

## ②4 Realizability Interpretation of Coinductive Definitions and Program Synthesis with Streams

M. Tatsuta(東北大学, 日本)

### 発表要旨

本稿では, streamを扱うようなプログラムの性質を形式化できるような論理体系を定義し, realizability interpretationを利用してプログラムを合成する方法を確立することが目的である。

一般にstreamやそのextensional equalityのように最大浮動点によって定義される性質は, 述語のcoinductive definitionによって形式化することが可能である。ここでは, このような機能を備えた論理体系として, TID, を与え, これらの性質がこの論理体系の中でどのように表現されるかを示す。

また, プログラムを合成する一つの方法として, realizability interpretationがある。ここで使用したのはGraysonのq-realizabilityを拡張したものであるが, それによってcoinductiveに定義された述語についてもrealizerが定義できて, しかもそのrealizability interpretationは健全性定理を満たすことを証明する。

更に, これを使ってstreamを含むプログラムをどのように合成するかについて, その方法を具体的に説明する。

### 質疑応答

質問: streamが, それに対応するtransformationの最大不動点として意味づけられるということですが, この関係には高階論理の概念が含まれるのではないのでしょうか。なぜなら, 例えばプロセス代数において強等価関係はプロセス間のある関係を定義するもので, やはりあるapplicationの最大浮動点としての意味を持ち, これは高階論理の概念に拡張されます。

要するに, 私の質問は, この論理体系の基になっているのは1階の論理なのではないでしょうか。もしそうなら, それを高階の論理に拡張することは可能なのでしょうかということです。

回答: この論理体系では, 最大不動点による意味づけを形式化することができます。また, これはおっしゃるように1階の論理がもとになっています。しかし, これを高階の論理に拡張することは不可能です。なぜなら私は本研究の目的のために, 高階の論理は必要ないと考えます。というのは高階論理はとても強すぎるからです。そのために, 私は1階の論理と述語のcoinductive definition, それに一般のinductionを使ったのです。

質問: プログラムの仕様をある形の論理式に制限しているという点について, あなたはその制限が自然であると言っていますが, その点についてもう少し詳しく説明して頂けますか?

回答: この制限は, 定理(仕様を記述した論理式)を証明できるようにするために付けたものです。しかし, そのような制限を加えても, 例えば高山さんの発表に出てきたmapstream functionなどは表現可能です。ですから, この条件を加えても特に問題はないと考えたので, 自然であると言いました。