Incremental Data Uncertainty Handling Using Evidence Combination: A Case Study on Maritime Data Reasoning

Mena Habib
Chair Databases
University of Twente
The Netherlands
m.b.habib@ewi.utwente.nl

Brend Wanders
Chair Databases
University of Twente
The Netherlands
b.wanders@utwente.nl

Jan Flokstra
Chair Databases
University of Twente
The Netherlands
flokstra@cs.utwente.nl

Maurice van Keulen
Chair Databases
University of Twente
The Netherlands
m.vankeulen@utwente.nl

***Abstract*—Semantic incompatibility is a conflict that occurs in the meanings of data. In this paper, we propose an approach for data cleaning by resolving semantic incompatibility. Our approach applies a dynamic and incremental enhancement of data quality. It checks the coherency/conflict of the newly recorded facts/relations against the existing ones. It reasons over the existing information and comes up with new discovered facts/relations. We choose maritime data cleaning as a validation scenario.**

I. Introduction

Data cleaning deals with detecting, removing, correcting or handling errors and inconsistencies in data in order to improve its quality. According to [1], there are two types of data incompatibility, structural incompatibility and semantic incompatibility. Structural incompatibility occurs when the corresponding attributes are defined in a different ways in the databases that need to be integrated. Semantic incompatibility occurs when similarly defined attributes take on different values in different databases.

In this paper, we focus on cleaning semantic incompatibility. We represent data quality problems with uncertainty in the data [2]. Our proposed approach applies a dynamic and incremental enhancement of data quality at every insertion of new data. It checks the coherency/conflicts of the newly inserted record against the existing data. Since the data is probabilistic, this amounts to reducing the uncertainty by removing worlds that are semantically incoherent.

It is the aim of this paper to automate the reasoning process by providing facilities to faithfully record facts and observations about vessels and the surrounding uncertainty about them, which allows reasoning about semantic coherency. To facilitate reasoning over uncertain observations and facts we use probabilistic dependency event's tree. Furthermore, for the scalability purpose, we introduces our probabilistic variant of datalog. Datalog is a knowledge representation and query language based on a subset of Prolog. It allows the expression of facts and rules. Rules specify how more facts can be derived from other facts. Our probabilistic variant of datalog allows the expression of uncertain facts and rules through the use of uncertainty annotations. Through the uncertainty annotations several dependency relations can be expressed between the facts and rules.

II. Related Work

A. Data Cleaning

Data cleaning research has been started since the evolution of database systems. Many approaches have been developed to clean data in different ways. In this section, we will focus on the efforts that tackled semantics data cleaning in single data source.

Semantic integrity in databases is discussed by some researchers. Yakout et al. [3] used constraint repair technique to reduce the inconsistency and improve the data quality. Their approach consults the user on the updates that are most likely to be beneficial in improving data quality. Furthermore, it also uses machine learning methods to identify and apply the correct updates directly to the database without the actual involvement of the user on these specific updates. Similar to our approach, Volkovs et al. [4] introduced a continuous data cleaning framework that can be applied to dynamic data and constraint environments. Their approach permits both the data and its semantics to evolve and suggests repairs based on the accumulated evidence to date. The authors use not only the data and the constraints as evidence, but also the past repairs chosen and applied by the user.

B. Maritime Data Management

Most of the maritime data management approaches focus on AIS data cleaning and prediction. Vespe et al. [5] proposed an approach that utilizes historical and real-time AIS data, and aimed at incrementally learning motion patterns without any specific a prior contextual information. Similarly Wijaya and Nakamura [6] used Apache HBase to store, process, and analyze a large amount of spatio-temporal data generated by shipboard AIS transponders with the objective to predict the behavior of ships navigating through heavily trafficked fairways around the gates of busy harbors.

The imperfection and the sources of errors in the maritime domain are discussed by Harati-Mokhtari et al. [7]. Katsilieris et al. [8] address the inference problem of whether a received AIS data are trustworthy or not with the help of radar measurements and other information from the tracking system.

