

2.3 節までのまとめ

以下、簡単に Gillies 氏のテキストを振り返ります。

¶ 帰納主義と反証主義：科学的方法については対立する二つの理論がある。一つはベーコンに代表される帰納主義であり、もう一つはポッパーに代表される反証主義である。前者が与えられたデータからの機械的な推論による一般法則の導出という方法を重んじるのに対して、後者は科学者のひらめきによる仮説の形成とデータによる反証という方法を重んじる。科学史の事例を見ると、ベーコンが考えたような純粋に機械的な推論によって重要な発見がなされている例はなさそうに思われる。しかしながら機械学習と呼ばれる AI の分野の近年の発展はこの状況を変化させてきている。

¶ 機械学習：機械学習は「チューリング的伝統」の中で発展してきた AI の一分野である。これは、決まったアルゴリズムにしたがってデータと背景知識から一般化を導出するものであり、ベーコン的帰納法と適切に呼ばれうるものであるかもしれない。

¶ チューリング的伝統：チューリング的伝統は、(1) 論理の使用と、(2) 実際問題への応用の重視によって特徴付けられる。機械学習にとって (1) が重要であるのは、論理主義の提唱者のフレイゲが考えたように、推論のステップを厳密に隙間なく構成することによって、推論のメカニズムをシミュレートできるからである。これを行わずに単に人間の帰納的推論をなぞるだけでは、人間がすでに発見したことを再発見することは出来るかもしれないが、新しい事実の発見には繋がらない。

¶ エキスパート・システムと Feigenbaum's bottleneck：チューリング的伝統の実用的側面は、エキスパート・システムに現れる。これは人間のエキスパートによって行われていた推論・判断を機械によって行わせる試みである。このようなシステムを作る際には、まず人間のエキスパートがどのようにして彼らの仕事を行っているかを明確なルールとして定式化しなければいけないが、これは手間がかかる上に、エキスパート自身がそのようなルールをはっきりとは意識しておらず、うまく説明できないことが多い。これが Feigenbaum's bottleneck といわれる問題である。ところがこのようなルールを明示化するのに機械学習が有用であることが分かってきた。

¶ ID3：Quilan によって開発された ID3 は、対象の集合と属性の集合が与えられたときに、その対象の集合をある分類に関してポジティブな事例とネガティブな事例に完全に分けるような属性の束を見つけるよう設計された機械学習システムである。ID3 の顕著な特徴は general-to-specific 的、トップ・ダウン的な性格を持っているということである。ID3 は与えられた対象の集合に対して何らかの属性による分割を試みる。こうして新しく作られた部分集合に対して再び別な属性を

適用し、さらに分割をすすめる。この過程を続けて、最終的にすべての部分集合がポジティブな事例かネガティブな事例のどちらかしか含まないようにであれば、そのような分類を可能にした属性の束が最初に与えられた分類を特徴付ける性質であると考えることができる¹。

ID3 の短所は属性ベースで設計されているためにデータとして一項述語しか使えないことである。ID3 でも十分に役に立つ場合もあるが、しかし一階述語論理の資源、つまり多項関係を使いたい場合もある。そこで一階述語論理に基づいたシステムが開発されている。

大体以上が前期にやった話でした。Gillies 氏のテキストはこの先、関係ベースの機械学習システム GOLEM の説明にはいります。GOLEM はエキスパート・システムに必要な規則を生み出すだけでなく、与えられたデータと背景知識から、決まったアルゴリズムとヒューリスティックによって新しい自然法則を発見することができること、それゆえベーコンの考えた機械的な帰納法のモデルになりうるものであることが論じられます。

GOLEM は Robinson のレゾリューション原理を基にした規則を使っているということですが、そのところがあまり詳しく説明されていないので、そもそものレゾリューション原理がどのようなものであり、それが機械学習にどのように応用されているのかを以下で簡単にレポートします。

1 レゾリューション原理

1.1 前置き

AI 研究において、論理学と AI の関係、あるいは前者が後者に対して持つ意義は、一つの大きなテーマである。AI が人間の知性のモデルであると考えられるならば、この問題は、論理学が人間の思考にとってどういう意義を持っているのか、というより大きな問題にかかわってくる。現実の状況における判断と行動という側面を考えた場合、論理的思考というものの重要性・意義については疑問が持たれてきた。例えば「フレーム問題」に関する議論がある。行為者が何らかの目的をもって行動するとき、彼は無限の情報を含む環境の中で、自らの目的にとって関連性の高い情報をとりだし、そこに基づいて推論・判断し、そして行為しなければならない。このような場合、論理学というものはほとんど役に立たないと考えられている。人間の思考や行動における環境との相互作用が重視されてくるにつれて、人間的思考のモデルとしての論理学という考え方は、ますます旗色が悪くなっている。人間は与えられたデータと背景知識から仮説を立て、その仮説から予測を引き出し、その予測に基づいて行動する。この過程は必ずしも論理的・演繹的なものではない。

しかしその一方で、論理プログラミングに基づいた機械学習システムが近年大きな成功を収めてきている。これらのシステムの中には決められたアルゴリズムに従いながら帰納的推論——与

¹正確に言えばこの属性の束は、個々の属性を原子式として捉えた場合、選言標準形で表されるであろう。

えられた例からの一般化——を行うものがある。例えば Quinlan の ID3 がそのようなシステムの一つである。ID3 の短所はそれが一項述語しか利用できないため、応用の範囲が限られているということであった。そこで Quinlan は関係をベースにした学習システム FOIL を開発した。FOIL も ID3 と同様にトップ・ダウン的性格を持っている。またエントロピーに関連付けられた情報利得 information gain の規準に基づいている点も ID3 と同様である。

