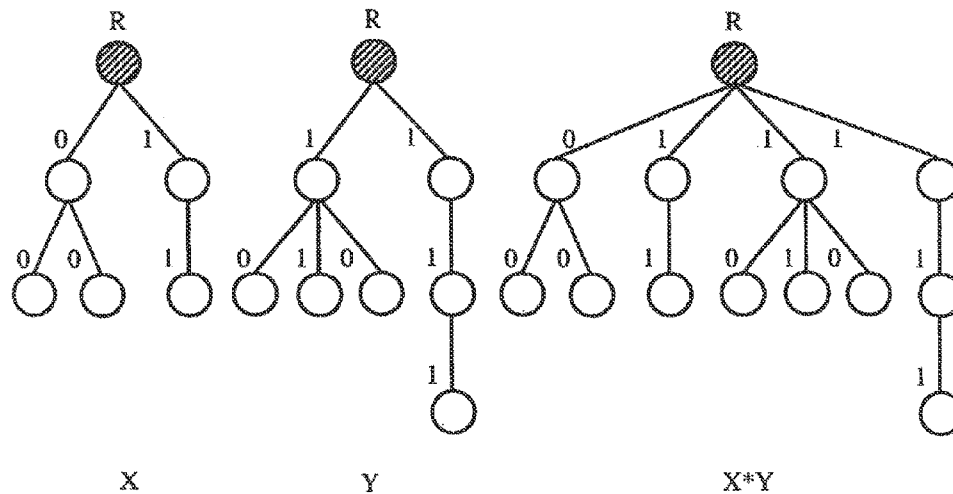


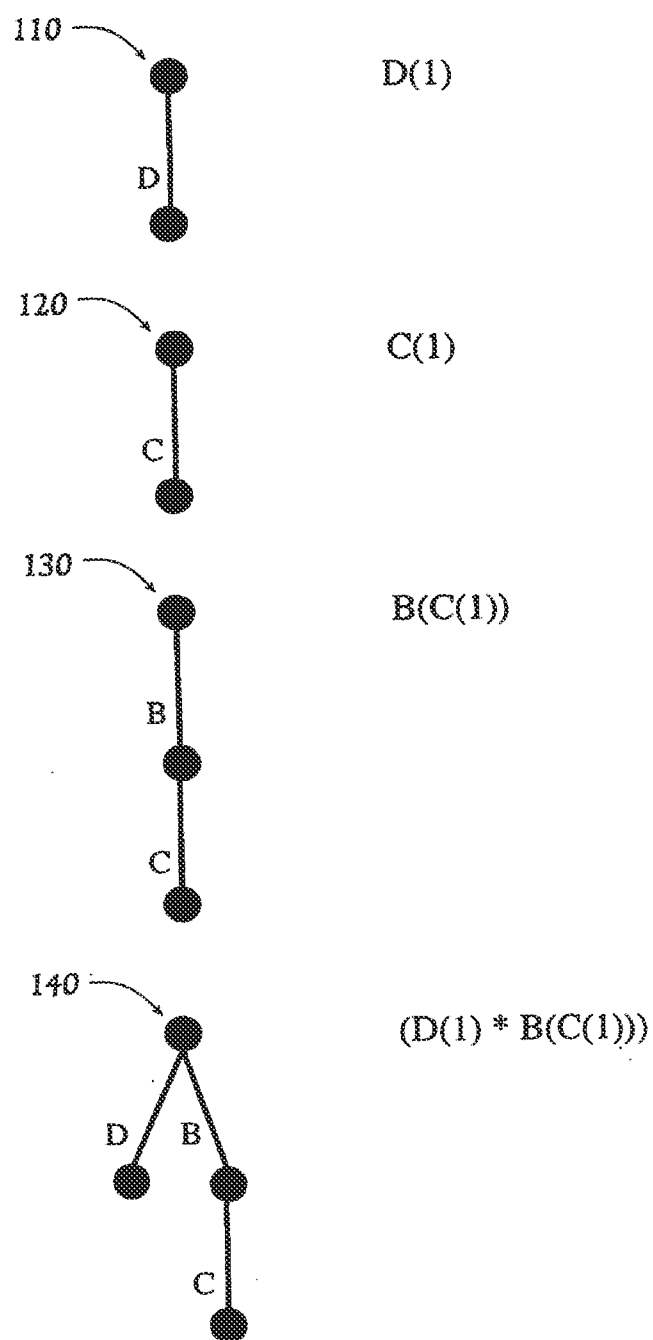


US 20220374447A1

(19) **United States**(12) **Patent Application Publication**
LeTourneau(10) **Pub. No.: US 2022/0374447 A1**(43) **Pub. Date: Nov. 24, 2022**(54) **METHOD AND/OR SYSTEM FOR
MANIPULATING TREE EXPRESSIONS**(71) Applicant: **Robert T. and Virginia T. Jenkins as
Trustees of the Jenkins Family Trust**
Dated Feb.8, 2002, Sacramento, CA
(US)(72) Inventor: **Jack J. LeTourneau**, Ventura, CA (US)(21) Appl. No.: **17/690,576**(22) Filed: **Mar. 9, 2022****Related U.S. Application Data**(63) Continuation of application No. 16/379,674, filed on
Apr. 9, 2019, now Pat. No. 11,314,766, which is a
continuation of application No. 15/250,118, filed on
Aug. 29, 2016, now Pat. No. 10,325,031, which is a
continuation of application No. 14/635,836, filed on
Mar. 2, 2015, now Pat. No. 9,430,512, which is a
continuation of application No. 14/086,837, filed on
Nov. 21, 2013, now Pat. No. 9,043,347, which is a
continuation of application No. 12/578,411, filed on
Oct. 13, 2009, now Pat. No. 8,626,777, which is acontinuation of application No. 11/006,841, filed on
Dec. 7, 2004, now Pat. No. 7,627,591.(60) Provisional application No. 60/623,280, filed on Oct.
29, 2004.**Publication Classification**(51) **Int. Cl.****G06F 16/25** (2006.01)**G06F 16/21** (2006.01)**G06F 16/28** (2006.01)**G06F 16/22** (2006.01)**G06F 16/901** (2006.01)**G06F 16/2453** (2006.01)(52) **U.S. Cl.**CPC **G06F 16/258** (2019.01); **G06F 16/211**
(2019.01); **G06F 16/282** (2019.01); **G06F**
16/2246 (2019.01); **G06F 16/9027** (2019.01);
G06F 16/24535 (2019.01); **Y10S 707/99942**
(2013.01)

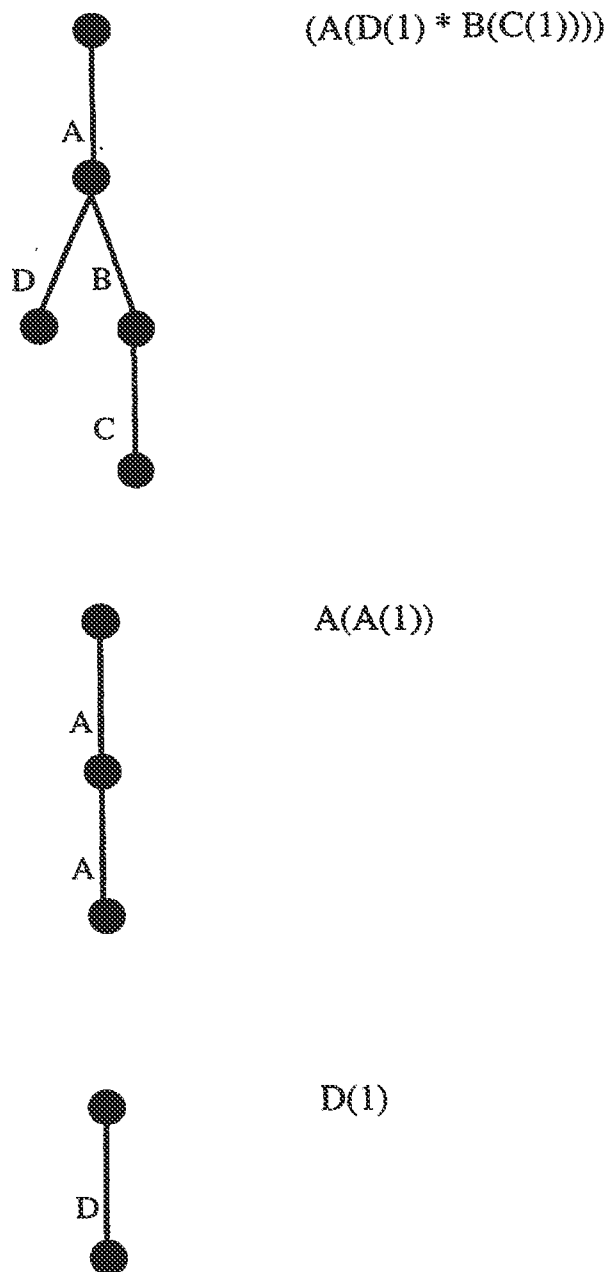
(57)

ABSTRACTEmbodiments of methods and/or systems of manipulating
tree expressions are disclosed.



