# 文字列探索に適したAVL木の拡張とその評価に関する研究

別記様式第３号－１

# 学位論文の要旨

| 氏　　名 | 竹之下　朗 |
|---|---|
| 学位論文題目 | 文字列探索に適したAVL木の拡張とその評価に関する研究 |

本論文では，木構造の探索操作における効率性の向上を目的として5分木に拡張したAVL木を提案し、これに対して様々な考察や評価を行った。拡張したAVL木のデータ構造は、平衡性を部分的に満たす5分木構造であり、データ探索において語の文字列を1文字ずつ比較し、前方が一致する文字列を部分木とする特徴を持っている。数値実験により、拡張したAVL木の構築時間は既存のAVL木の約70%、B木の約50%であり、探索の効率性が向上されることが確認できた。更にデータ構造の領域量の削減と奇数分木への一般化について考察した。

第１章「研究の背景とアルゴリズムの評価法」では、文字列探索に関する研究の背景について述べ、アルゴリズムと計算量についての概念、アルゴリズムの様々な評価法を説明している。

第２章「基本的なデータ構造と木構造」では、計算機上でのデータ構造と木構造について説明し、それらの詳細な構造や関連するアルゴリズムは主にC言語を使用して記述している。基本的なデータ構造として、リスト・配列・スタック・キューを取り上げ、それぞれのデータ構造の特徴と長所・短所について述べる。さらに木構造の性質や定義について説明し、代表的な木構造である2分探索木、AVL木、B木、トライ、パトリシアの基本的な構造やC言語での実装について詳述している。

第３章「5分木に拡張したAVL木の提案」では、既存のAVL木を文字列探索に適応させたデータ構造を提案する。5分木に拡張したAVL木は、2分木のAVL木を拡張したもので、リスト構造で平衡性を一部満たす木構造である。データ構造は、文字列を配列で格納し動的に挿入・削除できるようにリスト構造を採用している。拡張したAVL木では、アルゴ

リズムは基数探索法を用いて文字列を1節点につき最大2文字を比較し、比較回数を抑えるために木を平衡化している。この拡張したAVL木に対し、その定義、時間計算量、各操作などについて考察する。また、構造を満たすためにラベルというデータではない識別用の節点を導入している。このラベルの数が増加すると探索時間や領域量が増加すると考えられるが、文字が10進数以上であればラベルの割合は1%以下であり、実際の環境では、ほとんど影響がないことが分かった。この他に基数探索法の特徴から前部分が一致する文字列を探索できること等を述べている。

第４章「拡張したAVL木の評価」では、拡張したAVL木、既存の各木をC言語で作成し、重要な項目について数値比較を行っている。比較には、2分探索木、AVL木、5分木のB木を対象とした。2分探索木、AVL木との比較項目は、(1)構築時間、(2)領域量、(3)深さ、(4)比較回数、(5)平衡回数、(6)ラベル数である。5分木に拡張したAVL木は、2分木から多分木へと拡張したことにより木の深さ(3)が浅くなることから、(1)(4)(5)の値が小さくなり、その結果として探索時間が短くなる長所をもつ。5分木に拡張したAVL木の短所は(2)領域量であり、文字が10進数で文字列の長さが100桁の場合では、2分木のAVL木の約1.2倍である。5分木のB木との比較では、B木の高さが最も小さくなる場合を仮定するとデータ数が$10^{14}$個以上であり、5分木に拡張したAVL木はB木よりも領域量は理論的に小さくなることが分かった。文字が10進数で文字列の長さが100桁のデータ数10,000,000個のランダムなデータに対する数値実験では、構築時間は約47%、比較回数は約11%でいう良好な結果が得られた。この場合、5分木に拡張したAVL木の領域量はB木の約36%になった。

