

定理証明支援系による曲線の定性表現手法の形式化と証明

高橋 和子 山下 拓真 今井 宜洋

本発表では、曲線の定性表現およびその上での推論について、定理証明支援系 Rocq を使って形式化し証明を行う。具体的な対象は、方向と凸性を属性としてもつ単位セグメントを連結して構成される *scurve* と呼ばれる曲線が、2 次元平面上に交差することなく埋め込むことができるかどうかを判定する手法の正当性である。これにはすでに手証明が与えられているが、厳密な形式化は行われていない。本発表は、Rocq で曲線の定性表現を 2 次元平面上に埋め込むための形式化およびそのときに成り立つ性質の証明を行うための見通しを立てることを目的として 2 つのことを行う。まず、記号列として表現された *scurve* の 2 次元平面への埋め込みを形式化し、特定の *scurve* はらせん状にしか描くことができないという性質を証明するために必要な補題を抽出する。次に、判定のための *scurve* の回転の向きを使った簡約操作を形式化しこれに関する性質を証明する。

1 はじめに

画像データを対象とする研究分野においては最近では機械学習によるアプローチが主となっているが、一般に大量のデータと学習時間がかかることや、結果の信頼性が保証されないなどの欠点もある。相補的なアプローチとして画像データの特定の性質に着目した記号表現や論理推論を中心とした定性空間推論手法がある [3][10][2][13]。

形状に着目した定性空間推論手法の多くは、画像上のオブジェクトの輪郭を複数個の切片に分割し、各切片の形状の特徴を記号で表現し、それらの列としてオブジェクトの概形を表している [1][5][9][18]。逆に、定義に従った記号列が与えられたとき、2 次元平面上に対応する図が描ける保証はなく、また、描けたとしても自己交差したりらせん状になると、自然界のオブ

ジェクトの輪郭としては妥当でない。しかし、このような側面を対象とした研究はほとんど見当たらない。

高橋らは定性空間推論の一手法として、方向と凸性を属性としてもつ曲線分を単位セグメントとし複数の単位セグメントを連結していくことで尖点を持たない円滑な曲線を記号列 *scurve* として表現し、その上で形状などを推論する手法を提案した [17]。また、このように構成された記号列から平面上への写像を考え、らせん状にならないように描ける写像が存在するかどうかを判定する効率的手法として簡約を提案し [20][14]、その性質を示した [15]。これらの正当性にはすでに手証明が与えられているが、厳密な証明になっていない。本発表では、これを定理証明支援系 Rocq (Coq) を使って形式化を試みる。定理証明支援系において、曲線の定性表現から 2 次元平面への埋め込みを扱った事例は著者らの知る限り存在しない。本発表ではその最初のアプローチとして、形式化と証明の見通しを立てることを目的とする。

Coq は代表的な定理証明支援系の 1 つであり、2023 年に Rocq と改称された [4]。Rocq は型付き λ 計算に基づいた論理体系であり OCaml を拡張した言語で定義や定理を記述し、タクティックと呼ばれる書き換え規則を適用することで証明が進行する対話型の

* Formalization and Proof for Qualitative Representation of Curves Using the Interactive Theorem Prover.

This is an unrefereed paper. Copyrights belong to the Author(s).

Kazuko Takahashi, 関西学院大学工学部, School of Engineering, Kwansei Gakuin University.

Takuma Yamashita, Yoshihiro Imai, proof_ninja, proof_ninja, Inc..

証明支援系である．ライブラリも充実しており 4 色問題 [6] や保証済みコンパイラ CompCert [8] をはじめとして多くの応用事例が報告されている．

本研究では具体的に以下の 2 つの形式化と証明を行う．まず，記号列として表現された *scurve* の 2 次元平面への埋め込みを形式化し，特定の *scurve* はらせん状にしか描くことができないという証明に必要な補題を抽出する．次に，判定のための *scurve* に対する簡約操作を形式化しこれに関する性質を証明する．

読者は Rocq の基礎知識があるものとして話をすすめるが，詳細な証明技術を論じるのではなく基本的な考え方と証明の概略のみを示す．

本発表は以下のように構成される．第 2 節で曲線の定性表現と 2 次元平面への埋め込み，および簡約について紹介する．第 3 節でこの定性表現の体系および簡約の Rocq による形式化と性質の証明について述べる．第 4 節で関連研究と比較し，最後に第 5 節で結論と今後の課題について述べる．

2 曲線の定性表現の体系

2.1 *scurve*

2 次元平面上の有向曲線のセグメントで，方向と曲率が一定であるようなものの集合を *CURVES* とする．*CURVES* の要素はその方向と凸性のみに注目してその形状を定性的に表す．正確なサイズや曲率は無視する．

