Representing Public Transport Schedules as Repeating Trips

Romans Kasperovics, Michael H. Böhlen, Johann Gamper
Free University of Bozen-Bolzano
Via Sernesi 1, I-39100, Bozen-Bolzano, Italy
{kasperovics,boehlen,gamper}@inf.unibz.it

Abstract

The movement in public transport networks is organized according to schedules. The real-world schedules are specified by a set of periodic rules and a number of irregularities from these rules. The irregularities appear as cancelled trips or additional trips on special occasions such as public holidays, strikes, cultural events, etc. Under such conditions, it is a challenging problem to capture real-world schedules in a concise way.

This paper presents a practical approach for modelling real-world public transport schedules. We propose a new data structure, called repeating trip, that combines route information and the schedule at the starting station of the route; the schedules at other stations can be inferred. We define schedules as semi-periodic temporal repetitions, and store them as pairs of rules and exceptions. Both parts are represented in a tree structure, termed multislice, which can represent finite and infinite periodic repetitions. We illustrate our approach on a real-world schedule and we perform in-depth comparison with related work.

1 Introduction

Public transport schedules are complex objects that combine spatial aspects (e.g., the specific route of a bus) and temporal aspects (e.g., the time when a bus enters/exits each station). For many on-line services it is important to store the complete information about the schedules including irregularities, e.g., cancelled or additional trips which appear on special occasions, such as holidays, cultural events, strikes and nature disasters. Traditionally, schedules are represented as sets of periodic rules making up the regular schedule, however, the actual situation forces to introduce a number of irregularities from these rules. The problem arises when one needs to represent and query such schedules in a uniform way. Example 1 illustrates a real-world schedule that we use as a running example in this paper.

Example 1 *A fragment of a real-world schedule of bus line 1 on the route "Kardaun" – "Cadornastraße" is shown in Fig. 1. The schedule consists of periodic rules describing the movement of buses on working days (columns marked with "W"), alternative rules for Sundays and public holidays (columns marked with "+"), cancellations of particular buses, and additional buses. The working days are defined as weekdays from Monday to Saturday except public holidays.*

	W	W	+	W	W	...
Kardaun	07:19	07:30	07:44	07:45	08:00	...
Brennerstr.	07:23	07:34	07:48	07:49	08:04	...
Bahnhof	07:27	07:38	07:52	07:53	08:08	...
Sernesiplatz	07:30	07:41	07:55	07:56	08:11	...
Cadornastr.	07:34	07:45	07:59	08:00	08:15	...

* No buses on 2007-11-24 at 19:44, and 19:59 due to road works, and on 2008-01-06 at 07:44 due to holiday's market.
** Additional bus on 2007-09-30 at 07:05 due to cable car problems.

Figure 1. A fragment of a real-world bus schedule.

In practice, the problem of representing real-world schedules is usually solved with ad-hoc solutions. In the last 20 years the research community has proposed a number of general approaches how to represent complex temporal repetitions, such as schedules. The most advanced approaches [1, 6, 8] do not explore the applications in public transport networks and do not show how the temporal repetitions can be linked with spatial information.

We propose a new data structure, called *repeating trip*, which is able to capture periodic schedules with irregularities and link them to the spatial information. The main idea of a repeating trip is to store the temporal repetition of trips only at the starting station. The route information together with time offsets from the starting station are stored in a structure, termed *relative trip*. Having a temporal repetition at the starting station and a relative trip, the repetitions at other stations can be inferred. We represent temporal repetitions as *mixed recurrences* which are sets of cou-

1530-1311/08 $25.00 © 2008 IEEE
DOI 10.1109/TIME.2008.26

pled rules and exceptions. Both rules and exceptions can describe finite or infinite periodic time sequences and they both are specified in a uniform way with *multislices*, which is a compact representation of periodic and finite temporal repetitions.

