博士（工 学） 林 雄二

学 位 論 文 題 名

ソフトウェアの形式的洗練化と検証に関する基礎的研究

学位論文内容の要旨

一般にアルゴリズムは、データ操作と実行順序の制御の両面を含んでいる。プログラム内のデータ操作の部分と、動的な振舞いの部分を区別して表現する方法は、問題の分析，設計段階に有効である。筆者の提案するIO正則式は、構成の基本単位が、評価順序に意味を持たないデータ間の静的な関係式（論理式）であり、正則表現によって、評価順序が意味を持つ動的な振舞いの側面を表現している。これはジャクソン法におけるジャクソン構造図に対応する式でもある。本論文では、IO正則式に基づいたソフトウウェアの形式的洗練化方法および検証方法を提案する。

論文は全体が８章から構成されており、第１章では本研究への動機と研究内容の概要を述べる。

第２章ではプログラミングやソフトウエア開発方法論の現状と課題を考察し、提案するIO正則式の意義を述べる。

第３章では、仕様記述言語Ｚと洗練化を紹介し、Ｚが潜在的にもつ問題点を考察する。近年、形式的技法が研究され、なかでも形式的に仕様記述を行うための言語Ｚ，VDM，Ｂなどが脚光を浴びている。特に、Ｚは現実的な問題に適用され、その効果が評価されている。Ｚはスキーマによってプログラムの静的な論理関係部分を表現すると共に、操作の事前状態，事後状態を表現することによって状態の変化を記述する。このようなスキーマ表現によって、問題が形式的に表されると、形式的記述が仕様を満たしていることの検証（Verification）、仕様に過不足や矛盾が無いかの妥当性検査(Validation)を効果的に行うことができる。形式的仕様記述を利用している事例の多くは、この検証と妥当性検査の部分で効果を上げている。Ｚのような形式的仕様記述には、システムの動的性を記述する部分が不足している。複数のスキーマで表された仕様に対し、スキーマの実行順序関係は直接的には表現されていない。実行順序を表すには、スキーマが持つ変数に状態を保持させて、個々の状態において、どのスキーマが実行されるかを表現する必要がある。Ｚのスキーマを基本項とするIO正則式によって仕様を表現しておくことは、スキーマの実行順序、反復、選択などの構造が表現されるので、この問題に対する一つの解決法になる。特に、環境との相互作用のある問題に対しては、仕様に環境からの入力や環境への出力の実行順序を盛り込んでおく必要がある。

第４章では、IO正則式を用いたプログラムの洗練化手法を提案する。Ｚなどの形式的仕様を、洗練化によって段階的にプログラムへ変換しようという研究も進められている。これは、洗練化計算(Refinement Calculus)として知られる研究分野である。現在の洗練化計算は、Ｚのような形式的仕様から制御構造を見いだし、最後はプログラム（あるいは、Dijkstraの護衛コマンドプログラム）に変換しようというものである。洗練化計算の表現は、論理式で表された仕様とプログラムの制御文（while文，if文など）とを組合わせた表現（仕様文）を用いた記述法を基

にしている。これは、洗練化においては、仕様段階とは異なった表現法が必要になることを意味している。ソフトウェア開発において、多くの方法論が実践化されずに終わるのは、それら方法論の記述方法が、開発のそれぞれの段階で異なっている点にも原因があるとの指摘もされている。例えば、構造化分析における仕様記述ツール（データフローダイアグラム）などは、構造化設計の段階には全く利用されず、代わりにモジュール構造図などの異なるツールが必要とされる。IO正則式で表現することは、仕様から設計まで、統一した記述方法を用いて開発を進めることを意味する。そのことで、各段階で単一のスキーマを変換するのみならず、スキーマの合成に対しての変換を進めることが可能である。また、従来の洗練化の過程では、プログラムに必要な制御条件（while文の判定条件など）を順次開発していく必要があるのに対し、IO正則式では、それら条件を切り出すことは最終のプログラミング段階にまかせることになる。このことはちょうど、ジャクソン法が、JSP構造図の段階には、プログラムの制御文の条件式に腐心する必要がないことに符号する。従来の洗練化手法では、仕様からプログラムまでを洗練化によって導出することをねらいとしているが、必ずしも最終段階まで洗練化することが適当とはいえない。ある段階でプログラミングに移行し、開発されたプログラムを検証する方法がむしろ現実的である場合が多い。

第５章で、IO正則式の仕様に対し、Hoare論理の下にプログラムの正当性検証を行う方法を提案する。Ｚにおいては、1つのプログラムに複数のスキーマを羅列したものが対応していることが、プログラムの正当性検証を困難なものにしている。IO正則式では、プログラムの正当性を元のIO正則式の仕様に照らして検証することも可能になる。それはIO正則式がプログラムの構造に対応するからである。IO正則式を用いた洗練化手法とこの検証法を組合わせて、洗練化とプログラミングを両方向から進める開発が可能となる。

