

Progress in Informatics ABSTRACTS

No.10, March 2013

Contents

Special issue: Advanced Programming Techniques for Construction of Robust, General and Evolutionary Programs

Guest Editorial

- 1 ——— **Advanced programming techniques for construction of robust, generic and evolutionary programs**
 堅牢性と汎用性を備えた進化型プログラムを構築するための先進的プログラミング技術
 Zhenjiang HU, Shin-Cheng MU, Stephanie WEIRICH

Research Papers

- 3 ——— **Distributive laws of directed containers**
 有向コンテナの分配法則
 Danel AHMA, Tarmo UUSTALU
- コンテナを使用すれば、さまざまなクラスのデータ型の位置と形状をうまく表現することができる。我々は最近、ある1つの形状におけるそれぞれの位置が、もう1つの形状（平たく言えば、その位置に基づく副形状）を決定するような状況を一般的に説明するための特別なケースとして、有向コンテナを導入した。コンテナは充満忠実関手を介して集合の圏上の関手に変換されるが、有向コンテナは充満かつ忠実なコモナドを表す。実際のところ、有向コンテナはコモナド構造を運ぶコンテナそのものである。有向コンテナは、モノイドを一般化したもの（依存型バージョン）と見なすこともできる。この圏のコンテナは（集合関手の圏同様に）モノイド構造の組み合わせを運ぶが、一般に有向コンテナは（コモナドと同様）組み合わせを行わない。本論文では、2つの有向コンテナ間における分配法則の概念を構築するが、これは、2つのコモナド間における分配法則の概念に相当する。さらに、分配法則に基づく有向コンテナの構成構造について詳しく説明する。これは、2つのモノイドの Zappa-Szép 積を一般化する結果となる。

- 19 ——— **Equational reasoning about programs with general recursion and call-by-value semantics**

一般再帰および値呼び意味論を用いたプログラムに関する等式推論

Garrin KIMMELL, Aaron STUMP, Harley D. EADES, Peng FU, Tim SHEARD, Stephanie WEIRICH, Chris CASINGHINO, Vilhelm SJÖ BERG, Nathan COLLINS, Ki YUNG AHN

依存型プログラミング言語は、不変量を型としてコード化することによって、検証とプログラミングを統合化するメカニズムを備えている。従来、依存型言語は型理論に基づくもので、証明とプログラムの間の直接的な関係はカーリー・ハワード同型対応を基礎としていた。しかし、この対応関係では、論理的な安定性を確保するために言語を正規化する必要があるため、それなりの代償が強いられるものだった。Trellys は現在開発中の値呼びによる依存型プログラミング言語で、型理論と、一般再帰、Type: Type、および任意のデータ型を含む不安定なプログラム機能を統合化するように設計されている。本論文では、Trellys 用の1つのコア言語設計の概要を説明するとともに、発散の可能性があるプログラムについて安定した推論を促進する重要な言語構成要素の使用法を示す。

- 47 ——— **Wander types: A formalization of coinduction-recursion**

ワンダー型: 余帰納-再帰の定式化

Venanzio CAPRETTA

ワンダー型 (Wander Types) は、帰納-再帰定義の余帰納版である。ワンダー型は、特定の型のコンストラクタとその型自体に関する関数を同時に指定することによって定義される。この型のコンストラクタは関数コンポーネントを参照することができ、その関数自体はコンストラクタに関するパターンマッチングによって与えられる。ワンダー型は、帰納-再帰型とは2つの点で異なる。まず、要素の構造が整礎である必要はないので、コンストラクタを無限に適用することができる。また、関数の定義における再帰呼び出しを構造的に小さい引数に基づいて行う必要はない。ワンダー型は、いくつかの既知の型構成要素を一般化する。関数コンポーネントを使用すれば、データ分岐の方法を制御することができる。これは、適切な関数定義を通じて整礎性を整えることにより、余帰納と帰納の両方を実現する。ワンダー型の特別な例としては以下のようなものがある: 単純な帰納型と再帰型、帰納-再帰型、混合的な帰納-余帰納型、連続ストリームプロセッサ。

65 ——— **Modularising inductive families**

帰納的集合族のモジュール化

Hsiang-Shang KO, Jeremy GIBBONS

依存型プログラムを作成するプログラマーには、帰納的集合族 (inductive family) を使用して制約条件とデータ構造を一体化することが推奨される。制約条件は状況に応じてさまざまなものが使用されるので、データ構造が同じでも異なるデータ型が使われることがある。たとえば、シーケンスは要素の長さや順番によって制約されるので、シーケンスの基本的なデータ構造が同じでも、「ベクトル」や「並べ替えリスト」といった異なるデータ型が使われることになる。構造的には同様なこれらのデータ型に関する共通演算をモジュール方式で実装することは、長年にわたる課題だった。ここではデータ型に依存しないソリューションを提案する。このソリューションにおいて我々は、データ型とそのより洗練されたデータ型との間にある同型の族を、データ型の改良として公理化した。また、マクブライド (McBride) の言うところの「オーナメント」をこのような改良に変換できることを示す。オーナメントから導いたデータ型の改良により、演算の関連特性を制約条件ごとに個別に明らかにすることができる。さらに、プログラマーが基本的なデータ型に適用するいくつかの制約条件を選択し、それらの制約条件を取り入れて新しいデータ型を合成すれば、その演算を定期的にアップグレードして、合成したデータ型を使用することができる。

89 ——— **An expressive bidirectional transformation language for XQuery view update**