Taking the schedule from Example 1, the relative trip is (("Kardaun", 0), ("Brennerstr.", 4), ("Bahnhof", 8), ("Sernesiplatz", 11), ("Cadornastr.", 15)). Figure 2 illustrates the schedule of bus line 1 at the station "Kardaun" as a mixed recurrence. Buses on weekdays from Monday to Saturday form a periodic rule with a weekly period. Public holidays make an exception in the weekly schedule along with cancelled trips on 2007-11-24 at 19:44 and 19:59. Buses on holidays form another rule with just one exception on 2008-01-06 at 07:44. As the last pair of rules and exceptions we have an extra trip on 2007-09-30 at 07:05 with no exceptions. Here, $\lambda_1, \ldots, \lambda_6$ refer to the formal representations of these rules and exceptions with multislices, which are given later on in this paper (see Fig. 3 and Fig. 4).

Rules	Exceptions
λ_1: Mon-Sat buses	λ_2: Holidays, 2007-11-24-19:44, 2007-11-24-19:59
λ_3: Buses on Sundays and holidays	λ_4: 2008-01-06-07:44
λ_5: 2007-09-30-07:05	λ_6: empty

Figure 2. Mixed recurrence representing the schedule of line 1 at the station "Kardaun".

In this paper we identify real-world schedules as semi-periodic temporal repetitions. We define multislices as a core part of our representation of semi-periodic temporal repetitions. We show how multislices can be combined into mixed recurrences to represent any semi-periodic temporal repetitions. We show how the temporal and spatial aspects of schedules can be captured using repeating trips.

The rest of the paper is organized as follows. After some preliminary definitions in Sec. 2 we introduce and define multislices, mixed recurrences, and repeating trips in Sec. 3. Section 4 discusses related work and Sec. 5 draws conclusions and points to future work.

2 Basic Notions

We assume a discrete model of time, where the time domain, T, is an infinite, discrete, and totally ordered set of time instants. For readability, we denote the time instants as timestamps, e.g., 2006-09-16-07:19, 2006-09-16-07:23, 2006-09-16-07:27. We assume an isomorphism between the integers, $\mathbb{Z}$, and T, so we can refer the elements of T with their integer indexes.

A *temporal repetition* is an association of the same data item(s) with a possibly infinite subset of T. For example, the schedule of bus line 1 at station "Kardaun" is a temporal repetition, where "line 1, Kardaun" is the repeating data item and the departure times $\{\ldots,$ 2006-09-16-07:19, 2006-09-16-07:23, 2006-09-16-07:27, $\ldots\}$ are an infinite subset of T. A *flat representation* of temporal repetition explicitly enumerates all time instants. A *compact representation* of temporal repetition specifies a subset of T without enumerating all its elements. A temporal repetition is *finite* if the associated subset of T is finite. A temporal repetition is *periodic* if the integer indexes of the associated subset of T form a periodic set.

A set $\tilde{S} \subseteq \mathbb{Z}$ is a *periodic set* if there exists a positive integer p, called a period, such that for all $n \in \mathbb{Z}$ and for all $i \in \tilde{S}$, $i + np \in \tilde{S}$. Let $\tilde{S}$ be a periodic set with a period p, and let j be an element of $\tilde{S}$. The set $S' = \{i \mid i \in S \wedge j \leq i < j + p\}$ is called a *repeating subset* of $\tilde{S}$. Then, $\{S'\}_p$ is a compact representation, called *base representation*, of $\tilde{S}$. For example, $\{15, 30\}_{60}$ represents the periodic set $\{\ldots,$ -30, 15, 30, 75, $\ldots\}$.

Many real-world temporal repetitions are related to time granularities such as minutes, hours, days, weeks, months and years. It was shown [3] that multi-granular compact representations are more compact than equivalent base representations that use no granularities. A *time granularity* G is a pair $(\mathcal{L}_G, \mathcal{M}_G)$, where $\mathcal{L}_G \subseteq \mathbb{Z}$ is an index set and $\mathcal{M}_G$ is a mapping from $\mathcal{L}_G$ to non-empty subsets of T such that for all $i, j \in \mathcal{L}_G$, if $i < j$ then all elements in $\mathcal{M}_G(i)$ are smaller than all elements in $\mathcal{M}_G(j)$. We denote time granularities with 3 letters, e.g., "hou" for hours, "wee" for weeks, "mth" for months, "hol" for holidays, etc.

