Structural and Content Queries on the Nested Sets Model

Gianmaria Silvello

Department of Information Engineering, University of Padua, Italy
silvello@dei.unipd.it

Abstract. Hierarchical structures are pervasive in computer science. They are a fundamental means for modeling many aspects of reality and for managing wide corpora of data and digital resources. How to model and manage data hierarchies is a major theme in database theory and practice. The most important hierarchical structure is the tree, which has been widely studied, analyzed, and adopted in several contexts and scientific fields over time. In this paper we describe an alternative approach for modeling and managing hierarchies, called the Nested Sets Model (NS-M) and we describe how it is possible to reconsider structural and content queries by exploiting this model.

1 Motivation

Hierarchical structures are pervasive in computer science. They are a fundamental means for modeling many aspects of reality and for managing wide corpora of data and digital resources. How to model and manage data hierarchies is a major theme in database theory as well as in database applications [1] and it has been extensively treated in the context of database abstractions, the relational model, the complex value model, object-oriented database systems, and semistructured data and *Extensible Markup Language* (XML).

In the database field and more generally in computer science, the fundamental structure used to model and represent hierarchies is the tree. The tree has been formalized and deeply studied, its properties are well understood, and many of the algorithms based on it run in polynomial time. Therefore, the tree has been exploited by researchers and developers in many different fields to model and solve their problems. In this sense, trees are considered "*the most important nonlinear structures that arise in computer science*" [6].

In this article we propose an alternative approach for modeling and managing hierarchies, called Nested Sets Model (NS-M), which is based on the inclusion between sets as a means to capture hierarchical relations. The foundational idea underlying this model is to express the hierarchical relationships between objects through the inclusion property between sets, in place of the binary relation between nodes exploited by the tree. Therefore, the pivotal question we want to address is how structural and content queries on hierarchies can be handled with a data model other than the tree. In order to answer this question we define the

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

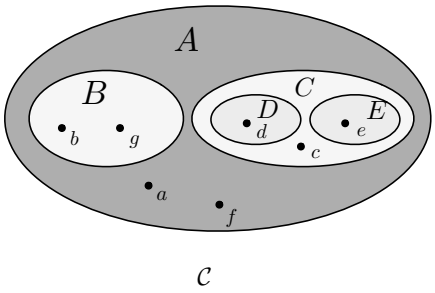

Fig. 1. A sample Nested Sets Collection (NS-C) mapped from a tree represented by means of an Euler-Venn diagram.

NS-M (Section 2.1), we describe how it allows us to handle structural and content components of a hierarchy (Section 2.2) and we prove its properties (Section 2.3). In Section 3 we specify basic structural and content query operations by highlighting how they can be handled by the NS-M; and, in Section 4 we draw some final remarks and future works.

2 The Nested Sets Model (NS-M)

2.1 Definition of the Model

The NS-M is formally defined as a collection of subsets where specific conditions must hold [2,4].

Definition 1 *Let A be a set and let $\mathcal{C}$ be a collection of subsets of A. Then $\mathcal{C}$ is a **Nested Sets Collection** (NS-C) if:*

$$A \in \mathcal{C}, \tag{2.1}$$

$$\forall H, K \in \mathcal{C} \mid H \cap K \neq \emptyset \Rightarrow H \subseteq K \vee K \subseteq H. \tag{2.2}$$

Therefore, we define a NS-C as a collection of subsets where two conditions must hold. The first condition (2.1) states that set A which contains all the subsets of the collection must belong to the NS-C itself. The second condition states the intersection of every couple of sets in the NS-C is not the empty-set only if one set is a proper subset of the other one. [1]

The NS-C is represented by means of an Euler-Venn diagram as we can see in Figure 1, which represents a sample NS-C composed of five nested sets:

[1] The graphical representation of a tree as a collection of nested sets was originally presented in [6] with no formal definition of a "nested sets model"; afterwards, this representation has been exploited in [3] to explain an alternative way to solve recursive queries over trees by using an integer intervals encoding. Also [5] proposed the same idea to solve recursion in relational databases.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

$\mathcal{C} = \{A, B, C, D, E\}$. We can see that A is the *top set* of $\mathcal{C}$ (i.e. the common superset of all the sets in $\mathcal{C}$) and thus Condition 2.1 is respected, and all the sets are either disjoints or one is a proper subset of the other as required by Condition 2.2.

