Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 10 (2012) 366 – 373

Procedia Computer Science

The 3rd International Conference on Ambient Systems, Networks and Technologies (ANT)

Spatio-Temporal Querying in Smart Spaces

Vivek Menon[a,*], Bharat Jayaraman[b], Venu Govindaraju[b]

[a]*Amrita Vishwa Vidyapeetham, Kochi 682 041, India*
[b]*University at Buffalo, Buffalo, NY 14260, USA*

Abstract

A 'smart space' is one that automatically identifies and tracks its occupants using unobtrusive biometric modalities such as face, gait, and voice in an unconstrained fashion. Information retrieval in a smart space is concerned with information about the location of people at various points in time. Towards this end, we abstract a smart space by a probabilistic state transition system in which each state records the probabilities of presence of a set of individuals who are present in various zones of the smart space. We formulate a data model based upon an occupancy relation with a real-valued probability attribute and describe some of the spatio-temporal queries in SQL and CLP(R), focusing on the computation of probabilities, an aspect that is novel to this model. We define concepts of precision and recall to quantify the performance of this model based on its ability to answer various spatio-temporal queries and discuss results from our experimental prototype.

Keywords: Smart Spaces, Abstract Framework, Biometrics, Recognition, Retrieval, Precision, Recall, Data Model, Spatio-temporal Queries, CLP(R)

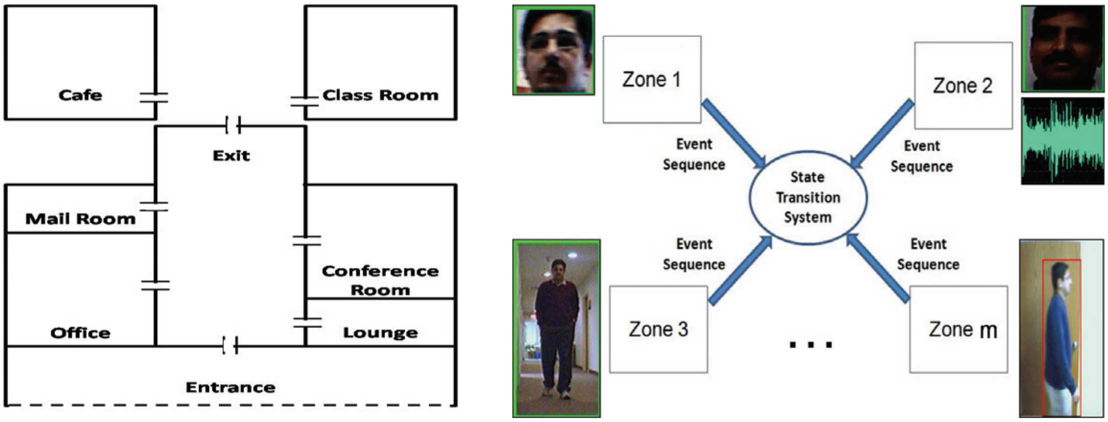
1. Introduction

The goal of our research [1] is to develop indoor smart spaces that can recognize and track their occupants as unobtrusively as possible and answer queries about their whereabouts. The sensors of interest in our work are video cameras and microphones that capture biometric modalities such as face, gait, and voice in an unconstrained fashion. The output of the proposed system would be responses to various spatio-temporal queries such as 'Where was X last seen?', 'What is the probability that Y and Z met in the high-security zone between 6 pm and 7 pm?', etc. Automated approaches to transforming multimedia data into a form suitable for information retrieval is a very challenging problem as it spans multiple research areas. Such queryable smart spaces are very important and beneficial in settings ranging from homes for the elderly or disabled, office workplaces, and in larger areas such as department stores, shopping complexes, train stations, and airports.

*Corresponding Author
Email addresses: `vivek_menon@cb.amrita.edu` (Vivek Menon), `bharat@buffalo.edu` (Bharat Jayaraman), `govind@buffalo.edu` (Venu Govindaraju)

doi:10.1016/j.procs.2012.06.048

This paper is based upon an abstract model of the behavior of a multimodal smart space in terms of a state transition system: states, events, and a transition function [2]. The state captures who is present in the different regions, or zones of the space. The state changes upon an event, i.e., the movement of an occupant from one zone to another. An event abstracts a biometric recognition step - whether it is face recognition, voice recognition, etc. - and is represented as a set of pairs $\langle o, p(o)\rangle$ where $p(o)$ is the probability that occupant o has been recognized at this event. Thus, the state information is also probabilistic in nature. The transition function takes as input a state and an event, and determines the next state by assigning revised probabilities to the occupants based upon the probabilities in the event. Figure 1 depicts the architecture of a smart space.