FOIL のようなシステムに対して Plotkin, Muggleton らはやはり関係をベースにしながら、specific-to-general 的、ボトム・アップ的な性格を持った機械学習システムを開発した。彼らのシステムはレゾリューション原理 the resolution principle (以下 RP) という原理を利用している。これはそもそも J. A. Robinson によって提案された論理体系の推論規則として採用されていたものである。Robinson 自身はこの体系を、コンピューターによる定理証明プログラムの理論的な枠組みとして構築したのであるが、Muggleton らは RP による新しい命題の導出過程を逆転させることによって、与えられたデータから新しい述語を生成させるプログラムを作り上げた。また RP 以外にも、Robinson のシステムで使われた置換 substitution, 統合 unification, 包摂 subsumption などの概念は Plotkin, Muggleton らのシステムにおいて重要な役割を果たしている。

RP の基本的なアイディアは次のようなものである。ある命題（正確には RP は命題ではなく、リテラル literals と呼ばれる命題の集合（この集合はクローズ clauses と呼ばれる）の集合を適用の対象とする）に RP を繰り返し適用することによって、もし元の命題が矛盾を含むものであれば、その矛盾を明白にするというものである。通常の論理学の体系が、公理から推論規則によって定理を導出していくのに対して、Robinson の体系は、任意の命題についてその反証を探すことで、もとの命題が定理であるかどうかを確かめる、という方法を採用している。もちろんこれはいわゆる分析的反証法的一种だが、Robinson の方法の利点は、それが RP のみを適用することから成り立っているということである。RP は通常の述語論理の公理系や証明論における推論規則と比べると非常に複雑であるが、Robinson の論文のタイトルにもあるように、RP は基本的にコンピューターに実行させるように作られた規則であり、その上では利点がある。

以下、Robinson (1965) に基づいて RP の概略を述べる。

1.2 シンタクス

この体系は通常の 1 階述語論理とは異なり、妥当性と証明の概念ではなく、充足不可能性と反証の概念に基づいて立てられている。1 階論理の文の有限集合 S が充足可能であるかどうかを決定するには、それぞれの文は存在量化子を持たない冠頭標準形²であると想定すれば十分である。なおかつそれぞれの文のマトリクス³は、原子式⁴もしくは原子式の否定を選言肢として持つ選言であると想定しても良い。

² 量化子が文の頭についている形。

³ 冠頭標準形の文から量化子を取り除いた残り。

⁴ 論理結合子を含まない論理式。

そこでこの体系は文法的な単位としてこの特殊な形式の文の有限集合を持つ。冠頭量子子は省略される。というのもそれはすべての自由変項に対する全称量子子であると決まっているからだ。さらには選言記号も使われない。問題としている集合がリテラル（後述）が複数含んでいるならば、その選言が私たちが考慮している命題であるとわかるからである。例えば集合 S が $A(x), \neg B(y, x)$ を含んでいたら、この集合は $\forall x \forall y (A(x) \vee \neg B(y, x))$ という命題を表している。

定義 1.1 (変項). 以下の記号が変項である。アルファベットの順序で並べてある：

$$u, v, w, x, y, z, u_1, v_1, w_1, x_1, y_1, z_1, u_2, \dots, \text{ etc.}$$

定義 1.2 (関数記号). 0 以上の n それぞれについて、以下の記号が n 度の関数記号である。アルファベットの順序で並べてある：

$$a^n, b^n, c^n, d^n, e^n, f^n, g^n, h^n, k^n, a_1^n, b_1^n, \dots, \text{ etc.}$$

$n = 0$ の場合は上添字は省略しても良い。0 度の関数記号は個体定項である。

定義 1.3 (述語記号). 0 以上の n それぞれについて、以下の記号が n 度の述語記号である。アルファベットの順序で並べてある：

$$P^n, Q^n, R^n, P_1^n, Q_1^n, R_1^n, P_2^n, \dots, \text{ etc.}$$

$n = 0$ のとき上添字は省略しても良い。

定義 1.4 (否定記号). $\neg$.

定義 1.5 (記号のアルファベットの順序). すべての記号の集合はアルファベットの順序によって整列される⁵。これは上の並べ方の規約に以下の規則を付け加えることで定義される。変項は関数記号に先行する。関数記号は述語記号に先行する。述語記号は否定記号に先行する。度数のより低い関数記号は度数のより高い関数記号に先行する。度数のより低い述語記号は度数のより高い述語記号に先行する。

定義 1.6 (項). 変項は項である。 n 度の関数記号とその後ろについた n 個の項からなる記号の列は項である（ただし $n \geq 0$ ）。

定義 1.7 (原子式). n 度の述語記号とその後ろについた n 個の項からなる記号の列は原子式である（ただし $n \geq 0$ ）。

定義 1.8 (リテラル *Literals*). 原子式はリテラルである。 A が原子式ならば、 $\neg A$ はリテラルである。前者を正のリテラル、後者を負のリテラルということもある。

⁵ある集合 S の元の間に順序関係 $\preceq$ が定義されており、 S の任意の二つの元 a, b に関して $a \preceq b$ か $b \preceq a$ のどちらかが成り立ち、また S の任意の部分集合において、 $\preceq$ に関する最小元が存在するとき、 S は $\preceq$ によって整列されるといふ。

定義 1.9 (補式). A が原子式ならば、 A と $\neg A$ という二つのリテラルは互いの補式であるといい、またこの二つは、どちらの順序でも、相補的対を形成するという。 L がリテラルであるとき、 $\bar{L}$ によって L の補式を表す場合がある。

定義 1.10 (クローズ *Clauses*). リテラルの有限集合 (空でも良い)。空のクローズは $\square$ によって指示される。

定義 1.11 (ホーン・クローズ *Horn clauses*). 正のリテラルを一つだけ持つクローズ。ホーン・クローズだけからなる集合を論理プログラムという場合がある⁶。

定義 1.12 (基礎リテラル). 変項を含まないリテラル。

定義 1.13 (基礎クローズ). 基礎リテラル以外を含まないクローズ。命題クローズという場合もある。 $\square$ も基礎クローズである。

定義 1.14 (適格な表現 *Well-formed expressions*). 項とリテラルは適格な表現である (以下「WFE」)。それ以外に WFE はない。見やすさのために、関数記号と述語記号の後ろにくる項を $()$ でくくり、もし項が複数あるなら間をコンマで区切ることにする。このとき上添字を省略することができる。例えば $P^2yh^3bv f^1x$ と書く代わりに $P(y, h(b, v, f(x)))$ と書く。

