

**Q**  
**Applied Compositional Thinking for Engineers (ACT4E)**



**Session 12 - Computation**

**Questions & Answers**

**Q: What's the title of the song?**

Rino Gaetano - Sforivano le viole

<https://www.youtube.com/watch?v=IYSSrL3dWH4>

loool it's so good :D

**Q: not continuous space means not connected space?**

JL: I think what is meant here by continuous space is something like the “continuum” (e.g. the real numbers) vs discrete spaces such as e.g. a finite set with a poset structure. In co-design, the latter type of spaces are *also* allowed as part of the optimization scenario. (So yes, maybe connected is a good topological adjective here.)

**Q: What is a good reference on Kleene's algorithm? Andrea's work talks about it but I'm looking for a more direct description. Is there a specific page pointer into Davey & Priestly? Part of the challenge is that nowhere in D&P is there any reference to an algorithm by that name. (Looking at a copy in hand...) GZ: added below**

JL: I think the book “Introduction to Lattices and Order” by Davey and Priestley treats this topic.

GZ: B. Davey and H. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2002. DOI: 10.1017/cbo9780511809088.(8.15, CPO fixpoint theorem I, page 183)

E. Manes and M. Arbib. *Algebraic approaches to program semantics*. Springer-Verlag, 1986. DOI: 10.1007/978-1-4612-4962-7.

**Q: Is there software available to solve monotone co-design problems?**

GZ: Yes, it's called PyMCDP <https://co-design.science/software/>. Release for public use will be announced.

**Q: Is there a connection between monotone codesign problems, and submodular set function optimization? Eg how do we formulate “diminishing returns” problems in economics**

[https://en.wikipedia.org/wiki/Utility\\_functions\\_on\\_indivisible\\_goods?](https://en.wikipedia.org/wiki/Utility_functions_on_indivisible_goods?)

**Any references?**

GZ: See answer in class. Interesting question, we haven't looked into it. Reach out if you are interested in working on it.

**Q: Trying to understand how computation and compositionality of solvers relates to things such as**

[https://en.wikipedia.org/wiki/Overlapping\\_subproblems](https://en.wikipedia.org/wiki/Overlapping_subproblems)

[https://en.wikipedia.org/wiki/Optimal\\_substructure](https://en.wikipedia.org/wiki/Optimal_substructure)

[https://en.wikipedia.org/wiki/Dynamic\\_programming](https://en.wikipedia.org/wiki/Dynamic_programming)

**Any thoughts on this?**

GZ: This is an active area of research in our group. I don't have an answer for the exact relations as of now, but I can tell that these are very interesting questions.