

# 矩形同士の埋め込み型重ね合わせについての定性空間推論

小西貴子<sup>†</sup> 高橋和子<sup>†</sup>

<sup>†</sup> 関西学院大学理工学研究科

〒669-1337 兵庫県三田市学園2丁目1番地

E-mail: †{t.konishi,ktaka}@kwansei.ac.jp

あらまし 表示したい部分 (*WHITE*) と隠したい部分 (*BLACK*) を持ち、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述べる。複数枚の矩形の *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か否かの判定条件と、条件を満たす場合に重ね合わせの結果として得られる図形の推論方法を示す。重ね合わせ方は二通り考えられ、本発表では埋め込み型について考察する。また、この二通りの方法を使って構築した矩形の重ね合わせ推論システムも示す。

キーワード 定性空間推論, 定性的知識表現, 矩形詰め込み問題

## Qualitative Spatial Reasoning on Rectangle Superposition by Embedding

Takako KONISHI<sup>†</sup> and Kazuko TAKAHASHI<sup>†</sup>

<sup>†</sup> Graduate School of Science and Technology, Kwansai Gakuin University

2-1, Gakuen, Sanda, Hyogo, 669-1337 Japan

E-mail: †{t.konishi,ktaka}@kwansei.ac.jp

**Abstract** This paper discusses a superposition of qualitative rectangles whose size and edge ratios are not fixed, and some parts of which should be visible (*WHITE*) and other parts should be hidden (*BLACK*). We investigate the conditions under which a superposition succeeds so that all *WHITE*s are visible and *BLACK*s are hidden when multiple rectangles are given, as well as the reasoning on generating the resultant figure. There are two types of the operations of superposing two rectangles. We discuss the one by embedding one rectangle into part of the other rectangle. We also show the system of superposing multiple rectangles developed using these operations.

**Key words** qualitative spatial reasoning, qualitative knowledge representation, rectangle packing.

### 1. はじめに

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 [1], ラベルの配置, VLSI のレイアウトなど様々な分野に応用されている。本研究ではそのような問題の1つマルチウィンドウの配置問題に着目する。マルチウィンドウの配置問題は一般の最適配置問題では考慮されていない以下の点に注意する必要がある。(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。(i) に関してはオブジェクトを縮小したり縦横比を変更したりすることで解が見つかる場合があるため、サイズや位置を固定的な数値で表すよりもオブジェクトそのものやオブジェクト同士の相対的位置関係を定性的に表現する方が望ましい。そのため、一般の詰め込み問題で使われている方法をそのまま適用するのは難しい。加えて、一般の詰め込み問題では重ね合わせまで取り扱ったものはない。そこで、

本研究では定性空間推論 (Qualitative Spatial Reasoning) の枠組み [2] [3] を用いてオブジェクトを記号表現し、それに対して論理的推論を行う。これによって推論過程が明確になるとともに、計算量も抑えることができる。定性空間推論については人工知能分野で様々な枠組みが提案されている [3] [4] [5] [6] 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 (*WHITE*) と隠したい部分 (*BLACK*) から成り、サイズや辺の縦横比が可変である複数枚の矩形の重ね合わせを考える。複数枚の矩形のうち、1枚は矩形全体が *WHITE*、その他の矩形は *BLACK* と *WHITE* から成るとき、すべての *WHITE* を表示し、かつすべての *BLACK* を隠す重ね合わせ方が存在するか否かを推論する。

このとき、二通りの重ね合わせ方が考えられる。

一つは *puton* という重ね合わせ方で、これは *BLACK* の形を矩形に近づけるという発想で重ね合わせを行なうものである。(図 1(a)) の場合、*unit X* のもつ *BLACK* の下部に *unit Y* を重ね合わせることににより、*BLACK* を矩形にすることができる。このように、*puton* の結果は、全体の形が矩形にならない場合もある。

