May 1997

The Decidability of Some Restricted Implication Problems for Path Constraints

Peter Buneman
University of Pennsylvania

Wenfei Fan
University of Pennsylvania

Scott Weinstein
University of Pennsylvania, weinstein@cis.upenn.edu

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-97-15.

The Decidability of Some Restricted Implication Problems for Path Constraints

Abstract
In [10], we introduced a path constraint language and established the undecidability of its associated implication problems. In this paper, we identify several fragments of the language, and establish the decidability of the implication and finite implication problems for each of these fragments in the context of semistructured databases. In addition, we demonstrate that these fragments suffice to express important semantic information such as extent constraint, inverse relationships and local database constraints commonly found in object-oriented databases. We also show that these fragments are useful for, among other things, query optimization.

Comments
University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-97-15.

The Decidability of Some Restricted Implication Problems for Path Constraints

Peter Buneman[*] peter@central.cis.upenn.edu

Wenfei Fan[†] wfan@saul.cis.upenn.edu

Scott Weinstein[‡] weinstein@linc.cis.upenn.edu

Department of Computer and Information Science
University of Pennsylvania

May 1997

Abstract

In [10], we introduced a path constraint language and established the undecidability of its associated implication problems. In this paper, we identify several fragments of the language, and establish the decidability of the implication and finite implication problems for each of these fragments in the context of semistructured databases. In addition, we demonstrate that these fragments suffice to express important semantic information such as extent constraint, inverse relationships and local database constraints commonly found in object-oriented databases. We also show that these fragments are useful for, among other things, query optimization.

1 Introduction

The representation of data as a rooted edge-labeled graph has gained enormous popularity recently in semistructured data. It has proven to be useful for a wide range of applications such as integrating heterogeneous data sources (Lorel [3], MSL [22]), querying biological data (UnQL [9]) and querying the Web (W3QS [17], WebSQL [19], STRUQL [14]). See [1] for a survey. The graph in Figure 1, taken from [10], provides an example of such a representation of a database. In the graph, the root node r indicates a (persistent) entry point into the database, the vertices represent data entities, and the edges are labeled with attribute names.

As it stands, the graph representation of data does not provide full information about the structure of the data. In response to this problem, in [10] we presented a class of path inclusion constraints, P, for the graph data model. These path constraints are capable of expressing natural integrity constraints that are a fundamental part of the semantics of the data. For example, by taking edge labels as binary relations, the following semantic relations can be expressed as constraints of P.

Extent Constraints. Given the database depicted in Figure 1, one would expect the following constraints to hold:

$$\forall c\, (\exists s\, (Students(r, s) \wedge Taking(s, c)) \rightarrow Courses(r, c))$$
$$\forall s\, (\exists c\, (Courses(r, c) \wedge Enrolled(c, s)) \rightarrow Students(r, s))$$

[*]This work was partly supported by the Army Research Office (DAAH04-95-1-0169) and NSF Grant CCR92-16122.
[†]Supported by an IRCS graduate fellowship.
[‡]Supported by NSF Grant CCR-9403447.

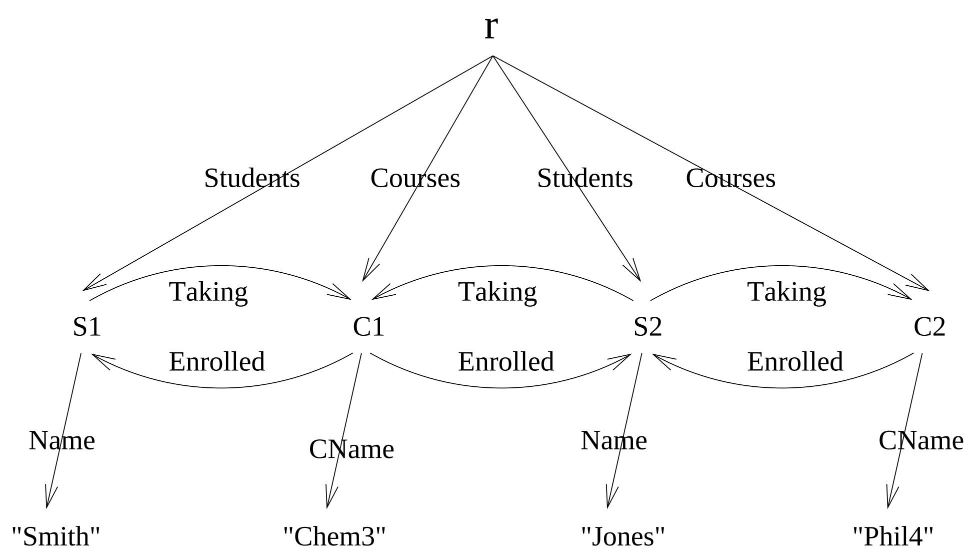

Figure 1: Representation of a student/course database

That is, any course taken by a student must be a course that occurs in the database "extent" of courses, and any student enrolled in a course must be a student that similarly occurs in the database.

Inverse Constraints. The inverse relationship between `Taking` and `Enrolled` is expressed as:

$$\forall\, s\; c\; (Students(r, s) \wedge Taking(s, c) \rightarrow Enrolled(c, s))$$
$$\forall\, c\; s\; (Courses(r, c) \wedge Enrolled(c, s) \rightarrow Taking(s, c))$$

Such constraints are common in object-oriented databases [18, 5, 11].

Local Database Constraints. In database integration one often wants to perform the most trivial integration and include one database as a component of another. Suppose, for example, we want to build a database which is a set, `Schools`, of school databases described above. Now we may want certain constraints to hold on components of this database. For example, the "extent constraints" described above now hold on each member of the set `Schools`. Here we refer to a component database such as a member of the set `Schools` as a *local database* and its constraints as *local database constraints*. Extending our graph representation by adding edges labeled with `Schools` from the new root node to the roots of local databases, the local extent constraints are:

$$\forall\, d\; c\;\; (Schools(r, d) \wedge \exists\, s\, (Students(d, s) \wedge Taking(s, c)) \rightarrow Courses(d, c))$$
$$\forall\, d\; s\;\; (Schools(r, d) \wedge \exists\, c\, (Courses(d, c) \wedge Enrolled(c, s)) \rightarrow Students(d, s))$$

Path inclusion constraints have been studied in [4]. However, the constraints of [4] cannot express, for example, the inverse relationship and the local database constraints described above.

Path constraints of P are useful for a number of reasons. For semistructured data, in particular, these constraints can be used for optimizing queries and for imposing some form of structure on the data.

There has been work in optimization techniques for queries on semistructured data. In [9], a lambda calculus for semistructured data is presented. This yields a framework for graph transformations which, in turn, allows an optimized evaluation of UnQL queries. In [24], a query decomposition method is proposed as an efficient query evaluation strategy on distributed data sources. In [3], extensions to the optimization techniques for generalized path expressions in object-oriented databases developed by [2, 12] are considered for semistructured data. Recently, Abiteboul and Vianu investigated query optimization by using path constraints [4].

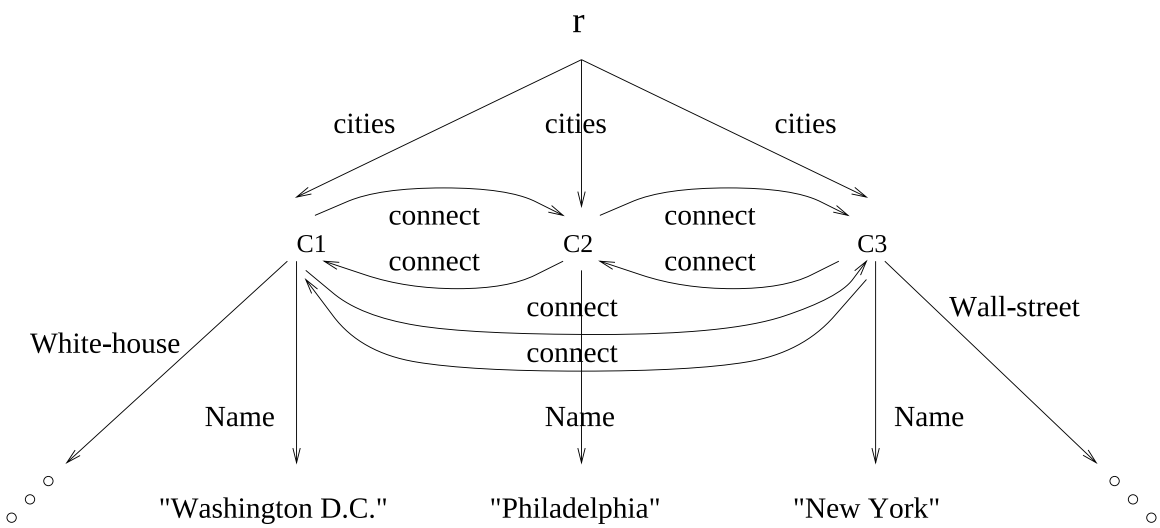

Figure 2: An example tour database

In the spirit of [4], here we demonstrate how to use path constraints of P to optimize queries. Suppose, for example, we want to find all the students taking the course "Chem3" in the database depicted in Figure 1. Given the inverse and extent constraints described above, we are able to express this (in OQL [11] syntax) as:

```
select s
from   r.Courses c, c.Enrolled s
where  c.CName = "Chem3"
```

Without the inverse and extent constraints, one may have to traverse the database extent of students to answer the query.

As another example, consider the tour database in Figure 2. Suppose we want to find all the cities connected to Philadelphia via one or more `connect` edges. Given the path constraints of P below:

$$\forall x\, y \;\; (cities(r,x) \wedge\, \exists z\, (connect(x,z) \wedge connect(z,y)) \rightarrow connect(x,y))$$
$$\forall x\, y \;\; (cities(r,x) \wedge connect(x,y) \rightarrow connect(y,x))$$

we are able to write the query as:

```
select c
from   r.cities p, p.connect c
where  p.Name = "Philadelphia"
```

Without these constraints, it is inevitable to formulate the query in some recursive form. Note that the path constraints given above are not examples of the constraints of [4].

Structural information about semistructured data is useful for query formulation and optimization. It also facilitates browsing of the data. In [8], a schema of a semistructured database is defined by means of graphs and simulation. Using the graph schemas, [15] provides optimization techniques for queries with regular path expressions. The problem of inferring structure in semistructured data is considered in [20, 21]. In [20], an algorithm is developed for approximately classifying objects into a type hierarchy. In [21], an approach to schema discovery by traversing navigation paths is presented.

Path constraints offer another means to add structure to semistructured data. The extent and inverse constraints described above, for example, convey semantics commonly found in object-oriented databases. As another example, consider the following constraints for a Web database of a school:

$$\forall x\, y\, (Dept(r, x) \wedge TA(x, y) \rightarrow Student(x, y))$$
$$\forall x\, y\, (Dept(r, x) \wedge TA(x, y) \rightarrow Employee(x, y))$$
$$\forall x\, y\, (Dept(r, x) \wedge (Student(x, y) \wedge Employee(x, y)) \rightarrow TA(x, y)) \quad (\dagger)$$

Here r indicates the home page of the school, which has links to the home pages of departments in the school. The home page of a department is in turn linked to the home pages of employees, students and teaching assistants in the department. Abusing object-oriented database terms, these constraints indicate that

- TA of a department is a "subclass" of both Student and Employee of the department; and
- the "extent" of TA is the intersection of the "extents" of Student and Employee.

The first two constraints above are in P and the constraint $(\dagger)$ is in P^c, which is a mild generalization of P. Again, these cannot be stated as constraints of [4].

To take advantage of these path constraints, it is desirable to be able to reason about them. However, in [10] we have shown that despite the simple syntax of the constraint language P, its associated implication problem is r.e. complete and its finite implication problem is co-r.e. complete. These undecidability results motivate our search for decidable fragments of P which retain sufficient expressive power to make them of interest from a database perspective. In this paper, we identify several fragments of P which suffice to express at least the constraints we have described above, and we establish the decidability of the implication problems associated with each of these fragments.

