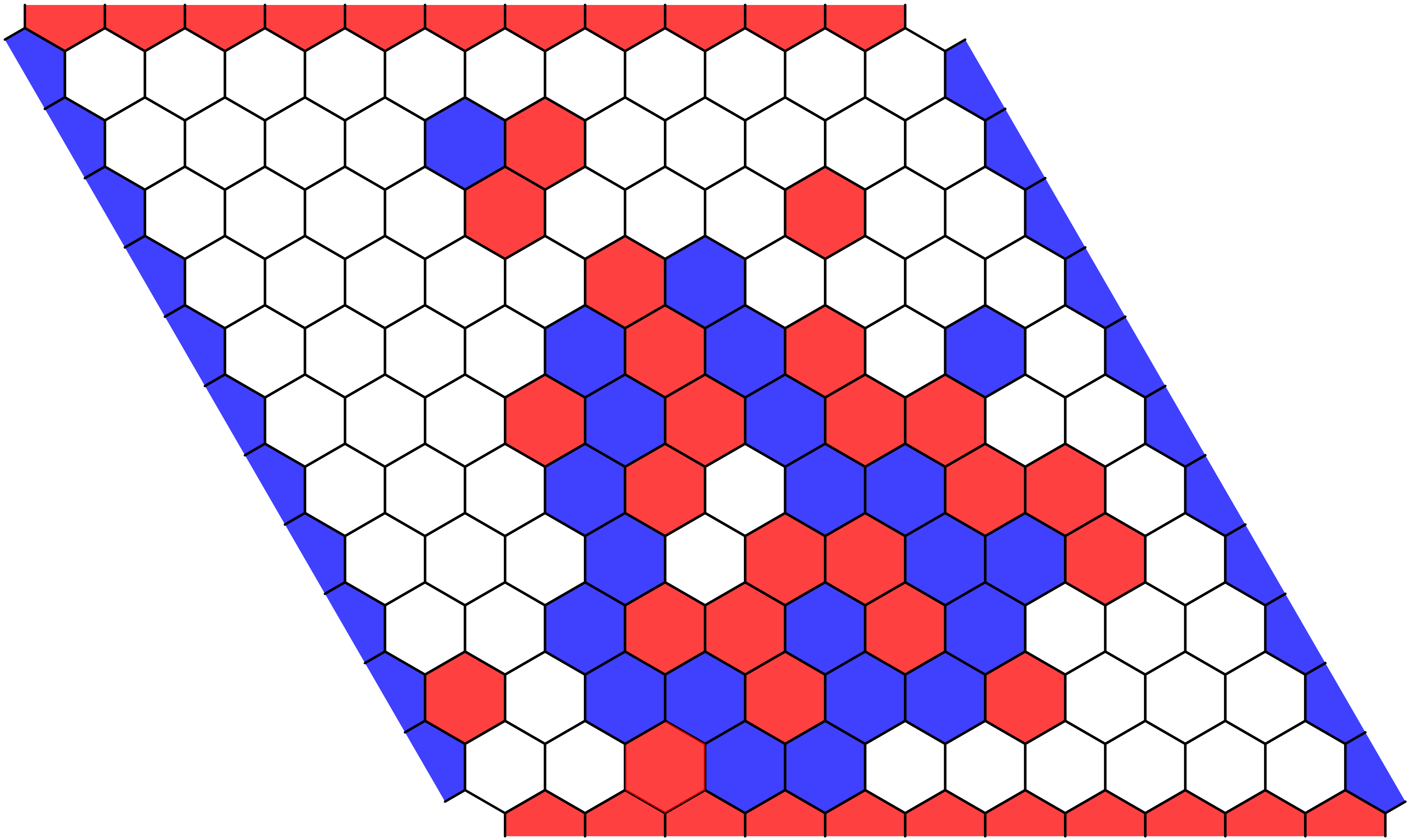
HEX (ヘックス)



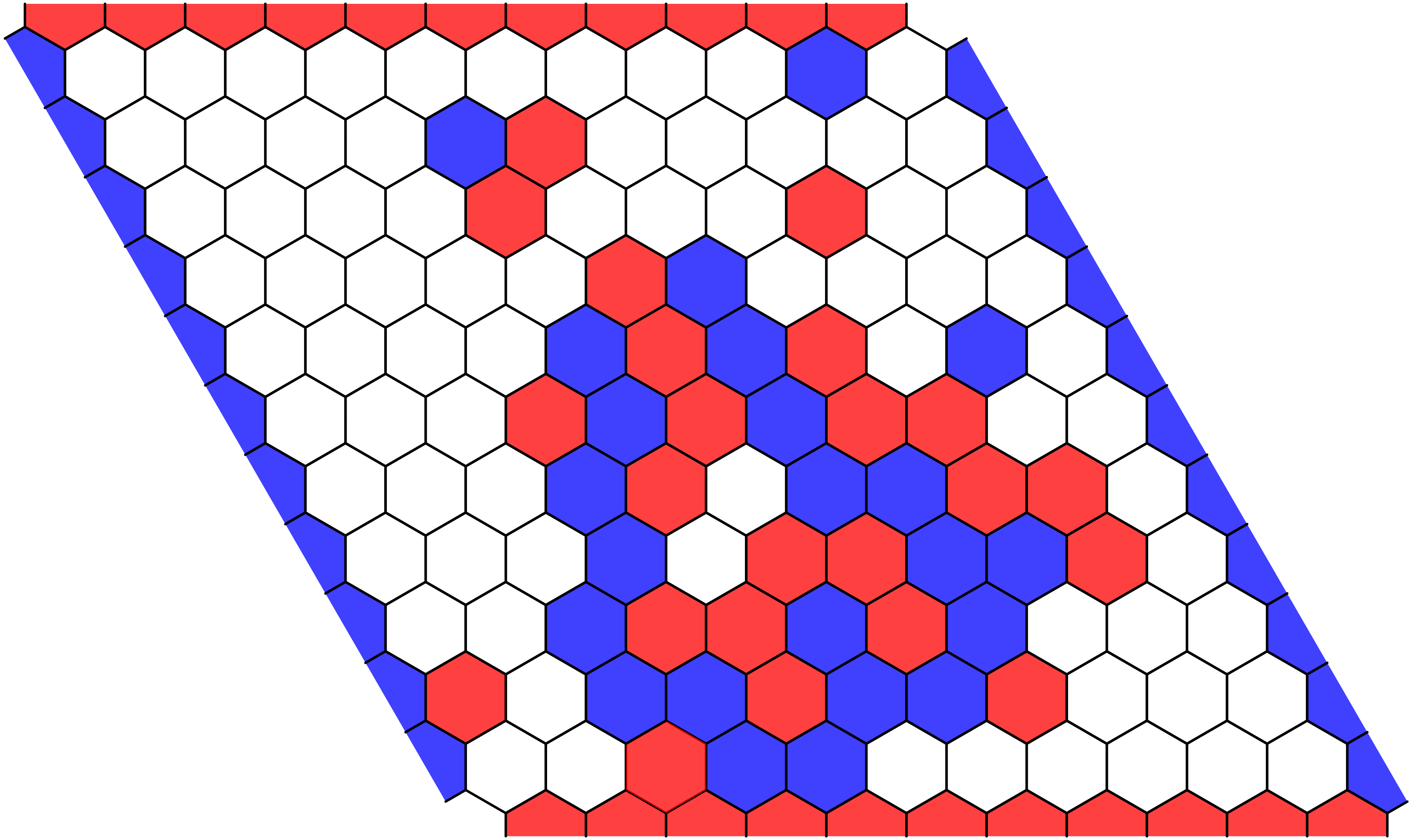
HEX (ヘックス)

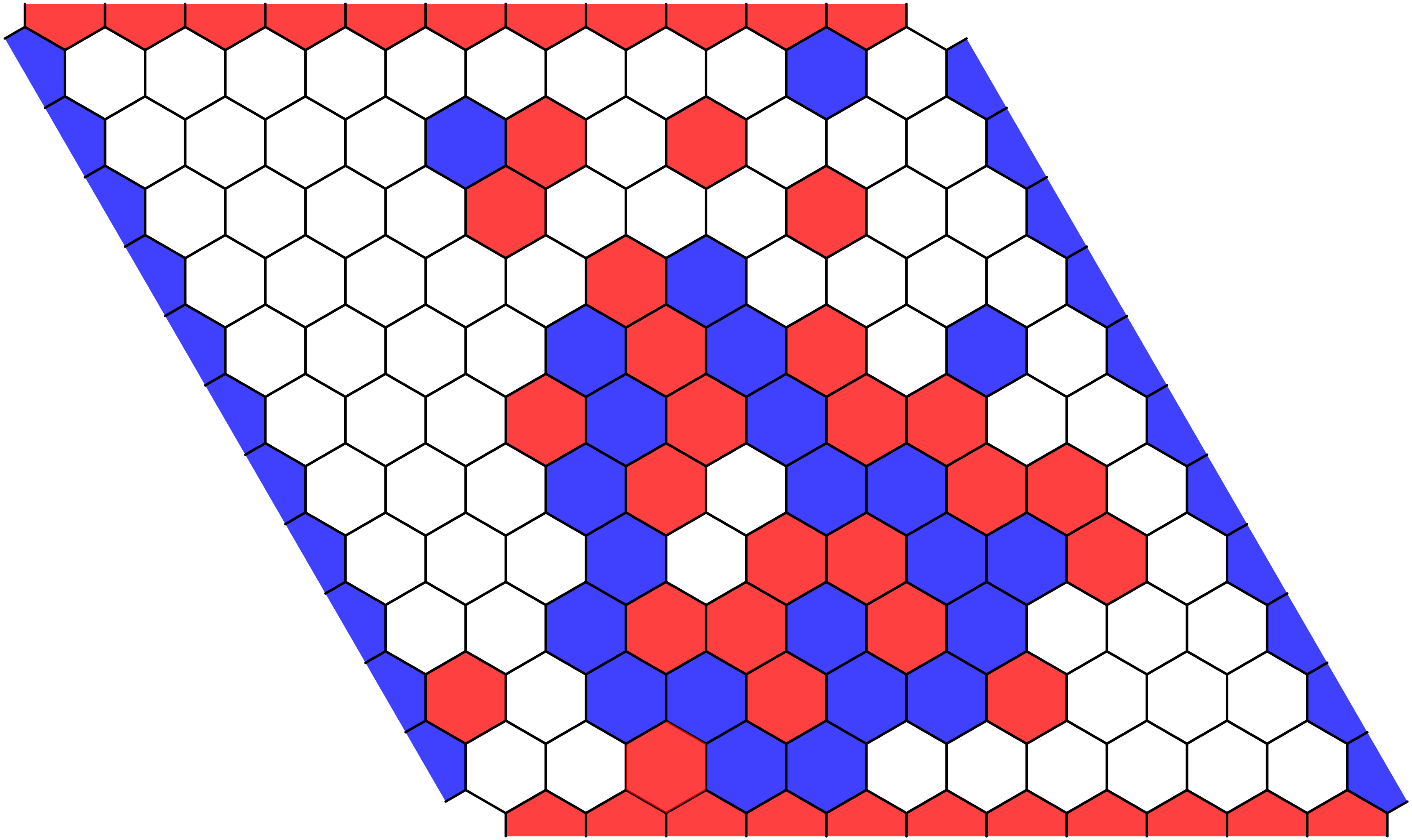


HEX (ヘックス)

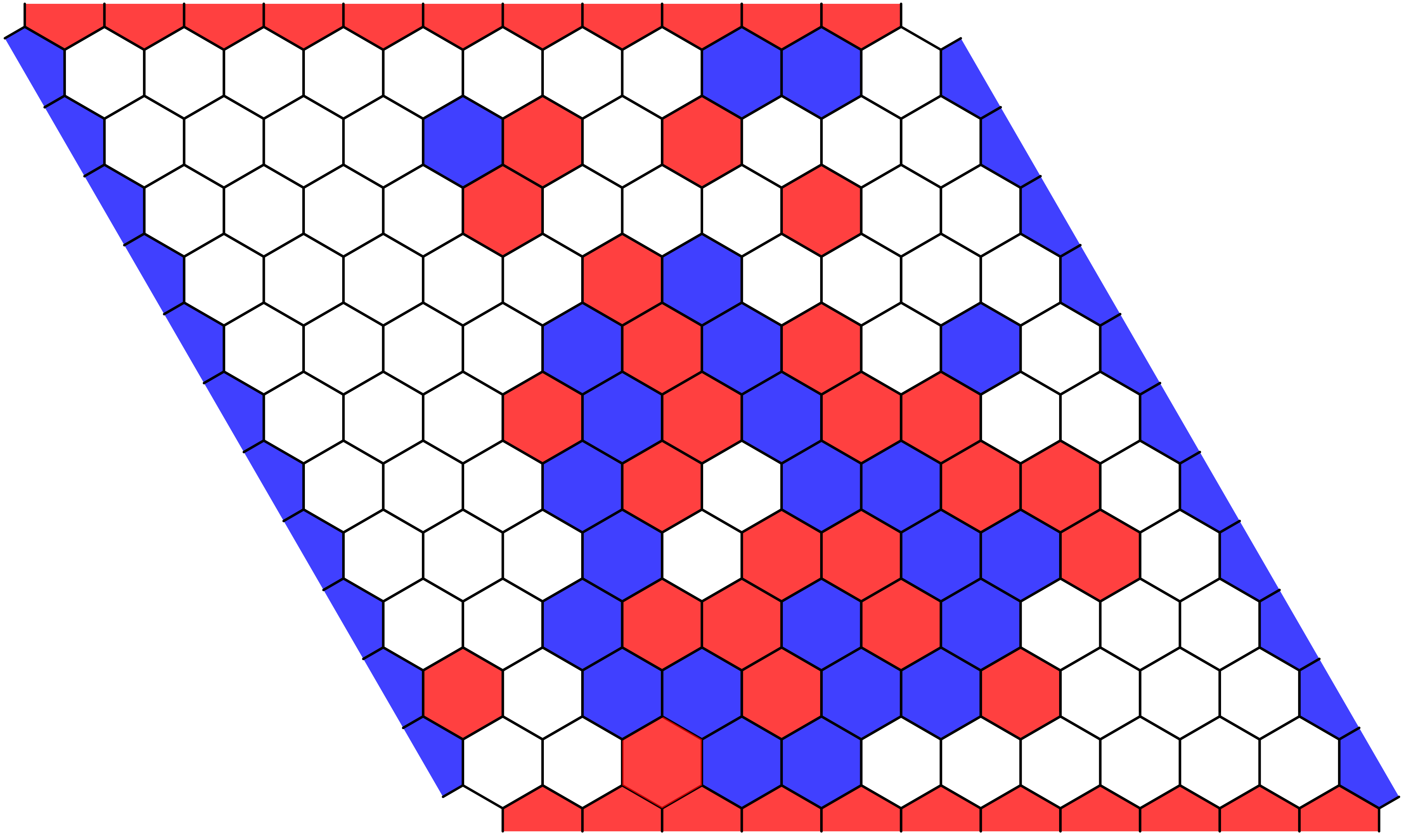


HEX (ヘックス)

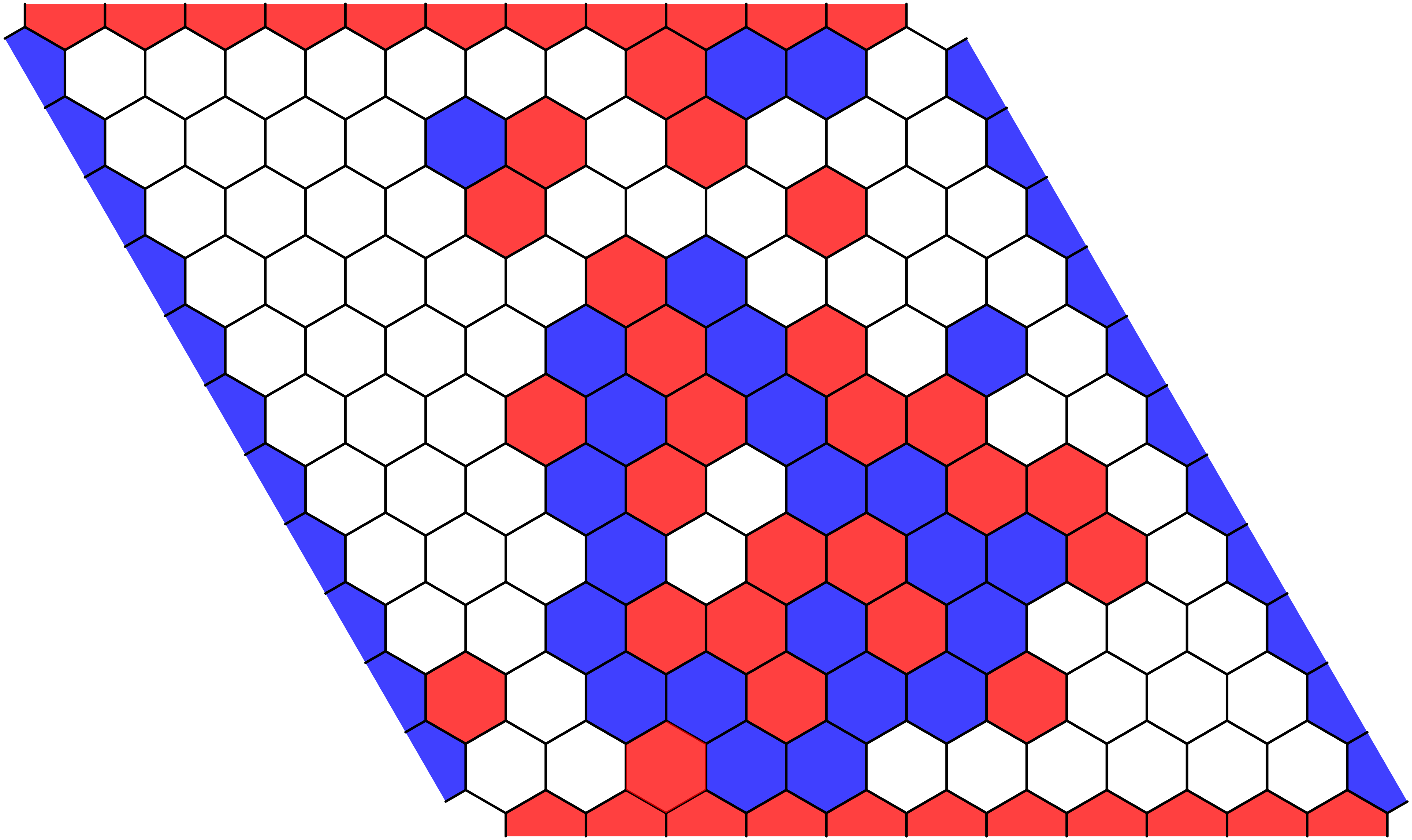




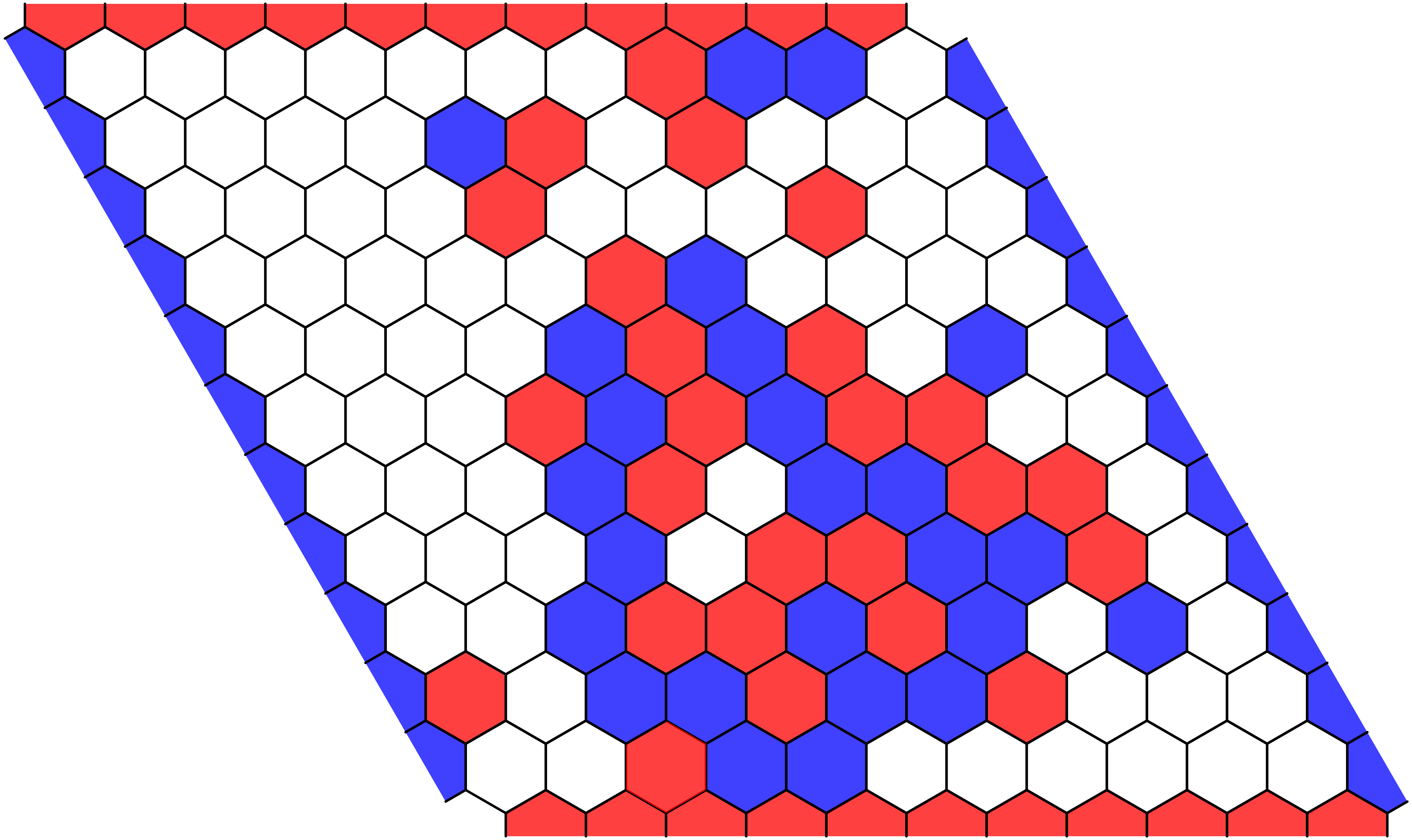
HEX (ヘックス)