もう一方は *embed* という重ね合わせ方で、1 枚の矩形全体をもう 1 枚の矩形に埋め込み、かつ *puton* とは異なる重ね合わせ方という発想で重ね合わせを行なうものである。(図 1(b)) の場合、*unit X* のもつ *BLACK* の上部に *unit Y* 全体を埋め込むことで、新たな矩形を得ることができる。*embed* の結果は、必ず全体の形が矩形になる。

このように、それぞれの操作を用いることによって、結果として異なる図形が得られる。

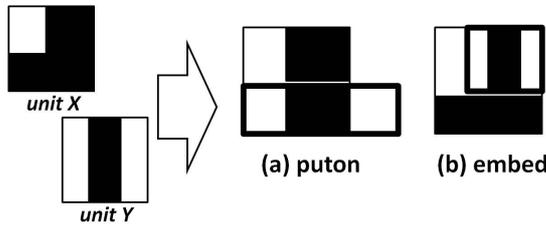


図 1 二通りの重ね合わせ方

著者らはこれまでに、対象となる矩形の定性表現を提案した。また、*puton* の重ね合わせ方については、制約を満たす重ね合わせが成功するために矩形を満たすべき条件を明らかにし、条件を満たす場合に重ね合わせの結果として得られる図形の推論方法も示した [7]。

本発表では、*embed* の重ね合わせ方について考察する。前研究と同様の方法で定式化することで重ね合わせが成功するための判定条件と得られる図形の推論方法を示す。さらに、この二つを用いて構築した複数の矩形重ね合わせ推論システムを示す。

本論文の構成は以下の通りである。2. 章では、対象とする単位図形である *unit* の定義と制約条件、表現方法について述べる。3. 章では埋め込み型の重ね合わせについて説明する。4. 章では、矩形の重ね合わせ推論システムについて述べる。5. 章では、本研究のまとめと今後の課題について述べる。

## 2. 対称の表現

### 2.1 対象とする単位図形

本研究で対象とする図形の単位を *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。

連結した *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 *region* と呼び、それぞれ

'*w*', '*b*', '*g*' と記す。

### 2.2 *valid unit* の制約条件

本研究では、*valid* である *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* として以下のように名付け、名付けたもの (*W*, *B*, *I1*, *I2*, *L1*, *L2*, *T1*, *T2*, *PLUS*) をタイプと呼ぶ。

(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 unit* と呼ぶ。*basic unit* 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* は連結している。

(iii) 全ての *white region* は矩形である。

さらに、*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。

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

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

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

*basic unit* と *new unit* を、*valid* である、という。すべての *valid unit* を、図 2 に示す。

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

### 2.3 *unit* の表現方法

*W* 以外の全ての *unit* は、以下のように *core region* を定義する。

*B*, *straight-plate-unit* および *WI* については、*BLACK* 全体を *core region* とする (図 3(a))。

*WI* を除く *cross-plates-unit* については、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 3(b))。

