

矩形同士の重ね合わせについての定性空間推論システムの拡張

小西 貴子 高橋 和子

サイズや辺の縦横比が可変な矩形同士の重ね合わせについて、定性空間推論の枠組を使った定式化の拡張について述べる。単位となる矩形は表示したい部分 (WHITE) と隠したい部分 (BLACK) を持ち、これらの構成によって定性的に表現される。このような矩形が複数枚与えられた時、BLACK をすべて隠し、かつ WHITE はすべて表示する重ね合わせ方が存在するための条件および重ね合わせ方を考える。本発表では、矩形内の BLACK, WHITE の形に対する制約条件を緩和した場合のこのような条件や重ね合わせ方について述べ、これまでの定式化を大きく変更することなく、より複雑なパターンをもつ矩形が扱えるように拡張できることを示す。

1 はじめに

PC ユーザは複数のウィンドウを画面上に開いて作業することが多いが、限られた画面上で複数のウィンドウを開くため、ウィンドウの位置や大きさを変更したり、ウィンドウを重ね合わせる必要が頻繁に生じる。その作業のために時間を取られたり、必要な部分がすべて前面にくるような配置にするためには手間がかかる場合も多い。そこで本研究では、マルチウィンドウの配置のように重ね合わせを伴う矩形の配置問題について考える。

限られた空間にオブジェクトを最適配置する問題は詰め込み問題の 1 つとしてこれまで多くの研究がなされており [2], ラベルの配置 [5][8] や、VLSI のレイアウト [7] など様々な分野に応用されている。しかし、一般の最適配置問題では以下の点は考慮されていない。(i) オブジェクトのサイズが可変である。(ii)

オブジェクト同士の重ね合わせが可能である。マルチウィンドウの配置問題では、オブジェクトのサイズや辺の縦横比を変更することで、解が見つかる場合がある。また、必要な部分を表示し不要な部分を隠すような重ね合わせ方を考える必要がある。そのため、従来のオブジェクトの最適配置問題の解法をそのまま応用するのは難しい。

本研究では、定性空間推論 (Qualitative Spatial Reasoning) の枠組み [1] を用いて上の 2 点を考慮したオブジェクトの配置問題に対する推論方法について考察する。

定性空間推論の枠組みでは、オブジェクトのサイズや位置を定量的に数値で表さずに、オブジェクトやオブジェクト同士の相対的な位置関係を定性的に記号表現し、それに対して論理的推論を行う。この枠組みは、サイズが可変なオブジェクトを扱うのに適しているだけでなく、推論過程が明確であり、さらに計算量も抑えられるという利点がある。定性空間推論については人工知能分野を中心に様々な枠組みが提案されている [3][4][9][10] が、重ね合わせまで考慮したものはない。本研究では重ね合わせまで扱えるような定性的表現を提案し、その上で複数のオブジェクト配置に関する推論方法を示す。

本研究では、表示したい部分と隠したい部分から成

An Extension of the Qualitative Spatial Reasoning System on Rectangle Superposition.

Takako Konishi, Kazuko Takahashi, 関西学院大学 理工学研究科, Graduate School of Science & Technology, Kwansei Gakuin University.

連絡先: 関西学院大学大学院理工学研究科情報科学専攻
〒 669-1337 三田市学園 2 丁目 1 番地
E-mail: t.konishi@kwansei.ac.jp

る複数枚の矩形の重ね合わせを考える．矩形は定性的に表現され，矩形の大きさと矩形内の縦横の辺の比は可変に扱うことができる．表示したい部分は全て表示し，かつ隠したい部分はすべて隠す重ね合わせ方を，定性的な表現を用いた枠組みの上で推論する．