The rest of the paper is organized as follows. In Section 2, we recall the formal definition of the path constraint language P from [10], and present a mild generalization of P, P^c. In Sections 3, 4 and 5, we identify several fragments of P which share the following properties. First, they each properly contain the set of word constraints investigated in [4]. Second, each of them fails to be included in two-variable first-order logic, the fragment of first-order logic consisting of all relational sentences with at most two distinct variables. Third, they allow the formulation of many semantic relations which are of interest from the point of view of database theory. And finally, they each possess decidable implication problems. Section 6 shows that the decidability results for these fragments of P also hold for the analogous fragments of P^c.

2 Preliminaries

In this section, we first recall the definitions of the data model and the constraint language P from [10], and then present a mild generalization of P.

We assume the standard notations used in first-order logic [13].

2.1 The data model

In the same spirit of OEM [23, 3] and UnQL [9], we model semistructured databases as rooted edge-labeled directed graphs. These graphs are represented as (finite) first-order logic structures of signature

$$\sigma = (r,\ E),$$

where r is a constant denoting the root and E is a finite set of binary relations denoting the edge labels. The constant r indicates an entry point into the databases.

2.2 Paths

A path can be represented as a logic formula with two free variables.

Definition 2.1: A path is a formula $\alpha(x, y)$ having one of the following forms:

- $x = y$, denoted $\epsilon(x, y)$ and called an *empty path*;
- $K(x, y)$, where $K \in E$; or
- $\exists z(K(x, z) \wedge \beta(z, y))$, where $K \in E$ and $\beta(z, y)$ is a path.

Here the free variables x and y denote the tail and head nodes of the path, respectively. We write $\alpha(x, y)$ as α when the parameters x and y are clear from the context. ■

The *concatenation* of paths $\alpha(x, z)$ and $\beta(z, y)$, denoted $\alpha(x, z) \cdot \beta(z, y)$ or simply $\alpha \cdot \beta$, is defined by:

$$\alpha(x,z) \cdot \beta(z,y) \;=\; \begin{cases} \beta(x,y) & \text{if } \alpha = \epsilon \\ \exists z(K(x,z) \wedge \beta(z,y)) & \text{if } \alpha = K \\ \exists u(K(x,u) \wedge (\alpha'(u,z) \cdot \beta(z,y))) & \text{if } \alpha(x,z) = \exists u(K(x,u) \wedge \alpha'(u,z)) \end{cases}$$

The *length* of path α, $|\alpha|$, is defined by:

$$|\alpha| \;=\; \begin{cases} 0 & \text{if } \alpha = \epsilon \\ 1 & \text{if } \alpha = K \\ 1 + |\beta| & \text{if } \alpha = K \cdot \beta \end{cases}$$

A path ρ is said to be a *proper prefix* of ϱ, denoted $\rho \prec_p \varrho$, iff there exists a path λ such that $\lambda \neq \epsilon$ and $\varrho = \rho \cdot \lambda$.

A path ρ is said to be a *prefix* of ϱ, denoted $\rho \preceq_p \varrho$, iff $\rho \prec_p \varrho$ or $\rho = \varrho$.

Similarly, ρ is said to be a *suffix* of ϱ, denoted $\rho \preceq_s \varrho$, iff there exists λ such that $\varrho = \lambda \cdot \rho$.

2.3 Path constraints

The path constraint language P is formalized as follows.

Definition 2.2: A *path constraint* φ is an expression of either the *forward* form

$$\forall\, x\, y\, (\alpha(r, x) \wedge \beta(x, y) \rightarrow \gamma(x, y)),$$

or the *backward* form

$$\forall\, x\, y\, (\alpha(r, x) \wedge \beta(x, y) \rightarrow \gamma(y, x)),$$

where α, β, γ are paths, called the *prefix*, *left tail* and *right tail* of φ, and denoted by $pf(\varphi)$, $lt(\varphi)$ and $rt(\varphi)$, respectively.

A path constraint is called a *forward constraint* if it is of the forward form, and a *backward constraint* if it is of the backward form.

The set of all path constraints is denoted by P. ■

For example, all the path constraints presented in the last section, except (†), are constraints in the set P.

We call a path constraint φ in P a *simple path constraint* if $pf(\varphi) = \epsilon$. That is, φ is of either the form

$$\forall\, y\, (\beta(r, y) \rightarrow \gamma(r, y)),$$

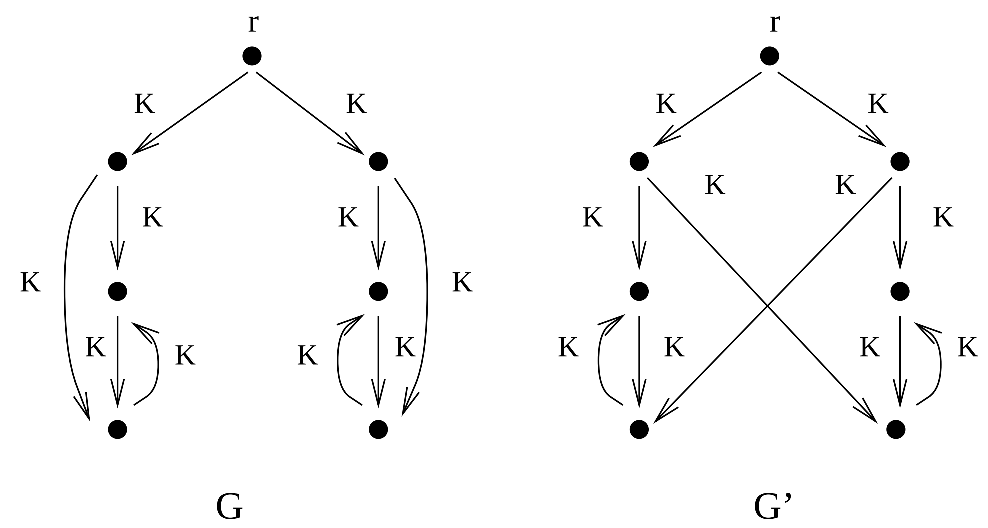

Figure 3: Structures distinguishable by P

or the form

$$\forall y\,(\beta(r,y) \to \gamma(y,r)).$$

The set of all simple path constraints is denoted by P_s.

A proper subclass of simple path constraints, called *word constraints* and denoted by P_w, was introduced and investigated in [4]. A word constraint can be represented as

$$\forall y\,(\beta(r,y) \to \gamma(r,y)),$$

where β and γ are paths.

2.4 Path constraint implication

We borrow the standard notions of model and implication from first-order logic [13].

Let G be a structure and φ a P_c constraint. We use $G \models \varphi$ to denote that G *satisfies* φ (i.e., G *is a model of* φ). Let Σ be a set of P_c constraints. We use $G \models \Sigma$ to denote that G *satisfies* Σ (i.e., G *is a model of* Σ). That is, for every $\phi \in \Sigma$, $G \models \phi$.

Let $\Sigma \cup \{\varphi\}$ be a finite subset of P_c. We use $\Sigma \models \varphi$ to denote that Σ *implies* φ. That is, for every structure G, if $G \models \Sigma$, then $G \models \varphi$. Similarly, we use $\Sigma \models_f \varphi$ to denote that Σ *finitely implies* φ. That is, for every finite structure G, if $G \models \Sigma$, then $G \models \varphi$.

In the context of semistructured databases, the *implication problem* for P is the problem of determining, given any finite set $\Sigma \cup \{\varphi\}$ of sentences in P, whether $\Sigma \models \varphi$. The *finite implication problem* for P is the problem of determining, given any finite subset $\Sigma \cup \{\varphi\}$ of P, whether $\Sigma \models_f \varphi$.

As observed by [4], every word constraint (in fact, every simple path constraint) can be expressed by a sentence in two-variable first-order logic (FO^2), the fragment of first-order logic consisting of all relational sentences with at most two distinct variables (see [16, 7] for in-depth presentations of FO^2). Recently, [16] has shown that the satisfiability problem for FO^2 is NEXPTIME-complete by establishing that any satisfiable FO^2 sentence has a model of size exponential in the length of the sentence. The decidability of the implication and finite implication problems for word constraints (and for simple constraints) follows immediately. In fact, [4] directly establishes (without reference to the embedding into FO^2) that the implication problems for word constraints are in PTIME.

In contrast to word constraints, many path constraints in P are not expressible in FO^2.

Example 2.1: Consider the structures G and G' given in Figure 3. It is easy to verify, using the 2-pebble Ehrenfeucht-Fraïssé style game [6], that G and G' are equivalent in FO^2. However, G and G' are distinguished by the path constraint

$$\varphi = \forall\, x\, y\, (K(r, x) \wedge K(x, y) \rightarrow \exists z(K(x, z) \wedge K(z, y))),$$

because $G \models \varphi$ but $G' \not\models \varphi$. This shows that φ is not expressible in FO^2. ∎

2.5 Conjunctive path constraints

Next, we present a mild generalization of P.

Definition 2.3: A *conjunctive path constraint* ϕ is an expression of either the *forward* form

$$\forall\, x\, y\, (\bigwedge_{\alpha \in A} \alpha(r, x) \wedge \bigwedge_{\beta \in B} \beta(x, y) \rightarrow \gamma(x, y)),$$

or the *backward* form

$$\forall\, x\, y\, (\bigwedge_{\alpha \in A} \alpha(r, x) \wedge \bigwedge_{\beta \in B} \beta(x, y) \rightarrow \gamma(y, x)),$$

where A, B are non-empty finite sets of paths, and are denoted by $pf(\phi)$ and $lt(\phi)$, respectively. Here γ is a path, denoted by by $rt(\phi)$.

The set of all conjunctive path constraints is denoted by P^c. ∎

For example, all the constraints given in the last section, including (†), are constraints of P^c.

Every path constraint of P is a conjunctive path constraint of P^c. As an immediate corollary of the undecidability results established in [10], we have the following.

Corollary 2.1: The implication problem for P^c is r.e. complete, and the finite implication problem for P^c is co-r.e. complete. ∎

3 Prefix restricted implication

In this section, we establish the decidability of a restricted form of the implication problems for P.

3.1 Definition

The implication problems for simple path constraints, which are known to be decidable, can be viewed as a restricted form of the implication problems for P. More specifically, the implication problems for P_s are the implication problems for P under the following restriction: for any finite subset of P in the implication problems, the prefix of each constraint in the subset is the empty path.

By replacing this prefix restriction with a weaker one, we define the prefix restricted implication problems for P as follows.

Definition 3.1: A *prefix restricted subset* of P is a finite subset of P in which the prefixes of all the constraints have the same length.

The *prefix restricted (finite) implication problem for* P is the problem of determining, given any prefix restricted subset $\Sigma \cup \{\varphi\}$ of P, whether all the (finite) models of Σ are also models of φ. ∎

Obviously, the implication problems for word constraints are special cases of the prefix restricted implication problems for P. Moreover, in contrast to word constraint implication, prefix restricted implication cannot be stated in two-variable first-order logic. A convenient argument for this is that

$\{\varphi\}$, where φ is the constraint given in Example 2.1, is a prefix restricted subset of P. However, φ is not expressible in FO^2.

Many cases of integrity constraint implication commonly found in databases are examples of the prefix restricted implication problem for P. Among these are implications for inverse constraints and local database constraints. As an example, consider the set consisting of the two local inverse constraints in the school databases described in Section 1:

$$\forall\, s\ c\ (\exists\, d\ (Schools(r,d) \wedge Students(d,s)) \wedge Taking(s,c) \to Enrolled(c,s))$$
$$\forall\, c\ s\ (\exists\, d\ (Schools(r,d) \wedge Courses(d,c)) \wedge Enrolled(c,s) \to Taking(s,c))$$

and the constraint

$$\forall\, s_1\ s_2\ (\exists\, d\ (Schools(r,d) \wedge Students(d,s_1)) \wedge \epsilon(s_1,s_2) \to \exists\, c\ (Taking(s_1,c) \wedge Enrolled(c,s_2))).$$

This set is a prefix restricted subset of P.

Another example of prefix restricted implication is the implication of the constraint

$$\forall\, x\ y\ (cities(r,x) \wedge \exists\, z\ (connect(x,z) \wedge connect(z,y)) \to connect(y,x))$$

from:

$$\forall\, x\ y\ (cities(r,x) \wedge \exists\, z\ (connect(x,z) \wedge connect(z,y)) \to connect(x,y))$$
$$\forall\, x\ y\ (cities(r,x) \wedge connect(x,y) \to connect(y,x))$$

3.2 Decidability

We next establish the following theorem.