図 1 において左の 3 つは定性的に同一であり，右の 2 つとはそれぞれ異なると見なす．

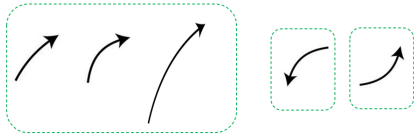


図 1 セグメントの定性的な分類

$S_v = \{n, s\}$, $S_h = \{e, w\}$, $Conv = \{cx, cc\}$ および $Dir = S_v \cup S_h$ とする．ここで，記号 n, s, e, w はそれぞれ上，下，右，左を表し， cx, cc はそれぞれ上に凸，下に凸を表す．真横を向くものは n, s のい

れかに分類し，真上や真下を向くものは e, w のいずれかに分類する．また，直線は考えない．各集合の要素に対してその集合内の相補的な要素を割り当てる関数を i とする．すなわち $i(s) = n, i(n) = s, i(e) = w, i(w) = e, i(cx) = cc, i(cc) = cx$ である．

$X \in CURVES$ を**セグメント**と呼び，その定性的な形状を $V \in S_v, H \in S_h, C \in Conv$ である 3 項組 (V, H, C) で表す．これを X の**定性表現**と呼ぶ．

$X, Y \in CURVES$ に対して $X = (V, H, C)$, $Y = (V', H', C')$ をそれぞれ X, Y の定性表現とする．*CURVES* の上の関係 $\sim$ を $V = V', H = H', C = C'$ が成り立つときのみ $X \sim Y$ であると定義すると， $\sim$ は *CURVES* 上の同値関係になる．その結果，*CURVES* は 8 つの同値類に分割される．これを $S = CURVES / \sim$ とおく．

$X \in S$ を**単位セグメント**と呼び，その始点と終点をそれぞれ $init(X)$, $term(X)$ と表記する．

尖点を持たない連続で有限な有向曲線をここでは *scurve* と呼ぶ．単位セグメントを連結することで *scurve* を生成する．

$X, Y \in S$ に対して $init(Y)$ と $term(X)$ を同一視することで *scurve* が得られるとき， X と Y は**直接連結可能である**といい， $dc(X, Y)$ と記述する．また，連結した結果を $X \cdot Y$ と表す．

X, Y の正確な曲率は無視するため，同一のクラスに属する単位セグメントのペアは直接連結可能であり，この 2 つを連結して得られる新たな曲線は尖点を持たない単一セグメントと見なす (図 2(a))．そうでない場合，セグメントのペア X, Y は，直接連結において変曲点または極点を形成する． $ifl(X, Y)$ は $X \cdot Y$ が変曲点を持つことを表し (図 2(b))， $xtr(X, Y, D)$ は，極点を持つことを表す (図 2(c))．

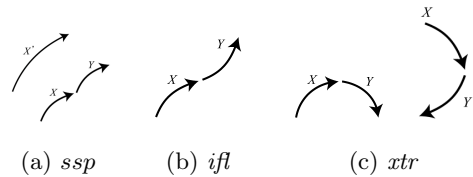


図 2 直接連結可能な単位セグメントのペア

直接連結は以下の規則に従う。

- $dc(X, Y)$ iff
 $ssp(X, Y)$ or $ifl(X, Y)$ or $xtr(X, Y, D)$ ($D \in Dir$).
- $ifl(X, Y)$ iff
 $X = (V, H, C)$ and $Y = (V, H, i(C))$.
- $xtr(X, Y, V)$ iff
 $X = (V, H, C)$ and $Y = (i(V), H, C)$.
- $xtr(X, Y, H)$ iff
 $X = (V, H, C)$ and $Y = (V, i(H), i(C))$.

$X, Y \in \mathcal{S}$ が直接連結可能でない場合は、 X と Y の間に $\mathcal{S}$ の要素を補間することで X, Y 間の *scurve* を得る。

$X_1, \dots, X_n \in \mathcal{S}$ ($n \geq 2$) に対して X_j, X_{j+1} ($\forall i; 1 \leq i \leq n-1$) が直接連結可能ならば、 X_1, X_n は **連結可能** といい、連結の結果得られる曲線を $X_1 \dots X_n$ と表す。

$X_1 \dots X_n$ ($n \geq 2$) ただし $X_{i-1} \neq X_i$ ($\forall i; 2 \leq i \leq n$) に対して $X_1 \dots X_n$ を X_1 から X_n への *scurve* と呼び、 $X_1 - X_n$ と表す。