A granularity G *groups into* a granularity H, denoted as $G \trianglelefteq H$, if for each $j \in \mathcal{L}_H$, there exists a subset S of $\mathcal{L}_G$ such that $\mathcal{M}_H(j) = \bigcup_{i \in S} \mathcal{M}_G(i)$. For instance, min $\trianglelefteq$ hou and hou $\trianglelefteq$ day. A granularity G *groups periodically into* a granularity H if $G \trianglelefteq H$ and there exists $R \in \mathbb{N}$, such that for all $i \in \mathcal{L}_H$, $\{i\}_R \subseteq \mathcal{L}_H$ and $\bigcup_{k \in \{i\}_R} \mathcal{M}_G(k)$ is a periodic set. For example, days group periodically into years with a period of 400 years.

We constraint the set of all possible granularities, similarly to [6], assuming the existence of a granularity B, that groups periodically into all granularities with an infinite index set and groups into all granularities with a finite index set. In addition, we assume there is a granularity O with $\mathcal{L}_O = \{0\}$ and $\mathcal{M}_O(0) = T$.

A *conversion* from a granularity G to a granularity H, denoted $\downarrow^{G,H}$, is a mapping from $\mathcal{L}_G$ to $\mathcal{L}_H$. There are several reasonable ways to define $\downarrow^{G,H}$ [4, 6, 7]. We use subscript indexes to distinguish between different conversions, e.g., $\downarrow_0^{G,H}$, $\downarrow_1^{G,H}$, etc.

3 Multislices and Repeating Trips

We represent public transport schedules as a set of repeating trips. A repeating trip is composed of a relative trip and a mixed recurrence. A mixed recurrence is composed of several multislices which describe positive occurrences (rules) and negative occurrences (exceptions). A multislice is a compact representation of a finite or an infinite periodic subset of T which uses multiple time granularities.

3.1 Multislices

The elements of T can be specified with sequences of pairs $((G_1, l_1), \ldots, (G_n, l_n))$, where G_i are granularities and l_i are elements of $\mathcal{L}_{G_i}$. This observation was first used in [5]. For example, a sequence $((\mathrm{wee}, 27), (\mathrm{day}, 6), (\mathrm{hou}, 7), (\mathrm{min}, 44))$ specifies the elements of T belonging to a minute 44 of hour 7 on Sunday of week 27. In such a sequence, the first pair selects an index at the given granularity level (e.g., week 27) and each next pair selects granules at their granularity level *relatively* to the previous pair (e.g., day 6 of week 27). For evaluating such sequences, termed *vector labels* [6], it is necessary to know how a granularity conversion is defined and how the relative indexes are translated to absolute indexes. For example, day 6 of week 27 has an absolute index 195.

For two granularities G, H, a *translation*, $\Delta^{G,H}$, is a function from $\mathcal{L}_G \times \mathcal{L}_H$ to $\mathcal{L}_H$, such that, given an absolute index of G and a relative index of H, $\Delta^{G,H}$ returns an absolute index of H. There are several reasonable ways to define $\Delta^{G,H}$ [5]. We use subscript indexes to distinguish between different translations, e.g., $\Delta_0^{G,H}$, $\Delta_1^{G,H}$, etc. Having defined $\Delta^{G_i,G_{i+1}}$ for $i \in [1, n-1]$, a vector label $((G_1, l_1), \ldots, (G_n, l_n))$ specifies a subset of T equal to $\mathcal{M}_{G_n}(\Delta^{G_{n-1},G_n}(\Delta^{G_{n-2},G_{n-1}}(\ldots, l_{n-1}), l_n))$.

Definition 1 A *multislice* is a labeled tree where both nodes and edges have labels. For each node u, the label is a pair (G_u, X_u), where G_u is a granularity and $X_u \subseteq \mathcal{L}_{G_u}$ is a selector. For each edge (u, v), the label is an index k of some translation $\Delta_k^{G_u,G_v}$.