Theorem 3.1: The prefix restricted implication and finite implication problems for P are decidable. ∎

The idea of the proof is to show that the satisfiability and finite satisfiability problems for the set

$$S_p = \{\bigwedge \Sigma \wedge \neg\varphi \mid \Sigma \cup \{\varphi\} \text{ is a prefix restricted subset of } P\}$$

are decidable. That is, we show that it is decidable to determine, given any $\psi \in S_p$, whether there is a (finite) structure such that $G \models \psi$.

Recall the following notion from [7].

Definition 3.2 [7]: A recursive class X of first-order logic sentences has the *small model property* for satisfiability iff there exists a recursive function s such that for each $\psi \in X$, if ψ is satisfiable, then ψ has a finite model of size at most $s(|\psi|)$, where $|\psi|$ stands for the length of ψ. ∎

To show the decidability of the satisfiability and finite satisfiability problems for S_p, it suffices to establish the small model property for S_p. To do this, we use a path label criterion to characterize whether a structure satisfies a sentence of S_p. More specifically, given a structure G and a sentence ψ of S_p, we label each node of G with paths in ψ. The path label of G, $LB(G,\psi)$, is the collection of the labels of all the nodes in G. This path label has the following properties:

- for any structure H, if $LB(H,\psi) = LB(G,\psi)$, then $H \models \psi$ iff $G \models \psi$; and
- there is a structure H of size at most $2^{2^{2|\psi|}}$, such that $LB(H,\psi) = LB(G,\psi)$.

In the remainder of this section, we present the path label criterion and show that it has the properties described above.

3.3 A path label criterion

We first define the path labels, and then discuss their properties.

Path labels

Given a structure G and a sentence ψ in S_p, we define a path label $LB(G, \psi)$ to characterize whether $G \models \psi$.

Let $G = (|G|, r^G, E^G)$ and $\psi = \bigwedge \Sigma \wedge \neg\varphi$. We use the following sets to denote the paths in ψ:

$$\begin{array}{rcl} Paths_\alpha(\psi) & = & \{pf(\phi) \mid \phi \in \Sigma \cup \{\varphi\}\} \\ Paths_\beta(\psi) & = & \{lt(\phi) \mid \phi \in \Sigma \cup \{\varphi\}\} \\ Paths_\gamma^+(\psi) & = & \{rt(\phi) \mid \phi \in \Sigma \cup \{\varphi\},\ \phi \text{ is a forward constraint}\} \\ Paths_\gamma^-(\psi) & = & \{-rt(\phi) \mid \phi \in \Sigma \cup \{\varphi\},\ \phi \text{ is a backward constraint}\} \\ Paths_{(\beta,\gamma)}(\psi) & = & Paths_\beta(\psi) \cup Paths_\gamma^+(\psi) \cup Paths_\gamma^-(\psi) \end{array}$$

Here the notation $-\rho$ denotes the pair $(-, \rho)$. We use this notation merely to distinguish the occurrence of a path as the right tail of a backward constraint as opposed to a forward constraint.

For each node a in $|G|$, we define a path label using paths in $Paths_\alpha(\psi)$ and $Paths_{(\beta,\gamma)}(\psi)$. This label consists of a pair of sets. The first component of the pair is the set of paths from r^G to a which are in $Paths_\alpha(\psi)$. That is,

$$lb_\alpha(a, G, \psi) = \{\rho \mid \rho \in Paths_\alpha(\psi),\ G \models \rho(r^G, a)\}.$$

The second is a collection of sets of paths in $Paths_{(\beta,\gamma)}(\psi)$. Each set consists of the paths between the node a and a node in $|G|$. More specifically, for each $b \in |G|$, let:

$$\begin{array}{rcl} lbs_\beta(a, b, G, \psi) & = & \{\rho \mid \rho \in Paths_\beta(\psi),\ G \models \rho(a, b)\} \\ lbs_\gamma(a, b, G, \psi) & = & \{\rho \mid \rho \in Paths_\gamma^+(\psi),\ G \models \rho(a, b)\} \\ & & \cup \{-\rho \mid -\rho \in Paths_\gamma^-(\psi),\ G \models \rho(b, a)\} \\ lbs_{(\beta,\gamma)}(a, b, G, \psi) & = & lbs_\beta(a, b, G, \psi) \ \cup \ lbs_\gamma(a, b, G, \psi) \end{array}$$

The second component of the label is defined by:

$$lb_{(\beta,\gamma)}(a, G, \psi) = \{lbs_{(\beta,\gamma)}(a, b, G, \psi) \mid b \in |G|\}$$

More precisely, we define the *label of node a in G w.r.t. ψ* by:

$$lb(a, G, \psi) = \begin{cases} (\emptyset,\ \emptyset) & \text{if } lb_\alpha(a, G, \psi) = \emptyset \\ (lb_\alpha(a, G, \psi),\ lb_{(\beta,\gamma)}(a, G, \psi)) & \text{otherwise} \end{cases}$$

The *label of G w.r.t. ψ* is defined by

$$LB(G, \psi) \ = \ \{lb(a, G, \psi) \mid a \in |G|\}.$$

Every label $l \in LB(G, \psi)$ is a pair of sets. We refer to the first component of l as $lb_\alpha(l)$, and the second as $lb_{(\beta,\gamma)}(l)$. In addition, we use the following notations:

$$\begin{array}{rcl} LB_\alpha(G, \psi) & = & \{lb_\alpha(l) \mid l \in LB(G, \psi)\} \\ LB_{(\beta,\gamma)}(G, \psi) & = & \{lb_{(\beta,\gamma)}(l) \mid l \in LB(G, \psi)\} \end{array}$$

We now consider a special case of $LB(G,\psi)$. If ψ involves simple constraints only, i.e., $\Sigma \cup \{\varphi\}$ is a subset of P_s, then $Paths_\alpha(\psi) = \{\epsilon\}$. Thus we have:

$$LB(G,\psi) = \begin{cases} \{(\epsilon,\, lb_{(\beta,\gamma)}(r^G, G, \psi))\} & \text{if } |G| \text{ is a singleton set} \\ \{(\epsilon,\, lb_{(\beta,\gamma)}(r^G, G, \psi)),\ (\emptyset,\, \emptyset)\} & \text{otherwise} \end{cases}$$

In this case, the cardinality of $LB(G,\psi)$ is at most 2.

Properties of the path labels

The most important property of $LB(G,\psi)$ is that it characterizes whether $G \models \psi$.

Let G be a structure and ψ a sentence, as described above. We say that $LB(G,\psi)$ *satisfies* ψ iff it satisfies the following conditions.

- For each $\phi \in \Sigma$, $LB(G,\psi)$ satisfies ϕ. That is, for any $l \in LB(G,\psi)$ and $s \in lb_{(\beta,\gamma)}(l)$, if $pf(\phi) \in lb_\alpha(l)$ and $lt(\phi) \in s$, then
  - $rt(\phi) \in s$ if ϕ is a forward constraint, and
  - $-rt(\phi) \in s$ if ϕ is a backward constraint.
- $LB(G,\psi)$ does not satisfy φ. That is, there exists $l \in LB(G,\psi)$ and $s \in lb_{(\beta,\gamma)}(l)$, such that $pf(\varphi) \in lb_\alpha(l)$, $lt(\varphi) \in s$, and
  - $rt(\varphi) \notin s$ if φ is a forward constraint, and
  - $-rt(\varphi) \notin s$ if φ is a backward constraint.

Lemma 3.2: For any structure G and any sentence $\psi \in S_p$, $G \models \psi$ iff $LB(G,\psi)$ satisfies ψ. ■

Proof: Let $\psi = \bigwedge \Sigma \wedge \neg\varphi$. It suffices to show that for each $\phi \in \Sigma \cup \{\varphi\}$, $G \models \phi$ iff $LB(G,\psi)$ satisfies ϕ. Without loss of generality, assume that all the constraints in $\Sigma \cup \{\varphi\}$ are forward constraints. The proof for the backward case is analogous.

(1) Assume $G \models \phi$, we want to show that $LB(G,\psi)$ satisfies ϕ.

Suppose, for *reductio*, that $LB(G,\psi)$ does not satisfy ϕ. That is, there exist $l \in LB(G,\psi)$ and $s \in lb_{(\beta,\gamma)}(l)$, such that $pf(\phi) \in lb_\alpha(l)$ and $lt(\phi) \in s$, but $rt(\phi) \notin s$. By the definition of the path labels, there exist $a, b \in |G|$, such that $lb(a, G, \psi) = l$ and $lbs_{(\beta,\gamma)}(a, b, G, \psi) = s$. Hence,

$$G \models pf(\phi)(r^G, a) \wedge lt(\phi)(a, b) \wedge \neg rt(\phi)(a, b).$$

This contradicts the assumption.

(2) Conversely, assume $G \not\models \phi$. we want to show that $LB(G,\psi)$ does not satisfy ϕ.

Suppose, for *reductio*, that $LB(G,\psi)$ satisfies ϕ. That is, for each $l \in LB(G,\psi)$ and each $s \in lb_{(\beta,\gamma)}(l)$, if $pf(\phi) \in lb_\alpha(l)$ and $lt(\phi) \in s$, then $rt(\phi) \in s$. However, since $G \models \neg\phi$, there exist $a, b \in |G|$, such that

$$G \models pf(\phi)(r^G, a) \wedge lt(\phi)(a, b) \wedge \neg rt(\phi)(a, b).$$

Hence, by the definition of the path labels, $pf(\phi) \in lb_\alpha(a, G, \psi)$, $lt(\phi) \in lbs_{(\beta,\gamma)}(a, b, G, \psi)$, but $rt(\phi) \notin lbs_{(\beta,\gamma)}(a, b, G, \psi)$. Let $l = lb(a, G, \psi)$ and $s = lbs_{(\beta,\gamma)}(a, b, G, \psi)$. Clearly, l and s contradict the assumption. ■

From Lemma 3.2 follows immediately the corollary below.

Corollary 3.3: For all structures G, H, and any sentence $\psi \in S_p$, if $LB(G,\psi) = LB(H,\psi)$, then $G \models \psi$ iff $H \models \psi$. ■

The size of a path label

We next examine the cardinality of $LB(G,\psi)$. We use $|S|$ to denote the cardinality of a set S. Given a sentence $\psi \in S_p$, where $\psi = \bigwedge \Sigma \wedge \neg\varphi$, it is easy to verify that

$$\begin{aligned} |Paths_\alpha(\psi)| &\leq |\psi|, \\ |Paths_{(\beta,\gamma)}(\psi)| &\leq |\psi|. \end{aligned}$$

For any structure G and any $l \in LB(G,\psi)$, $lb_\alpha(l)$ is a subset of $Paths_\alpha(\psi)$ and $lb_{(\beta,\gamma)}(l)$ is a subset of the power set of $Paths_{(\beta,\gamma)}(\psi)$. Therefore,

$$|LB(G,\psi)| \leq 2^{|\psi| + 2^{|\psi|}}.$$

In particular, if ψ involves simple constraints only, then $|LB(G,\psi)| \leq 2$.

We define the *prefix length of* ψ, $s_\alpha(\psi)$, to be $|pf(\varphi)|$. Note that the prefixes of all the constraints in $\Sigma \cup \{\varphi\}$ have the same length.

3.4 The small model property

Next, we establish the small model property for S_p. Using the path label criterion described above, it suffices to show the following.

Proposition 3.4: For each structure G and each sentence ψ in S_p, there is a structure H, such that

1. the size of H is at most $2^{2^{2|\psi|}}$; and
2. $LB(H,\psi) = LB(G,\psi)$.

■

The proof of the proposition requires two lemmas and the following notation.

Definition 3.3: Let G be a structure, m be a natural number and $a \in |G|$. The *m-neighborhood of a in G* is the structure $G(a) = (|G(a)|, r^{G(a)}, E^{G(a)})$, such that

- $|G(a)| = \{c \mid c \in |G|,$ there is path ρ, $|\rho| \leq m$ and either $G \models \rho(a,c)$ or $G \models \rho(c,a)\}$;
- $r^{G(a)} = a$; and
- for all $b, c \in |G(a)|$ and each $K \in E$, $G(a) \models K(b,c)$ iff $G \models K(b,c)$.

