数理科学広域演習 IV レポート

高橋 和音 *

2017 年 1 月 5 日

概要

SamurAI coding 2016–17 において，索敵処理を SAT Solver で実装する．本レポートでは，制約をいかに変数と論理式に変換し SAT Solver の入力に直したかを解説する．

1 はじめに

1.1 SamurAI Coding 2016–17 の概要

SamurAI Coding は情報処理学会が主催するプログラミングコンテストである．2012 年度以降，毎年度開催されている．その開催理由は，公式サイトに以下の通りに記されている [情報 c].

> 情報処理学会は若い世代から将来第一線の研究者や開発者になりうる，また世界市場を舞台に活躍できる人材を育てることを目的として，(中略) 今年度も SamurAI Coding 2016–17 を開催いたします．ゲームをテーマにした人工知能およびプログラミングのスキルを競い，(中略) 参加エンジニア・プログラマはその能力が世界で通用するか本コンテストを通じて試すことができます．

* 東京大学大学院 数理科学研究科 数理科学専攻 博士後期課程 2 年．Email: kazune@ms.u-tokyo.ac.jp

本年度は任天堂のゲーム「スプラトゥーン」を模したゲームを題材とした AI プログラミングコンテストである [情報 b].

> SamurAI Coding はプログラマが AI プログラミングの技倆を競うコンテストです．コンテストでは，それぞれ 3 人のサムライからなるふたつの軍団が *SamurAI 3x3 2016* (サムライ スリー オン スリー 2016) という名のゲームを戦い，各 AI プログラムは同一軍団の 3 人のサムライを制御します．

SamurAI Coding 2016–17 のルールは公式ドキュメント [情報 a] に記されている．このレポートでも，このドキュメントの用語は断り無く使用する．

1.2　SAT Solver との関係

SAT Solver は，論理式を与えると，それを真にする変数の組み合わせが存在するかしないかを調べ，仮に存在するならばその変数の組み合わせを 1 通り与えるというものである．詳しくは次節で定義する．$k \geq 3$ のとき，k-SAT 問題は NP 完全問題であることは有名である．近年では，複数の工夫により SAT Solver の高速化が進んでいる．また，SAT Solver を使う側の視点でも恩恵がある．問題を一度論理式に直せば，その問題の効率的な解法がわからずとも SAT Solver に与えれば高速に解が得られる．汎用的な様々な応用も研究されている．これらの事情は [wek16]，[Knu15] に詳しい．

今回は，SAT Solver の AI プログラミングコンテストへの応用を目指す．SamurAI Coding 2016–17 の AI を組む上で様々な処理が必要であるが，そのうち，索敵部分の処理を SAT Solver で実装する構想を練った．本レポートでは，敵の行動とフィールドの様々な制約をいかに変数と論理式に変換し SAT Solver の入力に直したかを解説する．

数学的な工夫として，「直前の行動開始地点と行動終了地点が違う」という条件を SAT Solver の入力に直そうとする際に，$O\left(2^{R^2}\right)$ 個の節を入力する必要に迫られたが，別の制約を駆使して $O(R^2)$ 個へと簡約化した．後述の (3.5) と

(3.6) を参照されたい．

1.3　数理科学広域演習 IV との関係

数理科学広域演習 IV で，本実装を解説するドキュメント，すなわちこのレポートを整備することとした．ドキュメントを整備した目的は，次のとおりである．

通常のプログラムでは，様々なデータ構造やアルゴリズムを利用するため，ある程度手だれた者であれば処理内容に見当がつくであろう．しかし，本構想を実装したプログラムは，入力から論理式をつくり， SAT Solver へ与え解かせ，その出力を処理するだけである．また，その論理式へ至る思考の多くも，初等的とはいえ数学的な内容となっている．したがって，完成されたプログラムだけを読んでも，せいぜい論理式を把握できるだけで，「意味」を理解するのは極めて困難であると考えられる．そこで，解説をするドキュメントを整備することとした．

2016 年度の数理科学広域演習 IV では，「ドキュメント」というテーマで各自が活動した．私は，実習時間を使って，本実装を考案し本ドキュメントを整備した．

2　準備

論理式を記述する際に使われる記号を述べ，以降で必要な考察をする．本レポートを通し，命題論理では二重否定則を採用する．「無限」は全く議論する必要がないので，選択公理は採用しても採用しなくとも同じである．