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

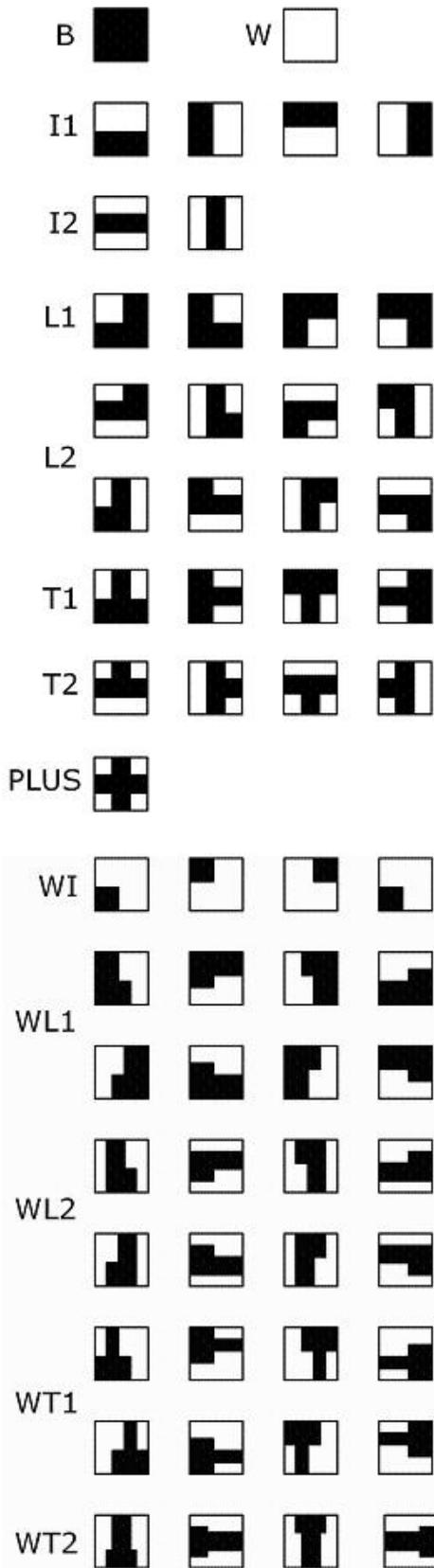


図2 validである unit

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. これを unit の表現とよぶ.

$core\ region$  から見て各方向にある region を,  $b, w, g$  を用

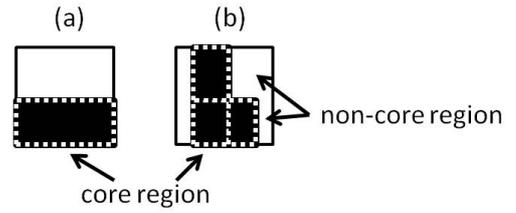


図3 unit の core region

いて表し,  $black\ region$  の先に  $white\ region$  が存在する場合のみ  $b(w)$  と表す.

例えば, 図4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  $white\ region$  があることを表している.

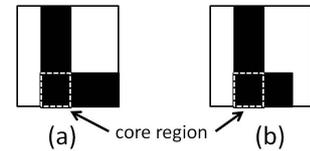


図4 unit の表現方法

2枚の unit の表現が一致する場合, その2枚の unit は同じタイプで, かつ同じ向きであるという.

### 3. unit の重ね合わせ: EMBED

$BLACK$  は全て隠し, かつ  $WHITE$  は全て表示するという要求を満たす2枚の unit の重ね合わせ方には,  $puton$  と  $embed$  の二通りの方法がある.

$puton$  は2枚の unit の  $core\ region$  同士を重ね合わせる操作であり,  $embed$  は1枚の unit 全体を, もう1枚の unit の一部に埋め込む操作である(図5). ここでは,  $embed$  の定式化と特徴について述べる.

$embed$  の結果は,  $puton$  と一致する場合とそうでない場合がある.  $embed$  により,  $puton$  とは異なるタイプの valid な unit が得られる場合,  $embed$  が成功する, という.

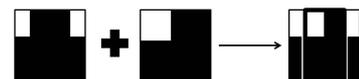


図5 embed の例

$embed$  の場合,  $puton$  と違い  $core\ region$  が関係しない重ね合わせ方をするため, すべての場合に共通な判定式は存在しない.

そのため, 今回は  $basic\ unit$  については unit の組み合わせごとに判定式を示し, valid である unit が得られる組み合わせについてはその結果を示す.  $new\ unit$  については,  $embed$  によって得られる結果のみを表にまとめる.