第６章では、IO正則式の適用分野としてのデータフローネットワークを紹介し、データフローネットワークの意味を考察する。データフローネットワークは複数のプロセスをチャネルで結んだ計算モデルである。個々のプロセスは入力データストリームを出力データストリームに変換する機構である。プロセスの機能を外部から見たとき、それは、ストリームからストリームへの関数とみなすことができる。データフローネットワークという計算モデルにおいて基礎になっている性質は不動点との関係である。すなわち、プロセスを関数とみた場合に、ネットワークに生ずるストリームは関数の不動点であるという性質である。具体的に、プログラム実行の下で、不動点がどのような性質に該当するか、すなわち、有限で計算が終了する場合、ストリームが通信チャネルに残って終了した場合、などに対する不動点の意味を考察した。

第７章では、IO正則式表現の下でのデータフローネットワークの検証を提案している。データフローネットワークにおいて、個々の入力データにプロセスがどう反応するかを盛り込む立場では、順次受け取る入力データに伴ってどのように状態を変え、どのような出力を送出するかを表現しなければならない。そのような表現として、IO正則式は効果的である。データフローネットワークの個々のプロセスをIO正則式で表現することにより、ネットワーク全体の性質の検証をネットワーク内を流れるストリームの不動点の性質を利用して行うことが可能となり、筆者はそのための手法を開発した。

最後の第８章は、まとめとして研究成果全体を評価している。

学位論文審査の要旨

主　査　教　授　宮　本　衛　市
副　査　教　授　嘉　数　侑　昇
副　査　教　授　大　内　　東
副　査　教　授　和　田　充　雄

学位論文題名

ソフトウェアの形式的洗練化と検証に関する基礎的研究

一般にアルゴリズムは，データ操作と実行制御の両面を含んでいる。プログラム内のデータ操作の部分と，動的な振舞いの部分を区別して表現する方法は，問題の分析,設計段階に有効である。著者の提案するIO正則式は，構成の基本単位が評価順序に意味を持たないデータ間の静的な関係式(論理式)であり，正則表現によって評価順序が意味を持つ動的な振舞いの側面を表現している。論文では，IO正則式に基づいたソフトウェアの形式的洗練化方法および検証方法を提案している。

論文は全体が８章から構成されており，第１章では，本研究への動機と研究内容の概要を述べている。第２章では，プログラミングやソフトウエア開発方法論の現状と課題を考察し，提案するIO正則式の意義を述べている。第３章では，仕様記述言語Ｚと洗練化を紹介し，Ｚが潜在的にもつ問題点を考察している。

第４章では，IO正則式を用いたプログラムの洗練化手法を提案している。現在一般に知られている洗練化計算(Refinement Calculus)は，Ｚのような形式的仕様から制御構造を見い出し，最後はプログラムに変換しようというものである。洗練化計算の表現は，論理式で表された仕様とプログラムの制御文（while文,if文など）とを組合わせた記述法を基にしている。これは，洗練化においては，仕様段階とは異なった表現法が必要になることを意味している。これに対し，著者の提案する洗練化方法の特長は以下のとおりである。

・IO正則式で表現することによって，仕様から設計まで，統一した記述方法を用いて開発を進められる。

・各段階で単一のスキーマを変換するのみならず，スキーマの合成に対しての変換を進めることが可能である。

第５章では，IO正則式の仕様に対し，Hoare論理の下でプログラムの正当性検証を行う方法を提案している。Ｚにおいては，1つのプログラムに複数のスキーマを羅列したものが対応しており，これがプログラムの正当性検証を困難なものにしているのに対し，ＩＯ正則式による仕様表現はこの検証に有効な手段を与えており，以下のような特長を持つ。

・IO正則式がプログラムの構造に対応するので，プログラムの正当性を元のIO正則式の各構造に照らして検証することが可能になる。すなわち，多くの場合，部分的な検証を行うことで全体を証明することになる。
・局部的なプログラム構造はＩＯ正則式の仕様として与えられているので，多くの場合，ループ不変式に腐心する必要がない。
・IO正則式を用いた洗練化手法とこの検証法を組合わせて，洗練化とプログラミングを両方向から進める開発が可能となる。

第６章では，IO正則式の適用分野としてのデータフローネットワークを紹介し，データフローネットワークの意味を考察している。データフローネットワークという計算モデルにおいて基礎になっている性質は不動点との関係である。すなわち，プロセスを関数とみた場合に，ネットワークに生ずるストリームは関数の不動点であるという性質である。著者は，プログラム実行の下で不動点がどのような性質に該当するか，すなわち，有限で計算が終了する場合，またはストリームが通信チャネルに残って終了した場合などに対する不動点の意味を明らかにしている。

第７章では，IO正則式表現の下でのデータフローネットワークの検証を提案している。データフローネットワークにおいて，個々の入力データにプロセスがどう反応するかを盛り込む立場では，順次受け取る入力データに伴ってどのように状態を変え，どのような出力を送出するかを表現しなければならない。そのための表現としてＩＯ正則式が有効であることを述べ，さらに，データフローネットワークのプロセスと，仕様であるチャネル上のストリームとをＩＯ正則式で表現し，データフローネットワークが仕様を満たすことを証明する方法を提案している。この方法は非同期の分散アルゴリズムなどに適用が可能であり，著者は最大値発見問題などへの適用例を示している。

これを要するに，著者は，ＩＯ正則式がプログラムにおける計算過程を表現する有効なモデルであることを示し，それに基づいた洗練化，検証に関して新知見を得たものであり，ソフトウェア工学の発展に対して貢献するところ大なるものがある。

よって著者は，北海道大学博士（工学）の学位を授与される資格あるものと認める。