

# Symbolic Trajectories in SECONDO: Pattern Matching and Rewriting

Fabio Valdés<sup>1</sup>, Maria Luisa Damiani<sup>2</sup>, and Ralf Hartmut Güting<sup>1</sup>

<sup>1</sup> FernUniversität Hagen, Germany

<sup>2</sup> Università degli Studi di Milano, Italy

{fabio.valdes,rhg}@fernuni-hagen.de  
mdamiani@dico.unimi.it

**Abstract.** In this paper, we introduce a novel data model for representing symbolic trajectories along with a pattern language enabling both the matching and the rewriting of trajectories. We illustrate in particular the trajectory data type and two operations for querying symbolic trajectories inside the database system SECONDO. As an important application of our theory, the classification and depiction of a set of real trajectories according to several criteria is demonstrated.

## 1 Introduction

Recently, pattern matching with movement history has been studied extensively [1,2]. Since raw GPS records are inconvenient for most matching applications, [3,4,5] and especially [6] focus on trajectories containing semantic information.

A symbolic trajectory is a sequence of temporally annotated labels each of which is a semantically meaningful description, e.g., a street or city name, an activity, or a means of transportation. Inside our database system SECONDO [7,8], a symbolic trajectory is called a moving label, since similar data types (moving point, moving real, etc.) are supported. SECONDO converts GPS data into a moving label by matching the segments onto a map and storing the according street names from OpenStreetMap [9]. For ease of exposition, we provide a short movement history of a person inside the city of Dortmund in nested list syntax:

```
((("2012-03-31-13:17:01" "2012-03-31-13:18:21" TRUE FALSE) "Hansastr.")  
  (("2012-03-31-13:18:21" "2012-03-31-13:20:37" TRUE FALSE) "Kampstr.")  
  (("2012-03-31-13:20:37" "2012-03-31-13:21:07" TRUE FALSE) "Brueckstr.")  
  (("2012-03-31-13:21:07" "2012-03-31-13:21:47" TRUE FALSE) "Hohe Luft")  
  (("2012-03-31-13:21:47" "2012-03-31-13:21:57" TRUE FALSE) "Bissenkamp"))
```

Each line of the quoted moving label corresponds to a so-called unit label, i.e., a combination of a time interval and a description. The boolean expressions indicate whether the start and end instant belong to the interval, respectively.

We created and implemented the operators `matches` and `rewrite`, the former for matching and the latter for rewriting a symbolic trajectory, both available in SECONDO. For using the first one, the user has to provide a mobility pattern, arbitrarily many additional conditions, and a symbolic trajectory. The result is

true if and only if the specified pattern matches the trajectory, thus the operator yields a boolean value and may be applied as a selection criterion for filtering trajectories fulfilling a specific pattern – and related conditions, if specified – from a large database relation. For example, from all the trips of a person during the last 12 months, we may obtain exactly the paths from home to work.

The second operator requires the same input as `matches`, extended by a rewrite part consisting of result variables and optional assignments. Again, a matching decision is computed, but `rewrite` focuses on the fact that there may be numerous matching possibilities (e.g., if a place is visited repeatedly inside a trajectory). Its result is a set of moving labels, more exactly, according to the selection of the user, a specific part of the trajectory is returned, and due to the possible matching ambiguity, there may be several results. In addition, these results can be rewritten in order to enrich them with further knowledge.

The main contributions of our research, compared to the abovementioned related work, are the expressiveness of our pattern language allowing the use of SECONDO database queries for filtering moving labels as well as the ability to rewrite parts of the moving label. The latter enables the user to classify large sets of symbolic trajectories concerning business trips to particular customers, migration behaviors of different bird species, positions of components for an automatic manufacturing process, or private journeys to special destinations. A comprehensive example is presented below.

In Section 2, we introduce the pattern language for matching and rewriting a symbolic trajectory. The demonstration is reported in Section 3.

## 2 Pattern Language

This section is dedicated to the language for specifying a `matches` or `rewrite` query in SECONDO. As mentioned above, we distinguish four parts of the input. This and the next section refer to the following example:

```
X ( _ "Alte Teichstr." ) Y * Z ( _ "Alte Teichstr." ) // get_duration(X.time)
+ get_duration(Y.time) + get_duration(Z.time) < (duration (0 1200000))
=> A // A.label := "short walk", A.start := X.start, A.end := Z.end
```

*Unit Patterns.* We consider the first query line until `//` containing a sequence of unit patterns, each of which may be assigned a variable by prepending it. For both operators, this part is crucial for the matching decision and thus mandatory. Each unit pattern has one of the forms  $(t\ l)$ ,  $((t\ l))$ ,  $+$ , or  $*$ , where  $t$  is a time interval either in semantic or in numerical form (e.g., `afternoon`, `2012-05-12~2012-05-13-23:45:00`) and  $l$  is a label.<sup>1</sup> A simple pattern  $(t\ l)$  matches a unit label  $(t_u\ l_u)$  if and only if  $t_u \subset t$  and  $l = l_u$ . An underscore acts as a wildcard, i.e., guarantees a time or label match, respectively. The second form has the same matching criteria but refers to one or more unit labels, as long as a matching is possible. The two remaining alternatives match any sequence of unit labels and differ insofar as  $*$  may also match no unit label at all.

The values of the assigned variables are accessed in the following parts.

---

<sup>1</sup> The user may specify sets of time intervals resp. labels instead of single ones.

*Conditions.* Subsequent to the unit patterns, the user may append conditions (until `=>`) obeying the following two rules: Each condition has to be evaluable into a boolean result by `SECONDO` and to contain at least one expression of the form `v.attr`, where `v` is one of the variables from the previous part, and `attr` is one of the attributes `label`, `time`, `start`, `end`, `card` (the latter denotes the number of unit labels matched by the respective unit pattern) of this variable determined by the matching. Notably, several unit patterns may be linked in one condition.