The semantics of multislices is definable through a mapping to sets of vector labels. Each path $((G_1, X_1), \ldots, (G_n, X_n))$ from the root to a leaf in a multislice defines a set of vector labels $((G_1, l_{1,i}), \ldots, (G_n, l_{n,i}))$, where $\forall j \in [1, n] : l_{j,i} \in X_j$.

To represent our bus schedule we use only translations $\Delta_0^{G,H}$ defined as $\Delta_0^{G,H}(i, j) = \downarrow_0^{G,H}(i) \cap \{k \mid k = j + \delta(i)\}$, where $\downarrow_0^{G,H}(i) = \{j \mid \mathcal{M}_H(j) \subseteq \mathcal{M}_G(i)\}$ and $\delta(i) = \min(\downarrow_0^{G,H}(i))$ if $\min(\downarrow_0^{G,H}(i)) \neq -\infty$ and 0 otherwise. For example, $\Delta_0^{\mathrm{wee},\mathrm{day}}(27, 6) = 195$. We assume that for all nodes u, X_u is either finite or infinite periodic subset of $\mathcal{L}_{G_u}$. Since we use the same $\Delta^{G,H}$, we do not show the edge labels in the examples.

Example 2 *Figure 3 shows a multislice representation of buses on weekdays from Monday to Saturday from Example 1. A path ((wee, $\mathcal{L}_{wee}$), (day, [0-5]), (hou, {7}), (min, {19, 30, 45})) describes the following infinite set of vector labels.*

$$\{\ldots, ((\mathrm{wee}, 0), (\mathrm{day}, 0), (\mathrm{hou}, 7), (\mathrm{min}, 19)),$$
$$((\mathrm{wee}, 0), (\mathrm{day}, 0), (\mathrm{hou}, 7), (\mathrm{min}, 30)),$$
$$((\mathrm{wee}, 0), (\mathrm{day}, 0), (\mathrm{hou}, 7), (\mathrm{min}, 45)),$$
$$((\mathrm{wee}, 0), (\mathrm{day}, 1), (\mathrm{hou}, 7), (\mathrm{min}, 19)), \ldots\}$$

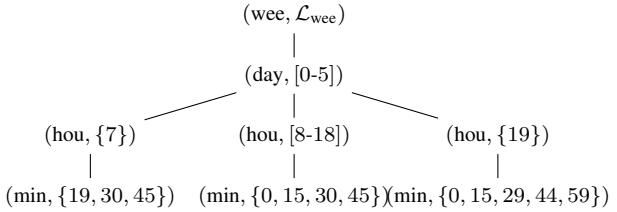

Figure 3. Multislice λ_1.

It can be shown that having our assumptions, a multislice can represent a finite subset of T, an infinite periodic subset of T, or a union of both.

3.2 Mixed Recurrences

In this section we define semi-periodic temporal repetitions which is a formal object to describe real-world schedules. We then define mixed recurrences which is a compact representation of semi-periodic temporal repetitions. A temporal repetition is semi-periodic if the integer indexes of the associated subset of T form a semi-periodic set.

Definition 2 A set $S \subseteq \mathbb{Z}$ is semi-periodic if it can be represented as a set algebra expression $(\tilde{S}_1 \setminus F_1) \cup (\tilde{S}_2 \setminus F_2) \cup \cdots \cup (\tilde{S}_{n-1} \setminus F_{n-1}) \cup F_n$ for some finite n. Here, $\tilde{S}_1, \ldots, \tilde{S}_{n-1}$ are periodic sets and $F_1, \ldots, F_n$ are finite sets.

For example, set $\{\ldots, -30, 15, 30, 75, \ldots\} \setminus \{15\}$ is semi-periodic.