2.1　定義

記号 2.1.　x, y を命題とする．

1.　「x の否定」を $\neg x$ と書く．

2. 「x かつ y」のことを $x \wedge y$ と書き，「x または y」のことを $x \vee y$ と書く．
3. 「x ならば y」のことを $x \Rightarrow y$ と書く．

記号 2.2.　I を集合とし，$i \in I$ に対し x_i は命題とする．

1. 「任意の $i \in I$ に対し x_i である」のことを

$$\bigwedge_{i \in I} x_i$$

と書く．$I = n \in \mathbb{N}$ が有限集合であるとき，

$$\bigwedge_{i \in I} x_i = x_0 \wedge x_1 \wedge \cdots \wedge x_{n-1}$$

である．

2. 「$i \in I$ が存在し x_i である」のことを

$$\bigvee_{i \in I} x_i$$

と書く．$I = n \in \mathbb{N}$ が有限集合であるとき，

$$\bigvee_{i \in I} x_i = x_0 \vee x_1 \vee \cdots \vee x_{n-1}$$

である．

定義 2.3.　1.　x が真か偽の値を取る変数であるとき，x を単に**変数**という．

2. x を変数とする．このとき，x の**リテラル**とは，命題 x，および，命題 $\neg x$ のことである．
3. **節**とは，有限個のリテラル $l_1, \ldots, l_n$ を用いて

$$\bigvee_{i=1}^{n} l_n = l_1 \vee l_2 \vee \cdots \vee l_n$$

と表される命題である．この節の長さを n と定義する．

定義 2.4.　$x_1, \ldots, x_n$ を変数とする．

1. SAT Solver の**入力**は，$x_1, \ldots, x_n$ のリテラルから作られた有限個の節 $C_1, \ldots, C_m$ を用いて

$$X = \bigwedge_{i=1}^{m} C_i = C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

と表される命題である．$C_1, \ldots, C_m$ の長さがすべて k であるとき，X を ***k*-SAT 問題**という．

2. X を入力した際の SAT Solver の**出力**は，次の (a), (b) のいずれかである．
   (a) 偽である．この場合は，X を真にする $(x_1, \ldots, x_n)$ の真偽値の組が存在しないことを表す．
   (b) 真である．この場合は，X を真にする $(x_1, \ldots, x_n)$ の真偽値の組が存在することを表す．この場合, 更に追加で, X を真にする $(x_1, \ldots, x_n)$ の真偽値の組を 1 つ返す．

2.2　「ならば」の変換

SAT Solver に投げる命題に条件を落とし込むときに，「ならば」で発想すると自然に記述できるように思う．ところが「ならば」は SAT Solver には直接入力はできない．そこで「ならば」の変換に必要な補題を用意する．

補題 2.5.　a, b, c を変数とする．以下が成立する．

$$(a \wedge b) \vee c = (a \vee c) \wedge (b \vee c).$$

証明.　もっとかっこいい方法があるかもしれないが，私は (a, b, c) の真偽 8 パターンを確かめた．a が真のときと $(a, b, c) =$ (真, 真, 偽) のときは両辺真，残りは偽である．■

系 2.6.　a, b を変数とする．以下が成立する．

$$a \Rightarrow b = (\neg a \vee b).$$

証明. $a \Rightarrow b$ は $\neg a \vee (a \wedge b)$ と同値である．あとは補題 2.5 を用いると，次式がしたがう．

$$\neg a \vee (a \wedge b) = (\neg a \vee a) \wedge (\neg a \vee b) = (\neg a \vee b). \qquad \blacksquare$$

3 索敵アルゴリズムの実装

以下では，$R = 14$ とする．

3.1 用語の定義

まず，考察しやすいように，以下では「戦場」「領有」という用語を使用しない．スプラトゥーンに習って用語を置き換える．

定義 3.1. 1. 戦場を**床**と読み替える．
2. 区画の領有のことを，床を**塗る**と表現する．
3. 誰が区画を領有しているかを示すために，**色**という表現を使用する．例えば，敵の色，敵軍の色，味方の色などという．

以下では，次の前提を置く．

注意 3.2. 1. SAT Solver を使って武将の位置を特定を試みるのは，以下の (i), (ii) の条件をみたすときおよび，そのときのみとする．
   (i) 直前のターンの床の状況と現在の床の状況を比べた際に，ある地点の色が敵軍の武将の色に変化したことを確認した．
   (ii) その武将の現在の位置が不明である．
2. 自分の全てのターンで，上記の SAT Solver 使用有無の判定を行う．以下では， SAT Solver 使用有無の判定が行われた結果，使用することが決定したと仮定し，考察をすすめる．
3. SAT Solver の出力を見て，武将の位置を特定することを目標とする．

