# Logics of Polynomial-Time Computation

Bjarki Holm

Easter 2010

One of the main open questions in descriptive complexity theory is whether there is a logic that can "capture PTIME" — that is, a logic that can express exactly all polynomial-time computable properties of finite structures. By a theorem of Fagin we know that the class NP is captured by the existential fragment of second-order logic. Finding a logic for PTIME would therefore allow logical (in particular, model-theoretic) techniques to be applied to one of the most important questions of computational complexity, that whether PTIME = NP.

Over the past three decades, there have been various attempts to define a logic for PTIME. Most of these attempts have focused on building PTIME "from below", by considering extensions of first-order and fixed-point logics. While the general problem is still wide open, there have recently been a number of partial results on capturing PTIME on restricted classes of structures, such as on graphs with certain nice properties. In this lecture we will review some of this previous work and introduce a few of the current directions of research. Our focus will be on issues of order-invariance and choice, as we explain in more detail below.

**The difficulty of making choice.** Many of the apparent barriers to finding a logical characterisation of PTIME can be related to issues concerning *ordering* and *choice*. It is known that on the class of finite ordered structures (that is, structures equipped with a linear ordering of their elements), we have logics for PTIME as well as many of the complexity classes contained therein. Without an ordering, however, these logics fail to express some very basic properties of low complexity. This gap in expressive power of logics on ordered *vs* unordered structures is generally due to their inability to make arbitrary choice. Choice and ordering can be seen as two sides of the same coin: in the presence of an ordering, we can simulate choice from a set by picking the least element, say, and equipped with an unrestricted choice, we can always impose an ordering on a set by repeatedly choosing one element after another. The ability to make an arbitrary choice is an important part of many common algorithms (e.g. "**while** $S$ is not empty, **choose** a vertex $v$ of degree $\geq 2\ldots$"). For standard computational machine models, this is no difficulty as the input to a computation usually comes with a built-in ordering (the sequence of bits on the input tape of a Turing machine, say). Therefore, in trying to find a logical characterisation of complexity classes — which are, after all, defined in terms of machine models — we are faced with an inherent mismatch between the logical side of computation and the machine model of computation. This apparent mismatch will be the theme of this lecture, as we will discuss topics concerning order-invariance, canonical orderings and counting. Finally, we will briefly introduce some recent attempts to define computational machine models that explicitly forbid arbitrary choice.

**Desirable previous knowledge.** None, I will aim to have this accessible to all Part II/ACS/PhD students (despite some earlier adverts).

1. Grohe, M. The quest for a logic capturing PTIME. *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science (LICS '08)*, pp. 267-271, 2008.

2. Immerman, N. Descriptive complexity: A logician's approach to computation. *Notices of the American Mathematical Society*, vol. 42, no. 10, pp. 1127–1133, 1995.