(a) (b)

Fig. 1: Architecture of a Multimodal Smart Space

The state transition system model provides a natural basis for retrieval of answers in response to various queries about the whereabouts of occupants in the smart space. In this paper, we formulate a data model based upon an occupancy relation with a real-valued probability attribute and show how to formulate spatio-temporal queries using the well-known SQL database query language, focusing on the computation of probabilities, an aspect that is novel to this model. We then show how to formulate more complex queries in a constraint-based extension of logic programs, called CLP(R), which permits general recursive queries and reasoning over real-valued variables and arithmetic operations.

We also formulate precision and recall metrics in a query-dependent manner, since the performance of a smart space is ultimately determined by how well it can respond to the queries that is posed to it. We provide examples for calculation of query-dependent performance metrics based on results from simulation runs using our experimental prototype [3] of an eight-zone university building with 45 registered occupants where each of the frequented areas is mapped as a separate zone and named accordingly entrance, office, mail room, lounge, conference room, classroom, cafeteria and exit as shown in Figure 1a.

Our results confirm that the state transition model serves as a concise abstraction of a smart space and that spatio-temporal querying using CLP(R) is very effective in dealing with the query formulation involving probabilistic data from the state transition system. The rest of this paper is organized as follows. Related work is surveyed in section 2, the details of the data model and query formulation is discussed in section 3, constraint based queries are discussed in section 4. Query-dependent performance metrics are discussed in section 5 and conclusions in section 6.

2. Related Work

The data in smart spaces is fundamentally probabilistic and spatio-temporal in nature since people are moving between different zones over a period of time and we are interested in their trajectories. Hence the data models and query languages of interest in a smart space are probabilistic and spatio-temporal.

There has been considerable research on temporal query languages [4], spatial databases as well as spatio-temporal databases over the past two decades. Location-based systems have been a major driver for the interest in moving object databases (MOD), and their associated data models, query languages, indexing, and uncertainty [5, 6, 7].

In addition to the challenges involved in spatio-temporal databases, research into probabilistic databases has gained momentum over the years due to the emergence of a broad range of applications that need to manage large and imprecise data sets in domains such as sensor networks [8] and various pervasive computing scenarios [9, 10]. The conventional database management systems are incapable of handling large volumes of imprecise data associated with an increasing number of new applications, as imprecision is modeled in a probabilistic manner. The existing rich query languages coupled with some of the event detection engines such as Cayuga [11], SASE [12] or SnoopIB [13] are capable of extracting sophisticated patterns from event streams, though these languages require the data to be precise.

A probabilistic database management system (ProbDMS) [14] stores large volumes of probabilistic data and supports complex queries in addition to the standard features supported by conventional database management systems. Recent work [15, 16, 17] on probabilistic data streams has investigated queries of varying complexity. Extensions to SQL with provision for uncertain matches and ranked results have been proposed in [18, 19], though with certain restrictions.

Our research on querying in smart spaces [1] makes crucial use of probabilistic and temporal concepts, while the spatial issues are treated more in a qualitative (symbolic) than a quantitative (geometric) manner.

3. Data Model and Query Formulation

We begin with a presentation of the data model underlying our query language. While the data model is basically relational in nature, it departs from the standard relational model in the use of a real-valued attribute for the probability. Since the underlying events occur at discrete points in time, we assume that time is discretized as a totally ordered set of hour-minute points $(00:00, 00:01, \ldots, 23:59)$ over a 24-hour period. This can be extended to cover multiple days, months, and years in a straightforward way, as necessary.

