

Top-Down Progressive Computing

Yiyu Yao and Jigang Luo

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
{yyao, luo226}@cs.uregina.ca

Abstract. A top-down, step-wise progressive computing model is presented as a mode of granular computing. Based on a multilevel granular structure, progressive computing explores a sequence of refinements from coarser information granulation to finer information granulation. A basic progressive computing algorithm is introduced. Examples of progressive computing are provided.

1 Introduction

In the search for new paradigms of computing, there is a recent surge of interest, under the name of granular computing [1, 9, 13], in computations using multiple levels of abstraction and granulation. To a large extent, the majority of existing studies are restricted to rough sets, fuzzy sets, cluster analysis, and classical divide and conquer methods [12, 16], and aim at solving specific problems. There are urgent needs to move beyond these limitations, to define scope and goals of granular computing, and to demonstrate the unique contributions of granular computing.

The triarchic theory of granular computing [15, 16, 18, 19] attempts to promote granular computing as a multidisciplinary and an interdisciplinary study. It presents a conceptual model of granular computing based on the notions of multilevel and multiview granular structures and the granular computing triangle. The multilevel requires that a granular structure consists of a family of integrative levels with different granularity [17]; the multiview requires that it may be necessary to consider a family of multilevel structures with each representing a different view. The granular computing triangle suggests that a study of granular computing consists of three parts, namely, the philosophical position, the methodology foundations, and computation paradigms of granule computing. They represent structured thinking, structured problem solving, and structured information processing, respectively. A study of granular computing serves two purposes for human and machine problem solving [18]. Granular computing for humans focuses on granule-based methodology of problem solving. It aims at empowering humans with granular computing methodology. Granular computing for machines concentrates on mechanized granular information processing. It aims at implementing human-inspired intelligent information processing systems.

Granular thinking may improve human problem solving and granular information processing may improve machine problem solving.

The triarchic theory describes granular computing at a much abstract, or coarse-grain, level of granularity. To make the model practically useful, we must further develop and refine the theory at more concrete and fine-grain levels. More specifically, we must develop and formulate concrete methodology and computational algorithms of granular computing based on multilevel granular structures. A unified framework for studying multilevel, set-theoretic granular structures is given in a recent paper [20]. A multilevel granular structure suggests three possible modes of computation, namely, top-down, bottom-up and middle-out approaches. The main objective of this paper is to outline a top-down, step-wise progressive computing mode and examine several examples.

2 A Model of Top-Down Progressive Computing

Top-down progressive computing is guided by and explores a multilevel granular structure in structured thinking and structured information processing. In some situations, one can first explicitly construct a multilevel granular structure and then process the structure from top levels with larger granularity to bottom levels with smaller granularity. More often than not, one combines the construction and processing of a multilevel granular structure simultaneously in a top-down manner.

2.1 Multilevel Granular Structures

A multilevel granular structure is a fundamental concept in granular computing. Basic ingredients of a granular structure are granules, levels, connections of granules and connections of levels. A granule is a unit, or a focal point, of discussion at a particular level. A level is populated by granules of the same granularity or similar nature. Levels are partially ordered by their granularity. Every level is a representation of a problem and may serve a particular purpose. A multilevel granular structure gives multiple representations of the same problem.

The basic ideas of multilevel granular structure can be illustrated by maps with different granularity. A world map can be seen from the level of countries. Every country is treated as a basic unit and a world map is represented by the countries and their connections. At the next level, there is a map for each country, where every city is a basic unit. A map is represented by cities and their connections. At yet another level, each city has its own map where different regions are basic units. In this way, more detailed maps can be further developed. Many versions of maps thus provide a multilevel granular structure.

The construction and interpretation of granules and levels depend on specific problems. Nevertheless, it is still possible to list some desirable properties about a multilevel granular structure [17]:

- Granules in a particular level are relatively independent or loosely related. Each granule provides a local, partial description and all granules in the level collectively provide a global, complete description.

