論文の内容の要旨

論文題目　Abstraction and Search in Verification by State Exploration
状態探査による検証における抽象化と探索

氏　名　高橋 孝一

ネットワークのインフラストラクチャを含むセイフティ・クリティカルなシステムが増えるに従って，ソフトウェア検証の重要性は今後ますます大きくなって行くと考えられる．状態の自動的な探索を基礎とするモデル検査法は，検証技術の一つとして，ハードウェア検証の分野で広く利用されており，ソフトウェアの検証に対してもその利用が期待されている．しかし，ソフトウェアはハードウェアに比べて状態空間が巨大であり，ソフトウェア検証に対してモデル検査法を直接には適用できない場合が多い．従って，ソフトウェアのモデル検査を成功させるためには，状態空間を小さくする技術が不可欠である．状態空間を小さくする最も重要な方法の一つとして抽象化があり，ハードウェアのモデル検査に対しては成功している．本論文は，ソフトウェア検証を行うために，抽象化の技術を付加したモデル検査法，すなわち，抽象モデル検査法について探求することを目標としている．

本論文の第３章では，モデル検査法における抽象化の有効性を調べるために，一つのケーススタディとして，並行ごみ集めのアルゴリズムの検証を行う．並行ごみ集めアルゴリズムの正しさは簡単に証明できるものではなく，抽象モデル検査のような自動検証が有効な手段となる．このケーススタディを通じて，抽象モデル検査における二つの重要な課題が明らかになる．一つは，抽象化によって元のアルゴリズムの性質が失われないことを保証する理論的基礎が必要なことである．もう一つは，問題に応じて適切な抽象化をどのように構成するかという課題である．

第４章では，抽象モデル検査の理論的基礎を与える．プログラム意味論の分野で研究されていた詳細化の理論を適用することによって，モデル検査と抽象化の関係を理論的に解明する。

並行ごみ集めアルゴリズムを検証するケーススタディでは，ヒープの状態を抽象化する必要がある．ヒープは複雑なリンク構造を成し抽象化の可能性も無限にあるため，ヒープの抽象化をどのように定義すべきかは自明ではない．第５章では，リンク構造を抽象化するための一般的な技術を開発する．正則表現を用いることにより，並行ごみ集めアルゴリズムの抽象モデル検査に用いた抽象化を一般化する．この技術は，第３章で与えた二番目の課題の一つの答えになっている．

モデル検査法は，与えられたソフトウェアが仕様を満たしているかどうかを自動検証するだけでなく，与えられた仕様を満たすソフトウェアを自動合成する問題にも適用可能なことが期待される。第６章では，アルゴリズムを自動合成することを目標に，モデル検査を用いてアルゴリズムの探索を行う方法を提案する．そのケーススタディとして，並行ごみ集めのアルゴリズムと排他制御のアルゴリズムの発見を試みる．第７章では，アルゴリズムの探索の効率を上げるために，シンボリックなモデル検査法をアルゴリズムの集合に適用する．この方法を用いることにより，モデル検査の一回の実行でアルゴリズムの集合を探索することが可能である．さらに探索空間を小さくするため，この方法に抽象化技術を適用する．すなわち，シンボリックな探索と抽象化を組み合わせる．この方法のケーススタディとして，相互排除のアルゴリズムの探索を行う．

最後に、抽象モデル検査法の全プロセスの正しさを形式的に証明することを目指す研究について述べる．特に，抽象モデル検査で用いられる抽象化は，抽象モデル検査全体の正しさを保証するために，いくつかの条件を満たしていな

ければならない．理想的には，これらの条件は形式的に証明されるべきである．第８章では，並行ごみ集めの抽象モデル検査で用いられた抽象化の正しさの形式的証明を与える。形式的証明は，HOL と呼ばれる証明支援系の上で開発された．形式的証明と関連して，第９章では，証明支援系のための環境に関する研究についても述べる．まず，証明支援系のユーザ・インタフェースのためのガイドライトして，“proving as editing paradigm”と呼ぶ考えを提案する．このガイドラインは，第８章の形式的証明に際して実際に活用されたものである．また，プログラム意味論の分野で頻出する可換図をそのまま扱うことのできるグラフィカル・ユーザ・インタフェースについても述べる．