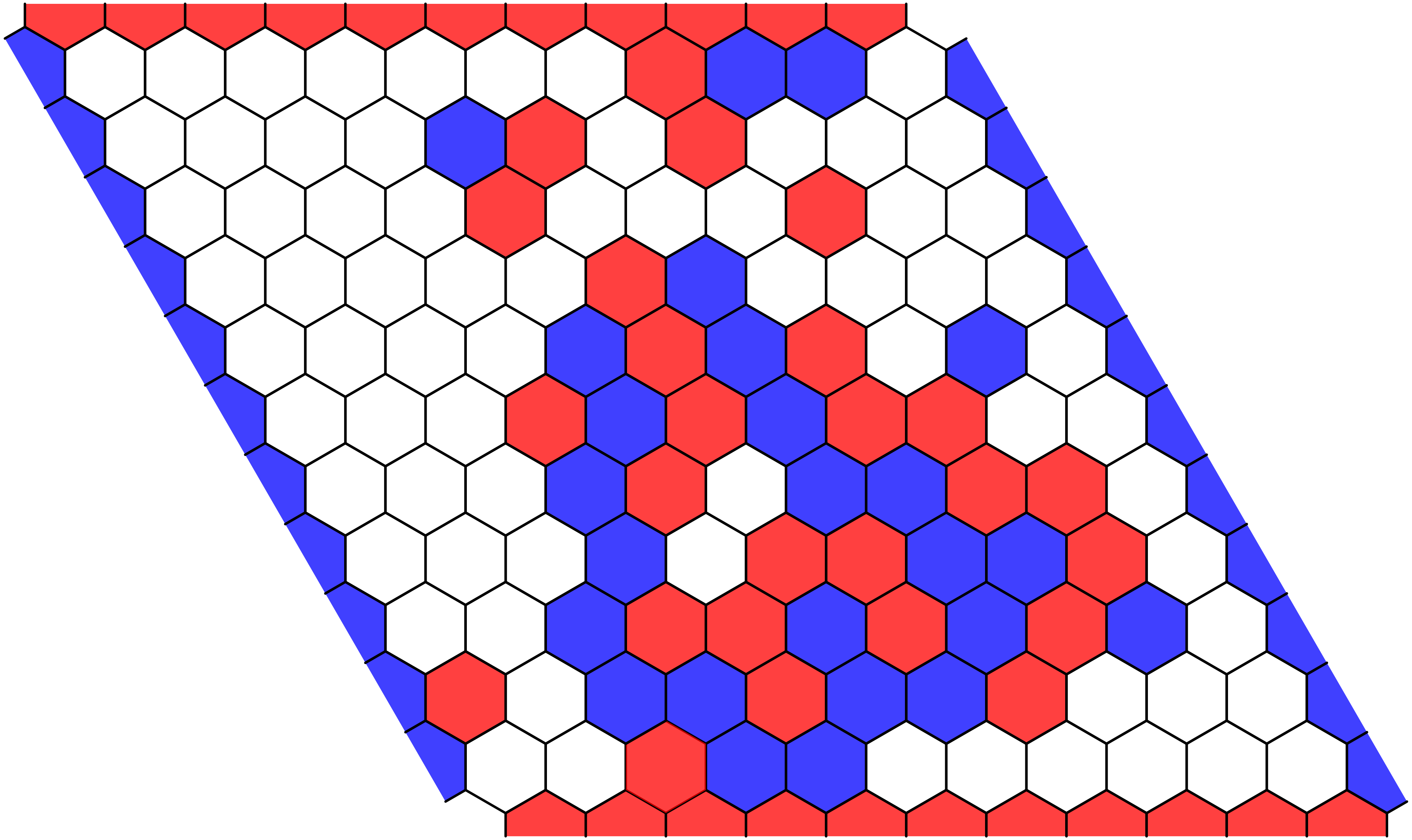
HEX (ヘックス)



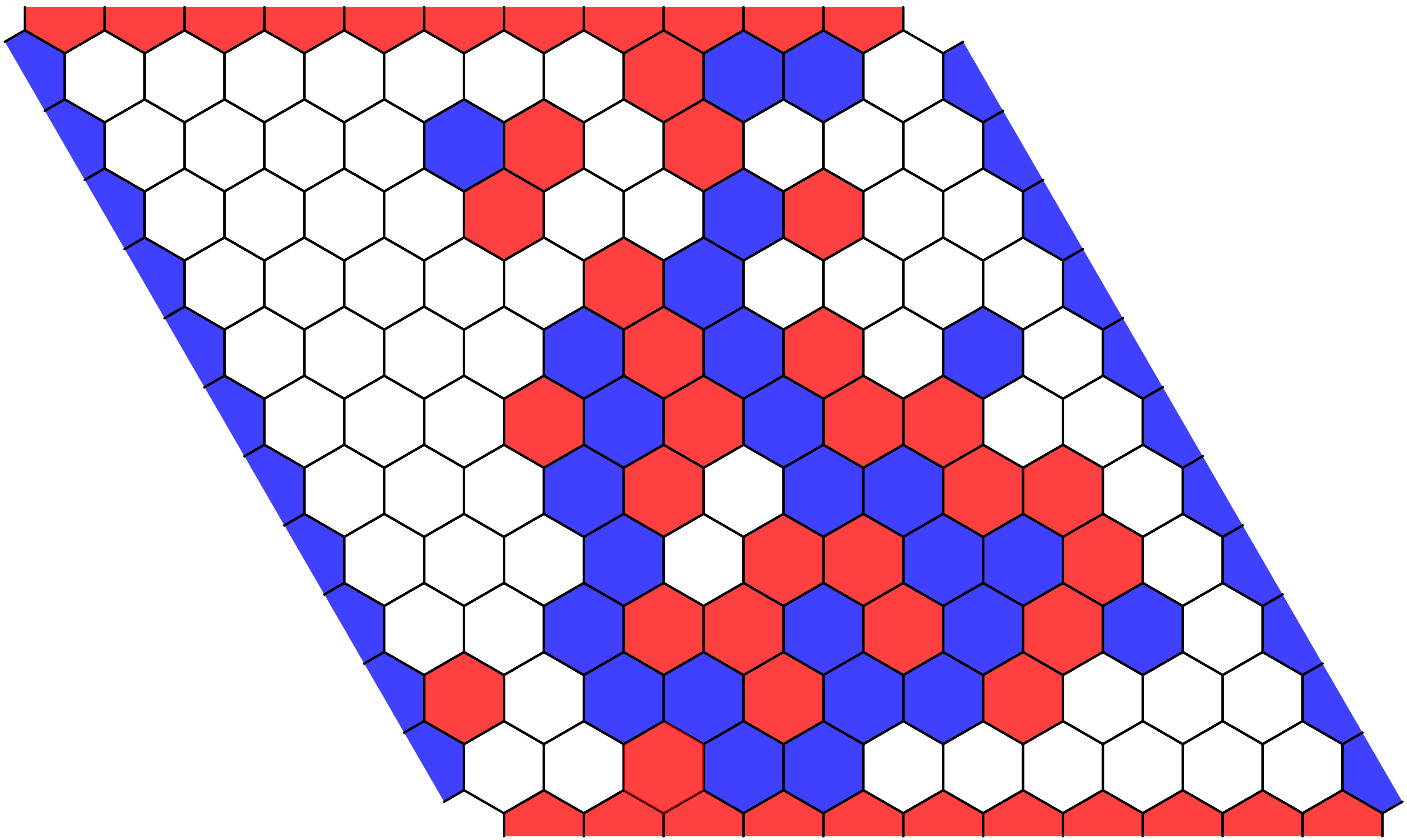
HEX (ヘックス)



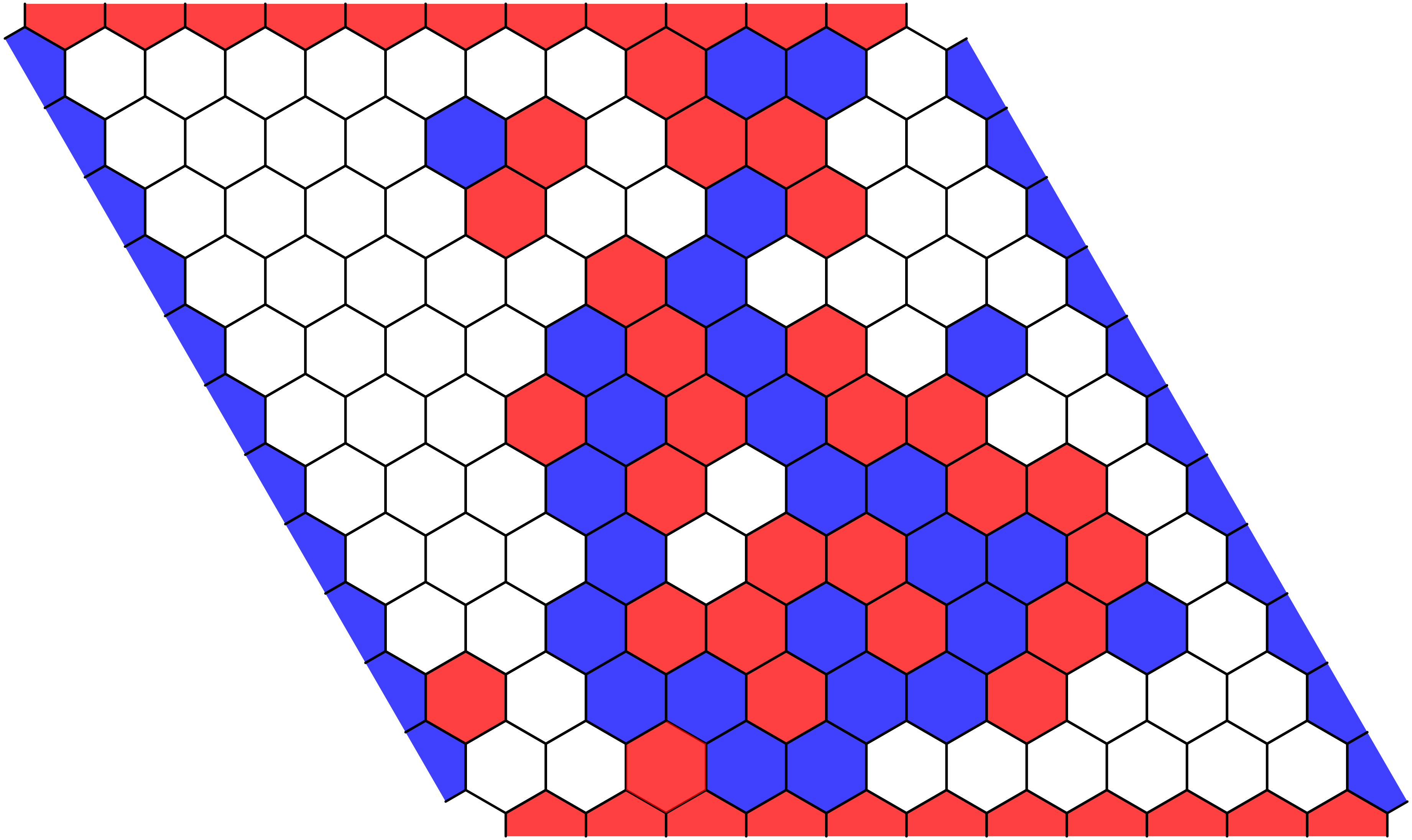
HEX (ヘックス)



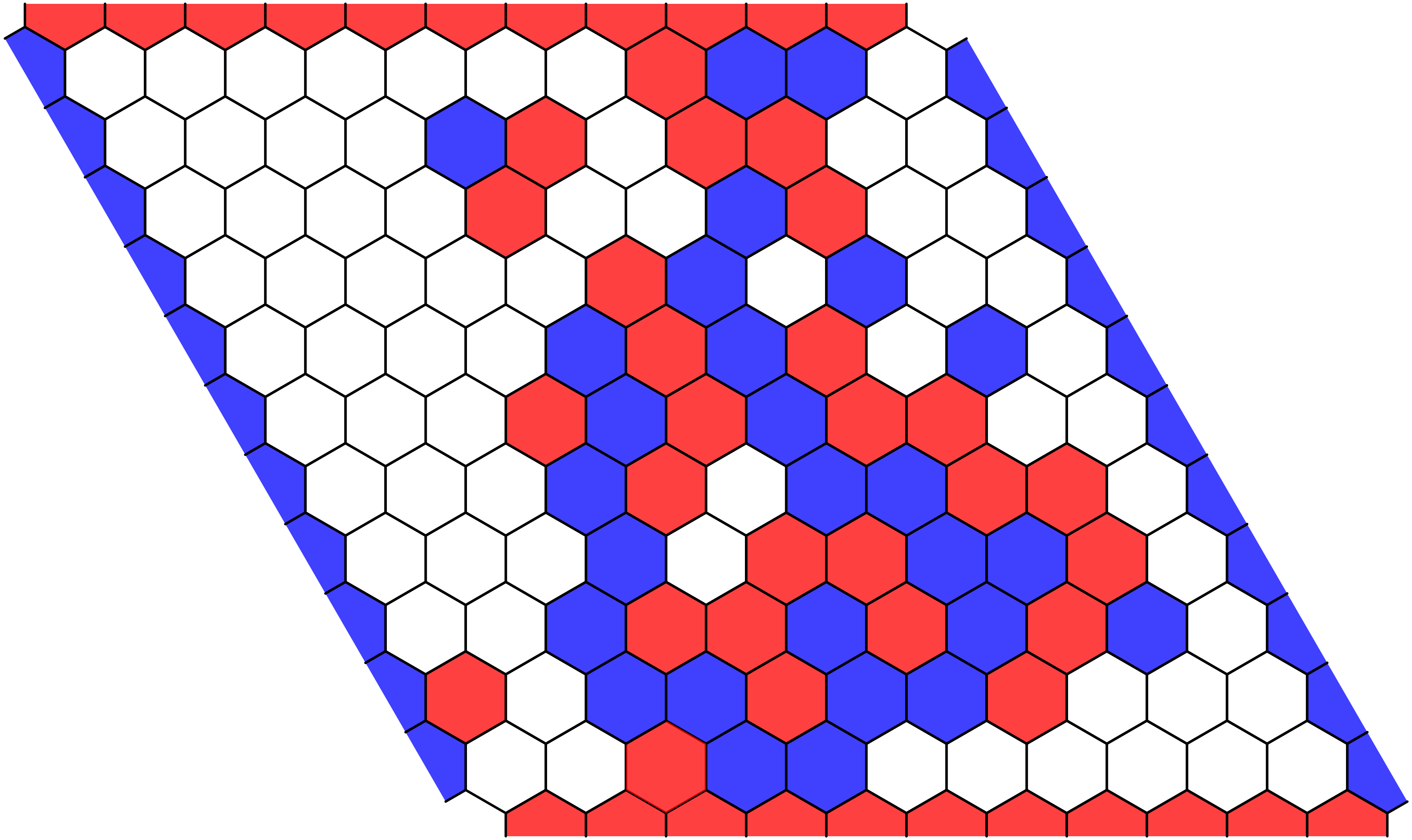
HEX (ヘックス)



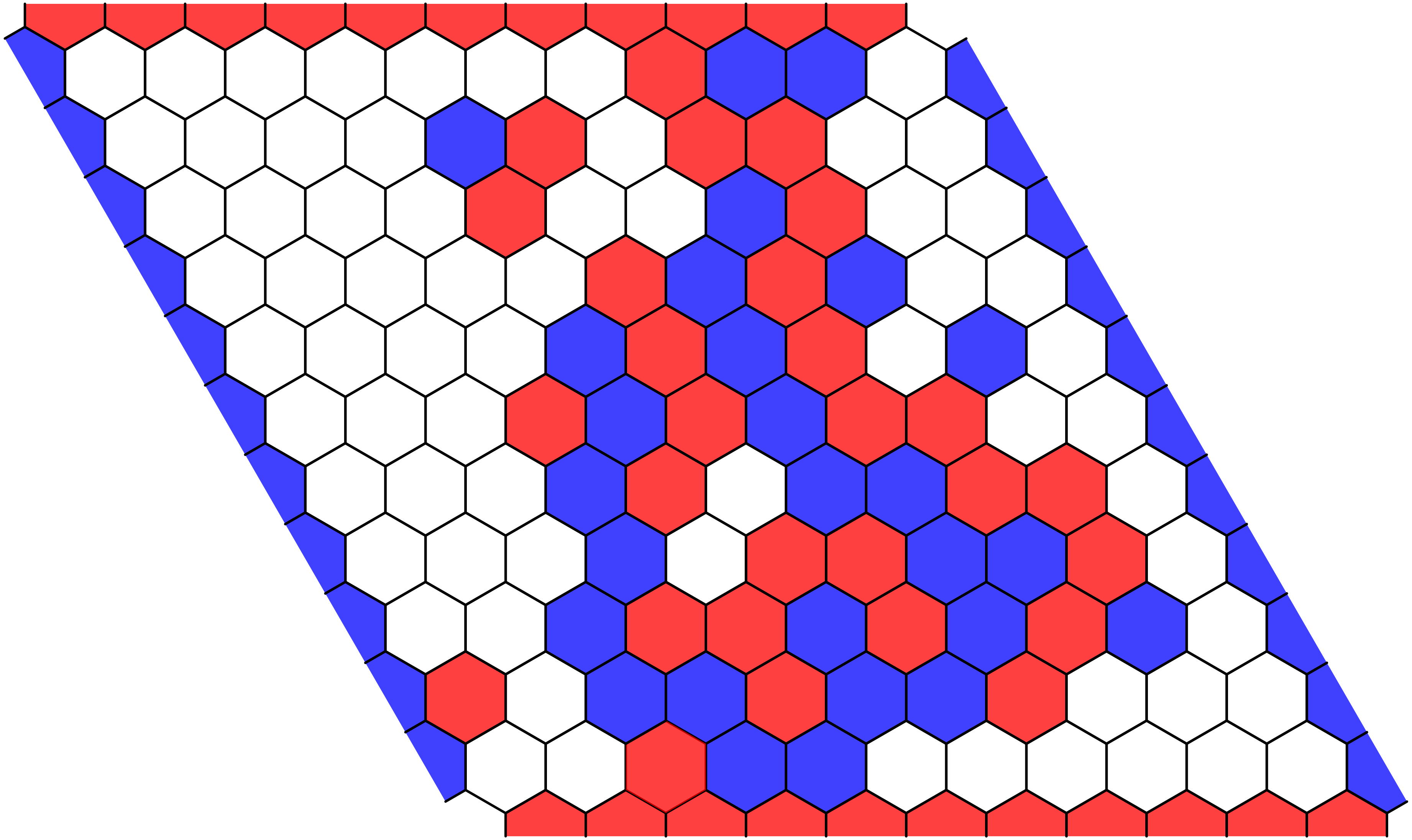
HEX (ヘックス)