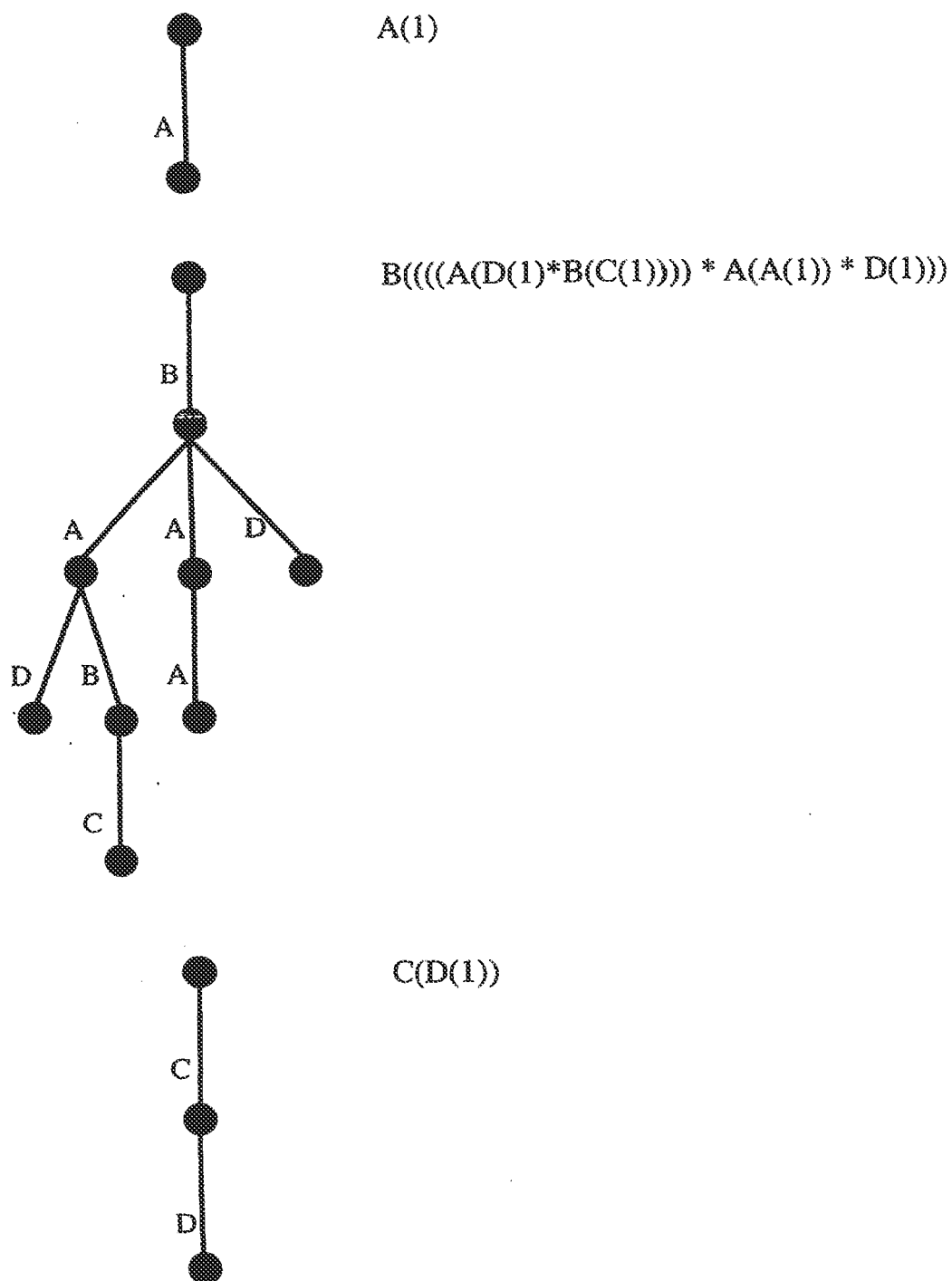
100

Fig. 1



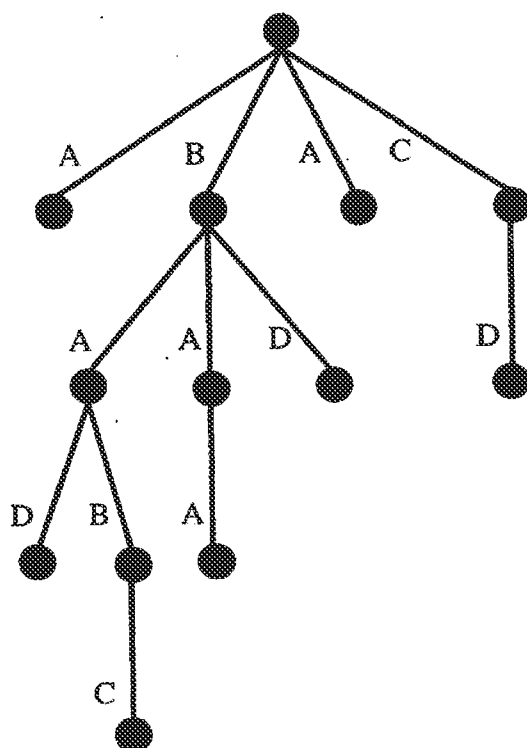
200

Fig. 2

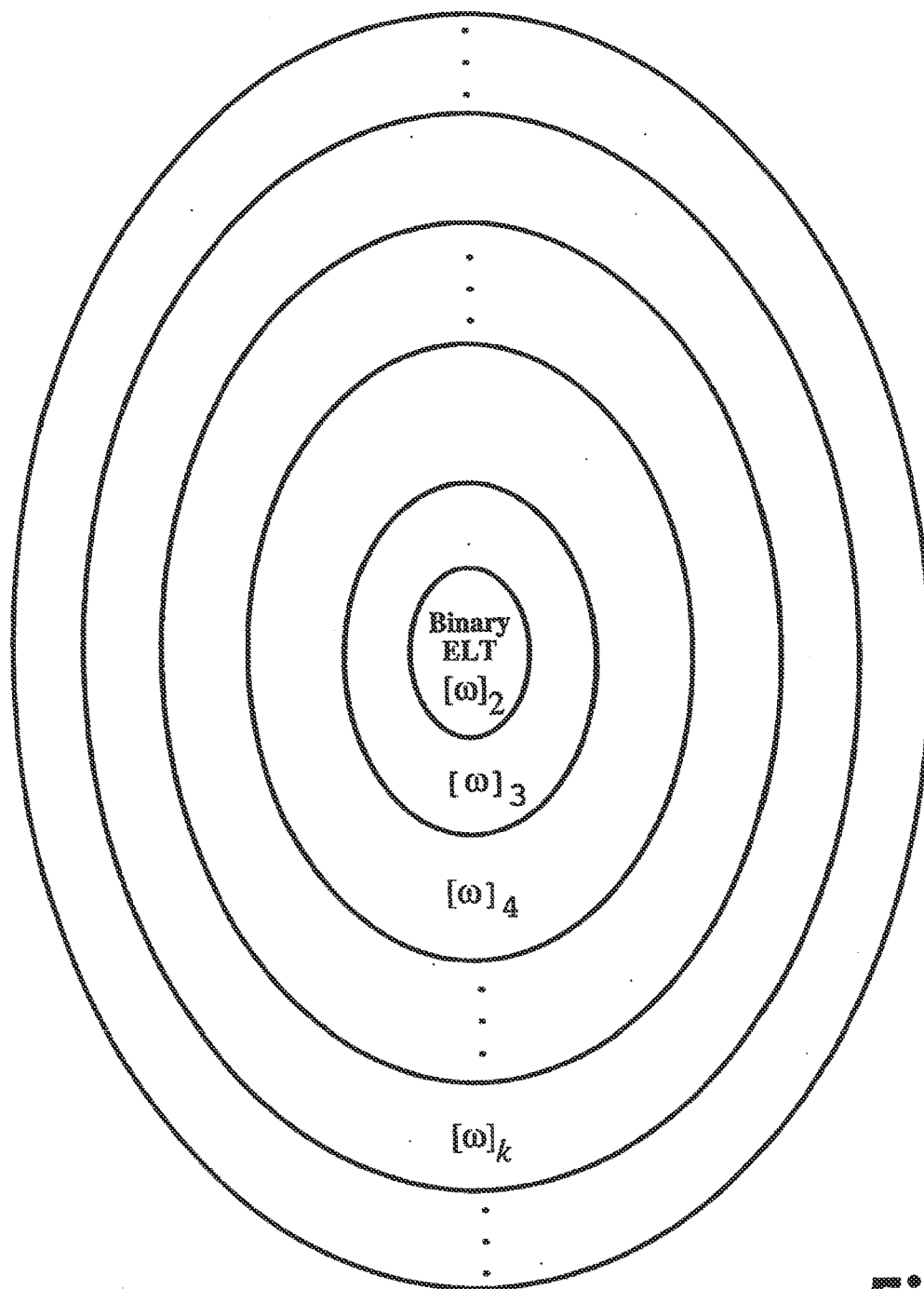


300

Fig. 3



$((A(1) * B(((A(D(1)*B(C(1))))*A(A(1))*D(1)))) * A(1)) * C(D(1)))$



500

Fig. 5

$$\begin{aligned}Q(0) &\Rightarrow 2 \\Q(1) &\Rightarrow 3 \\Q(2) &\Rightarrow 5 \\Q(3) &\Rightarrow 7 \\Q(4) &\Rightarrow 11 \\Q(5) &\Rightarrow 13 \\Q(6) &\Rightarrow 17 \\Q(7) &\Rightarrow 19 \\Q(8) &\Rightarrow 23 \\Q(9) &\Rightarrow 29 \\Q(10) &\Rightarrow 31 \\Q(11) &\Rightarrow 37 \\Q(12) &\Rightarrow 41 \\&\vdots \\&\vdots \\&\vdots\end{aligned}$$