以下に、それぞれの場合について説明する。

### 3.1 basic unit 同士の embed

unit  $r = \langle r_1, r_2, r_3, r_4 \rangle$ , unit  $r' = \langle r'_1, r'_2, r'_3, r'_4 \rangle$  を basic unit とする。  $r$  に  $r'$  全体を埋め込む場合を考える。

$r'$  全体を  $r$  の一部に埋め込み unit の表現  $\langle r_1, r_2, r_3, r_4 \rangle$  を返す関数  $embed$  を以下のように定義する。

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

$embed : R \times R \rightarrow R \cup \{fail\}$

まず、 $embed$  で用いる関数  $e$  を導入する。

関数  $e$  を以下のように定義する。

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

$V$  の中で、basic unit に対応する部分集合を

$V_b \subseteq V$

とする。

$V_b = \{b, w, g\}$

$e : V_b \times V_b \rightarrow V_b$

$e(g, v) = v$  where  $v \in V_b$

$e(v, g) = v$  where  $v \in V_b$

$e(b, b) = b$

$e(w, w) = w$

$e(b, w) = w$

$e(w, b) = w$

basic unit 同士の  $embed$  の場合、以下のように cross-plates-unit と straight-plate-unit で場合分けをして考える。

(1)  $r, r'$  のタイプが同じ場合

(2)  $r, r'$  共に cross-plates-unit の場合

(3)  $r, r'$  共に straight-plate-unit の場合

(4)  $r, r'$  のうち、片方が cross-plates-unit, もう片方が straight-plate-unit で、plate 全体に埋め込む場合

(5)  $r, r'$  のうち、片方が cross-plates-unit, もう片方が straight-plate-unit で、plate の一部に埋め込む場合

#### 3.1.1 $r, r'$ のタイプが同じ場合

2 枚の unit のタイプが同じ場合、unit の向きも同じならば、 $embed$  を行なったとき 2 枚の unit と同じタイプの unit を得ることができる。また、向きが異なる場合は  $embed$  を行なうことができない。

以下にその式を示す。

$\bigwedge_{i=1, \dots, 4} (r_i = r'_i)$  ならば

$embed(r, r') = \langle r_1, r_2, r_3, r_4 \rangle$

それ以外の場合

$embed(r, r') = fail$

#### 3.1.2 cross-plates-units 同士の embed

$r$  のタイプが  $L1, L2, T2$ ,  $r'$  のタイプが  $PLUS$  の場合、 $embed$  が成功しないのは明らかである。それ以外の場合は、基本的な考え方として 2 枚の unit それぞれにおいて  $WHITE$  が core region から見て少なくとも 1 つ同じ方向にあれば  $embed$

が成功する。以下に、例外については述べる。

$r$  のタイプが  $L2$ ,  $r'$  のタイプが  $T1$  の場合、2 枚の unit の向きによって、unit それぞれの core region から見て  $WHITE$  の方向が一致していても、 $embed$  が成功する場合としない場合がある (図 6)

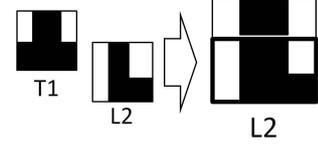


図 6  $T1$  と  $L2$  の組み合わせで成功する例

$r$  のタイプが  $L2$ ,  $r'$  のタイプが  $L1$  の場合、core region から見て  $WHITE$  の方向が 1 つも一致していなくても、重ね合わせを行なえる場合が二通りある (図 7, 8)。

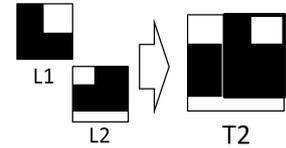


図 7  $L1$  と  $L2$  の組み合わせの例外 1

図 7 の場合、 $L1$  において core region から見た  $WHITE$  の方向と、 $L2$  において core region から見た  $WHITE$  の方向が左右または上下対称方向にあれば、 $embed$  が成功する。

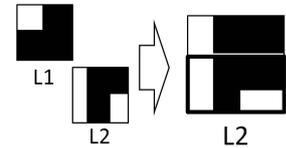


図 8  $L1$  と  $L2$  の組み合わせの例外 2

#### 3.1.3 $r, r'$ 共に straight-plate-unit の場合

$r, r'$  共に straight-plate-unit の場合は、 $I1$  と  $I2$  の組み合わせしか  $embed$  は成功しない。

$I1$  と  $I2$  の組み合わせの場合、2 枚の unit それぞれにおいて core region から見た  $WHITE$  の方向が一致していれば  $embed$  は成功する (図 9)。