- A granular structure offers multiple representations and descriptions of the same problem. Representations at different levels must be consistent with each other. A problem in a lower level must have a corresponding problem in a higher level; a solution to a higher level problem can serve as a guide for searching for a solution to the same problem at a lower level.
- Although different levels in a granular structure represent the same problem, different vocabularies and languages may be used to define and interpret granules at different levels and different methods may be used for processing at different levels.
- Levels are partially ordered by their granularity. A lower level granulation has more detailed information than a higher level granulation. That is, a lower level granulation is a refinement of a higher level granulation, and a higher level is an abstraction of a lower level. Granules in a particular level are used to explain larger granules in the next higher level and, at the same time, are explained by smaller granules in the next lower level.
- The number of levels is not prefixed. It is possible to combine several levels into one or to split a level into more levels in the construction of a granular structure.
- Interactions between granules are mostly restricted to three levels, namely, the current level, the next lower level, and the next higher level. This restriction would greatly reduce the complexity of problem solving.
- There are bidirectional influences between two adjacent levels. A higher level typically controls and specifies its next lower level. Conversely, a lower level may also cause restructuring of its next higher level.
- There are at least two basic operations for level transformation. A refinement or zoom-in operation transforms a representation or description in a higher level into one in the next lower level. An important feature of this operation is the addition or fill-in of details. A coarsening, abstraction, or zoom-out operation transforms a representation or description in a lower level into one in the next higher level. An important feature is the omission of details so that only crucial information is preserved.

A granular structure therefore provides structured descriptions of a problem with multiple levels of abstraction and detail. This useful structure serves as a basis of structured approaches of granular computing.

2.2 A Basic Progressive Computing Algorithm

By the refinement-coarsening relationship, we can organize different granulations into a hierarchy. In this paper, we consider a sequence of granulations and levels. If G_2 is a refined granulation of G_1 , and G_3 is a refined granulation of G_2 , \dots , and G_n is a refined granulation of G_{n-1} , then the sequence $G_1, G_2, \dots, G_{n-1}, G_n$ is an n -level granular structure. One way to use this multilevel granular structure is top-down progressive computing. The main idea is to process granulations level-wise from the coarsest granulation G_1 to the finest granulation G_n .

Top-down progressive computing involves two basic tasks, namely, to build a multilevel granular structure and to compute with the structure. We can build a

multilevel granular structure from top to bottom in a progressive manner. Once a multilevel granular structure is constructed, the same top-down progressive computing mode is used to find a solution to a problem. We find an inaccurate solution in a coarser granulation and use the solution as a guide for finding a more accurate solution in a finer granulation. This process is repeated to refine progressively a solution until a satisfactory solution is found. As mentioned earlier, the two tasks can be performed simultaneously in a top-down progressive way.

Alternatively, a granular structure may be constructed in a bottom-up way and is processed in a top-down mode. The bottom-up mode is a data-driven process when a general understanding of a problem is lacking; the top-down model is a conceptually guided process when an in-depth understanding is available. Both modes are effectively used in human cognition and problem solving [6]. In the context of computer programming, Knuth [8] concisely summarizes that "... top-down and bottom-up were opposing methodologies: one more suitable for program exposition and the other more suitable for program creation."

The top-down progressive computing mode may speed up the problem solving process. The number of granules in a higher level is usually smaller than that in a lower level. Finding a solution in a higher level is relatively faster. Once a solution in a higher level is found, it can serve as a guide to find a solution in a lower level, which may avoid unnecessary work in a lower level.

In specifying a top-level, generic progressive computing algorithm, we consider the following basic ingredients:

- Multiple representations of a problem. This is the basis of progressive computing. In fact, progressive computing is a sequence of refinements from coarse descriptions at higher levels to refined descriptions at lower levels.
- Refinement operation on granulations. A granulation refinement operation is needed so that one can transform a coarser description into a finer description. It is important that a refinement operation preserves certain consistency properties of different descriptions.
- Refinement operation on solutions. A solution refinement operation revises an inaccurate, approximate solution in a higher level into a more accurate solution in a lower level.
- Evaluation function. An evaluation function measures the fitness of a solution obtained in a level. Once a satisfactory solution is obtained, one can stop the progressive computing process.

By utilizing these ingredients, a basic progressive computing algorithm (BPC) is given in Algorithm 1. In BPC, the granulation refinement and solution refinement operations are denoted by *granulation_refinement* and *solution_refinement*, respectively. G_0 and P_0 are some initial values for granulation (G) and solution (P). In the very first step, $G_1 = \text{granulation_refinement}(G_0)$ produces the coarsest granulation G_1 in the granulation sequence G_1, G_2, G_3, \dots and $P_1 = \text{solution_refinement}(P_0, G_1)$ is the very first solution in the solution sequence P_1, P_2, \dots . BPC uses a function *fitness* to evaluate a solution, and the process terminates once a satisfactory solution is found. It should be noted that

sometimes the refined granulation itself is a solution. In this case, the solution-refinement step may be simply removed by using G_k as a solution.