重ね合わせの例を図 1 に示す．この図では，黒い部分は隠したい部分，白い部分は表示したい部分を表している．(a) では，まず B の上に A を重ね合わせる．その際，A の大きさを変え，B の下側に重ね合わせる．そして，残った黒い部分に C を重ねることで，黒い部分は全て隠し，白い部分は全て表示する重ね合わせ方ができる．(b) の場合は，どのような重ね合わせ方をしても，黒い部分を全て隠すことができない．

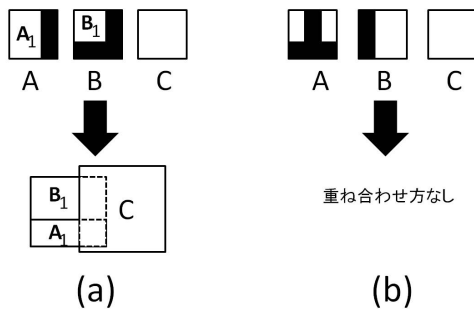


図 1 矩形の重ね合わせ方の例

矩形内の白黒の部分の切り分けのパターンによってオブジェクトを分類して定性的に表現し，それぞれについて重ね合わせ方を考える．著者らはすでに基本的なパターンについては定性的な表現方法を示し，表示したい部分は全て表示し，かつ隠したい部分は全て隠す重ね合わせ方が存在するための条件および重ね合わせについて定式化を行った [6]．基本的なパターンでは，表示したい部分が矩形をしているもののみとしていたが，本発表では，その制約条件をゆるめ，例えば図 2 のようなパターンも対象とできるようにした表現方法と重ね合わせの推論方法について述べる．結果として，[6] で述べた定式化を大きく変更することなく対象が拡張できる．

本論文の構成は以下の通りである．2 章では対象とする図形である *unit* の定義と制約条件について述べ

る．3 章，4 章では *unit* の重ね合わせについて説明する．5 章では本研究のまとめと今後の課題について述べる．

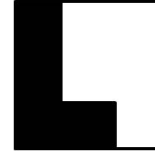


図 2 表示したい部分が矩形でない例

2 対象とするもの

本研究で対象とする図形を *unit* と呼ぶ．*unit* は矩形で，表示させる部分 *WHITE* と隠される部分 *BLACK* に分かれている．*BLACK* は，*core region* と *non-core region* に分かれており，これについては後で述べる．*unit* の外部は *GRAY* と呼ぶことにする．*unit* の大きさ，*unit* 内の縦横の辺の比率は可変である．

離れていない *WHITE* を，それぞれ *white region* と呼び，また *core region*，*non-core region* はそれぞれ *black region* と呼ぶ．*GRAY* は *gray region* と呼ぶ．これらの *white region*，*black region*，*gray region* をそれぞれ '*w*'，'*b*'，'*g*' と記す．

2.1 *unit* の制約条件

本研究では，*unit* を以下の条件のもとに定義する．幅 a ，高さ b の白い矩形があるとき， $x*b(0 \leq x \leq a)$ と $a*y(0 \leq y \leq b)$ で得られる 2 枚の黒いプレートを考える．このプレートを白い矩形に n 枚 ($0 \leq n \leq 2$) 埋め込んで得られるものを *unit* とし，以下のように名付ける．

- (0) プレートを埋め込まない．(W)
- (1) 1 枚だけプレートを埋め込む．(B, I1, I2)
- (2) 2 枚ともプレートを埋め込む．(L1, T1, PLUS)
- (3) L1 と T1 を拡張し，長さ $a(b)$ の辺をもつ黒いプレートの $a(b)$ の辺の外側に， $a*c(b*c)$ (ただし c は定数) で得られる白いプレートを接合す

る . (L2 , T2)

このように定義された *unit* を *basic units* と呼ぶ .
basic units は以下の制約条件を満たす .

- (i) *unit* の全体の形は矩形である .
- (ii) *BLACK* は繋がっており , 離れて存在しない .
- (iii) 全ての *white region* は矩形である .