Finite Rooted Unordered 2 Valued Edge-Labeled Trees

Signature = $\langle 2, 2, 1 \rangle$

Alphabet = $\{0, 1, A, B, *\}$

Expressions

$$\begin{aligned} (x * y) &= (y * x); \\ ((x * y) * z) &= (x * (y * z)); \end{aligned}$$

$$(0 * x) = 0;$$

$$(1 * x) = x;$$

$$A(0) = 1;$$

$$B(0) = 1.$$

Core Model Isomorph (based upon the natural number universe)

$$A(x) = Q(2x - 2), x \text{ not } 0;$$

$$B(x) = Q(2x - 1), x \text{ not } 0;$$

$$(x * y) = (x \text{ multiplied by } y).$$

Finite Rooted Unordered 3 Valued Edge-Labeled Trees

Signature = $\langle 2, 3, 1 \rangle$
Alphabet = $\{0, 1, A, B, C, *\}$

Expressions

$$(x * y) = (y * x);$$

$$((x * y) * z) = (x * (y * z));$$

$$(0 * x) = 0;$$

$$(1 * x) = x;$$

$$A(0) = 1;$$

$$B(0) = 1;$$

$$C(0) = 1.$$

Core Model Isomorph (based upon the natural number universe)

$$A(x) = Q(3x - 3), x \text{ not } 0;$$

$$B(x) = Q(3x - 2), x \text{ not } 0;$$

$$C(x) = Q(3x - 1), x \text{ not } 0;$$

$$(x * y) = (x \text{ multiplied by } y).$$

Finite Rooted Unordered 4 Valued Edge-Labeled Trees

Signature = $\langle 2, 4, 1 \rangle$

Alphabet = $\{0, 1, A, B, C, D, *\}$

Expressions

$$\begin{array}{l} 910 \rightarrow (x * y) = (y * x); \\ 920 \rightarrow ((x * y) * z) = (x * (y * z)); \end{array}$$

$$930 \rightarrow \begin{cases} (0 * x) = 0; \\ (1 * x) = x; \end{cases}$$

$$940 \rightarrow \begin{cases} A(0) = 1; \\ B(0) = 1; \\ C(0) = 1; \\ D(0) = 1. \end{cases}$$

Core Model Isomorph (based upon the natural number universe)

$$A(x) = Q(4x - 4), x \text{ not } 0;$$

$$B(x) = Q(4x - 3), x \text{ not } 0;$$

$$C(x) = Q(4x - 2), x \text{ not } 0;$$

$$D(x) = Q(4x - 1), x \text{ not } 0;$$

$$(x * y) = (x \text{ multiplied by } y).$$

900

Fig. 9

Finite Rooted Unordered 5 Valued Edge-Labeled Trees

Signature = $\langle 2, 5, 1 \rangle$

Alphabet = $\{0, 1, A, B, C, D, E\}$

Expressions

$$(x * y) = (y * x);$$

$$((x * y) * z) = (x * (y * z));$$

$$(0 * x) = 0;$$

$$(1 * x) = x;$$

$$A(0) = 1;$$

$$B(0) = 1;$$

$$C(0) = 1;$$

$$D(0) = 1;$$

$$E(0) = 1.$$

Core Model Isomorph (based upon the natural number universe)

$$A(x) = Q(5x - 5), x \text{ not } 0;$$

$$B(x) = Q(5x - 4), x \text{ not } 0;$$

$$C(x) = Q(5x - 3), x \text{ not } 0;$$

$$D(x) = Q(5x - 2), x \text{ not } 0;$$

$$E(x) = Q(5x - 1), x \text{ not } 0;$$

$$(x * y) = (x \text{ multiplied by } y).$$

Finite Rooted Unordered 6 Valued Edge-Labeled Trees

Signature = $\langle 2, 6, 1 \rangle$

Alphabet = $\{0, 1, A, B, C, D, E, F * \}$

Expressions

$$(x * y) = (y * x);$$

$$((x * y) * z) = (x * (y * z));$$

$$(0 * x) = 0;$$

$$(1 * x) = x;$$

$$A(0) = 1;$$

$$B(0) = 1;$$

$$C(0) = 1;$$

$$D(0) = 1;$$

$$E(0) = 1;$$

$$F(0) = 1.$$

Core Model Isomorph (based upon the natural number universe)

$$A(x) = Q(6x - 6), x \text{ not } 0;$$

$$B(x) = Q(6x - 5), x \text{ not } 0;$$

$$C(x) = Q(6x - 4), x \text{ not } 0;$$

$$D(x) = Q(6x - 3), x \text{ not } 0;$$

$$E(x) = Q(6x - 2), x \text{ not } 0;$$

$$F(x) = Q(6x - 1), x \text{ not } 0;$$

$$(x * y) = (x \text{ multiplied by } y).$$

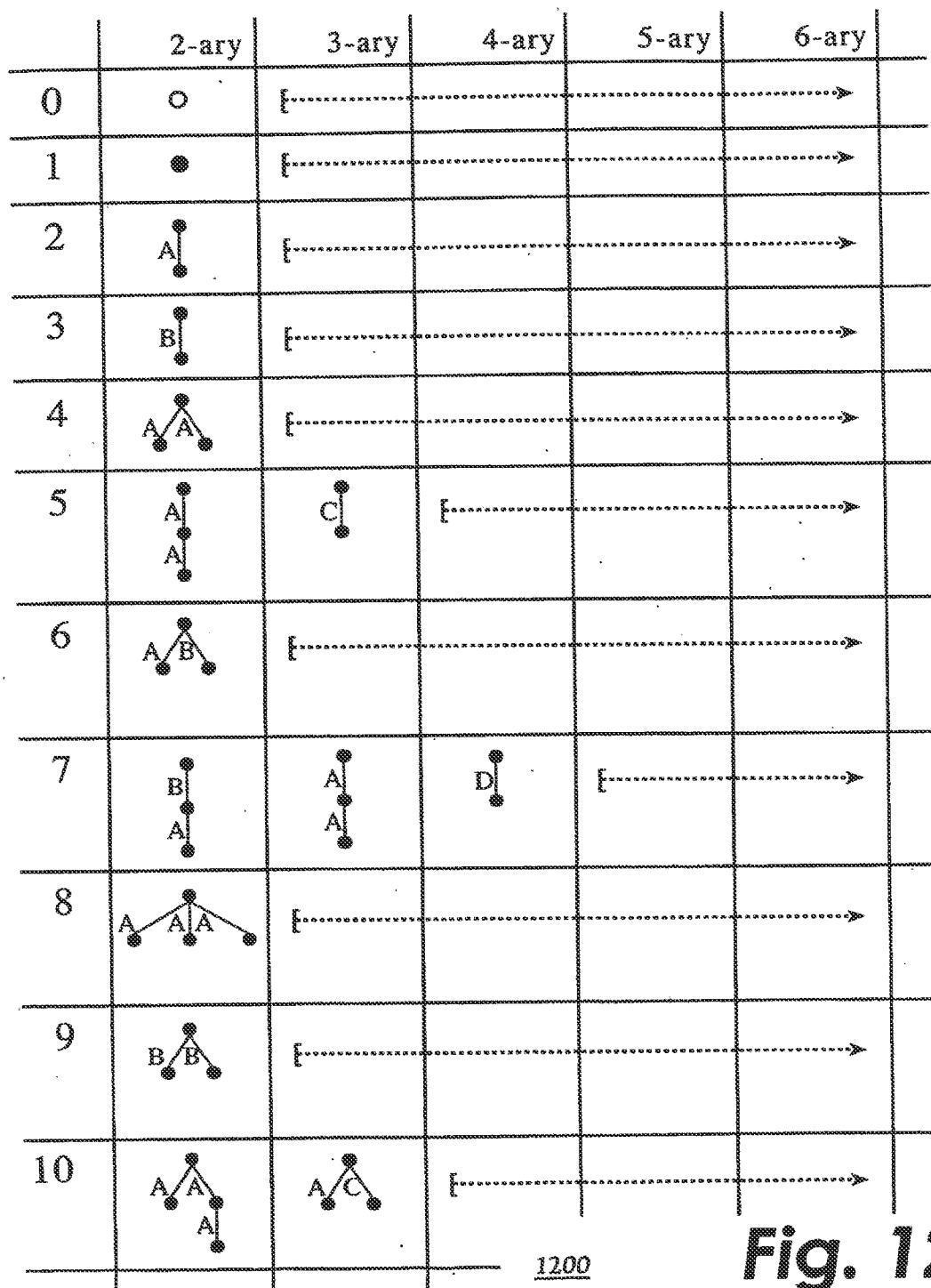
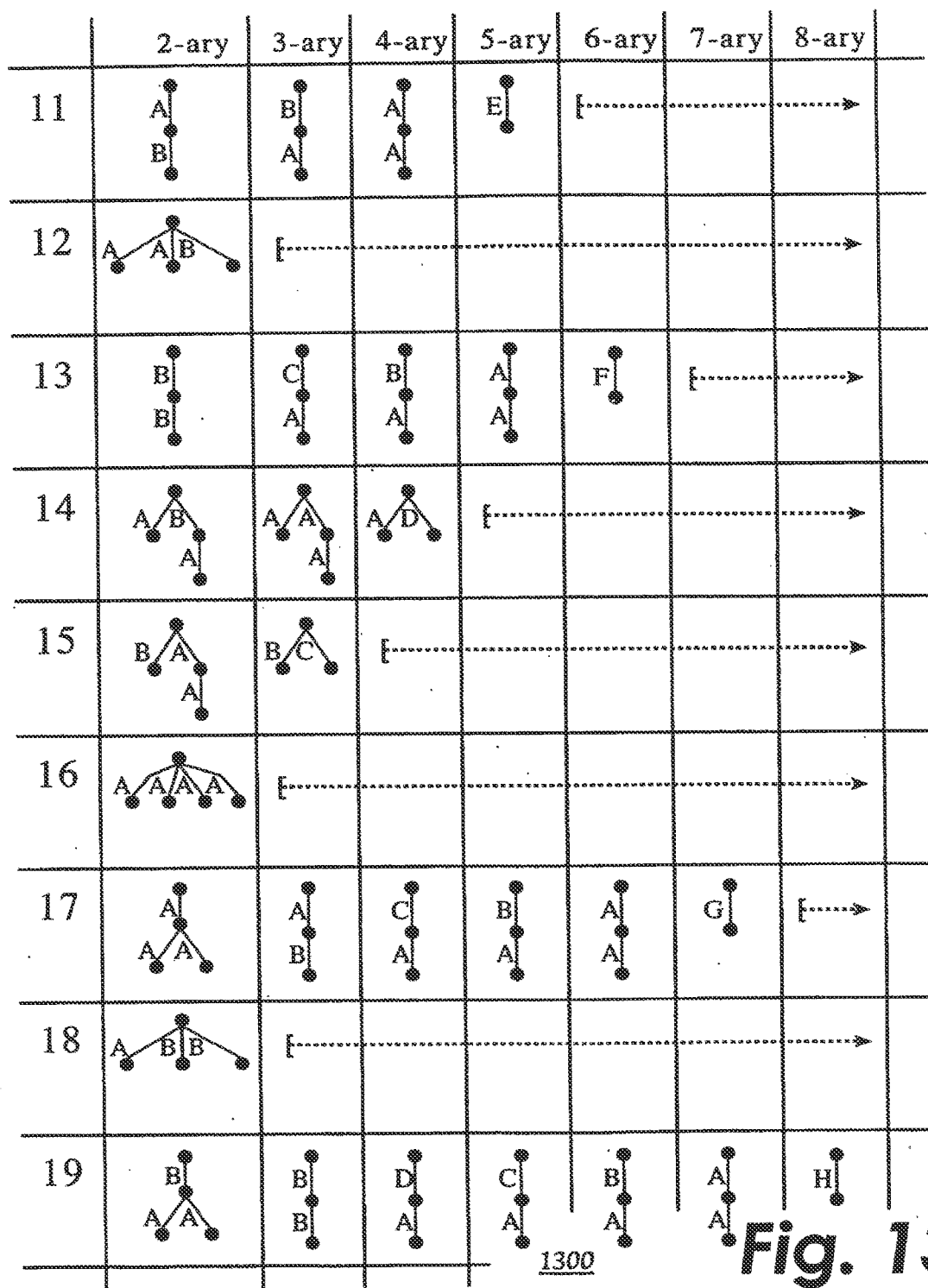
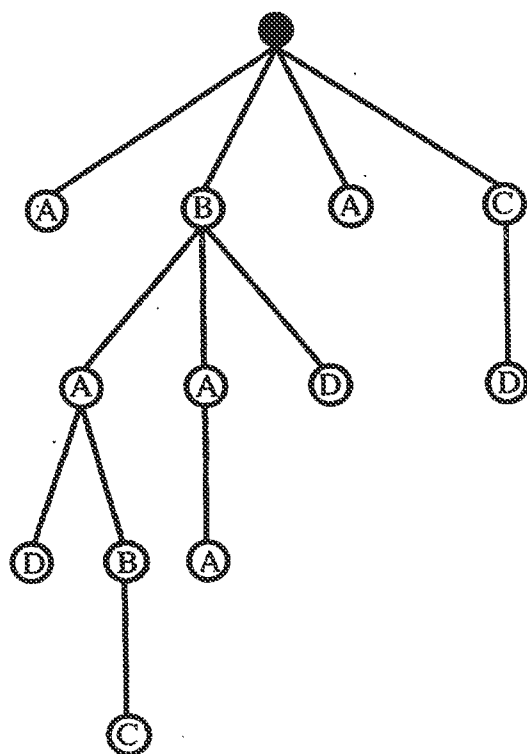