定義 1.15 (WFE の辞書的順序). すべての WFE は辞書的順序に整列される。この順序は次の規則によって決定される。 A が B に先行するのは、 A が B よりも短い、もしくは同じ長さで、 A と B が異なる記号を持つ最初の位置で、 A が B よりもアルファベットの順序で先行する記号を持つ場合である。

1.3 充足可能性と基礎レゾリューション、いくつかの定理

定義 1.16 (エルブラン宇宙 *Herbrand universes*). 任意のクローズの集合 S に対して、 S のエルブラン宇宙 $\mathcal{H}_S$ と呼ばれる、基礎項からなる次のような集合が結び付けられる。まず F は S に現れるすべての関数記号の集合であるとする。 F が 0 度の関数記号を含んでいるなら、 S の関数語彙は F である。もし含んでいなければ、 S の関数語彙は $\{a\} \cup F$ である。このとき $\mathcal{H}_S$ は、 S の関数語彙のみから構成されるすべての基礎項の集合である。例えば S の関数語彙が $\{a, f^1, g^1\}$ ならば $\mathcal{H}_S$ は $\{a, f(a), g(a), f(f(a)), f(g(a)), g(f(a)), g(g(a)), \dots\}$ である。

定義 1.17 (充満 *Saturation*). S は任意のクローズの集合、 P は任意の項の集合とする。このとき $P(S)$ によって、 P 上での S の充満を表す。これは S の要素に現れる変項を P の要素によって置き換えて得られるすべての基礎クローズの集合である。ただし同じ変項の現れはすべて同じ項で置き換えられるものとする。

⁶13 頁, 例 2.7 参照

定義 1.18 (モデル). 相補的な対をふくまない基礎リテラルの集合. M がモデル, S が基礎クローズの集合であるとする. S に属するすべての C が M の要素を含んでいるとき, M は S のモデルであるという.

定義 1.19 (充足可能性). クローズの集合 S にモデルが存在するとき, S は充足可能である. そうでないとき, S は充足不可能である.

定義から明らかなように, $\square$ を含むクローズの集合は充足不可能である. また空集合は充足可能である. 空でないクローズはそこに含まれるリテラルを選言肢とする選言に対して全称量化を施した閉式として解釈される.

定義 1.20 (基礎リゾルヴェント). C と D が基礎クローズであり, $L \subseteq C, M \subseteq D$ はともにシングルトン (単元集合) であり, その要素が相補的な対を形成するとき, 基礎クローズ $(C - L) \cup (D - M)$ は C と D の基礎リゾルヴェントと呼ばれる.

R が C と D の基礎リゾルヴェントであるとき, 明らかに $\{C, D\}$ の任意のモデルは $\{C, D, R\}$ のモデルでもある. 任意の基礎クローズの対は高々有限の基礎リゾルヴェントしか持たない (クローズは有限集合であることに注意).

定義 1.21 (基礎レゾリューション). S は基礎クローズからなる任意の集合であるとする. このとき S の基礎レゾリューション $\mathcal{R}(S)$ は, S の要素になっているすべての基礎クローズと, S の任意の二つの要素のすべての基礎リゾルヴェントからなる集合である.

定義 1.22 (n 次基礎レゾリューション). S は基礎クローズからなる任意の集合であるとする. このとき S の n 次基礎レゾリューション $\mathcal{R}^n(S)$ は以下のように定義される. $\mathcal{R}^0(S) = S$; 任意の $n \geq 0$ に対して, $\mathcal{R}^{n+1}(S) = \mathcal{R}(\mathcal{R}^n(S))$.

定理 1.23 (Herbrand's Theorem). S はクローズからなる任意の有限集合であり, $\mathcal{H}_S$ はそのエルブラン宇宙であるとする. このとき, S が充足不可能なのは, $\mathcal{H}_S(S)$ のある有限な部分集合が充足不可能なときであり, そしてそのときに限る.

この定理は次のようにも表現できる.

定理 1.24. S はクローズからなる任意の有限集合であるとする. S が充足不可能なのは, $\mathcal{H}_S$ のある有限な部分集合 P に対して, $P(S)$ が充足不可能なときである.

さらに S が充足不可能であり, その反証が存在するならば, 定理の条件を満たす最小限の P を構成することができる. そのような P は S についての *proof set* と呼ばれる.

定理 1.25 (Ground Resolution Theorem). S が基礎クローズからなる任意の有限集合であるとき, S が充足不可能であるのは, ある $n \geq 0$ に対して, $\mathcal{R}^n(S)$ が $\square$ を含むときであり, そしてそのときに限る.

定理 1.26. S がクローズからなる任意の有限集合であるとき、 S が充足不可能であるのは、 $\mathcal{H}_S$ のある有限な部分集合 P とある $n \geq 0$ に対して、 $\mathcal{R}^n(P(S))$ が $\square$ を含むときであり、そしてそのときに限る。

補題 1.27. S がクローズからなる任意の集合であり、 P が $\mathcal{H}_S$ の任意の部分集合であるとき、 $\mathcal{R}(P(S)) \subseteq P(\mathcal{R}(S))$ 。

系 1.28. S がクローズからなる任意の集合であり、 P が $\mathcal{H}_S$ の任意の部分集合であるとき、すべての $n \geq 0$ に対して、 $\mathcal{R}^n(P(S)) \subseteq P(\mathcal{R}^n(S))$ 。

定理 1.29. S がクローズからなる任意の有限集合であるとき、 S が充足不可能であるのは、 $\mathcal{H}_S$ のある有限の部分集合 P とある $n \geq 0$ に対して、 $P(\mathcal{R}^n(S))$ が $\square$ を含むときであり、そしてそのときに限る。

1.4 置換, 統合, レゾリューション, いくつかの定理

定義 1.30 (置換要素). V が任意の変項であり、 T は V と異なる任意の項であるとき、 T/V を置換要素という。 V を要素 V/T の変項といい、 T を要素 V/T の項という。