2.2 *scurve* の 2 次元平面への埋め込み

$\mathcal{S}$ から *CURVES* への写像を考える。

1. 2 次元平面上の曲線 $\mathbf{X} \in \text{CURVES}$ の定性表現を $X \in \mathcal{S}$ とするとき、 $\mathbf{X}$ を X の埋め込みと呼ぶ。 $init(\mathbf{X})$, $term(\mathbf{X})$ はそれぞれ 2 次元平面上における $\mathbf{X}$ の始点と終点の位置を表す。
2. $X_1 \dots X_n$ を *scurve* $X_1 - X_n$, $\mathbf{X}_i$ ($1 \leq i \leq n$) を X_i の埋め込みとする。任意の i ($1 \leq i \leq n-1$) に対して、 $term(\mathbf{X}_i)$ と $init(\mathbf{X}_{i+1})$ が同一位置に配置されているとき、 $X_1 \dots X_n$ を *scurve* $X_1 - X_n$ の埋め込みという。

単一の *scurve* に対してその埋め込みは無限個存在するが、埋め込み方によってらせん状になる場合とそうでない場合がある。

scurve $X - Y$ の 2 次元平面への埋め込み $\mathbf{X} - \mathbf{Y}$ に対して、 $\mathbf{X}, \mathbf{Y}$ の曲率を保持したまま $\mathbf{X}$ の始点方向および $\mathbf{Y}$ の終点方向に延長して得られる (無限) 曲線を考える。この曲線が自己交差しない場合、その埋め込みは**開 (open)** といい、自己交差する場合は**閉 (closed)**

という。開埋め込みが存在するときその *scurve* を**許容可能 (admissible)** という。また、*scurve* が空列の場合も許容可能とする。

図 3 はそれぞれ $X = (n, e, cx)$, $Z = (s, e, cx)$, $Y = (s, w, cc)$ の場合の *scurve* $X \cdot Z \cdot Y$ の開埋め込みと閉埋め込みを示している。開埋め込みが存在するので $X \cdot Z \cdot Y$ は許容可能である。

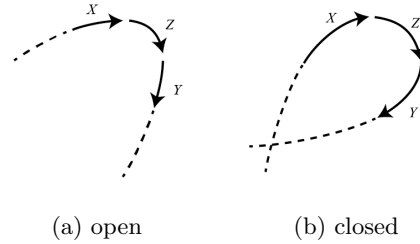


図 3 $X \cdot Z \cdot Y$ の 2 種類の埋め込み

一方、 $X = (n, e, cx)$, $Z_1 = (s, e, cx)$, $Z_2 = (s, w, cc)$, $Y = (n, w, cc)$ の場合、*scurve* $X \cdot Z_1 \cdot Z_2 \cdot Y$ には閉埋め込みしか存在しない (図 4)。したがって $X \cdot Z_1 \cdot Z_2 \cdot Y$ は許容可能ではない。

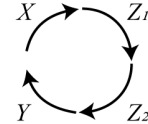


図 4 閉埋め込みしか存在しない *scurve*

2.3 簡約と許容可能性判定

scurve の許容可能性は、それを辿ったときの回転の向きを考えることで判定可能なので、以下では *scurve* の向きを対象に議論する。

$X \in \mathcal{S}$ の向きを時計回り (+)/半時計回り (-) のいずれかに決める関数 *orn* を定義し、さらに、*scurve* の向きをそれを構成するセグメントの向きの列と定義する。

- (単位セグメントの向き) $X \in \mathcal{S}$ に対して
 $orn(X) = '+'$ iff $X = (n, e, cx)$, (s, e, cx) ,

(s, w, cc) or (n, w, cc) ; $orn(X) = '-'$ iff $X = (s, w, cx), (s, e, cc), (n, e, cc)$ or (n, w, cx) .

- (*scurve* の向き) $X_1, \dots, X_n \in \mathcal{S}$ に対して $orn(X_1 \dots X_n) = orn(X_1) \dots orn(X_n)$.

たとえば図 4 において $orn(X \cdot Z_1 \cdot Z_2 \cdot Y) = ++++$ である.

scurve p の向き $orn(p)$ に出現する $+$ と $-$ の個数の差を $orn(p)$ の回転差と呼ぶ.

また, 集合 $\{+, -\}$ の上で相補となる要素を割り当てる関数 inv を $inv(+)=-, inv(-)=+$ と定義し, これらを逆向き方向と呼ぶ.

scurve の向きを簡約することによって, 与えられた任意の長さの向きの列の許容可能性を判定する. 簡約規則は 2 つあり, 1 つは向きが $+-+$ (または $-+-$) の部分列を $+$ (または $-$) で置き換えるもの, もう 1 つは $++--$ (または $--++$) の部分列を $+-$ (または $-+$) で置き換えるものである.

[簡約規則]

σ_1, σ_2 を向きの列, $s_1, s_2, s_3, s_4 \in \{+, -\}$ とする.