Some semi-periodic sets cannot be expressed with less than n components which follows from the formula $(\tilde{S}_1 \setminus F_1) \cup (\tilde{S}_2 \setminus F_2) = ((\tilde{S}_1 \ominus \tilde{S}_2) \setminus (F_1 \cup F_2)) \cup ((S_1 \cap S_2) \setminus (F_1 \cap F_2))$. Here, $\tilde{S}_1, \tilde{S}_2$ are any periodic sets, F_1, F_2 are any finite sets and $\tilde{S}_1 \ominus \tilde{S}_2 = (\tilde{S}_1 \setminus \tilde{S}_2) \cup (\tilde{S}_2 \setminus \tilde{S}_1)$. However, the majority of schedules, including the schedule from Example 1, have simple form $(\tilde{S}_1 \setminus F_1) \cup F_2$.

Definition 3 A *mixed recurrence* is a finite set of pairs $(\lambda_\mathrm{r}, \lambda_\mathrm{e})$, where both $\lambda_\mathrm{r}, \lambda_\mathrm{e}$ are multislices and λ_r represents rules and λ_e represents exceptions.

Example 3 *The departure times at the station "Kardaun" in our running example can be represented with a mixed recurrence* $\rho_1 = \{(\lambda_1, \lambda_2), (\lambda_3, \lambda_4), (\lambda_5, \lambda_6)\}$. *Figure 4 illustrates the mixed recurrence as a tree, where the root node is labeled with the union operator and the first level nodes are labeled with the set difference operator; multislice* λ_1 *is illustrated in Fig. 3.*

Similar as for multislices, a mixed recurrence evaluates to a set of time points. Let $\mathcal{I}(\lambda)$ denote the time points represented by a single multislice λ. The time points represented by the mixed recurrence $\{(\lambda_{\mathrm{r},1}, \lambda_{\mathrm{e},1}), \ldots, (\lambda_{\mathrm{r},n}, \lambda_{\mathrm{e},n})\}$ are defined as $(\mathcal{I}(\lambda_{\mathrm{r},1}) \setminus \mathcal{I}(\lambda_{\mathrm{e},1})) \cup \cdots \cup (\mathcal{I}(\lambda_{\mathrm{r},n}) \setminus \mathcal{I}(\lambda_{\mathrm{e},n}))$

Any semi-periodic set can be mapped directly to a mixed recurrence representing each component $(\tilde{S} \setminus F)$ with a pair of multislices. Any mixed recurrence represents a semi-periodic set, because for any periodic sets $\tilde{S}_1, \tilde{S}_2$ and any finite sets F_1, F_2, $(\tilde{S}_1 \cup F_1) \setminus (\tilde{S}_2 \cup F_2) = ((\tilde{S}_1 \setminus \tilde{S}_2) \setminus F_2) \cup (F_1 \setminus (\tilde{S}_2 \cup F_2))$. Note, that $\tilde{S}_1 \setminus \tilde{S}_2$ is always a periodic set and $F_1 \setminus (\tilde{S}_2 \cup F_2)$ is always a finite set.

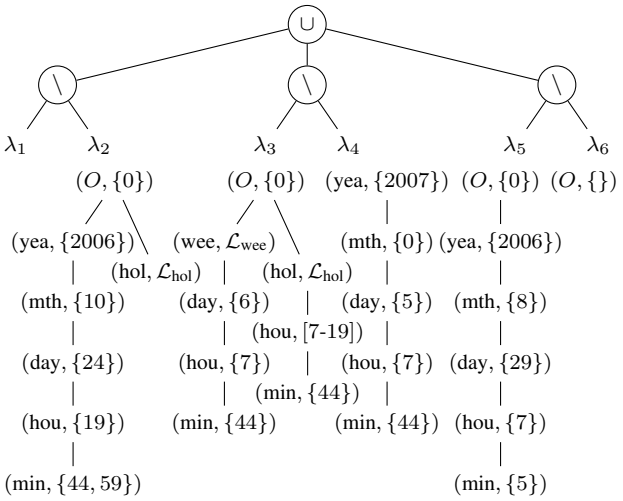

Figure 4. Mixed recurrence ρ_1.

3.3 Repeating Trips

A repeating trip is at the same time a way to represent multiple temporal repetitions at different stations in shorter form and to connect the time instants at different locations, because they describe one spatio-temporal phenomena – a trip of a vehicle.