考察に必要な用語を定義する．

定義 3.3.　1.　変化後の色を持った武将を，以下では単に**敵**と呼ぶ．

2.　敵が直前の 1 ターンでした一連の動作を**行動**という．この定義は定義 3.7 の直前まで有効であるものとする．
3.　敵が直前の 1 ターンを開始した際にいた位置を**行動開始地点**という．敵が直前の 1 ターンを終了した際にいた位置を**行動開始地点**という．
4.　敵が直前の 1 ターンで塗りを行った位置を**塗った位置**という．

今の制約では，塗った位置はちょうど 1 箇所である．詳しくは命題 3.19 で述べる．

以下では方角を自然数で表現する．

注意 3.4.　方角は，以下の通りに自然数に対応させる．0: 南，1: 東，2: 北，3: 西．

3.2　行動した「地点」に関する入力

ここでは，行動開始地点，行動終了地点，塗った地点に関する基本的な条件を落とし込む．実際に床を見てそれらを決定するプロセスは次小節にまわす．

まずは，行動地点に関する変数を導入する．

記号 3.5.　$0 \leq i, j \leq R$ とする．$k = 0, 1, 2, 3$ とする．

1.　s_{ij} を「i 行 j 列が行動開始地点ならば真，そうでないならば偽」とする．
2.　e_{ij} を「i 行 j 列が行動終了地点ならば真，そうでないならば偽」とする．
3.　d_{ijk} を「i 行 j 列方角 k で塗ったならば真，そうでないならば偽」とする．

注意 3.6.　1 ターン内にできる敵の行動は，顕現・隠伏の違いを除けば真である s_{ij}，e_{ijk}，d_{ijk} の組み合わせに対応する．

注意 3.6 でも指摘した通り，(s, e, d) の組み合わせは，ルール上の意味での「敵の 1 ターンの行動」に対応する．そこで，「行動」という用語を定義し直す．

定義 3.7.　1.　以降，**行動**とは真である s_{ij}，e_{ij}，d_{ijk} の組み合わせをいう．特にどの元が真であるのかを問題としない場合は (s, e, d) とも記す．

2.　ある行動が**行動可能**であるとは，その行動が 1 ターンのコスト内で行動可能であることをいう．

最初に，当然だが，どこかの地点で行動開始，どこかの地点で塗り，どこかの地点で行動終了していないといけない．これらの条件はそれぞれ

$$(\mathrm{Allow})_s = \bigvee_{i=0}^{R} \bigvee_{j=0}^{R} s_{ij},$$

$$(\mathrm{Allow})_e = \bigvee_{i=0}^{R} \bigvee_{j=0}^{R} e_{ij},$$

$$(\mathrm{Allow})_d = \bigvee_{i=0}^{R} \bigvee_{j=0}^{R} \bigvee_{j=0}^{3} d_{ijk}$$

と表せる．

次に，これも当然なことだが，2 つ以上のマスが行動開始地点であってはならない．このことは SAT Solver に明示的に入力される必要がある．つまり，全ての $(i, j) \neq (k, l)$ に対し，$\neg(s_{ij} \wedge s_{kl})$ である．すなわち，

$$(\mathrm{Forbid})_s = \bigwedge_{(i,j)\neq(k,l)} \neg(s_{ij} \wedge s_{kl})$$

である．これを SAT の入力に合うように変形すると，

$$(\mathrm{Forbid})_s = \bigwedge_{(i,j)\neq(k,l)} (\neg s_{ij} \vee \neg s_{kl})$$

である．これは e, d も同様である．すなわち，

$$(\mathrm{Forbid})_e = \bigwedge_{(i,j)\neq(k,l)} (\neg e_{ij} \vee \neg e_{kl}),$$

$$(\mathrm{Forbid})_d = \bigwedge_{(i,j,k)\neq(l,m,n)} (\neg e_{ijk} \vee \neg e_{lmn})$$