That is, $G(a)$ is the restriction of G to $|G(a)|$ with a as the new root. ■

Given a structure G and a sentence ψ in S_p, the first lemma below proves the existence of a structure G_α which has the following properties.

- $LB_\alpha(G_\alpha,\psi) = LB_\alpha(G,\psi)$. In addition, for each $l \in LB(G,\psi)$, there is a distinguished node $a_l \in |G_\alpha|$ such that $lb_\alpha(a_l, G_\alpha, \psi) = lb_\alpha(l)$.
- For each $a \in |G_\alpha|$, if $lb_\alpha(a, G_\alpha, \psi) \neq \emptyset$, then a does not have any outgoing edge. That is, for each $K \in E$ and $b \in |G_\alpha|$, $G_\alpha \models \neg K(a,b)$.

We shall proceed to construct the structure H described in Proposition 3.4 such that G_α is the $s_\alpha(\psi)$-neighborhood of r^H in H. This ensures that $LB_\alpha(H,\psi) = LB_\alpha(G,\psi)$.

Lemma 3.5: For any structure G and sentence $\psi \in S_p$, there is a structure $G_\alpha = (|G_\alpha|, r^{G_\alpha}, E^{G_\alpha})$, such that

1. the size of G_α is at most $|\psi| + 2^{|\psi| + 2^{|\psi|}}$;
2. there is a subset L_α of $|G_\alpha|$, such that
   (a) there is a bijection $f : LB(G, \psi) \to L_\alpha$, such that for each $l \in LB(G, \psi)$,
      i. $lb_\alpha(l) = lb_\alpha(f(l), G_\alpha, \psi) = \{\rho \mid \rho \text{ is a path, } G_\alpha \models \rho(r^{G_\alpha}, f(l))\}$,
      ii. for each $K \in E$ and $b \in |G_\alpha|$, $G_\alpha \models \neg K(f(l), b)$;
   (b) for each $b \in |G_\alpha| \setminus L_\alpha$,
      i. $lb_\alpha(b, G_\alpha, \psi) = \emptyset$,
      ii. there is a unique path ρ such that $G_\alpha \models \rho(r^{G_\alpha}, b)$. In addition, $|\rho| < s_\alpha(\psi)$.

■

Proof: Let $I_\alpha(\psi) = \bigcup_{\varrho \in Paths_\alpha(\psi)} \{\rho \mid \rho \prec_p \varrho\}$. Here $\rho \prec_p \varrho$ stands for that ρ is a proper prefix of ϱ, as defined in Section 2. We construct G_α using $LB(G, \psi)$ and $I_\alpha(\psi)$ as follows. For each $\rho \in I_\alpha(\psi)$, let a_ρ be a distinguished node, and for each $l \in LB(G, \psi)$, let a_l be a distinguished node. Let

- $L_\alpha = \{a_l \mid l \in LB(G, \psi)\}$;
- $|G_\alpha| = L_\alpha \cup \{a_\rho \mid \rho \in I_\alpha(\psi)\}$;
- $r^{G_\alpha} = \begin{cases} a_\epsilon & \text{if } s_\alpha(\psi) \geq 1 \\ a_{lb(r^G, G, \psi)} & \text{otherwise;} \end{cases}$
- for all $a, b \in |G_\alpha|$ and $K \in E$, $G_\alpha \models K(a, b)$ iff there exists $\rho \in I_\alpha(\psi)$, such that $a = a_\rho$ (i.e., $a \notin L_\alpha$), and one of the following conditions is satisfied:
  - there exists $\varrho \in I_\alpha(\psi)$, such that $b = a_\varrho$ (i.e., $b \notin L_\alpha$), and $\varrho = \rho \cdot K$; or
  - there exists $l \in LB(G, \psi)$, such that $b = a_l$ (i.e., $b \in L_\alpha$), and there exists $\varrho \in lb_\alpha(l)$, such that $\varrho = \rho \cdot K$.

The structure G_α is basically a rooted acyclic directed graph (see Figure 4). It has the following properties.

- The restriction of G_α to $\{a_\rho \mid \rho \in I_\alpha(\psi)\}$ is a tree of height $s_\alpha(\psi) - 1$. For each node a_ρ in the tree, there is a single path ρ from the root r^{G_α} to a_ρ.
- At level $s_\alpha(\psi)$, there are $|LB(G, \psi)|$ many nodes. Each of these nodes is uniquely marked with a label $l \in LB(G, \psi)$. In addition, it does not have any outgoing edges, and all its incoming edges are from leaves of the tree mentioned above.

We now verify that G_α indeed meets all the requirements of the lemma.

(1) *The size of* G_α.

Let $size(A)$ denote the size of a structure A. Since $|L_\alpha| = |LB(G, \psi)| \leq 2^{|\psi| + 2^{|\psi|}}$ and $|I_\alpha(\psi)| \leq |\psi|$, $size(G_\alpha)$ is at most

$$|\psi| + 2^{|\psi| + 2^{|\psi|}}.$$

In particular, when $s_\alpha(\psi) = 0$, $|LB(G, \psi)| \leq 2$ and $size(G_\alpha)$ is at most 2.

(2) *The properties of* L_α.

The bijection f from $LB(G, \psi)$ to L_α can be defined by:

$$l \mapsto a_l.$$

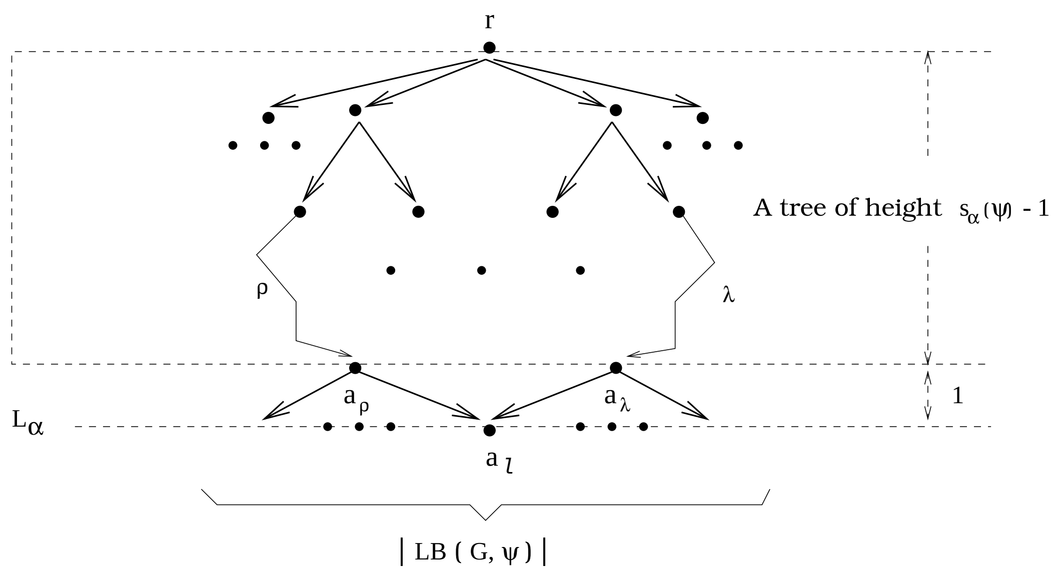

Figure 4: The structure G_α in Lemma 3.5

To verify the other properties of L_α, first observe the following simple fact.

Claim: For any $\varrho \in I_\alpha(\psi)$, $\{\rho \mid \rho$ is a path, $G_\alpha \models \rho(r^{G_\alpha}, a_\varrho)\} = \{\varrho\}$.

This claim can be verified by a straightforward induction on $|\varrho|$. From this claim and the construction of G_α, the second statement of the lemma follows. ■

The next lemma deals with $LB_{(\beta,\gamma)}(G,\psi)$. Given a label l in $LB(G,\psi)$, it constructs a structure $G_l = (|G_l|, r^{G_l}, E^{G_l})$ such that

$$lb_{(\beta,\gamma)}(r^{G_l}, G_l, \psi) = lb_{(\beta,\gamma)}(l).$$

We shall construct the structure H described in Proposition 3.4 in such a way that for each l in $LB(G,\psi)$, G_l is part of H, and moreover,

$$lb_{(\beta,\gamma)}(r^{G_l}, H, \psi) = lb_{(\beta,\gamma)}(r^{G_l}, G_l, \psi).$$

Lemma 3.6: Let G be a structure and ψ a sentence in S_p. For each label l in $LB(G,\psi)$, there is a structure G_l, such that

1. the size of G_l is at most $2^{|\psi|}$; and
2. $lb_{(\beta,\gamma)}(r^{G_l}, G_l, \psi) = lb_{(\beta,\gamma)}(l)$.

■

Proof: We give a filtration argument. To do this, we need the following notations.

First, we define the following sets:

$$\begin{aligned}
I^+(\psi) &= \bigcup_{\varrho \in Paths_\beta(\psi) \cup Paths^+_\gamma(\psi)} \{\rho \mid \rho \preceq_p \varrho\} \\
I^-(\psi) &= \bigcup_{-\varrho \in Paths^-_\gamma(\psi)} \{-\rho \mid \rho \preceq_s \varrho\} \\
I(\psi) &= I^+(\psi) \cup I^-(\psi)
\end{aligned}$$

Here $\rho \preceq_p \varrho$ denotes that ρ is a prefix of ϱ, and $\rho \preceq_s \varrho$ denotes that ρ is a suffix of ϱ, as described in Section 2. It is easy to verify that $|I(\psi)| \leq |\psi|$.

Second, by $l \in LB(G, \psi)$, there exists $a \in |G|$ such that

$$lb(a, G, \psi) = l.$$

Using a, we define a mapping g from $|G|$ to the power set of $I(\psi)$, such that for each $b \in |G|$,

$$g(b) \mapsto \{\rho \mid \rho \in I^+(\psi),\, G \models \rho(a,\, b)\} \,\cup\, \{-\rho \mid\, -\rho \in I^-(\psi),\, G \models \rho(b,\, a)\}.$$

Using the mapping g, we define an equivalence relation $\sim$ on $|G|$ such that

$$b \sim b' \quad \text{iff} \quad g(b) = g(b').$$

Let $[b]$ denote the equivalence class of b with respect to $\sim$. We proceed to construct a σ-structure $G_l = (|G_l|,\, r^{G_l},\, E^{G_l})$ whose nodes are these equivalence classes: Let

- $|G_l| = \{[b] \mid b \in |G|\}$;
- $r^{G_l} = [a]$;
- for all $o_1, o_2 \in |G_l|$ and $K \in E$, $G_l \models K(o_1, o_2)$ iff there exist $b_1, b_2 \in |G|$, such that $[b_1] = o_1$, $[b_2] = o_2$, and $G \models K(b_1,\, b_2)$.

We next show that G_l is indeed the structure desired.
(1) *The size of* G_l.
For each $b \in |G|$, $g(b) \subseteq I(\psi)$. Since $|I(\psi)| \leq |\psi|$, the size of G_l is at most $2^{|\psi|}$.

(2) $lb_{(\beta,\gamma)}(r^{G_l}, G_l, \psi) = lb_{(\beta,\gamma)}(l)$.
It suffices to show the following claim.

Claim 1: For each $b \in |G|$, $lbs_{(\beta,\gamma)}(r^{G_l}, [b], G_l, \psi) = lbs_{(\beta,\gamma)}(a, b, G, \psi)$.

For if Claim 1 holds, then

$$\begin{aligned} lb_{(\beta,\gamma)}(r^{G_l}, G_l, \psi) &= \{lbs_{(\beta,\gamma)}(r^{G_l}, c, G_l, \psi) \mid c \in |G_l|\} \\ &= \{lbs_{(\beta,\gamma)}(r^{G_l}, [b], G_l, \psi) \mid b \in |G|\} \\ &= \{lbs_{(\beta,\gamma)}(a, b, G, \psi) \mid b \in |G|\} \\ &= lb_{(\beta,\gamma)}(a, G, \psi). \\ &= lb_{(\beta,\gamma)}(l). \end{aligned}$$

To verify Claim 1, it suffices to show the following.

Claim 2: For any $b \in |G|$ and $\rho \in I^+(\psi)$, $G \models \rho(a,\, b)$ iff $G_l \models \rho(r^{G_l},\, [b])$.

Claim 3: For any $b \in |G|$ and $-\rho \in I^-(\psi)$, $G \models \rho(b,\, a)$ iff $G_l \models \rho([b],\, r^{G_l})$.