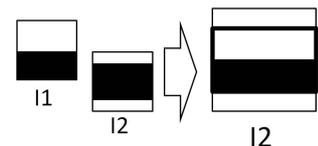


図 9  $I1$  と  $I2$  が関係する場合

#### 3.1.4 $r, r'$ のうち、片方が cross-plates-unit, もう片方が straight-plate-unit で、plate 全体に埋め込む場合

$r, r'$  が共に cross-plates-unit, または straight-plate-unit の場合は、 $r'$  は  $r$  の  $BLACK$  の plate 全体に埋め込む場合にし

か *embed* が成功しないが, *cross-plates-unit*, *straight-plate-unit* がまじった場合は, *plate* の一部に埋め込むことで成功する場合もある.

$r, r'$  のうち, 片方が *cross-plates-unit*, もう片方が *straight-plate-unit* で, *plate* 全体に埋め込む場合, *straight-plate-unit* である *unit* が  $I1$ , *cross-plates-unit* である *unit* が  $L1, T1, L2, T2$  の場合に *embed* が成功する. その場合,  $I1$  の *core region* から見て *WHITE* がある方向に,  $L1$  または  $T1$  の *core region* から見て *GRAY* が, もしくは,  $L2$  または  $T2$  の *core region* から見て *WHITE* が存在すれば *embed* は成功する (図 10).

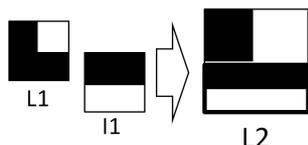


図 10  $I1$  が関係する場合

### 3.1.5 $r, r'$ のうち, 片方が *cross-plates-unit*, もう片方が *straight-plate-unit* で, *plate* の一部に埋め込む場合

ここまでの検討においては, すべて前面にくる *unit* の *core region* が, そのまま結果として得られる *unit* の *core region* となっていた. しかし, *straight-plate-unit* を *cross-plates-unit* の *plate* の一部に埋め込む場合, *cross-plates-unit* の *core region* の上に埋め込まれることはない. (図 11). そのため, *core region* を中心とした表現方法を用いて判定することは難しい. この場合は, 背面にくる *unit* でパターン分けを行ない, それぞれの組み合わせに対して *embed* を適用した.

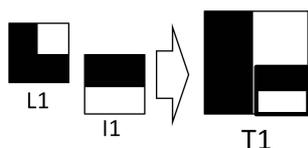


図 11  $I1$  を他の *unit* の *plate* の一部に埋め込む場合の例

*basic unit* 同士の *embed* で得られる結果をまとめたものを表 1 に示す.

表 1 *basic unit* 同士の *embed* で得られる *unit*

前面 背面	$I1$	$I2$	$L1$	$L2$	$T1$	$T2$	PLUS
$I1$	$I1$	$I2$	$L2$	$L2$	$T2$	$T2$	U
$I2$	$I2$	$I2$	U	U	U	U	U
$L1$	$L1$ $L2$ $T1$	$T1$	$L1$	$L2$	$T1$	U	U
$L2$	$L2$ $T2$	$T2$	$L2$ $T2$	$L2$	$L2$ $T2$	U	U
$T1$	$T1$ $T2$	$T1$	$T1$	U	$T1$	$T2$	PLUS
$T2$	$T2$	$T2$	$T2$	U	$T2$	$T2$	U
PLUS	U	U	U	$T2$	PLUS	$T2$	PLUS

### 3.2 *new unit* が関係する *embed*

*new unit* が関係する *embed* を考える.