である．

次に，移動はせいぜい 1 マスしかできないことを条件に加える．k, l はそれぞれ 0 以上 R 以下とする．しばらく，点 (k, l) から 0 または 1 しか離れていないマス (i, j) について $\bigvee$ をとったものを $\bigvee_{i,j}$ と表記することにする．素朴には，x_{ij} を変数としたとき，

$$\bigvee_{i,j} x_{ij} = x_{k(l+1)} \lor x_{(k+1)l} \lor x_{k(l-1)} \lor x_{(k-1)l}$$

のことだが，(k, l) が端のマスだったら (i, j) の幾つかの点は実際には床の外かもしれない．その部分は実装の際に適宜外すことにする．

さて，行動開始地点が点 (k, l) のとき，行動終了地点がせいぜい 1 マスしか離れていないことは，任意の (k, l) に対して

$$s_{kl} \Rightarrow \left(\bigvee_{i,j} e_{ij} \right)$$

であると書ける．補題 2.6 より，これは

$$\neg s_{kl} \lor \left(\bigvee_{i,j} e_{ij} \right)$$

と同値である．(k, l) について $\bigwedge$ をとって，

$$(\text{start-end}) = \bigwedge_{k=0}^{R} \bigwedge_{l=0}^{R} \left(\neg s_{kl} \lor \left(\bigvee_{i,j} e_{ij} \right) \right)$$

が求める条件である．

次に塗った位置と方角についての条件を書く．つまり「塗った位置は，開始地点か終了地点である」ということである．すなわち，任意の (i, j, k) に対し，

$$d_{ijk} \Longrightarrow s_{ij} \lor e_{ij}$$

である．補題 2.6 より，これは

$$\neg d_{ijk} \lor (s_{ij} \lor e_{ij})$$

と同値である．すなわち，条件は

$$(\text{drawpoint}) = \bigwedge_{i=0}^{R} \bigwedge_{j=0}^{R} \bigwedge_{k=0}^{3} (\neg d_{ijk} \vee s_{ij} \vee e_{ij})$$

である．

注意 3.8.　ここまでの制約をまとめると，以下の通りである．

- 行動開始地点と行動終了地点がただ 1 つ存在する．
- 行動開始地点と行動終了地点はせいぜい 1 マスしか離れていない．
- 塗った地点と塗った方向がただ 1 つ存在する．
- 塗った地点は，行動開始地点または行動終了地点である．

3.3　塗った地点の制約

ここから先は，各ターン毎の床の状況を鑑みて，d_{ijk} に制限を加える．この小節では，$0 \leq i, j \leq R$，および，$k = 0, 1, 2, 3$ は固定する．

床の状況を SAT Solver に入力する変数を用意する．

記号 3.9 (塗りのための床からの制約変数).　$0 \leq l, m \leq R$ とする．

1. α_{lm} を「点 (l, m) が知覚されており，かつ，現在敵の色で塗られているなら真，そうでないなら偽」とする．より具体的に言うと，以下のいずれかの場合は真である．
   - 点 (l, m) が敵の色で塗られていると知覚されている場合．

   また，以下のいずれかの場合は偽である．
   - 点 (l, m) が味方軍の色，あるいは，敵以外の敵軍の色で塗られていると知覚されている場合．
   - 点 (l, m) が誰にも塗られていないと知覚されている場合．
   - 点 (l, m) がそもそも知覚されていない場合．
2. β_{lm} を「点 (l, m) が知覚されており，かつ，現在敵の色で塗られていない

なら真，そうでないなら偽」とする．より具体的に言うと，以下のいずれかの場合は真である．

- 点 (l, m) が味方軍の色，あるいは，敵以外の敵軍の色で塗られていると知覚されている場合.
- 点 (l, m) が誰にも塗られていないと知覚されている場合.

また，以下のいずれかの場合は偽である．

- 点 (l, m) が敵の色で塗られていると知覚されている場合.
- 点 (l, m) がそもそも知覚されていない場合.