定義 1.31 (置換). 置換要素の任意の有限集合 (空である場合も許す) で、各要素の変項がどれも異なっているとき、この集合を置換という。 P が項からなる任意の集合で、置換 θ の要素の項がすべて P に属するとき、 θ は P 上の置換であるという。要素として $T_1/V_1, \dots, T_k/V_k$ を持つ置換を $\{T_1/V_1, \dots, T_k/V_k\}$ と書く。このとき要素の順序はいつでもよい。ギリシャ文字の小文字を置換を表すために用いる。特に ϵ は空置換を表す。

定義 1.32 (例化). E は任意の有限の記号列であり、 $\theta = \{T_1/V_1, \dots, T_k/V_k\}$ は任意の置換であるとする。このとき θ による E の例化とは、 E における変項 V_i ($1 \leq i \leq k$) の各現れを、項 T_i の現れによって置き換える操作である。その結果として得られる記号列を E の θ による例と呼び、 $E\theta$ によって表す。 C が記号列の任意の集合であり、 θ が置換であるとする、 C の θ による例とは、 $\{E\theta | E \in C\}$ である。

定義 1.33 (標準化). C は記号列の有限集合であり、 $V_1, \dots, V_k$ は C に現れるすべての異なる変項をアルファベットの順序で並べたものである。このとき、 C の x -標準化は、置換 $\{x_1/V_1, \dots, x_k/V_k\}$ であり、 ξ_C と表す。また C の y -標準化は置換 $\{y_1/V_1, \dots, y_k/V_k\}$ であり、 η_C と表す。

定義 1.34 (置換の合成). $\theta = \{T_1/V_1, \dots, T_k/V_k\}$ と λ は任意の二つの置換であるとする。 λ' は λ の要素のうち、その変項が $V_1, \dots, V_k$ のどれとも異なっているものの集合であるとする。また θ' は、 $T_i\lambda$ が V_i とは異なっているような要素 $T_i\lambda/V_i$ ($1 \leq i \leq k$) すべてからなる集合であるとする。このとき $\theta' \cup \lambda'$ を θ と λ の合成といい、 $\theta\lambda$ によって表す。

事実 1.35. 置換の合成は結合性を持つ．つまり任意の置換 σ, θ, λ について $(\sigma\theta)\lambda = \sigma(\theta\lambda)$ が成り立つ．また E が任意の表現であり， θ, λ が任意の置換であるとする， $(E\theta)\lambda = E(\theta\lambda)$ が成り立つ．したがってこれらの括弧を省略しても構わない．

事実 1.36. θ, λ を任意の置換とする．すべての表現 E について， $E\theta = E\lambda$ ならば $\theta = \lambda$ が成り立つ．

事実 1.37. 任意の記号列の集合 A, B と置換 λ について， $(A \cup B)\lambda = A\lambda \cup B\lambda$ が成り立つ．

定義 1.38 (不一致集合 *Disagreement sets*). A は WFE の任意の集合であるとする． A に属する WFE の部分 WFE で， A に属する WFE のすべてが同じ記号を共有しているのではない，最初の場所に現れるものすべての集合を A の不一致集合という．

例 1.39.

$$\begin{aligned} A &= \{P(x, h(x, y), y), P(x, k(y), y), P(x, a, b)\} \\ A \text{ の不一致集合} &= \{h(x, y), k(y), a\} \end{aligned}$$

定義 1.40 (統合 *Unification*). A が WFE の任意の集合であり， θ が置換であるとする． $A\theta$ がシングルトンであるとき， θ は A を統合する，あるいは θ は A の統合子であるといわれる．統合子を持つ WFE の任意の集合は，統合可能であるという．

明らかに， θ が A を統合するが A がシングルトンではないのであれば， θ は A の不一致集合を統合する．

定義 1.41 (統合アルゴリズム). WFE からなる任意の有限かつ空でない集合 A に対して適用される以下の手続きを，統合アルゴリズムという．

手順 1. $\sigma_0 = \epsilon, k = 0$ とし，手順 2 へ進め．

手順 2. $A\sigma_k$ がシングルトンでないならば手順 3 に進め．そうでなければ $\sigma_A = \sigma_k$ として停止せよ．

手順 3. $A\sigma_k$ の不一致集合 B_k の要素を辞書的順序で並べたときに，最初に来るものを V_k とし，二番目に来るものを U_k とせよ． V_k が変項であり U_k の中に現れないのであれば， $\sigma_{k+1} = \sigma_k\{U_k/V_k\}$ とし， k に 1 加えて，手順 2 に戻れ．そうでなければ停止せよ．

定義 1.42 (最汎統合子 *Most general unifiers*). A は WFE の有限かつ空でない任意の集合． A の統合アルゴリズムが手順 2 で停止するとき，その結果として得られる置換 σ_A は， A の最汎統合子という．またこのとき A は最汎統合可能であるという．

定義 1.43 (キー・トリプル *Key triples*). リテラルの有限集合 L, M, N の順序付けられた三つ組 $\langle L, M, N \rangle$ は，以下の条件を満たすとき，クローズ C, D の順序付けられた対 $\langle C, D \rangle$ のキー・トリプルという．

- (i) L と M は空ではなく, $L \subseteq C, M \subseteq D$.
- (ii) N は, 集合 $L\xi_C \cup M\eta_D$ の要素である原子式か, 要素の補式である原子式からなる集合.
- (iii) N は最汎統合子 σ_N によって最汎統合可能である.
- (iv) 集合 $L\xi_C\sigma_N$ と集合 $M\eta_D\sigma_N$ はシングルトンであり, その要素は互いに補式である.

定義 1.44 (リゾルヴェント). C と D がクローズであり, $\langle L, M, N \rangle$ が $\langle C, D \rangle$ のキー・トリプルであるとき, クローズ $(C - L)\xi_C\sigma_N \cup (D - M)\eta_D\sigma_N$ を, C と D のリゾルヴェントと呼ぶ.

注意 1.45. 明らかに $\langle C, D \rangle$ は高々有限個のキー・トリプルを持つ. したがって C, D は高々有限個のリゾルヴェントを持つ.