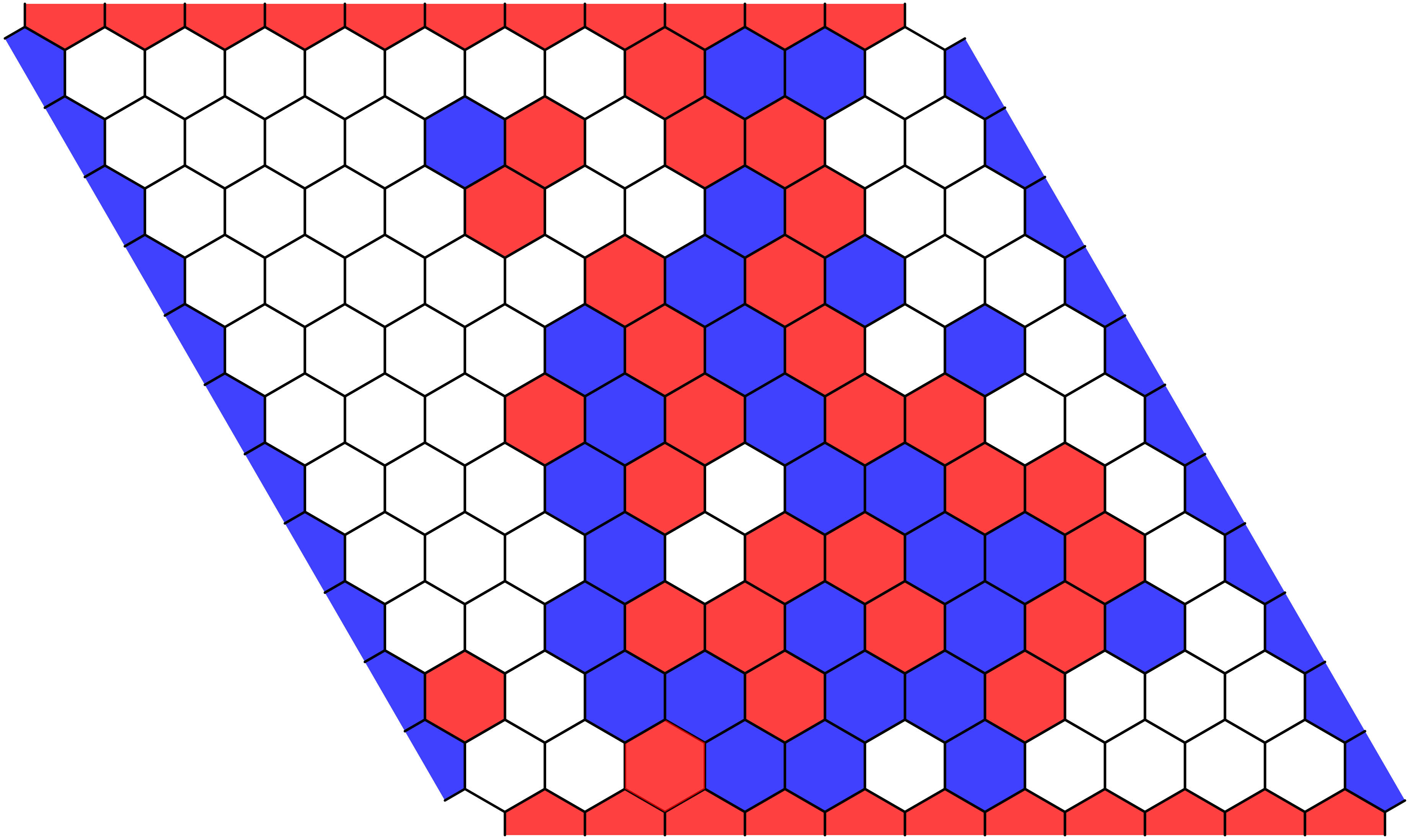
HEX (ヘックス)



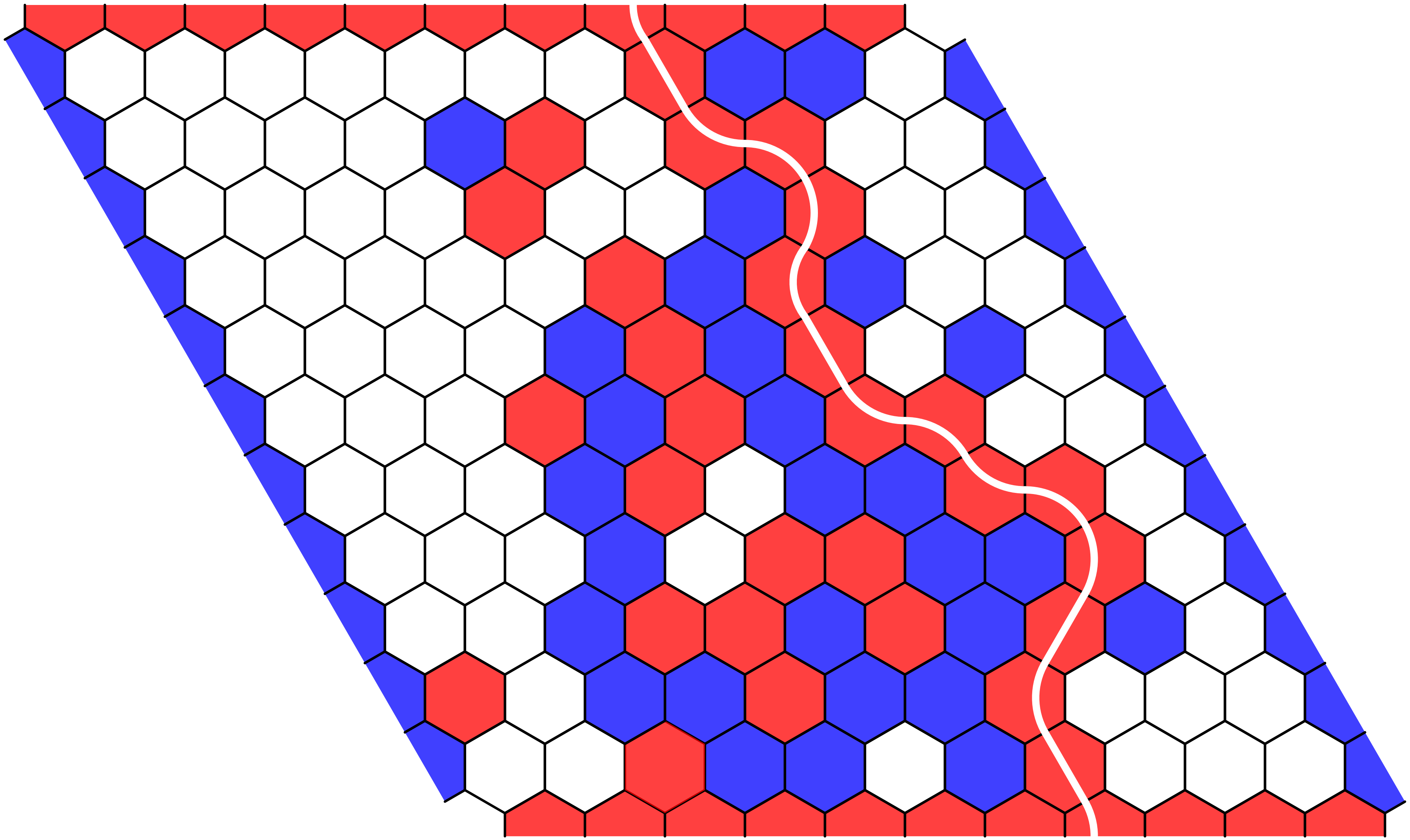
HEX (ヘックス)



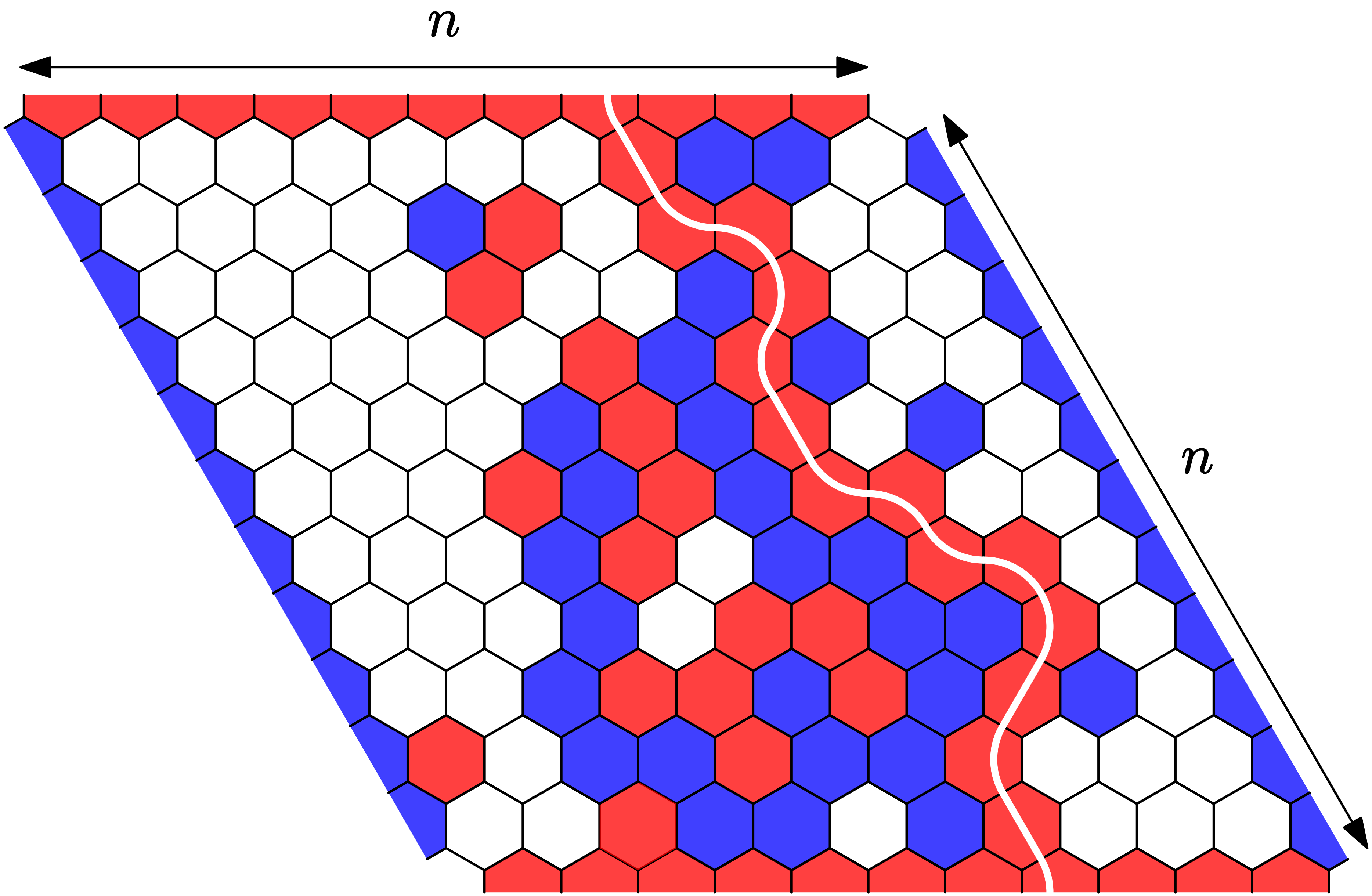
HEX (ヘックス)



HEX (ヘックス)



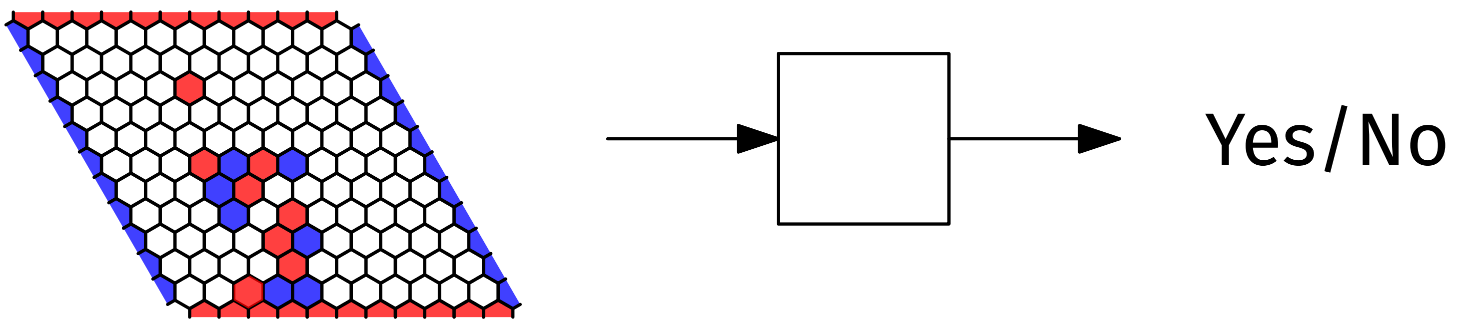
HEX (ヘックス)



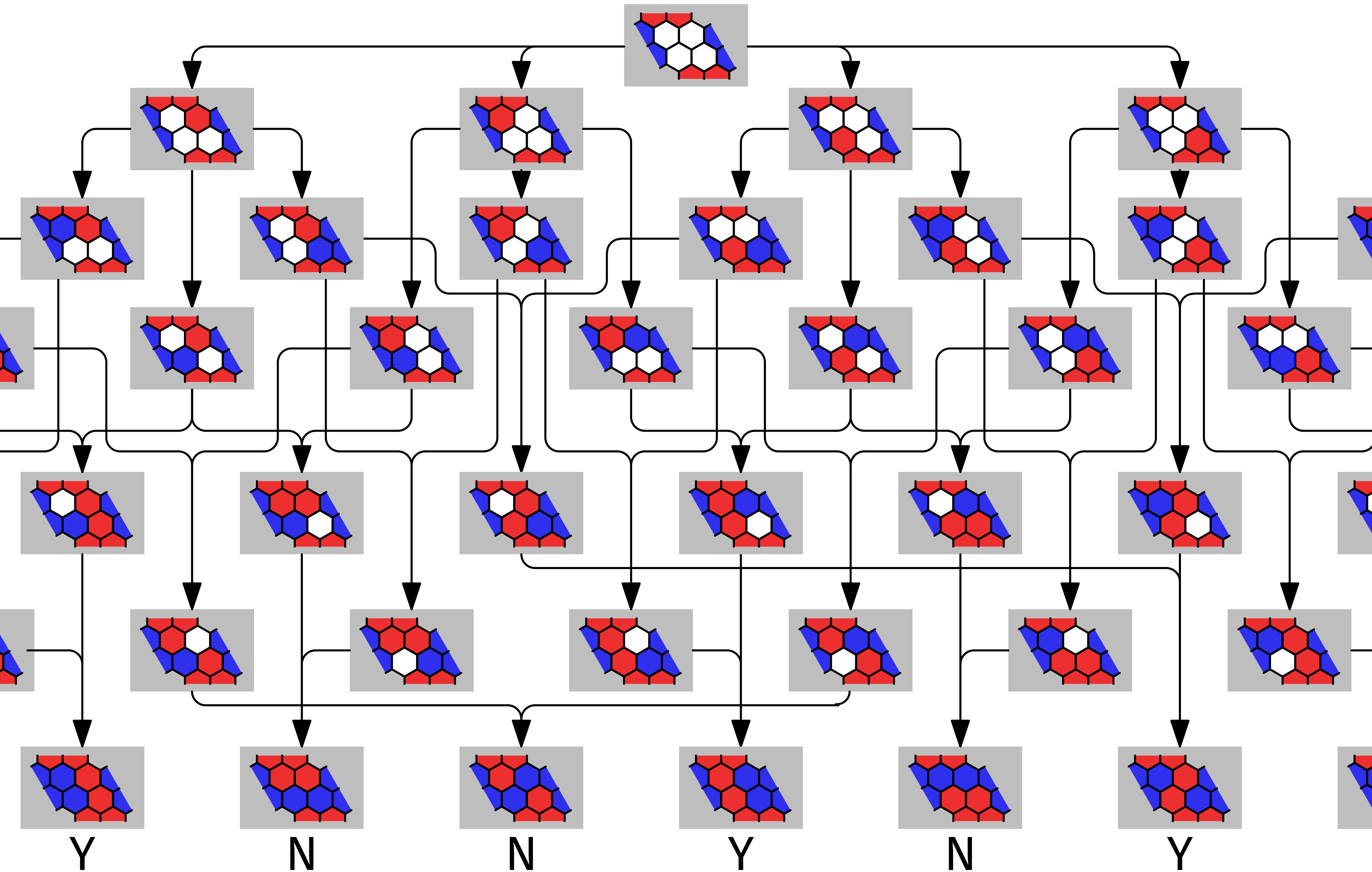
問題 HEX

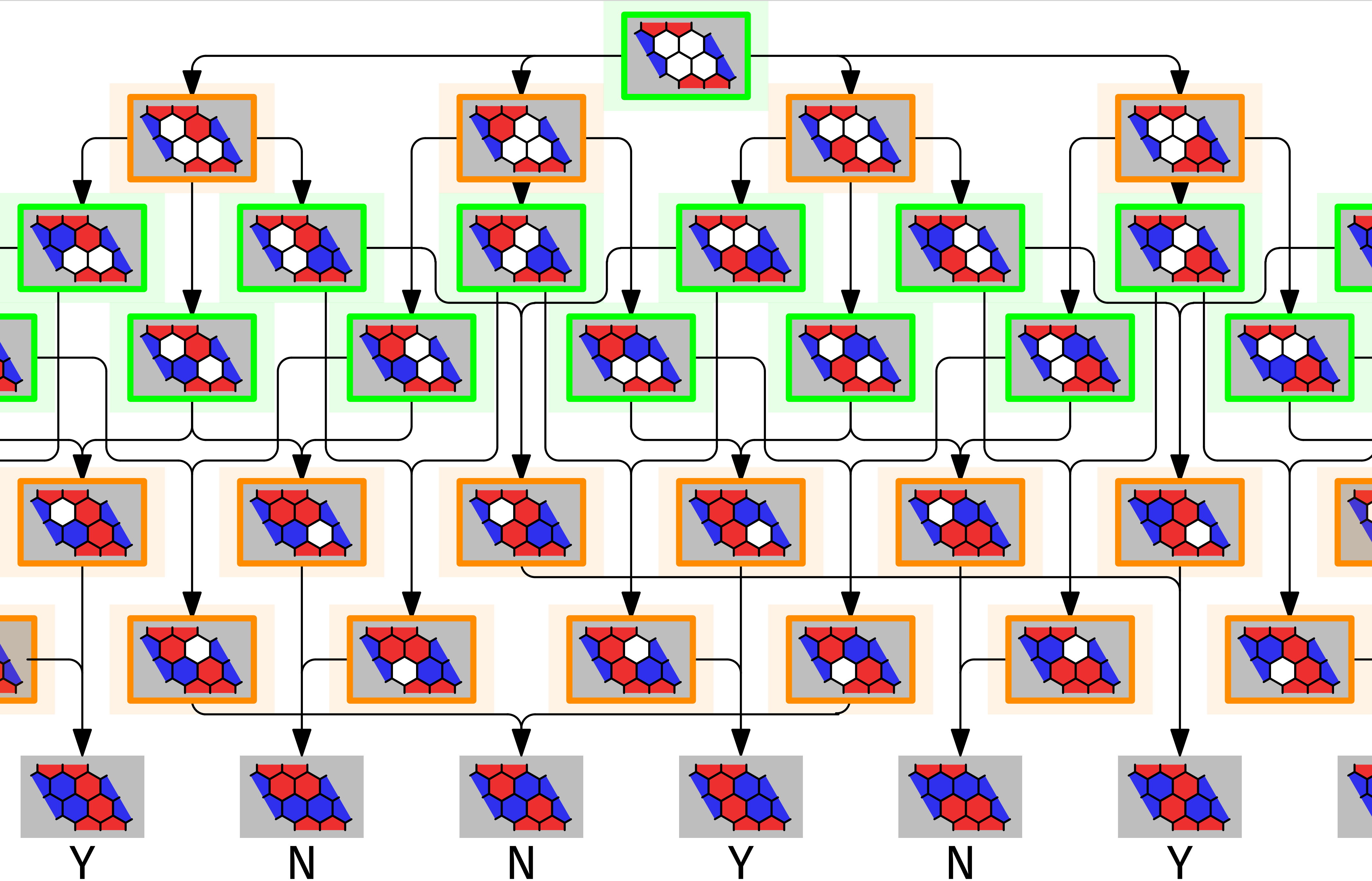
入力： $n \times n$ の HEX の途中局面
出力： 赤が必勝 $\Rightarrow$ Yes
青が必勝 $\Rightarrow$ No

