Concept-Based Student Assessment

Piyush Maheshwari and Nandan Parameswaran

***Abstract*—In this paper, we suggest that a student's answer must be assessed not only for the correctness of the answer but also for the correctness of her understanding of the underlying concepts. Assessing answers for conceptual correctness produces marking that is fairer than assessing for correctness of answer. We propose a concept-based methodology for assessing student answers using concept maps.**

***Index Terms*—Concept map, assessment, education.**

I. Introduction

Teaching involves presenting concepts to students using definitions, illustrations and examples of applications. One of the challenges in teaching is to assess a student's understanding of the concepts. Often it is not clear if the student has understood a concept exactly the way it was taught to him by a teacher. For example, while teaching a programming language, often there is a difference in what an instructor defines as the concept *type* (as a set of objects along with a set of operations) and how a student understands it (often, as a set of objects only). Thus, when a teacher defines STACK as a type, the student may confuse it with a particular instance of STACK with certain contents. Further, typically in an assessment, for example by marking in a written examination, assessing an answer that is completely correct is often easier than assessing an answer that is only partially correct since there is no well-defined scheme for making partial assessment. Typically an answer is marked for the correctness of its answers. Further, in assessment techniques based on problem solving typically assess task and data flow in a student's solution.

In this paper, we suggest that a student answer must be marked not only for the correctness of answer but also for the correctness of his/her understanding of the concepts. Assessing answers for conceptual correctness produces marking that is fairer than marking for correctness of answer. We propose a concept-based methodology for assessing student answers by marking using concept maps.

The rest of the paper is organized as follows. In Section II, we define concepts and concept maps, and discuss extracting concept maps from a given text description. Section III presents our methodology for concept map based marking of student answers. In Section IV, we show results of applying our methodology to a student answers selected from a set of computer science students. Section V discusses the related work and Section VI concludes the paper.

Manuscript received March 11, 2016; revised July 13, 2016.

P. Maheshwari is with the Department of Engineering and Architecture, Amity University Dubai, Dubai International Academic City, UAE (e-mail: pmaheshwari@amityuniversity.ae).

N. Parameswaran is with the School of Computer Science and Engineering, the University of New South Wales, Sydney, Australia (e-mail: paramesh@cse.unsw.edu.au).

II. Concept Maps

Knowledge representation using symbols depends crucially on the notion of concepts where concepts may often model reality. Human thoughts and understanding are filled with concepts. Understanding and knowledge acquisition, which are the corner stones of education, rely to a large extent on the explicit use of concepts. It is thus necessary that assessment techniques be based on concept understanding. Concept-based assessment assesses the student's mastery over concepts presented in a course.

Intuitively, a concept is an idea of something formed by mentally combining all its characteristics [1]. In this paper, we characterize a concept by its name and a set of attributes. Concepts often have relationships between them; some are domain independent and some are domain dependent. For example, bubble sort algorithm is a concept that may be defined as a concept as follows:

bubble-sort

- name: Bubble-sort
- attributes
  1) is-a: Algorithm
  2) input: set of numbers
  3) output: ordered set of numbers
  4) :O(n^2)

In general, the attributes will depend on the domain and applications. We can similarly define another concept called *Algorithm*:

algorithm

- name*: Algorithm*
- attributes
  1) is-a: Procedure
  2) input: set of objects
  3) output: set of objects
  4) method: sequence of steps that are executable with finite resources

We can view *is-a* as a relationship between the concepts Bubble-sort and Algorithm. In general, a concept may be related to more than one concept.

A concept map is a graph where each node is labeled with a concept and each edge is labeled with a relationship between the two nodes that the edge connects. In this paper, however, by concept map we specifically refer to a concept graph that we extract from the description consisting of text, mathematical expressions, programs and figures that we normally come across in the field of computing. We now discuss how concept graphs can be drawn from a given description.

doi: 10.18178/ijiet.2017.7.7.929

III. METHOD FOR EXTRACTING A CONCEPT MAP

Input: A description consisting of text, mathematical expressions, programs and figures.