Algorithm 1: BPC (Basic Progressive Computing Algorithm)

```

1 begin
2    $k \leftarrow 0$ ;
3   set  $G_0$  and  $P_0$ ;
4   repeat
5      $k \leftarrow k + 1$ ;
6      $G_k = \text{granulation\_refinement}(G_{k-1})$ ;
7      $P_k = \text{solution\_refinement}(P_{k-1}, G_k)$ ;
8      $F_k = \text{fitness}(P_k)$ ;
9   until  $F_k$  satisfies a certain condition ;
10 end

```

3 Examples of Progressive Computing

The top-down progressive computing is applicable in many fields and can be used by both humans and machines. In this section we give some examples that use the top-down progressive computing mode. The first two examples demonstrate how humans use the top-down progressive computing, the last three examples demonstrate how machines use top-down progressive computing.

3.1 Progressive Computing as a Methodology of Problem Solving for Humans

Computer Programming The top-down progressive computing mode can be used in the process of writing a computer program [5]. The program construction consists of a sequence of refinement steps. In each step, a given task is broken up into a number of subtasks. Each refinement in the description of a task may be accompanied by a refinement of the description of the data that constitutes the means of communication between the subtasks [11]. There is a multilevel granular structure in developing a program and granulations at different levels are presented by different languages. The top level is the natural language description of the program, the second high level is given by some model language such as UML or flow chart, the next level granulation is detailed specifications of the modules and algorithms that are described by pseudo-codes and natural language, the lowest level granulation is the program that is written by a programming language. This multilevel granular structure can be built up by the top-down progressive computing mode.

Scientific Writing The top-down progressive computing mode is also used in scientific writing [2, 14]. A scientific paper can be viewed as a multilevel granular structure, the top level is the thesis of the paper, the lowest level is the scientific paper itself, and the medium levels are various outlines of the paper. One level description is given by section headings. Another level of description is given by sub-section headings. In the top-down progressive computing mode for scientific writing, the top granulation is a statement of the thesis. The thesis may be explained by a few section headings or statements in the second level. Every section can be further divided into sub-sections to form the next level granulation. This dividing process continues until we get the entire paper.

Remarks Although computer programming and scientific writing are very different at a lower level, they are much the same at a higher level. Both of them can be considered as applications of the same methodology of progressive computing. When a program or a paper is finished, one typically only sees the final product rather than all intermediate drafts. Unfortunately, the methodology used by a programmer or a scientist is not directly visitable from a program or a paper. One goal of granular computing is to make such methodology more explicit and better described and documented so that many more people can use them. For example, the mode of progressive program development and scientific writing can be easily used in producing mathematical proofs. The same progressive mode can also be used in building models or analyzing data in many other applications.

3.2 Progressive Computing as a Paradigm of Information Processing for Machines

Variable Precision Logic Variable precision logic [7] is to reason with incomplete information and under time constraints. It handles trade-offs between the precision of inferences and the computational efficiency of deriving them. A two level top-down progressive computing mode can be used in the reasoning with variable precision logic.

The variable precision logic is based on augmented production rules of the following form:

$$\begin{array}{ll}
 \textit{if} & \langle \textit{premise} \rangle \\
 \textit{then} & \langle \textit{decision} \rangle : \gamma \\
 \textit{unless} & \langle \textit{censor} \rangle
 \end{array} \tag{1}$$

where $\langle \textit{premise} \rangle$ is a major condition, $\langle \textit{censor} \rangle$ is a minor condition, γ is a certainty value that is bigger than 0.5 and less than 1, and $\textit{decision}$ is a result of the variable precision logic. There are three types of inferences in the variable precision logic:

- If $\textit{premise}$ holds and \textit{censor} is unknown, give $\textit{decision}$ with certainty γ .
- If $\textit{premise}$ holds and \textit{censor} holds, give $\neg \textit{decision}$ with certainty 1.
- If $\textit{premise}$ holds and \textit{censor} does not hold, give $\textit{decision}$ with certainty 1.