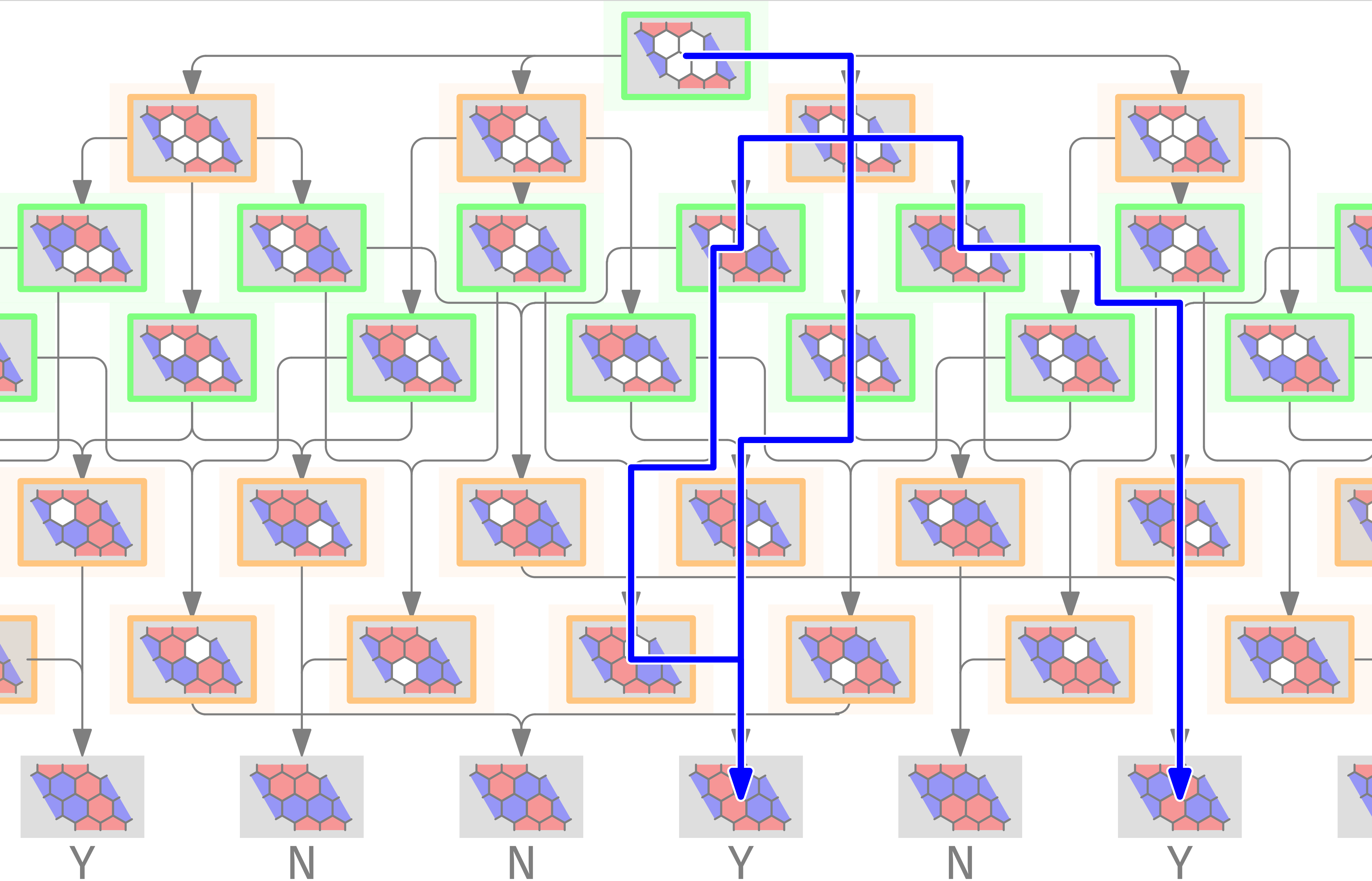
既知：HEX では，どちらかが必勝 (引き分けにならない)



問題 HEX：配位グラフ



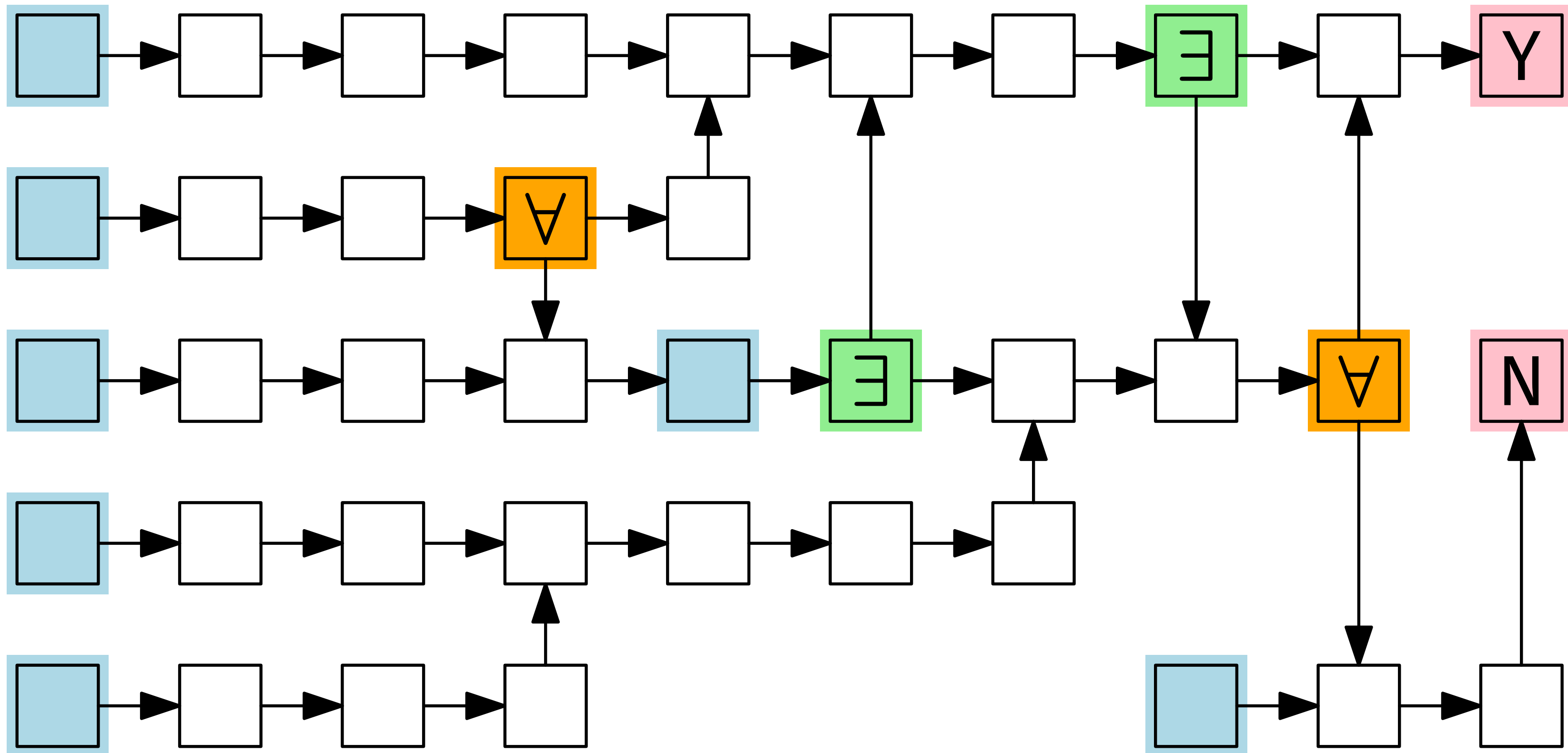

Y
N
N
Y
N
Y


Y
N
N
Y
N
Y

本日の内容

1. **交代性計算とは？**
2. AP $=$ PSPACE と APSPACE $=$ EXPTIME
3. QBF：交代性から見た PSPACE 完全問題
4. 難しい問題の例

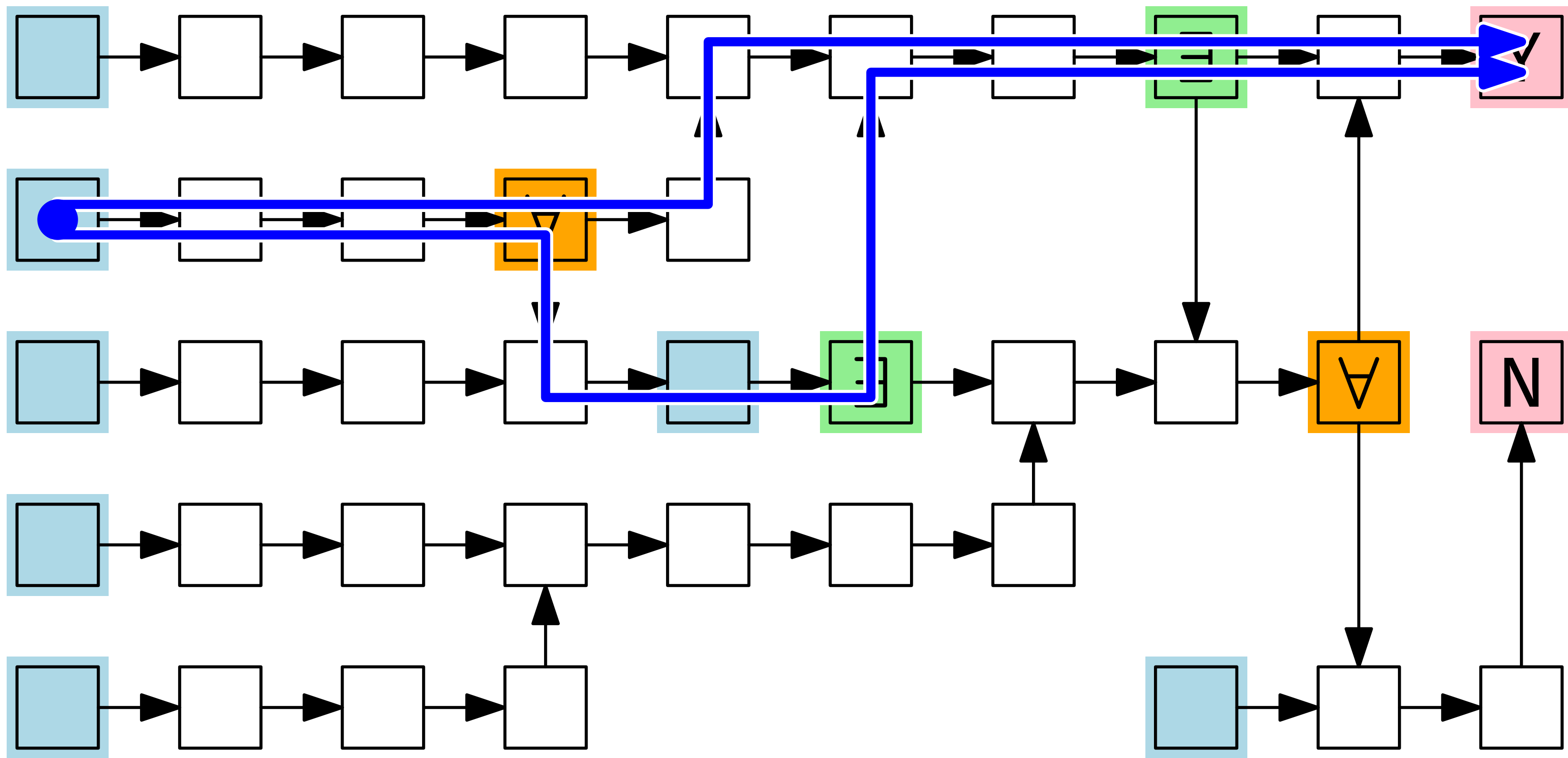
交代性：時点状況から見る



5 種類の時点状況

- 開始状況
- 停止状況
- 決定性状況
- 存在状況 (非決定性)
- 全称状況 (非決定性)

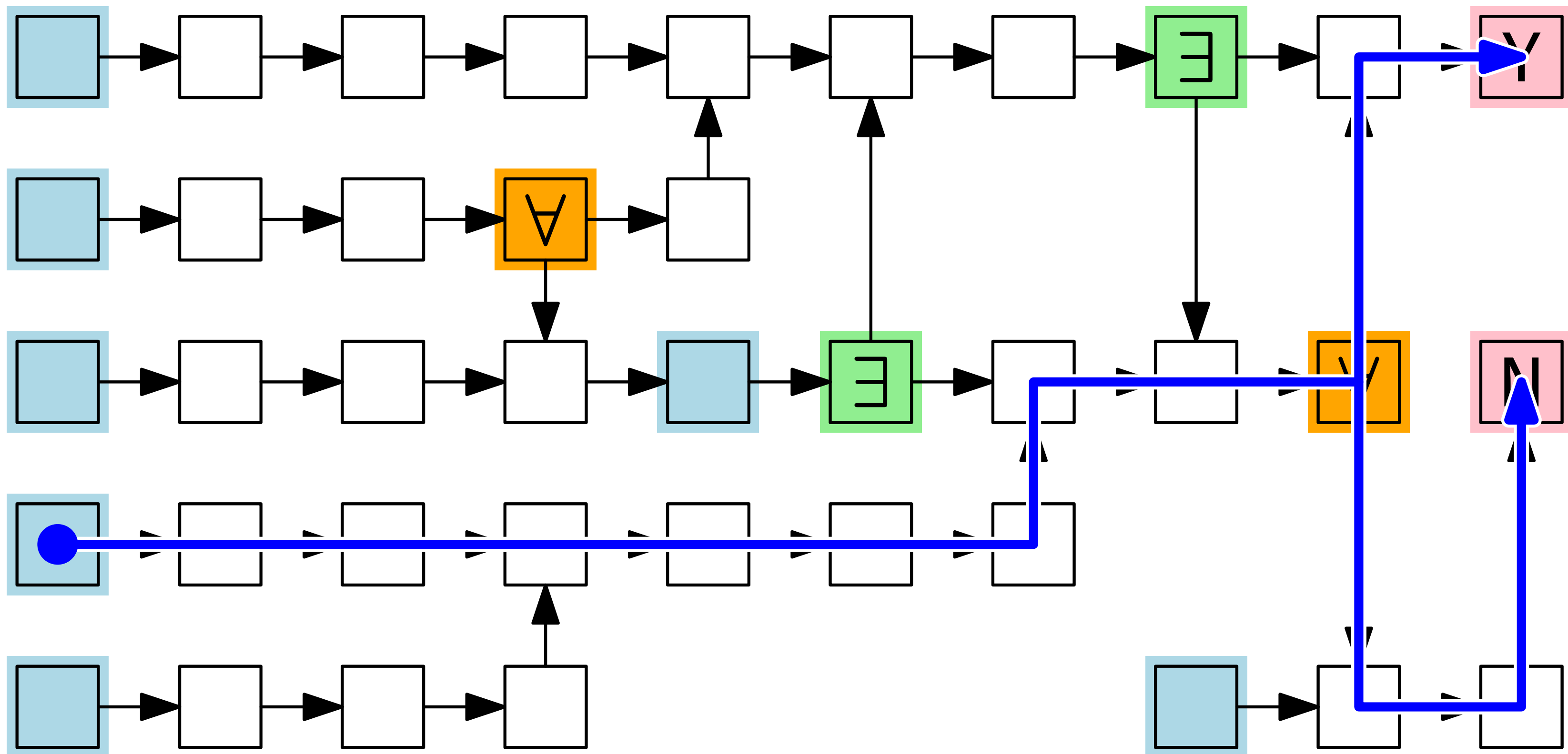
交代性：時点状況から見る



5 種類の時点状況

- 開始状況
- 停止状況
- 決定性状況
- ∃ 存在状況 (非決定性)
- ∀ 全称状況 (非決定性)

交代性：時点状況から見る



5 種類の時点状況

- 開始状況
- 停止状況
- 決定性状況
- 存在状況 (非決定性)
- 全称状況 (非決定性)

交代性を持つ計算

交代性を持つプログラムでは

命令 guess-∃ と guess-∀ を使える

- 有限集合 X に対して，guess-∃(X) とすると，X の要素を 1 つ好きなように選べる
- 有限集合 X に対して，guess-∀(X) とすると，X の要素をすべて考えて分岐する

それらにかかる時間 $= O(\log |X|)$ ステップ

交代性における guess-∃ $=$ 非決定性における guess

交代性 $=$ alternation (名詞)

交代的 $=$ alternating (形容詞)

(Chandra, Kozen, Stockmeyer ’81)

交代性を持つ計算：例

この質問に正しく答えたい

集合 X $= \{1, 4, 6, 9\}$ は次の性質を満たすか？
任意の $a \in$ X に対して，ある $b \in$ X が存在して，
$a + b = 10$ となる

非決定性

```
str = "Yes"
foreach a in X:
  b = guess(X):
    if a + b != 10:
      str = "No"
    end
  end
return str
```

交代性

```
a = guess-∀(X)
b = guess-∃(X)
if a + b == 10:
  return "Yes"
else:
  return "No"
end
```

交代性を使って解くとは？

定義 (非形式)：交代性アルゴリズム

判定問題 P を **解く交代性アルゴリズム** とは,
任意の入力 I に対して，次を行うもの

- 存在状況に到達したとき，guess をうまく選んで次が成り立つ
  1. どの経路も必ず停止状況に到達する
  2. I が Yes インスタンス $\Rightarrow$ すべての経路が Yes の停止状況に到達
  3. I が No インスタンス $\Rightarrow$ ある経路が No の停止状況に到達

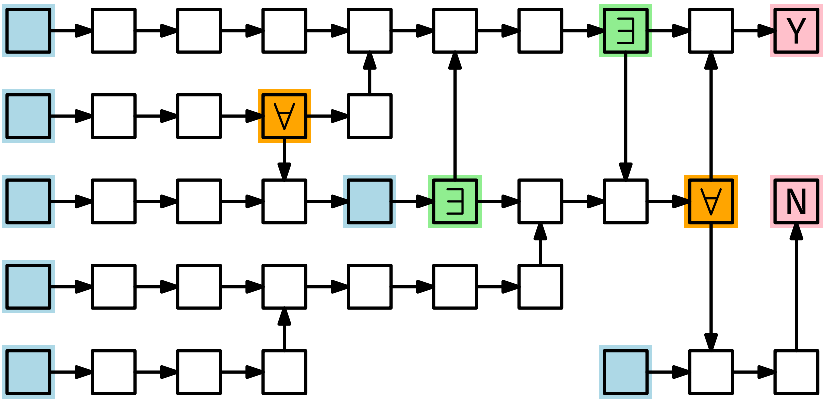
交代性計算における時間計算量

P は判定問題，A は P を解く交代性アルゴリズム

定義：交代性アルゴリズムの時間計算量

アルゴリズム A の **時間計算量** (time complexity) とは，
符号長 n 以下の入力 I に対して，
I に対応する開始時点から停止時点に至る最大経路長の
I に関する最大値

仮定：アルゴリズムの各行は定数個のステップしか含まない



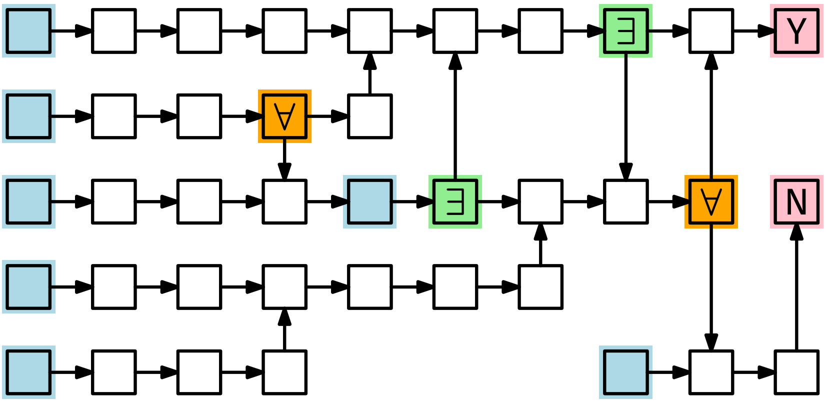
交代性計算における領域計算量

P は判定問題，A は P を解く交代性アルゴリズム

定義：交代性アルゴリズムの領域計算量

アルゴリズム A の **領域計算量** (space complexity) とは，
符号長 n 以下の入力 I に対して，
I に対応する開始時点から到達できる時点状況における
作業領域使用量の I に関する最大値 (ビット)