定義 1.46 (レゾリューション). S はクローズからなる任意の集合であるとする. このとき S のレゾリューション $\mathcal{R}(S)$ は, S の要素と, S の要素のリゾルヴェントすべてからなる集合である.

定義 1.47 (n 次レゾリューション). S はクローズからなる任意の集合とする. S の n 次レゾリューション $\mathcal{R}^n(S)$ は, n 次基礎レゾリューションとまったく同様に定義される.

注意 1.48. 前注意から分かるように, クローズからなる任意の集合 S に関してある $n \geq 0$ が存在し, $m \geq n$ なる任意の m について $\mathcal{R}^n(S) = \mathcal{R}^m(S)$ が成り立つ.

定理 1.49 (Resolution Theorem). S がクローズからなる任意の有限集合であるとき, S が充足不可能であるのは, ある $n \geq 0$ に対して, $\mathcal{R}^n(S)$ が $\square$ を含むときであり, そしてそのときに限る.

定理 1.50 (Unification Theorem). A は WFE の有限かつ空でない任意の集合とする. A が統合可能であれば A は最汎統合子 σ_A によって最汎統合可能である. なおかつ A の任意の統合子 θ について, ある置換 λ が存在し, $\theta = \sigma_A\lambda$ が成り立つ.

補題 1.51. S はクローズからなる任意の集合, P は $\mathcal{H}_S$ の任意の部分集合とする. このとき $\mathcal{R}(P(S)) \subseteq P(\mathcal{R}(S))$ が成り立つ.

定義 1.52 (包摂 Subsumption). C, D が空でない二つの異なるクローズであり, ある置換 θ について, $C\theta \subseteq D$ が成り立つとき, C は D を包摂するという.

注意 1.53. クローズをそこに含まれるリテラルからなる選言と同一視したとき, 二つのクローズ C, D の間に集合としての包含関係 $C \subseteq D$ が成り立つならば, リテラルの選言としての論理的帰結関係 $C \models D$ が成り立つ.

事実 1.54. C を任意の統合可能なクローズとし, σ_C をその最汎統合子であるとする. このとき C の任意の統合子 θ について, $C\sigma_C$ は $C\theta$ を包摂する.

注意 1.55. C が D を包摂するとき, 「 C は D よりも一般的である」と言うことにすれば, 上の事実は, 最汎統合子によって統合されたクローズは, 他のいかなる統合子によって統合されたクローズよりも一般的である, と言い換えることができる. 最汎統合子という名前はこの性質に由来している.

1.5 RP と反証

定義 1.56 (レゾリューション原理 (RP)). 任意の二つのクローズ C と D から C と D のリゾルヴェントを推論してよい.

RP は Robinson の体系の唯一の推論規則である.

定義 1.57 (反証 *refutation*). クローズの有限列 $B_1, \dots, B_n$ が以下の条件 (a)(b) を満たすとき, この列はクローズの集合 S の反証であるという.

- (a) 各 B_i ($1 \leq i \leq n$) について, B_i は S の要素であるか, この列の前にある二つのクローズのリゾルヴェントであるかのどちらかである.
- (b) B_n は $\square$ である.

Resolution Theorem から直ちに, クローズからなる有限集合 S が充足不可能なのは, S の反証が存在するときであり, そしてそのときに限る, ということが導かれる. したがって Resolution Theorem はこの体系における完全性定理である.

例 1.58. 集合 A が次のような二つのクローズ, C_1 と C_2 を含んでいる.

$$\begin{aligned} C_1 &= \{Q(x, g(x), y, h(x, y), z, k(x, y, z))\} \\ C_2 &= \{\neg Q(u, v, e(v), w, f(v, w), x)\} \end{aligned}$$

このとき A は反証 $C_1, C_2, \square$ を持つ. $\langle C_1, C_2 \rangle$ はキー・トリプル $\langle C_1, C_2, N \rangle$ を持つことに注意せよ. ただしここで N は次の二つの原子式を含む集合である.

$$\begin{aligned} &Q(x_1, g(x_1), x_2, h(x_1, x_2), x_3, k(x_1, x_2, x_3)) \\ &Q(y_1, y_2, e(y_2), y_3, f(y_2, y_3), y_4) \end{aligned}$$

これらは C_1, C_2 に含まれるリテラルもしくはその補式にそれぞれ x -標準化, y -標準化を施して得られた原子式である. N を最汎統合する最汎統合子 σ_N は, 統合アルゴリズムを以下のように遂行することで得られる.

1. $\sigma_0 = \epsilon, k = 0$.
2. $B_0 = \{x_1, y_1\}, \sigma_1 = \sigma_0\{y_1/x_1\}$.
3. $N\sigma_1 = \{Q(y_1, g(y_1), x_2, h(y_1, x_2), x_3, k(y_1, x_2, x_3)), \\ Q(y_1, y_2, e(y_2), y_3, f(y_2, y_3), y_4)\}$.
4. $B_1 = \{y_2, g(y_1)\}, \sigma_2 = \sigma_1/\{g(y_1)/y_2\}$.
5. $N\sigma_2 = \{Q(y_1, g(y_1), x_2, h(y_1, x_2), x_3, k(y_1, x_2, x_3)), \\ Q(y_1, g(y_1), e(g(y_1)), y_3, f(g(y_1), y_3), y_4)\}$.
6. $B_2 = \{x_2, e(g(y_1))\}, \sigma_3 = \sigma_2\{e(g(y_1))/x_2\}$.

