Towards Compact and Tractable Automaton-based Representations of Time Granularities

Ugo Dal Lago[1], Angelo Montanari[2], and Gabriele Puppis[2]

[1] Dipartimento di Scienze dell'Informazione, Università di Bologna
Mura Anteo Zamboni 7, 40127 Bologna, Italy
dallago@cs.unibo.it

[2] Dipartimento di Matematica e Informatica, Università di Udine
via delle Scienze 206, 33100 Udine, Italy
{montana,puppis}@dimi.uniud.it

Outline

- Time Granularities
- Representation Formalisms
- String-based and Automaton-based Approaches
- A Relevant Problem: The Granule Conversion Problem
- Optimizing Automaton-based Representations

Time Granularities

Motivations:

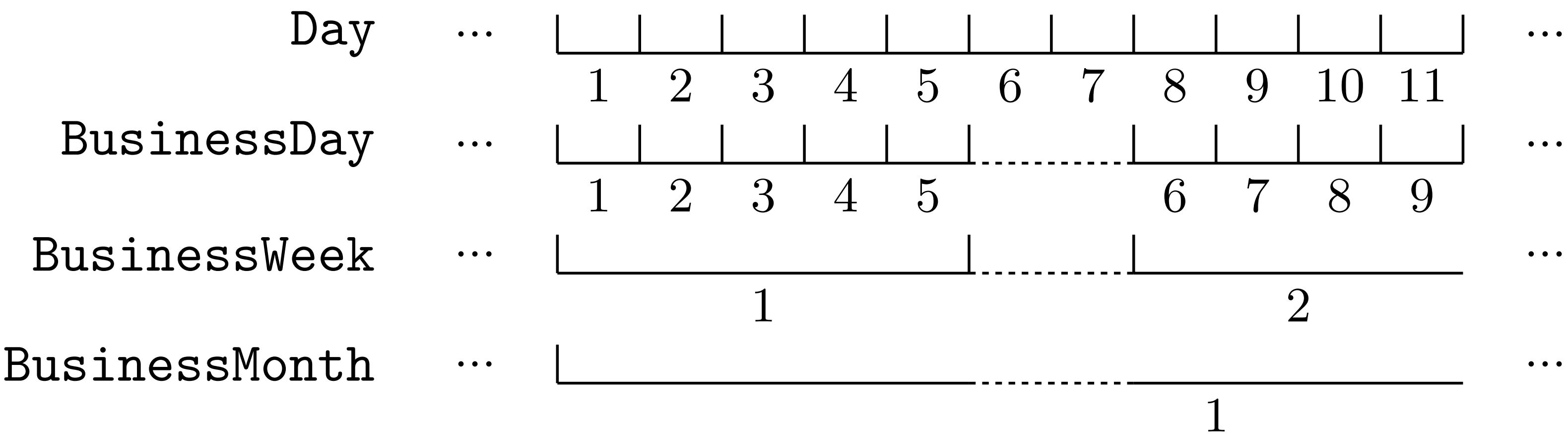
- **relational databases**:
  to express temporal information at different time granularities, to relate different granules and convert associated data (queries)
- **artificial intelligence**:
  to reason about temporal relationships, e.g, to check consistency and validity of temporal constraints at different time granularities (temporal CSPs)
- **data mining**:
  to discover temporal relationships between collected events, to derive implicit information from such relationships

Time Granularities

Definition $G : \mathbb{Z} \rightarrow 2^T$ is a *granularity* iff

- $(T, <)$ is a linearly ordered set of temporal instants,
- $t_x < t_y$ whenever $x < y$, $t_x \in G(x)$, and $t_y \in G(y)$.

A *granule* of G is a non-empty set $G(x)$ and $x \in \mathbb{Z}$ is said to be its *label*.



Representation Formalisms (1)

We cannot finitely represent all the granularities over an infinite temporal domain, so we have to restrict ourselves to a proper subclass of structures.