(r1) $s_1 = s_3 = inv(s_2)$ のとき, $\sigma_1 s_1 s_2 s_3 \sigma_2$ を $\sigma_1 s_1 \sigma_2$ と置き換える.

(r2) $s_1 = s_2 = inv(s_3) = inv(s_4)$ のとき, $\sigma_1 s_1 s_2 s_3 s_4 \sigma_2$ を $\sigma_1 s_1 s_4 \sigma_2$ と置き換える.

scurve の向き σ の部分列に可能な限り (r1), (r2) を適用して得られる向きの列を σ の簡約形と呼ぶ.

任意の向きの列に対して以下の性質が成り立つ.

命題 1 (簡約の性質) 1. (停止性)

簡約過程は停止する.

2. (回転差保持)

簡約において回転差は保持される.

3. (簡約形)

(a) 簡約形は, 両端にそれぞれ高々 1 個の逆向き方向 $inv(x)$ にはさまれた同一の向き x の 0 個以上の列である.

(b) 任意の向きの列に対して, 両端にある向きはいずれも簡約過程で変化しない.

4. (合流性)

簡約規則の適用順にかかわらず簡約形は一意的に定まる.

5. (許容可能性の保持)

簡約過程において許容可能性は保持される.

一般に, 2 次元平面上の連続曲線の回転角の和が 2π 以上になればその曲線は 2 次元平面上で交わることが知られている. 回転差が 4 以上の向きの列はこれに相当し, 対応する *scurve* は許容可能でない. また, 向きの列の長さが 8 以上ならば回転差は 4 以上になる. 回転差が 3 以下の場合の許容可能性に関しては以下の命題が成り立つ.

命題 2 (許容可能性の判定) 任意の向きの列に対して, 回転差が 3 以下のとき得られる簡約形のパターンは図 5 に示す 11 個に限る. ただし, $+$ と $-$, および向きの列を前から見たときと後ろから見たときに対称なものは同一のものと考えている. この中で許容可能でないものは図で NG とかかれた 2 個のみである.

3 Rocq による形式化

前節で説明した曲線の定性表現の体系を Rocq で形式化する.

3.1 scurve

まず, 単位セグメントを上下方向, 左右方向, 凸性の 3 つの属性をもつ項として定義する.

```
Inductive V : Set :=
| n : V
| s : V
.
Inductive H : Set :=
| e : H
| w : H
.
Inductive C : Set :=
| cx : C
| cc : C
.
Definition PrimitiveSegment := (V*H*C)%type
```

直接連結を単位セグメント上の関数として定義する.

```
Inductive dc : PrimitiveSegment ->
PrimitiveSegment -> Prop :=
| DIfl v h c : dc (v, h, c) (v, h, i_c c)
| DXtrvN h : dc (n, h, cx) (s, h, cx)
| DXtrvS h : dc (s, h, cc) (n, h, cc)
| DXtrhN h : dc (n, h, cc) (n, i_h h, cx)
| DXtrhS h : dc (s, h, cx) (s, i_h h, cc)
.
```

続いて連結および *scurve* を単位セグメントの列として定義する (図 6). 単位セグメントを, *scurve* の

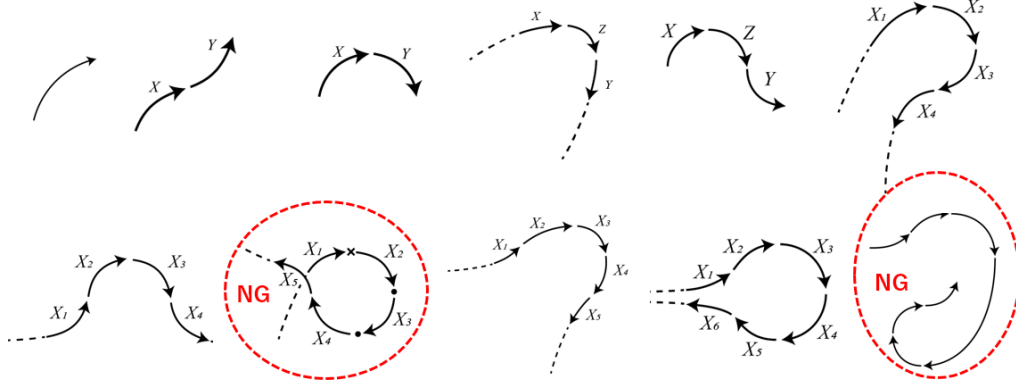


図 5 回転差 3 以下の簡約形の全パターン

先頭セグメントと連結可能な場合にそこに連結するように, *scurve* を再帰的に定義している.