2.2 Separating Structure and Content

From the *structural point-of-view*, a collection of subsets is represented by the sets in the collection and their inclusion dependencies. A collection of subsets at the *intensional level* is defined by its structure. Let us consider an example, we can say that $\mathcal{C} = \{A, B, C\}$ where $B \subseteq A$, $C \subseteq A$ and $B \nsubseteq C \wedge C \nsubseteq B$ is a NS-C because it respects conditions 2.1 and 2.2 of Definition 1. In this way, we know the structure of the collection and we know which relationships hold between the sets.

When we consider a collection of subsets $\mathcal{C}$ from the the *content point-of-view*, it means that we refer to its *extensional level*. From the *content point-of-view* a collection of subsets $\mathcal{C}$ is represented by the *extension* of the sets composing it; the properties of the sets are then verified by inspecting the sets and verify the elements that they contain. In this case, we say that the content of a collection of subsets defines the *extension* of such a collection. Therefore, we can say that $\mathcal{C} = \{A, B, C\}$ where $A = \{a, b, c, d\}$, $B = \{b\}$ and $C = \{c, d\}$ is the extension of a NS-C. In the next example we can see a NS-C defined at the intensional level which at the extensional level is satisfied by two different NS-C.

Example 1 *Let us consider the following NS-C defined at the intensional level:* $\mathcal{C} = \{A, B, C, D\}$ *where* $B \subseteq A$, $C \subseteq A$, $D \subseteq C$ *and* $B \nsubseteq C \wedge C \nsubseteq B$. *Then,* $A = \{a, b, c, d, e\}$, $B = \{b\}$, $C = \{c, d, e\}$, $D = \{d, e\}$ *is a valid instance for* $\mathcal{C}$, *as well as* $A = \{a, b, c, d, e, f\}$, $B = \{c, d\}$, $C = \{b, e, f\}$ *and* $D = \{f\}$; *indeed, they both satisfy the specified structural conditions.*

Both the structural and the content point-of-view are important for the treatment of the NS-M. We exploit the structure defined at the intensional level to define the properties of NS-M; whereas we exploit the extensional level to perform set operations which manipulate the content of the subsets composing the collections.

In the following we make extensive use of the concepts of collection of proper subsets and supersets and of direct subsets and supersets. Let $\mathcal{C}$ be a collection of sets and $A \in \mathcal{C}$ be a set, we define:

- $\mathcal{S}^-(A) = \{B \in \mathcal{C} \ : \ A \subset B\}$ to be the **collection of proper supersets** of A in $\mathcal{C}$;
- $\mathcal{S}^+(A) = \{B \in \mathcal{C} \ : \ B \subset A\}$ to be the **collection of proper subsets** of A in $\mathcal{C}$.
- $\mathcal{D}^-(A) = \{B \in \mathcal{C} \ : \ \big((A \subset B) \wedge (\nexists E \in \mathcal{C} \mid A \subset E \subset B)\big)\}$ to be the **collection of direct supersets** of A in $\mathcal{C}$.
- $\mathcal{D}^+(A) = \{B \in \mathcal{C} \ : \ \big((B \subset A) \wedge (\nexists E \in \mathcal{C} \mid B \subset E \subset A)\big)\}$ to be the **collection of direct subsets** of A in $\mathcal{C}$.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

2.3 Properties of the Model

Many properties of the NS-M are derived by the straightforward application of set theory as we show in the following example which takes into account the intensional level.

Example 2 *Let $\mathcal{C}$ be a NS-C. For all $H, K \in \mathcal{C} \mid H \subseteq K$ we can derive from set theory that $H \cup K = K$ and $H \cap K = H$. As well as we can say that for all $H, K \in \mathcal{C} \mid H \nsubseteq K \wedge K \nsubseteq H \Rightarrow H \setminus K = H \wedge K \setminus H = K$.*

In this example we see that the sets in a NS-C behave exactly as one would expect under the operations of union, intersection and set difference. Let us see an example which shows how these operations behave at the extensional level.