次に , *unit* を構成する 2 枚のプレートのうち 1 枚の長さを短くすることで , 扱える *unit* の種類を増やす . 2 枚のプレートの定義を変更し , 以下に示す条件を満たす n 枚 ($0 \leq n \leq 2$) のプレートを埋め込むことによって得られる *unit* を考える .

$$\begin{cases} x * b(0 \leq x \leq a) \\ x' * y(x \leq x' \leq a)(0 \leq y \leq b) \\ x * y'(0 \leq x \leq a)(y \leq y' \leq b) \\ a * y(0 \leq y \leq b) \end{cases}$$

これによって , *white region* の形として矩形以外のものも認められ , *white region* が図 3 , 図 4 のようなものも対象となる . 対象は広がるが , 図 4 のような *white region* を扱うと , 既存の表現方法のみでは *unit* を一意的に表すことが難しくなり , 表現方法を根本的に変更する必要がある . *basic units* については , 一意的な提案方法を既に提案しており , 今回は既存の表現方法を保つことを目的としているため , 図 3 に示される形のみに制限する .

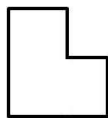


図 3 *white region* の制約条件

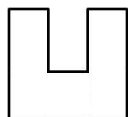


図 4 扱わない *white region* の例

この条件によって得られる *unit* を *new units* という . *new units* として新たに *WI* , *WL1* , *WL2* , *WT1* , *WT2* の 5 つのタイプが得られる .

basic units と *new units* を , *valid* である , という . 全ての *valid unit* を , 図 5 に示す .

I1-type , *I2*-type を *straight-plate-units* , *W*-type , *B*-type , *straight-plate-units* 以外の *unit* を *cross-plates-units* と呼ぶ .

2.2 *unit* の表現方法

W-type 以外の全ての *unit* は , 以下のように *core region* を定義する .

B-type と *straight-plate-units* は , *BLACK* 全体を *core region* とする (図 6) .

cross-plates-units は , 2 つのプレートが交わっている部分を *core region* とする . また , *core region* に含まれない部分を , *non-core region* とする (図 7) .

unit X の *core region* を $Core_x$ と書く .

valid である *unit* は , $Core_x$ から見て , 上 , 右 , 下 , 左方向にある *region* を用いて , 一意的に表現することができる . *basic units* では , *core region* のすぐ隣の *region* の種類 b , w , g で表せばよいが , *new units* に対しては , すぐ隣が *black region* の場合 , その先に *white region* が存在する場合を区別する必要がある . このような場合は $b(w)$ と表す .

例えば , 図 8 の場合 , (a) は $\langle b, b, g, w \rangle$, (b) は $\langle b, b(w), g, w \rangle$ となる . (a) の *unit* は , $Core_x$ から見て , 上方向と右方向には *black region* , 下方向には *gray region* , 左方向には *white region* があることを表している . (b) では , 右方向に *black region* がありその先に *white region* があることを示している .

以下に , *region* の種類 , *unit* の表現の集合 , *unit* のタイプの集合をそれぞれ表す V , R , T を示し , R から T への関数 *type* を定義する .

$$V = \{b, w, g, b(w)\}$$

$$R = \{\langle r_1, r_2, r_3, r_4 \rangle \mid \text{valid である } unit \text{ の表現}\}$$