Definition (State Relation): *Given a space with occupants $O = o_1 \ldots o_n$, zones $Z = z_1 \ldots z_m$, and biometric events occurring at distinct increasing times $T = t_1 \ldots t_x$, the states of the smart space can be represented as a relation $state(time, occupant, zone, probability)$, where occupant $\in O$, zone $\in Z$, and time $\in T$, where $T \subseteq$ {00:00, 00:01, ..., 23:59}, a discrete totally ordered set of time units. The attribute probability $\in \mathcal{R}$, the set of real numbers, and is functionally dependent on the other three attributes. The state relation satisfies the following integrity constraint: $\forall t\, \forall i\, \Sigma\{p : (\exists z)\, state(t, o_i, z, p)\} = 1$.*

The *state* relation shown in table 1a is based on the states of the smart space. The tuples in the *state* relation correspond to those time units at which the events occur. At each such time t, a state of a smart space is represented by a set of $m \times n$ tuples corresponding to all possible zone-occupant pairs.

state(10:15, o_1, entrance, 0.08)
...
state(10:15, o_5, external, 0.03)

(a) State Relation

occupancy(10:15, 10:19, o_1, entrance, 0.08)
...
occupancy(10:15, 10:19, o_5, external, 0.03)

(b) Occupancy Relation

Table 1: Sample State and Occupancy Relations

Based on the *state* relation, we define an *occupancy* relation that characterizes the data model and forms the basis for the formulation of various queries.

Definition (Occupancy Relation): *Given a smart space with $O = o_1 \ldots o_n$, $Z = z_1 \ldots z_m$, we define an occupancy relation $occupancy(start, end, person, zone, probability)$, where start and end define a time interval. The attribute probability $\in \mathcal{R}$, the set of real numbers, and is functionally dependent on the other* *four attributes.*

Table 1b is a snapshot of the *occupancy relation*. This relation satisfies the integrity constraint that, for any given occupant o and time-interval, the sum of o's probabilities across all zones = 1 for this time-interval.

We first formulate a couple of simple queries in SQL (Structured Query Language) and then discuss in the next section the use of CLP (constraint logic programming) for more complex queries. Our focus will be on queries involving the computation of probabilities, as this is the novel part of the work. Below is the syntax of the most basic form of SQL queries:

SELECT *attributes* FROM *relations* WHERE *condition*

We will use the *occupancy* relation defined earlier as the basis for formulating queries. The tuples of the relations that satisfy the *condition* are selected and the relevant attributes are returned as the result. The *condition* is typically a conjunction of simpler tests that serve as a basis for tuple selection. There are numerous extensions to the basic syntax outlined above, in order to perform aggregate operations, grouping, ordering, etc.

Query 1. What is the probability that an occupant o_3 was in the lounge at 3:00 pm?
Answer: For this query, it is necessary to check the occupancy relation for the presence of o_3 in the lounge at a time interval that contains 3:00 pm. If o_3 was detected to be in the lounge, at most one tuple in occupancy will satisfy the query; otherwise the query returns without any answer. For simplicity, we assume overloaded comparison operators <=, <, =, etc., that are defined on time values.

```
SELECT  prob
FROM    occupancy
WHERE   person = o3 and
        zone = lounge and
        from <= 15:00 <= to
```

Query 2. What is the probability that occupant o_7 was in the lounge during 10:00 am to 11:00 am?
Answer: The probability that an occupant was not in the lounge at a given time is the sum of the probabilities that he was in one of the other zones at this time (since the sum of the probabilities across all zones = 1 at any given time). Since there could be multiple sub-intervals within 10:00 am to 11:00 am during which o_7 was in the lounge (with different probabilities), the answer to the query is 1 minus the product of the probabilities that he was not in the lounge during every such sub-interval.

```
(1 - PROD(
SELECT  SUM(prob) as sumprob
FROM    occupancy
WHERE   person = o7 and
        zone ≠ lounge and
        10:00 <= from and to <= 11:00
GROUP BY from)
)
```

4. Constraint Based Queries

Logic programs offer a more expressive query paradigm than SQL because they permit the formulation of general recursive queries. SQL offers the guarantee that all queries will terminate, an important requirement for a database query language. The subset of Horn clauses called Datalog, which is essentially Horn clauses without function symbols [20], also has the strong termination property and has been studied extensively in the literature. Since our underlying data model uses a real-valued attribute for probability along with operations for comparison and arithmetic, it is more natural to adopt the paradigm of constraint logic programming over reals, CLP(R) [21], rather than Horn clauses. Essentially, CLP(R) extends Horn clauses by generalizing unification to constraint satisfaction. Typically, CLP(R) systems provide solvers for linear equalities and inequalities; non-linear equations and inequations are deferred until one or more variables become bound and they become linear. They also support aggregation predicates, such as *min*, *max*, *sum*, *count*, etc., and we will make use of such operations in our formulations as well.