3.2 2次元平面上への埋め込み

埋め込んだ曲線の連続性を扱うために, $\mathbf{R}$ 上の閉区間 $[0, 1]$ を動く媒介変数を使って 2 次元平面上の x, y 座標をそれぞれ表す. そのために, セグメントを $\mathbf{R}$ から $\mathbf{R}^2$ への写像とし, 閉区間 $[0, 1]$ から 2 次元平面上への関数として対応が取れるようにした.

Definition $\text{Segment} := \mathbf{R} \rightarrow \mathbf{R} * \mathbf{R}.$

あるセグメントに対して, $[0, 1]$ 区間上の任意の数に対応する平面上の座標がそのセグメント上にあることを示す述語 *onSegment* を定義する.

Definition $\text{onSegment} (\text{seg} : \text{Segment})$
 $(rr : \mathbf{R} * \mathbf{R}) :=$
 $\text{exists } (t : \mathbf{R}), 0 \leq t \leq 1 \wedge \text{seg } t = rr.$

これを使ってセグメントの始点と終点の位置関係や両者の間に必ず点が存在するなど基本的な性質を記述する. 直観的に自明と思われるものはひとまず公理として仮定することで証明をすすめる. 図 7 に連続性に関して仮定した公理の一部を示す.

scurve の埋め込みを定義する (図 8). *scurve* に単位セグメントを連結したものと, セグメントのリストに始点と終点を一致させるようにセグメントを連結したものが対応するように埋め込む関係として定義

している.

開埋め込み/閉埋め込みの定義をする. セグメントのリストを始点および終点方向に延長した曲線に対応させる. そして, $[0, 1]$ 上の異なる 2 点で, 対応するセグメント上の点が一致するものが存在するときに閉埋め込みと定義する. 開埋め込みはその否定で表す.

Parameter $\text{extend} :$
 $\text{list Segment} \rightarrow (\mathbf{R} \rightarrow \mathbf{R} * \mathbf{R}).$
Definition $\text{close_extended} (c : \text{Segment}) :=$
 $\text{exists } (t1\ t2 : \mathbf{R}), t1 < t2 \wedge c\ t1 = c\ t2.$
Definition $\text{close} (ls : \text{list Segment}) : \text{Prop} :=$
 $\text{close_extended } (\text{extend } ls).$

許容可能性を以下のように定義する.

Definition $\text{admissible} (lp : \text{scurve}) : \text{Prop} :=$
 $\text{exists } ls : \text{list Segment},$
 $(\text{embed_scurve } lp\ ls \wedge \sim \text{close } ls).$

図 4 で示した $X \cdot Z_1 \cdot Z_2 \cdot Y$ の列, 図 5 で NG と記したセグメント数が 5 個のものおよび 7 個のものあわせて 3 つの例が許容可能でないことを証明する.

ここでは連続性に関する公理を利用して証明をすすめたが, 主として *onSegment* を使って記述した多くの補題を導入する必要があった. 図 9 に導入した補題の例を示す. これらの意味は以下の通りである. 補題 (1) は, 埋め込んだ曲線の最初のセグメントや最後のセグメントについて, ある x/y 座標より大きい/小さい任意の座標に対して, その x/y 座標を持つ点がセグメント上にあることを表す. 補題 (2) は, *scurve*

```

Inductive dc_pseg_hd: PrimitiveSegment -> list PrimitiveSegment -> Prop :=
| DcNil: forall (x:PrimitiveSegment), dc_pseg_hd x nil
| DcCons: forall (x y:PrimitiveSegment) (l:list PrimitiveSegment), dc x y ->
dc_pseg_hd x (y::l)
.

Definition connect (s: PrimitiveSegment) (lp: scurve) (A: dc_pseg_hd s (proj1_sig lp)) :
scurve := exist _ (s :: (proj1_sig lp)) (IsScurveCons s (proj1_sig lp) (proj2_sig lp) A).

Inductive is_scurve: list PrimitiveSegment -> Prop :=
| IsScurveNil: is_scurve nil
| IsScurveCons: forall (x:PrimitiveSegment) (xs:list PrimitiveSegment) , is_scurve xs ->
dc_pseg_hd x xs -> is_scurve (x :: xs).

Definition scurve : Set := { l: list PrimitiveSegment | is_scurve l }.

```

図 6 *scurve* の定義

(1) セグメントの始点と終点の位置関係
Axiom n_end_relation: forall (s: Segment) (h: H) (c: C),
embed (n, h, c) s -> snd (init s) < snd (term s).

(2) セグメント上の点とセグメントの始点・終点の位置関係
Axiom s_onseg_relation: forall (s1: Segment) (h:H) (c:C) (x y: R),
embed (s, h, c) s1 -> onSegment s1 (x, y) -> snd (term s1) <= y /\ y <= snd (init s1).