Definition 4 Let $\mathcal{S}$ be a set of stations. A *trip*, τ, is a finite ordered list, $\tau = ((s_1, t_1), \ldots, (s_m, t_m))$, where $(s_i, t_i) \in \mathcal{S} \times T$ for $i = 1, \ldots, m$ and $t_i < t_{i+1}$ for $i = 1, \ldots, m-1$.

Example 4 *A trip of a bus leaving the station "Kardaun" on 2007-11-24 at 07:19 is (("Kardaun", 2007-11-24-07:19), ("Brennerstr.", 2007-11-24-07:23), ("Bahnhof", 2007-11-24-07:27), ("Sernesiplatz", 2007-11-24-07:30), ("Cadornastr.", 2007-11-24-07:34)).*

Definition 5 Let $\tau = ((s_1, t_1), \ldots, (s_m, t_m))$ be a trip. The *relative trip*, $\bar{\tau}$, of trip τ is defined as $\tau = ((s_1, o_1), \ldots, (s_m, o_m))$ where $o_j = t_j - t_1$ represents the offset (in time) from the departure time at the starting station of the bus.

Example 5 *A relative trip of a bus on the route "Kardaun" – "Cadornastr." is* $\bar{\tau}_1$ *=(("Kardaun", 0), ("Brennerstr.", 4), ("Bahnhof", 8), ("Sernesiplatz", 11), ("Cadornastr.", 15)).*

Definition 6 A *repeating trip* is a pair $(\rho, \bar{\tau})$ where ρ is a mixed recurrence and $\bar{\tau}$ is a relative trip.

Example 6 *The representation of bus line 1 as a repeating trip is given as* $(\rho_1, \bar{\tau}_1)$*, where* ρ_1 *is the mixed recurrence from Example 3 and* $\bar{\tau}_1$ *is a relative trip from Example 5. The represented trips of buses can be obtained by adding each value* $t \in T$ *represented by* ρ_1 *to the offsets of the relative trip* $\bar{\tau}_1$*. For example, having* t_1 *=2007-11-24-07:19 we get a trip in Example 4.*

4 Related Work

Work [6] defines a powerful algebraic language, called *calendar algebra*, for specifying time granularities. In terms of this work a schedule at some station can be specified as a time granularity. Original language is defined on right-infinite time domain, but if we omit this restriction, any mixed recurrence can be translated to an expression of calendar algebra. Compared to calendar algebra, mixed recurrences are more optimized for the representation of schedules and are more "space-friendly". The representation of multislice λ_1 from Fig. 3 in calendar algebra takes 30 operators and 47 constants, comparing to 34 constants using multislices.

An approach used in FBS system [9], which we refer as *day-marking*, is the most common way to represent bus schedules in relational databases. Figure 5 illustrates a fragment of the same schedule as in Fig. 1 represented with day-marking. Here, "W" stands for working days and "+" stands for Sundays and holidays. Column *trip* contains identifiers that link together several tuples into a single trip. Columns *src* and *dst* stand for the source and the destination and show the linking between stations. Columns *dep* and *arr* show departure and arrival times.

trip	src	dst	dm	dep	arr
1	Kardaun	Brennerstr.	W	07:19	07:23
1	Brennerstr.	Bahnhof	W	07:23	07:29
2	Kardaun	Brennerstr.	W	07:30	07:34
2	Brennerstr.	Bahnhof	W	07:34	07:40
3	Kardaun	Brennerstr.	+	07:44	07:48
3	Brennerstr.	Bahnhof	+	07:48	07:54
4	Kardaun	Brennerstr.	W	07:45	07:49
4	Brennerstr.	Bahnhof	W	07:49	07:53
...	...	...	...	...	...

Figure 5. Relational representation of a bus schedule with day-marking approach.

A representation using day markers can be easily translated to multislices having "W" and "+" defined as granularities. For example, the departures at the station "Kardaun" from the first 2 rows in Fig. 5 can be translated to an equivalent repeating trip ($\{(((\mathrm{W}, \mathcal{L}_\mathrm{W}), (\mathrm{hou}, \{7\}), (\mathrm{min}, \{23\})),((O, \{\}))\}$, (("Kardaun", 0), ("Brennerstr.", 4), ("Bahnhof", 10))).

