Choiceless Polynomial Time: Introduction and Update

Andreas Blass

Microsoft Research
Redmond, WA 98052

and

University of Michigan
Ann Arbor, MI 48109

ablass@umich.edu

Ordered Structures:

- Represented directly as input to Turing machine.
- Polynomial time captured by first-order logic plus least-fixed-point operator. (Immerman & Vardi)

Unordered Structures

- Add an ordering to use as input.
- Answer must be independent of order. (Chandra & Harel)
- No logic known to capture PTIME.
- Conjecture (Gurevich): No logic captures PTIME.

PTIME computation on unordered structures can use an ordering, but must give the right answer regardless of the ordering.

How much can one compute without even using an ordering?

Choiceless Polynomial Time

(Blass, Gurevich, Shelah, 1999)

Intention:

- Prohibit introducing an ordering.
- Equivalently, prohibit arbitrary choices.
- Allow everything else.
  - Parallelism
  - Fancy data structures

Implementation:

- Work with input structure plus hereditarily finite sets over it.
- Compute using abstract state machines.
- Polynomially bound the number of computation steps.
- Polynomially bound the number of active elements.

A state of an abstract state machine is a first-order structure. The machine's program tells how to update certain "dynamic" function symbols. The program is executed repeatedly until the computation is complete. Active elements are (1) the arguments and values involved in updates and (2) all members of active elements.

Some Details

The vocabulary of the structure $HF(A)$ of hereditarily finite sets over an input structure A has symbols for the relations of A and symbols for basic set-theoretic notions: $\in$, $\varnothing$, $\bigcup$, $\{-,-\}$, the set of atoms (i.e., A), and "the unique element of".

Additional *dynamic* function symbols, initially constant with value $\varnothing$, are modified during the computation. They include `Halt` and `Output`.

Terms are built as in first-order logic with the additional constructor

$$\{t(x) : x \in r : \varphi(x)\}.$$

Rules are built from *updates* of dynamic symbols f,

$$f(t_1, \ldots, t_r) := t_0,$$

by conditional branching and parallel combination of the form

$$\texttt{do for all } x \in r, R(x) \texttt{ enddo}.$$

What can be computed in Choiceless Polynomial Time?

Partial positive answer:
Anything PTIME that depends only on ordering a tiny part of the input structure. "Tiny" means approximately $\log n / \log \log n$ in an input structure of size n.

Partial negative answer: Not much.

- $\tilde{\text{C}}$PT can't count.
- Not even modulo 2.
- It can't decide whether a bipartite graph has a perfect matching.
- Shelah proved the zero-one law for $\tilde{\text{C}}$PT-computable properties.

What if you add counting to $\tilde{\text{C}}$PT? Now what can you compute? In particular, can you compute all PTIME properties? Gurevich's conjecture says no.

First Attempt: Bipartite Matching

The examples where $\tilde{\text{C}}$PT can't decide whether bipartite graphs have perfect matchings are very simple ones, easily decided once counting is available. But the general bipartite matching problem is nowhere near that simple, and it was once expected to support Gurevich's conjecture by being undecidable in $\tilde{\text{C}}$PT+Counting.

Theorem 1 (Shelah)**.** *Bipartite matching is decidable in $\tilde{C}PT$+Counting.*

The non-bipartite case remains an open problem.

Second Attempt: Cai-Fürer-Immerman Graphs

Cai, Fürer, and Immerman (1992) exhibited pairs of graphs (with an additional, preorder structure) that are not isomorphic but very difficult to tell apart. A formula of first-order logic plus the least-fixed-point operator cannot tell the two graphs in a pair apart, once the pair is sufficiently large compared to the formula.

Blass, Gurevich, and Shelah (2002) noted that sufficiently padded versions of the Cai-Fürer-Immerman graphs can be distinguished by $\tilde{\mathrm{C}}$PT+Counting, even though first-order logic plus least-fixed-point still can't tell them apart. The unpadded case was left open.

Theorem 2 (Rossman)**.** *The Cai-Fürer-Immerman graphs can be distinguished in* $\tilde{C}PT$*.*

Surprisingly, counting isn't needed. In many cases, the preorder of the graphs isn't needed either.

Theorem 3 (Dawar & Richerby)**.** *The Cai-Fürer-Immerman graphs cannot be distinguished in* $\tilde{C}PT$*+Counting with any fixed bound on the set-theoretic rank of the sets used in* $HF(A)$*.*

Moral of this story:
The availability of fancy data structures, in the form of hereditarily finite sets of arbitrarily high rank, makes a real difference.

Third Attempt: Determinants

Blass, Gurevich, and Shelah (2002) showed how to decide, in $\tilde{\text{C}}$PT+Counting, whether a given matrix over a finite field or over the integers is singular. When the field is $\mathbb{Z}/2$, this suffices to compute the determinant, but for other finite fields and for the integers, the determinant question was left open. (We did get that, over the integers, the set of all prime divisors of the determinant is computable in $\tilde{\text{C}}$PT+Counting, but we didn't get the multiplicities or the sign.)

Recently, Rossman noticed that Csanky's algorithm for determinants over the integers works in $\tilde{\text{C}}$PT+Counting, and I added that one can the compute determinants over finite fields by lifting to quotients of polynomial rings over the integers.

Summary of Attempts

Various PTIME problems have been proposed as possibly not computable in $\tilde{\text{C}}$PT+Counting. In all cases where we know the answer, the problem has turned out to be in $\tilde{\text{C}}$PT+Counting — but sometimes by **very** clever proofs, and not in any uniform way that might generalize to a large class of problems.

Speculation on Bijective Proofs

A bijective proof of a combinatorial identity $|A| = |B|$ is one that exhibits an explicit bijection between A and B.

An example: $\sum_{k \text{ odd}} \binom{n}{k} = \sum_{k \text{ even}} \binom{n}{k}$ for all $n \geq 1$.

Non-bijective, algebraic proof: The difference between the two sides is

$$\sum_{k} \binom{n}{k} (-1)^k = (1 + (-1))^n = 0$$

by the binomial theorem.

Bijective proof: We exhibit a bijection between the collection of odd-sized subsets of $\{1, 2, \ldots, n\}$ and the collection of even-sized subsets. Send any set X to $X \cup \{1\}$ if $1 \notin X$ and to $X - \{1\}$ if $1 \in X$.

Dishonest bijective proof: Use the algebraic proof, and then "exhibit" the lexicographically first bijection.

Trying to Stop Dishonesty

The bijection should depend only on the data given in the problem. In the example, the ordering of the underlying set $\{1, 2, \ldots, n\}$ was artificially introduced. Neither it, nor the derived lexicographic ordering, is really available.

But is the specific element 1, used in the (honest) bijective proof available?

In fact, a bijection between the even-sized and odd-sized subsets of a given S is exactly as "available" as a single odd-sized subset of S.

Feldman and Propp (1995) studied problems about explicitness of bijections, from the point of view of invariance under permutations.

An Oxymoron?

A. Blass and B. E. Sagan, "Bijective proofs of two broken circuit theorems," J. Graph Theory 10 (1986) 15–21.

The notion of a broken circuit in a graph presupposes a linear ordering of the set of edges. That's nearly equivalent to a linear ordering of the set of vertices. What can prevent the "lexicographically first bijection" dishonesty in this situation?

Invariance considerations don't do the job. Suggestion: Computability considerations. The bijection should be easily computable from the data. $\tilde{\mathrm{C}}$PT, perhaps with counting or with additional help, and perhaps with stricter resource bounds, may provide a precise and reasonably honest definition of "bijective proof".