Definition (CLP(R)): *A CLP(R) program is a collection of rules, which are one of two forms:*

$p(\bar{t})$
$p(\bar{t}) :- p_1(\bar{t}_1) \ldots p_k(\bar{t}_k)$

where each p is a user-defined predicate and each $p_1 \ldots p_k$ may be user-defined or may be one of a pre-defined set of builtin constraint predicates, such as $\leq$, $\geq$, etc. The terms $\bar{t}$ and $\bar{t}_i$ for $1 \leq i \leq k$ include ordinary terms as in Horn clauses as well as terms composed from real numbers, variables, and the usual arithmetic operators.

Note that <= and => are overloaded operators and we use them in this paper for also comparing time units.

Query 3. What is the probability that o_3 and o_5 met in the lounge today? Assume that "met" means "being in the same zone at the same time".
Answer: The probability that o_3 and o_5 met in the lounge today is 1 minus the probability that o_3 and o_5 did not meet in the lounge. For any interval, the probability of not having met in the lounge during this interval is 1 minus product of the probabilities of their being in the lounge during this interval – predicate q3 returns this probability for every overlapping interval.

```
query3(1-Prob) :-
        prod(P, q3(P), Prob).
q3(1 - Prob1*Prob2) :-
        occupancy(From1, To1, o3, lounge, Prob1),
        occupancy(From2, To2, o5, lounge, Prob2),
        overlaps(From1,To1, From2, To2)
overlaps(F1,T1,F2,T2) :-
        F1 <= F2, F2 <= T1.
overlaps(F1,T1,F2,T2) :-
        F2 <= F1, F1 <= T2.
```

Query 4. What is the longest contiguous duration during which occupant o_1 was in the office?
Answer: We define the contiguous occupancy in a zone recursively and use this definition in order to define the longest duration.

```
query4(Duration) :-
        setof(F,occupancy(F,_,o1,office,_), FromSet),
        max(D,q4(FromSet,D), Duration).
q4(FS,D) :-
        member(F,FS),
        contiguous(F,T,o1,office),
        D = T - F.
contiguous(F,T,P,Z) :-
        occupancy(F,T1,P,Z, _),
        T2 = T1 + 0:01,
        contiguous(T2,T,P,Z).
contiguous(F,T,P,Z) :-
        \+ occupancy(F,_,P,Z,_),
        T = F - 0:01.
```

It is straightforward to extend the above definition of `contiguous` so that the average probability during this period is also included. Other extensions include the incorporation of distances between adjacent zones and spatial queries that make use of this distance information. As can be seen from the above formulations and possible extensions, the CLP(R) is a powerful paradigm for probabilistic spatio-temporal queries.

5. Query-dependent Performance Metrics

The query-dependent characterization involves evaluating the performance of the model from an information retrieval perspective based on its ability to answer spatio-temporal queries about the space and its occupants. While query-independent [3] is holistic and involves performance characterization at a system level, the scope of query-dependent performance characterization is restricted to the spatio-temporal dimensions that are either explicit or implicit from the query of interest. At a very granular level, the window of interest for evaluating the performance may only concern an occupant's presence in a particular zone at a specific point in time to a more broad level that may concern the entire sequence of states of the state transition model.

The performance metrics for any given query of interest are defined in terms of the ground truth, which is a set of true answers associated with the query. The nature of the response set may vary depending on the type of query posed and may comprise of basic entities or attributes of the smart space such as occupants, zones, probabilities of presence, time of occurrence or derived attributes such as duration of presence, tracks, etc., which are based on relations that can be defined as part of the data model. The response set involving the occupants in a zone is defined in terms of *recognition threshold* θ; only those persons with a probability $\geq \theta$ are assumed to be present. For a state where a person's probability in two or more zones is $\geq \theta$, the zone with the highest probability is taken as the zone of his presence.

Definition (Ground Truth): *Given n occupants* $O=\{o_1 \ldots o_n\}$ *and an event sequence* $e_1 \ldots e_x$*, then the ground truth,* GT*, is a sequence* $o_{i_1} \ldots o_{i_x}$ *where each index* $i_1 \ldots i_x$ *lies in the range* $1 \ldots n$*.*

The ground truth basically states which person was the true occupant in question for each biometric event. We first define occupancy-based precision and recall, as follows.