$$T = \{B', W', I1', I2', L1', L2', T1', T2', PLUS', WI', WL1', WL2', WT1', WT2'\}$$

$$type : R \rightarrow T$$

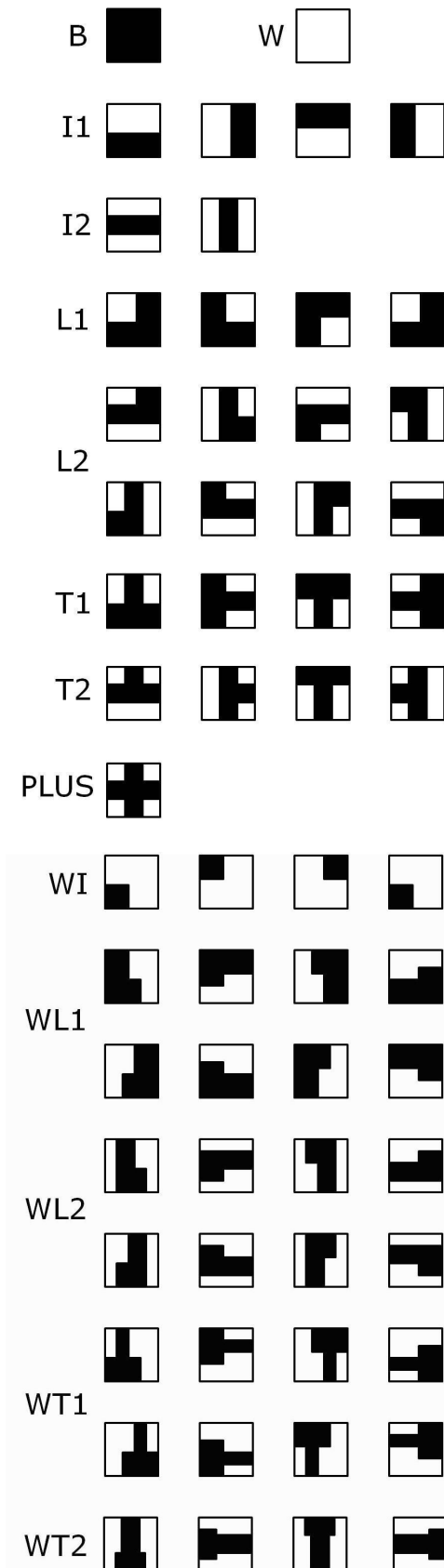


図 5 valid である unit

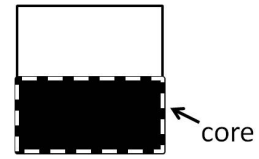


図 6 straight-plate-units の core region

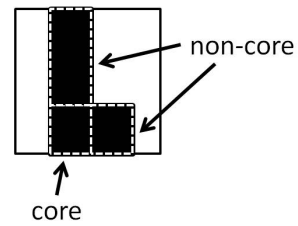


図 7 cross-plates-units の core region

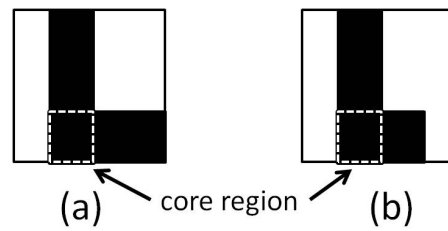


図 8 unit の表現方法

$type(\langle g, g, g, g \rangle)$	= 'B'
$type(\langle w, w, w, w \rangle)$	= 'W'
$type(\langle w, g, g, g \rangle)$	= 'I1'
$type(\langle w, g, w, g \rangle)$	= 'I2'
$type(\langle b, g, g, b \rangle)$	= 'L1'
$type(\langle b, g, w, b \rangle)$	= 'L2'
$type(\langle b, b, g, b \rangle)$	= 'T1'
$type(\langle b, b, w, b \rangle)$	= 'T2'
$type(\langle b, b, b, b \rangle)$	= 'PLUS'
$type(\langle w, w, g, g \rangle)$	= 'WI'
$type(\langle b, b(w), g, g \rangle)$	= 'WL1'
$type(\langle b, b(w), g, w \rangle)$	= 'WL2'
$type(\langle b, b(w), g, b \rangle)$	= 'WT1'
$type(\langle b, b(w), g, b(w) \rangle)$	= 'WT2'