7. $N\sigma_3 = \{Q(y_1, g(y_1), e(g(y_1))), h(y_1, e(g(y_1))), x_3, k(y_1, e(g(y_1))), x_3),$
 $Q(y_1, g(y_1), e(g(y_1))), y_3, f(g(y_1), y_3), y_4)\}.$
8. $B_3 = \{y_3, h(y_1, e(g(y_1)))\}, \sigma_4 = \sigma_3\{h(y_1, e(g(y_1)))/y_3\}.$
9. $N\sigma_4 = \{Q(y_1, g(y_1), e(g(y_1))), h(y_1, e(g(y_1))), x_3, k(y_1, e(g(y_1))), x_3),$
 $Q(y_1, g(y_1), e(g(y_1))), h(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1)))), y_4)\}.$
10. $B_4 = \{x_3, f(g(y_1), h(y_1, e(g(y_1))))\}, \sigma_5 = \sigma_4\{f(g(y_1), h(y_1, e(g(y_1))))/x_3\}.$
11. $N\sigma_5 = \{Q(y_1, g(y_1), e(g(y_1))), h(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1))))),$
 $k(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1))))),$
 $Q(y_1, g(y_1), e(g(y_1))), h(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1))))), y_4\}.$
12. $B_5 = \{y_4, k(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1))))),$
 $\sigma_6 = \sigma_5\{k(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1)))))/y_4\}.$
13. $N\sigma_6 = \{Q(y_1, g(y_1), e(g(y_1))), h(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1))))),$
 $k(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1))))),$
 $Q(y_1, g(y_1), e(g(y_1))), h(y_1, e(g(y_1))), f(g(y_1), h(y_1, e(g(y_1))))), y_4\}.$
14. $B_6 = \emptyset, \sigma_N = \sigma_6.$

このとき $C_1\xi_{C_1}\sigma_N$ と $C_2\eta_{C_2}\sigma_N$ はシングルトンで、その要素は互いに補式になっている。

2 機械学習における RP の応用

レゾリューションは、クローズの集合 S からリゾルヴェント C を導出する規則である。 S のある部分集合 $\{C_1, C_2\}$ が与えられたとき、これに対してリゾルヴェントは存在しないかもしれないし、複数存在するかもしれない。しかしいずれにせよこの導出の手続きは有限である。特に C_1, C_2 のどちらかがシングルトンの場合はこの手続きは決定的 determinisitic なものになる。この推論の過程を逆にして、 C_1 と C が与えられたとき、 C_1 と C_2 のリゾルヴェントが C になるような C_2 を導出するような機械的な手続きが与えられるだろうか？ この過程には定項を変項に変えるという操作が含まれる。したがってこれは具体的な対象に関するデータが与えられているときに、その前提となる一般的な仮説を構築する手続きと考えられる。このような手続きが与えられるならば、それはベーコン的な帰納法を指示する強い根拠を与えるだろう。ただし気をつけなければならないのは、 S が充足可能であれば、 C も充足可能であるのに対して、この逆は必ずしも成り立たないので、ここにおいても仮説が反証されるかどうかのテストが不可欠だということである。

当然予想されることであるが、このような手続きは完全に決定的にはならない。しかしながら様々な制約をつけることで非決定性を減らすことは出来る。それらの制約はいわばヒューリスティックとして機能するものである。

2.1 逆レゾリューション

Muggleton と Buntine は RP を逆転させた過程によって一階述語を新しく「発明」する CIGOL というシステムを作った（この名前は LOGIC のつづりを逆にしたものである）。以下, Muggleton and Buntine (1997) を部分的に要約する。

定義 2.1 (場所). 項あるいはリテラル, あるいはそれらの列の中に項が現れる場所は次のように再帰的な仕方で定義される。

- (i) $f(t_1, \dots, t_n)$ の中の場所 $\langle a_1 \rangle$ にある項は t_{a_1} である。
- (ii) $f(t_1, \dots, t_n)$ の場所 $\langle a_1, \dots, a_m \rangle$ にある項は t_{a_1} の中の場所 $\langle a_2, \dots, a_m \rangle$ にある項である。

定義 2.2 (逆置換). t は何らかの項あるいはリテラルであり, θ はある置換であるとする。このとき, $t\theta\theta^{-1} = t$ であるとき, θ^{-1} は θ の逆置換であるといわれる。

例えば $\theta = \{t_1/v_1, \dots, t_n/v_n\}$ であるとするこの逆置換は,

$$\theta^{-1} = \{v_1/(t_1, \{p_{1,1}, \dots, p_{1,m_1}\}), \dots, v_n/(t_n, \{p_{n,1}, \dots, p_{n,m_n}\})\}$$

と表される。ただし p_{i,m_j} は t 中の変項 v_i の場所を表している。

例 2.3. $L = \text{likes}(x, \text{brother}(x)), \theta = \{\text{john}/x\}$ とすると,

$$L\theta = \text{likes}(\text{john}, \text{brother}(\text{john})), \theta^{-1} = \{x/(\text{john}, \{\langle 1 \rangle, \langle 2, 1 \rangle\})\}$$

ここで $\langle 1 \rangle, \langle 1, 2 \rangle$ は, L の中の変項 x の現れる場所を表している。

定義 2.4 (θ -差). 二つの項もしくはリテラル t_1, t_2 が共通の変項を持っていないとする。このとき, $t_1\theta = t_2$ であり, かつ $\text{domain}(\theta) \subseteq \text{vars}(t_1)$ であるとき, 置換 θ を t_1 と t_2 の θ -差と呼ぶ⁷。それ以外の場合は差は未定義である。差が定義されているとき, それを中置関係 $\theta = t_1 -_{\theta} t_2$ として表す。実際に二つの項の間の θ -差は一意的であり, 以下のように再帰的な仕方で定義される。

- (i) $v_1 -_{\theta} t_2 = \{t_2/v_1\}$ if v_1 が変項。
- (ii) $f(r_1, \dots, r_n) -_{\theta} f(s_1, \dots, s_n) = \bigcup_{1 \leq i \leq n} (r_i -_{\theta} s_i)$ if f が n 度の述語もしくは関数記号。

$t_1 -_{\theta} t_2$ が定義されているとき, t_1 は t_2 を包摂するという。

注意 2.5. ここで定義された項またはリテラルの間の包摂の概念と, 前節で定義されたクローズの間の包摂の概念との間には次のような関係がある。 C, D はクローズの空でない集合であり, $C = \{L_1, \dots, L_n\}, D = \{L'_1, \dots, L'_m\}$ とする。 C が D を前節の意味で包摂するのは任意の L_i についてある L'_j が存在して, L_i が L'_j をここでの意味で包摂するときであり, そしてそのときに限る。

⁷ θ が置換であるとき, θ に属するすべての構成要素の変項の集合を θ のドメインといい, $\text{domain}(\theta)$ と書く。また任意の表現 e に含まれる変項すべての集合を $\text{vars}(e)$ によって表す。また表現からなる任意の集合 $E = \{e_1, \dots, e_n\}$ について $\text{vars}(E) = \bigcup_{1 \leq k \leq n} \text{vars}(e_k)$ とする。

例 2.6. 項 $t_1 = plus(x, y)$ は項 $t_2 = plus(3, 4)$ を包摂する. というのも $t_1 -_\theta t_2 = \{3/x, 4/y\}$ と定義されるから.