*Results.* In this required part, ranging from `=>` to `//`, the user controls the `rewrite` operator's result by specifying the output variables, either known from the unit patterns section or new. The latter must be enriched with information in the following part. For every matching possibility (if any), the unit labels assigned to the output variables are merged into a moving label which is returned.

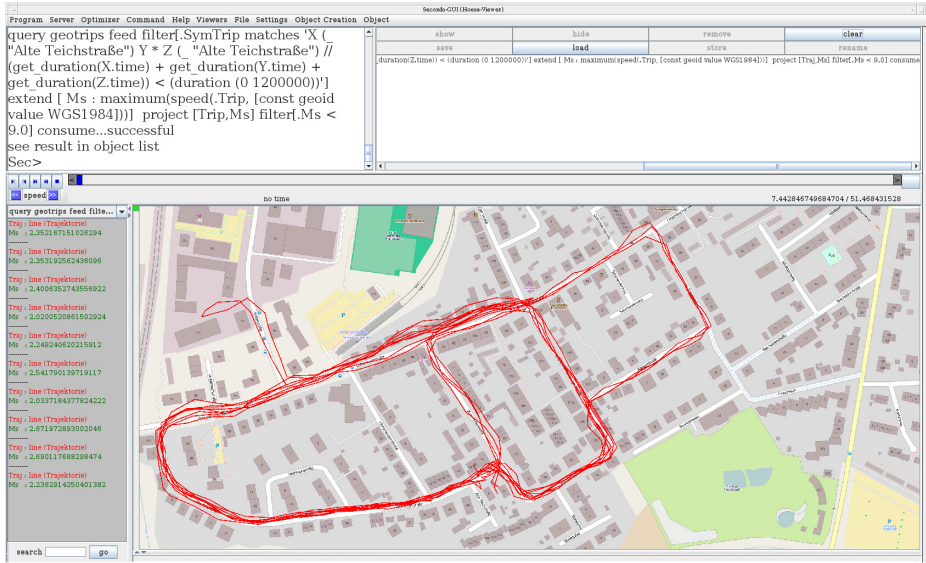
*Assignments.* Finally (fourth line after `//`), the user may rewrite the result by assigning new values (either by referring to values from other variables or by typing them in `SECONDO` syntax) to certain result unit labels. Left from the assignment symbol `:=`, a `v.attr` expression (as explained in the conditions part, although the use of `card` is not allowed here) is required. The assigned value has to be a `SECONDO`-evaluable expression of the respective data type.

### 3 Demonstration

In the following, we demonstrate the functionality of the operators. The used relation contains 169 moving labels of different lengths from one person, covered by car, train, bike, or foot, including trips from home to work or leisure walks. Our objective is to extract all walks starting and ending in "Alte Teichstr." (1) of less than 20 minutes (2) from the relation and to rewrite the resulting moving labels, in order to have only one unit per walk (3) containing the respective time interval and the label "short walk" (4). This is obtained by executing `rewrite` with the pattern from section 2 in `SECONDO`. For the sake of brevity, only the query parts related to our operators are discussed. Initially, the operator converts the unit pattern sequence into an NFA whose states represent the unit patterns, except for the accepting state which is active only in case of a complete match with the moving label. This automaton performs filter step (1), after which the variable bindings are stored. Subsequently, the `v.attr` parts of the condition (note that 20 minutes are entered as 0 days and 1,200,000 ms) are replaced by the respective values, such that the expression can be evaluated by `SECONDO`, excluding too long walks (2). The result part contains one new variable, hence each resulting moving label consists of one unit label (3). Since the variable `A` is not attached to a unit pattern, the assignment section has to provide its components with data that are either user-typed ("short walk") or dependent on the trajectory and on the variable bindings (`x.start`, `z.end`) (4). We present one of the resulting moving labels:

```
((("2012-04-02-08:10:05" "2012-04-02-08:26:27" TRUE TRUE) "short walk"))
```

The runtime of this query amounts to 0.047 sec on an AMD Phenom II X6 3.3 GHz running openSUSE 11.4, with 8 GBytes of memory. The walks remaining after the selections (1) and (2) are depicted in Figure 1.



**Fig. 1.** A screenshot of the filtered walks from the SECONDO GUI

Note that SECONDO is freely available and may be downloaded from [8].

## References

1. Hadjieleftheriou, M., Kollios, G., Bakalov, P., Tsotras, V.J.: Complex spatio-temporal pattern queries. In: Proc. VLDB, pp. 877–888 (2005)
2. Mokhtar, H.M.O., Su, J.: A query language for moving object trajectories. In: Proc. SSDBM, pp. 173–184 (2005)
3. Vieira, M.R., Bakalov, P., Tsotras, V.J.: Querying trajectories using flexible patterns. In: Proc. EDBT, pp. 406–417 (2010)
4. Vieira, M.R., Bakalov, P., Tsotras, V.J.: Flextrack: A system for querying flexible patterns in trajectory databases. In: Int. Symp. SSTD, pp. 475–480 (2011)
5. Vieira, M.R., Tsotras, V.J.: Complex motion pattern queries for trajectories. In: ICDE Workshops, pp. 280–283 (2011)
6. du Mouza, C., Rigaux, P.: Mobility patterns. *GeoInformatica* 9(4), 297–319 (2005)
7. Güting, R.H., Behr, T., Düntgen, C.: Secondo: A platform for moving objects databases research and for publishing and integrating research implementations. *IEEE Data Eng. Bull.* 33(2), 56–63 (2010)
8. <http://dna.fernuni-hagen.de/Secondo.html> (2012)
9. <http://www.openstreetmap.org> (2012)