TABLE I: Vessels knowledge base facts about ships called "ZANDER"

ID	Knowledge base ID	Name	IMO
A	7092	ZANDER	7712767
B	56789	ZANDER	N/A
C	178040	ZANDER	N/A
D	226495	ZANDER	N/A
E	400705	ZANDER	8922553

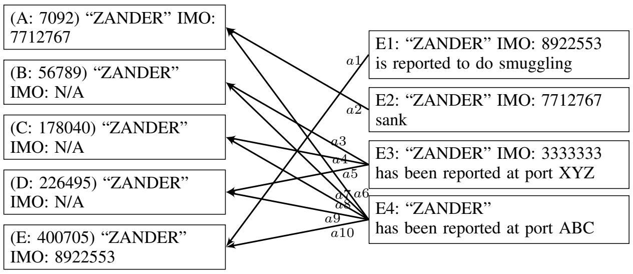
(A: 7092) "ZANDER" IMO: 7712767
(B: 56789) "ZANDER" IMO: N/A
(C: 178040) "ZANDER" IMO: N/A
(D: 226495) "ZANDER" IMO: N/A
(E: 400705) "ZANDER" IMO: 8922553
E1: "ZANDER" IMO: 8922553 is reported to do smuggling
E2: "ZANDER" IMO: 7712767 sank
E3: "ZANDER" IMO: 3333333 has been reported at port XYZ
E4: "ZANDER" has been reported at port ABC
a1
a2
a3
a4
a5
a6
a7
a8
a9
a10

Fig. 1: Global situation representation.

C. *Probabilistic Data Reasoning*

In the past three decades a growing number of probabilistic logics have been proposed and refined. Among these works, ProbLog by De Raedt et al. [9] and probabilistic datalog by Fuhr [10] invite special mention. ProbLog extends Prolog by labeling each clause with a probability. An approximation approach is then used to calculate the probability of each answer. Fuhr's probabilistic datalog is similar in that it allows the attachment of probabilities to clauses. Probability calculations for the answers is done through the principle of inclusion-exclusion.

III. Maritime Scenario

To help maritime decision makers, it is required to integrate data coming from different sources and reason over such diverse data. In this scenario, we simulate a situation where maritime data are flowing from different sources into a central database. The scenario describes a set of observations about vessels at different locations at different times. The idea is to show the contrast in the situation handling between local and global interpretations of the available knowledge.

A. *Vessels' Knowledge base*

Table I shows the portion of the vessels' knowledge base collected from www.vesselfinder.com that represents the basic information about ships with name "ZANDER". For simplicity, we assigned a simple ID to each ship (A, B, .. , E). As we can see, there are 5 vessels having the name "ZANDER". For three of them, the IMO (International Maritime Organization) number is missing. The IMO number is a unique reference for the ship. It should be manually entered at the time of installation of AIS on the vessel. However, the IMO number might have been entered incorrectly [7]. Furthermore, the knowledge base can be also incomplete as in our case.

B. *Events*

The coast guards at port "ABC" have reported a ship with the name "ZANDER" to be passing by. The coast guards want to check if this ship might be suspicious. Given the vessels knowledge base, this ship can be one of five vessels called "ZANDER". The probability that it could be any vessel of the five vessels is $\frac{1}{5}$. This probability represents the local interpretation of the current situation. This local interpretation does not take into account other observations and reports previously made.

Figure 1 summarizes the observations regarding ships called "ZANDER" reported by intelligence surveillance at different locations at different times. The figure shows the possible interpretations of each individual observation. Here, we describe all the reported observations along with their local interpretations.

- One year prior to the coast guards observation at port "ABC", the vessel "ZANDER" with IMO number 8922553 (ID=E) is reported to do smuggling. There is only one possible vessel entity that could be linked to this observation as the IMO number is known in this case. The edge $a1$ links this event to its entity. We will later use these edges in the reasoning process.
- Three weeks prior to the current observation, the vessel "ZANDER" with IMO number 7712767 (ID=A) sank. Again only one possible vessel entity that could be linked to this observation.
- One day prior to the current observation, the vessel "ZANDER" with IMO number 3333333 has been reported to be passing by the port "XYZ". There is no vessel with name "ZANDER" in knowledge base having IMO number 3333333. However, there exist three vessels with unknown IMO number. The edges $a3$, $a4$ and $a5$ link this events to its candidate entities.