Output: A concept map

Method:

Step 1: Identify all items that are considered relevant as concepts for assessment.

Step 2: Create a concept node for each concept identified through the keywords.

Step 3: Identify all items that are considered relevant as relations (between two concepts) for assessment.

Step 4: Create an arc between the relevant concepts for each relation and label the arc with the name of the relation.

Step 5: Repeat Step 1 through Step 5 this time identifying collections of items.

Step 1 through Step 4 will give concepts and relations that are explicitly present in the description. Often, however, concepts and relations are also embedded in the description implicitly and they are identified in Step 5.

We distinguish two types of nodes: *primitive concepts* and *abstract concepts*. A concept is said to be abstract if it is defined using other concepts, otherwise it is said to be primitive. For example, in programming, Queue is considered to be an abstract concept while Integer a primitive concept. (In a concept map, primitive nodes may not have outgoing arcs.)

A. Text based Descriptions — Single Concept

In this, the text is in the form of description containing only English text without mathematical expressions, programs and figures, and the text describes a single concept. Given such a text based description, we identify keywords, phrases, and sentences that either directly or indirectly correspond to a concept or relation. We will provide an example below.

Example 1:

A text based description, that defines a concept called Tree, is given below.

Definition: A binary tree is a collection of nodes. The collection can be empty, which is sometimes denoted as NULL. Otherwise, a tree consists of a distinguished node r, called the root, and zero, one or two (sub) binary trees.

In the description above, we identify the keywords and phrases as concepts and relations as listed in Table I.

These concepts are adequate for our purpose. (Note that there may be other concepts and relations present implicitly in the description, but they are considered redundant for our purpose.) We now can draw the graph for the concept Binary-tree, as shown in Fig. 1 where the node Binary-tree is called the root node (Additional nodes and relations have been added for the sake of completeness).

B. Algorithmic Description

In this, the given description is in the form of an algorithm. We consider algorithms given in pseudo code.

Example 2

Consider the pseudo code description of a binary tree search algorithm in Fig. 2 below.

Fig. 3 shows a concept map for ordered binary tree search with Binary-tree-search as the root node, and it shows Binary-tree-search as an abstract concept consisting of several sub concepts.

TABLE I: CONCEPTS AND RELATIONS OF BINARY TREE

Keyword or phrase	**Concept or relation?**	**Name**
binary tree	Concept	Binary-tree
collection of	Relation	collection- of
node	Concept	Node
collection	Concept	Collection
empty	Concept	Empty
consists of	Relation	Consists-of
distinguished node	Concept	Uniqueness
root	Concept	Root
(sub) binary tree	Concept	Sub-tree

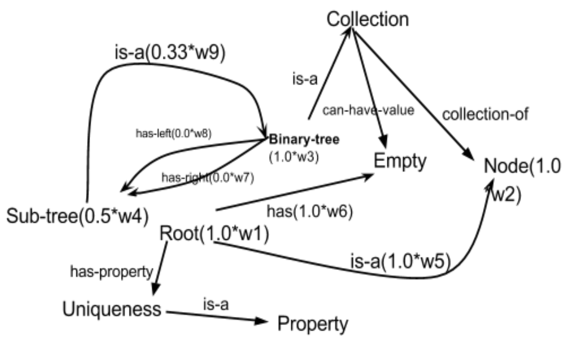

Fig. 1. Concept map for binary-tree. The concepts property, empty and node are primitive concepts. (Note that their nodes do not have outgoing arcs showing that they do not depend on other concepts).

```
def search_recursively(key, node):
    if node is None or node.key == key:
        return node
    elif key < node.key:
        return search_recursively(key, node.left)
    else:  # key > node.key
        return search_recursively(key, node.right)
    return None
```

Fig. 2. Binary tree search algorithm pseudo code.