XQuery 画面更新用の表現力豊かな双方向変換言語

Dongxi LIU, Zhenjiang HU, 武市 正人

本論文では、表現力豊かな双方向 XML 変換言語について述べるとともに、その言語を使用し、実体化された XQuery 画面を通じて XML データを更新するという問題について検討する。この双方向言語の変換は、2つの方向で行うことができる。1つが順方向で、この場合は、それらの変換によって XML ソースから実体化された表示が作成される。これに対し逆方向では、変換により、表示上の更新を反映することでソースが更新される。この双方向言語によって XQuery の解釈を行う際には、順方向の実行で XML クエリを行い、逆方向の実行後に XML の更新を行う。ここでは、双方向変換における良いふるまいを特徴づけるために、拡張ラウンドトリップ性を提案する。この性質は、表現力豊かな双方向変換言語を実現するため、柔軟性をより高めるものである。表示された XML への挿入の更新に伴う問題を詳細な例によって分析し、表示への挿入が行われた場合の逆方向変換の補助として、新たにデータ型情報を使用する。この双方向言語用に、XML 用の再帰的正規表現型を含むシステムを考案した。データ型についてよく考慮されたプログラムでは、逆方向の実行後もソース型が維持される。我々の方法によるプロトタイプは、複数の XQuery ユースケースにおいて実装され、テストされている。

131 ——— **GRoundTram: An integrated framework for developing well-behaved bidirectional model transformations**

GRoundTram: ラウンドトリップ性 (well-behavedness) を満たす双方向モデル変換開発のための統合フレームワーク

日高 宗一郎, 胡 振江, 稲葉 一浩, 加藤 弘之, 中野 圭介

双方向モデル変換は、二つのモデルの間の一貫性の維持に有用であり、ソフトウェア開発におけるモデル間同期、ラウンドトリップエンジニアリング、ソフトウェア進化など多くの応用が期待されている。しかし、このように魅力的な用法があるにもかかわらず、体系的な開発を支援する実用的なツールがないため、広く用いられていないのが現状である。本稿では、この問題を解決すべく GRoundTram という統合環境を提案する。GRoundTram はラウンドトリップ性 (well-behavedness) を満たし効率的な双方向モデル変換の合成可能な開発のために注意深く設計され実装されている。GRoundTram は整礎な双方向変換の枠組の上に構築され、双方向モデル変換のコーディングのためのユーザフレンドリな言語や、モデルと変換両方の妥当性の確認 (validation) のための新たなツール、効率を改善するための最適化機構、そして双方向変換の振舞いを検査する強力なデバッグ環境を備えている。GRoundTram は我々以外の研究グループにも利用され、そこでの成果から GRoundTram の実用上の有用性が示されている。

Research Reports149 ——— **NII Shonan Meeting Report: Dependently Typed Programming**

NII 湘南会議報告: 依存型プログラミング

Shin-CHENG MU, Conor MCBRIDE, Stephanie WEIRICH

157 ——— **NII Shonan Meeting Report: Automated Techniques for Higher-Order Program Verification**

NII 湘南会議報告: 高階プログラム検証のための自動化技術

小林 直樹, Luke ONG, David Van HORN

Regular Paper

Research Paper

167 — An efficient exact algorithm for the Minimum Latency Problem

最小遅延時間問題のための効果的な厳密解アルゴリズム
Ha BANG BAN, Kien NGUYEN, Manh CUONG NGO,
Duc NGHIA NGUYEN

最小遅延時間問題 (Minimum Latency Problem : MLP) は、数多くの実用的用途を持つ組合せ最適化問題の 1 つである。一般的に、MLP は NP 困難であることが明らかになっている。この問題を解くための 1 つの方法が、厳密解アルゴリズムである。しかし、近年提案されているアルゴリズムは、節点数が 26 という小規模問題にのみ適用されるものである。本論文では、より大規模な MLP を解くための新しい厳密解アルゴリズムについて論じる。ここで論じるアルゴリズムは分枝限定法に基づくもので、効率化手法を改善する 2 つの新しい規則を採用している。我々は、いくつかのデータセットに関してアルゴリズムの評価を行った。その結果は、40 節点までの問題を正確に解けることを示している。

175 — Load-balanced routing with selective even traffic splitting

選択式均等分散における負荷分散方式
本間 奨, 角田 俊一, 大木 英司

本論文は、従来の non-split S-OSPF よりネットワーク輻輳率を低減させ、かつ、split S-OSPF より比較的容易に分散するために、even-split S-OSPF を提案する。Split S-OSPF では、隣接エッジノードにおいて最適な分散比を用いて転送する。しかし、非均等な割合で分割して転送を行うため、実装が複雑となる。一方、Non-split S-OSPF では、エッジノードにおいて隣接ノードを 1 つしか選択しないため、split S-OSPF より転送が単純であるものの、ネットワーク輻輳率が上昇するという欠点がある。提案する even-split S-OSPF は、隣接ノードにトラヒックを均等に分散することにより、split S-OSPF よりトラヒックの転送を単純化させるとともに、non-split S-OSPF よりネットワーク輻輳率を低減させる。このトラヒックを均等に転送させる隣接ノードを選ぶ最適化問題は、整数計画問題に定式化される。整数計画問題を実時間で解くには難しいため、実時間で解くための発見的な手法を開発した。この発見的な手法を用いて性能を評価した結果、even-split S-OSPF は、小規模ネットワークにおいて non-split よりネットワーク輻輳率を低減でき、かつ、大規模ネットワークにおいて split S-OSPF と同等の性能が得られた。