3. γ_{lm} を「点 (l, m) を敵が直前に塗ったことが確実ならば真，そうでないなら偽」とする．つまり，現在敵の色で塗られているものの敵の行動の直前は敵の色で塗られていない場合に真とし，それ以外は偽とする．
4. δ_{lm} を「点 (l, m) を敵が直前に塗らなかったことが確実ならば真，そうでないなら偽」とする．つまり，現在敵の色以外で塗られており，敵の行動の直前も同じ色で塗られていた場合は真とし，それ以外は偽とする．

注意 3.10.　記号 3.9 の $\{\alpha_{lm}, \beta_{lm}, \gamma_{lm}, \delta_{lm}\}$ は全て SAT Solver に真か偽かを入力するものとする．例えば，α_{lm} が真である場合には 1 つのリテラルからなる節 α_{lm} を，偽である場合には 1 つのリテラルからなる節 $\neg\alpha_{lm}$ を入力する．

注意 3.11.　考察を進めた結果，$\{\alpha_{lm}\}$ は使用しなかった．

さて，塗ったときの敵の地点と方向は d_{ijk} で表現されているが，この情報を塗ったマスの情報に予めおきかえておくことが以下では必要になる．つまり，以下で定義する変数 g_{lm} を媒介して，d_{ijk} ではなく g_{lm} で制約を記述する．

記号 3.12 (塗った地点の情報を媒介する変数).
1. 敵が (i, j) の地点で方角 k に塗ったときに塗られる地点の集合を D_{ijk} とする．
2. $0 \leq l, m \leq R$ とする．g_{lm} を「点 (l, m) が敵が直前の行動で塗った地点ならば真，そうでないならば偽」とする．

注意 3.13.　ここでは敵は固定されているので，D_{ijk} は (i, j, k) の組により一意

に定まる．

注意 3.14.　g_{lm} は，どの d_{ijk} が成立するかで真か偽かが決まるのであり，床の情報とは関係がないことに注意されたい．g_{lm} と床の情報 β_{lm}，γ_{lm}，δ_{lm} との整合性の条件式を以下で与える．これにより，g を媒介として床の情報から d に制約を与えることができる．

さてまず，d_{ijk} の情報を g_{lm} の情報に置き換える条件式を書く．これは記号 3.12 より，

$$d_{ijk} \Rightarrow \left(\bigwedge_{(l,m)\in D_{ijk}} g_{lm}\right) \wedge \left(\bigwedge_{(l,m)\notin D_{ijk}} \neg g_{lm}\right)$$

である．系 2.6 より，

$$\neg d_{ijk} \vee \left(\left(\bigwedge_{(l,m)\in D_{ijk}} g_{lm}\right) \wedge \left(\bigwedge_{(l,m)\notin D_{ijk}} \neg g_{lm}\right)\right)$$

と同値である．補題 2.5 及び (i,j,k) が固定されていたことを考慮すると，求める条件は

$$\text{(param-trans)} = \bigwedge_{i,j,k} \left(\left(\bigwedge_{(l,m)\in D_{ijk}} (\neg d_{ijk} \vee g_{lm})\right) \wedge \left(\bigwedge_{(l,m)\notin D_{ijk}} (\neg d_{ijk} \vee \neg g_{lm})\right)\right) \tag{3.1}$$

である．

次に，β_{ij}，γ_{ij}，δ_{ij} と g_{ij} の関係式を導く．次の関係がある．

1. 敵が直前に塗り替えたことが確実な点は，敵が直前の行動で塗った点である．
2. 敵が直前の行動で塗った点は，敵の色で塗られた点，または，知覚できていない点である．
3. 敵が直前に塗り替えなかったことが確実な点は，敵が直前の行動で塗った点ではない．

言うまでもなくこれらは，任意の (l, m) に対して以下が成立することを意味している．

1. $\gamma_{lm} \Rightarrow g_{lm}$.
2. $g_{lm} \Rightarrow \neg\beta_{lm}$.
3. $\delta_{lm} \Rightarrow \neg g_{lm}$.

補題 2.5，系 2.6 より，以下が得られる．

$$\begin{aligned}
(\text{field})_\gamma &= \bigwedge_{l=0}^{R} \bigwedge_{m=0}^{R} (\neg\gamma_{lm} \vee g_{lm}), \\
(\text{field})_\beta &= \bigwedge_{l=0}^{R} \bigwedge_{m=0}^{R} (\neg g_{lm} \vee \neg\beta_{lm}), \\
(\text{field})_\delta &= \bigwedge_{l=0}^{R} \bigwedge_{m=0}^{R} (\neg\delta_{lm} \vee \neg g_{lm}).
\end{aligned}$$

3.4　隠遁・顕現の条件

前小節までで，隠遁と顕現の条件以外の制約を考慮に入れた．本小節では，隠遁と顕現の条件からありえない組み合わせを排除し，その行動が，定義 3.7.2. で定義した行動可能であることを保証する．