It may be noted that it does not capture all aspects of the search algorithm, but only those concepts that may be considered relevant for our assessment.

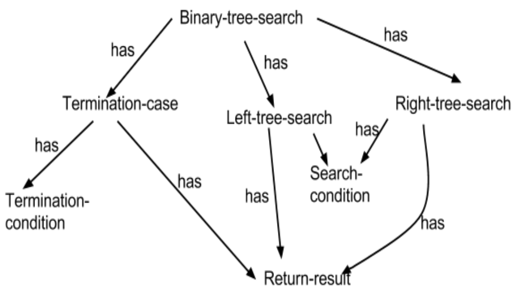

Fig. 3. Concept map for Binary-tree-search.

C. Data Structure Diagrams

In computer science, often diagrams are used in descriptions. Data structure diagrams are used to show complex organizations of data and the relationship between the data elements.

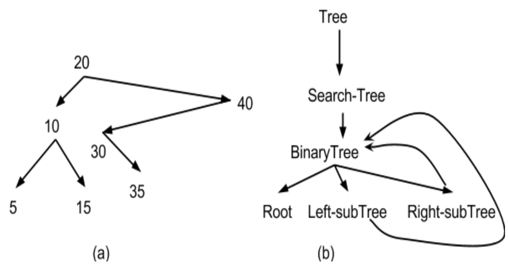

Fig. 4. (a) Binary tree and (b) its concept map.

Fig. 4 shows a binary tree and its concept map. Concept map is drawn in relation to a set of evaluation criteria. Thus, the concept map in Fig. 4(b), with it root node Tree, can be used only for assessing the presence or absence of the concepts in the map. Thus, it is possible that there may be more than one binary tree for a given concept map. In order to specify a particular tree structure, a set of axioms are needed along with a concept map. In this example, we need the following axioms: (a) Left-subtree and Right-subtree may be empty; (b) root of the overall tree is 20.

D. Mathematical Derivations

In student assessment, we often come across derivations which an assessor needs to verify. Fig. 5 shows a simplified concept map with root node Derivation. Derivation is defined as a sequence of valid steps where each step is in the form of a mathematical equality.

E. Text based Description — Multiple Concepts

Often in an assessment, students provide long textual descriptions that tend to describe multiple concepts. This for example occurs when a student describes the function of a device such as a mechanical device, the working of an algorithm, etc. Such descriptions may give rise to concept graphs that have multiple root nodes with concepts implicitly defined. We consider an example below.

Although 1,000n is larger than n^2 for small values of n, n^2 grows at a faster rate, and thus n^2 will eventually be the larger function. The turning point is n = 1,000 in this case. The first definition says that eventually there is some point n_0 past which $c*f(n)$ is always at least as large as T(n), so that if constant factors are ignored, f(n) is at least as big as T(n). In our case, we have T(n) = 1,000n, f(n) = n^2, n_0 = 1,000, and c = 1. We could also use n_0 = 10 and c =100. Thus, we can say that 1,000n = O(n^2) (order n-squared). This notation is known as Big-Oh notation. Frequently, instead of saying "order . . . ," one says "Big-Oh . . . ." [2].

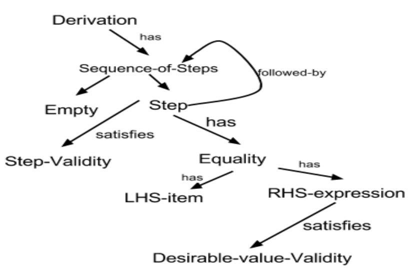

Fig. 5. Concept map for derivation.

The text above is an illustration of how to express a given function in terms of an important concept called the Big-Oh notation from the field of Algorithms. In this process, the description uses several other concepts as shown below. In order to obtain the concept map for the description above, we first need to simplify the description by retaining those concepts that we consider relevant to assessment (in this case) and eliminating the rest. The resulting description is shown below.