3 unit の重ね合わせ : PUTON

2 枚の *unit* の *core region* 同士を重ね合わせる操作を考える。例えば、図 9 の 2 枚の *unit* にこの操作を行うと、 $Core_x$ の前面に $Core_y$ を重ね合わせ、*unit Z* を得ることができる。この操作では *core region* 同士を重ね合わせるため、この部分が新たに $Core_z$ となり、ここを中心に新たに得られる *unit* は表現される。この場合、新たに得られた *unit Z* は、 $\langle w, b, g, g \rangle$ と表現される。ある *region* にある *region* を重ねた結果得られる *region* を返す関数を *on* とする。

さらに、図 9 の *unit Z* は、BLACK の形が矩形であるにも関わらず、BLACK 全体が *core region* になっていない。そのため、*core region* 同士を重ね合わせたあと BLACK の形が矩形になった場合、BLACK 全体を *core region* にする操作が必要である。この操作を表す関数を *merge* と呼ぶ。*merge* は、*unit* の表現 r が与えられたとき、BLACK の形が矩形の場合は、BLACK 全体を *core region* とした表現を返し、それ以外の場合は r をそのまま返す関数である。例えば、図 9 の *unit Z* に *merge* を行うと、BLACK 全体を *core region* にし、*unit Z* の表現は $\langle w, g, g, g \rangle$ となる。

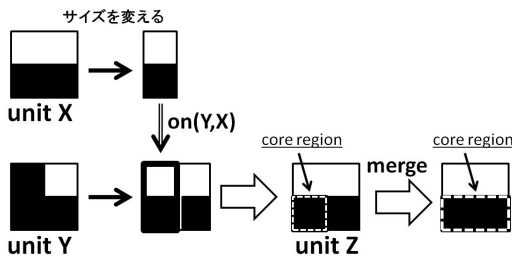


図 9 unit の重ね合わせ : puton

関数 *on*, *merge* はそれぞれ以下のように定義される。

$$on : V \times V \rightarrow V \cup \{fail\}$$

$$\begin{aligned} on(b, b) &= b \\ on(b, w) &= w \\ on(w, w) &= fail \\ on(w, b) &= fail \\ on(b, b(w)) &= b(w) \\ on(w, b(w)) &= fail \\ on(b(w), w) &= w \\ on(b(w), b) &= b(w) \\ on(b(w), b(w)) &= b(w) \\ on(g, v) &= v \text{ where } v \in V \\ on(v, g) &= v \text{ where } v \in V \end{aligned}$$

fail は、*white region* の前面に *gray region* 以外の *region* が重なり、失敗することを意味する。

$r = \langle r_1, r_2, r_3, r_4 \rangle$ とする。

r が $\bigwedge_{i=1, \dots, 4} (r_i \neq fail)$ を満たすとき、*merge* を以下のように定義する。

$$merge(r) = \langle r'_1, r'_2, r'_3, r'_4 \rangle \text{ とする。}$$

ただし、 $r_i = b^*$ は $r_i = b \vee r_i = b(w)$ を表し、 $r'_i = g^*$ は $r_i = b$ のとき g , $r_i = b(w)$ のとき w を表す。

$$merge : V^4 \rightarrow R$$

$$merge(r) = \begin{cases} \langle g^*, r_2, g^*, r_4 \rangle & \text{if } r_1 = b^* \wedge r_2 \neq b \wedge r_3 = b^* \wedge r_4 \neq b \\ \langle r_1, g^*, r_3, g^* \rangle & \text{if } r_1 \neq b \wedge r_2 = b^* \wedge r_3 \neq b \wedge r_4 = b^* \\ \langle g^*, r_2, r_3, r_4 \rangle & \text{if } r_1 = b^* \wedge r_2 \neq b \wedge r_3 \neq b \wedge r_4 \neq b \\ \langle r_1, g^*, r_3, r_4 \rangle & \text{if } r_1 \neq b \wedge r_2 = b^* \wedge r_3 \neq b \wedge r_4 \neq b \\ \langle r_1, r_2, g^*, r_4 \rangle & \text{if } r_1 \neq b \wedge r_2 \neq b \wedge r_3 = b^* \wedge r_4 \neq b \\ \langle r_1, r_2, r_3, g^* \rangle & \text{if } r_1 \neq b \wedge r_2 \neq b \wedge r_3 \neq b \wedge r_4 = b^* \\ \langle r_1, r_2, r_3, r_4 \rangle & \text{otherwise} \end{cases}$$

