Artifact: A Computational Interpretation of Compact Closed Categories: Reversible Programming with Negative and Fractional Types

Chao-Hong Chen and Amr Sabry

January 20, 2021

This document contains a list of claims in the paper and corresponding code.

2 Core Reversible Language: Π

- The syntax of Π in Fig.1 is formalized in **Pi/Syntax.agda**.

2.1 Abstract Machine Semantics

- The δ function in Fig.2 is defined in **Pi/Opsem.agda:L47**.
- The well-formed continuation stacks in Fig.3 are defined in **Pi/Opsem.agda:L71**.
- The machine states in Def.1 are defined in **Pi/Opsem.agda:L81**.
- The reduction relation in Fig.4 is defined in **Pi/Opsem.agda:L86**.
- Lem.2 is proved in **Pi/NoRepeat.agda:L13**.
- Lem.3 is proved in **Pi/NoRepeat.agda:L29**.
- Def.4 is defined in **Pi/Eval.agda:L76**.
- Def.5 is defined in **Pi/Eval.agda:L121**.
- Thm.6 is proved in **Pi/Properties.agda:L31**.

2.2 Interpreter

- The interpreter in Fig.5 is defined in **Pi/Interp.agda:L9**.
- Thm.7 is proved in **Pi/Properties.agda:L49**.

3 Termination of Reversible Abstract Machines

- The reversible abstract machine in Def.8 is defined in **RevMachine.agda:L8**.
- Lem.9 is proved in **RevNoRepeat.agda:L125**.
- Lem.10 is proved in **Pi/Eval.agda:L13**.
- Thm.11 is proved in **Pi/Eval.agda:L76**.
- The partial reversible abstract machine in Def.12 is defined in **RevMachine.agda:L15**.
- Lem.13 is proved in **PartialRevNoRepeat.agda:L136**.

4 Space and Time Resources and Trade-Offs

- $\#\sigma$ is defined in **TimeSpace.agda:L71**.
- The examples at the end of this section are in **TimeSpace.agda:L80-87**.

5 Negative Types: Π^m

5.1 Abstract Machine Semantics

- The syntax of Π^m is formalized in **Pi-/Syntax.agda**.
- Def.14 is defined in **Pi-/Opsem.agda:L84**.
- The transition rules in Fig.6 are defined in **Pi-/Opsem.agda:L91**.

5.2 Properties

- Lem.15 is proved in **Pi-/NoRepeat.agda:L20**.
- Lem.16 is proved in **Pi-/NoRepeat.agda:L119**.
- Lem.17 is proved in **Pi-/Eval.agda:L23**.
- Π^m is a reversible abstract machine is proved in **Pi-/NoRepeat.agda:L223**.
- Def.18 is generalized in Def.20.
- Def.20 and generalized Thm.19 are in **Pi-/Eval.agda:L172**. The proof relies on the finitness of execution traces for Π^m which follows from the finitness of Π^m machine states and the non-repeating lemma for reversible abstract machines (Lem.9).
- Def.21 is defined in **Pi-/Eval.agda:L177**.
- Thm.22 is proved in **Pi-/Properties.agda:L133**.

5.3 Interpreter

- The interpreter is defined in **Pi-/Interp.agda:L12**.
- Thm.23 is proved in **Pi-/Properties.agda:L193**.

5.4 Compact Closed Category

- Thm.24 is proved in **Pi-/Category.agda:L299**.
- Thm.25 does not hold because it does not satisfy the uniqueness condition required by inverse category, which is missing from the proof and previous formalization (Thanks Cole Comfort for pointing this out). The counterexample is in **Pi-/Category.agda:L303**. All arguments in the proof still hold, which is proved in the module at **Pi-/Properties.agda:L991**.
- The code for the remark in the end is in **Pi-/Category.agda:L353-410**.

6 Fractional Types: Π^d

6.1 Abstract Machine Semantics

- The syntax of Π^d is formalized in **PiFrac/Syntax.agda**.
- Def.26 is defined in **PiFrac/Opsem.agda:L103**.
- The transition rules in Fig.8 are defined in **PiFrac/Opsem.agda:L109**.

6.2 Properties

- Lem.27 is proved in **PiFrac/NoRepeat.agda:L17**.
- Lem.28 is proved in **PiFrac/NoRepeat.agda:L38**.
- Lem.29 is proved in **PiFrac/Eval.agda:L18**.
- Π^d is a partial reversible abstract machine is proved in **PiFrac/NoRepeat.agda:L113**.
- Def.30 and Thm.31 is in **PiFrac/Eval.agda:L96**.
- Def.32 and Thm.33 is in **PiFrac/Eval.agda:L153**.
- Thm.34 is proved in **PiFrac/Properties.agda:L20**.

6.3 Interpreter

- The interpreter is defined in **PiFrac/Interp.agda:L15**.
- Thm.35 is proved in **PiFrac/Properties.agda:L40**.

6.4 Compact Closed Category

- Thm.36 is proved in **PiFrac/Category.agda:L110**.

7 Combining Negative and Fractional Types: $\Pi^{\mathbb{Q}}$

7.1 Abstract Machine Semantics

- The syntax of $\Pi^{\mathbb{Q}}$ is defined in **PiQ/Syntax.agda**.
- Def.37 is defined in **PiQ/Opsem.agda:L115**.
- The transition rules in Fig.9 are defined in **PiQ/Opsem.agda:L123**.

7.2 Properties

- Lem.38 is proved in **PiQ/NoRepeat.agda:L17**.
- Lem.39 is proved in **PiQ/NoRepeat.agda:L135**.
- Lem.40 is proved in **PiQ/Eval.agda:L24**.
- $\Pi^{\mathbb{Q}}$ is a partial reversible abstract machine is proved in **PiQ/NoRepeat.agda:L250**.
- Def.41 and Thm.42 is in **PiQ/Eval.agda:L196**. This proof relies on the finitness of execution traces for $\Pi^{\mathbb{Q}}$ which follows from the finitness of $\Pi^{\mathbb{Q}}$ machine states and the non-repeating lemma for partial reversible abstract machines (Lem.13).
- Def.43 is in **PiQ/Eval.agda:L201**.
- Thm.44 is proved in **PiQ/Properties.agda:L217**.

7.3 Interpreter

- The interpreter is defined in **PiQ/Interp.agda:L20**.
- The equivalence between interpreter and machine semantics is proved in **PiQ/Properties.agda:L402**.

8 Programming with Negative and Fractional Types

- All examples except for the SAT solver are in **PiQ/Examples.agda**.
- The implementation of the SAT solver is in **PiQ/SAT.agda**.