第５章「5分木に拡張したAVL木の領域量の削減と一般化」では、第４章の数値実験で明らかになった領域量の問題と奇数分木へのAVL木への拡張について考察している。本論文で提案したAVL木の短所としては領域量の増加であったが、これを削減する方法を提案している。領域量の削減には、ポインタの数を減らすことと節点の要素になる文字列の削減で対応した。ポインタ数の削減については、節点がラベルになる場合、データ用の配列は構造を維持する2文字だけ必要であり、データが格納されている節点では中央部分木へのポインタが不要である。一方、文字列の削減については節点に文字列全体ではなく、

一致しない後方の文字列のみを保持させることで削減を行った。これにより、従来の5分木構造のAVL木より10進数50桁の場合で、67.74%の削減が可能であることが判明し、既存のAVL木の約90%、B木（5分木）の約40%の領域量で構築可能であるという結果を確認している。また、ある節点において最大$k$桁の比較を行うと、$2k$+1分木として一般化できることを示した。

第６章「まとめと今後の課題」では、前章までのまとめと今後の課題について述べている。

別記様式第13号

# 論 文 審 査 の 要 旨

| 報告番号 | 理工研 第348号 | 氏名 | 竹之下 朗 |
|---|---|---|---|
| 審査委員 | 主査 | 新森 修一 | |
| | 副査 | 古澤 仁 | 青木 敏 |
| | | | |

学位論文題目　文字列探索に適したAVL木の拡張とその評価に関する研究
(Studies on Proposal and Evaluation of Extended AVL Tree Suitable for Character String Search)

審査要旨

提出された学位論文及び論文目録等を基に学位論文審査を実施した。本論文は、文字列探索に適したAVL木の拡張とその評価に関する研究について述べたものであり、全文6章から構成されている。

第1章を序論とし、文字列探索に関する研究の背景について述べ、アルゴリズムと計算量についての概念、アルゴリズムの様々な評価法を述べている。第2章では、木構造の性質や定義について説明し、代表的な木構造である2分探索木、AVL木、B木、トライ、パトリシアの基本的な構造や特徴、並びにC言語での実装について詳述している。

第3章では、既存のAVL木を5分木に拡張したAVL木を新たに提案し、文字列探索に適したデータ構造であること示している。拡張したAVL木は、与えられた文字列を1節点につき最大2文字ずつ比較し、比較回数を抑えるために木の平衡化を行っている。また、拡張したAVL木の定義、時間計算量、各種基本操作、構造を維持するための識別用の節点（ラベル）の導入と影響などについて考察している。

第4章では、数値実験により拡張したAVL木の評価を行っている。既存のAVL木、5分木のB木を主な比較対象とし、構築（挿入）時間、比較回数、領域量などについて詳細な検証を行っている。10進数10～100桁の一千万個のランダムなデータに対して、構築時間は既存のAVL木の65％程度、B木の50％程度であり、構築時間は探索時間に大きく依存することから、探索の効率性の向上を確認している。一方、領域量については、B木の約35％と良好な結果が得られたが、既存のAVL木の約1.2倍の領域量が必要であると言う問題点を指摘している。

第5章では、前章の数値実験で明らかになった領域量の問題点の改善と奇数分木へ一般化について考察している。領域量の削減には、拡張したAVL木の性質を活かした方法を提案しており、ポインタの数を減らすことと節点の要素になる文字列の削減で対応している。ポインタ数の削減は、節点の種類に応じて最低限確保すること、節点がラベルになる場合、データ用の配列は構造を維持する2文字だけ必要であることを利用し、また、文字列の削減は節点に文字列全体ではなく、一致しない後方の文字列のみを保持させることで削減を行っている。これにより、第3章で提案したAVL木より10進数50桁の場合で、約68%の削減が可能であることが示された。また、ある節点において最大$k$桁の比較を行うと、$2k$+1分木として一般化できることを示し、基本的操作の方法を考察している。第6章では、本研究で得られた結論を示し、今後の研究の発展性について提起している。

