A Graph-based Approach to Analyze and Compare Computer Science Curricula for Primary and Lower Secondary Education

Stefan Pasterk and Andreas Bollin
Alpen-Adria-Universität Klagenfurt, Austria
stefan.pasterk@aau.at, andreas.bollin@aau.at

To teach topics of computer science in kindergarten or primary schools, curricula, educational standards and/or competency models were developed and in some countries, like Switzerland or Australia, already established. These models show differences with regard to focus, content, structure and number of skills or competencies which makes a comparison a complex task. This contribution introduces a technique and framework to comprehensibly evaluate different curricula, standards and competency models for computer science education in primary and lower secondary schools.

A Graph-based Approach

Extraction

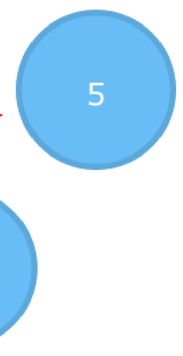
Computing Practice and Programming (CPP)
Grades K–3 (L1:3.CPP)
The student will be able to:

1. Use technology resources to conduct age-appropriate research.
2. Use developmentally appropriate multimedia resources (e.g., interactive books and educational software) to support learning across the curriculum.
3. Create developmentally appropriate multimedia products with support from teachers, family members, or student partners.
4. Construct a set of statements to be acted out to accomplish a simple task (e.g., turtle instructions).
5. Identify jobs that use computing and technology.
6. Gather and organize information using concept-mapping tools.

[1]

Computational Thinking: (CT)
Grades K–3 (L1:3.CT)
The student will be able to:

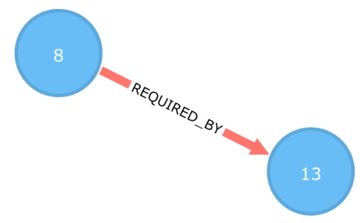
1. Use technology resources (e.g., puzzles, logical thinking programs) to solve age-appropriate problems.
2. Use writing tools, digital cameras, and drawing tools to illustrate thoughts, ideas, and stories in a step-by-step manner.
3. Understand how to arrange (sort) information into useful order, such as sorting students by birth date, without using a computer.
4. Recognize that software is created to control computer operations.
5. Demonstrate how 0s and 1s can be used to represent information.



The idea behind this new methodology is to represented knowledge items of different curricula, educational standards or competency models for computer science as single vertices of a graph including an unique id (displayed by a number in the vertices).

Connection

„Required by“-relation:



„Expanded by“-relation:

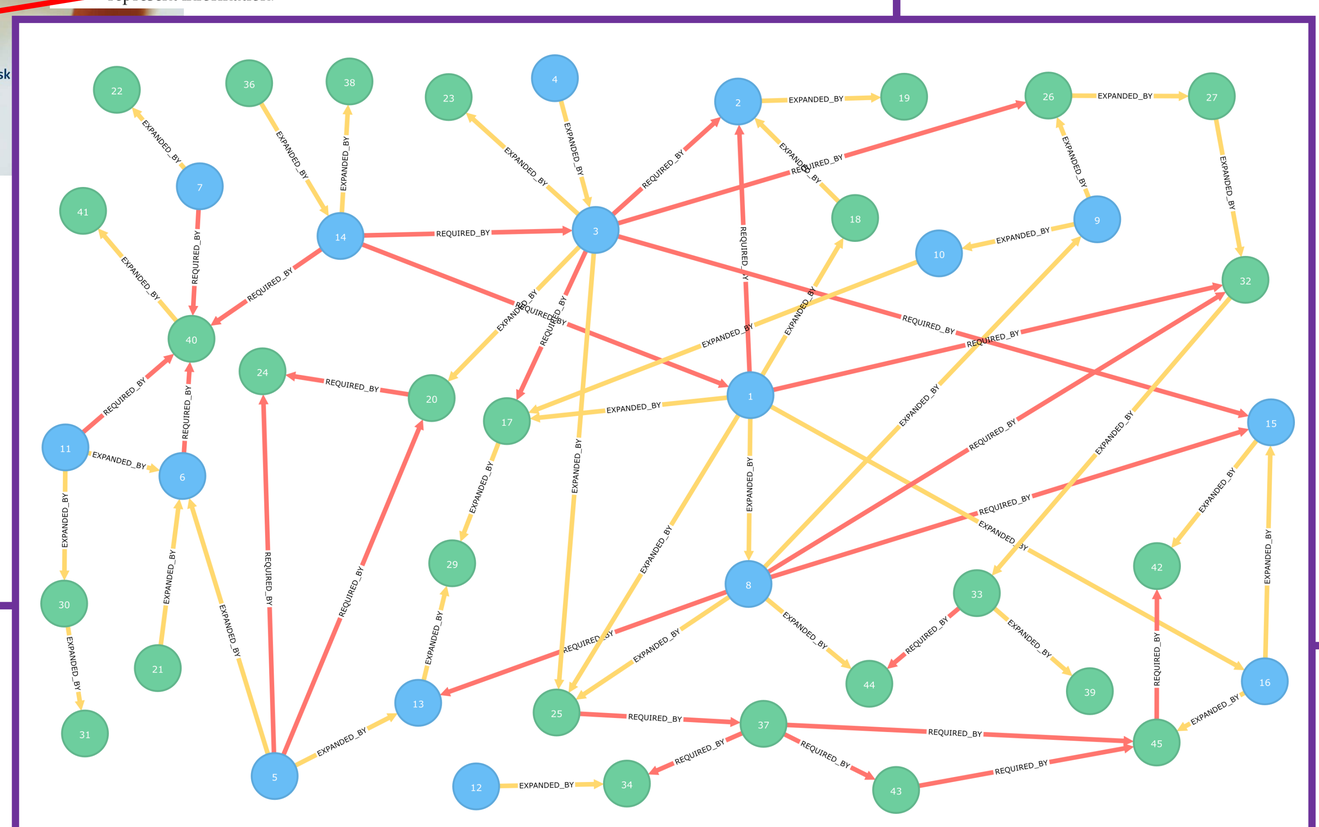


Fig.1: The CSTA K-12 Computer Science Standards (2011) from grade K to 6 represented as graph and categorized by level 1 (blue vertices) and level 2 (green vertices)

The edges of the graph represent the relations between the knowledge items within a curriculum, educational standard or competency model. For this, we classified the relations as either being a ”required by” or an ”expanded by” relation, leading to labeled and directed edges in the graph. The ”required by” relation indicates that one vertex is required by another one, and the ”expanded by” relation represents either a generalization or a specialization relationship.

Metrics

We consider the following different graph-theoretic measures in the analysis to get information about important vertices, complexity and structure.

	Metric	Calculation	Meaning
Important vertices	Maximum degree	Number of edges at a vertex	Central knowledge item
	Source	Vertices without incoming edges	Knowledge item to start with
	Sink	Vertices without outgoing edges	Possible final know-ledge item of a Grade
Complexity	Size	Number of vertices (|V|)	Number of knowledge items
		Number of edges (|E|)	Number of relations
	Density	$\frac{\lvert E\rvert * 100}{(\lvert V\rvert * (\lvert V\rvert - 1))}$	Ratio between number of vertices and number of edges
Structure	Isolated vertices	Vertices without edges	Knowledge items without Connections
	Connected components	Number of not connected sub-graphs	Possible independent topics

Results

Fig.2 contains a comparison of the generated graphs for the curricula from Australia (AUS) and Switzerland (CHE). Experts categorized the knowledge elements of both curricula into “digital literacy”, represented by red vertices, and “computer science”, represented by blue vertices. Table 2 shows the values for the metrics.

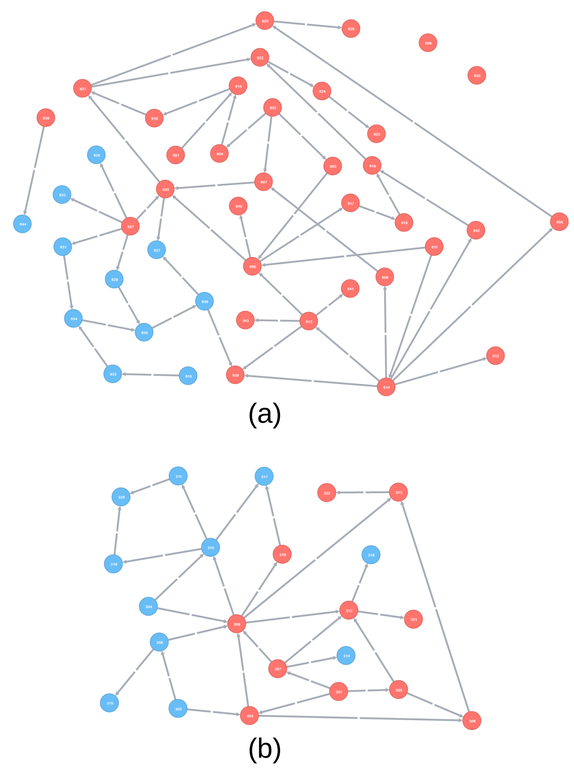


Fig.2: The curricula from Switzerland (a) and from Australia (b) for five to twelve years old students.

	CHE	AUS
Max. degree	7	8
Number of sources	8	3
Number of sinks	13	7
Number of knowledge items	44	22
Number of relations	51	30
Density	2.7	6.5
Isolated vertices	2	0
Connected components	1	4

Future Work

The content of the curricula, educational standards and competency models will be broken down to their basic knowledge items and they will be categorized into knowledge areas. Further a platform (gecko.aau.at) is developed, to collaboratively evaluate graphs representing existing curricula and to give the possibility to develop own learning paths.

References:
[1] D. Seehorn, S. Carey, B. Fuschetto, I. Lee, D. Moix, D. O’Grady-Cunniff, B. B. Owens, C. Stephenson, and A. Verno, “Csta k–12 computer science standards: Revised 2011,” New York, NY, USA, Tech. Rep., 2011.