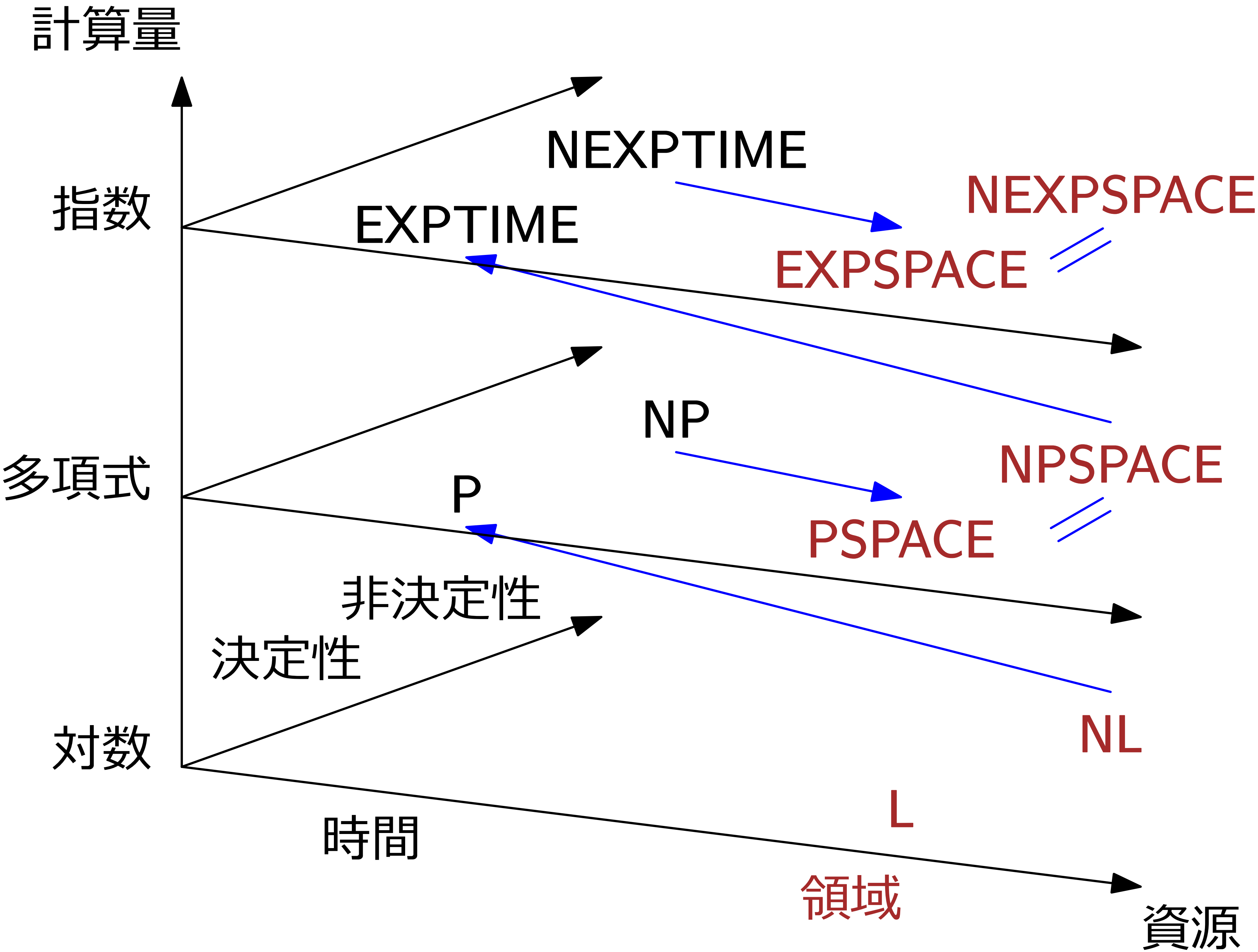
仮定：アルゴリズムの各行は定数個のステップしか含まない



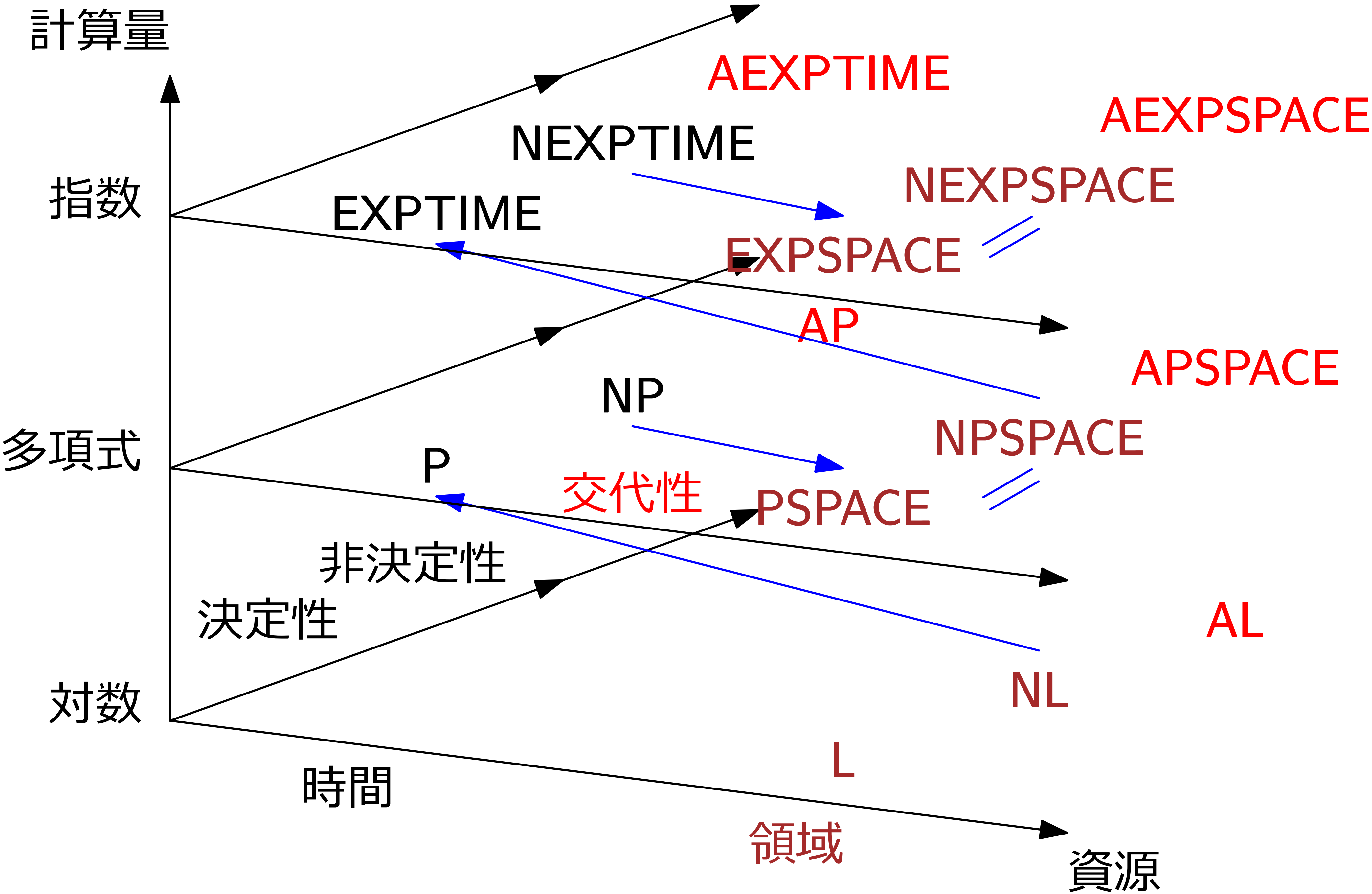
交代性に関わる計算複雑性クラス

クラス	定義
	交代性を持つアルゴリズムによって
AP	多項式時間で解ける判定問題全体のクラス
AEXPTIME	指数時間で解ける判定問題全体のクラス
AL	対数領域で解ける判定問題全体のクラス
APSPACE	多項式領域で解ける判定問題全体のクラス
AEXPSPACE	指数領域で解ける判定問題全体のクラス

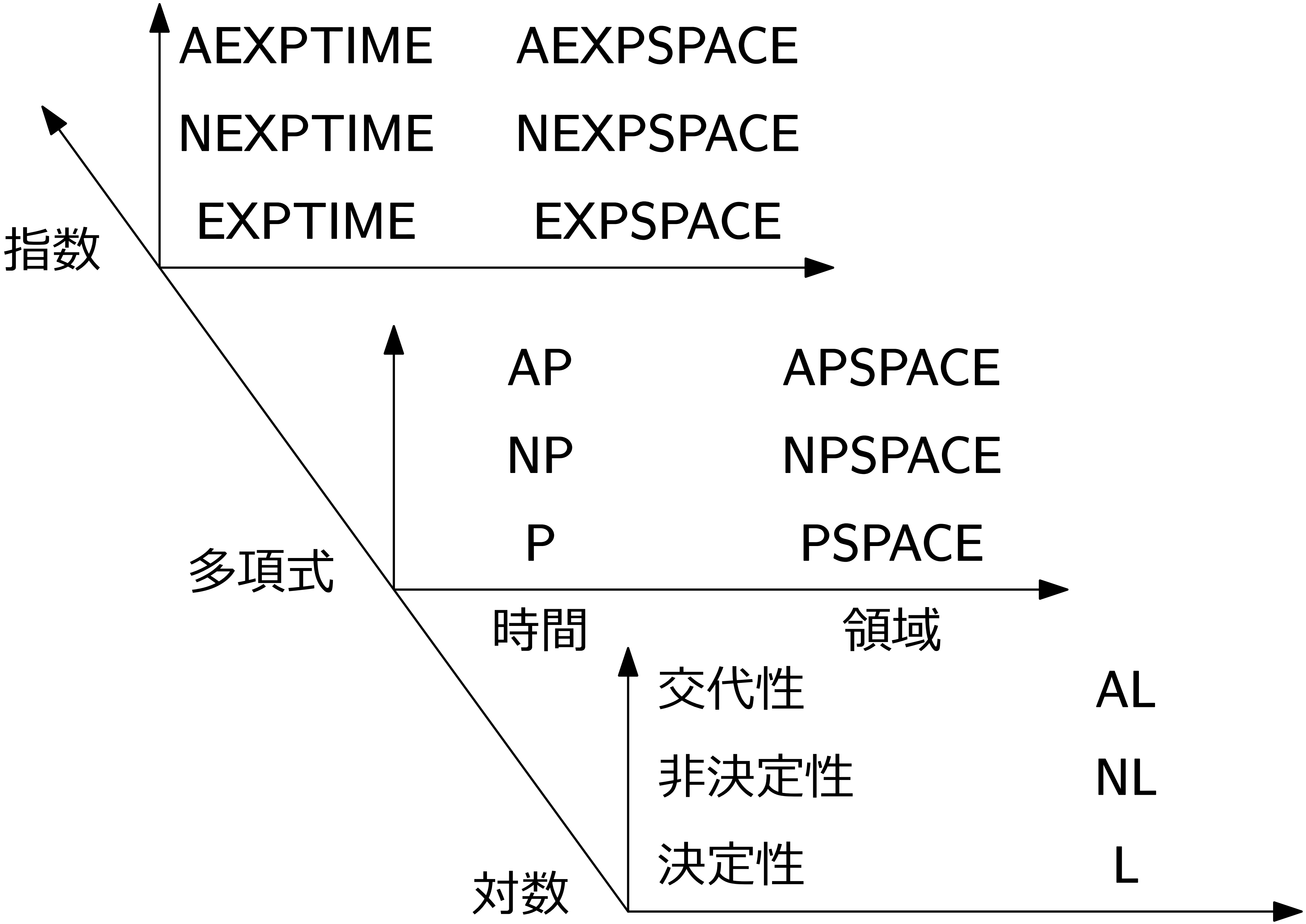
三次元的？ に書くと


計算量
指数
多項式
対数
NEXPTIME
EXPTIME
NEXPSPACE
EXPSPACE
NP
P
NPSPACE
PSPACE
非決定性
決定性
NL
L
時間
領域
資源

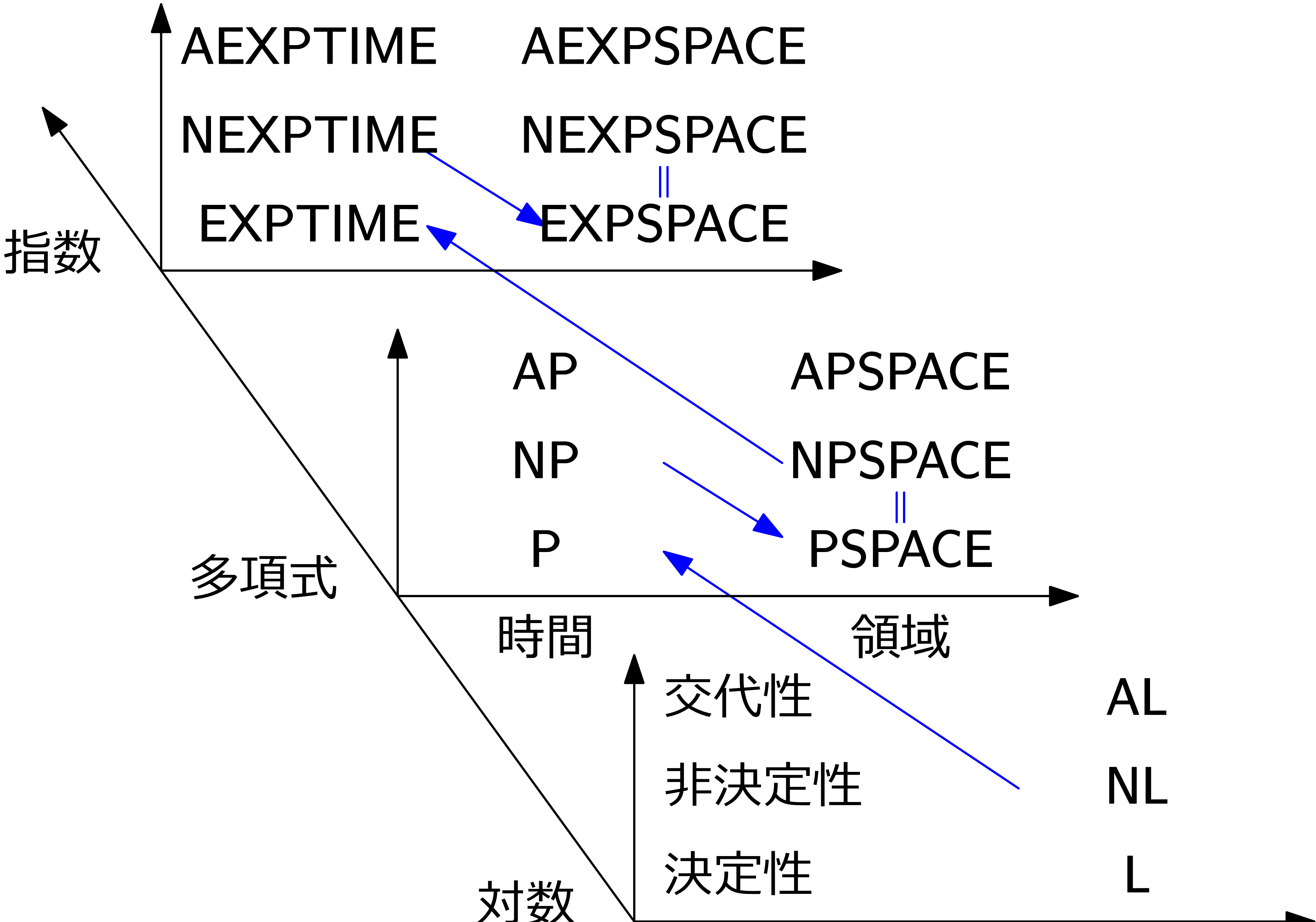
三次元的？ に書くと


計算量
AEXPTIME
AEXPSPACE
NEXPTIME
指数
NEXPSPACE
EXPTIME
EXPSPACE
AP
APSPACE
NP
NPSPACE
多項式
P
交代性
PSPACE
非決定性
決定性
AL
NL
対数
L
時間
領域
資源

フロアごと？に書くと

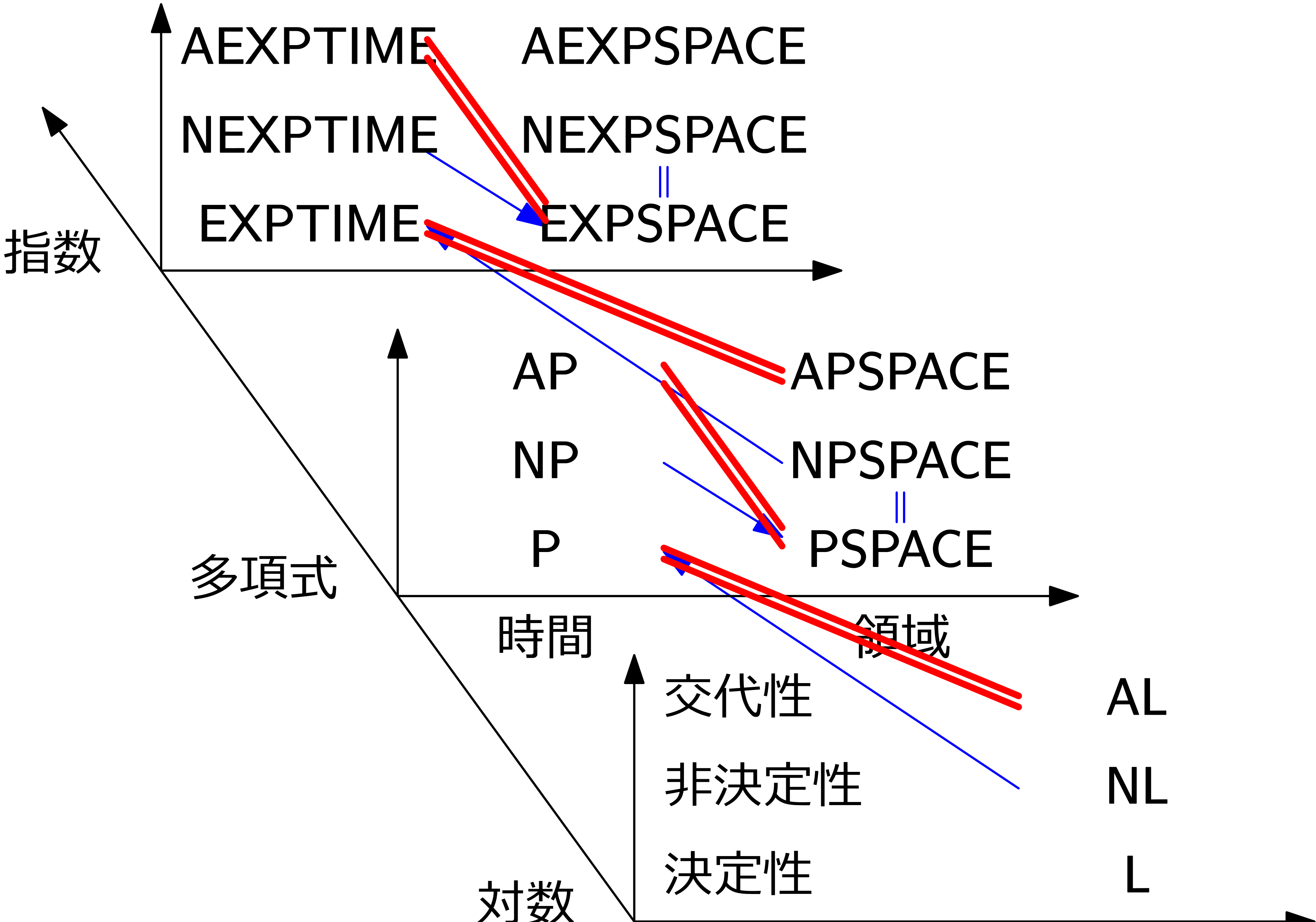

AEXPTIME
AEXPSPACE
NEXPTIME
NEXPSPACE
EXPTIME
EXPSPACE
指数
AP
APSPACE
NP
NPSPACE
P
PSPACE
多項式
時間
領域
交代性
AL
非決定性
NL
決定性
L
対数

フロアごと？に書くと



AEXPTIME
AEXPSPACE
NEXPTIME
NEXPSPACE
EXPTIME
EXPSPACE
指数
AP
APSPACE
NP
NPSPACE
P
PSPACE
多項式
時間
領域
交代性
AL
非決定性
NL
決定性
L
対数

フロアごと？に書くと


AEXPTIME
AEXPSPACE
NEXPTIME
NEXPSPACE
EXPTIME
EXPSPACE
指数
AP
APSPACE
NP
NPSPACE
P
PSPACE
多項式
時間
領域
交代性
AL
非決定性
NL
決定性
L
対数

本日の内容

1. 交代性計算とは？
2. AP = PSPACE と AL = P
3. QBF：交代性から見た PSPACE 完全問題
4. 難しい問題の例

AP = PSPACE

性質 (Chandra, Kozen, Stockmeyer ’86)

AP = PSPACE

つまり，「交代性多項式時間 = 決定性多項式領域」

同様に，AEXPTIME = EXPSPACE も正しい

標語的にいうと

時間$(f(n))$ + 交代性 = 領域$(f(n))$

AP = PSPACE

性質 (Chandra, Kozen, Stockmeyer ’86)