*embed* を行なった結果を, 表 2 に示す. なお, ここで示している結果は, *new unit* が関連するもののみで, *basic unit* 同士の重ね合わせから *basic unit* が得られるものは示していない. また, 結果が複数ある組み合わせは, 重ね合わせる 2 枚の *unit* の向きや埋め込む位置を変えることによって, 異なる結果が得られることを示している (図 12).

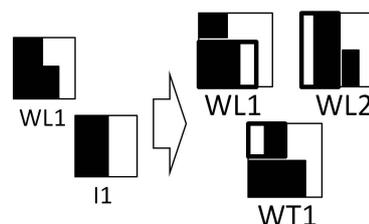


図 12 結果が複数ある組み合わせの例

この表から, *basic unit* と違い, *new unit* では同じタイプのものを重ね合わせても結果が得られない場合 (WL2) や, 2 つの結果が得られる場合 (WT1) があることがわかる (図 13).

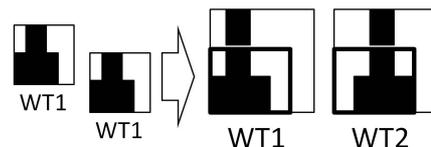


図 13 同じタイプのを重ね合わせて結果が 2 つ得られる場合 (WT1)

*new unit* と *basic unit* の重ね合わせを考えると, *new unit* を背面にする方が, *basic unit* を背面にするより *embed* が成功する場合が多い.

PLUS の *unit* は, どの *unit* とも成功しない.

表 2 *new unit* と *basic unit* の *embed*

前面 背面	$I1$	$I2$	$L1$	$L2$	$T1$	$T2$	PLUS	WI	WL1	WL2	WT1	WT2
$I1$	WI WL1	WT1	WL1	WL2	WT1	U	U	WI	WL1 WL2	WL2	WT1 WT2	WT2
$I2$	WL2	WT2	WL2	WL2	WT2	U	U	U	WL2	WL2	WT2	WT2
$L1$	WL1	U	U	U	U	U	U	U	WL1	U	WT1	U
$L2$	WL2	WT1	WT1	U	WT1	U	U	WI	WL2	U	WT1	U
$T1$	WT1	WT2	U	U	U	U	U	U	U	WL2	WT1	WT2
$T2$	U	U	U	U	U	U	U	U	U	U	U	U
PLUS	U	U	U	U	U	U	U	U	U	U	U	U
WI	WI	U	U	U	U	U	U	WI	U	U	U	U
WL1	WL1 WL2 WT1	L2 WT1	WL1	WL2	WT1	U	U	WI	WL1	U	WT2	U
WL2	WL2 WT2	WL2 WT2	WL2	WL2	WT2	U	U	U	WL1	U	WT2	U
WT1	WT1 WT2	WL2 WT1	WT1	L2	WT1	$T2$	U	U	U	WL2	WT1 WT2	WT2
WT2	WT2	WL2 WT2	WT2	WL2	WT2	U	U	U	U	WL2	WT2	WT2

### 3.3 *embed* の性質

以下に *embed* の性質をまとめる [7].

- *puton* で重ね合わせが成功し, かつ *valid* である *unit* が

得られる 2 枚の *unit* の組み合わせは、必ず *embed* でも同じ結果が得られる。

- *plate* の一部に埋め込む場合は、*cross-plates-unit* に *straight-plate-unit* を埋め込む場合しか成功しない。
- 2 枚の *unit* の重ね合わせ方として、今回 *puton* 以外に *embed* の定式化を行なったが、この 2 つと定性的に異なる重ね合わせが存在するか否かはまだわかっていない。
- *embed* に関しては、タイプの中に閉じた部分集合が存在しない。

#### 4. 重ね合わせ推論システム

本研究では、*puton* と *embed* 双方の操作を用いて、 $n$  枚の *unit* の重ね合わせ方と順番を推論するシステムを構築した。 $n$  枚の *unit* の重ね合わせに関しては、 $n$  枚の *unit* で重ね合わせが成功するとき、必ずその中で重ね合わせが成功する 2 枚の組み合わせが存在するため、2 枚の *unit* の重ね合わせを用いて再帰的に推論を行なうことができるということを証明している [7]。