1) Identify Dominating term
- $1{,}000n$ is larger than n^2 for small values of n;
- n^2 grows at a faster rate;
- n^2 will eventually be the larger function;

2) Identify $T(n)$ and $f(n)$
- $T(n) = 1{,}000n$, $f(n) = n^2$;

3) Identify the turning point
- $n_0 = 1{,}000$, and $c = 1$; or $n_0 = 10$ and $c = 100$;

4) Claim: $T(n) = O(f(n))$
- $1{,}000n = O(n^2)$.

Fig. 6 shows a concept map for a further simplified description, where we have shown four major concepts (bold font). Each concept is an Identification process which follows a certain temporal order. (Note that for the sake of simplicity, we have not shown the sub steps such as 1a, 1b, etc.)

Using the concept maps, we can now examine the student description of the same set of concepts and relations and look for similarity. Ideally, concepts must be identical and relations between them must also be identical. However, not all concepts and relations may be present in the student description. For example, with respect to the concept Node in Fig. 1, the student description may not have the concept corresponding to the concept Empty.

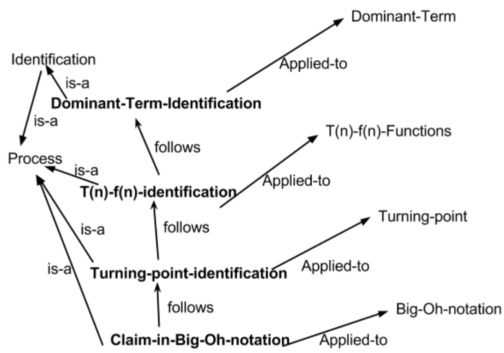

Fig. 6. Concept map for multiple concepts description.

Also, certain concepts C, *subtree1* for example, may not be present in the identical form. Thus, it may be necessary to infer them to obtain a partial match. In such cases, we look for a concept C′ that may be approximately equivalent to C (for example, *node*), match C with C′ and obtain a partial match. Thus, matching *subtree1* against *node* provides a partial match with a score s < 1.

IV. Assessment Using a Concept Map

We now discuss how assessment of a student description is done using a concept map.

A method for assessing student answer using a given concept map

Input: Concept map for a sample correct answer and a student answer.

Output: An assessment score.

Method:

1) Consider a node v_0 of the graph chosen arbitrarily where the concept Cv_0 associated with this node is a primitive concept.
2) Examine the student answer and check if Cv_0 is defined. If Cv_0 is defined explicitly, then assign a full score of 1 to the node v_0. Otherwise, Cv_0 may be implicitly defined using several phrases and sentences. If Cv_0 is implicitly defined in the student answer, then assign a score of p where p is the inverse measure of implicitness. If neither explicit description nor the implicit description of Cv_0 is present, then assign a score of 0.
3) Repeat the above process for every node of the graph.
4) Repeat Steps 1 and 2 above for every edge in the concept graph.
5) Compute the overall score as $(s_1*w_1 + s_2*w_2 + \ldots + s_n*w_n)/(w_1+ w_2\ldots+ w_n)$ where s_i is the score of the node v_i, w_i is the weight attached to v_i and n is the total number of nodes in the tree.

We now illustrate the method above using an example.

Student Answer (a sample):

A binary tree is made of nodes, where each node contains a "left" reference, a "right" reference, and a data element. The topmost node in the tree is called the root. Every node (excluding a root) in a tree is connected by a directed edge from exactly one other node. This node is called a parent. On the other hand, each node can be connected to arbitrary number of nodes, called children. Nodes with no children are called leaves, or external nodes. Nodes which are not leaves are called internal nodes. Nodes with the same parent are called siblings.

Using the concepts in Fig. 1, we start matching each node with the text above. Our first task is to identify the nodes that are considered essential for assessment. In Fig. 1, we consider the following as essential nodes: Node, Root, Binary-tree and Sub-tree. Other nodes are not considered essential for assessment in this example.