まずは，SAT Solver に入力する変数を用意する．

記号 3.15.　π_0 を，敵の開始地点が判明していないならば真，そうでないならば偽とする．

注意 3.16.　ゲーム中は，自分のターン毎に敵の位置が判明しているか否かが与えられる．現在の敵の位置，すなわち，直前の敵の行動の行動終了地点が判明している場合は，そもそも SAT Solver を発動する必要はない．直前の敵の行動の行動開始地点が判明している場合は，顕現動作を省略できるので，SAT Solver には 1 つのリテラルからなる節 $\neg\pi_0$ を入力する．また，直前の敵の行動の行動開始地点を (i, j) とするとき，1 つのリテラルからなる節 s_{ij} を入力する．直前

の敵の行動の行動開始地点が判明していない場合は，SAT Solver には 1 つのリテラルからなる節 π_0 を入力する．

記号 3.17 (隠伏・顕現のための床からの制約変数).　$0 \le l, m \le R$ とする．

1. κ_{lm} を「点 (l, m) が知覚されており，かつ，現在敵軍の色で塗られているなら真，そうでないなら偽」とする．より具体的に言うと，以下のいずれかの場合は真である．
   - 点 (l, m) が敵軍の色で塗られていると知覚されている場合．

   また，以下のいずれかの場合は偽である．
   - 点 (l, m) が味方軍の色で塗られていると知覚されている場合．
   - 点 (l, m) が誰にも塗られていないと知覚されている場合．
   - 点 (l, m) がそもそも知覚されていない場合．
2. κ'_{lm} を 1 ターン前の κ_{lm} とする．最初のターンでは全て偽とする．
3. λ_{lm} を「点 (l, m) が知覚されており，かつ，現在敵軍の色で塗られていないなら真，そうでないなら偽」とする．より具体的に言うと，以下のいずれかの場合は真である．
   - 点 (l, m) が味方軍の色で塗られていると知覚されている場合．
   - 点 (l, m) が誰にも塗られていないと知覚されている場合．

   また，以下のいずれかの場合は偽である．
   - 点 (l, m) が敵軍の色で塗られていると知覚されている場合．
   - 点 (l, m) がそもそも知覚されていない場合．
4. λ'_{lm} を 1 ターン前の λ_{lm} とする．最初のターンでは全て偽とする．

記号 3.17 で導入した変数についても，注意 3.10 に準ずる．

ここまでの情報から，敵の行動開始地点に制約がかかる．つまり，敵の行動開始地点が不明であるのに，知覚できていて敵軍の色で塗られていない箇所が敵の行動開始地点であることはない．すなわち，任意の (i, j) に対し，以下が成立する．

$$(\pi_0 \wedge \lambda'_{ij}) \Rightarrow \neg s_{ij}.$$

系 2.6 を用いると，これは以下と同値である．

$$\neg\pi_0 \vee \neg\lambda'_{ij} \vee \neg s_{ij}.$$

よって，以下が成立する．

$$(\text{start-hidden}) = \bigwedge_{i=0}^{R} \bigwedge_{j=0}^{R} \left(\neg\pi_0 \vee \neg\lambda'_{ij} \vee \neg s_{ij}\right).$$

同様に，行動終了地点についても，以下が成立する．

$$(\text{goal-hidden}) = \bigwedge_{i=0}^{R} \bigwedge_{j=0}^{R} \left(\neg\lambda_{ij} \vee \neg g_{ij}\right).$$

隠遁と顕現の条件からありえない組み合わせを排除するために，以下の変数を用意する．

記号 3.18.　1.　r_0 を，開始地点で隠伏状態であるならば真，そうでないならば偽とする．

2.　r_1 を，終了地点で隠伏状態であるならば真，そうでないならば偽とする．

3.　r_2 を，開始地点と終了地点が異なっているならば真，そうでないならば偽とする．

命題 3.19.　ある行動が行動可能であるための条件は，以下の節が真であることである．

$$\neg r_0 \vee \neg r_1 \vee \neg r_2. \tag{3.2}$$

証明.　ある行動がコスト 7 以内である条件を書き下せばよい．今回の SAT Solver を発動させる前提条件として，敵の行動の前後で土地の所有に変化があることが前提条件になっている．そこで，「領有」の動作が少なくとも 1 回は行われている．領有は 4 コストであるから，ちょうど 1 回行われている．