Fig. 12



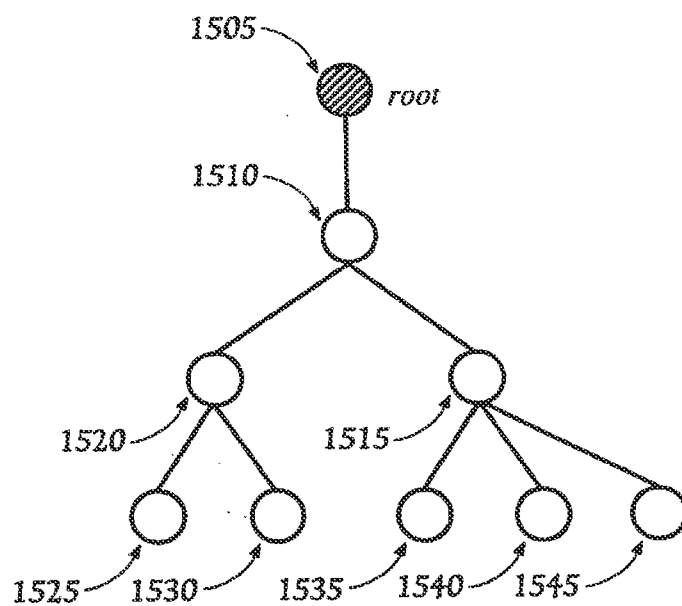
1300

Fig. 13



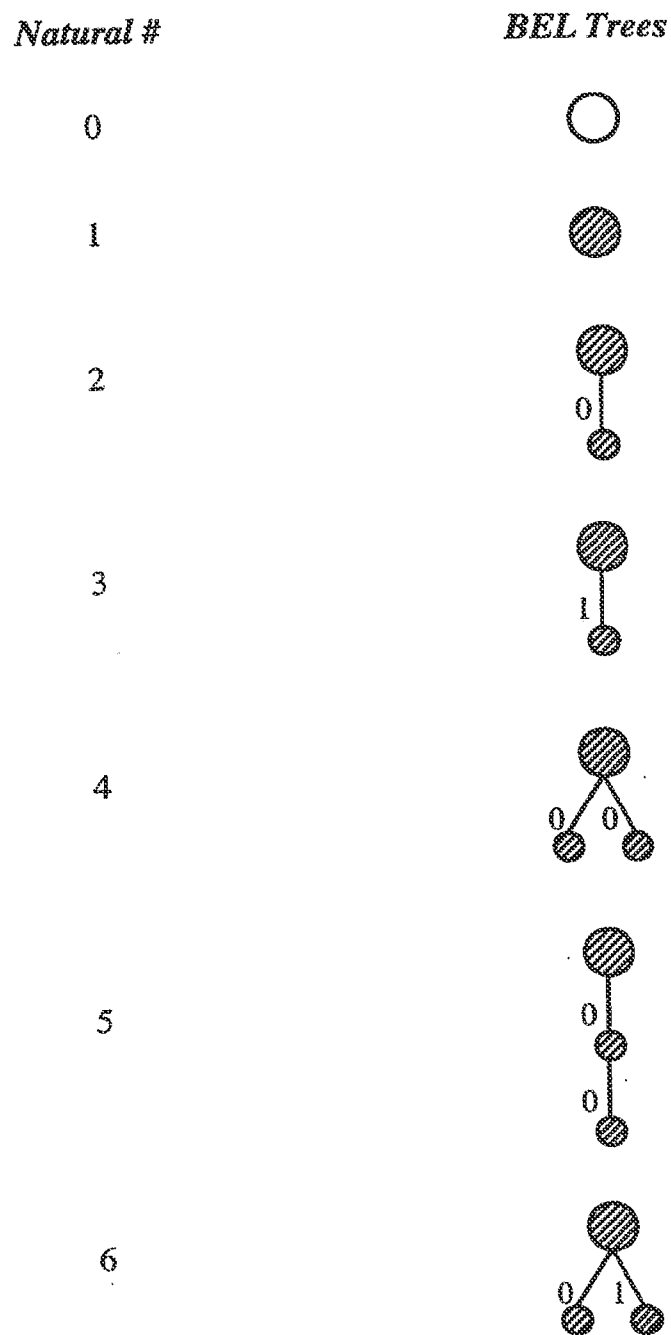
1400

Fig. 14



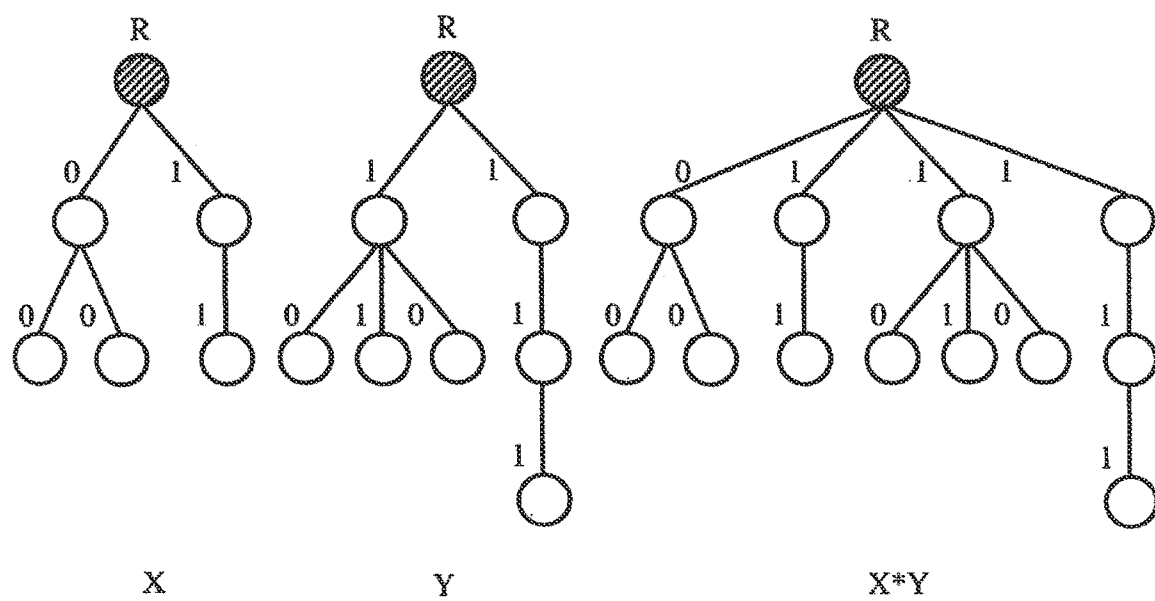
1500

Fig. 15



1600

Fig. 16



1700

Fig. 17

METHOD AND/OR SYSTEM FOR MANIPULATING TREE EXPRESSIONS

[0001] This disclosure claims priority pursuant to 35 USC 119(e) from U.S. provisional patent application Ser. No. 60/623,280, filed on Oct. 29, 2004, by LeToumeau, titled, “METHOD AND/OR SYSTEM FOR MANIPULATING TREE EXPRESSIONS,” assigned to the assignee of the presently claimed subject matter.