The main problem of day-marking approach is its inability to deal with irregularities. For example, an additional bus on 2007-09-30 at 07:05 can be added introducing a specific day marker just for 2007-09-30. In order to deal with a finite number of cancelled trips one needs to introduce a table of exceptions which is a half-way to the approach we are proposing in this paper.

In some cases the problem of exceptions is solved by generating a flat representation in a time span from 6 months to 2 years. All exceptions are then edited by hand. Such an approach denies the possibility to query the schedules outside the given time span. We made a comparison of sizes of representations of a real-world bus network using repeating trips, day-marking (without finite exceptions), and the flat representation. The network has 334 bus stops in both directions, 14 bus lines and 44 bus routes in total. On each route the buses make from 0 to 68 trips a day. Table 1 summarizes the results of the comparison.

Table 1. Size of representation of schedules of a real-world public transport network.

Representation	No. of tuples	Size (KB)
Repeating trips	44	176
Day-marking	65 800	3 948
Flat (2 years period)	20 189 913	403 798

Work [2] proposes a formalism, called *linear repeating points*, for handling infinite temporal data and uses a train schedule as a running example. A value of the representation consists of two linear functions and a set of simple linear constraints. This combination is powerful enough to express semi-periodic sets, but the lack of granularities leads to huge representations. For example, there are 12 yearly holidays. The period of years is 4 (in the time span 1900-2100). If we take 50 bus routes having 20 stops per route and 100 trips a day, the size of the representation is already $4{\cdot}12{\cdot}50{\cdot}20{\cdot}100 = 4800000$ tuples just for the holidays, not taking into account regular working days, Sundays, Saturdays, school days, etc.

5 Conclusions and Future Work

In this paper we addressed the problem of representing real-world public transport schedules. The main challenges were: (1) the representation of periodic schedules with irregularities, such as cancelled trips or additional trips, and (2) the incorporation of spatial information.

We presented a new data structure, called repeating trip, that fulfills both requirements. We compared the repeating trips with other works on representations of temporal repetitions and found that our approach has some features that are not offered by the others, most importantly the representation of exceptions and the spatial part.

Future work on top of our new data structure includes the design and implementation of algorithms for common operations in the public transport domain and the experimental evaluation of these algorithms.

References

[1] R. Chandra, A. Segev, and M. Stonebraker. Implementing calendars and temporal rules in next generation databases. In *Proceedings of ICDE'94*, pages 264–273, Washington, DC, USA, 1994. IEEE Computer Society.

[2] F. Kabanza, J.-M. Stevenne, and P. Wolper. Handling infinite temporal data. In *PODS*, pages 392–403, 1990.

[3] R. Kasperovics and M. H. Böhlen. Querying multi-granular compact representations. In *DASFAA*, pages 111–124, 2006.

[4] M. C. Keet. *A Formal Theory of Granularity*. Phd thesis, Free University of Bozen-Bolzano, Italy, April 2008.

[5] B. Leban, D. D. McDonald, and D. R. Forster. A representation for collections of temporal intervals. In *Proceedings of AAAI'86*, pages 367–371, August 1986.

[6] P. Ning, X. S. Wang, and S. Jajodia. An algebraic representation of calendars. *Ann. Math. Artif. Intell.*, 36(1-2):5–38, 2002.

[7] H. J. Ohlbach. Periodic temporal notions as 'tree partitionings'. Forschungsbericht/research report PMS-FB-2006-11, Institute for Informatics, University of Munich, 2006.

[8] P. Terenziani. Symbolic user-defined periodicity in temporal relational databases. *IEEE Trans. Knowl. Data Eng.*, 15(2):489–509, 2003.

[9] C. Weber, D. Brauer, V. Kolmorgen, M. Hirschel, S. Provezza, and T. Hulsch. *Fahrplanbearbeitungssystem FBS – Anleitung*. iRFP, September 2006.