For if these claims hold, then from $Paths_{(\beta,\gamma)}(\psi) \subseteq I(\psi)$ and the definition of $lbs_{(\beta,\gamma)}$ follows Claim 1.

We next show Claim 2 by induction on $|\rho|$. Similarly, Claim 3 can be verified.
Base case: $|\rho| = 0$. That is, $\rho = \epsilon$. By the definition of g, it is straightforward to verify that $G_l \models \epsilon(r^{G_l},\, [b])$ iff $g(b) = g(a)$ iff $\epsilon \in g(b)$ iff $b = a$ iff $G \models \epsilon(a,\, b)$. Therefore, Claim 2 holds in this case.

Inductive step: Assume the claim for $|\rho| = m$.

We next show that the claim holds for $|\rho| = m + 1$. That is, ρ is of the form $\varrho \cdot K$, where $\varrho \in I^+(\psi)$, $|\varrho| = m$ and $K \in E$.

First, suppose that $G \models \rho(a, b)$. Then there exists $c \in |G|$, such that

$$G \models \varrho(a, c) \wedge K(c, b).$$

By the induction hypothesis,

$$G_l \models \varrho(r^{G_l}, [c]).$$

Moreover, by $G \models K(c, b)$ and the definition of G_l, we have

$$G_l \models K([c], [b]).$$

Therefore, $G_l \models \rho(r^{G_l}, [b])$.

Conversely, assume that $G_l \models \rho(r^{G_l}, [b])$. Then there exists $o \in |G_l|$, such that

$$G_l \models \varrho(r^{G_l}, o) \wedge K(o, [b]).$$

By the definition of G_l and $G_l \models K(o, [b])$, there exist $o_1, b_1 \in |G|$, such that $[o_1] = o$, $[b_1] = [b]$, and

$$G \models K(o_1, b_1).$$

In addition, since $G_l \models \varrho(r^{G_l}, o)$ and $[o_1] = o$, by the induction hypothesis, we have that

$$G \models \varrho(a, o_1).$$

Therefore, $G \models \rho(a, b_1)$. That is, $\rho \in g(b_1)$. By $[b_1] = [b]$, we have that $g(b_1) = g(b)$. Therefore, $\rho \in g(b)$. Hence $G \models \rho(a, b)$.

This completes the proof of Lemma 3.6. ■

Finally, we prove Proposition 3.4. As mentioned earlier, given a structure G and a sentence ψ in S_p, we define the structure H described in Proposition 3.4 in such a way that

- the structure G_α in Lemma 3.5 is the $s_\alpha(\psi)$-neighborhood of r^H in H;
- for each $l \in LB(G, \psi)$, G_l in Lemma 3.6 is part of H such that
  - $r^{G_l} = f(l)$, where f is the function specified in Lemma 3.5,
  - $lb_{(\beta,\gamma)}(r^{G_l}, H, \psi) = lb_{(\beta,\gamma)}(r^{G_l}, G_l, \psi) = lb_{(\beta,\gamma)}(l)$, and
  - $lb_\alpha(r^{G_l}, H, \psi) = lb_\alpha(l)$.

Note that the proof below uses the restriction on prefixes described in Definition 3.1.

Proof of Proposition 3.4: Given a structure G and a sentence ψ in S_p, let G_α be the structure specified in Lemma 3.5, and for each $l \in LB(G, \psi)$, let G_l be the structure specified in Lemma 3.6. Without loss of generality, assume that $|G_l| \cap |G_\alpha| = \emptyset$ and $|G_l| \cap |G_{l'}| = \emptyset$ if $l \neq l'$. We build structure $H = (|H|, r^H, E^H)$, as follows.

- $|H| = |G_\alpha| \cup \bigcup_{l \in LB(G,\psi)} (|G_l| \ \setminus \ \{r^{G_l}\});$
- $r^H = r^{G_\alpha};$

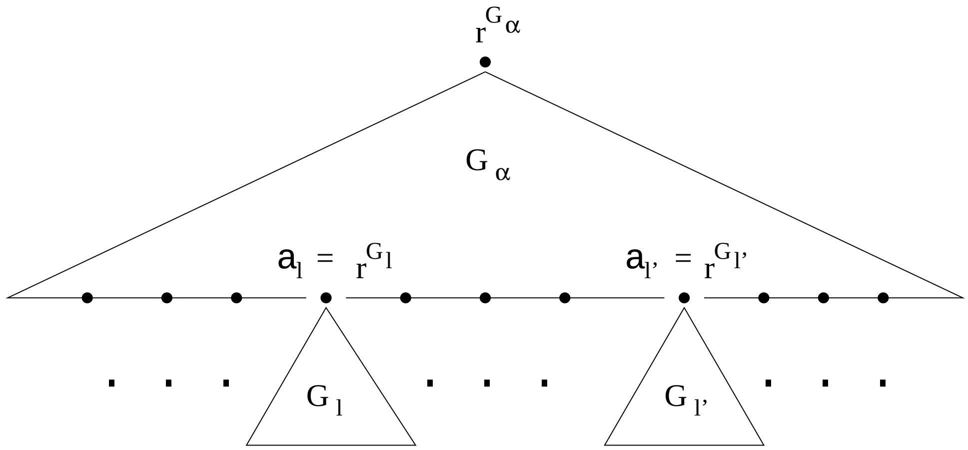

Figure 5: The structure H in Proposition 3.4

- For all $a, b \in |H|$ and each $K \in E$, $H \models K(a,b)$ iff one of the following conditions is satisfied:
  - $a, b \in |G_\alpha|$ and $G_\alpha \models K(a,b)$;
  - For some $l \in LB(G,\psi)$, $a, b \in |G_l|$ and $G_l \models K(a,b)$;
  - Let L_α be the subset of $|G_\alpha|$ and f the function specified in Lemma 3.5. For some $l \in LB(G,\psi)$,
    - $a = f(l)$, $b \in |G_l|$ and $G_l \models K(r^{G_l}, b)$; or
    - $b = f(l)$, $a \in |G_l|$ and $G_l \models K(a, r^{G_l})$; or
    - $a = b = f(l)$ and $G_l \models K(r^{G_l}, r^{G_l})$.

Intuitively, H is built from G_α and G_l's by identifying $f(l)$ with r^{G_l} for each $l \in LB(G,\psi)$. See Figure 5 for the structure H.

We now show that H is indeed the structure desired.

(1) *The size of H.*

Obviously,

$$size(H) = size(G_\alpha) + \sum_{l \in LB(G,\psi)} size(G_l) - |LB(G,\psi)|.$$

By Lemmas 3.5 and 3.6, the size of $|H|$ is at most

$$|\psi| + 2^{|\psi| + 2^{|\psi|}}\, 2^{|\psi|},$$

which is no larger than $2^{2^{2|\psi|}}$.

Note that when $s_\alpha(\psi) = 0$, the size of H is at most $2^{|\psi|}$.

(2) $LB(H,\psi) = LB(G,\psi)$.

It suffices to show the claim below.

Claim: Let L_α be the set and f the function specified in Lemma 3.5. They have the following properties.

1. For each $a \in |H| \setminus L_\alpha$, $lb(a, H, \psi) = (\emptyset,\ \emptyset)$.

2. For each $l \in LB(G,\psi)$, $lb(f(l), H, \psi) = l$.

For if the claim holds, then $LB(G,\psi) \subseteq LB(H,\psi)$. In addition, by Lemma 3.5, f is a bijection between $LB(G,\psi)$ and L_α. Therefore, $LB(H,\psi) = LB(G,\psi)$.

To show the claim, first observe the following simple facts about H, which are immediate from the definition of H.

Fact 1: For any $l \in LB(G,\psi)$ and $a, b \in |H|$, if $b \in |G_l|$ and $a \notin |G_l|$, then for each path ρ,

- $H \models \rho(a,b)$ iff there are paths λ and ϱ, such that $\rho = \lambda \cdot \varrho$ and $H \models \lambda(a, f(l)) \wedge \varrho(f(l), b)$;
- $H \models \rho(b,a)$ iff $a = f(l)$ and $G_l \models \rho(b, r^{G_l})$.

Fact 2: For each $l \in LB(G,\psi)$ and any $a \in |H|$, if there exists path ρ such that $H \models \rho(f(l), a)$, then either $a \in |G_l|$ or $a = f(l)$.

Fact 3: For each $l \in LB(G,\psi)$, for any path ρ and node $a \in |G_l| \cap |H|$,

$$
\begin{aligned}
H \models \rho(f(l), a) \quad &\text{iff} \quad G_l \models \rho(r^{G_l}, a), \\
H \models \rho(a, f(l)) \quad &\text{iff} \quad G_l \models \rho(a, r^{G_l}), \\
H \models \rho(f(l), f(l)) \quad &\text{iff} \quad G_l \models \rho(r^{G_l}, r^{G_l}).
\end{aligned}
$$

Using the facts above, we examine the following cases.

Case 1: $a \in |G_\alpha| \setminus L_\alpha$. By Facts 1 and 2, all the paths from r^H to node a are in G_α. By Lemma 3.5, there is only one path from r^H to a, and the length of the path is less than $S_\alpha(\psi)$. By the definition of the path labels and the restriction on prefixes described in Definition 3.1, $lb(a, H, \psi) = (\emptyset,\ \emptyset)$.

Case 2: *For some* $l \in LB(G,\psi)$, $a \in |G_l| \setminus \{r^{G_l}\}$. By Fact 1, if there is path ρ from r^H to node a, then there must be paths λ and ϱ, such that $\rho = \lambda \cdot \varrho$ and $H \models \lambda(a, f(l)) \wedge \varrho(f(l), b)$. In addition, since $a \neq f(l)$, we have $\varrho \neq \epsilon$. By Lemma 3.5, $|\lambda| = s_\alpha(\psi)$. Hence $s_\alpha(\psi) < |\rho|$. Therefore, by the definition of the path labels and the restriction on prefixes described in Definition 3.1, $lb(a, H, \psi) = (\emptyset,\ \emptyset)$.

Case 3: $a \in L_\alpha$. That is, $a = f(l)$ for some $l \in LB(G,\psi)$. By Lemma 3.5 and Fact 1, $lb_\alpha(a, H, \psi) = lb_\alpha(l)$. By Facts 2, 3 and Lemma 3.6, we have $lb_{(\beta,\gamma)}(a, H, \psi) = lb_{(\beta,\gamma)}(l)$. Hence $lb(a, H, \psi) = l$.

Therefore, the claim holds.

This completes the proof of Proposition 3.4 ■

4 Sublanguage P_β

In this section, we present a sublanguage of P and establish the decidability of its associated implication problems.

4.1 Definition

Some cases of path constraint implication are not examples of the prefix restricted implication. For instance, the set consisting of the two extent constraints and the two inverse constraints for school databases given in Section 1 is not a prefix restricted subset of P.

The constraints in the last example, however, are in the sublanguage P_β defined below.

Definition 4.1: A *β-restricted path constraint* φ is a constraint in P with $|lt(\varphi)| \leq 1$. That is, either $lt(\varphi) = \epsilon$, or $lt(\varphi) = K$ for some $K \in E$.

The set of all simple path constraints and all β-restricted path constraints is denoted by P_β. ■

Note that the class of word constraints is a proper subset of P_β. In addition, not all constraints in P_β are expressible in two-variable first-order logic. Indeed, the constraint φ given in Example 2.1 is in P_β, but is not in FO^2.

4.2 The implication problems for P_β

The decidability of the implication problems for P_β is established by the following.

Theorem 4.1: The implication and finite implication problems for P_β are decidable. ■

In the same way as in the proof of Theorem 3.1, we show Theorem 4.1 by establishing the small model property for the following set of sentences:

$$S(P_\beta) = \{\bigwedge \Sigma \wedge \neg\varphi \mid \varphi \in P_\beta,\ \Sigma \subset P_\beta,\ \Sigma \text{ is finite}\}.$$

To do this, we give a filtration argument. Given a satisfiable sentence ψ in $S(P_\beta)$, we find the set of paths in ψ and use a path labeling mechanism similar to the one employed in the proof of Theorem 3.1. More specifically, let G be a model of ψ. We use the paths in ψ to label each node of G, and therefore, obtain the label of G with respect to ψ. The cardinality of this label is determined only by $|\psi|$, the length of ψ. We then construct a structure H, such that H and G have the same label with respect to ψ, and moreover, $H \models \psi$. In addition, each node of H has a unique path label. The size of H is, therefore, bounded by the cardinality of the label of G with respect to ψ, which is at most $2^{|\psi|}$. Thus the small model property is established.