BACKGROUND

[0002] This disclosure is related to manipulating tree expressions

[0003] In a variety of fields, data or a set of data, may be represented in a hierarchical fashion. This form of representation may, for example, convey information, such as particular relationships or patterns between particular pieces of data or groups of data and the like. However, manipulating and/or even recognizing specific data representations or patterns is not straight-forward, particularly where the data is arranged in a complex hierarchy. Without loss of generality, examples may include a database, and further, without limitation, a relational database. Techniques for performing operations on such databases or recognizing specific patterns, for example, are computationally complex, time consuming, and/or otherwise cumbersome. A need, therefore, continues to exist for improved techniques for performing such operations and/or recognizing such patterns.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Subject matter is particularly pointed out and distinctly claimed in the concluding portion of the specification. The claimed subject matter, however, both as to organization and method of operation, together with objects, features, and advantages thereof, may best be understood by reference of the following detailed description when read with the accompanying drawings in which:

[0005] FIG. 1 is a schematic diagram of embodiments of several unordered edge-labeled trees and symbolic expressions mathematically representing the tree embodiments;

[0006] FIG. 2 is a schematic diagram of embodiments of several unordered edge-labeled trees and symbolic expressions mathematically representing the tree embodiments;

[0007] FIG. 3 is a schematic diagram of embodiments of several unordered edge-labeled trees and symbolic expressions mathematically representing the tree embodiments;

[0008] FIG. 4 is a schematic diagram illustrating an embodiment of an unordered edge-labeled tree and a symbolic expression mathematically representing the tree embodiment;

[0009] FIG. 5 is a schematic diagram illustrating all N-valued rooted, unordered edge-labeled trees, where N is a natural numeral greater than or equal to 2;

[0010] FIG. 6 is a table providing an embodiment of a function that relates natural numerals to composite numerals;

[0011] FIG. 7 is a table providing symbolic expressions for an embodiment of rooted, unordered, 2-valued edge-labeled trees;

[0012] FIG. 8 is a table providing symbolic expressions for an embodiment of rooted, unordered, 3-valued edge-labeled trees;

[0013] FIG. 9 is a table providing symbolic expressions for an embodiment of rooted, unordered, 4-valued edge-labeled trees;

[0014] FIG. 10 is a table providing symbolic expressions for an embodiment of rooted, unordered, 5-valued edge-labeled trees;

[0015] FIG. 11 is a table providing symbolic expressions for an embodiment of rooted, unordered, 6-valued edge-labeled trees;

[0016] FIGS. 12 and 13 provide an embodiment of a table relating the natural numerals and embodiments of different tree views;

[0017] FIG. 14 is a schematic diagram of an unordered node labeled tree;

[0018] FIG. 15 is a schematic diagram of an unordered edge labeled tree;

[0019] FIG. 16 is a table illustrating an embodiment of a relationship between BELTs and natural numerals; and

[0020] FIG. 17 is a schematic diagram illustrating an embodiment of merging two edge-labeled trees.

DETAILED DESCRIPTION

[0021] In the following detailed description, numerous specific details are set forth to provide a thorough understanding of the claimed subject matter. However, it will be understood by those skilled in the art that the claimed subject matter may be practiced without these specific details. In other instances, well-known methods, procedures, components and/or circuits have not been described in detail so as not to obscure the claimed subject matter.

[0022] Some portions of the detailed description which follow are presented in terms of algorithms and/or symbolic representations of operations on data bits or binary digital signals stored within a computing system, such as within a computer or computing system memory. These algorithmic descriptions and/or representations are the techniques used by those of ordinary skill in the data processing arts to convey the substance of their work to others skilled in the art. An algorithm is here, and generally, considered to be a self-consistent sequence of operations and/or similar processing leading to a desired result. The operations and/or processing involve physical manipulations of physical quantities. Typically, although not necessarily, these quantities may take the form of electrical and/or magnetic signals capable of being stored, transferred, combined, compared and/or otherwise manipulated. It has proven convenient, at times, principally for reasons of common usage, to refer to these signals as bits, data, values, elements, symbols, characters, terms, numbers, numerals and/or the like. It should be understood, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels. Unless specifically stated otherwise, as apparent from the following discussion, it is appreciated that throughout this specification discussions utilizing terms such as “processing”, “computing”, “calculating”, “determining” and/or the like refer to the actions and/or processes of a computing platform, such as a computer or a similar electronic computing device, that manipulates and/or transforms data represented as physical electronic and/or magnetic quantities and/or other physical quantities within the computing platform’s processors, memories, registers, and/or other information storage, transmission, and/or display devices.

[0023] In a variety of fields, data or a set of data, may be represented in a hierarchical fashion. This form of representation may, for example, convey information, such as particular relationships or patterns between particular pieces of data or groups of data and the like. However, manipulating and/or even recognizing specific data representations or patterns is not straight-forward, particularly where the data is arranged in a complex hierarchy. Without loss of generality, examples may include a database and further, without limitation, a relational database. Techniques for performing operations on such databases or recognizing specific patterns, for example, are computationally complex, time consuming, and/or otherwise cumbersome. A need, therefore, continues to exist for improved techniques for performing such operations and/or recognizing such patterns.

[0024] As previously discussed, in a variety of fields, it is convenient and/or desirable to represent data, a set of data and/or other information in a hierarchical fashion. In this context, such a hierarchy of data shall be referred to as a “tree.” In a particular embodiment, a tree may comprise a finite, rooted, connected, acyclic graph. Likewise, such trees may be either ordered or unordered. Here, ordered refers to the notion that there is an ordering or precedence among nodes attached to a common node corresponding to the order of the attached nodes shown in a graphical illustration. An unordered tree is illustrated here, for example, in FIG. 15 by embodiment 100. As illustrated, the root of this particular embodiment encompasses node 105. In addition to 105, there are eight other nodes designated 110 to 145, respectively. Likewise, the nodes are connected by branches referred to, in this context, as edges. Thus, the nodes of this tree are connected by eight edges. This embodiment, therefore, illustrates a finite tree that is rooted by node 105. Furthermore, the nodes are connected, meaning, in this context, that a path exists between any two nodes of the tree. The tree is likewise acyclic, meaning here, that no path in the tree forms a complete loop.

[0025] As previously suggested, in a variety of contexts, it may be convenient and/or desirable to represent a hierarchy of data and/or other information using a structure, such as the embodiment illustrated in FIG. 15. One particular embodiment, without loss of generality, of a tree may include edges that are labeled with data and/or other values. Likewise, in one particular embodiment, such data and/or values may be limited to a particular set of data. For example, in this context, a binary edge labeled tree refers to a tree in which the data and/or values comprise binary data, that is, in this example, either a binary one or a binary zero. Likewise, alternatively, the edges of a tree may be labeled with three values, such as 0, 1, 2. Continuing, the edges may be labeled with four values, five values, etc. In this context, the class of all trees in which the edges are labeled with a specific number of distinct values, that is, in this context, values chosen from a set having a specific number of distinct elements, shall be referred to as edge-labeled trees (ELTs). It is likewise noted that such trees are not limited to being labeled with the numerals previously described. Any distinctly identifiable labels may be employed; however, in this context, it shall be understood that employing numerals to label the edges is sufficiently general to encompass any sort of data labels that may be desirable, regardless of their form.

[0026] To reiterate, in this context, a tree comprises an edge labeled tree if each edge of the string or tree respec-

tively stores a value or single piece of data. Likewise, in this context, two nodes are employed to support an edge holding a single piece of data. At this point, it is worth noting that trees having nodes and edges, such as previously described, may be represented in a computing platform or similar computing device through a data structure or a similar mechanism intended to capture the hierarchical relationship of the data, for example. It is intended that all such embodiments are included within the scope of the claimed subject matter.

[0027] It is noted that binary edge labeled trees (BELTs) may be listed or enumerated. See, for example, U.S. provisional patent application Ser. No. 60/543,371, titled “Manipulating Sets of Hierarchical Data,” filed on Feb. 8, 2004, by J.J. LeTourneau, and assigned to the assignee of the current provisional application. This is illustrated, here, for example, in FIG. 16. It is noted that this particular figure also includes the associated natural numerals. The association of such numerals for this particular embodiment should be clear based at least in part on previously cited U.S. provisional patent application Ser. No. 60/543,371. However, it is, of course, again noted that the claimed subject matter is not limited in scope to employing the approach or approaches described in aforementioned U.S. provisional patent application Ser. No. 60/543,371, U.S. provisional patent application Ser. No. 60/543,371 is provided simply as an example of listing or enumerating unordered BELTs.

[0028] However, for this particular embodiment, although the claimed subject matter is not limited in scope in this respect, a method of enumerating a set of unordered trees may begin with enumeration of an empty binary edge labeled tree and a one node binary edge labeled tree. Thus, the empty tree is associated with the natural numeral zero and has a symbolic representation as illustrated in FIG. 16 (circle). Likewise, the one node tree, which holds no data, is associated with the natural numeral one and has a graphical representation of a single node. For higher positive natural numbers, ordered trees may be generated by a process described, for example, in “The Lexicographic Generation of Ordered Trees,” by S. Zaks, *The Journal of Theoretical Computer Science*, Vol. 10(1), pp63-82, 1980, or, “Enumerating Ordered Trees Lexicographically,” by M.C. Er, *Computation Journal*, Vol. 28, Issue 5, pp 538-542, 1985. This may be illustrated, for example in FIG. 16, as described in more detail below.

[0029] As illustrated, for this particular embodiment, and as previously described, the empty tree has zero nodes and is associated with the natural numeral zero. Likewise, the one node tree root comprises a single node and is associated with the natural numeral one. Thus, to obtain the tree at position two, a root node is attached and connected to the prior root node by an edge. Likewise, here, by convention, the edge is labeled with a binary zero. If, however, the tree formed by the immediately preceding approach were present in the prior enumeration of trees, then a similar process embodiment is followed, but, instead, the new edge is labeled with a binary one rather than a binary zero. Thus, for example, to obtain the binary edge labeled tree for position three, a new root node is connected to the root node by an edge and that edge is labeled with a binary one.