IV. Data Cleaning and Coherency Reasoning

To assess the danger of the vessel "ZANDER" that is passing by the port "ABC", we need to reason over the global picture that includes all the observations related to the current situation. The local picture shows that the probability of a threat caused by the observed vessel is zero as there is no information available for the coast guards except the observed name of the vessel. By providing the coast guards with a vessels KB, they find that the observed vessel can be one out of five vessels called "ZANDER". If the coast guards get to know the piece of information that the vessel "ZANDER" with ID=E is reported before to do smuggling activities, the probability of a threat caused by the observed vessel jumps to $\frac{1}{5}$. Providing the coast guards with the complete global picture will help them in resolving the ambiguities and in giving them a better understanding of the situation. The global picture is represented with an event's dependency tree which shows the conditional probabilities and the search space of the dependent events. Dependent events are the ones which affect the search space of other events. For example, the events of type "*a seen vessel*" are dependent on each other because the same vessel can not be seen at two at the same time. Similarly, the events "*sank*" and "*a seen vessel*" are dependent. Dependent events are classified into absolute events and uncertain events. Edge cleaning is done either by eliminating the edge completely or by updating the edge probability.

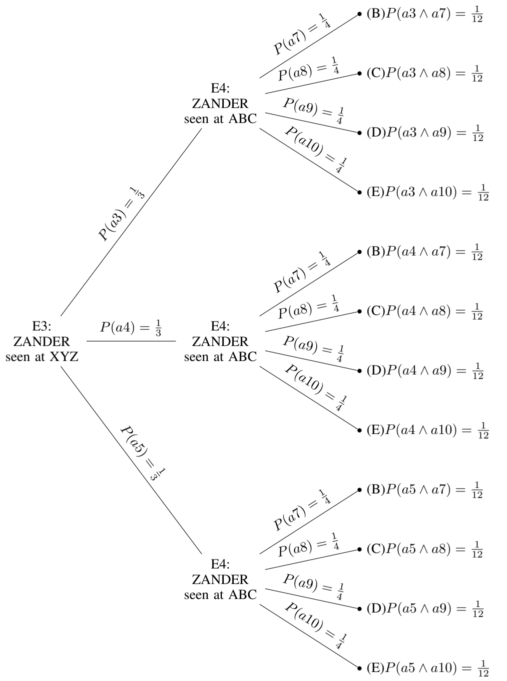
E3:
ZANDER
seen at XYZ
$P(a3) = \frac{1}{3}$
$P(a4) = \frac{1}{3}$
$P(a5) = \frac{1}{3}$
E4:
ZANDER
seen at ABC
$P(a7) = \frac{1}{4}$
$P(a8) = \frac{1}{4}$
$P(a9) = \frac{1}{4}$
$P(a10) = \frac{1}{4}$
(B)$P(a3 \wedge a7) = \frac{1}{12}$
(C)$P(a3 \wedge a8) = \frac{1}{12}$
(D)$P(a3 \wedge a9) = \frac{1}{12}$
(E)$P(a3 \wedge a10) = \frac{1}{12}$
E4:
ZANDER
seen at ABC
$P(a7) = \frac{1}{4}$
$P(a8) = \frac{1}{4}$
$P(a9) = \frac{1}{4}$
$P(a10) = \frac{1}{4}$
(B)$P(a4 \wedge a7) = \frac{1}{12}$
(C)$P(a4 \wedge a8) = \frac{1}{12}$
(D)$P(a4 \wedge a9) = \frac{1}{12}$
(E)$P(a4 \wedge a10) = \frac{1}{12}$
E4:
ZANDER
seen at ABC
$P(a7) = \frac{1}{4}$
$P(a8) = \frac{1}{4}$
$P(a9) = \frac{1}{4}$
$P(a10) = \frac{1}{4}$
(B)$P(a5 \wedge a7) = \frac{1}{12}$
(C)$P(a5 \wedge a8) = \frac{1}{12}$
(D)$P(a5 \wedge a9) = \frac{1}{12}$
(E)$P(a5 \wedge a10) = \frac{1}{12}$

Fig. 2: Events' tree before applying cleaning rules.