(3) セグメントが平面上の 2 点を通る時、その間にある x/y 座標を取ると、その x/y 座標の点がセグメント上に存在する
Axiom exist_between_x_pos: forall (seg: Segment) (x1 x2 y1 y2 x: R),
onSegment seg (x1, y1) -> onSegment seg (x2, y2) -> y1 <= y2 -> x1 <= x -> x <= x2 ->
exists y:R, onSegment seg (x, y) /\ y1 <= y <= y2.

図 7 連続性に関する性質

```

Inductive embed_scurve : scurve -> list Segment -> Prop :=
| EmbedScurveNil : embed_scurve (exist _ nil IsScurveNil) nil
| EmbedScurveSgle : forall (ps:PrimitiveSegment) (s:Segment),
embed ps s
-> embed_scurve (connect ps (exist _ nil IsScurveNil) (DcNil ps)) (s::nil)
| EmbedScurveCons : forall (ps:PrimitiveSegment) (lp: scurve)
(A: dc_pseg_hd ps (proj1_sig lp)) (s1 s2: Segment) (ls: list Segment),
embed ps s1
-> embed_scurve lp (s2::ls)
-> term s1 = init s2
-> embed_scurve (connect ps lp A) (s1 :: s2 :: ls).

```

図 8 *scurve* の埋め込み

が、 (n, e, cx) を最初の単位セグメントにもち、右方向にいくらか進み、最後は左方向に進んだあと $(n, w, -)$ で終わる場合、右に行く往路と左に行く復路で共通の x に対して y 復路が y 往路より短い場合、このセグメントはどこかで交わることを表す。

これらの補題は時間をかければ証明可能だと思われるが現時点では未証明である。これらが証明されると 3 例とも同じ道筋によって証明が完了する。

3.3 簡約操作の性質

まず、回転の向きの列として *Direction* という項を

定義する (図 10)。

続いて簡約規則を *Direction* のリスト上の関数として定義する (図 11)。

簡約操作については 2.3 節に示した命題 1 の 5 つの性質をそれぞれ形式化する (図 12)。このうち 1 から 4 の証明に取り組み、いずれも証明は完了した。

1, 2 についての証明は単純である。

3 の簡約形については、手証明においては + と −、および向きの列を前から見たときと後ろから見たときの対称性を考慮し、1 つの代表例を証明していたが、Rocq においては対象性を考慮せずすべてのパターン

```

補題 (1)
Lemma extended_segment_init_n : forall (ls: list Segment) (h: H) (y x1 y1: R),
  let head := head_seg ls default_segment in embed (n, h, cx) head ->
  onHeadSegment head (x1, y1) ->
  y <= y1 ->
  exists (x: R), onHeadSegment (head_seg ls default_segment) (x, y) /\
  (match h with e => x <= x1 | w => x1 <= x end).

補題 (2)
Lemma end_nwcx_close: forall sc s_start s_end c x y1 y2 l s1 center s2 r,
  embed_scurve sc (l ++ [s1] ++ center ++ [s2] ++ r) ->
  head (l ++ [s1]) = Some s_start ->
  embed (n, e, cx) s_start ->
  last ([s2] ++ r) default_segment = s_end ->
  embed (n, w, c) s_end ->
  all_same_h (l ++ [s1]) e ->
  all_same_h ([s2] ++ r) w ->
  onSegment s1 (x, y1) ->
  onSegment s2 (x, y2) ->
  y2 < y1 ->
  close (l ++ [s1] ++ center ++ [s2] ++ r).

```

図 9 許容可能でない証明で導入した補題

```

Inductive Direction : Set :=
| Plus : Direction
| Minus : Direction
.

Definition orn (x:PrimitiveSegment) : Direction :=
match x with
| (n, e, cx) | (s, e, cx) | (s, w, cc) | (n, w, cc) => Plus
| (s, w, cx) | (s, e, cc) | (n, e, cc) | (n, w, cx) => Minus
end.

Definition rotation_difference (ds: list Direction) : Z :=
fold_right Z.add 0 (map (fun d =>
match d with
| Plus => 1
| Minus => -1
end
) ds).

```

図 10 向きに関する定義

```

Inductive Rule : list Direction -> list Direction -> Prop :=
| R1Plus : Rule [Plus; Minus; Plus] [Plus]
| R1Minus : Rule [Minus; Plus; Minus] [Minus]
| R2Plus : Rule [Plus; Plus; Minus; Minus] [Plus; Minus]
| R2Minus : Rule [Minus; Minus; Plus; Plus] [Minus; Plus]
.

Inductive ReduceDir : list Direction -> list Direction -> Prop :=
| RDRefl : forall ds, ReduceDir ds ds
| RDTrans : forall ds ds' ds'', ReduceDirStep ds ds' -> ReduceDir ds' ds'' ->
ReduceDir ds ds''

Definition Reduce (p p': list PrimitiveSegment): Prop :=
ReduceDir (map orn p) (map orn p').