First, we observe that in Fig. 1, the concepts Node, Empty, and Property are primitive as they do not have support from other concepts. Other concepts are non-primitive concepts. For example, the concept Root is non-primitive since it is defined as a Node. We choose arbitrarily the primitive concept Root and look for its presence in the student answer in a valid semantic context. It is present in the answer (marked bold face), and the score for this node is 1. We similarly notice that the concept Node and Binary-tree are present (bold face). However, the concept Sub-tree is not present explicitly; however, it is present implicitly (italicized bold face). So, we assign a partial score of 0.5. We next verify if the nodes occur in the right context of relationship. These relationships are:

is-a (Root, Node): present, score s=1.0

has (Binary-tree, Root): present; score s=1.0

has-left(Binary-tree, Sub-tree): incorrect relationship; score=0.0 [italicized bold face]

has-right(Binary-tree, Sub-tree):incorrect relationship; score = 0.0

is-a(Subtree, Binary-tree):present implicitly; score = 0.33 [deeper implicitly]

Thus, aggregating similarity values over nodes and edges, we can obtain the overall value of similarity by weighted multiplication and normalization:

$$(1.0*w_1+1.0*w_2+...+0.33*w_9)/(w_1+w_2+...+w_9)$$

Assuming $w_1= w_2=...= w_9=1$, we have 5.3/9 = 0.58

Thus, the semantic gap between the concept map in Fig. 1 and the student answer is: 1–0.58 = 0.42.

V. Related Work

Work done in e-learning has influenced the way teaching can be done in terms of customized pedagogy, student model, collective learning, and flexible assessment [3], [4]. Senthil *et al.* have proposed techniques for assessing short answers using ontology mapping techniques [5].

Studies in psychology show that concepts play a central role in human understanding [6]. Concepts as formal entities have been investigated in areas such as concept graphs and ontology. Ontology is defined as an explicit specification of a conceptualization [7]. Considerable work done in the field of ontology has focused on techniques for automatically identifying concepts and the relationships amongst themselves in real world applications [8]. Ontologies have been widely used in medicine [9], engineering [10], and education [11]. Similarity between concepts has also been a topic of intense research that has resulted in a wide variety of algorithms and implemented systems [12]-[13]. Conceptual modeling and ontologies has increasingly found its way in the field of education [14]-[16] and has created considerable challenges in modeling and presentation. To our knowledge, concept map as the formal basis for student assessment has not been attempted so far.

VI. Conclusion

We applied our methodology for assessing student answers in a course on Design and Analysis of Algorithms in the field of computer science and marked the student answers using concept maps. The score thus obtained was considered as a measure of *understanding of the underlying concepts* by the student. The difference between the maximum mark and the mark that was obtained through assessment was considered as a measure of *lack of knowledge*.

Our methodology is useful not only in marking but also in structuring questions and in the design of lecture of materials. It may even be suggested that while organizing teaching materials, it should be clearly understood if a topic that is taught is assessable or not. In the current climate, it is reasonable to assume that students would lack motivation in learning something that is not going to be assessed. Our methodology also provides an objective way of assessing student performances. The methodology can be applied to other types of performances as well such as assignments, projects and seminar presentations. Also, our technique lends itself for easy implementation, and can help in organizing teaching slides based on concept flow.

Knowledge representation using symbols depend crucially on the notion of concepts where concepts model reality. Human thoughts and understanding are filled with concepts. Understanding and knowledge acquisition, which are the corner stones of education, are possible only with the use of concepts. It is thus necessary that assessment techniques be based on concept understanding. Concept based assessment assesses the student's mastery of concepts presented in a course. However, in assessment techniques based on problem solving typically assess the correctness of task and data flow only in a student's solution.

REFERENCES

[1] Dictionary.com. (Sept 2015). [Online]. Available: http://dictionary.reference.com/browse/concept

[2] M. A. Weiss, *Data Structures and Algorithm Analysis in C*, Section 2.1, Addison Wesley, 2000.