Possible approaches to model time granularities:

- **algebraic one**:
  relationships between granularities are represented by terms built up from a finite set of operators
  (e.g., $\texttt{Week} = \mathit{Group}_7(\texttt{Day})$ in the Calendar Algebra, see Bettini et al. ’00)
- **logical one**:
  granularities are defined by models of formulas in a given language (e.g., PLTL, see Combi et al., ’02)

Representation Formalisms (2)

- **string-based one**:
  relationships between temporal instants and granules are encoded by sequences of symbols from a given alphabet (e.g., granspecs, see Wijsen ’00)
- **automaton-based one**:
  automata are exploited to encode string-based descriptions of time granularities (e.g., Single-string Automata, see Dal Lago ’01)

We focus our attention on string-based and automaton-based approaches.

Fundamental Problems

- **Equivalence**:
  the problem of establishing whether two different representations define the same granularity
- **Conversion**:
  the problem of relating granules from different time granularities and converting associated data
- **Minimization**:
  the problem of computing the smallest representation(s) for a given granularity
- **Optimization**:
  the problem of computing the representation(s) on which crucial algorithms (e.g., conversion algorithms) run faster

String-based Approach (1)

Basic ingredients:

- a discrete temporal domain T
- restriction to *left bounded periodical* granularities, namely, granularities that have an initial granule and, ultimately, periodically group instants of the temporal domain
- a fixed alphabet $\{\blacksquare, \square, \wr\}$, where
  $\blacksquare$ represents instants covered by some granule,
  $\square$ represents gaps within and between granules,
  $\wr$ separates granules and defines labels

String-based Approach (2)

Example. The infinite word
$\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\square\square\wr\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\square\square\wr\ldots$
represents the granularity `BusinessWeek` in terms of `Day`.

Proposition. *Ultimately periodic words* over $\{\blacksquare, \square, \wr\}$ capture all the left bounded periodical granularities.

Remark. Such strings can be finitely represented by pairs (*granspecs*) of prefixes and repeating patterns.

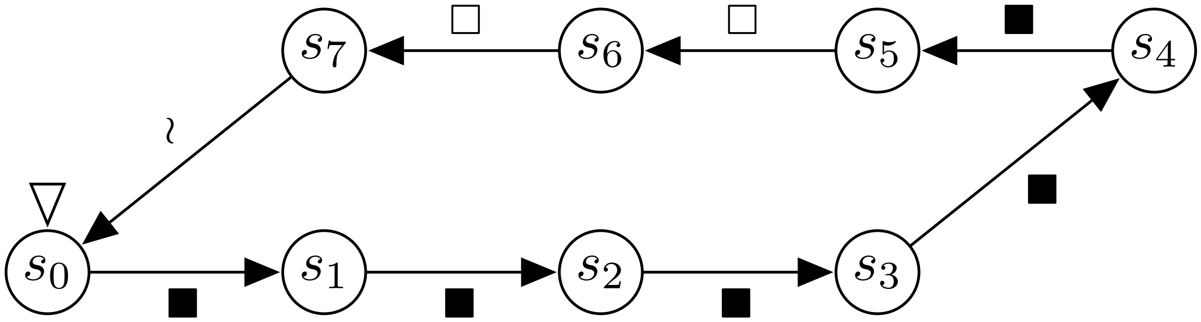
Example. $(\varepsilon, \blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\square\square\wr)$ is a granspec representing `BusinessWeek` in terms of `Day`.

Automaton-based Approach (1)

Connection between ultimately periodic words and automata:

Proposition. A single left bounded periodical granularity can be represented by an automaton recognizing a single string (SSA).

Example. An SSA representing `BusinessWeek` in terms of `Day`.



Problem. As for granspecs, a problem arises: such representations are *too large* with respect to inherently simple structure of granularities.

Automaton-based Approach (2)

Idea. We endow automata with *counters* and we use *primary and secondary transitions* to compactly encode redundancies of generated strings.