We first define the path labels, called *relative path labels*. Using the path labels, we then establish the small model property for $S(P_\beta)$.

4.3 Relative path label

Given a satisfiable sentence ψ of $S(P_\beta)$, where $\psi = \bigwedge \Sigma \wedge \neg\varphi$, we use the following sets to denote paths in ψ:

$$\begin{aligned}
Paths_{(\alpha,\beta)}(\psi) &= \{pf(\phi) \mid \phi \in \Sigma \cup \{\varphi\}\} \ \cup\ \{lt(\phi) \mid \phi \in \Sigma \cup \{\varphi\},\ \phi \in P_s\} \\
I_{(\alpha,\beta)}(\psi) &= \bigcup_{\varrho \in Paths_{(\alpha,\beta)}(\psi)} \{\rho \mid \rho \preceq_p \varrho\} \\
I(\varphi) &= \begin{cases} \{\rho \mid \rho \preceq_p rt(\varphi)\} & \text{if } \varphi \text{ is a forward constraint} \\ \{\rho \mid \rho \preceq_s rt(\varphi)\} & \text{if } \varphi \text{ is a backward constraint} \end{cases}
\end{aligned}$$

Here $\rho \preceq_p \varrho$ ($\rho \preceq_s \varrho$) means that ρ is a prefix (suffix) of ϱ, as defined in Section 2.

Let G be a model of ψ, $G = (|G|, r^G, E^G)$, and (a, b) be a pair of nodes in $|G|$ such that

$$G \models pf(\varphi)(r, a) \wedge lt(\varphi)(a, b) \wedge \neg rt(\varphi)(a, b)$$

if φ is a forward constraint, and

$$G \models pf(\varphi)(r, a) \wedge lt(\varphi)(a, b) \wedge \neg rt(\varphi)(b, a)$$

if φ is a backward constraint. This pair is referred to as a *witness of* $\neg\varphi$ *in* G.

For each $c \in |G|$, we label c with a pair. The first component of the pair is

$$ls_{(\alpha,\beta)}(c, G, \psi) = \{\rho \mid \rho \in I_{(\alpha,\beta)}(\psi),\ G \models \rho(r^G, c)\}.$$

The second component is defined to be:

$$ls_\varphi(c,a,G,\psi) \;=\; \begin{cases} \{\rho \mid \rho \in I(\varphi),\ G \models \rho(a,c)\} & \text{if } \varphi \text{ is a forward constraint} \\ \{\rho \mid \rho \in I(\varphi),\ G \models \rho(c,a)\} & \text{if } \varphi \text{ is a backward constraint} \end{cases}$$

The *path label of node c in G relative to ψ and a* is defined to be:

$$ls(c,G,\psi,a) = (ls_{(\alpha,\beta)}(c,G,\psi),\ ls_\varphi(c,a,G,\psi))$$

The *path label of G relative to ψ and a* is defined to be:

$$LS(G,\psi,a) = \{ls(c,G,\psi,a) \mid c \in |G|\}$$

Note that for each $c \in |G|$,

- $\epsilon \in ls_{(\alpha,\beta)}(c,G,\psi)$ iff $c = r^G$, and
- $\epsilon \in ls_\varphi(c,a,G,\psi)$ iff $c = a$.

We next examine the size of a relative path label.

Given a satisfiable sentence ψ of $S(P_\beta)$, where $\psi = \bigwedge \Sigma \wedge \neg\varphi$, let G be a model of ψ and (a,b) a witness of $\neg\varphi$ of G. Note that for each $c \in |G|$,

$$\begin{aligned} ls_{(\alpha,\beta)}(c,G,\psi) &\subseteq I_{(\alpha,\beta)}(\psi), \\ ls_\varphi(c,a,G,\psi) &\subseteq I(\varphi). \end{aligned}$$

In addition, it is easy to verify that

$$|I_{(\alpha,\beta)}(\psi)| + |I(\varphi)| \leq |\psi|.$$

Hence

$$|LS(G,\psi,a)| \leq 2^{|\psi|}.$$

The notion of relative path labels differs from the one described in Section 3.3 in the following aspects. First, relative path labels are defined for models of satisfiable sentences in $S(P_\beta)$, rather than for arbitrary structures. Second, the relative path label of a node in a structure involves only the paths between the node and two fixed nodes of the structure, whereas the one given in Section 3.3 contains paths related to all the nodes in the structure. As a result, a relative path label has a much smaller cardinality. Third, a relative path label does not characterize whether a structure is a model of a sentence in $S(P_\beta)$, but based on it we are able to form a filtration argument to establish the small model property for $S(P_\beta)$.

4.4 The small model property

Based on relative path labels we establish the following proposition, from which follows Theorem 4.1.

Proposition 4.2: Every satisfiable sentence ψ of $S(P_\beta)$ has a model of size at most $2^{|\psi|}$. ■

Proof: Let ψ be a satisfiable sentence in $S(P_\beta)$, where $\psi = \bigwedge \Sigma \wedge \neg\varphi$. Since ψ is satisfiable, there is a structure $G = (|G|, r^G, E^G)$ such that $G \models \psi$. It follows that there exist $a, b \in |G|$ such that (a,b) is a witness of $\neg\varphi$ in G. That is,

$$G \models pf(\varphi)(r,a) \wedge lt(\varphi)(a,b) \wedge \neg rt(\varphi)(a,b)$$

if φ is a forward constraint, and

$$G \models pf(\varphi)(r, a) \wedge lt(\varphi)(a, b) \wedge \neg rt(\varphi)(b, a)$$

if φ is a backward constraint.

Consider $LS(G, \psi, a)$. As in the proof of Lemma 3.6, we define an equivalence relation $\sim$ on $|G|$ by:

$$b \sim b' \quad \text{iff} \quad ls(b, G, \psi, a) = ls(b', G, \psi, a).$$

We denote the equivalence class of b with respect to $\sim$ as $[b]$. By taking these equivalence classes as nodes, we proceed to construct a σ-structure $H = (|H|, r^H, E^H)$ as follows:

- $|H| = \{[b] \mid b \in |G|\}$;
- $r^H = [r^G]$;
- for each $K \in E$ and $o_1, o_2 \in |H|$, $H \models K(o_1, o_2)$ iff there are $b_1, b_2 \in |G|$ such that $[b_1] = o_1$, $[b_2] = o_2$, and $G \models K(b_1, b_2)$.

We next show that $H \models \psi$, and moreover, the size of H is at most $2^{|\psi|}$.

(1) *The size of H.*

Obviously, $size(H)$ is at most $|LS(G, \psi, a)|$. Therefore, the size of H is at most $2^{|\psi|}$.

(2) $H \models \psi$.

It suffices to show following claims.

Claim 1: For any path ρ and all $c, d \in |G|$, if $G \models \rho(c, d)$, then $H \models \rho([c], [d])$.

Claim 2: For each $c \in |G|$, $ls(c, G, \psi, a) = ls([c], H, \psi, [a])$.

Using these claims, we show $H \models \psi$ as follows. The proofs of these claims will be given shortly.

We first show that $H \models \Sigma$. Suppose, for *reductio*, that there exists $\phi \in \Sigma$ such that $H \models \neg\phi$. Without loss of generality, assume that ϕ is a forward constraint (the argument for the backward case is analogous). Then there exist $c, d \in |H|$, such that

$$H \models pf(\phi)(r^H, c) \wedge lt(\phi)(c, d) \wedge \neg rt(\phi)(c, d).$$

We have two cases to consider.

Case 1: ϕ is a simple constraint. That is, $pf(\phi) = \epsilon$ and $c = r^H$.

In this case, the assumption is equivalent to

$$lt(\phi) \in ls_{(\alpha,\beta)}(d, H, \psi) \quad \text{and} \quad H \models \neg rt(\phi)(r^H, d).$$

By the definition of H, there exists $d_1 \in |G|$, such that $[d_1] = d$. By Claim 2,

$$ls_{(\alpha,\beta)}(d_1, G, \psi) = ls_{(\alpha,\beta)}(d, H, \psi).$$

Hence $lt(\phi) \in ls_{(\alpha,\beta)}(d_1, G, \psi)$. That is, $G \models lt(\phi)(r^G, d_1)$. Since $G \models \phi$, $G \models rt(\phi)(r^G, d_1)$. By Claim 1, we have $H \models rt(\phi)(r^H, d)$. This contradicts the assumption.

Case 2: ϕ is a β-restricted constraint, i.e., $|lt(\phi)| \leq 1$.

If $|lt(\phi)| = 0$, then $c = d$. Thus by the assumption,

$$pf(\phi) \in ls_{(\alpha,\beta)}(c, H, \psi) \quad \text{and} \quad H \models \neg rt(\phi)(c, c).$$

By the definition of H, there exists $c_1 \in |G|$, such that $[c_1] = c$. By Claim 2,

$$ls_{(\alpha,\beta)}(c_1, G, \psi) = ls_{(\alpha,\beta)}(c, H, \psi).$$

Thus $pf(\phi) \in ls_{(\alpha,\beta)}(c_1, G, \psi)$. That is, $G \models pf(\phi)(r^G, c_1)$. By $G \models \phi$, $G \models rt(\phi)(c_1, c_1)$. Thus by Claim 1, we have $H \models rt(\phi)(c, c)$. This contradicts the assumption.

If $|lt(\phi)| = 1$, then $lt(\phi) = K$ for some $K \in E$. By the assumption, we have

$$pf(\phi) \in ls_{(\alpha,\beta)}(c, H, \psi) \quad \text{and} \quad H \models K(c, d) \wedge \neg rt(\phi)(c, d).$$

By the definition of H, there exist nodes $c_1, d_1 \in |G|$, such that $[c_1] = c$, $[d_1] = d$ and $G \models K(c_1, d_1)$. By Claim 2, we have that

$$ls_{(\alpha,\beta)}(c_1, G, \psi) = ls_{(\alpha,\beta)}(c, H, \psi).$$

As a result, we have $G \models pf(\phi)(r^G, c_1)$. Thus $G \models pf(\phi)(r^G, c_1) \wedge K(c_1, d_1)$. By $G \models \phi$, $G \models rt(\phi)(c_1, d_1)$. Thus by Claim 1, we have that $H \models rt(\phi)(c, d)$. Again, this contradicts the assumption.

Therefore, $H \models \Sigma$.

We next show that $H \models \neg\varphi$. Since (a, b) is a witness of $\neg\varphi$ in G,

$$G \models pf(\varphi)(r^G, a) \wedge lt(\varphi)(a, b).$$

By Claim 1,

$$H \models pf(\varphi)(r^H, [a]) \wedge lt(\varphi)([a], [b]).$$

By Claim 2, we have that $ls_\varphi(b, a, G, \psi) = ls_\varphi([b], [a], H, \psi)$. As a result, when φ is a forward constraint, by $G \models \neg rt(\varphi)(a, b)$, we have that

$$H \models \neg rt(\varphi)([a], [b]);$$

and when φ is a backward constraint, by $G \models \neg rt(\varphi)(b, a)$, we have that

$$H \models \neg rt(\varphi)([b], [a]).$$

Therefore, $H \models \neg\varphi$.

We now show Claim 1 by induction on $|\rho|$.

Base case: If $|\rho| = 0$, then $c = d$. Hence clearly $[c] = [d]$. That is, $H \models \epsilon([c], [d])$.

Inductive step: Assume the claim for $|\rho| = m$.

We now consider ρ with $|\rho| = m + 1$. By $|\rho| = m + 1$, there exist a path ϱ and $K \in E$, such that $\rho = \varrho \cdot K$ and $|\varrho| = m$. By $G \models \rho(c, d)$, there exist a node $c' \in |G|$, such that

$$G \models \varrho(c, c') \wedge K(c', d).$$

By the induction hypothesis, $H \models \varrho([c], [c'])$. Furthermore, by the definition of H, we have $H \models K([c'], [d])$. Hence $H \models \rho([c], [d])$.

Finally, we show Claim 2 by *reductio*.