valid である 2 枚の *unit* を重ね合わせて新しい *unit* をつくる操作を *puton* と呼ぶ。これに対応する関数 *puton* は, *on* と *merge* を使って以下のように定義される。

valid である 2 枚の *unit*, X, Y の表現をそれぞれ $r = \langle r_1, r_2, r_3, r_4 \rangle$, $r' = \langle r'_1, r'_2, r'_3, r'_4 \rangle$ とする。

$puton : R \times R \rightarrow R$

$puton(r, r') =$

$merge(\langle on(r_1, r'_1), on(r_2, r'_2),$
 $on(r_3, r'_3), on(r_4, r'_4) \rangle)$

puton が成功するための条件として, まず (c1) を満たす必要がある。

$$(c1) \bigwedge_{i=1, \dots, 4} on(r_i, r'_i) \neq fail$$

さらに, *cross-plates-units* 同士の重ね合わせを考えると, *core region* の斜め方向の *region* についても考える必要があるため, 以下の (c2) の式を満たす必要がある。

(c2) 以下の式を満たす $i (1 \leq i \leq 4)$ が存在する。

$$r_i = r'_{i+2} = g$$

(ただし, $r'_5 = r'_1$, $r'_6 = r'_2$ とみなす。)

以後, r_i において $i \geq 5$ のときは, $r_i = r_{i-4}$ とみなす。

basic units に関しては, *cross-plates-units* 同士を重ね合わせる場合は (c1) と (c2), それ以外の場合は (c1) を満たせば *puton* が成功することは証明済みである [6]。

複数枚の *unit* の重ね合わせを考えると, 少なくとも 1 枚は W-type でないと解が存在しないことは明らかなので, 必ず 1 枚がそのような *unit* であるとする。今回は, W-type が 2 枚あるような場合は考えないものとする。

ここで, n 枚の *unit* から W-type を除いた $n-1$ 枚の重ね合わせを考える。

$n-1$ 枚の重ね合わせを考えると, 以下の 2 つの場合を考慮する必要がある。

(1) $n-1 = 2$ の場合

(2) $n-1 > 2$ の場合

(1) の場合 2 枚の *unit* を重ね合わせるので, その 2 枚の *unit* を重ね合わせた結果, 前面にくる *BLACK* の形が矩形になれば, 解がある, という。

(2) の場合, $n-1$ 枚のうち 2 枚を重ね合わせ, それにより得られる *unit* が valid であれば, それを新たな *unit* として加え, $(n-1)-1$ 枚の *unit* の重ね合わせとして再帰的に重ね合わせを続けることができる。

以上の (1), (2) の場合を判定する判定式を, 以下で説明する。

3.1 *BLACK* が矩形であるか否かの判定: effectiveness

unit Z が, 離れていない矩形の *BLACK* を 1 つだけでもつとき, *unit* Z は effective であるという。

unit Z がそのような *BLACK* をもつかどうかの判定式を以下に示す。

これは, *core region* の周りに *black region* が存在するか否かを判定している。*BLACK* が矩形ならば, *merge* によって *BLACK* 全体が *core region* になっているため, *core region* の周りに *black region* は存在しない。

$$(c3) \bigwedge_{i=1, \dots, 4} r_i \neq b$$

3.2 *unit* が valid であるか否かの判定: validity

2 枚の *unit* を重ね合わせて得られる新たな *unit* が valid であるか否かを判定する。*unit* が valid であるためには, (i) 全体の形が矩形になる, (ii) *WHITE* が制約条件を満たしている, を判定する必要がある。(i)(ii) をそれぞれ判定する式 (c4)(c5) を以下に示す。