Remark. Suitable restrictions on the rules managing counter updates and transition activations guarantee

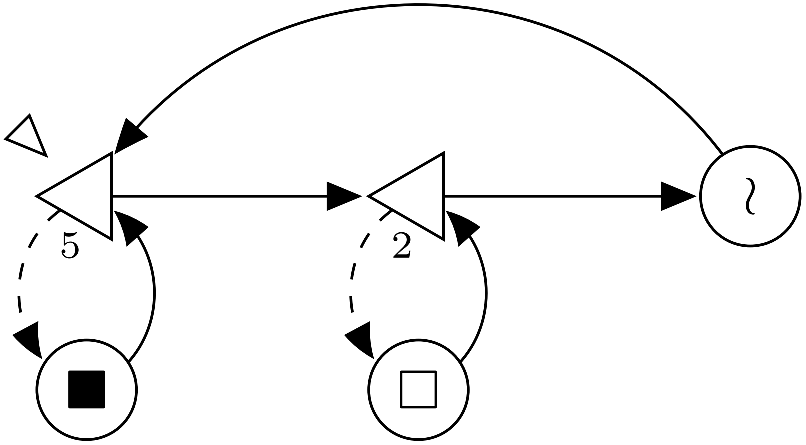
- the *same expressive power of SSA*;
- the *decidability of the equivalence problem*;
- *efficient manipulation* of representations (e.g. granule conversions).

Here the notion of *Restricted Labeled Automaton* (RLA) over finite and ultimately periodic words comes into play.

Automaton-based approach (3)

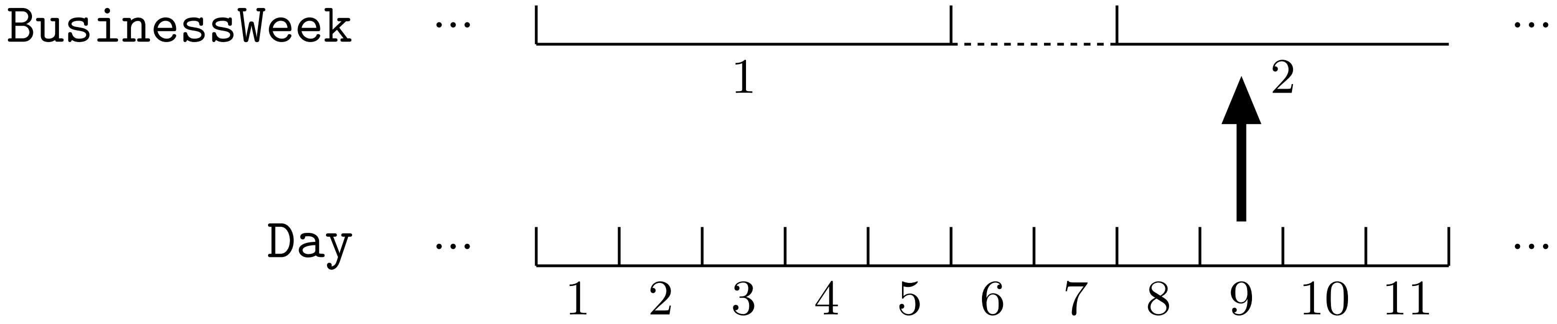
Distinctive features of RLA:

- states are partitioned into labeled and non-labeled states
- each non-labeled state is endowed with a single counter and it is the source of a secondary transition
- secondary transitions are activated iff the value of the the corresponding counter is positive
- counters can only be decremented whenever they are positive, otherwise they are given their initial value



Granule Conversions (1)

Example (Granule Conversion Problem):
compute the label of the business week covering the 9*-th day.*



Granule conversions can be reduced to problems over strings/automata.

Example. In the given example, the solution is given by 1 *plus the number of occurrences of* $\wr$ *in* $(\blacksquare^5\square^2\wr)^\omega$ *until the* 9*-th occurrence of* $\blacksquare$ *or* $\square$ (that is, 2).

Granule Conversions (2)

How RLA can be used to solve the problem?

- **straightforward solution**:
  mimic the automaton transitions until the addressed occurrence has been reached
- **wiser solution**:
  take advantage of nested repetitions in the run of the RLA in order to mimic *maximal periodic sequences* of transitions at once

$\Rightarrow$ The latter algorithm runs in *polynomial time* $\Theta(\|M\|)$, where M is the involved RLA and $\| \ \|$ is a suitable **complexity measure** envisaging the number of states and the structure of the transition functions.

The Optimization Problem

Problem. It is worth *minimizing the complexity* $\| \ \|$ in order to achieve the smallest running time for granule conversion algorithms.