以上、本論文は、文字列探索に適したAVL木の拡張とその評価に関する研究であり、5分木に拡張したAVL木を新たに提案し、基本的操作アルゴリズムの提起、既存のAVL木やB木など他の木構造との詳細な比較・検証を行い、時間計算量や領域計算量の観点からの優位性を明らかにしている。これは、大規模な文字列データを扱うデータベースシステムなどへの貢献が期待される。よって、審査委員会は博士（工学）の学位論文として合格と判定する。

別記様式第14号

# 最終試験結果の要旨

| 報告番号 | 理工研　第348号 | 氏　名 | 竹之下　朗 |
|---|---|---|---|
| 審査委員 | 主　査 | 新森　修一 | |
| | 副　査 | 古澤　仁 | 青木　敏 |
| | | | |

主査及び副査2名で構成される審査委員会は、平成23年1月31日に学位申請者「竹之下　朗」に対して、論文の内容について説明を求めた。これに引き続き、参加者を含めて質疑応答を行うとともに、関連事項について諮問を以下のように行った結果、いずれに対しても満足すべき回答が得られた。主な質疑応答は、以下の通りであった。

**質問1**：領域量の削減率について述べているが、削減率の定義と既存のAVL木との比較結果は？

**回答1**：削減率とは、本論文で提案している木構造の領域量を比較対象としている木構造の領域量で除した値を1から引いたものである。すなわち、比較対象としてるデータ構造の領域量に対して、本論文の構造で削減可能となった領域量の比率である。数値実験により既存のAVL木と比較した領域量の増大が問題であったが、ポインタ数の削減、最低限必要な文字列のみ格納する方法を用いることで、最終的な削減率は約10～50％程度であることが確認された。

**質問2**：数値実験で主に既存のAVL木とB木を対象にしているが、その理由は？

**回答2**：既存のAVL木の構造は、本論文で拡張する際の基本とした構造であることから比較対象にしている。また、B木については、広く利用されている多分木構造であり、本論文で提案した構造が5分木構造になっているので、5分木のB木を比較対象とした。

**質問3**：本論文の手法を一般のパソコン上で利用することが可能か、また効果が出るのか？

**回答3**：本論文で提案したデータ構造はC言語で実装を行ったが、一般のパソコンで利用することは可能である。また、かなり大規模な文字列データに対してでないと実感できるほどの効果はないかも知れないが、数文字程度で数万個程度以上の文字列データの探索であれば効果が期待できる。なお、数値実験では、データ構造全体の構築時間やあるデータを探索するのに要する時間とも、B木の50％程度で実行できることが確認されている。

**質問4**：データベースシステムやWeb上での情報検索に応用が期待できるか？

**回答4**：本質的には応用可能であると思われるが、本論文ではメモリ（主記憶）上に格納された文字列データに対しての探索を研究対象にしているので、ハードディスクなどの外部記憶装置に格納されたデータの探索への応用方法の確立は今後の課題と考えている。

**質問5**：提案するデータ構造の$2k$ +1分木への一般化について述べているが、5分木から$2k$ +1分木へ自然な形での一般化が可能か？

**回答5**：本論文の中で、5分木に拡張したAVL木の定義を示しているが、この定義は3分木、5分木、7分木といった奇数分木への再定義が可能な構造になっているので、$2k$+1分木への一般化は可能である。また、一節点で何文字比較するのかが$k$の値になる。この値が小さければ構造はシンプルになるが、木の高さが高くなり探索に要する時間が増大する傾向があると考えられる。

**質問6**：本論文の研究の発展性、今後の課題は？

**回答6**：拡張したAVL木の一般化と理論的性質の考察、領域量の削減方法の具体的な実装化とその効果の確認、大規模な日本語の文字列データに対する数値実験による検証、ダブル配列などの他のデータ構造との比較評価などが考えられる。

など、約10個の質問に対して的確に答えた。

以上の結果を受け、上記審査委員会は全員一致で、学位申請者は大学院博士後期課程の修了者としての学力ならびに見識を十分有するものと判断し、博士（工学）の学位を与えるに足る資格を持つと認めた。