重ね合わせを行う 2 枚の *unit*, X, Y の表現をそれぞれ $r = \langle r_1, r_2, r_3, r_4 \rangle$, $r' = \langle r'_1, r'_2, r'_3, r'_4 \rangle$, X, Y を重ね合わせて得られる *unit* Z の表現を $r = \langle r''_1, r''_2, r''_3, r''_4 \rangle$ とするとき, 以下の式 (c4)(c5) を満たす *unit* Z は valid である。

(c4) $r_i = r'_{i+2} = g$ となる $i(1 \leq i \leq 4)$ が存在し、その i が以下の (1) ~ (4) のいずれかの式を満たす。

- (1) $r_{i+1} = b \wedge r'_{i+1} \neq g \wedge r_{i+3} = g \wedge r'_{i+3} = g$
- (2) $r_{i+1} = g \wedge r'_{i+1} = g \wedge r_{i+3} = b \wedge r'_{i+3} \neq g$
- (3) $r_{i+1} = b \wedge r'_{i+1} \neq g \wedge r_{i+3} = b \wedge r'_{i+3} \neq g$
- (4) $r_{i+1} = g \wedge r'_{i+1} = g \wedge r_{i+3} = g \wedge r'_{i+3} = g$

(c5) 次のいずれかの式を満たす $i(1 \leq i \leq 4)$ は存在しない。

- (1) $r''_i = r''_{i+1} = b(w)$
- (2) $r''_i = r''_{i+1} = r''_{i+2} = w$
- (3) $(r''_i \neq g) \wedge (r''_{i+2} \neq g) \wedge (r''_{i+1} = b(w))$

cross-plates-units の前面に *straight-plate-units* を重ね合わせる場合、解がある重ね方は *puton* しかないことは証明済みである [6]。

4 unit の重ね合わせ : EMBED

unit 同士を重ね合わせるとき、*core* 同士を重ね合わせる以外の方法で、*valid* である *unit* を得ることができる場合がある。この重ね合わせ方を *embed* と呼ぶ。*embed* は、前面にくる *unit* 全体を背面にくる *unit* に埋め込む重ね合わせ方である。例えば T1 に L1 に埋め込むと、図 10 のように新たに T1 を得ることができる。

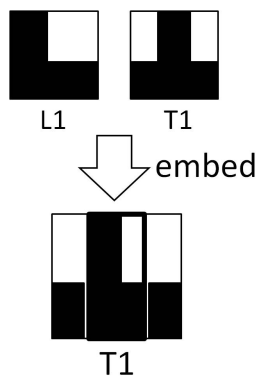


図 10 *embed* の例

basic units に対し *embed* をまとめたものが、表 1

である。行が背面にくる *unit*、列が前面にくる *unit* を示す。

ここで *unit* は、回転したものと対称であるものも含む。また、*puton* により同じタイプの *unit* が得られる場合は記載していない。

この表は、*embed* を行ない *valid* である *unit* が得られる場合、その得られる *unit* の種類を示している。'U' は、重ね合わせる *unit* をどの向きで重ね合わせても *valid* である *unit* が得られない場合である。

'U*' は、一見重ね合わせが可能であるように思われるが、*valid* である *unit* が得られない場合である。例えば T1 に L2 を重ね合わせる場合、図 11 のようになり (1) のラインと (2) のラインを揃えることは不可能であり、この *BLACK* は制約条件を満たすことができない。

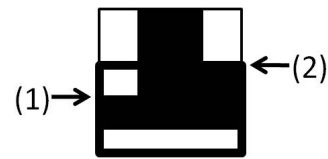


図 11 *embed* で得られる *valid* でない *unit*

また、*new units* に対しても、いくつかの場合 *embed* を行うことができる。例えば、図 12 の場合、WT2 に WT1 を埋め込むことによって新たに WT2 を得ることができる