レゾリューションの定義を少し修正して述べておこう. C_1, C_2 はクローズとする. L_1, L_2 はそれぞれ C_1, C_2 に属するリテラルであり, $\overline{L_1}$ と L_2 が最汎統合子 θ を持つものとする (定義 1.9 参照). このとき $C = (C_1 - \{L_1\})\theta \cup (C_2 - \{L_2\})\theta$ を C_1 と C_2 のリゾルヴェントといい, $C_1 \bullet C_2$ と書く. 置換 θ は $\theta = \theta_1\theta_2$ かつ $domain(\theta_1) \subseteq vars(C_1)$ かつ $domain(\theta_2) \subseteq vars(C_2)$ を満たすような, 構成部分 θ_1, θ_2 に一意的に分解される. ここから

$$C = (C_1 - \{L_1\})\theta_1 \cup (C_2 - \{L_2\})\theta_2 \quad (1)$$

が得られる.

レゾリューションは理論 (ここではクローズの集合と考える) から帰結を導き出すのに使われる. ある理論 T について, L が T の帰結であるのは $T \cup \{\overline{L}\}$ が充足不可能であると示されるときであり, そしてそのときに限る. 前章の定理 1.49 より, これは $T \cup \{\overline{L}\}$ に RP を有限回適用して反証が得られることと同値である.

例 2.7. 理論 T はクローズ $C_1 = \{larger(hammer, feather)\}, C_2 = \{denser(hammer, feather)\}, C_3 = \{heavier(x, y), \neg denser(x, y), \neg larger(x, y)\}$ からなるものとする. $C = \{heavier(hammer, feather)\}$ が T の帰結であることをは次のように示される. まず C_3 と $\{\neg heavier(hammer, feather)\}$ から RP によって, $C_4 = \{\neg denser(hammer, feather), \neg larger(hammer, feather)\}$ が得られる. 次に C_2 と C_4 から RP によって $C_5 = \{\neg larger(hammer, feather)\}$ が得られる. 最後に C_1 と C_5 から $\square$ が得られる. したがって $T \cup C$ には反証が存在する. T は明らかに充足可能なので, T から $heavier(hammer, feather)$ が帰結することが分かる⁸.

CIGOL は例から理論を構築する三つのタイプの演算子を使っている. 「V」演算子は新しい述語を導入せずにすでに存在しているクローズを一般化する. 「W」演算子は新しい補助的な述語を導入する. 短縮演算子は Plotkin の最小一般化の関係 (least general generalization relation) を適用することで単位クローズを生み出す, 「W」演算子の境界事例である. ここでは「V」演算子を取り上げる.

2.1.1 「V」演算子

Figure 3 はいくつかの「V」が繋がったものとみることができる. Figure 4 はこれらの「V」の一般的な形式を示している. RP は「V」の腕の先にある二つのクローズから, 「V」の下にあるクローズを導出する規則である. ここで説明されるメカニズムは逆に腕の一方にあるクローズと底にあるクローズから, もう一方の腕にあるクローズを導出するものである. Figure 4 では RP をかけられる C_1 のリテラルは正 (+), C_2 のリテラルは負 (-) である.

⁸Figure 3 参照.

C_1, C_2 を含むどんな理論においても C は余分である．というのもそれは C_1, C_2 から直接的に含意されるからである． C は廃棄することができるという事実より，「V」演算子は理論の単純化と簡略化を可能にする．

Figure 4 の C_1 と C から C_2 を構成する操作を吸収といい， C_2 と C から C_1 を構成する操作を同定という．この二つの操作をあわせて「V」演算という．フォーマルな定義は後に出される．以下では吸収の過程を説明する．吸収の操作によって C_1 と C から導出されるクローズ C_2 を，レゾリューションの商といい， $C_2 = C/C_1$ と書くことにする．

二つのクローズから RP によって導出されるクローズの数は有限であり，特にレゾリューションを受けるリテラルが決まれば，そのとき導出されるクローズは一つに決まる．しかしながら二つの与えられたクローズ C_1, C に対して， C_1 と RP によって C を導出できるようなクローズは無限にある⁹．そこで吸収の操作は，いかにしてそこから非決定性を減らすかがポイントになる．そのために操作を受けるクローズにはいくつかの制約が課される．

二つのクローズが共通のリテラルを持たないとき，この二つのクローズは分離可能であるという．RP を適用される二つのクローズに関して，以下では常に分離可能性が前提されているものとする．基礎ホーン・クローズ (p. 5, 定義 1.11 参照) のケースでは，分離可能性を想定すれば，レゾリューションの商は一意的に決まる．一階のクローズでは一般的に商は一意的には定まらない．しかし等式 (1) について考察することで，私たちはこの商に対する正確な制約を見つけることができる．(1) に対する単純な代数的操作によって次の等式が得られる．

$$C_2 = (C - (C_1 - \{L_1\})\theta_1)\theta_2^{-1} \cup \{L_2\} \quad (2)$$

さて $\theta_1\theta_2$ は $\overline{L_1}$ と L_2 の最汎統合子だから， $\overline{L_1}\theta_1 = L_2\theta_2$ である．したがって，

$$L_2 = \overline{L_1}\theta_1\theta_2^{-1} \quad (3)$$

(3) を (2) に代入することによって，

$$\begin{aligned} C_2 &= (C - (C_1 - \{L_1\})\theta_1)\theta_2^{-1} \cup \{\overline{L_1}\}\theta_1\theta_2^{-1} \\ &= ((C - (C_1 - \{L_1\})\theta_1) \cup \{\overline{L_1}\}\theta_1)\theta_2^{-1} \end{aligned} \quad (4)$$