したがって，残りの 3 コストで，「移動」「隠伏と顕現の切り替え」がどれだけできるかに考察が帰着される．隠伏に顕現状態でできないことは存在しないので，以下を仮定して良い．

- 顕現の動作が行われるならば，行動開始時点で行っていると仮定して良い．
- 隠伏の動作が行われるならば，行動終了時点で行っていると仮定して良い．

つまり隠伏と顕現の切り替えは，最大でも 2 回しか行われないとしてよい．また，行動は 2 コストかかるため，最大でも 1 回しか行われない．隠伏と顕現の切り替えは 1 コストである．したがって，3 コストでできないことは，「隠伏と顕現の切り替えを 2 回やり，かつ，移動を 1 回した場合」に限られる．最初の顕現動作があったことは r_0，移動動作があったことは r_2，最後の隠伏動作があったことは r_1 でわかる．(3.2) が行動可能の条件である．■

注意 3.20.　実際には，(3.2) ではなく，以下の節を入力する．

$$(\text{movable}) = \neg r_0 \vee \neg r_1 \vee \neg r_2 \vee \neg \pi_0. \tag{3.3}$$

以下の点に注意されたい．

1. 記号 3.18 で定められた r_0, r_1, r_2 それぞれ，真とも偽とも言えない場合は，SAT Solver 上はその変数は偽として取り扱い，(3.3) を真にする．この場合，命題 3.19 を考慮するに十分な条件がないので，本小節の内容は敵の位置を絞り込む手がかりにはならない．だから，r_0, r_1, r_2 それぞれ，「偽であるための条件」を考察する必要はない．
2. 以下では，直前の敵の行動の行動開始地点・行動終了地点が判明しないものと仮定する．つまり， SAT Solver は発動され，π_0 は真と仮定する．このとき，(3.3) は (3.2) と同値である．以下では，(3.2) を考察する．

記号 3.18.1. の条件を述べる．行動開始地点を (i, j) と仮定する．r_0 は真であるといえる条件は，κ'_{ij} が真である場合であり，またその場合に限る．詳しく言うと以下のとおりである．π_0 が真であるから敵の行動開始地点は認識できていない．その状態で，敵は行動開始地点では隠伏状態であるといえるのは，(i, j) が敵の色と認識できている場合である．以上を踏まえて，条件は任意の (i, j) に

対し，

$$(s_{ij} \wedge \kappa'_{ij}) \Rightarrow r_0$$

である．系 2.6 より，以下を SAT Solver に入力する．

$$(\text{Force})_{r_0} = \bigwedge_{i=0}^{R} \bigwedge_{j=0}^{R} (\neg s_{ij} \vee \neg \kappa'_{ij} \vee r_0).$$

同様に，r_1 については以下のとおりである．

$$(\text{Force})_{r_1} = \bigwedge_{i=0}^{R} \bigwedge_{j=0}^{R} (\neg g_{ij} \vee \neg \kappa_{ij} \vee r_1).$$

記号 3.18.3. の条件は厄介である．着想自体は単純である．行動開始地点と行動終了地点が一致しているとは，ある (i, j) について s_{ij} かつ e_{ij} が成立することである．したがって，記号 3.18.3. の条件は，

$$\neg \left(\bigvee_{i=0}^{R} \bigvee_{j=0}^{R} (s_{ij} \wedge e_{ij}) \right) \Rightarrow r_2$$

である．系 2.6 より，これは

$$\left(\bigvee_{i=0}^{R} \bigvee_{j=0}^{R} (s_{ij} \wedge e_{ij}) \right) \vee r_2$$

である．

$$P_{r_2} = \bigvee_{i=0}^{R} \bigvee_{j=0}^{R} (s_{ij} \wedge e_{ij}) \tag{3.4}$$

とおく．P_{r_2} を SAT Solver の入力に直接直すと，以下のとおりである．

$$P_{r_2} = \bigwedge_{\sigma \in \Sigma} \left(\bigvee_{i=0}^{R} \bigvee_{j=0}^{R} \sigma_{ij} \right). \tag{3.5}$$

ここで，

$$\Sigma = \left\{ \sigma \colon \{0, \ldots, R\}^2 \to \{s_{ij}, e_{ij}\}_{ij} \mid \sigma_{ij} \text{は } s_{ij} \text{ または } e_{ij}. \right\}.$$

さて，(3.5) において，右辺は $2^{(R+1)^2}$ 個の節からなっている．$R = 14$ であるから，これを制限時間内に入力することは不可能である．