In terms of top-down progressive computing, G_0 is the top granulation containing only one granule that is the information about *premise*. P_0 is a decision with certainty γ . $G_1 = \textit{granulation_refinement}(G_0)$ returns the lowest level granulation consisting of two granules, containing information about *premise* and *cursor*. The operation $\textit{solution_refinement}(P_0, G_1)$ returns a decision with certainty 1. The function $\textit{fitness}(P)$ indicates the certainty of P , where P is a decision.

Progressive Image Transmission Progressive image transmission is used in image transmission on the Internet. The principle of progressive image transmission is to transmit an image stage by stage, at every stage some specific information in the image is transmitted. When all the stages are transmitted, the transmission for the whole image is done. The progressive image transmission is a top-down progressive computing process. The information transmitted at every stage is a granulation of the whole image. The first transmitted image is the top granulation that contains the most important information and the finally transmitted image is the lowest granulation. The importance of the information in granulations becomes less as the levels of granulation move lower.

The granulation hierarchy of image can be created in different ways. One way is to create a granulation hierarchy by shape and color [10]. The first granulation is a grey scale image that contains the vague shapes of objects in the image, the second granulation is another grey scale image that gives detailed information about the shapes of objects, and the third granulation gives the color information for the image. Another way to create granulations is by the resolution of an image [3]. The lowest resolution image is the top granulation and the highest resolution image is the lowest granulation. One pixel in a lower resolution represents a few pixels in a higher resolution.

In terms of top-down progressive computing, G_0 is the top granulation that is a grey scale image or a lowest resolution image. P_0 is the granulation G_0 , as the solution is the granulation itself. $G_k = \textit{granulation_refinement}(G_{k-1})$ returns the next low level granulation of G_{k-1} , $P_k = G_k$, and $\textit{fitness}(P_k)$ measures the quality of the image represented by P_k , where $k \geq 1$.

Progressive Approximate Aggregate Query The progressive approximate aggregate query is used in multi-dimensional databases, which may be interpreted as a top-down progressive computing process [4]. The data is organized in a MRA-quad tree. Every leaf node contains statistic information of a record such as the minimum value, the maximum value in this record or the sum of all data in this record, every non-leaf node contains approximate statistic information of all its children. For example, Fig 1 is a MRA-quad tree, nodes 5, 6, 7, 8, and 9 are leaf nodes, and each represents a set of data and it records the exact information about the maximum value, the minimum value and the sum of this set of data. Node 2 is a non-leaf node. It is the union of data in node 5 and node 6, and records the approximate information about the maximum value, the minimum value and sum of the union of data in nodes 5 and 6. If we want

to get some approximate statistics of a group of data such as the union of node 5 and node 6, we can easily get it from the non-leaf node 2, but if we want to get accurate statistics of a group of data, we should iterate all the leaf nodes in these groups.

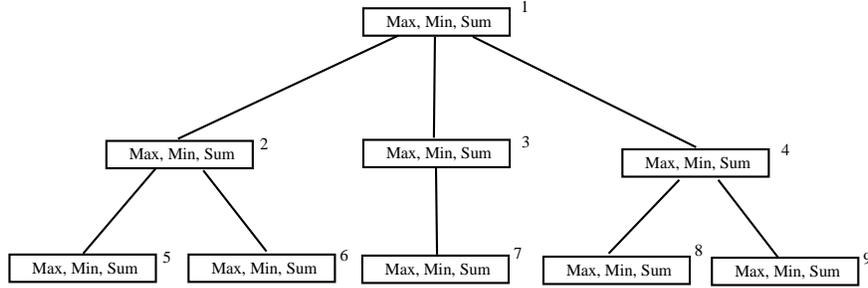


Fig. 1: MRA-quad tree

In terms of top-down progressive computing, G_0 is the top granulation that is the root node of the MRA-quad tree. P_0 is the statistics information recorded by the root node. $G_k = granulation_refinement(G_{k-1})$ returns all the children nodes of nodes in granulation G_{k-1} . $P_k = solution_refinement(P_{k-1}, G_k)$ calculates the statistics information in all the nodes in G_k and gives a resulting statistics information. Function $fitness(P_k)$ gives the accuracy of the statistics information represented by P_k .