[0030] Continuing with this example, to obtain the binary edge labeled tree for position four, observe that numeral four is the product of numeral two times numeral two. Thus, a union is formed at the root of two trees, where, here, each

of those trees is associated with the positive natural numeral two. Likewise, to obtain the binary edge labeled tree for position five, begin with the binary edge labeled tree for position two and follow the previously articulated approach of adding a root and an edge and labeling it with a binary zero.

[0031] In this context, adding a root node and an edge and labeling it binary zero is referred to as a “zero-push” operation and adding a root node and an edge and labeling it binary one is referred to as a “one-push” operation. Thus, referring again to FIG. 16, the one-push of the root tree is the tree at position three. This follows from FIG. 9 of previously referenced U.S. provisional patent application Ser. No. 60/543,371, since $Q((1*2)-1)=Q(1)=3$. Likewise, the tree at position five is the zero-push of the tree at position 2. Again, this follows from FIG. 9 of the previously referenced US provisional patent application, since $Q((2*2)-2)=Q(2)=5$.

[0032] In the embodiment just described, binary edge labeled trees use binary numerals “0” and “1.” However, the claimed subject matter is not limited in scope to binary edge labeled trees. For example, trees may employ any number of numeral combinations as labels, such as triplets, quadruplet, etc. Thus, using a quadruplet example, it is possible to construct trees, such as a zero-push of a particular tree, a one-push of that tree, a two-push of that tree, and a three-push of that tree. Thus, for such trees, edges may be labeled 0, 1, 2 or 3, etc., as previously described and as explained in more detail hereinafter.

[0033] The foregoing discussion has begun to characterize an algebra involving trees, in this particular embodiment, an algebra for unordered edge labeled trees or unordered ELTs, such as BELTs. The foregoing discussion defines a value zero, a zero node tree for this particular embodiment, a value one, a one node tree for this particular embodiment, and a monadic operation, previously described as zero-push. For example, alternatively, a “one-push” may be employed. For this embodiment, this is analogous, for example, to the convention that “0” represent “off” and “1” represent “on.” Alternatively and equivalently, “1” may be employed to represent “off,” and “0” may be employed to represent “on,” without loss of generality. For this particular embodiment, an additional operation may be characterized, a “merger” operation. The merger operation with respect to trees refers to merging two trees at their roots. This operation is illustrated, for example, in FIG. 17.

[0034] As will now be appreciated, the merger operation comprises a binary operator. Likewise, the constants zero/one, referred to above, may be viewed as an operation having no argument or as a zero valued argument operator or operation. Thus, this operation, in effect, returns the same value whenever applied. Here, for this particular embodiment, the constant value, or zero valued argument operation that returns “c” and is denoted as “c.” The merger operator is denoted as “*.”

[0035] FIG. 4 is schematic diagram illustrating an embodiment of an edge labeled tree, here a 4 valued edge labeled tree. In this particular embodiment, four distinct values pre employed to label the edges. Here, the labels comprising A, B, C and D, although, of course, the claimed subject matter is not limited to 4 valued edge labeled trees, to edge labeled trees, or to employing these particular edge labels. It is noted that the labels A, B, C, and D in this embodiment are similar to the labels binary 0 and binary 1 for BELTs. Below tree 400 is a symbolic expression mathematically representing

tree 400. Performing the operations indicated by the expression shown in FIG. 4 below tree 400 will provide a natural numeral that corresponds, for this particular embodiment, to this particular tree, as described in more detail hereinafter.

[0036] To assist in understanding the relationship between the symbolic expression shown in FIG. 4 and tree 400, for this particular embodiment, FIG. 1 provides an embodiment 110 of another tree. As illustrated, tree 110 comprises an edge label D connecting two nodes. For this particular context, this embodiment may be expressed symbolically as follows: D(1). Thus, a technique to describe the embodiment of tree 110 would refer to the “push” of the natural number 1. Here, for this particular embodiment, this particular push operation comprises the “D” push of 1, resulting in D being the label of the edge connecting the two nodes. More specifically, as previously described, a single node comprises the natural numeral 1 in this particular embodiment. To perform a push operation, an edge is attached to that node and labeled. Here, applying a D push, the label provided comprises the label D.

[0037] Continuing, the “C” push of “1” is illustrated as two nodes with an edge labeled C connecting the two nodes for tree embodiment 120. Applying similar reasoning provides an edge labeled tree embodiment 130 representing the following expression: B(C(1)). Likewise, for this particular embodiment, the operation of merger may be represented as “*”, as previously suggested. Thus, applying a merger operation provides tree embodiment 140 at the bottom of FIG. 1 corresponding, for this particular embodiment, to the following expression: (D(1)*B(C(1))). Applying similar reasoning to FIGS. 2 and 3 and the tree embodiments shown ultimately produces tree 400 illustrated in FIG. 4, along with the corresponding symbolic expression.

[0038] As the previous discussion suggests, here A, B, C and D comprise monadic operators and the merger operation comprises a binary operation. In U.S. provisional patent application No. 60/575,784, titled “Method and/or System for Simplifying Tree Expressions, such as for Pattern Matching,” filed May 28, 2004, by J.J. LeTourneau, assigned to the assignee of the current application, monadic operators similar to those described here were designed as successor operators, using the symbol S(x). Here, these monadic operators comprise multiple successive operators.

[0039] Previously, an embodiment for manipulating binary edge labeled trees or BELTs was described in connection with U.S. provisional patent application 60/543,371. In that context, binary edge labeled trees comprise finite rooted, unordered two valued edge labeled trees. Thus, for the particular embodiment of binary edge labeled trees described, the two values comprise “0” and “1,” although alternately they could comprise A and B, for example, or any other two values. Referring now to FIG. 5, a Venn diagram 500 is illustrated providing the set of all edge labeled trees, structured or organized in a particular manner here. In the center of the diagram, binary or two valued edge labeled trees are depicted as a subset. Furthermore, as illustrated, two valued edge labeled trees are also depicted as a subclass or subset of three valued edge labeled trees. Likewise, three valued edge labeled trees are depicted as a subclass or subset of four valued edge labeled trees and so forth. Thus, depending at least in part on the particular set of distinct values employed to label the edges, an edge labeled tree that employs two distinct values may comprise an example of a three valued edge labeled tree in which one of the values is

specifically not employed in the particular tree. As shall be explained in more detail hereinafter, this raises a question regarding proper interpretation of the data that the tree may represent or store. More specifically, an identical tree may represent different data depending at least in part on whether the tree is “viewed” as, to continue with this example, a two valued edge labeled tree or a three valued edge labeled tree. Thus, in this embodiment, we refer to this as the “view” of the particular tree. For example, a two valued edge labeled tree is referred to as view 2 and a three valued edge labeled tree is referred to as view 3, although, for example, the particular tree may not contain three different values. The view in this embodiment refers to the set of distinct values from which the labels may be selected, as previously described. FIG. 5 therefore depicts the set of all edge labeled trees as the union of all such edge labeled trees in which the edge values are selected from a set having a specific number of distinct values.

[0040] Previously in U.S. provisional application 60/543,371, an embodiment was demonstrated in which an association existed between natural numerals and binary edge labeled trees. For this particular embodiment, similar associations also exist, here between any N valued edge labeled tree and the natural numerals, where N is a numeral. Of course, many different associations are possible and the claimed subject matter is intended to cover all such associations regardless of the particular embodiment. Thus, for example, three valued edge label trees may be converted to numerals, four valued edge labeled trees may be converted to numerals and so forth. Thus, manipulations, such as those previously described, for example, in aforementioned provisional U.S. patent application Ser. No. 60/543,371, as well as additional manipulations, may be applied to N valued edge labeled trees, as described in more detail hereinafter.

[0041] As suggested in previously referenced U.S. provisional application 60/543,371, in one particular embodiment, when converting between binary edge labeled trees and numerals, a relationship was found to exist between a “push” operation and non-composite numerals. Thus, in this context, it may be convenient to define an operation indexed by the natural numerals that provides in ascending order the non-composite numerals, although, of course, the claimed subject matter is not limited in scope in this respect. Such an operation is depicted specifically in FIG. 6. As described in more detail hereinafter, this operation is also convenient in this context in connection with edge labeled trees of any number of distinct edge values.

[0042] As previously suggested in aforementioned U.S. provisional patent application NO. 60/575,784, a set of congruence operations on the set of tree expressions may be isomorphic to the set of finite, rooted, unordered binary edge labeled trees. Thus, or more particularly, under such an isomorphism, in the particular embodiment, a one-to-one relationship between the equivalence classes that satisfy the tree expressions and the finite, rooted, ordered binary edge labeled trees may exist.

[0043] Thus, beginning with binary edge labeled trees or finite rooted unordered two valued edge labeled trees, but continuing to edge labeled trees of higher numbers of values, such as, for example, three valued edge labeled trees, four valued edge labeled trees and so forth, for this embodiment, a similar association or relationship between tree expressions and edge labeled trees may be constructed. Thus, for this embodiment, with a set of operations that satisfies a set

of tree expressions an isomorphism with a set of finite rooted unordered N valued edge labeled trees results in a similar one to one relationship between the equivalence classes that satisfy the tree expressions and the edge labeled trees themselves. Likewise, by demonstrating that these tree expressions are also isomorphic with natural numerals, tree manipulations are able to be constructed for edge labeled trees using natural numerals, as had similarly been done for binary edge labeled trees. Thus, as shall become more clear hereinafter, manipulating tree expressions is isomorphic to manipulating numerals for this particular embodiment.