AP = PSPACE

つまり，「交代性多項式時間 = 決定性多項式領域」

同様に，AEXPTIME = EXPSPACE も正しい

証明：次の 2 つを証明すればよい

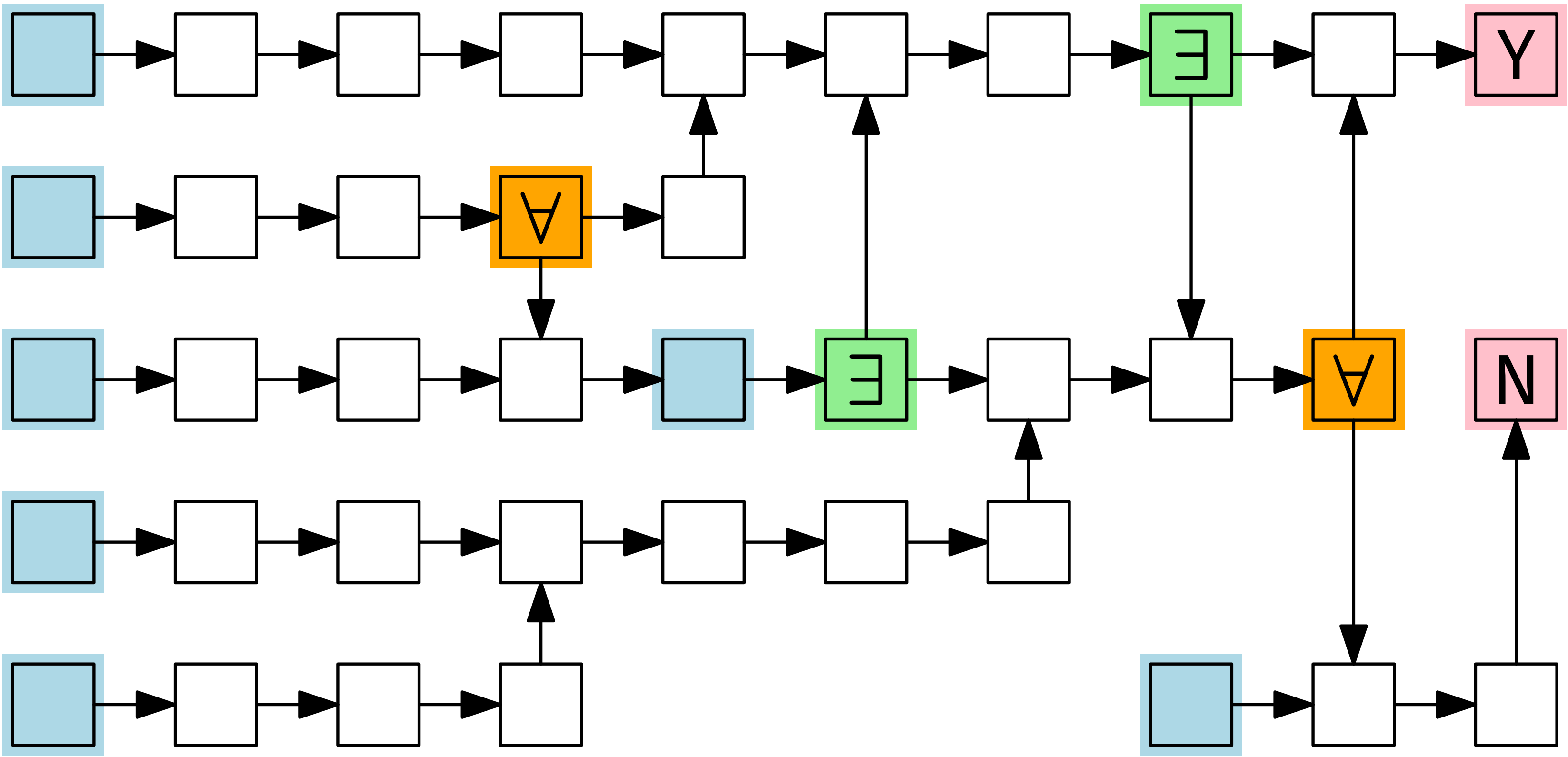
- AP $\subseteq$ PSPACE
- PSPACE $\subseteq$ AP

標語的にいうと

時間$(f(n))$ + 交代性 = 領域$(f(n))$

AP ⊆ PSPACE

証明は NP ⊆ PSPACE に似ている



時点状況のグラフにおいて深さ優先探索を行う

- 交代性多項式時間アルゴリズムでは, 各時点状況の作業領域量も多項式
- ∴ これは決定性多項式領域アルゴリズム

PSPACE $\subseteq$ AP (1)

証明は Savitch の定理 の証明 に似ている

NPSPACE $\subseteq$ PSPACE

証明の流れ

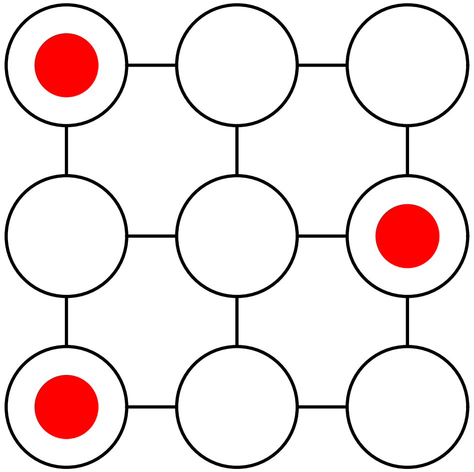
1. ある PSPACE 完全問題 P を考える
2. P を解く交代性多項式時間アルゴリズム A を作る

復習：独立集合遷移問題 (PSPACE 完全)

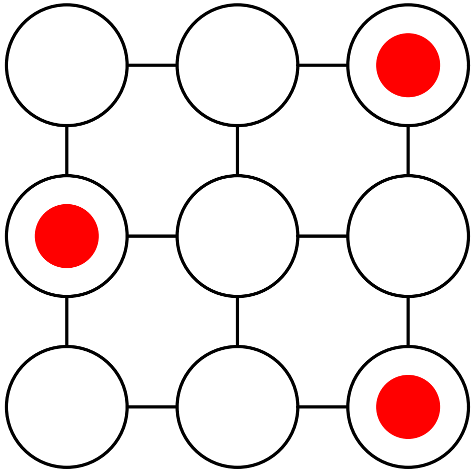
独立集合遷移問題

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



I_s



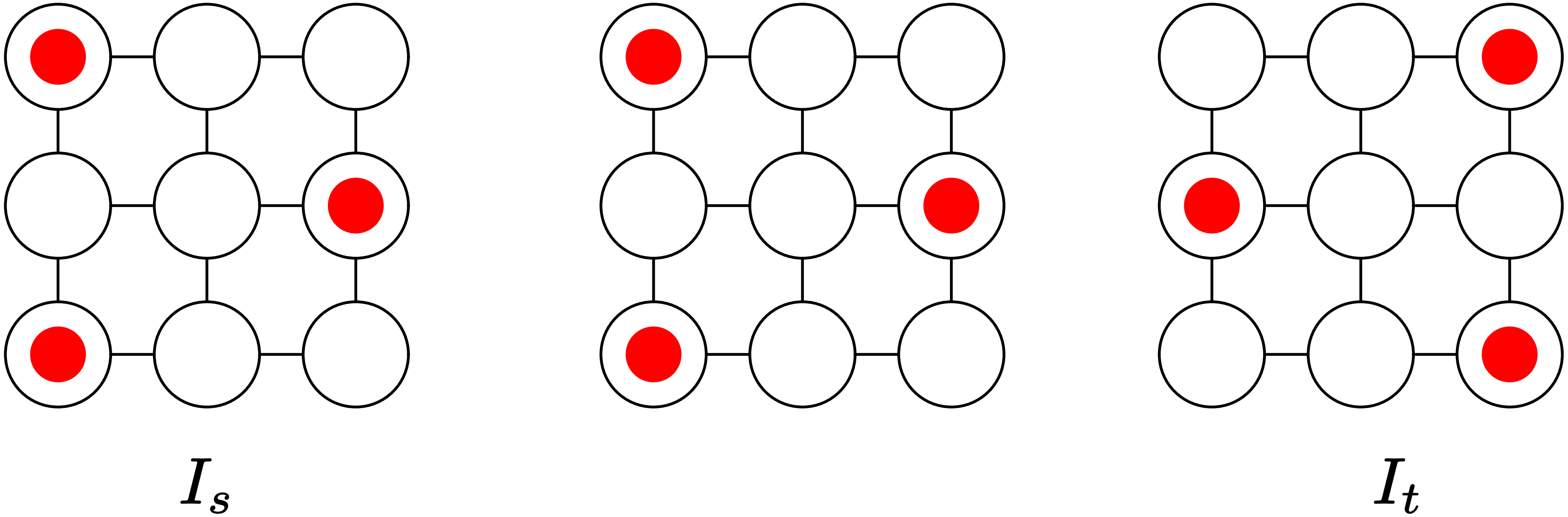
I_t

$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

独立集合遷移問題

入力： 無向グラフ $G=(V,E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No

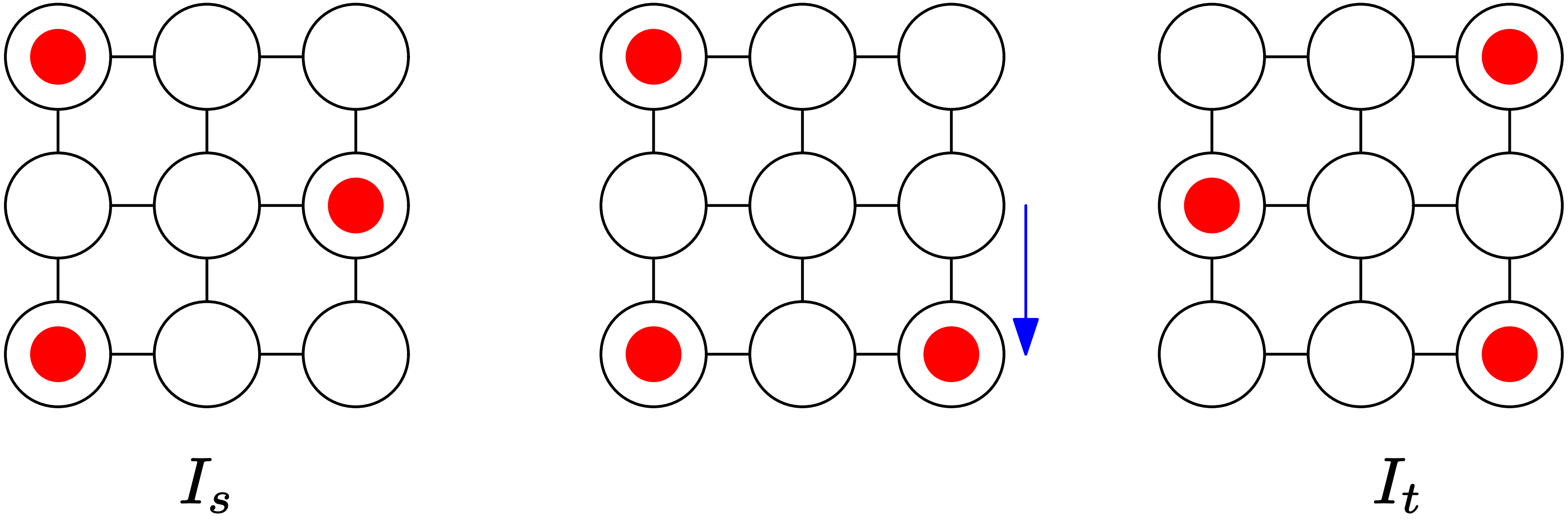


$I \subseteq V$ が $G=(V,E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

独立集合遷移問題

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



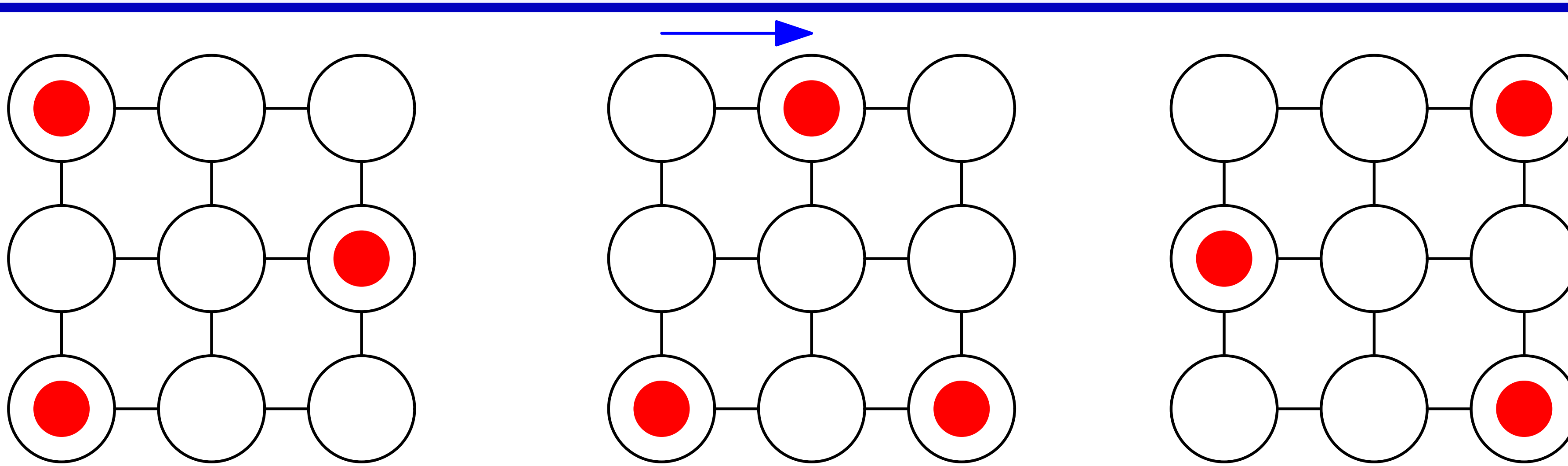
$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

復習：独立集合遷移問題 (PSPACE 完全)

独立集合遷移問題

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



I_s I_t

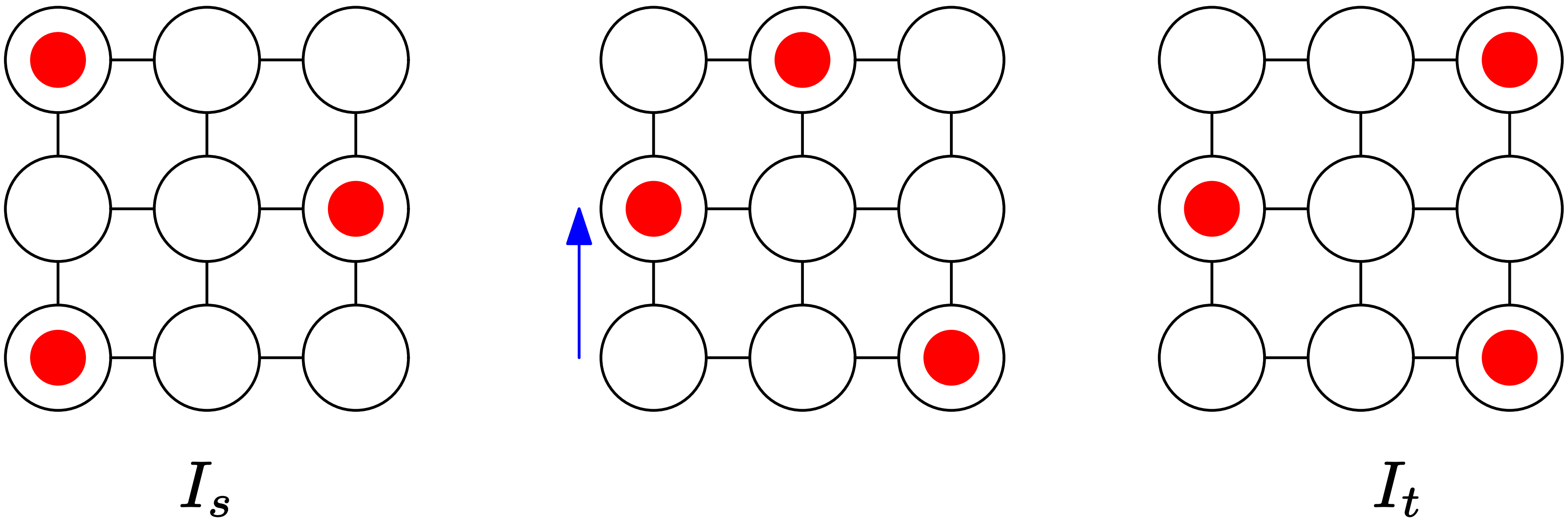
$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

復習：独立集合遷移問題 (PSPACE 完全)

独立集合遷移問題

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



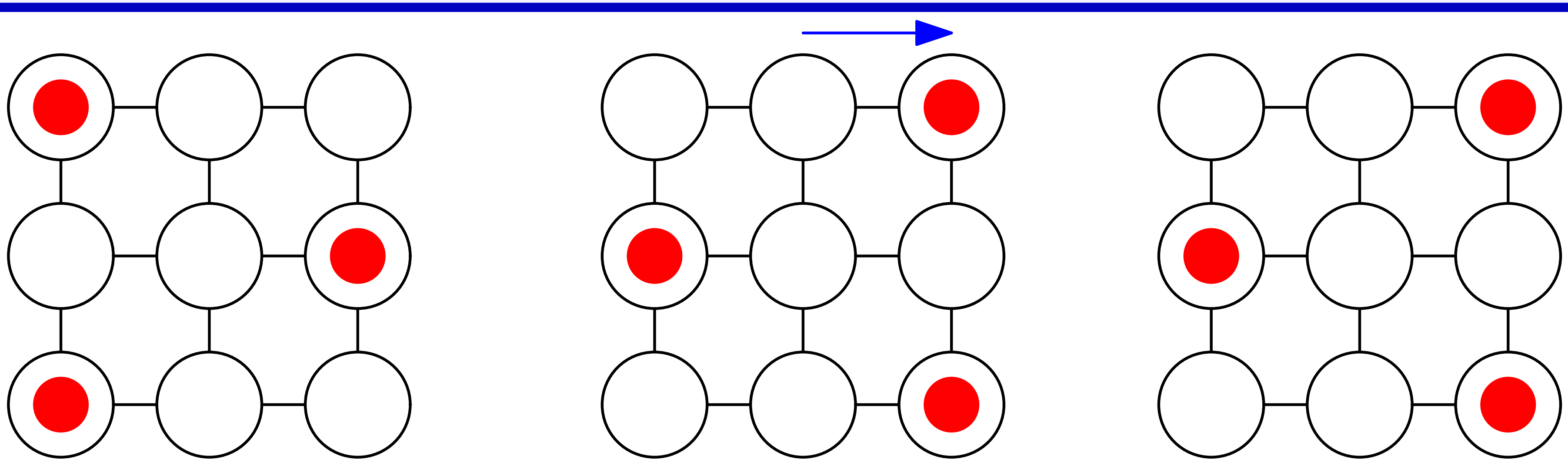
$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

復習：独立集合遷移問題 (PSPACE 完全)

独立集合遷移問題

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



I_s I_t

$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

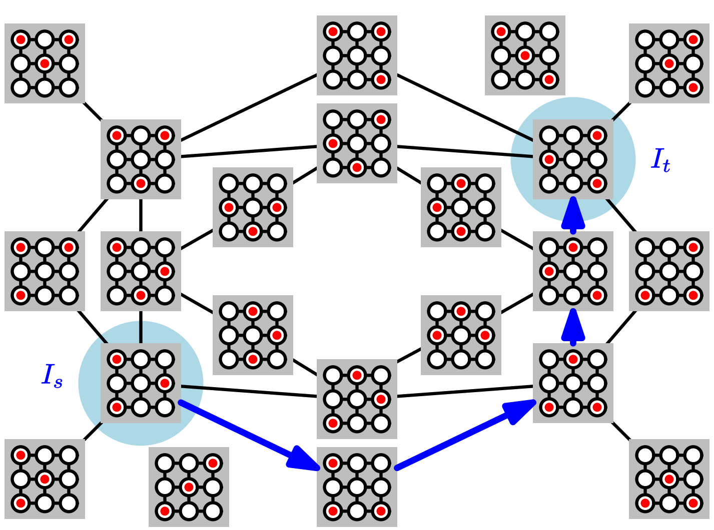
PSPACE ⊆ AP (2)

次の問題を $k = 2^{|V|}$ のとき交代性多項式時間で解ければ十分

ステップ数限定独立集合遷移問題

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$,
非負整数 $k \geq 0$

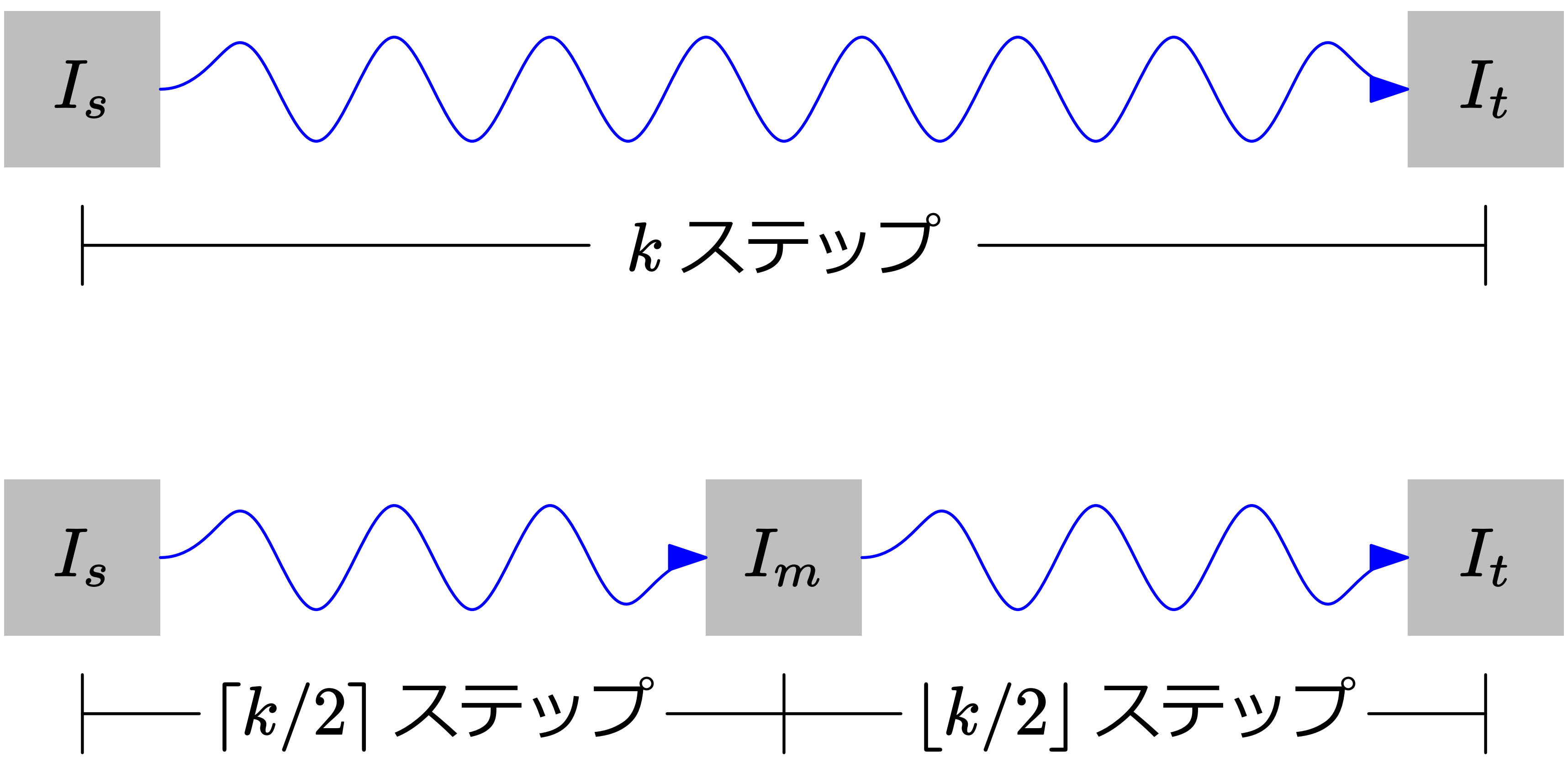
出力： I_s から I_t に k ステップ以内で遷移可能 $\Rightarrow$ Yes
そうではない $\Rightarrow$ No



