ブーリアン　インベッディング

1M－6

山口　人生
神奈川大学理学部情報科学科

1．はじめに

対象システムをモデル化する際、様々な抽象的構造が浮かび上がってくる。その中でも、“部分順序構造”（特別な場合として“木構造”を含む）と“ブール代数構造”は、共に広範囲の応用可能性を秘めている。ここで、ブール代数は、ある種の制約を満たす束のことであり、この意味で、部分順序構造の特殊例になっている。ところが、一方で、任意の部分順序構造はブール代数へ、1：1にembedできるという便利な性質がある。これにより、写像先のブール代数の上限、下限、補元の秩序を、元の部分順序構造が利用できることになる。今回の発表は、このような1：1 embeddingに関する話題である。以下、議論の過度の一般化を避けるため、考察の対象となる部分順序構造は有限集合であると仮定する。

2．Embedding

さて、任意の有限部分順序構造システムが与えられた時、それを何らかのブール代数システムへembedするには、様々なアルゴリズムを考えることができる。例えば、元の部分順序構造と比較して、ブール代数を十分大きく取りさえしたら、ほぼ常識的に、任意の部分順序構造はembedできることが判る。また、ある大きさのブール代数にembedできることが判ったならば、それ以上の大きさのブール代数には、必ずembedできることも明らかである。よって、ここでは、多種の可能性の中から、特に、行き先のブール代数の候補として、

「“できるだけ”小さなもの」・・・（1）

という制約を満たすようなアルゴリズムを考えることにする。この制約の“自然さ”は納得できるであろう。以下の課題は、この制約を如何に実現するかである。

さて、今、比較のため、上の制約を、次のような、“より厳密な”制約に置き換えた場合のことを考えてみよう。

「（1：1 embedできる）最小のもの」・・・（2）

実は、この制約を満たす最適アルゴリズムを構築するのは非常に難しい。条件分けによる、部分順序構造の類別化クラス数が膨大になるからである。例えば、与えられた部分順序構造を、（順序関係≦を上から下向きにとったグラフで表現して、）縦向きの“深さ”と横向きの“幅（広さ）”で類別しても、各ノードを結ぶ“順序配線”の種類により、さらに多種多様に分かれていくことから、場合分けの複雑さが推察できよう。（行き先のブール代数の配線は、大きさにより一意決定されている。）よって、準最適解として、上の条件（1）を満足するアルゴリズムを探さざるを得ない。

ここでの問題は、“できるだけ”の意味・内容を如何に解釈するかである。実は、この分野では、すでに、次のような結果が知られていた。

「与えられた部分順序構造（P，≦）が“separative”という条件を満たす時、Pのatomsを行き先のブール代数のatomsに写像するようなembeddingで、ブール代数が最小になるようなアルゴリズムが存在する。」

このようなアルゴリズムを実現するembeddingは“canonical embedding”と呼ばれる。ここで、

（P，≦）が“separative”

とは、Pが次のような条件を満たすことを言う。

$(\forall p,q\in P)(\neg(p\leq q) \rightarrow (\exists r\leq p)(\neg(\exists f(r)\in P)(f(r)\leq r \text{ and } f(r)\leq q)))$・・・（3）

上のアルゴリズムで付帯されている、「Pのatomsをブール代数のatomsに写す」という条件は、応用上、違和感のない納得のいくものである。よって、我々は、この結果を利用することにした。こ

Boolean Embedding
Jinsei Yamaguchi
Dept. of Information Science, Kanagawa Univ.

れにより、上の問題は、実質的に
「与えられた（P,≦）を如何にして、なるべく小さな separative 部分順序構造（P’,≦’）に拡張するか」
という課題に変換されることになる。

3．Separativization

さて、一口に“（P,≦）をなるべく小さな separative（P’,≦’）に拡張する”といっても様々な方式が考えられる。我々の採用した方式は、「Normal Separativization」というアルゴリズムである。以下、これを提示するが、その前に、使用する用語の準備をしておこう。

定義３－１．（P,≦）を有限部分順序構造とする。
１．Pの各点 p に対し、p の rank を次のように定める。
rank(p)＝max{length(ℓ) | ℓ は極大元から p までの path}
ここで、length(ℓ) ＝“ℓ 上の nodes 数”－１。
２．（P,≦）が separative でない時、（３）を満足しない、即ち、条件
$(\exists p,q \in P)(\neg (p \leqq q) \text{ and } (\forall r \leqq p)((\exists f(r) \in P)(f(r) \leqq r \text{ and } f(r) \leqq q)))$・・・（４）
を満たす pair (p,q)が少なくとも一つ存在する。この pair (p,q)を（P,≦）の non-separativity の“witness”と呼ぶ。以下、便宜上、（P,≦）の non-separativity の witness の集合を W（P）と書く。
３．(p,q) ∊ W（P）の時、条件（４）を満たす f(r) のうち、特に、f(p)を(p,q)に選択された“target”と呼ぶ。以下、便宜上、(p,q)に選択された targets の集合を T(p,q)と書く。
４．f(p) ∊ T(p,q)の時、p と f(p)を結ぶ path 上で、f(p)の直前の点 s を f(p)の“bearding point”と呼ぶ。以下、便宜上、f(p)の bearding points の集合を B(f(p))と書く。
５．任意の p∊P に対し、Pに無い新しい点 a を下向きに追加して、新しい部分順序構造
（P’,≦’）＝（P,≦）∪{(a,p)}
を得ることを、Pの点 p に“髭を生やす”という。
┤

以上の準備のもとで、

Normal Separativization

Step1．
If，（P,≦）が“直後の node が一つだけになる点”を持つ時
Then，それらの各点に髭を生やして、新しい（P’,≦’）を得る
Else，（P,≦）＝（P’,≦’）とする。
Step2．
If，（P’,≦’）が separative の時
Then，stop
Else，(p,q) ∊W（P’）で、特に rank(q)が最小で、rank(p)が最大のものを選ぶ。
この(p,q)に対し、f(p) ∊ T(p,q)で、rank(f(p))が最小のものを選ぶ。
この f(p)に対し、s∊B(f(p))で、rank(s)が最小のものを選ぶ。（この結果、s＝p となる。）
Step3．
s に髭を生やして、（P”,≦”）を得て、Step２へ行く。 ■

４．まとめ

任意の部分順序構造を最小のブール代数へ１：１ embed する準最適アルゴリズムを提案した。この種の問題で、ターゲットの“最小さ”を保存する最適アルゴリズムを求めることは非常に難しい。（原理上可能であるかどうかすら判っていない。）
このアルゴリズムに基づき、部分順序構造を“Boolean Complexity”で類別することが可能になる。
このアルゴリズムを、どのようにして実用的な場面で利用するかについては、また別の機会に発表する。

参考文献

[1]Yamaguchi, J., (1999) Boolean Embedding, Intern. J. Computer Math. Vol.71, pp 301-317.