Suppose that there exists $c \in |G|$, such that

$$ls(c, G, \psi, a) \neq ls([c], H, \psi, [a]).$$

Then we examine the following three cases.

Case 1: $ls_{(\alpha,\beta)}(c, G, \psi) \neq ls_{(\alpha,\beta)}([c], H, \psi)$.

To see this assumption leads to a contradiction, it suffices to show the claim below.

Claim 3: For any $\rho \in I_{(\alpha,\beta)}(\psi)$ and $c \in |G|$, $\rho \in ls_{(\alpha,\beta)}(c, G, \psi)$ iff $\rho \in ls_{(\alpha,\beta)}([c], H, \psi)$.

We show this claim by induction on $|\rho|$.

Base case: $|\rho| = 0$. That is, $\rho = \epsilon$. It is easy to see that

$$\begin{aligned} \epsilon \in ls_{(\alpha,\beta)}(c, G, \psi) \quad &\text{iff} \quad c = r^G, \\ \epsilon \in ls_{(\alpha,\beta)}([c], H, \psi) \quad &\text{iff} \quad [c] = [r^G]. \end{aligned}$$

Thus by the definition of f, we have that $\epsilon \in ls_{(\alpha,\beta)}(c, G, \psi)$ iff $\epsilon \in ls_{(\alpha,\beta)}([c], H, \psi)$.

Inductive step: Assume the claim for $|\rho| = m$. We next consider the claim for $|\rho| = m + 1$.

Suppose $\rho \in ls_{(\alpha,\beta)}(c, G, \psi)$. That is, $G \models \rho(r^G, c)$. Then by Claim 1, $H \models \rho(r^H, [c])$. That is, $\rho \in ls_{(\alpha,\beta)}([c], H, \psi)$.

Conversely, assume that $\rho \in ls_{(\alpha,\beta)}([c], H, \psi)$. Then there exist $d \in |H|$, $K \in E$ and path $\varrho \in I_{(\alpha,\beta)}(\psi)$, such that $\rho = \varrho \cdot K$, $|\varrho| = m$ and

$$H \models \varrho(r^H, d) \wedge K(d, [c]).$$

Because $H \models K(d, [c])$, by the definition of H, there exist nodes $c_1, d_1 \in |G|$ such that $[c_1] = [c]$, $[d_1] = d$ and $G \models K(d_1, c_1)$. Moreover, by the induction hypothesis, we have $\varrho \in ls_{(\alpha,\beta)}(d_1, G, \psi)$. That is, $G \models \varrho(r^G, d_1)$. Hence

$$\rho \in ls_{(\alpha,\beta)}(c_1, G, \psi).$$

By $[c_1] = [c]$, we have that $ls_{(\alpha,\beta)}(c, G, \psi) = ls_{(\alpha,\beta)}(c_1, G, \psi)$. Thus $\rho \in ls_{(\alpha,\beta)}(c, G, \psi)$.

Therefore, Claim 3 holds. As a result, the assumption in Case 1 leads to a contradiction.

Case 2: $ls_\varphi(c, a, G, \psi) \neq ls_\varphi([c], [a], G, \psi)$.

As for Case 1, it suffices to prove the following claim.

Claim 4: For each $\rho \in I(\varphi)$ and each $c \in |G|$, $\rho \in ls_\varphi(c, a, G, \psi)$ iff $\rho \in ls_\varphi([c], [a], H, \psi)$.

The proof of Claims 4 is similar to that of Claim 1, by induction on $|\rho|$. Here we assume that φ is a backward constraint. The proof for the forward case is analogous.

Base case: $|\rho| = 0$. That is, $\rho = \epsilon$. It is easy to see that

$$\begin{aligned} \epsilon \in ls_\varphi(c, a, G, \psi) \quad &\text{iff} \quad c = a, \\ \epsilon \in ls_\varphi([c], [a], H, \psi) \quad &\text{iff} \quad [c] = [a]. \end{aligned}$$

By the definition of f, we have that $\epsilon \in ls_\varphi(c, a, G, \psi)$ iff $\epsilon \in ls_\varphi([c], [a], H, \psi)$.

Inductive step: Assume the claim for $|\rho| = m$. We next consider the claim for $|\rho| = m + 1$.

Assume $\rho \in ls_\varphi(c, a, G, \psi)$. That is, $G \models \rho(c, a)$. Then by Claim 1, $H \models \rho([c], [a])$. That is, $\rho \in ls_\varphi([c], [a], H, \psi)$.

Conversely, assume that $\rho \in ls_\varphi([c], [a], H, \psi)$. Then there exist $d \in |H|$, $\varrho \in I(\varphi)$ and $K \in E$, such that $\rho = K \cdot \varrho$, $|\varrho| = m$ and

$$H \models K([c], d) \wedge \varrho(d, [a]).$$

Since $H \models K([c], d)$, by the definition of H, there are $c_1, d_1 \in |G|$ such that $[c_1] = [c]$, $[d_1] = d$ and $G \models K(c_1, d_1)$. By the induction hypothesis, we have that $\varrho \in ls_\varphi(d_1, G, \psi)$. That is, $G \models \varrho(d_1, a)$. Hence

$$\rho \in ls_\varphi(c_1, a, G, \psi).$$

By $[c_1] = [c]$, we have that $ls_\varphi(c, a, G, \psi) = ls_\varphi(c_1, a, G, \psi)$. Thus $\rho \in ls_\varphi(c, a, G, \psi)$.

Therefore, Claim 4 holds. Hence the assumption in Case 2 also leads to a contradiction.

This completes the proof of Proposition 4.2. ∎

5 Extended implications for P_β

In this section, we present a generalization of the implication problems for P_β and establish its decidability.

5.1 Definition

Consider the set consisting of the local extent constraints given in Section 1 and the local inverse constraints given in Section 3.1. This set is neither a prefix restricted subset of P nor a subset of P_β. However, the constraints in this set share the following property: all of them are constraints in schools databases in Figure 1 augmented with a common prefix `Schools`. In general, when represented in a global environment, path constraints in a local database are augmented with a common prefix.

This example motivates the following extension of P_β.

Definition 5.1: Let α be a path and φ be a constraint in P_β. The *extension of φ with prefix α*, denoted $\delta(\varphi, \alpha)$, is the constraint defined either by

$$\forall\, x\, y\, (\alpha \cdot pf(\varphi)(r, x) \wedge lt(\varphi)(x, y) \to rt(\varphi)(x, y))$$

when φ is of the forward form, or by

$$\forall\, x\, y\, (\alpha \cdot pf(\varphi)(r, x) \wedge lt(\varphi)(x, y) \to rt(\varphi)(y, x))$$

when φ is of the backward form, where $\cdot$ is the path concatenation operator, and pf, lt and rt are defined in Definition 2.2.

Let α be a path and Σ be a finite subset of P_β. The *extension of Σ with prefix α* is the subset of P defined by

$$\{\delta(\varphi, \alpha) \mid \varphi \in \Sigma\}.$$

Such a set is called a *prefix extended subset of P_β*.

The *extended (finite) implication problem for P_β* is the problem of determining, given any prefix extended subset $\Sigma \cup \{\varphi\}$ of P_β, whether all the (finite) models of Σ are also models of φ. ■

For instance, the set described in the last example is a prefix extended subset of P_β.

Note that the (finite) implication problem for P_β is a special case of the extended (finite) implication problem for P_β. As an immediate result, the implications for word constraints are special cases of the extended implications of P_β. Moreover, extended implications of P_β cannot be stated in two-variable first-order logic.

5.2 Decidability

In this section, we prove the decidability of the extended implication problems for P_β.

Theorem 5.1: The extended implication and finite implication problems for P_β are decidable. ■

We prove the theorem by reduction to the implication problems for P_β, whose decidability is established by Theorem 4.1.

Let *Paths* denote the set of all paths, and let

$$S_e(P_\beta) = \{\bigwedge \Sigma \wedge \neg\varphi \mid \Sigma \cup \{\varphi\} \text{ is a prefix extended subset of } P_\beta\}.$$

Recall the set $S(P_\beta)$ defined in the proof of Theorem 4.1. The *prefix extension function from $S(P_\beta)$ to $S_e(P_\beta)$* is the mapping

$$f : S(P_\beta) \times Paths \to S_e(P_\beta),$$

defined by

$$f(\bigwedge \Sigma \wedge \neg\varphi,\ \alpha) \mapsto \bigwedge_{\phi\in\Sigma} \delta(\phi,\alpha) \wedge \neg\delta(\varphi,\alpha).$$

To prove Theorem 5.1, it suffices to show the proposition below.

Proposition 5.2: Let ψ be a sentence in $S(P_\beta)$, α a path, and f the prefix extension function from $S(P_\beta)$ to $S_e(P_\beta)$. Then

1. ψ is satisfiable iff $f(\psi,\alpha)$ is satisfiable;
2. ψ is finitely satisfiable iff $f(\psi,\alpha)$ is finitely satisfiable. In addition, if ψ has a finite model of size N, then $f(\psi,\alpha)$ has a finite model of size $N+|\alpha|$. ■

For if Proposition 5.2 holds, then $S_e(P_\beta)$ has the small model property for satisfiability. More specifically, given $\phi \in S_e(P_\beta)$, we can determine a path α and $\psi \in S(P_\beta)$ in linear time, such that $\phi = f(\psi,\ \alpha)$. In addition, $|\phi| \le |\psi| + |\alpha|$. If ϕ is satisfiable, then by Proposition 5.2, so is ψ. By Proposition 4.2, ψ has a model of size at most $2^{|\psi|}$. Thus again by Proposition 5.2, ϕ has a model of size at most $2^{|\psi|} + |\alpha|$, which is no larger than $2^{|\phi|}$.

We next show Proposition 5.2.

Proof: We only prove (2) of the proposition. The proof of (1) is similar.

First notice that if $|\alpha| = 0$, then $f(\psi,\alpha) = \psi$. Obviously, the proposition holds in this case. Hence in the sequel, we assume that $|\alpha| \ge 1$.

Let $\psi = \bigwedge \Sigma \wedge \neg\varphi$, and let

$$R_\alpha = \{\rho \mid \rho \text{ is a path, } \rho \prec \alpha\}.$$

Here $\rho \prec \alpha$ means that ρ is a proper prefix of α, as described in Section 2. The proof of the proposition is carried out as follows.

(1) Suppose that ψ has a finite model $G = (|G|,\ r^G,\ E^G)$. We show that $f(\psi,\alpha)$ has a finite model $H = (|H|,\ r^H,\ E^H)$, and moreover, the size of H, $size(H)$, is $size(G) + |\alpha|$.

We construct H as follows. For each $\rho \in R_\alpha$, let c_ρ be a distinguished node not in $|G|$. Let

- $|H| = |G| \cup \{c_\rho \mid \rho \in R_\alpha\}$;
- $r^H = c_\epsilon$;
- For all $a, b \in |H|$ and each $K \in E$, $H \models K(a,b)$ iff one of the following conditions is satisfied:
  - there exists $\rho \in R_\alpha$, such that $a = c_\rho$ and $b = c_{\rho\cdot K}$ and $\rho \cdot K \in R_\alpha$; or
  - there exists $\rho \in R_\alpha$, such that $\alpha = \rho \cdot K$ and $a = c_\rho$ and $b = r^G$; or
  - $a, b \in |G|$ and $G \models K(a,b)$.

Obviously, $size(H) = size(G) + |\alpha|$.

To show that $H \models f(\psi,\alpha)$, first observe the following simple facts, which are immediate from the construction of H.

Fact 1: $\{c \mid c \in |H|,\ H \models \alpha(c_\epsilon, c)\} = \{r^G\}$.

Fact 2: For each $a \in |G|$ and each $c \in |H| \setminus |G|$, there exists no path ρ such that $G \models \rho(a,c)$.

Next, we show that $H \models \bigwedge_{\phi\in\Sigma} \delta(\phi,\alpha) \wedge \neg\delta(\varphi,\alpha)$.