Remark 1. There may be complexity-optimal automata which are not size-optimal, so

$$\textit{optimization problem} \neq \textit{minimization problem}.$$

Remark 2. There may be many different automata which are equivalent and complexity-optimal, so

there isn't a unique solution to the optimization problem.

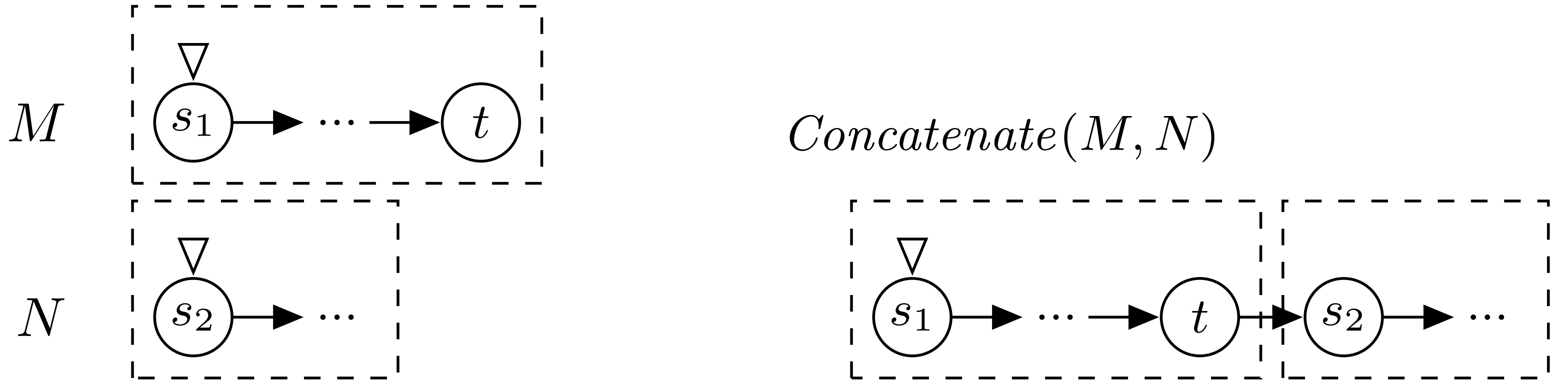
Closure Properties of RLA (1)

Idea. We cope with the optimization problem by using *dynamic programming*.

Proposition. The class of RLA is closed under

- **Concatenation**
  given two RLA recognizing u and v, it generates an RLA recognizing $u \cdot v$
- **Iteration**
  given an RLA recognizing u, it generates an RLA recognizing u^ω
- **Repetition**
  given an RLA recognizing u and a positive integer k, it generates an RLA recognizing u^k

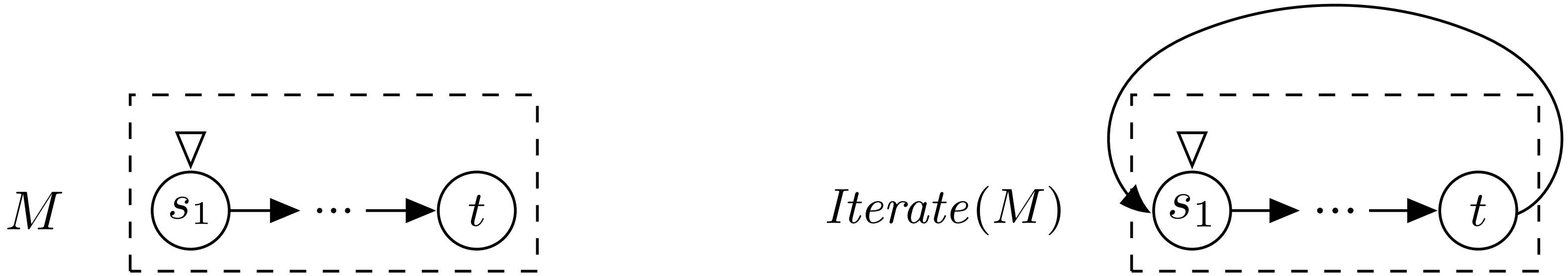
Closure Properties of RLA (2)