Remarks The three examples are from different domains; but they share the same underlying information processing mode, namely, top-down progressive processing. Their unification stems from the fact that a multilevel granular structure is used in each of them. With the introduction of multiple representations, it is possible to compute in a level-wise manner. A crucial issue of top-down progressive computing is therefore the construction of multilevel granular structures.

4 Conclusion

There is an urgent need for developing unique and flexible methods and algorithms by which the philosophy and principles of granular computing can be explicitly and clearly explained. In this paper, we outline a top-down progressive computing model as a mode of granular computing. As a methodology, progressive computing can be used by humans for problem solving; as a computation paradigm, progressive computing can be used by machines for information processing. A basic progressive computing algorithm, BPC, is introduced. Several examples of progressive computing are examined with respect to both human and machine problem solving and information processing.

References

1. Bargiela, A., Pedrycz, W.(Eds.): Human-Centric Information Processing Through Granular Modelling. Springer, Berlin (2009)
2. Flower, L.: Problem-Solving Strategies for Writing. Harcourt Brace Jovabovich, Inc., New York (1981)
3. Ireton, M., Xydeas, C.: A progressive encoding technique for binary images. In: IEE Colloquium on Low Bit Rate Image Coding, 11/1-11/4 (1990)
4. Lazaridis, I., Mehrotra, S.: Progressive approximate aggregate queries with a multi-resolution tree structure. In: Proceedings of the 2001 ACM SIGMOD international conference on Management of data, 401-412 (2001)
5. Ledgard, H., Gueras, J., Nagin, P.: PASCAL with Style: Programming Proverbs. Hayden Book Company, Inc., Rechelle Park, NJ (1979)
6. Lindsay, P.H. and Norman, D.A.: An Introduction to Psychology (2nd edition). Academic Press, New York (1977)
7. Michalskia, R.S., Winstonb, P.H.: Variable precision logic. Artificial Intelligence 29: 121-146 (1986)
8. Knuth, D.E.: Literate programming. The Computer Journal 27: 97-111 (1984)
9. Pedrycz, W., Skowron, A., Kreinovich, V.(Eds.): Handbook of Granular Computing. Wiley Interscience, New York (2008)
10. Tong, F.H., Zhang, D.: A new progressive colour image transmission scheme for the World Wide Web. Computer Networks and ISDN Systems 30: 2059-2064 (1998)
11. Wirth, N.: Program development by stepwise refinement. Communications of the ACM 14: 221-227 (1971)
12. Yao, J.T.: A ten-year review of granular computing. In: Proceedings of the 2007 IEEE International Conference on Granular Computing, 734-739 (2007)
13. Yao, J.T.(Ed.): Novel Developments in Granular Computing, Applications for Advanced Human Reasoning and Soft Computation. Information Science Reference, Herskey, PA (2010)
14. Yao, Y.Y.: Structured writing with granular computing strategies. In: Proceedings of the 2007 IEEE International Conference on Granular Computing, 72-77 (2007)
15. Yao, Y.Y.: A unified framework of granular computing. In: Pedrycz, W., Skowron, A. and Kreinovich, V. (Eds.), Handbook of Granular Computing, Wiley, New York, 401-410 (2008)
16. Yao, Y.Y.: Granular computing: past, present and future. In: Proceedings of the 2008 IEEE International Conference on Granular Computing, 80-85 (2008)
17. Yao, Y.Y.: Integrative levels of granularity. In: Bargiela, A. and Pedrycz, W. (Eds.), Human-Centric Information Processing Through Granular Modelling, Springer, Berlin, 31-47 (2009)
18. Yao, Y.Y.: Human-inspired granular computing. In: Yao, J.T. (Ed.) Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation, Information Science Reference, Herskey, PA, 1-15 (2010)
19. Yao, Y.Y.: Artificial intelligence perspectives on granular computing. In: Pedrycz, W. and Chen, S.-M.(Eds.), Granular Computing and Intelligent Systems Design with Information Granules of Higher Order and Higher Type, Springer, Berlin, 17-34 (2011)
20. Yao, Y.Y., Miao, D.Q., Zhang, N., and Xu, F.F.: Set-theoretic models of granular structures. In: Proceedings of the 5th International Conference on Rough Sets and Knowledge Technology, LNCS(LNAI) 6401, 94-101 (2011)