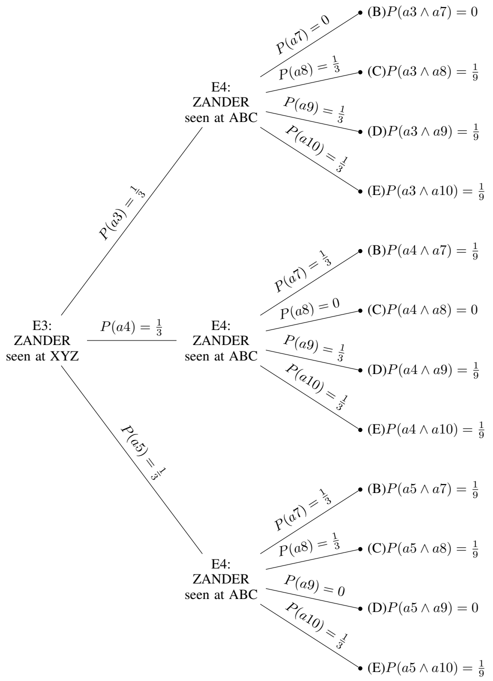

Fig. 3: Events' tree after applying cleaning rules.

*a) **Absolute event::*** is the certain event that is linked to an entity with a probability of 1.0. In our scenario, we have two absolute events; the event *E1* and the event *E2*. The event *E1* is not a dependent event (i.e. it is not conflicting with other events). Hence, no edge cleaning can be done based on this event. The event *E2* is a dependent event as it is of type "*a seen vessel*". Hence, for this event, we apply edge elimination method for data cleaning. The ship that is reported at port "ABC" can not be the same ship that sank three weeks ago. To formalize, $a6 \oplus a2$ (we use the symbol $\oplus$ to represent mutual exclusion). Hence, the edge $a6$ should be removed leaving only four possibilities for the event *E4*. As a consequence, we redistribute the probability mass function of event *E4*. The probability of each of $a7$, $a8$, $a9$, and $a10$ is updated to be $\frac{1}{4}$ instead of $\frac{1}{5}$.

*b) **Uncertain event::*** is the event whose affected entity is uncertain (ambiguous). In our scenario, events *E3* and *E4* are uncertain and dependent. To handle the uncertainty and enhance the data quality we build the events' tree. The process of building the tree and updating the edges probabilities is described as follows:

1) Check the candidate entities of the event of interest. In this case, event *E4* is the event of interest and vessels *B*, *C*, *D* and *E* are the candidate entities.
2) Recursively find all the dependent events that affect the candidate entities of the event of interest. In our case, *E3* is the only dependent event that affects the candidate entities of the event of interest *E4*.
3) Order the dependent events ascendingly according to the time of their occurrence. *E3* is followed by *E4*.
4) Build the events' tree where the root is the earliest event to take place (*E3* in our case) and the children are the next occurring event. We keep building the tree until we reach the event of interest (*E4* in our case). The leaves are the candidate entities of the event of interest. The edges represent the candidate entities of the parent event labeled by the probabilities of each candidate entity. Figure 2 shows the events' tree as described above. The tree shows the 12 possible choice combination of our scenario. From the tree, the probability that the reported vessel at port ABC is the one that is reported before to do smuggling (vessel E) equals $\frac{1}{12} + \frac{1}{12} + \frac{1}{12} = \frac{1}{4}$.
5) Update the probabilities of the tree's edges using the following set of coherency rules that control the search space: a) $a7 \oplus a3$. The ship that is reported at port "ABC" can not be the ship with ID=B if it was the one that has been reported at the port "XYZ" one day ago. This means that if $a3$ is $true$ then $a7$ should be $false$ and hence, either $a8$ or $a9$ or $a10$ is $true$. Similarly, b) $a8 \oplus a4$ and c) $a9 \oplus a5$.
6) After applying the coherency rules, we redistribute the probability mass function to update the probabilities of the tree's edges. Figure 3 shows the events' tree after updating the edges' probabilities.

From the tree, we find that the probability of having a smuggling ship at port "ABC" is $\frac{1}{3}$. As we can see, using only the local situation information we got a probability of *zero* that the ship passing by the port "ABC" is suspicious. by taking into account all the available relevant evidences, the probability jumped to $\frac{1}{3}$ which would give an alert to the coast guards that there is a chance of a smuggling activity, so the guards could take a suitable action.

V. Probabilistic Datalog

Although the aforementioned approach is effective, it is not scalable. This is why we introduce the usage of a probabilistic datalog for better scalability.