Example 3 *Let $\mathcal{C} = \{A, B, C\}$ be a NS-C, where $B \subseteq A$ and $C \subseteq B$. Then let us consider the following instance: $A = \{a, b, c, d, e\}$, $B = \{c, d, e\}$ and $C = \{e\}$. Then, $B \cup C = \{c, d, e\} = B$ and $B \cap C = \{e\} = C$.*

Let us consider a NS-C $\mathcal{C}$, the next proposition shows that for all $H \in \mathcal{C}$, H has at most one direct superset.

Proposition 1 *Let $\mathcal{C}$ be a NS-C. Then, $\forall H \in \mathcal{C}, |\mathcal{D}^-(H)| \leq 1$.*

Proof. Ab absurdo suppose that $\exists H \in \mathcal{C}$ such that $|\mathcal{D}^-(H)| > 1 \Rightarrow \exists K, L \in \mathcal{D}^-(H) \mid H \subseteq K \wedge H \subseteq L \wedge L \nsubseteq K \wedge K \nsubseteq L \Rightarrow K \cap L = H \Rightarrow \mathcal{C}$ is not a NS-C (condition 2.2 of Definition 1).

The following corollary to this proposition shows that the set with minimum cardinality in the collection of supersets of H is its direct superset.

Corollary 2 *Let $\mathcal{C}$ be a NS-C, $H \in \mathcal{C}$ be a set, $\mathcal{S}^-(H)$ be the collection of proper supersets of H and $K \in \mathcal{S}^-(H)$ where $\forall L \in \mathcal{S}^-(H), |K| \leq |L|$ be the subset with minimum cardinality in $\mathcal{S}^-(H)$. Then, $\mathcal{D}^-(H) = K$.*

Proof. We know from Proposition 1 that $|\mathcal{D}^-(H)| \leq 1$. Then, ab absurdo suppose that $\forall L \in \mathcal{S}^-(H), |K| \leq |L|$ and that $\exists W \in \mathcal{S}^-(H) \mid \big((|W| > |K|) \wedge (\mathcal{D}^-(H) = W)\big)$. This means that $H \subseteq W \wedge H \subseteq K$ and by definition of NS-M $W \subseteq K \vee K \subseteq W$. If $W \subseteq K \Rightarrow |W| < |K|$; if $K \subseteq W \Rightarrow |K| < |W|$. So if $\mathcal{D}^+(H) = W \Rightarrow |W| < |K|$.

The next proposition proves that the direct subsets of H are always disjoints.

Proposition 3 *Let $\mathcal{C}$ be a NS-C and $H \in \mathcal{C}$ be a set, then $\forall K, L \in \mathcal{D}^+(H), K \cap L = \emptyset$.*

Proof. Ab absurdo suppose that $K \cap L \neq \emptyset \Rightarrow K \cap L = W$ such that $|W| \geq 1 \wedge W \nsubseteq K \wedge W \nsubseteq L \Rightarrow \mathcal{C}$ is not a NS-C.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

3 An Insight on Structural and Content Queries

When we deal with a NS-C, we can distinguish between **structural** and **content** queries. Structural queries ask for the relationships between the sets in a collection of subsets – e.g. "Return all the sets in $\mathcal{C}$ which are supersets of the set $H \in \mathcal{C}$", in this case we are not interested in the actual content of the sets, instead we just want to know which sets are supersets of H. Content queries ask for the actual content (the elements) of the sets in a collection of subsets – e.g. "Return all the elements in $\mathcal{C}$ belonging to supersets of $H \in \mathcal{C}$" or "Return all the elements in $\mathcal{C}$ belonging to subsets of $H \in \mathcal{C}$". A possible data structure to implement these queries has to take into account the structural and the content components of the NS-M. it is possible to propose a dictionary-based data structure which from the structural point-of-view, stores the inclusion dependencies between the sets; and, from the content point-of-view, it stores the materialization of the sets (i.e. the elements of each set).