Definition (Precision Recall - Occupancy based): *Given a space with m zones, n occupants* $O = \{o_1 \ldots o_n\}$*, an event sequence* $E = e_1 \ldots e_x$*, and ground truth* GT*. For an occupancy-based query* Q*, suppose* Rel_o *is the set of relevant occupants that satisfies the query as per* GT*, and* Ret_o *is the set of retrieved occupants as per the data model and occupancy relation. Then,* $precision_o = |Ret_o \cap Rel_o| \,/\, |Ret_o|$ *and* $recall_o = |Ret_o \cap Rel_o| \,/\, |Rel_o|$*.*

This definition can be extended to queries that determine durations or time intervals. Essentially, each time interval $< t_1, t_2 >$ can be regarded as a discrete set of time points $\{t_1,\ t_1 + 0 : 01, \ldots t_2\}$. This leads to the following definition.

Definition (Precision Recall - Time based): *Given a space with m zones, n occupants* $O = \{o_1 \ldots o_n\}$*, an event sequence* $E = e_1 \ldots e_x$*, and ground truth* GT*. For an time-based query* Q*, suppose* Rel_t *is the set of relevant time units that satisfy the query as per* GT*, and* Ret_t *is the set of retrieved time units as per the data model and occupancy relation. Then,* $precision_t = |Ret_t \cap Rel_t| \,/\, |Ret_t|$ *and* $recall_t = |Ret_t \cap Rel_t| \,/\, |Rel_t|$*.*

We discuss the query dependent performance metrics by considering a fairly typical query and evaluating the track based precision recall metrics for varying number of occupants (n = 5, 10, 15, 20, 25) of the smart space. The query in question 'Who all were present in the cafeteria between 10:00 am - 11:00 am?' is an example of a spatio-temporal query and retrieves the set of occupants present in a zone during a specified time interval. After varying the number of occupants for each simulation run while keeping the recognition threshold constant at θ=0.4, we evaluate the performance metrics associated with this query for varying time intervals (8 hour, 4 hour, 2 hour, 1 hour, 30 minutes, 15 minutes) over the day. This process is repeated for all the other zones of the smart space illustrated in figure 1a.

Table 2 summarizes the performance metrics associated with this class of spatio-temporal query for varying number of occupants in the smart space. The average precision and average recall are computed from precision recall values over varying time intervals.

No. of Occ	Avg. Precision	Avg. Recall
5	0.88	0.69
10	0.87	0.77
15	0.87	0.77
20	0.85	0.78
25	0.83	0.69

Table 2: Query-based Performance Metrics for Varying Number of Occupants

6. Conclusion and Future Work

We have presented an approach to processing spatio-temporal queries related to the whereabouts of occupants in a smart space. An important characteristic of such spaces is that the information regarding the whereabouts of its occupants is fundamentally probabilistic in nature due to the uncertainty associated with unconstrained biometric recognition using unobtrusive modalities. Therefore, we model the states of the smart space using an occupancy relation which records the probability that a given individual is present at a certain location for a certain duration. While probabilities are modeled as a real number in our data model, time is modeled discretely reflecting the discrete nature of the underlying events. We formulated probabilistic queries in SQL and CLP(R) to show how information can be retrieved. The state transition system model provides a natural basis for keeping track of the effect of various events that occur in the smart space. The states in turn serve as an effective basis for the retrieval of answers in response to various queries about the whereabouts of occupants in the smart space. We also show how the performance of a smart space can be defined using precision-recall and tailored to the needs of information retrieval.

We have presented a variety of spatio-temporal queries in this paper. As part of our future work, we propose to investigate a class of interesting queries which may be called *what-if queries*, e.g., “If A was known to be in the lounge at noon with certainty, what is the probability that B was also present at that time?” This query cannot be answered without re-initializing the state and occupancy relations. If A was in the lounge at noon with certainty it means that A was present with certainty in the zone preceding his entrance to the lounge. Inductively, we can say that A was present with certainty in all zones in his track leading up to entrance to the security zone at or preceding noon. Thus, we need to define a revised event sequence in which the probability of A is 1.0 for each event in his track leading up to lounge at noon, and the probabilities of all other occupants for each such event is 0. The above query shows the deep interconnection between retrieval, reasoning, and recognition. A ‘what-if’ retrieval query that declares knowledge about an event causes the redefinition of one or more biometric events, thereby triggering the state transition system to compute a new set of states, which the retrieval system uses to determine a new occupancy relation for answering the query.