A datalog program consists of a set of rules and facts called a knowledge base n. A rule $r = (A^h \leftarrow A_1, \ldots, A_n)$ is a horn clause representing the knowledge that A^h is true if all A_i are true. A fact is a rule without body $(A^h \leftarrow\ )$. We call an atom grounded if it only features constant terms. Semantic entailment for our datalog is defined as the Herbrand Base: all ground atoms that can be derived as a logical consequence from the set of rules.

Our probabilistic datalog is based on the idea of possible worlds. A probabilistic knowledge base is a set of possible worlds $\mathbf{W} = \{W_1, \ldots, W_k\}$ where each world $W \in \mathbf{W}$ is a conventional knowledge base. We extend datalog by labeling rules with a propositional sentence that describes in which possible worlds the rule is present. The atoms of the sentence are of the form $\omega{=}n$. An atom $\omega{=}n$ holds for a world iff the world has the same assignment of value n to variable ω. For example the sentence '$x{=}2 \vee y{=}1$' holds for all worlds that assign 2 to x or 1 to y. A rule is present in a world if the rule's sentence holds for that world.

By attaching a probability $P(\omega{=}n)$ to each specific assignment of value to variable the probability of a possible world can be calculated as the product of the probabilities of the variable assignments for that world. [11] presents a formal overview of our probabilistic datalog. Our preliminary investigation of the probabilistic datalog approach is positive.

VI. Knowledge Representation

Figure 4 shows the representation of our scenario on datalog. The situation representation is split into the following sections:

- Factual data: where all the knowledge base facts are represented.
- Observations: where all the intelligence surveillance observations/reports are represented.
- Reasoning: where a set of rules is defined to resolve the ambiguities and discover new relations/facts. The power of datalog goes here. For example, the 'seen' rule resolves the identity of the observed vessel. The rule 'discover_new_imo' assigns vessels with no IMO to a newly discovered IMO that does not exist in the knowledge base. Data cleaning is done through some rules like 'can_be_in' which assumes that ship may exist in some port if it is not reported at another port and if it is not removed from the knowledge base (because it sank or got out of service).
- Query: represents our information need. The rule 'smuggling' tries to find the probability that the ship reported at some port is doing smuggling activities.

The propositional sentences attached to each rule are noted in '[' and ']', lack of a sentence indicates that the rule holds in all possible worlds. Probability attachments are noted in obvious annotations.

For the current version of datalog, the automatic calculation of the fact probabilities is not available yet. That is why we have to calculate these probabilities outside the datalog and assign them to random variables associated with the rules. Another limitation with the current version is that it does not handle time data due to a lack of arithmetic predicates. Currently, we assume that any observed vessel at some port cannot be existing at another port.

VII. Conclusion and Future Work

In this paper, we presented an approach to resolve semantic incompatibility of data. Our approach builds up evidences needed to resolve the ambiguity of the current situation. It checks the coherency/conflict of new observations against the existing data. It also reasons over the existing information and comes up with new discovered facts/relations to construct a global picture for the current situation. For this purpose, we use a probabilistic variant of datalog that allows for the expression of uncertain facts and rules through the use of uncertainty annotations. We validated our approach on maritime data cleaning scenario. For the future work, we want to automatically calculate fact probabilities inside datalog. We want also to represent, handle and reason over time data.

Acknowledgment

This paper is part of the Federated Security Shield (FedSS) European project funded by ITEA-3.

References

[1] A. Chatterjee and A. Segev, "Data manipulation in heterogeneous databases," *SIGMOD Rec.*, vol. 20, no. 4, pp. 64–68, Dec. 1991. [Online]. Available: http://doi.acm.org/10.1145/141356.141385

[2] M. van Keulen, "Managing uncertainty: The road towards better data interoperability," *IT - Information Technology*, vol. 54, no. 3, pp. 138–146, May 2012.

[3] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas, "Guided data repair," *Proc. VLDB Endow.*, vol. 4, no. 5, pp. 279–289, Feb. 2011. [Online]. Available: http://dx.doi.org/10.14778/1952376.1952378

[4] M. Volkovs, F. Chiang, J. Szlichta, and R. Miller, "Continuous data cleaning," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, March 2014, pp. 244–255.