ところが，今回の場合，$\{s_{ij}\}$ と $\{e_{ij}\}$ には制約がある．

- $\{s_{ij}\}$ は，あるただ 1 つの (i, j) のみ真であり，他は偽である．
- $\{e_{ij}\}$ も同様である．

これらは，$(\mathrm{Allow})_s$ かつ $(\mathrm{Forbid})_s$ かつ $(\mathrm{Allow})_e$ かつ $(\mathrm{Forbid})_e$ として，SAT Solver に § 3.2 で入力済みである．

補題 3.21.　$(\mathrm{Allow})_s$ かつ $(\mathrm{Forbid})_s$ かつ $(\mathrm{Allow})_e$ かつ $(\mathrm{Forbid})_e$ の仮定のもとで，P_{r_2} は以下の P'_{r_2} と同値である．

$$P'_{r_2} = \bigwedge_{\sigma \in \Sigma'} \left(\bigvee_{i=0}^{R} \bigvee_{j=0}^{R} \sigma_{ij} \right). \tag{3.6}$$

ここで，

$$\Sigma' = \{\sigma \in \Sigma' \mid \text{ある } (i', j') \text{ について } \sigma_{i'j'} = e_{i'j'}. \quad \text{それ以外の } (i, j) \text{ について } \sigma_{ij} = s_{ij}.\}.$$

証明.　$\Sigma' \subset \Sigma$ より，$P_{r_2} \Rightarrow P'_{r_2}$ である．$P'_{r_2} \Rightarrow P_{r_2}$ を示すために，$\neg P_{r_2} \Rightarrow \neg P'_{r_2}$ を示す．

$(\mathrm{Allow})_s$ かつ $(\mathrm{Forbid})_s$ かつ $(\mathrm{Allow})_e$ かつ $(\mathrm{Forbid})_e$ より, ある $(i_1, j_1), (i_2, j_2)$ が存在し，以下が成立する．

$$s_{ij} = \begin{cases} \text{真} & (i, j) = (i_1, j_1), \\ \text{偽} & (i, j) \neq (i_1, j_1), \end{cases}$$

$$e_{ij} = \begin{cases} \text{真} & (i, j) = (i_2, j_2), \\ \text{偽} & (i, j) \neq (i_2, j_2). \end{cases}$$

(3.4) にもどれば，P_{r_2} が偽であることは $(i_1, j_1) \neq (i_2, j_2)$ であることと同値である．

$\sigma \in \Sigma$ を

$$\sigma_{ij} = \begin{cases} e_{ij} & (i, j) = (i_1, j_1), \\ s_{ij} & (i, j) \neq (i_1, j_1) \end{cases}$$

と定める．すると，$\sigma \in \Sigma'$ である．このとき，

$$\bigvee_{i=0}^{R} \bigvee_{j=0}^{R} \sigma_{ij}$$

は偽である．よって P'_{r_2} は偽である．$\neg P_{r_2} \Rightarrow \neg P'_{r_2}$ が示された． ■

(3.6) の右辺の節の個数は $(R + 1)^2 = 225$ であるから，P_{r_2} の代わりに P'_{r_2} を採用することができる．すなわち，簡約化ができたことになる．結局，入力するべき条件は，以下のとおりである．

$$(\mathrm{Force})_{r_2} = \bigwedge_{\sigma \in \Sigma'} \left(\left(\bigvee_{i=0}^{R} \bigvee_{j=0}^{R} \sigma_{ij} \right) \vee r_2 \right).$$

参考文献

[Knu15] Donald E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 6: Satisfiability*. Addison-Wesley Professional, 1 edition, 12 2015.

[wek16] Sat 技術の進化と応用 〜パズルからプログラム検証まで〜. 情報処理, Vol. 57, No. 8, pp. 702–737, jul 2016.

[情報 a] 情報処理学会プログラミングコンテスト委員会. Samurai 3x3 2016 ルールとプログラムインタフェース. `http://samuraicoding.info/rules-jp.pdf`.

[情報 b] 情報処理学会プログラミングコンテスト委員会. Samurai coding のルール. `http://samuraicoding.info/rules-jp.html`.

[情報 c] 情報処理学会プログラミングコンテスト委員会. 情報処理学会 国際 人工知能プログラミングコンテスト: Samurai coding 2016–17. `http://samuraicoding.info/index-jp.html`.