5 おわりに

本発表では、サイズや辺の縦横比が可変な矩形同士の重ね合わせについて、定性空間推論の枠組みを使った定式化の拡張について述べた。この結果、矩形内の *BLACK*、*WHITE* の形に対する制約条件を緩和することで、より複雑なパターンの *unit* に対しても推論が可能になった。

具体的には以下の点で拡張が必要だったが、これらの拡張はこれまでの定式化を大きく変更することなく行えるものである。

- *valid* である *unit* の表現方法

fg \ bk	I1	I2	L1	L2	T1	T2	PLUS
I1	I1	I2	T1 L1 L2	T2 L2	T1 T2	T2	U*
I2	I2	I2	T1	T2	T1	T2	U
L1	L2	U	L1	L2 T2	T1	T2	U*
L2	L2	U	L2	L2	U*	U*	U
T1	T2	U	T1	T2	T1	T2	PLUS
T2	T2	U	U	U	T2	T2	U
PLUS	U	U	U	U	PLUS	U	PLUS

表 1 embed で得られる unit

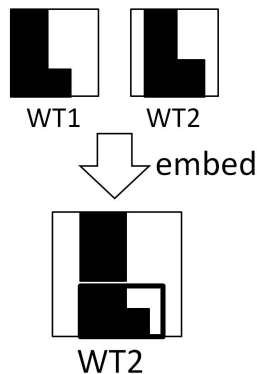


図 12 new units に対する embed の例

- puton 操作
- 重ね合わせの結果得られた unit が valid である判定条件

今後は、基本パターンで成立していたいくつかの性質が今回の拡張によっても保たれていることを証明することと、embed を用いた重ね合わせについて詳細に考察することを考えている。また、さらに複雑なパターンをもつ unit に対しての拡張も目指す。

参考文献

[1] Cohn, A. G. and Renz, J. : Qualitative Spatial Representation and Reasoning, *Foundations of Ar-*

- tificial Intelligence*, Vol. 3 (2008) , pp. 551–596.
- [2] Dowsland, K. A. and Dowsland, W. B. : Packing problems, *European Journal of Operational Research*, Vol. 56 (1992), pp. 2–14.
- [3] Egenhofer, M. J. and Franzosa, R. D. : Point-Set Topological Relations, *International Journal for Geographical Information Systems*, 1991, pp. 161–174.
- [4] Freksa, C. : Qualitative Spatial Reasoning, *Cognitive and Linguistic Aspects of Geographic Space*, 1991, pp. 361–372.
- [5] Freeman, H. : Computer name placement, in *Geographical Information Systems 1*, D. J. Maguire, M. F. Goodchild, and D. W. Rhind, Eds. John Wiley, 1991, pp. 449–460.
- [6] Konishi, T. and Takahashi, K. : Symbolic Representation and Reasoning for Rectangles with Superposition, *The Third International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2011)*, 2011, pp. 71–76.
- [7] Lapaugh, A. S. : Layout algorithm for VLSI design, *ACM Computing Surveys*, vol. 28, no. 1 (1996), pp. 59–61.
- [8] Li, J. , Plaisant, C. and Shneiderman, B. : Data object and label placement for information abundant visualizations, in *Proceedings of the Workshop of New Paradigms Information Visualization and Manipulation (NPV98)*, 1998, pp. 41–48.
- [9] Randell, D. A. , Cui, Z. and Cohn, A. G. : A Spatial Logic Based on Regions and Connection, *International Conference on the Principles of Knowledge Representation and Reasoning*, 1992, pp. 165–176.
- [10] Takahashi, K. and Sumitomo, T. : The Qualitative Treatment of Spatial Data, *International Journal on Artificial Intelligent Tools*, Vol. 16 (2007), pp. 662–682 .