[0044] Previously, binary edge labeled trees have been discussed. For purposes of illustration, it shall instructive to now discuss another type of edge labeled tree, such as four valued edge labeled trees. For example, FIG. 9 corresponds to finite rooted unordered four valued edge labeled trees, which were also discussed previously with respect to FIG. 4. Thus, for this example embodiment, an algebra may be constructed that is isomorphic to the natural numerals for such four valued edge labeled trees. Likewise, similar algebras may be constructed by use a similar set of tree expressions, as shown, for example, by FIGS. 7, 8, 10 and 11. The similarity of these expressions allows us to write a schema or generalized description and thereby cover all such similar algebras.

[0045] Thus, similar to an approach previously described, FIG. 9 provides a set of constants and operators here, constants 0 and 1, monadic operators A, B, C, and D and binary operator *. Thus, we designate this algebra with the signature $\langle 2, 4, 1 \rangle$ as a result. The expressions for this particular embodiment are provided in FIG. 9. The first expression, 910, denotes commutativity and the second expression, 920, denotes associativity. Likewise, the next two expressions, 930, define the relationship of the merger of the constants with any other value. The next four expressions, 940, define the monadic operators A, B, C and D.

[0046] Thus, for this embodiment, these expressions therefore define a set of edge labeled trees with particular properties. Specifically, the properties are isomorphic to the natural numerals. Thus, as shall be demonstrated further, for this embodiment, four valued edge labeled trees, for example, may be manipulated using natural numerals.

[0047] At least in part because natural numerals are isomorphic to N valued trees, a way to depict this relationship for this embodiment is illustrated by FIGS. 12 and 13. Previously, a particular view for a particular edge labeled tree, for this embodiment, was discussed. In these figures, each column represents a different potential view for a set of edge labeled trees for this embodiment. Likewise, each row provides the edge labeled tree in the view corresponding to the column for the natural number on the left-hand side of FIG. 12.

[0048] For example, column one shows the trees with for two valued edge labeled trees edges, otherwise referred to as binary edge labeled trees. Thus, as previously described, no nodes corresponds to “0”. Continuing, a single node corresponds to “1” or to “root”. Likewise, the numerals two and three in this view turn out to be push operations. In this case, the numeral 2 is the tree corresponding to the A push of one denoted A(1). Likewise, the numeral 3 is the tree corresponding to the B push of one, denoted B(1).

[0049] For this embodiment, these relationships may also be confirmed by referring back to FIG. 7. Here, the operations A(x) and B(x) as defined in terms of the function,

previously defined in connection with FIG. 6. Thus, using these expressions, to determine the push of 1 denoted $A(1)$, as provided in FIG. 7, this is $Q((2*1)-2)$. This provides $Q(0)$ or the value 2, as demonstrated from FIG. 6. As similar result may be obtained for $B(1)$. Referring to FIG. 7, this corresponds to $Q((21)-1)$, or $Q(1)$, again from FIG. 6, the value 3.

[0050] A similar relationship may be established for three valued edge labeled trees, described by the expressions provided in FIG. 8, for example. Referring again to FIGS. 12 and 13, the edge labeled trees corresponding to these expressions are depicted in the second column. It is noted that the tree structures using this notation are the same between the first column and the second column for the numerals from zero to four. However, a difference is noted between the first column and the second column at numeral 5. Thus, for view 3, the numeral 5 is the C push of 1. More particularly, again referring to FIG. 8, $C(1)$ equals $Q(3*1)-1$, or $Q(2)$. From FIG. 6, the corresponding value is 5, as previously suggested. Similarly, looking at column 3 of FIG. 12, for view 5, the D push of 1 is numeral 7.

[0051] Thus, for this embodiment, regardless of the “view” of the edge labeled trees, there is a unique one to one correspondence, here, an association embodiment, between the natural numerals and that set of edge labeled trees. This embodiment, therefore, provides the capability to manipulate and combine edge labeled trees of different view. For example, for two edge labeled trees from two different views, one of the edge labeled trees may be converted so that the two edge labeled trees are in the same view. Once in the same view, the trees may be manipulated, such as by a merger, for example. Likewise, in an alternative embodiment, both trees may be converted to numerals, the numerals may be manipulated and then the manipulated numerals may be converted back to edge labeled trees of a particular view. Likewise, the edge labeled trees may be converted to any desirable view.

[0052] It is likewise noted that for this particular embodiment one way of manipulation an edge labeled tree is to apply a push operation to the edge labeled tree. Likewise, as previously described, for this embodiment, a push operation comprises adding an edge and labeling it. Assuming for this embodiment that the labels for the edge labeled tree comprise numerals, the label for a particular view will be a numeral that is less than the view itself. For example if the view is 5 then the set of distinct values to label an edge comprises 0, 1, 2, 3 or 4. Of course, this is merely one potential embodiment and the claimed subject matter is not limited in scope in this respect. For example, as previously described, letters may be employed. Likewise, any set of values where each value is distinct may be employed and remain within the scope of the claimed subject matter.

[0053] A similar set of manipulations may be applied to node labeled trees rather than edge labeled trees. Thus, node labeled trees may be represented in different views, may be converted to the same view, may be converted to numerals, combined, and converted back to a node labeled tree of a particular view. Likewise, a push operation may be applied to a node labeled tree, as previously described for edge labeled trees.

[0054] Although the claimed subject matter is not limited in scope in this respect, one technique for implementing this approach may be to apply a table look up approach. For example, a table providing different embodiments associat-

ing different views to natural numerals may be employed. Of course, the claimed subject matter is not limited in scope in this respect. For example, instead, a table look-up may be employed for the operation Q and the expressions previously described may be applied to perform manipulations, such as those previously illustrated, for example.

[0055] Techniques for performing table look-ups are well-known and well-understood. Thus, this will not be discussed in detail here. However, it shall be appreciated that any and all of the previously described and/or later described processing, operations, conversions, transformations, manipulations, etc. of strings, trees, numerals, data, etc. may be performed on one or more computing platforms or similar computing devices, such as those that may include a memory to store a table as just described, although, the claimed subject matter is not necessarily limited in scope to this particular approach. Thus, for example, a hierarchy of data, such as a tree as previously described, for example, may be formed. Likewise, operations and/or manipulations, as described, may be performed; however, operations and/or manipulations in addition to those described or instead of those described may also be applied. It is intended that the claimed subject matter cover such embodiments.

[0056] As described in prior embodiments, one technique for manipulating edge labeled trees includes converting such trees to natural numerals, performing manipulation of the natural numerals, and converting back to an edge labeled tree of a particular view. Furthermore, as described above, one technique for such conversions may include table look-up, as described above. Likewise, in another embodiment, it may be possible to convert a natural numeral directly to an edge labeled tree using a table look-up for the operation Q, previously described. For example, if it were desirable to convert the natural numeral 61 to an edge labeled tree in view 4, the numeral could be factored and the factors converted to trees. In this example, 61 is a non-composite, so, using a table look-up, $Q(17)$ provides 61. Thus, 61 is a push of 17. Using the expressions provided on FIG. 9, for example, we may determine whether 61 is the A, B, C, or D push of 17 and so forth. Likewise, for this particular embodiment, previously an example of converting between an edge labeled tree of a particular view and a natural numeral was provided.

[0057] Embodiments of a method of manipulating tree expressions have a variety of potentially useful applications. As described previously, trees provide a technique for structuring and/or depicting hierarchical data. Thus, for example, trees may be employed to represent language sentence structures, computer programs, algebraic formulae, molecular structures, family relationships and more. For example, one potential application of such a tree reduction technique is in the area of pattern matching. See, for example, “A VLSI Architecture for Object Recognition using Tree Matching” K. Sitaraman, N. Ranganathan and A. Ejnoui; Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP’02) 2000; “Expressive and efficient pattern languages for tree-structured data” by Frank Never and Thomas Schwentick; Proceedings of the Nineteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 2000. Thus, in pattern matching, substructures, in the form of a tree, for example, may be located within a larger structure, also in the form of a tree, referred to in this context as the target. This may be accomplished by comparing the

structures; however, typically, such a comparison is complex, cumbersome, and/or time consuming.

[0058] Of course, the claimed subject matter is not limited to unordered edge labeled trees. For example, as described in previously cited U.S. provisional patent application 60/543,371, binary edge labeled trees and binary node labeled trees may be employed nearly interchangeably to represent substantially the same hierarchy of data. In particular, a binary node labeled tree may be associated with a binary edge labeled tree where the nodes of the binary node labeled tree take the same values as the edges of the binary edge labeled tree, except that the root node of the binary node labeled tree may comprise a node having a zero value or a null value. Thus, rather than employing edge labeled trees (ELTs), the previously described embodiments may alternatively be performed using node labeled trees (NLTs). One example of a NLT is illustrated in the diagram of FIG. 14 by tree 1400. As one example embodiment, operations and/or manipulations may be employed using edge labeled trees and the resulting edge labeled tree may be converted to a node labeled tree. However, in another embodiment, operations and/or manipulations may be performed directly using node labeled trees.

[0059] In accordance with the claimed subject matter, therefore, any tree, regardless of whether it is edge labeled, node labeled, non-binary, a feature tree, or otherwise, may be manipulated and/or operated upon in a manner similar to the approach of the previously described embodiments. Typically, different views shall be employed, depending at least in part, for example, upon the particular type of tree. Furthermore or alternatively, as described in the previously referenced U.S. provisional patent application 60/543,371, a node labeled tree in which the nodes are labeled with natural numerals or data values may be converted to an edge labeled tree. Furthermore, this may be accomplished with approximately the same amount of storage. For example, for this particular embodiment, this may involve substantially the same amount of node and/or edge data label values. However, for convenience, without intending to limit the scope of the claimed subject matter in any way, here, operations and/or manipulations and the like have been described primarily in the context of ELTs.

