Abstract

PPL (PTIME Program Logic) is a language that expresses all PTIME queries on programs. Further, there is an efficient algorithm that turns sentences in this language into programs.

Motivation

- Can we optimize this program in a certain way?
- Requirements:
  - Get the answer fast
  - Generate a program to get the answer
  - Generate this program fast

Proof Technique

- Use Immerman + Vardi Theorem:
  - FOL+LFP expresses all PTIME queries on ordered structures
  - Turn a program into an ordered structure
  - Requirement:
    - Transform fast
    - Keep structure similar to program

Proof Flow

Create a function from programs to structures of a certain type

Show the structures are ordered

Apply Immerman-Vardi

Show the function is an injection

Show the function is PTIME

Use the language to ask questions about the encoded program

Further Studies

- Is there a more natural language for these queries?
- Is there a way to guarantee the generated programs are as efficient as possible?
- Can a similar tactic be used to actually provide optimizations

Uses

PTIME Verification:
- Demonstrates a question can be answered quickly?

Program Generation:
- Given a question, returns a program that can answer the question quickly

Example Answer

Yes, as x=x_9, y=x_10, z=z_11

Sentence Example

Is 0 ever added to a variable on the right?

Is there a variable where the next character is a plus, and the character after that is a 0

Are there three words, x, y, z, where y comes after x, z comes after y, x is a variable, y is a plus, and z is a 0.

Example Answer

Yes, as x=x_9, y=x_10, z=z_11

Sentence Example

Is 0 ever added to a variable on the right?

Is there a variable where the next character is a plus, and the character after that is a 0

Are there three words, x, y, z, where y comes after x, z comes after y, x is a variable, y is a plus, and z is a 0.