が得られる． C と C_1 が与えられたとき，等式 (4) において非決定性の源は三つある． $L_1, \theta_1, \theta_2^{-1}$ である．この非決定性を減らすために C_1 がシングルトンであると仮定する．つまり $C_1 = \{L_1\}$ とする．これによって (4) は $C_2 = (C \cup \{\overline{L_1}\}\theta_1)\theta_2^{-1}$ に単純化され，非決定性は θ_1 と θ_2^{-1} の選択に減らされる．

θ_2^{-1} を決定的でなく構成することを考えてみよう．逆置換の定義により，私たちは $(C \cup \{\overline{L_1}\}\theta_1)$ の中のどの項が異なる変項に写像されるかを決めなければならない． C に現れる項 t と， $\{\overline{L_1}\}\theta_1$

⁹次のような状況と類比的に考えればよい．集合 A, B が与えられたとき $A \cap B$ は一意的に決まる．しかし A, C が与えられたとき， $A \cap B = C$ となるような B は無限に存在し，一意的には決まらない．

に現れる $t_1\theta_1$ は, θ_2^{-1} に現れるのと同じの変項 v に写像されるとする. ただしここで t_1 は $\{\overline{L_1}\}$ の項とする. このことは $t_1\theta_1 = t$ を含意する. 明らかにこの場合 $t_1 -_\theta t$ は定義されており, θ_1 の部分集合である. これは C と $C_1 = \{L_1\}$ が与えられたときに C_2 を仮説として立てる, 決定的でないアルゴリズムを示唆している (Algorithm 1). Algorithm 1 において, 集合 S の分割 Π というのは共通部分をもたない集合の集合であり, 要素になっている集合 (ブロックという) の和が S に等しくなるようなものを意味している. つまり例えば $S = \{a, b, c\}$ に対して, $\{\{a\}, \{b, c\}\}$ は S の一つの分割である.

Algorithm 1: 吸収を計算する非決定的アルゴリズム

Input: クローズ C と $C_1 = \{L_1\}$

$TP = \{(t, p) | t \text{ は } (C \cup \{\overline{L_1}\}) \text{ の場所 } p \text{ にある項}\}$ であるとせよ.

TP' は TP のある部分集合であるとせよ.

TP' の分割 Π を以下のように構成せよ.

Π のそれぞれのブロック $B = \{(r, p_1), \dots, (r, p_n)\} \cup \{(s, q_1), \dots, (s, q_m)\}$ は次のようなものであるとせよ.

s は r を包摂し,

すべての (r, p_i) は C の項に対応し,

(s, p_j) は $\{L_1\}$ の項に対応する.

上のすべてのブロック B に対して $\theta_1 = \bigcup (s -_\theta r)$ とせよ.

$\theta_2^{-1} = \{(r, \{p_1, \dots, p_n, q_1, \dots, q_m\}) / v | \text{上のすべてのブロック } B \text{ に対して}\}$ とせよ.

ただし v はすべて異なる変項で, $(C \cup \{\overline{L_1}\})$ にはないものとする.

Output: $C_2 = (C \cup \{\overline{L_1}\})\theta_1\theta_2^{-1}$

Algorithm 1 は非決定的なので, これを実行するには搜索ベースのアルゴリズムとして定式化しなおさなければならない (省略). これは単純性と簡略性をヒューリスティックとして用いる.

例 2.8. $C = \{x < s(s(x))\}$ とし, $C_1 = \{y < s(y)\}$ とする. ただし $s(z)$ は z の後続者を意味する. このとき

$$(C \cup \{\overline{L_1}\}) = \{x < s(s(x)), \neg(y < s(y))\},$$

$$TP = \{(x, \langle 1, 1 \rangle), (x, \langle 1, 2, 1, 1 \rangle), (s(x), \langle 1, 2, 1 \rangle),$$

$$(s(s(x)), \langle 1, 2 \rangle), (y, \langle 2, 1 \rangle), (y, \langle 2, 2, 1 \rangle), (s(y), \langle 2, 2 \rangle)\}.$$

TP' は非決定的に $\{(s(s(x)), \langle 1, 2 \rangle), (s(y), \langle 2, 2 \rangle)\}$ とする. Π は自明な分割 $\{\{(s(s(x)), \langle 2, 2 \rangle)\}\}$ とする. このとき,

$$\theta_1 = \{y/s(x)\},$$

$$\theta_2^{-1} = \{(s(s(x)), \{\langle 1, 2 \rangle, \langle 2, 2 \rangle\}) / z\},$$

$$C_2 = \{(x < z), \neg(s(x) < z)\}.$$

これは確かにレゾリューション関係 $C = C_1 \bullet C_2$ を満たす.

参考文献

- Gabbay, Dov M., Hogger, C. J. and Robinson, J. A. (eds.): *Handbook of Logic in Artificial Intelligence and Logic Programming. Vol. 1. Logical Foundations*, Clarendon Press, 1993.
- Mendelson, E: *Introduction to Mathematical Logic*, fourth edition, Chapman & Hall, 1997.
- Muggleton, S: Inductive logic programming. Muggleton (ed.) *Inductive Logic Programming*, Academic Press, 1992, pp. 3-27.
- Muggleton, S and Buntine, W.: Machine invention of first-order predicates by inverting resolution. Muggleton (ed.) *Inductive Logic Programming*, Academic Press, 1992, pp. 261-280.
- Plotkin, G. D.: A note on inductive generalization. Bernard Meltzer and Donald Michie (eds.), *Machine Intelligence 5*, Edinburgh University Press, 1969, pp. 153-163.
- Robinson, J. A.: A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery*, **12**, no. 1 (January, 1965), pp. 23-41.