```

図 11 簡約規則の定義

を陽に記述し場合分けして証明を行った。

4 の合流性については、複数の簡約規則が適用可能な部分列が重なっていない場合は簡単に証明できる。重なっている場合はいずれを先に適用しても最終的に

は同一の形に簡約されることを示す。これについても対称性を考慮せず全パターン 13 個を陽に記述し、これ以外のパターンがないこと、および、各パターンにおいて成り立つことを証明した。

```

1. 停止性
Lemma termination :
  forall x, exists y, ReduceDir x y /\ ~ CanReduceDirStep y.

2. 回転差保持
Lemma rotation_difference_preservation:
  forall (ds ds': list Direction), ReduceDir ds ds' ->
    rotation_difference ds = rotation_difference ds'.

3(a). 簡約形
Lemma reduced_form : forall (x: list Direction), ~ CanReduceDirStep x
  -> exists (l m n:nat), (l <= 1) /\ (n <= 1) /\
    (x = (List.repeat Plus l) ++ (List.repeat Minus m) ++ (List.repeat Plus n) \/
      x = (List.repeat Minus l) ++ (List.repeat Plus m) ++ (List.repeat Minus n)).

3(b). 簡約形
Lemma ReduceDir_preserve_both_ends: forall ds ds',
  ReduceDir ds ds' -> head ds = head ds' /\ last_opt ds = last_opt ds'.

4. 合流性
Lemma ReduceDir_local_confluence src dst1 dst2:
  ReduceDirStep src dst1 ->
  ReduceDirStep src dst2 ->
  have_common_reduce dst1 dst2.

5. 許容可能性の保持
Lemma AdmissibleDirs_preserve ds ds':
  AdmissibleDirs ds -> ReduceDir ds ds' -> AdmissibleDirs ds'.

```

図 12 簡約の性質

次に、命題 2 を形式化する (図 13).

対称性を考慮しなければ図 5 に示したパターンは 28 個になる. 任意の長さの向きの列として可能なパターンを生成していき, それぞれについて 28 個のいずれかと一致するか, そうでなければ簡約形の条件に矛盾することを示すことで証明は完了した. この証明においては, 整数計算を自動的に行うタクティック `lia` の利用が有効だった.

4 関連研究

定理証明支援系を使った図形の性質の証明は 4 色問題をはじめとして多くあるが [6], 初等幾何の性質を扱ったものやトポロジーとして完全に抽象化したものが多く, 記号表現を対象としたものはほとんどない. 高橋らは図形の構成要素をオブジェクトとしてみなした定性空間表現である PLCA を Coq で形式化した [11][16]. Grüniger らは自動定理証明器 Prover9 を使って領域の包含関係を定性的に取り扱う体系のモデル論的な考察をしている [7]. これらの研究はいずれも記号表現と図形の対応関係を証明支援系を使って形式化する試みであるが, 離散構造を対象としている. 本研究の大きな特徴は, 記号表現した図形データと平面上の連続データを対応させていることであり,

形式化も独特のものになっている.

5 おわりに

本発表では曲線の定性表現の Rocq による形式化について述べた. 具体的には, 方向と凸性を属性としてもつ単位セグメントを連結して構成される *scurve* について以下の 2 つを行った.

1. *scurve* の 2 次元平面への埋め込みを形式化し, 特定の *scurve* が許容可能でないことの証明のために必要な公理を用意し補題を作成した.
2. *scurve* の回転の向きを使った簡約手法を形式化してその性質を証明し, 簡約形が (対称性を考慮しなければ) 28 個のいずれかになることを証明した.

定理証明支援系では曲線の定性表現と 2 次元平面を関係づける事例は存在せず, この結果は最初のアプローチとして定理証明支援系の応用分野の拡張に寄与すると考える.

Rocq のコード一式は [12] に掲載している. 定義, 証明を含めてすべてで約 2200 行になっているが, 今回は証明の長さや効率などは考慮していない. 今回公理や補題で残している部分を完成させるとコード量は大幅に増加する. また, 証明にあたっては `ssre-`

簡約形の全パターン

Definition all_admissibles :=

```
[
  (* + *)
  [Plus];
  [Minus];
  (* +- *)
  [Plus; Minus];
  [Minus; Plus];
  (* ++ *)
  [Plus; Plus];
  [Minus; Minus];
  :
  :
  [Minus; Plus; Plus; Plus; Plus; Plus; Plus; Minus];
  [Plus; Minus; Minus; Minus; Minus; Minus; Plus]
].
```

命題 2

Theorem reduced_admissible_form :

forall ds dsr, AdmissibleDirs ds -> reduced ds dsr -> In dsr all_admissibles.