[3] R. Siddiqi and C. J. Harrison, "Improving teaching and learning through automated shortanswer marking," *IEEE Transactions on Learning Technologies*, vol. 3, no. 3, pp. 237-249, 2010.

[4] S. Jordan and T. Mitchell, "e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback," *British Journal of Educational Technology*, vol.40, no. 2, pp. 371-385, 2009.

[5] V. S. Kumaran and A. Sankar, "Towards an automated system for short-answer assessment using ontology mapping," *International Arab Journal of e-Technology*, vol. 4, no. 1, January 2015.

[6] B. Smith, "Beyond concepts: Ontology as reality representation," *Proceedings of FOIS 2004*.

[7] T. R. Gruber, "A translation approach to portable ontologies," *Knowledge Acquisition*, vol. 5, no. 2, pp.199-220, 1993.

[8] M. Hazman, El-Beltagy, and A. Rafea, "Survey of ontology learning approaches," *International Journal of Computer Applications*, vol. 22, no. 9, May 2011.

[9] G. Colombo, D. Merico, G. Boncoraglio, F. D. Paoli, J. Ellul, G. Frisoni *et al*., "An ontological modeling approach to cerebrovascular disease studies: The NEUROWEB case," *Journal of Biomedical Informatics*, vol. 43, no. 4, pp. 469-484, 2010.

[10] B. Swartout, R. Patil, K. Knight, and T. Russ, "Toward distributed use of large-scale ontologies," in *Proc. the Symposium on Ontological Engineering of AAAI*, pp. 138-148, AAAI, 1996.

[11] P. Brusilovsky, S. Sosnovsky, and O. Shcherbinina, "Quiz guide: Increasing the educational value of individualized self-assessment quizzes with adaptive navigation support," in *Proc. e-Learn 2004*, Washington, DC, USA: AACE, 2004, pp. 1806-1813.

[12] R. Thiagarajan, G. Manjunath, and M. Stumptner, "Computing semantic similarity using ontologies," presented at the International Semantic Web Conference (ISWC), 2008, Karlsruhe, Germany.

[13] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research*, vol. 11, pp. 95-130, 1999.

[14] N. Kitani and S. Yukita, "The educational uses of mathematical ontology and the searching tool," presented at 38th ASEE/IEEE Frontiers in Education Conference, 2008.

[15] International Conference on Formal Ontology and Information Systems, Turin, 4-6 November 2004.

[16] S. Corcho and A. Gomez-Perez, "A Roadmap to Ontology Specification Languages, Rose Dieng and Olivier Corby (eds.)," *Knowledge Engineering and Knowledge Management. Methods, Models and Tools*, Springer, Berlin, pp. 80-96, 2000.

P. Maheshwari is a well-recognized thought leader in the area of enterprise and solutions architecture, cloud-, IoT- and SOA-based software systems, system integration, and mobile-enabled applications. He has notable experience in the field of architecture design, innovations in distributed systems and problem solving in complex technological environments. He had extensively consulted for ICT companies like Ericsson, IBM and Pitney Bowes where he had contributed to end-to-end solution architecture design for business applications using mobile services, sensor networks and e-commerce platforms.

He had extensively lectured in computer science and information technology at the *University of New South Wales*, Sydney and *Griffith University*, Brisbane on the topics related to software architectures, parallel & distributed computing, and computer algorithms. In the past (1991-2006), he had held several research grants from the Australian Research Council (ARC) and well-funded projects in peer-to-peer computing from Microsoft Australia and other private companies.

Dr. Maheshwari earned his Ph.D. in computer science from the *University of Manchester*, U.K.. He also has a B.Eng. and M.Eng. from *Indian Institute of Technolgy Roorkee* (IIT-R), India. He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE) and Computer Society.

N. Parameswaran is an adjunct senior lecturer in the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. He works in the area of ontology based intelligent agent technology, wireless sensors applications, natural language processing, and video event annotations. His teaching interest includes artificial intelligence, internet programming, and multiagent systems.