PSPACE ⊆ AP (3)

ステップ数限定独立集合遷移問題：解法のアイディア

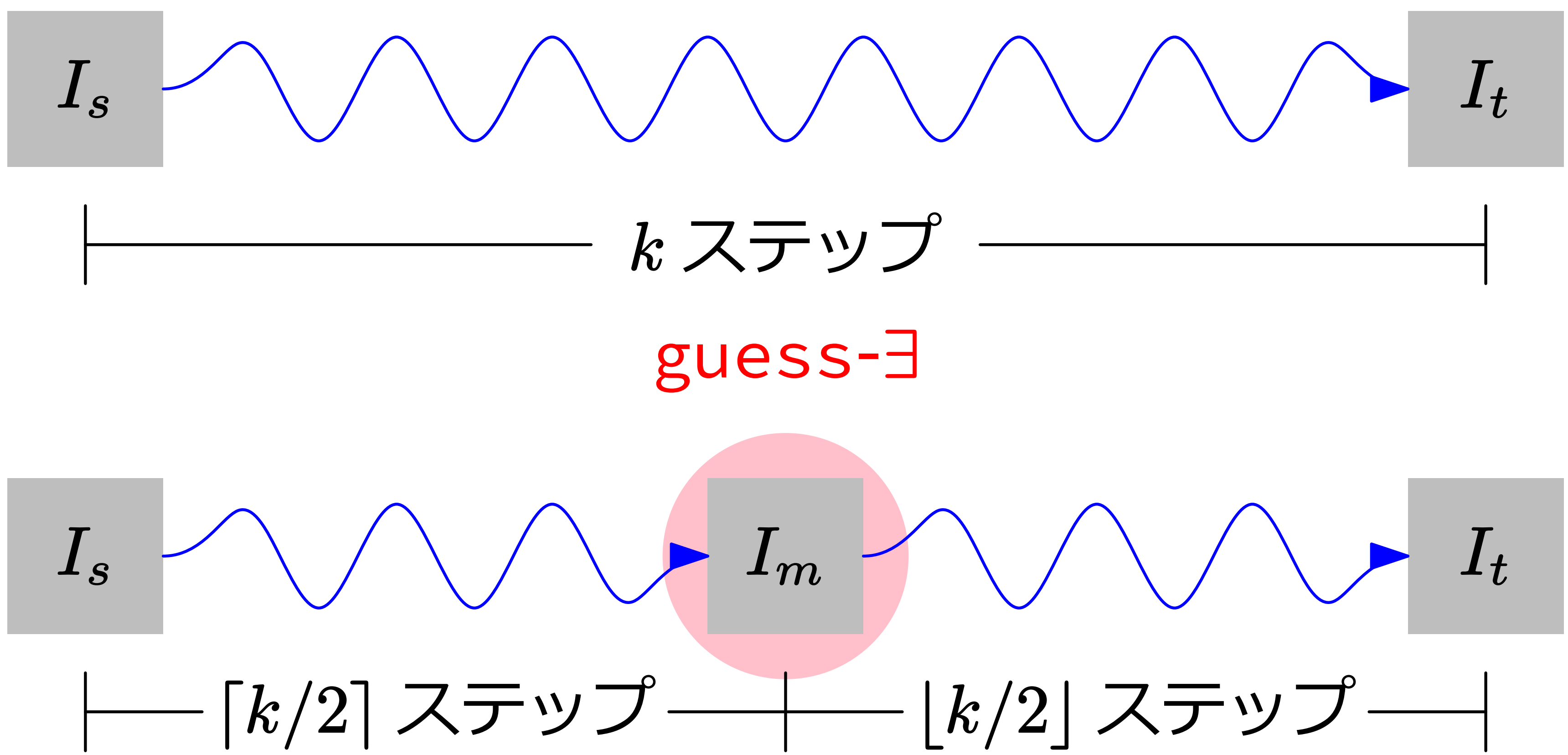
k ステップの遷移を
$\lceil k/2 \rceil$ ステップの遷移と $\lfloor k/2 \rfloor$ ステップの遷移に分ける



PSPACE $\subseteq$ AP (3)

ステップ数限定独立集合遷移問題：解法のアイディア

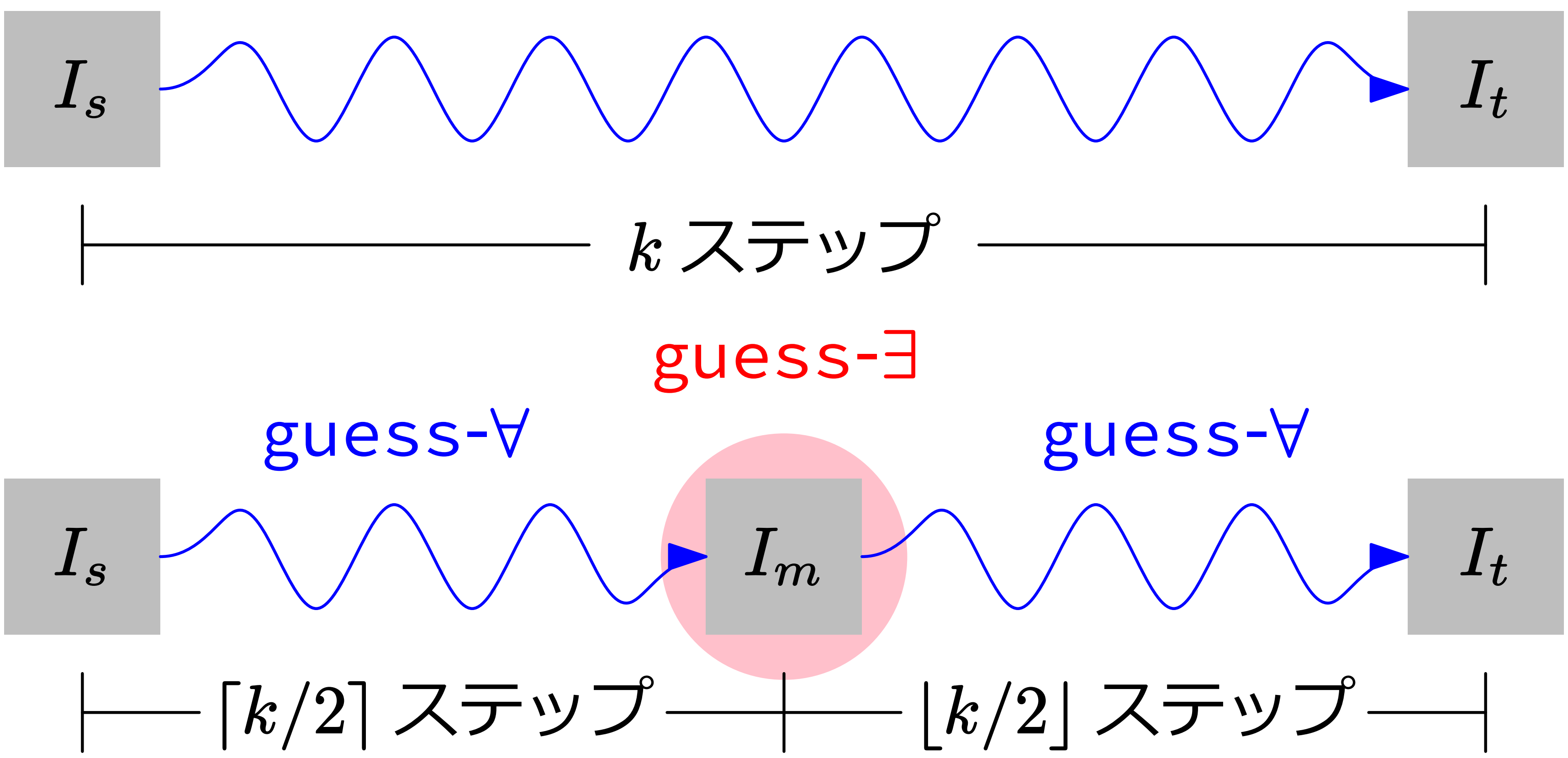
k ステップの遷移を
$\lceil k/2 \rceil$ ステップの遷移と $\lfloor k/2 \rfloor$ ステップの遷移に分ける



PSPACE $\subseteq$ AP (3)

ステップ数限定独立集合遷移問題：解法のアイディア

k ステップの遷移を
$\lceil k/2 \rceil$ ステップの遷移と $\lfloor k/2 \rfloor$ ステップの遷移に分ける

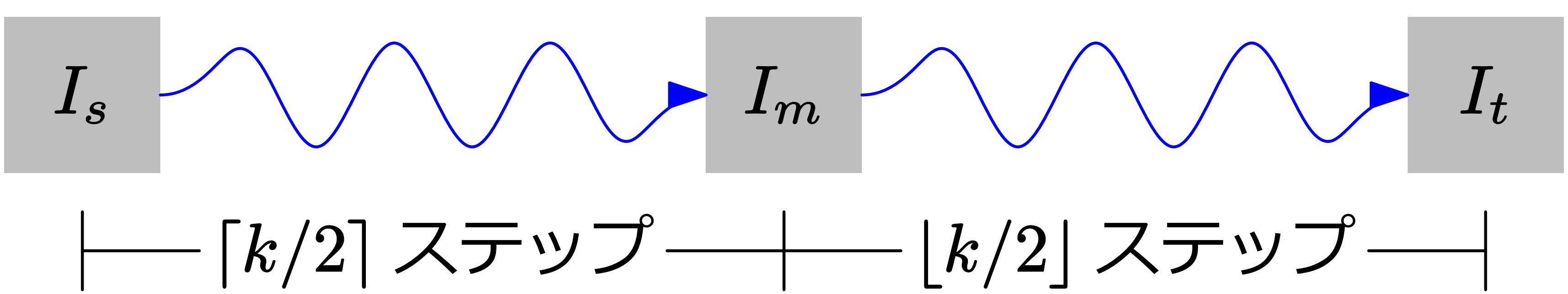


PSPACE $\subseteq$ AP (4)

アルゴリズム $A(I_s, I_t, k)$

1. $k \leq 1$ のとき，解くのは簡単
2. $k \geq 2$ のとき，
   $I_m :=$ guess-∃(すべての独立集合)
3. $i :=$ guess-∀($\{0, 1\}$)
   (a) $i = 0$ のとき，$A(I_s, I_m, \lceil k/2 \rceil)$ を実行
   (b) $i = 1$ のとき，$A(I_m, I_t, \lfloor k/2 \rfloor)$ を実行

これが正しい出力をすることは 今までの議論から分かる



PSPACE ⊆ AP (4)

アルゴリズム $A(I_s, I_t, k)$ 　　時間計算量を $t(k)$ とする

1. $k \leq 1$ のとき，解くのは簡単
2. $k \geq 2$ のとき，
   $I_m :=$ guess-∃(すべての独立集合)
3. $i :=$ guess-∀($\{0, 1\}$)
   (a) $i = 0$ のとき，$A(I_s, I_m, \lceil k/2 \rceil)$ を実行
   (b) $i = 1$ のとき，$A(I_m, I_t, \lfloor k/2 \rfloor)$ を実行

$t(0) = t(1) = O(|V|)$，$t(k) = O(|V|) + t(\lceil k/2 \rceil)$ if $k \geq 2$

$$\begin{aligned} t(2^{|V|}) &= O(|V|) + t(\lceil 2^{|V|}/2 \rceil) \\ &= \underbrace{O(|V|) + O(|V|) + \cdots + O(|V|)}_{O(|V|)\text{ 個}} = O(|V|^2) \end{aligned}$$

∴ これは交代性多項式時間アルゴリズム

AP = PSPACE：まとめ

性質 (Chandra, Kozen, Stockmeyer '86)

AP = PSPACE

つまり，「交代性多項式時間 = 決定性多項式領域」

同様に，AEXPTIME = EXPSPACE も正しい

標語的にいうと

時間$(f(n))$ + 交代性 = 領域$(f(n))$

性質 (Chandra, Kozen, Stockmeyer ’86)

AL $=$ P

つまり，「交代性対数領域 $=$ 決定性多項式時間」

同様に，APSPACE $=$ EXPTIME も正しい

証明：次の 2 つを証明すればよい

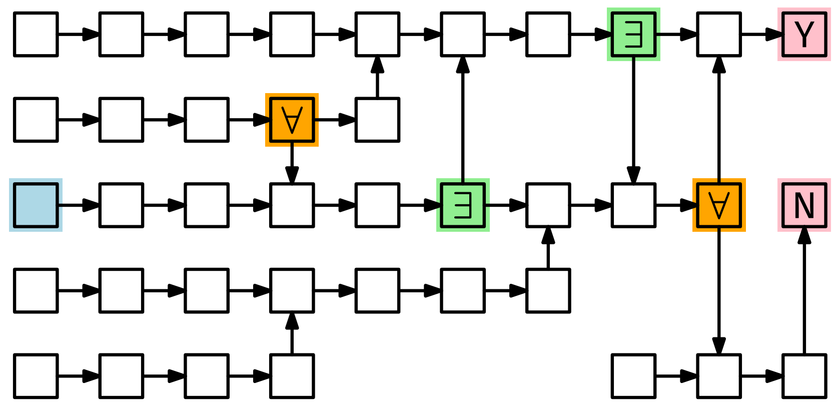
- AL $\subseteq$ P
- P $\subseteq$ AL

標語的にいうと

領域$(f(n))$ $+$ 交代性 $=$ 時間$(2^{f(n)})$

AL $\subseteq$ P

判定問題 P が交代性 $k \log n$ 領域で解けるとき，
ありうる計算状況の総数は $O(n^k)$



<u>アルゴリズム</u>

1. 計算状況のグラフを明示的に作成する
2. 計算状況のグラフ上で探索を行う
   (注：経路の数は多項式で収まらないかもしれない)

これは P を解く多項式時間アルゴリズム

P ⊆ AL (1)

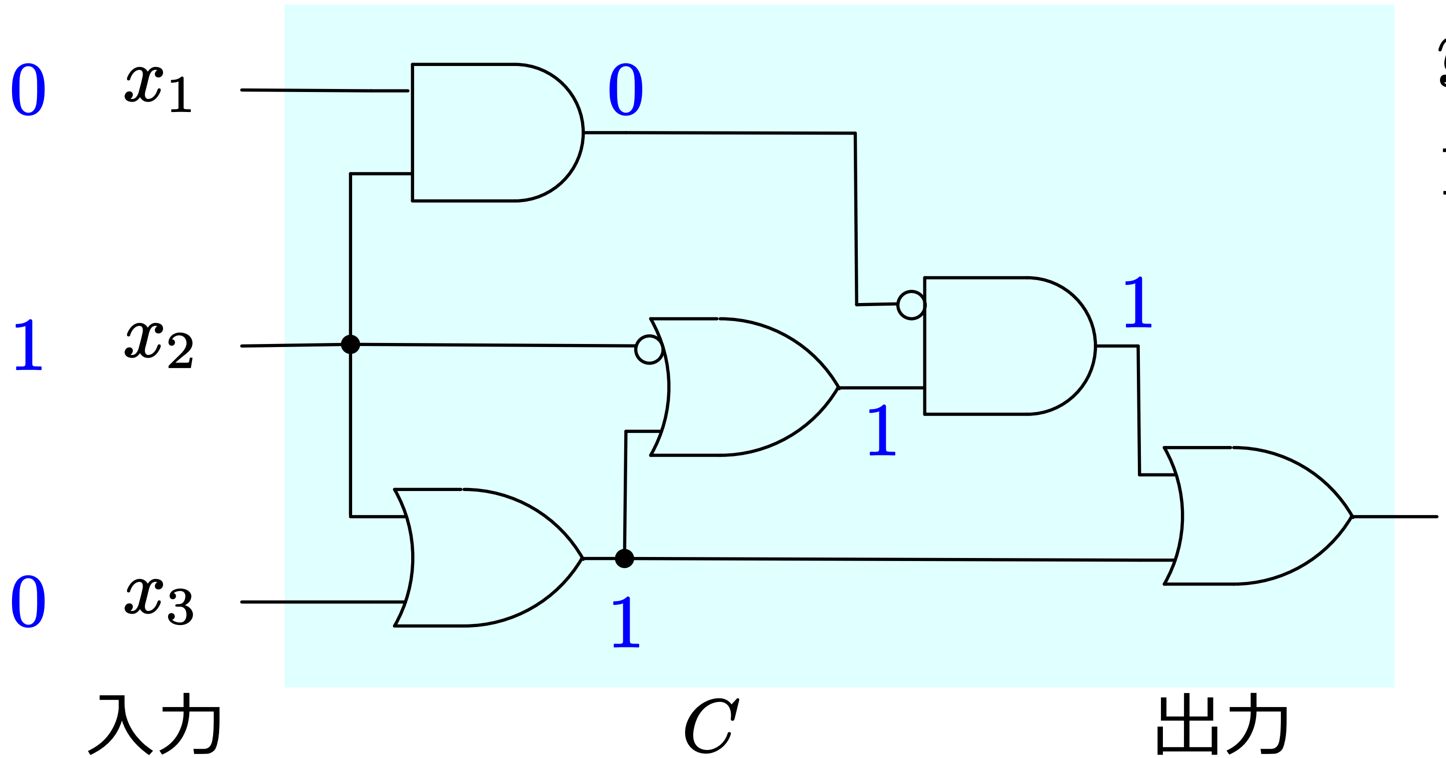
次の P 完全問題に対して，交代性対数領域アルゴリズムを作る

問題：論理回路評価問題 (circuit value problem)