このシステムでは、まず最初の画面 (図 14) で複数枚の *unit* を選択し、メニューから判定開始を押すと、推論結果が表示される。解がある場合は、*unit* を重ね合わせる順番、*puton* と *embed* のどちらの重ね合わせ方を用いたか、*embed* を用いた場合はどの位置に埋め込まれたかが表示される (図 15)。解がない場合は、解なしと表示される。

このシステムを用いることにより、一見重ね合わせが成功するか否かわからない *unit* の組み合わせに対して、簡単に推論結果を見ることができる。

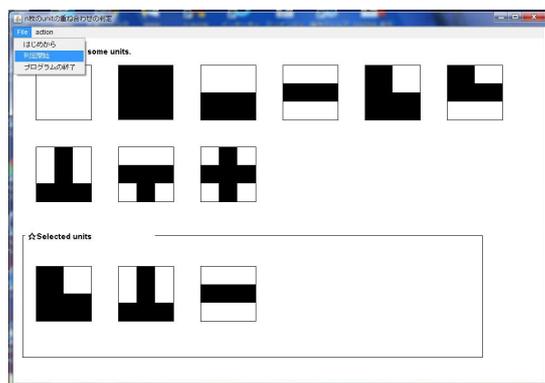


図 14 *unit* の選択画面

#### 5. おわりに

本研究では、隠したい部分と表示したい部分をもつ矩形に対し、既に定式化を行なっている重ね合わせ方とは別の重ね合わせ方 *embed* について、新たに定式化を提案した。これにより、 $n$  枚の *unit* の重ね合わせについて、*puton*、*embed* 双方の操作を用いた重ね合わせの推論システムを構築することができた。

今後の課題として、*puton*、*embed* 以外に、この 2 つと定性的に異なる重ね合わせ方が存在するか否かについての研究、またより複雑な *unit* への拡張等が挙げられる。

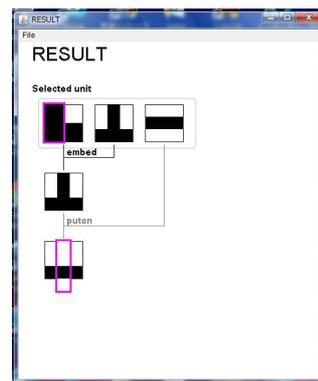


図 15 解がある場合の例

#### 文 献

- [1] Kathryn A. Dowsland, William B. Dowsland: Packing problems, *European Journal of Operational Research*, Vol. 56, pp. 2–14 (1992)
- [2] Anthony G. Cohn, Jochen Renz: Qualitative Spatial Representation and Reasoning, *Foundations of Artificial Intelligence*, Vol. 3, pp. 551–596 (2008)
- [3] Christian Freksa: Qualitative Spatial Reasoning, *Cognitive and Linguistic Aspects of Geographic Space*, pp. 361–372 (1991)
- [4] Max J. Egenhofer, Robert D. Franzosa: Point-Set Topological Relations, *International Journal for Geographical Information Systems*, pp. 161–174 (1991)
- [5] David A. Randell, Zhan Cui, Anthony G. Cohn: A Spatial Logic Based on Regions and Connection, *International Conference on the Principles of Knowledge Representation and Reasoning*, pp. 165–176 (1992)
- [6] Kazuko.Takahashi, Takao.Sumitomo: The Qualitative Treatment of Spatial Data, *International Journal on Artificial Intelligent Tools*, Vol. 16, pp. 662–682 (2007)
- [7] 小西 貴子: 矩形同士の重ね合わせに対する定性空間推論, 修士論文 (2012)
- [8] T. Konishi, and K. Takahashi, “Symbolic Representation and Reasoning for Rectangles with Superposition,” *The Third International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2011)*, pp. 71–76 (2011)