図 13 全簡約形の形

flect/mathComp [19]などのライブラリは使用していない。ライブラリを有効利用することで効率的な証明ができると思われる。

今後の大きな課題としては以下のものがある。

まず、許容可能性について公理や補題として残している部分の証明を完成させることである。これらの証明には多大な時間が必要だが、よいライブラリが利用できれば証明は効率的に進行するかもしれない。また、簡約過程の性質として許容可能性の保持の証明が未着手である。これは向きの列の上で証明できるものではないが、対偶をとって許容可能でないことを証明のゴールにすれば、許容不可能である 3 例の証明と同様の筋道で証明ができると考えている。

謝辞 本研究は JSPS 科研費補助金 JP24K15096 の助成を受けたものである。プロジェクト初期の段階で森口草介氏には有用なコメントをいただきました。また、Rocq による証明には杉泊直也氏、山口洋氏にご協力いただきました。記して感謝します。

参考文献

- [1] Cabedo, L. M. and Escrig, M. T.: A qualitative theory for shape representation and matching for design, *Proceedings of the Sixteenth European Conference on Artificial Intelligence, ECAI'2004*, 2004, pp. 858–862.
- [2] Chen, J., Cohn, A., Liu, D., Wang, S., Ouyang, J., and Yu, Q.: A survey of qualitative spatial representations, *The Knowledge Engineering Review*, Vol. 30(2013), pp. 106–136.
- [3] Cohn, A. and Renz, J.: Qualitative Spatial Representation and Reasoning, *Handbook of Knowledge Representation*, Elsevier, 2008, chapter 13.
- [4] coq: Rocq-prover, <https://rocq-prover.org/>.
- [5] Galton, A. and Meathrel, R.: Qualitative outline theory, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, (1999), pp. 1061–1066.
- [6] Gonthier, G.: Formal proof- the four color theorem, *Notices of American Mathematical Society*, Vol. 55, No. 11(2008), pp. 1382–1393.
- [7] Grüninger, M. and Aameri, B.: A New Perspective on the Mereotopology of RCC8, *The 13th International Conference on Spatial Information Theory (COSIT 2017)*, 2017, pp. 2:1–2:13.
- [8] Leroy, X.: A Formally Verified Compiler Back-end, *J. of Automated Reasoning*, Vol. 43, No. 4(2009), pp. 363–446.
- [9] Leyton, M.: A process-grammar for shape, *Artificial Intelligence*, Vol. 34(1988), pp. 213–247.
- [10] Ligozat, G.: *Qualitative Spatial and Temporal Reasoning*, Wiley, 2011.
- [11] Moriguchi, S., Goto, M., and Takahashi, K.: Towards Verified Construction for Planar Class of a Qualitative Spatial Representation, *SCSS 2016: 7th International Symposium on Symbolic Computation in Software Science*, 2016, pp. 117–129.
- [12] proof ninja: Scurve proof repository, <https://github.com/proof-ninja/coq-scurve/>.
- [13] Sioutis, M. and Wolter, D.: Qualitative spatial and temporal reasoning: current status and future challenges, *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 2021, pp. 4594–4601.
- [14] Takahashi, K.: Qualitative Formalization of a Curve on a Two-Dimensional Plane, *The 16th International Conference on Spatial Information Theory (COSIT 2024)*, 2024, pp. 4:1–4:19.

- [15] Takahashi, K.: Reasoning about the Embedded Shape of a Qualitatively Represented Curve, *SCSS 2024 WIP: 10th International Symposium on Symbolic Computation in Software Science - Work in Progress Workshop*, 2024, pp. 113–118.
- [16] Takahashi, K., Moriguchi, S., and Goto, M.: Formalization of a Surface Subdivision Allowing a Region with Holes without Coordinates, *11th International Workshop on Automated Deduction in Geometry (ADG2016)*, 2016, pp. 190–207.
- [17] Taniuchi, Y. and Takahashi, K.: Spatial Representation and Reasoning about Fold Strata: A Qualitative Approach, *15th International Conference, ICAART 2023, Revised Selected Papers*, 2024, pp. 244–266.
- [18] Tosue, M. and Takahashi, K.: Towards a qualitative reasoning on shape change and object division, *14th International Conference on Spatial Information Theory (COSIT 2019)*, 2019, pp. 7:1–7:15.
- [19] 萩原学, アフェルト・レナルド: *Coq/SSReflect/MathComp* による定理証明: フリーソフトではじめる数学の形式化, 森北出版, 2018.
- [20] 高橋和子: 曲線の定性的扱いと自己交差性の判定, *IPSJ, SIG-PRO-149*, 2024.