Neighborhood Models & Descriptive Complexity

Problem Statement

Descriptive complexity aims to organize complexity classes by the expressive power of the languages that are able to describe problems in them. But the field often overlooks the role that the class of models has to play in characterizing this expressive power. A classic argument is that expressivity in a logical language is machine-, resource-, or model-independent, and therefore classes such as P, NP, and PSPACE are made more robust by having this kind of resource-independent characterization (emphasis mine):

> In the beginning, there were two measures of computational complexity: time and space. From an engineering standpoint, these were very natural measures, quantifying the amount of physical resources needed to perform a computation. From a mathematical viewpoint, time and space were somewhat less satisfying, since neither appeared to be tied to the inherent mathematical complexity of the computational problem . . . [Fagin's Theorem] was a remarkable insight, for it demonstrated that the computational complexity of a problem can be understood as the richness of a language needed to specify the problem. **Time and space are not model-dependent engineering concepts, they are more fundamental.**
>
> —Immerman, *Descriptive complexity* [2]

> The technique of the proof . . . was reused by Fagin almost a quarter century later to prove his celebrated result that put the equality sign between the class NP and existential second-order logic, thereby providing a **machine-independent characterization** of an important complexity class.
>
> —Libkin, *Elements of finite model theory* [3]

> This result provided one of the first **machine independent characterizations** of an important complexity class – i.e. **one whose formulation does not make reference to a specific model of computation** . . . The availability of such characterizations is often taken to provide additional evidence for the mathematical robustness of classes like NP.
>
> —Dean, "Computational Complexity Theory" [1] (SEP)

But descriptive complexity classes *are* model-dependent — they depend just as much on the choice of model as they do the choice of language. Historically, these classes have been defined using graph models, i.e. classical models for first- and second-order logics. But this choice was somewhat arbitrary. For example, the theory could have been developed instead using more general *neighborhood space models*.

The purpose of these notes is to explore this possibility — to develop descriptive complexity using neighborhood models rather than graph models. What I have in mind is a set of alternative logical classes that parallel the classes $\mathrm{SO}\exists$, $\mathrm{FO}[\mathrm{LFP}]$, $\mathrm{FO}[\mathrm{DTC}]$, and so on. These in turn will give rise to a set of previously undiscovered computational complexity classes that likewise parallel the classes NP, P, L, and so on. I posit that these new classes are in many ways more natural than their counterparts, in a sense that I will make more precise in writing these notes.

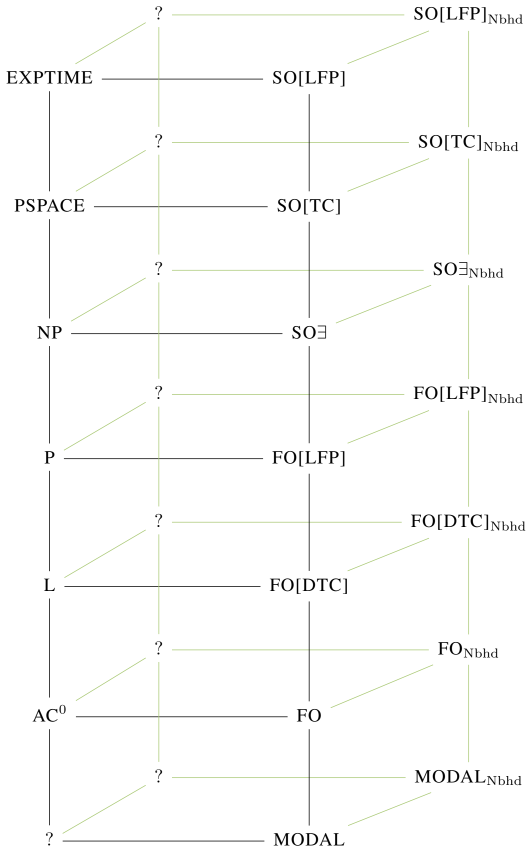
In concrete terms, I would like to answer the questions:

Question 1. What are the "neighborhood descriptive complexity" classes like? Are they actually distinct from their counterparts? (For instance, is $\mathrm{SO}\exists \subset \mathrm{SO}\exists_{\mathrm{Nbhd}}$? Try to give a *specific* problem in $\mathrm{SO}\exists_{\mathrm{Nbhd}}$ but not in $\mathrm{SO}\exists$.)

Question 2. What *computational* complexity classes correspond to the neighborhood classes? What are these classes like? (For instance, what class NP† is equivalent to SO$\exists_{\text{Nbhd}}$? Try to give a specific problem in NP† but not in NP. Additionally, try to give a specific problem that is NP†-complete.)

Question 3. Neighborhood models are in many ways more natural than relational models — for example, in modal logic, relational models carry with them the normal modal axioms, whereas neighborhood models are complete with respect to no axioms at all. In what ways does this sense of "natural" carry over in the complexity context? How are these neighborhood classes more natural than their counterparts?

The inclusion diagram to the right is a roadmap for what's ahead. The black lines show the classical complexity hierarchy alongside the descriptive complexity hierarchy. Classes below are contained in classes above, and horizontal lines indicate equality. The green lines extend the hierarchy to include the neighborhood classes and their computational counterparts. Note that the green inclusions **haven't been proven yet** — they illustrate the conjecture I'm chasing after in these notes.



Basic Setup & Definitions

Computational Complexity Classes

[define, as brief as possible, the basic complexity classes & (will need to define machines too!). I will also need to define reductions and completeness (give examples of complete problems for each!!)]

Descriptive Complexity Classes

[first, define traditional models for modal logic, then in Johan's style define FOL and SOL semantics as dynamic generalizations.]

Descriptive Complexity Classes using Neighborhood Models

[define neighborhood models for modal logic; we can generalize this idea straightforwardly to FOL and SOL; then define the descriptive complexity classes we get by using neighborhood models instead]

Why Consider Neighborhood Models?

[state some classical/already-known theorems that explain why neighborhood models are more general / more natural. Start with the very easy example of modal logic: Neighborhood models are able to express non-monotonicity in a way that relational models cannot. The interesting thing later will be to see how these results translate into insights for complexity theory.]

Progress So Far

[The very first results should focus on NP (SO$\exists$). First, show that the class for SO$\exists$ with neighborhood models is *distinct* from SO$\exists$ — by finding a problem expressible in the former that is not expressible in the latter! (I expect this will be a generalization of the non-monotonicity example from earlier)] [Prove that this new class is closed under FO-reductions (the plan is to essentially do Fagin's proof)] [Identify the class that corresponds to NP on the other side; as a sanity check, prove that it is distinct from NP (includes a problem that NP does not)] [Identify a problem that is complete for this new NP class] [Show that this problem is expressible in SO$\exists$ with neighborhood models] [Prove that the SO$\exists$ neighborhood class is $\subseteq$ this new NP class] [Put it all together to show that the new classes are equal]

[So we would have: A neighborhood model alternative to SO$\exists$ that is distinct from it; an alternative to NP that corresponds to it; example problems in each; an example *complete* problem; and a proof that they're equal] [I will accept this as a proof of concept that I can actually do all this.]

Todo List

□ sorry

References

[1] Walter Dean. "Computational Complexity Theory". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Fall 2021. Metaphysics Research Lab, Stanford University, 2021.

[2] Neil Immerman. *Descriptive complexity*. Springer Science & Business Media, 1998.

[3] Leonid Libkin. *Elements of finite model theory*. Vol. 41. Springer, 2004.