References

[1] V. Menon, B. Jayaraman, V. Govindaraju, The Three Rs of Cyberphysical Spaces, Computer 44 (9) (2011) 73–79.

[2] V. Menon, B. Jayaraman, V. Govindaraju, Biometrics Driven Smart Environments: Abstract Framework and Evaluation, in: Proc. of the 5th International Conference on Ubiquitous Intelligence and Computing (UIC-08), Springer-Verlag, 2008, pp. 75–89.

[3] V. Menon, B. Jayaraman, V. Govindaraju, Multimodal Identification and Tracking in Smart Environments, Journal of Personal and Ubiquitous Computing 14 (8) (2010) 685–694.

[4] R. Snodgrass, The Temporal Query Language TQuel, ACM Trans. Database Syst. 12 (2) (1987) 247–298.

[5] R. de Caluwe, G. deTre, G. Bordogna (Eds.), Spatio-Temporal Databases: Flexible Querying and Reasoning, Sprin, 2004.

[6] R. Guting, M. Schneider, Moving Object Databases, Morgan Kaufmann, 2005.

[7] N. Pelekis, E. Frentzos, N. Giatrakos, Y. Theodoridis, HERMES: Aggregative LBS via a Trajectory DB Engine, in: Proc. 28th Intl Conf. Management of Data (SIGMOD 08), 2008.

[8] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, W. Hong, Model-driven Data Acquisition in Sensor Networks, in: Proc. of VLDB '04, 2004, pp. 588–599.
[9] S. Das, D. J. Cook, A. Battacharya, T. L. Heierman, The Role of Prediction Algorithms in the MavHome Smart Home Architecture, IEEE Wireless Communications 9 (6) (2002) 77–84.
[10] J. F. McCarthy, T. D. Anagnost, EVENTMANAGER: Support for the Peripheral Awareness of Events, in: Proc. of Second International Symposium on Handheld and Ubiquitous Computing (HUC '00), 2000, pp. 67–96.
[11] A. Demers, J. Gehrke, M. Hong, M. Riedewald, W. White, Towards Expressive Publish/Subscribe Systems, in: Proc. of the 10th International Conference on Extending Database Technology (EBDT '06), 2006, pp. 627–644.
[12] E. Wu, Y. Diao, S. Rizvi, High-performance Complex Event Processing over Streams, in: Proc. of the ACM SIGMOD' 06, 2006, pp. 407–418.
[13] R. Adaikkalavan, S. Chakravarthy, SnoopIB: Interval-based Event Specification and Detection for Active Databases, Data and Knowledge Engineering 59 (1) (2006) 139–165.
[14] N. Dalvi, C. R, D. Suciu, Probabilistic Databases: Diamonds in the Dirt, Communications of the ACM 52 (7) (2009) 86–94.
[15] G. Cormode, M. Garofalakis, Sketching Probabilistic Data Streams, in: Proc. of the SIGMOD '07, 2007, pp. 281–292.
[16] T. S. Jayram, S. Kale, E. Vee, Efficient Aggregation Algorithms for Probabilistic Data, in: Proc. of Symposium on Discrete Algorithm (SODA '07), 2007, pp. 346–355.
[17] C. Re, J. Letchner, M. Balazinksa, D. Suciu, Event Queries on Correlated Probabilistic Streams, in: Proc. of ACM SIGMOD International Conference on Management of Data, 2008, pp. 715–728.
[18] S. Agrawal, S. Chaudhuri, G. Das, A. Gionis, Automated Ranking of Database Query Results, in: Proc. of CIDR '03, 2003.
[19] A. Motro, Vague: A User Interface to Relational Databases that Permits Vague Queries, ACM Transactions on Information Systems 6 (3) (1988) 187–214.
[20] S. Ceri, G. Gottlob, L. Tanca, What You Always Wanted to Know About Datalog (And Never Dared to Ask), IEEE Transactions of Knowledge and Data Engineering 1 (1) (1989) 146–166.
[21] J. Jaffar, S. Michaylov, P. Stuckey, R. Yap, The CLP(R) Language and System, ACM Transactions on Programming Languages and Systems (TOPLAS) 14 (3) (1992) 339–395.