M

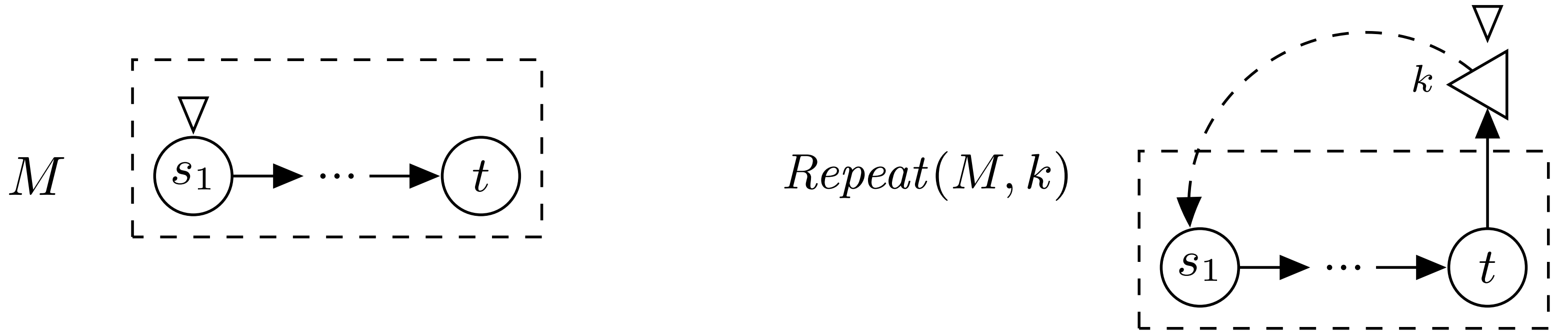
N

$Concatenate(M, N)$



M

$Iterate(M)$



M

$Repeat(M, k)$

Solving the Optimization Problem (1)

Remark. Unfortunately, there exist some RLA that cannot be generated by concatenation, iteration, and repetition.

However...

Proposition. For any RLA M, there exists an *equivalent* RLA M', having the *same complexity*, which can be built starting from basic automata and using concatenation, iteration, and repetition.

$\Rightarrow$ We can restrict ourselves to a proper expressively complete subclass of compound automata, known as *Sharing Free RLA* (SFRLA for short).

Solving the Optimization Problem (2)

Theorem. Given a (finite or ultimately periodic) word u, there is a complexity-optimal SFRLA that recognizes u which is generated by combining smaller complexity-optimal SFRLA recognizing substrings of u.

Furthermore, we can restrict our search only to

a finite number of possible compositions

$\Rightarrow$ We can effectively build a complexity-optimal SFRLA recognizing a given string in a bottom-up fashion .

Such an optimization algorithm takes *polynomial time*.

Conclusions

We defined time granularities and we introduced an alternative automaton-based formalism for representing and reasoning about them.

We identified some crucial problems involving time granularities, with a special attention to the granule conversion problem.

We focused on the efficiency of conversion algorithms and we tackled the optimization of automaton-based representations.

Finally, we exploited non-trivial properties of RLA in order to solve the optimization problem.

It is an open question if a similar approach can be applied to solve the minimation problem.

References

- C. Bettini, S. Jajodia, and X.S. Wang. *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer, 2000.
- C. Combi, M. Francheschet, and A. Peron. *A Logical approach to represent and Reason About Calendars*. Proceedings of the 9th International Symposium on Temporal Representation and Reasoning, IEEE Computer Society, 2002.
- U. Dal Lago and A. Montanari. *Calendars, Time Granularities, and Automata*. Proceedings of the 7th International Symposium on Spatial and Temporal Databases, vol. 2121 of LNCS, 2001.
- U. Dal Lago, A. Montanari, and G. Puppis. *Time Granularities, Calendar Algebra, and Automata*. Technical Report 4, Dipartimento di Matematica e Informatica, Università degli Studi di Udine, 2003.
- U. Dal Lago, A. Montanari, and G. Puppis. *Towards Compact and Tractable Automaton-based Representations of Time Granularities*. Technical Report 17, Dipartimento di Matematica e Informatica, Università degli Studi di Udine, 2003.
- J. Wijsen. *A String-based Model for Infinite Granularities*. Proceedings of the AAAI Workshop on Spatial and Temporal Granularities, AAAI Press, 2000.