



ストリーム処理への自動変換による 効率的な木構造処理

佐藤 亮介 東北大学大学院情報科学研究科/日本学術振興会特別研究員

[受賞論文]

Ordered Types for Stream Processing of Tree-Structured Data
Ryosuke Sato, Kohei Suenaga, Naoki Kobayashi (Tohoku Univ.)
Journal of Information Processing, Vol.19, pp.74-87 (2011)

このたび、標記の論文で本会論文賞という過大な賞をいただくこととなり、大変光栄に思う。本論文の共著者である小林、末永両氏、また、大変有益なご指摘をいただいた査読者の方々に心より感謝する。

本論文は、XML データなどの木構造データを、シンプルなプログラムで効率的に処理できるようにすることを目指したものである。コンピュータの発達に伴い、XML データのサイズは増加の一途をたどっており、複雑な処理を効率よく行える XML 処理手法が必要となっている。XML などの木構造データの処理手法として、木構造処理とストリーム処理の2つの処理方式が挙げられる。木構造処理では入力データをすべて読み込み、入力木をメモリ上に構築する。一方、ストリーム処理では、入力データをトークンの列として受け取り、逐次的に処理を進めていく。そのため、木構造処理は記述が容易だがメモリ効率が悪く、逆にストリーム処理はメモリ効率は良いが複雑になるという特徴がある。

本研究の基になった先行研究では、木構造処理およびストリーム処理の両方の利点を活かすため、木構造処理プログラムをストリーム処理プログラムに変換するというアプローチをとっている。与えられたプログラムが入力木の各ノードを左から右の深さ優先でちょうど一度ずつアクセスするのであれば、ストリーム処理への変換が容易にできる。順序付き線形型によって上記の性質が成り立つことを保証できるため、先行研究では、順序付き線形型を用いた型システムを用いて、入力プログラムがこの性質を満たすかどうかを検査し、性質を満たすプログラムをストリーム処理に変換していた。しかし、当然ながら、この性質を満たさないプログラムも多く

ある。そのようなプログラムを扱うため、この手法に対していくつかの拡張が行われてきた。拡張の1つとして、先の制約を満たさない部分にバッファリング処理を挿入するというものがある。これによって、どのような入力プログラムでもストリーム処理に変換ができるようになった。

本論文は、それらの拡張の一環であり、よりメモリ効率の良いストリーム処理プログラムへの変換を目指したものである。本研究では、順序付き線形型だけでなく、順序付き非線形型を考えることによって、無駄なバッファリングを減らしたより効率的なストリーム処理プログラムへの変換を実現している。順序付き非線形型を用いることで、この部分木はこの処理の中でしか使用されないといった解析が可能になる。この情報を用いることによってより柔軟なバッファリングが可能となり、結果としてよりメモリ効率の良いストリーム処理プログラムへの変換が可能となる。

本研究は、木構造処理プログラムの効率化という実践的な動機から生まれたものであるが、順序付き非線形論理という普段あまり見かけない論理体系を使っており、線形論理以外の部分構造論理の計算機科学への応用という観点からの面白さも味わっていただけだと思う。

本研究分野のさらなる発展を願うとともに、本賞の受賞を励みに、研究分野および社会の発展のため今後もより一層研究に尽力したい。本研究がその発展に少しでも寄与できれば幸いである。

(2012年4月16日受付)

佐藤 亮介 ryosuke@kb.ecei.tohoku.ac.jp
東北大学大学院情報科学研究科博士課程後期3年の課程在学中。日本学術振興会特別研究員。