If we consider a NS-C $\mathcal{C}$ and a set $H \in \mathcal{C}$, the **structural** query operations we point out are: Descendants(H) which returns the sets which are descendants of H, Ancestors(H) which returns the sets which are ancestors of H, Children(H) which returns the sets which are children of H, and Parent(H) which returns the set which is the parent of H. The worst-case scenario for all these structural queries is represented by a NS-C structured as a chain. In the case of the Descendants(H) query the worst-case input set H is the top set because the query has to return all the sets in the given NS-C. All other structural query operations can be implemented to run in constant time for whichever collection of subsets and input set by exploiting the described properties of NS-M. For instance, the Parent operation can be implemented in $O(1)$ time by exploiting the fact that every set in a NS-C has at most one direct superset as proved in Proposition 1.

The **content** query operations are: Elements(H) which returns the elements in the set H, DescendantElements(H) which returns the elements in the descendants of H, AncestorElements(H) which returns the elements in the ancestors of H, ChildrenElements(H) which returns the elements in the children of H, and ParentElements(H) which returns the elements in the parent of H.

The NS-M is built in such a way that each set contains all the elements of its descendants; therefore, it is possible to answer this query without browsing the collection of subsets (without browsing the hierarchy) or without passing through any structural query. We can see how these queries are independent with respect to their correspondent structural queries. In order to answer the content queries we do not need to browse the hierarchical structure of the collection of subsets.

4 Final Remarks and Future Works

By defining the NS-M we showed that it is possible to model hierarchies exploiting the inclusion property between sets in place of the binary relationship

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

between nodes adopted by the tree. We provided the theoretical basis for employing the NS-M as an alternative to the tree to model hierarchies, and for exploiting a set-theoretical environment for the definition of problems and solutions with hierarchies. It is possible to exploit the fact that content and structure components of hierarchies are modeled as independent building blocks of the NS-M to define structural and content queries. We gave a first insight from the theoretical point-of-view on how content queries do not strongly rely on structural queries as happens with the tree.

In the future we are going to conduct experiments on synthetic and real datasets to experimentally verify that both structural and content query operations are scalable and efficient in the NS-M. The experimental analysis will be aimed to show how the choice between the tree and the NS-M has to be made on an application basis by evaluating which operations are executed more frequently. It will be thus possible to investigate and propose data structures optimized for specific contexts and applications so that to gain further performances for the NS-M.

Acknowledgments

The author thanks Maristella Agosti and Nicola Ferro which supported this work and contributed to many of the presented ideas. The author would like to thank Carlo Meghini for his useful and accurate observations regarding the formalization and the properties of the NS-M. The PROMISE network of excellence[2] (contract n. 258191) projects, as part of the 7th Framework Program of the European Commission, have partially supported the reported work.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. M. Agosti, N. Ferro, and G. Silvello. The NESTOR Framework: Manage, Access and Exchange Hierarchical Data Structures. In *Proceedings of the 18th Italian Symposium on Advanced Database Systems*, pages 242–253. Società Editrice Esculapio, Bologna, Italy, 2010.
3. J. Celko. *Joe Celko's SQL for Smarties: Advanced SQL Programming*. Morgan Kaufmann, San Francisco, California, USA, 2000.
4. N. Ferro and G. Silvello. The NESTOR Framework: How to Handle Hierarchical Data Structures. In M. Agosti, J. Jose Borbinha, S. Kapidakis, C. Papatheodorou, and G. Tsakonas, editors, *Proc. 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2009)*, pages 215–226. Springer, Heidelberg, Germany, 2009.
5. M. Kamfonas. Recursive Hierarchies: The Relational Taboo! *The Relational Journal*, October/November 1992.
6. D. E. Knuth. *The Art of Computer Programming, third edition*, volume 1. Addison Wesley, Reading, MA, USA, 1997.

[2] `http://www.promise-noe.eu/`

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,