Relationship between computer science education and transfer of learning

Simone Martini

June 14, 2019

Computer science education research is gaining more and more importance and attention. Moreover, governments and policymakers are realizing the importance of introducing scientific and cultural elements of computer science (often referred to as “computational thinking” [Win06]) in K-12 education.

The introduction of a new subject that most people are not expert in always comes with oversimplifications and enthusiastic claims about its general value.

In these years, some misconceptions are spreading: mostly unverified claims state that studying computer science (CS) can automatically improve students’ results in other disciplines, or teach them domain-general problem-solving skills (e.g., abstraction, pattern recognition, problem decomposition, logical reasoning) or influence cognitive and affective aspects (e.g., motivation, self-efficacy, resilience, creativity).

These claims are not new to computer science education research: debates dates back to as early as the 1980s, in the context of the educational language LOGO [Pap80, PK84].

Some research and reviews were conducted in the early 90s, leading to negative [Pal90] or only slightly positive [LB91] conclusions.

More encouraging results were, however, obtained when the interventions were explicitly designed to transfer specific competencies [PR91, KC88].

In facts, educational research warns against automatic transfer and agrees transfer is difficult, especially between far domains and especially if activities are not explicitly designed for it [ABD+10, Nat00].

The proposed research can consider the relationship between CS and transfer on several dimensions.

Transfer is likely to happen more easily between near domains and applications (e.g., algorithms in CS and in Math; sequencing of instructions and
sequencing of a story at kindergarten level). Less likely is that transfer will automatically happen between CS and the so-called higher-order thinking skills, whose teachability is highly debated [TS13], or, even worse, to broader aspects like resilience [Lew17].

Moreover, different teaching approaches and tools can influence transfer. For example, the debate between instructionist and constructionist learning theories is deeply linked to the aforementioned LOGO research [PH91]. From one side, a more abstract and top-down instruction seems to foster transfer, since it teaches connections and applicability of the same knowledge in different situations, but, on the other side, a more hands-on bottom-up approach seems to help to create a deeper and more retainable knowledge. This is especially true for beginners, that naturally struggle to go past specific details and to grasp general laws during the initial phases of learning, especially when knowledge is transmitted rather than constructed [FR19].

Finally, successful examples of using CS (and in particular programming) mainly as a tool to foster the learning of other subjects are available: however, the amount of computer science learned by students is confined to what they strictly need to use in the main subject they are learning [Guz15].

Research on transfer and CS is recognized to be contradictory and outdated, and must be conducted in relation to modern learning tools and contexts [Sch16].

The research program will consist of one or more of the following.

1. Comprehensive literature review on the state of the art about the relationship between learning computer science and knowledge transfer to other domains, higher order thinking skills, and affective constructs.

2. Design, implementation, and evaluation of experiments (in K-12 education) to measure the effects of studying or not studying CS / CT on performances in standardized tests for:
   - other domains (e.g., literacy and numeracy)
   - general problem-solving skills
   - affective constructs (e.g., motivation, perseverance)

3. Design, implementation, and evaluation of experiments (in K-12 education) to measure the effects of using programming as a tool for integrated learning (especially in the context of STEM education).

4. Design, test, and evaluation of specific teaching approaches / tools / materials (in the context of K-12 education) to explicitly foster the transfer of skills from CT/CS to different contexts.
5. Design, implementation, and evaluation of experiments (in K-12 education) to study the influence of different pedagogical approaches (e.g., direct instruction vs. constructivist/constructionist approaches) and tools (e.g., visual vs. textual programming languages) on transfer of skills from CT/CS to different contexts.

6. Design, implementation, and evaluation of experiments (in the context of K-12 pre-service teacher training and in-service professional development) to study the relationship between CT/CS training and transfer of skills to different contexts.

References