[5] M. Vespe, I. Visentini, K. Bryan, and P. Braca, "Unsupervised learning of maritime traffic patterns for anomaly detection," in *Data Fusion Target Tracking Conference (DF TT 2012): Algorithms Applications, 9th IET*, May 2012, pp. 1–5.

[6] W. Wijaya and Y. Nakamura, "Predicting ship behavior navigating through heavily trafficked fairways by analyzing ais data on apache hbase," in *Computing and Networking (CANDAR), 2013 First International Symposium on*, Dec 2013, pp. 220–226.

[7] A. Harati-Mokhtari, A. Wall, P. Brooks, and J. Wang, "Automatic identification system (ais): Data reliability and human error implications," *Journal of Navigation*, vol. 60, pp. 373–389, 9 2007. [Online]. Available: http://journals.cambridge.org/article_S0373463307004298

[8] F. Katsilieris, P. Braca, and S. Coraluppi, "Detection of malicious ais position spoofing by exploiting radar information," in *Information Fusion (FUSION), 2013 16th International Conference on*, July 2013, pp. 1196–1203.

[9] L. De Raedt, A. Kimmig, and H. Toivonen, "Problog: A probabilistic prolog and its application in link discovery." in *IJCAI*, vol. 7, 2007, pp. 2462–2467.

[10] N. Fuhr, "Probabilistic datalog: Implementing logical information retrieval for advanced applications," *Journal of the American Society for Information Science*, vol. 51, no. 2, pp. 95–110, 2000.

[11] B. Wanders and M. van Keulen, "Revisiting the formal foundation of probabilistic databases," in *Proceedings of the 9th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-15)*, 2015.

```
% Factual data:
ship(sh7092).
ship_prop(ship_name, sh7092, ”ZANDER”).
ship_prop(ship_imo, sh7092, 7712767).

ship(sh56789).
ship_prop(ship_name, sh56789, ”ZANDER”).

ship(sh178040).
ship_prop(ship_name, sh178040, ”ZANDER”).

ship(sh226495).
ship_prop(ship_name, sh226495, ”ZANDER”).

ship(sh400705).
ship_prop(ship_name, sh400705, ”ZANDER”).
ship_prop(ship_imo, sh400705, 8922553).

port(p1).
port_prop(port_name, p1, ”ABC”).

port(p2).
port_prop(port_name, p2, ”XYZ”).


% Observations:
observe(smuggling_imo, 8922553). % This observation is about a single vessel
observe(sank_imo, 7712767).
observe(seen_imo, 3333333, ”XYZ”).
observe(seen_name, ”ZANDER”, ”ABC”).

% Reasoning:
seen(SH, PORT) :- observe(seen_imo, IMO, PN), ship_prop(ship_imo, SH, IMO), port_prop(port_name, PORT, PN),
     can_be_in(SH,PORT).
seen(SH, PORT) :- observe(seen_imo, IMO, PN), discover_new_imo(SH, IMO), port_prop(port_name, PORT, PN),
     can_be_in(SH,PORT).
seen(SH, PORT) :- observe(seen_name, NAME, PN), ship_prop(ship_name, SH, NAME), port_prop(port_name, PORT, PN
     ), can_be_in(SH,PORT) [e1=1].

@P(e1=1) = 0.5.
@P(e1=2) = 0.5.

sank(SH) :- observe(sank_imo, IMO), ship_prop(ship_imo, SH, IMO).
discover_new_imo(SH, IMO) :- observe(seen_imo, IMO, PORTName), ship(SH), ~has_imo(SH), port_prop(port_name,
     PORT, PORTName).
imo(IMO):- ship_prop(ship_imo, SH, IMO).
imo(IMO):- observe(seen_imo, IMO, PORTName).
known_smuggler(SH):- observe(smuggling_imo, IMO), ship_prop(ship_imo, SH, IMO).
smuggling(SH, PORT):- known_smuggler(SH), seen(SH, PORT).
can_be_in(SH, X):- ~removed(SH), ~seen(SH, Y), X != Y, port(X), port(Y),ship(SH).
has_imo(SH) :- ship_prop(ship_imo, SH, WHATEVER).
removed(SH):-  observe(sank_imo, IMO),ship_prop(ship_imo, SH, IMO).


% Query
smuggling(X, p1)?
```

Fig. 4: Datalog Knowledge Representation