[0060] In another embodiment, however, a particular tree may include null types or, more particularly, some node values denoted by the empty set. An advantage of employing null types includes the ability to address a broader array of hierarchical data sets. For example, without loss of generality and not intending to limit the scope of the claimed subject matter in any way, a null type permits representing in a database or a relational database, as two examples, situations where a particular attribute does not exist. As may be appreciated, this is different from a situation, for example, where a particular attribute may take on a numeral value of zero. Again, as described in the previously referenced U.S. provisional patent application 60/543,371, a tree with nulls, as described above, may be converted to a tree without nulls; however, the claimed subject matter is not limited in scope in this respect, of course. Thus, it may be desirable to be able to address both situations when representing, operating upon, manipulating and/or searching for patterns regarding hierarchical sets of data.

[0061] Likewise, in an alternative embodiment, a node labeled tree, for example, may comprise fixed length tuples of numerals. For such an embodiment, such multiple numer-

als may be combined into a single numeral, such as by employing Cantor pairing operations, for example. See, for example, *Logical Number Theory, An Introduction*, by Craig Smorynski, pp. 14-23, available from Springer-Verlag, 1991. This approach should produce a tree to which the previously described embodiments may then be applied.

[0062] Furthermore, a tree in which both the nodes and the edges are labeled may be referred to in this context as a feature tree and may be converted to an edge labeled tree and/or a node labeled tree. For example, without intending to limit the scope of the claimed subject matter, in one approach, a feature tree may be converted by converting any labeled node with its labeled outgoing edge to an ordered pair of labels for the particular node.

[0063] In yet another embodiment, for trees in which data labels do not comprise simply natural numerals, such as, as one example, trees that include negative numerals, such data labels may be converted to an ordered pair of numerals. For example, the first numeral may represent a data type. Examples include a data type such as negative, dollars, etc. As described above, such trees may also be converted to edge labeled trees, for example. However, again, this is provided for purposes of explanation and illustration. The claimed subject matter is not limited in scope to employing the approach of the previously referenced provisional patent application,

[0064] It will, of course, be understood that, although particular embodiments have just been described, the claimed subject matter is not limited in scope to a particular embodiment or implementation. For example, one embodiment may be in hardware, such as implemented to operate on a device or combination of devices, for example, whereas another embodiment may be in software. Likewise, an embodiment may be implemented in firmware, or as any combination of hardware, software, and/or firmware, for example. Likewise, although the claimed subject matter is not limited in scope in this respect, one embodiment may comprise one or more articles, such as a storage medium or storage media. This storage media, such as, one or more CD-ROMs and/or disks, for example, may have stored thereon instructions, that when executed by a system, such as a computer system, computing platform, or other system, for example, may result in an embodiment of a method in accordance with the claimed subject matter being executed, such as one of the embodiments previously described, for example. As one potential example, a computing platform may include one or more processing units or processors, one or more input/output devices, such as a display, a keyboard and/or a mouse, and/or one or more memories, such as static random access memory, dynamic random access memory, flash memory, and/or a hard drive. For example, a display may be employed to display one or more queries, such as those that may be interrelated, and or one or more tree expressions, although, again, the claimed subject matter is not limited in scope to this example.

[0065] In the preceding description, various aspects of the claimed subject matter have been described. For purposes of explanation, specific numbers, systems and/or configurations were set forth to provide a thorough understanding of the claimed subject matter. However, it should be apparent to one skilled in the art having the benefit of this disclosure that the claimed subject matter may be practiced without the specific details. In other instances, well-known features were omitted and/or simplified so as not to obscure the claimed

subject matter. While certain features have been illustrated and/or described herein, many modifications, substitutions, changes and/or equivalents will now occur to those skilled in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and/or changes as fall within the true spirit of the claimed subject matter.

1-126. (canceled)

127. A method of processing database content comprising:

manipulating one or more complex graphical expressions, the one or more complex graphical expressions comprise one or more labeled two-dimensional tree hierarchies;

configuring a first labeled two-dimensional tree hierarchy of the one or more labeled two-dimensional tree hierarchies in a first view of the one or more complex graphical expressions

configuring a second labeled two-dimensional tree hierarchy of the one or more labeled two-dimensional tree hierarchies in a second view of the one or more complex graphical expressions;

converting the second view of the second labeled two-dimensional tree hierarchy into a first view of the second labeled two-dimensional tree hierarchy;

and

combining the first labeled two-dimensional tree hierarchy in the first view and the second labeled two-dimensional tree hierarchy in the first view to form a combined labeled two-dimensional tree hierarchy of the one or more complex graphical expressions.

128. The method of claim **127**, wherein the conversion comprises identification of one or more numerical signal values associated with the first labeled two-dimensional tree hierarchy in the first view.

129. The method of claim **127**, wherein the conversion comprises identification of one or more numerical signal values associated with the second labeled two-dimensional tree hierarchy in the first view.

130. The method of claim **127**, wherein the conversion comprises identification of one or more numerical signal values associated with the second labeled two-dimensional tree hierarchy in the second view.

131. The method of claim **128**, wherein the identification of one or more numerical signal values associated with the first labeled two-dimensional tree hierarchy in the first view comprises of performing a table look up operation to determine the one or more numerical signal values associated with the first labeled two dimensional tree hierarchy in the first view.

132. The method of claim **129**, wherein the identification of one or more numerical signal values associated with the second labeled two-dimensional tree hierarchy in the first view comprises of performing a table look up operation to determine the one or more numerical signal values associated with the second labeled two-dimensional tree hierarchy in the first view.

133. The method of claim **130**, wherein the identification of one or more numerical signal values associated with the second labeled two-dimensional tree hierarchy in the second view comprises of performing a table look up operation to determine the one or more numerical signal values associated with the second labeled two dimensional tree hierarchy in the second view.

134. The method of claim **127**, wherein the converting comprises associating the first labeled tree hierarchy in the first view and/or the first labeled tree hierarchy in the second view using one or more numerical signal values via a look up table.

135. The method of claim **127**, wherein the converting the first labeled tree hierarchy in the first view to the first labeled tree hierarchy in the second view comprises using one or more numerical signal values associated with the view of the first labeled tree hierarchy via a look up table.

136. An apparatus comprising:

one or more processors coupled to one or more physical memory devices to store executable instructions and to store binary digital signal quantities, as physical memory states, wherein the executable instructions are to be accessible from the one or more physical memory devices for execution by the one or more processors;

wherein the one or more processors are to store in at least one of the physical memory devices, binary signal quantities, if any, that are to result from execution of the instructions on the one or more processors, wherein the one or more physical memory devices are also to store a database, or a portion thereof, wherein the executable instructions are to arrange database content;

wherein the instructions to arrange database content are executable to:

convert a first labeled tree hierarchy in a first view to the first labeled tree hierarchy in a second view; and

combine the first labeled tree hierarchy in the first view and the first labeled tree hierarchy in the second view to form a combined labeled tree hierarchy, wherein the first view is to be different from the second view.

137. The apparatus of claim **136**, wherein the instructions to arrange database content comprise executable instructions to access a data structure in the database, or the portion thereof, to identify one or more numerical signal values to be associated with the first labeled tree hierarchy in the first view.

138. The apparatus of claim **137**, wherein the instructions to access the data structure comprise executable instructions to perform a table look up operation to determine the one or more numerical signal values to be associated with the first labeled tree hierarchy in the first view.

139. The apparatus of claim **136**, wherein the instructions to arrange database content comprise executable instructions to compare one or more numerical signal values to be associated with the first labeled tree hierarchy in the first view and one or more other numerical signal values to be stored in the database, or the portion thereof, to detect a presence of the one or more numerical signal values, wherein a correspondence is to be indicative of a presence of content in the database, or the portion thereof, to correspond to the one or more numerical signal values and is to be indicative of one or more locations thereof in the database, or the portion thereof.

140. The apparatus of claim **136**, wherein the instructions to arrange database content comprise executable instructions to generate an association between the one or more labeled tree hierarchies and one or more numerical signal values;

wherein the executable instructions to generate the association comprise executable instructions to associate the first labeled tree hierarchy in the first view with at least one numerical signal value.

141. The apparatus of claim **136**, wherein the instructions to arrange database content comprise executable instructions so that the one or more labeled tree hierarchies are to comprise one or more labeled tree hierarchies to have one or more associated numerical signal values.

142. The apparatus of claim **136**, wherein the instructions to arrange database content comprise executable instructions so that the one or more labeled tree hierarchies are to comprise one or more edge-labeled tree (ELT) hierarchies to have one or more associated numerical signal values.

143. The apparatus of claim **136**, wherein the instructions to arrange database content comprise executable instructions so that the one or more labeled tree hierarchies are to comprise one or more node-labeled tree (NLT) hierarchies to have one or more associated numerical signal values.

144. The apparatus of claim **136**, wherein the instructions to convert the first labeled tree hierarchy in the first view to the first labeled tree hierarchy in the second view are executable to: associate the first labeled tree hierarchy in the first view and/or the first labeled tree hierarchy in the second view with one or more numerical signal values via a look up table.

145. The apparatus of claim **136**, wherein the instructions to convert the first labeled tree hierarchy in the first view to the first labeled tree hierarchy in the second view are executable to: convert the first labeled tree hierarchy in the first view to the first labeled tree hierarchy in the second

view substantially in accordance with one or more numerical signal values associated with the view of the first labeled tree hierarchy via a look up table.

146. An article comprising: a non-transitory storage medium that includes executable instructions stored thereon; wherein the instructions are executable by one or more processors to be coupled to one or more physical memory devices;

wherein the one or more physical memory devices are to store a database, or a portion thereof, wherein the executable instructions are to arrange database content; wherein the instructions to arrange database content are to:

convert a first labeled tree hierarchy in a first view to the first labeled tree hierarchy in a second view; and

combine the first labeled tree hierarchy in the first view and the first labeled tree hierarchy in the second view to form a combined labeled tree hierarchy, wherein the first view is to be different from the second view.

147. The article of claim **146**, wherein the instructions to arrange database content comprise executable instructions to access a data structure in the database, or the portion thereof, to identify one or more numerical signal values to be associated with the first labeled tree hierarchy in the first view.

* * * * *