入力： 論理回路 C，割当 α

出力： $C(\alpha) = 1 \Rightarrow$ Yes

$C(\alpha) = 0 \Rightarrow$ No



$y = C(x_1, x_2, x_3)$

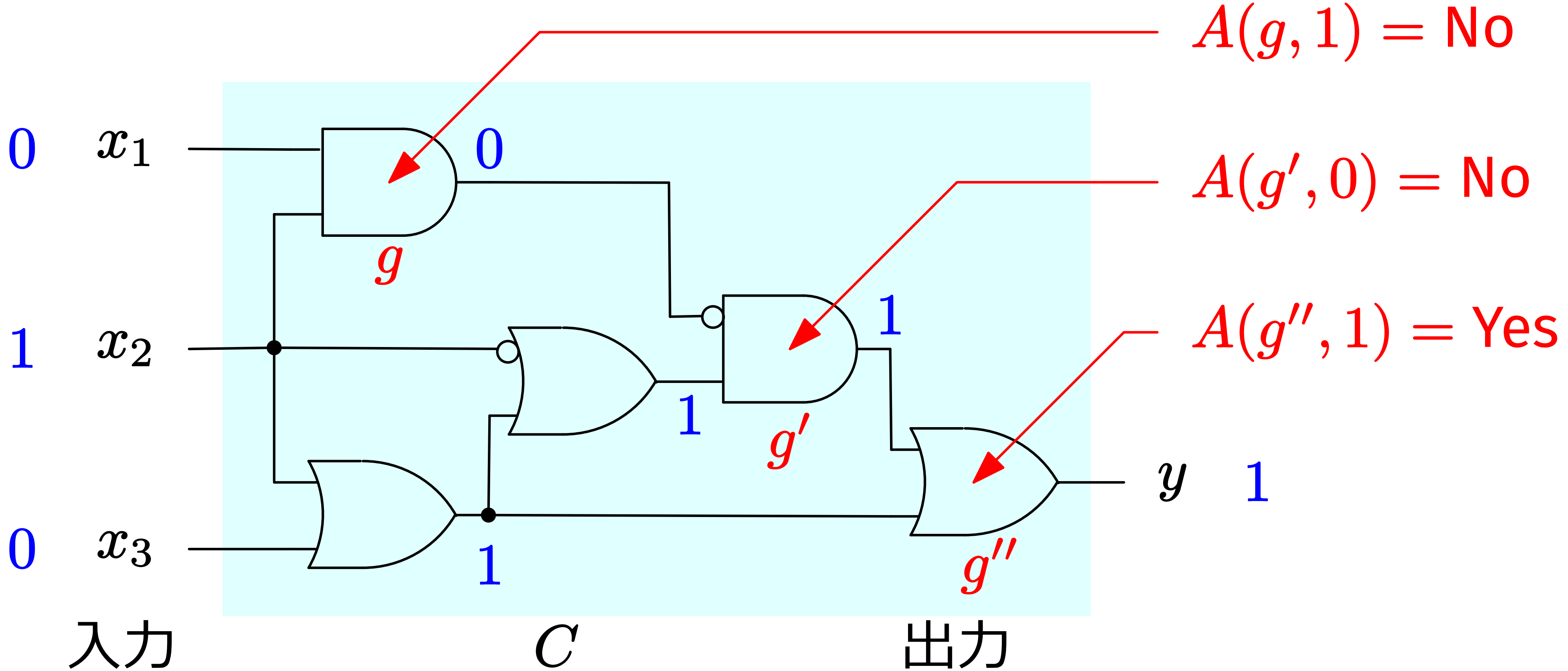
$1 = C(0, 1, 0)$

P ⊆ AL (2)

アルゴリズム A は再帰に基づく

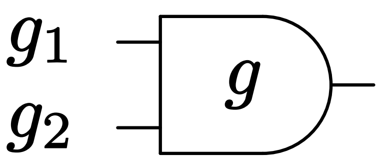
仕様：$A(g, b) =$ Yes $\Leftrightarrow$ 素子 g の出力が b である

- 回路 C に現れる論理素子 g
- 真理値 $b \in \{0, 1\}$



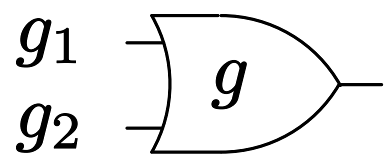
$A(g,b)$ の実装：素子 g の種類によって場合分け

g が AND 素子



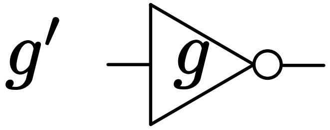
1. $i :=$ guess-$\forall(\{1,2\})$
2. $A(g_i, b)$ を実行

g が OR 素子



1. $i :=$ guess-$\exists(\{1,2\})$
2. $A(g_i, b)$ を実行

g が NOT 素子



1. $A(g', 1-b)$ を実行

g が入力

x_i

1. $x_i = 1 \Rightarrow$ Yes を出力
   $x_i = 0 \Rightarrow$ No を出力

AL = P : まとめ

性質 (Chandra, Kozen, Stockmeyer '86)

AL = P

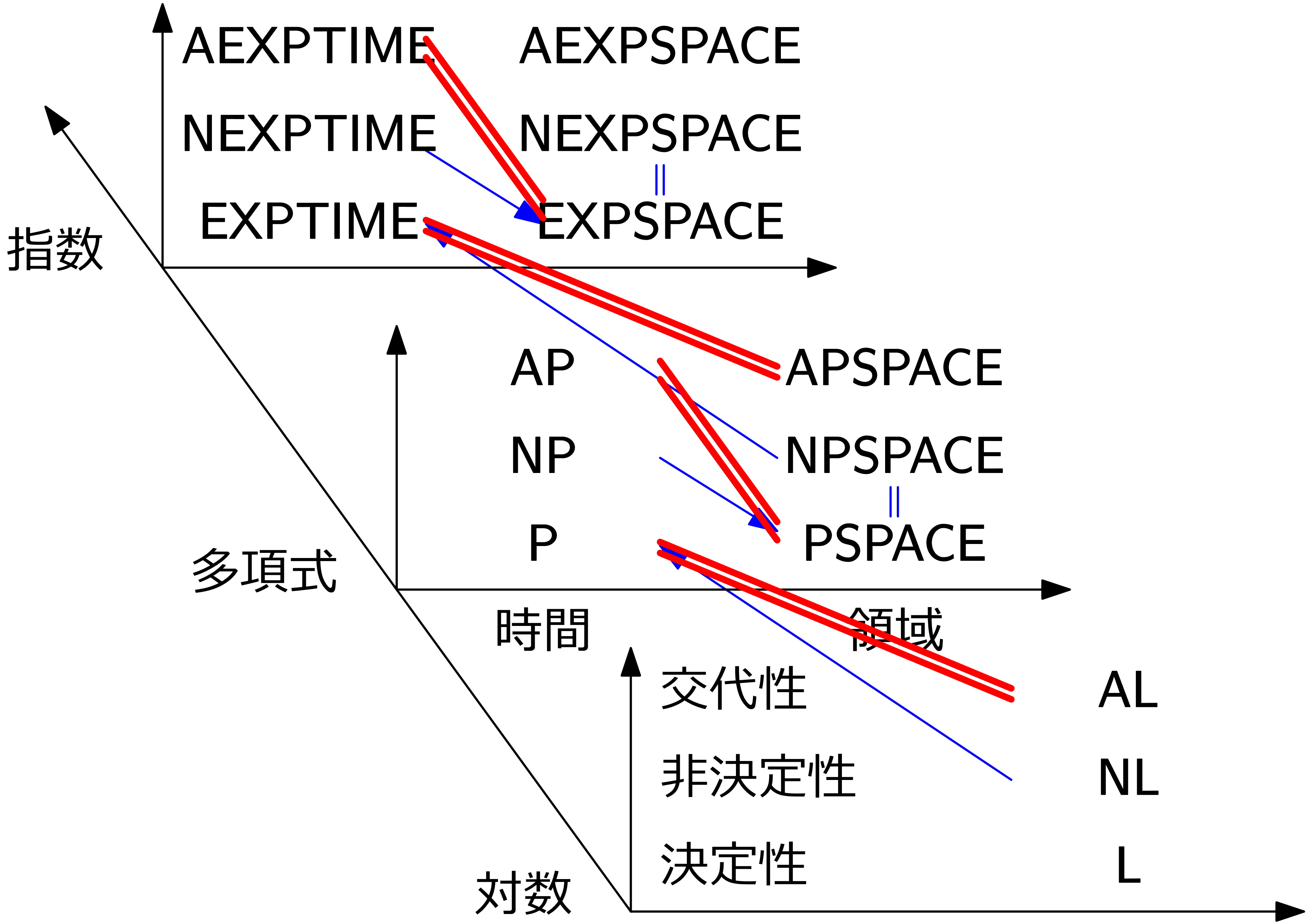
つまり，「交代性対数領域 = 決定性多項式時間」

同様に，APSPACE = EXPTIME も正しい

標語的にいうと

領域$(f(n))$ + 交代性 = 時間$(2^{f(n)})$

再掲：フロアごと？に書くと



本日の内容

1. 交代性計算とは？
2. AP = PSPACE と AL = P
3. **QBF：交代性から見た** PSPACE **完全問題**
4. 難しい問題の例

PSPACE 完全問題の気分 (2)

気分 (1) (第 8 回講義)

「圧縮された STCON」は PSPACE 完全になりがち

例：独立集合遷移問題

気分 (2) (今回)

「NP 完全問題 + 交代性」は PSPACE 完全になりがち

例：QBF (量化論理式問題，限量論理式問題)

標語的に言うと，「QBF = SAT + 交代性」

QBF

問題：QBF (量化論理式問題，限定論理式問題)

入力： 命題論理式 $\varphi(X)$ (X は変数集合)
X の分割 $X_1, X_2, \ldots, X_k$

出力： $\exists X_1 \, \forall X_2 \, \ldots \forall X_k \colon \varphi(X) = 1 \Rightarrow$ Yes
$\forall X_1 \, \exists X_2 \, \ldots \exists X_k \colon \varphi(X) = 0 \Rightarrow$ No

QBF $=$ Quantified Boolean Formula problem

($\exists, \forall$ を quantifier (量化子，限定子，限量子) と呼ぶ)

注 1：SAT は $X = X_1$ の場合に対応

注 2：X_k の前は「$\exists$」でもよい (k は偶数でも奇数でもよい)

性質 (証明しない) (Stockmeyer, Meyer ’73)

QBF は PSPACE 完全

QBF：例

問題：QBF (量化論理式問題，限定論理式問題)

入力： 命題論理式 $\varphi(X)$ (X は変数集合)
X の分割 $X_1, X_2, \ldots, X_k$

出力： $\exists X_1 \, \forall X_2 \, \ldots \forall X_k \colon \varphi(X) = 1 \Rightarrow$ Yes
$\forall X_1 \, \exists X_2 \, \ldots \exists X_k \colon \varphi(X) = 0 \Rightarrow$ No

例：$\varphi(x, y, z) = (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$

$$X = \{x, y, z\}, \quad X_1 = \{x\}, X_2 = \{y\}, X_3 = \{z\}$$

問題：QBF (量化論理式問題，限定論理式問題)

入力： 命題論理式 $\varphi(X)$ (X は変数集合)

X の分割 $X_1, X_2, \ldots, X_k$

出力： $\exists X_1 \, \forall X_2 \, \ldots \forall X_k \colon \varphi(X) = 1 \Rightarrow$ Yes

$\forall X_1 \, \exists X_2 \, \ldots \exists X_k \colon \varphi(X) = 0 \Rightarrow$ No

例：$\varphi(x, y, z) = (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$

$$X = \{x, y, z\}, \quad X_1 = \{x\}, X_2 = \{y\}, X_3 = \{z\}$$

2 人ゲームだと思うとよい

- $x = 1$ とする $\rightsquigarrow \varphi(1, y, z) = \overline{y} \vee \overline{z}$

問題：QBF (量化論理式問題，限定論理式問題)

入力： 命題論理式 $\varphi(X)$ (X は変数集合)

X の分割 $X_1, X_2, \ldots, X_k$

出力： $\exists X_1 \, \forall X_2 \, \ldots \forall X_k \colon \varphi(X) = 1 \Rightarrow$ Yes

$\forall X_1 \, \exists X_2 \, \ldots \exists X_k \colon \varphi(X) = 0 \Rightarrow$ No

例：$\varphi(x, y, z) = (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$

$$X = \{x, y, z\}, \quad X_1 = \{x\}, X_2 = \{y\}, X_3 = \{z\}$$

2 人ゲームだと思うとよい

- $x = 1$ とする $\rightsquigarrow \varphi(1, y, z) = \overline{y} \vee \overline{z}$
  - $y = 0$ とする $\rightsquigarrow \varphi(1, 0, z) = 1$
  - $y = 1$ とする $\rightsquigarrow \varphi(1, 1, z) = \overline{z}$

問題：QBF (量化論理式問題，限定論理式問題)

入力： 命題論理式 $\varphi(X)$ (X は変数集合)
X の分割 $X_1, X_2, \ldots, X_k$

出力： $\exists X_1 \, \forall X_2 \ \ldots \forall X_k \colon \varphi(X) = 1 \Rightarrow$ Yes
$\forall X_1 \, \exists X_2 \ \ldots \exists X_k \colon \varphi(X) = 0 \Rightarrow$ No

例：$\varphi(x, y, z) = (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$

$$X = \{x, y, z\}, \quad X_1 = \{x\}, X_2 = \{y\}, X_3 = \{z\}$$

2 人ゲームだと思うとよい

- $x = 1$ とする $\rightsquigarrow \varphi(1, y, z) = \overline{y} \vee \overline{z}$
  - $y = 0$ とする $\rightsquigarrow \varphi(1, 0, z) = 1$
  - $y = 1$ とする $\rightsquigarrow \varphi(1, 1, z) = \overline{z}$
    - $z = 0$ とする $\rightsquigarrow \varphi(1, 1, 0) = 1$ $\quad \Rightarrow$ 出力は Yes

問題：QBF (量化論理式問題，限定論理式問題)

入力： 命題論理式 $\varphi(X)$ (X は変数集合)
X の分割 $X_1, X_2, \ldots, X_k$

出力： $\exists X_1 \, \forall X_2 \, \ldots \forall X_k : \varphi(X) = 1 \Rightarrow$ Yes
$\forall X_1 \, \exists X_2 \, \ldots \exists X_k : \varphi(X) = 0 \Rightarrow$ No

QBF を解く交代性多項式時間アルゴリズム

1. X_1 の割当を guess-∃
2. X_2 の割当を guess-∀

⋮

k. X_k の割当を guess-∀

*. 得られた割当に対して，$\varphi(X)$ の値を評価して，
評価した値が 1 ならば Yes，0 ならば No を出力

HEX 再訪

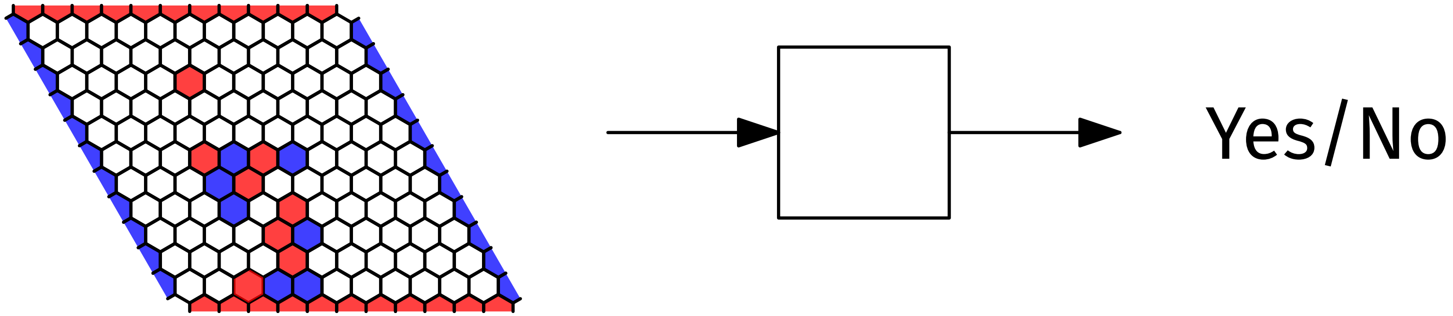
問題 HEX

入力： $n \times n$ の HEX の途中盤面

出力： 赤が必勝 $\Rightarrow$ Yes

青が必勝 $\Rightarrow$ No

既知：HEX では，どちらかが必勝 (引き分けにならない)



性質 (証明しない) (Reisch '81)

HEX は PSPACE 完全

「HEX $\in$ AP $=$ PSPACE」はいままでの議論から分かる

ゲームとパズルの計算複雑性

完全情報の場合 (盤面，手札など，すべてわかっている)

終局までの長さ	多項式で収まる	指数で収まる
各局面の符号長	多項式で収まる	多項式で収まる
1 人ゲーム (パズル)	NP (例：数独)	PSPACE (例：倉庫番)
交代性の付与		
2 人ゲーム	PSPACE (例：HEX)	

ゲームとパズルの計算複雑性

完全情報の場合 (盤面，手札など，すべてわかっている)

終局までの長さ	多項式で収まる	指数で収まる
各局面の符号長	多項式で収まる	多項式で収まる
1 人ゲーム (パズル)	NP (例：数独)	PSPACE (例：倉庫番)
↓ 交代性の付与		
2 人ゲーム	PSPACE (例：HEX)	EXPTIME (例：チェス)

本日の内容

1. 交代性計算とは？
2. AP = PSPACE と AL = P
3. QBF：交代性から見た PSPACE 完全問題
4. **難しい問題の例**

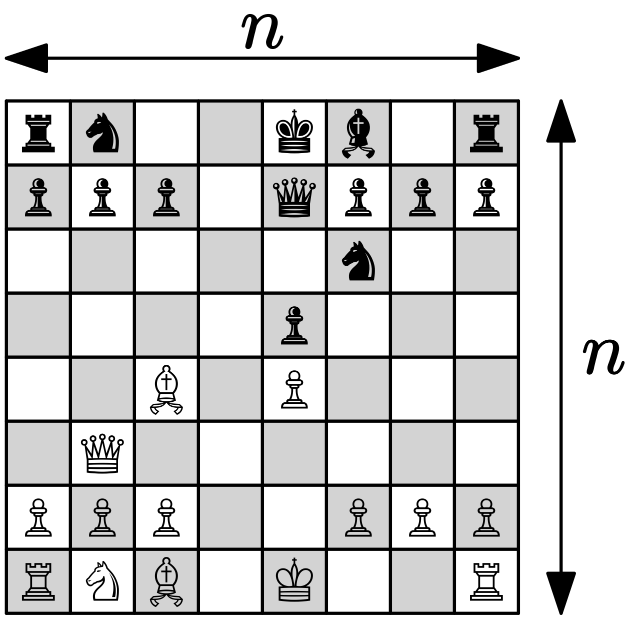
判定問題としてのチェス

問題：チェス

入力： $n \times n$ のチェスの途中局面
出力： 先手が必勝 $\Rightarrow$ Yes
それ以外 $\Rightarrow$ No

注意 1：チェスでは，引き分けがありうる

注意 2：3 回同一局面ルールや 50 手ルールはないとする



オペラゲーム (1858) より

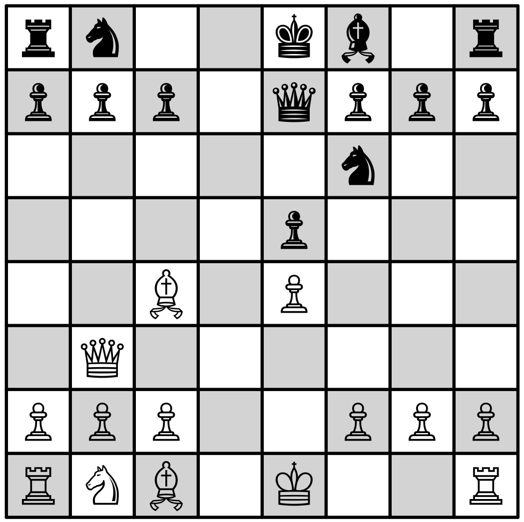
チェスの計算複雑性

問題：チェス

入力： $n \times n$ のチェスの途中局面
出力： 先手が必勝 $\Rightarrow$ Yes
それ以外 $\Rightarrow$ No

性質 (証明しない) (Fraenkel, Lichtenstein ’81)

チェスは EXPTIME 完全



オペラゲーム (1858) より

チェスの計算複雑性

問題：チェス

入力： $n \times n$ のチェスの途中局面

出力： 先手が必勝 $\Rightarrow$ Yes

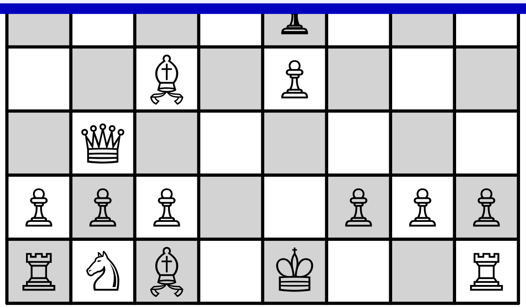
それ以外 $\Rightarrow$ No

性質 (証明しない) (Fraenkel, Lichtenstein ’81)

チェスは EXPTIME 完全

階層定理 (P $\neq$ EXPTIME) より

チェスは多項式時間で解けない



オペラゲーム (1858) より

完全情報の場合 (盤面，手札など，すべてわかっている)

終局までの長さ	多項式	指数
各局面の符号長	多項式	多項式
1 人ゲーム (パズル)	NP (数独)	PSPACE (倉庫番)
↓ 交代性の付与		
2 人ゲーム	PSPACE (HEX)	EXPTIME (チェス)

パズルとゲームの計算複雑性 (続)

完全情報の場合 (盤面，手札など，すべてわかっている)

終局までの長さ	多項式	指数	指数
各局面の符号長	多項式	多項式	指数
1 人ゲーム (パズル)	NP (数独)	PSPACE (倉庫番)	NEXPTIME
↓ 交代性の付与			
2 人ゲーム	PSPACE (HEX)	EXPTIME (チェス)	EXPSPACE

本日のまとめ

本日の内容

- **交代性計算** という計算モデルを導入する
- AP = PSPACE，AL = P という関係を証明し，交代性を使って計算複雑性を調べる方法を紹介する
- PSPACE 完全問題の別の例として QBF を導入する
- 交代性の視点から，パズルとゲームの計算複雑性を調査する

次回の予告

内容

- 交代性が限定された計算モデルを導入する
- 限定交代性を使って多項式階層 PH を導入する
- $\mathrm{P} = \mathrm{NP}$ と多項式階層の関係を証明する

クイズ

Q

1.

2.

3.

4.