First, suppose, for *reductio*, that there exists $\phi \in \Sigma$ such that $H \models \neg\delta(\phi,\alpha)$. Without loss of generality, assume that ϕ is a forward constraint (the argument for the backward case is analogous). Then there exist $a, b, c \in |H|$, such that

$$H \models \alpha(c_\epsilon, a) \wedge pf(\phi)(a,b) \wedge lt(\phi)(b,c) \wedge \neg rt(\phi)(b,c).$$

By Fact 1, $a = r^G$. By Fact 2, $b, c \in |G|$, and moreover, by the construction of H,

$$G \models pf(\phi)(a, b) \wedge lt(\phi)(b, c) \wedge \neg rt(\phi)(b, c).$$

That is, $G \models \neg\phi$. This contradicts the assumption that $G \models \psi$.

Second, since $G \models \psi$, $G \models \neg\varphi$. Without loss of generality, assume that φ is a forward constraint (the argument for the backward case is analogous). Hence there exist $b, c \in |G|$, such that

$$G \models pf(\varphi)(r^G, b) \wedge lt(\varphi)(b, c) \wedge \neg rt(\varphi)(b, c).$$

By Fact 1, $H \models \alpha(c_\epsilon, r^G)$. Hence by the construction of H,

$$H \models \alpha(c_\epsilon, r^G) \wedge pf(\varphi)(r^G, b) \wedge lt(\varphi)(b, c) \wedge \neg rt(\varphi)(b, c).$$

That is, $H \models \neg\delta(\varphi, \alpha)$.

Hence $H \models f(\psi, \alpha)$. Therefore, H is a finite model of $f(\psi, \alpha)$.

(2) Suppose that $f(\psi, \alpha)$ has a finite model $G = (|G|,\, r^G,\, E^G)$. We construct a finite model of ψ.

Without loss of generality, assume that φ is a forward constraint (the proof for the backward case is analogous). Since $G \models \neg\delta(\varphi, \alpha)$, i.e.,

$$G \models \exists\, x\, y\ (\alpha \cdot pf(\varphi)(r^G, x) \wedge lt(\varphi)(x, y) \wedge \neg rt(\varphi)(x, y)),$$

there exist $a, b, c \in |G|$, such that

$$G \models \alpha(r^G, a) \wedge pf(\varphi)(a, b) \wedge lt(\varphi)(b, c) \wedge \neg rt(\varphi)(b, c).$$

Let

$$m = max\{|pf(\phi)| + |lt(\phi)| + |rt(\phi)| \mid \phi \in \Sigma \cup \{\varphi\}\} \ + \ 1$$

and let $G(a)$ be the m-neighborhood of a in G, as described in Definition 3.3. Clearly, $G(a)$ is a finite structure. We next show that $G(a) \models \psi$.

We first show that $G(a) \models \neg\varphi$. Since $|pf(\varphi)| + |lt(\varphi)| < m$ and $|pf(\varphi)| + |rt(\varphi)| < m$, we have that $b \in |G(a)|$ and $c \in |G(a)|$. Thus by the definition of $G(a)$, we have

$$G(a) \models pf(\varphi)(a, b) \wedge lt(\varphi)(b, c) \wedge \neg rt(\varphi)(b, c).$$

That is, $G(a) \models \neg\varphi$.

Second, we show by *reductio* that for each $\phi \in \Sigma$, $G(a) \models \phi$. Suppose that there exists $\phi \in \Sigma$, such that $G(a) \models \neg\phi$. Without loss of generality, assume that ϕ is a forward constraint (the proof for the backward case is analogous). Then there exist $d, e \in |G(a)|$, such that

$$G(a) \models pf(\phi)(a, d) \wedge lt(\phi)(d, e) \wedge \neg rt(\phi)(d, e).$$

Thus by the definition of $G(a)$, we have

$$G \models \alpha(r^G, a) \wedge pf(\phi)(a, d) \wedge lt(\phi)(d, e) \wedge \neg rt(\phi)(d, e).$$

That is, $G \models \neg\delta(\phi, \alpha)$. This contradicts the assumption that $G \models f(\psi, \alpha)$.

This completes the proof of the proposition. ∎

Note that the proof of Proposition 5.2 does not use any special property of P_β, and therefore, still holds for arbitrary recursive subsets of P. More specifically, given any recursive subset X of P, we can define the function δ for sentences of X in the same way as in Definition 5.1. Similarly, the prefix extended subsets of X can also be defined. Let

$$\begin{aligned} S(X) &= \{\bigwedge \Sigma \wedge \neg\varphi \mid \Sigma \cup \{\varphi\} \text{ is a finite subset of } X\}, \\ S_e(X) &= \{\bigwedge \Sigma \wedge \neg\varphi \mid \Sigma \cup \{\varphi\} \text{ is a prefix extended subset of } X\}. \end{aligned}$$

Define the *prefix extension function from* $S(X)$ *to* $S_e(X)$ as the mapping

$$g : S(X) \times Paths \to S_e(X),$$

such that

$$g(\bigwedge \Sigma \wedge \neg\varphi,\, \alpha) \mapsto \bigwedge_{\phi \in \Sigma} \delta(\phi, \alpha) \wedge \neg\delta(\varphi, \alpha).$$

It is easy to see that the argument for Proposition 5.2 also provides a proof of the corollary below.

Corollary 5.3: Let X be a recursive subset of P, ψ a sentence in $S(X)$, α a path, and g the prefix extension function from $S(X)$ to $S_e(X)$. Then

1. ψ is satisfiable iff $g(\psi, \alpha)$ is satisfiable;
2. ψ is finitely satisfiable iff $g(\psi, \alpha)$ is finitely satisfiable. In addition, if ψ has a finite model of size N, then $g(\psi, \alpha)$ has a finite model of size $N + |\alpha|$. ■

6 Conjunctive path constraints

In this section, we show that the decidability results established in the previous sections also hold true for the conjunctive path constraints defined in Section 2.

We first define fragments of P^c analogous to the fragments of P discussed in the previous sections.

Definition 6.1: A finite subset Σ of P^c is called a *prefix restricted subset* of P^c iff for all ϕ, ψ in Σ, all the paths in $pf(\phi) \cup pf(\psi)$ have the same length.

The *prefix restricted (finite) implication problem for* P^c is the problem of determining, given any finite prefix restricted subset $\Sigma \cup \{\phi\}$ of P^c, whether all the (finite) models of Σ are also models of ϕ. ■

Definition 6.2: A *simple conjunctive path constraint* ϕ is a constraint of P^c with $pf(\phi) = \{\epsilon\}$.

A *β-restricted conjunctive path constraint* ϕ is a constraint of P^c such that for each $\beta \in lt(\phi)$, $|\beta| \leq 1$.

The set of all simple conjunctive path constraints and all β-restricted conjunctive path constraints is denoted by P^c_β. ■

Definition 6.3: Let ρ be a path and ϕ be a constraint in P^c_β. The *extension of* ϕ *with prefix* ρ, denoted $\delta(\phi, \rho)$, is the constraint in P^c defined either by

$$\forall\, x\, y\, (\bigwedge_{\alpha \in pf(\phi)} \rho \cdot \alpha(r, x) \wedge \bigwedge_{\beta \in lt(\phi)} \beta(x, y) \to rt(\phi)(x, y))$$

when ϕ is of the forward form, or by

$$\forall\, x\, y\, (\bigwedge_{\alpha \in pf(\phi)} \rho \cdot \alpha(r, x) \wedge \bigwedge_{\beta \in lt(\phi)} \beta(x, y) \to rt(\phi)(y, x))$$

when ϕ is of the backward form.

Let ρ be a path and Σ be a finite subset of P^c_β. The *extension of* Σ *with prefix* ρ is the subset of P^c defined by

$$\{\delta(\phi, \rho) \mid \phi \in \Sigma\}.$$

Such a set is called a *prefix extended subset of* P^c_β.

The *extended (finite) implication problem for* P^c_β is the problem of determining, given any prefix extended subset $\Sigma \cup \{\phi\}$ of P^c_β, whether all the (finite) models of Σ are also models of ϕ. ■

The following decidability results can be verified analogously to Theorems 3.1, 4.1 and 5.1, respectively.

Theorem 6.1: The prefix restricted implication and finite implication problems for P^c are decidable. ■

Theorem 6.2: The implication and finite implication problems for P^c_β are decidable. ■

Theorem 6.3: The extended implication and finite implication problems for P^c_β are decidable. ■

7 Conclusions

In [10], we introduced a path constraint language, P, and established the undecidability of its associated implication problems. In light of these undecidability results, in this paper we have identified several fragments of P which suffice to express many important integrity constraints such as local database constraints and inverse constraints. We have established the decidability of the implication problems associated with each of these fragments. We have also demonstrated the use of these fragments in optimizing query evaluation and in adding structure to semistructured data. In addition, we have investigated a generalization of P, P^c, and shown that the decidability results for the fragments of P investigated here also hold for the analogous fragments of P^c.

Acknowledgements. The authors thank Victor Vianu, Val Tannen and Susan Davidson for helpful discussions.

References

[1] S. Abiteboul. "Querying semi-structured data". In *Proc. ICDT*, 1997.

[2] S. Abiteboul, S. Cluet, V. Christophides, T. Milo, G. Moerkotte, and J. Siméon. "Querying documents in object databases". *Journal of Digital Libraries*, 1(1), 1997.

[3] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Weiner. "The lorel query language for semistructured data". *Journal of Digital Libraries*, 1(1), 1997.

[4] S. Abiteboul and V. Vianu. "Regular path queries with constraints", In *Proc. ACM Symp. on Principles of Database Systems*, 1997.

[5] F. Bancilhon, C. Delobel, and P. Kanellakis, editors. *Building an object-oriented database system: the story of O2*. Morgan Kaufmann, San Mateo, California, 1992.

[6] J. Barwise. "On Moschovakis closure ordinals". *Journal of Symbolic Logic*, 42:292-296, 1977.

[7] E. Börger, E. Grädel, and Y. Gurevich. *The classical decision problem*. Springer, 1997.

[8] P. Buneman, S. Davidson, M. Fernandez, and D. Suciu. "Adding structure to unstructured data". In *Proc. ICDT*, 1997.

[9] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu. "A query language and optimization techniques for unstructured data". In *Proc. ACM SIGMOD International Conf. on Management of Data*, pp. 505-516, 1996.

[10] P. Buneman, W. Fan, and S. Weinstein. "Some undecidable implication problems for path constraints". Technical Report MS-CIS-97-14, Department of Computer and Information Science, University of Pennsylvania, 1997.

[11] R. G. G. Cattell (ed.). *The object-oriented standard: ODMG-93* (Release 1.2). Morgan Kaufmann, San Mateo, California, 1996.

[12] V. Christophides, S. Cluet, and G. Moerkotte. "Evaluating queries with generalized path expressions". In *Proc. ACM SIGMOD International Conf. on Management of Data*, 1996.

[13] H. B. Enderton. *A mathematical introduction to logic*. Academic Press, 1972.

[14] M. Fernandez, D. Florescu, A. Levy, and D. Suciu. "A query language and processor for a web-site management system". In *Workshop on Management of Semistructured Data*, 1997.

[15] M. Fernandez and D. Suciu. "Query optimizations for semi-structured data using graph schemas". In *Proc. International Conf. on Data Engineering*, 1998.

[16] E. Grädel, P. Kolaitis, and M. Vardi. "On the decision problem for two-variable first-order logic". *Bulletin of Symbolic Logic*, 3(1): 53-69, March 1997.

[17] D. Konopnicki and O. Shmueli. "Draft of W3QS: a query system for the World-Wide Web". In *Proc. of VLDB*, 1995.

[18] C. Lamb, G. Landis, J. Orenstein, and D. Weinreb. "The ObjectStore Database system". *Comm. ACM*, 34(10): 51-63, October 1991.

[19] A. O. Mendelzon, G. A. Mihaila, and T. Milo. "Querying the World Wide Web". In *Proc. PDIS*, pp. 80-91, 1996.

[20] S. Nestorov, S. Abiteboul, and R. Motwani. "Inferring structure in semistructured data". In *Workshop on Management of Semistructured Data*, 1997.

[21] S. Nestorov, J. Ullman, J. Weiner, and S. Chawathe. "Representative objects: Concise representations of semistructured, hierarchical data". In *Proc. Thirteenth International Conf. on Data Engineering*, 1997.

[22] Y. Papakonstantinou, S. Abiteboul, and H. Garcia-Molina. "Object fusion in mediator systems". In *Proc. VLDB*, 1996.

[23] Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. "Object exchange across heterogeneous information sources". In *Proc. Eleventh International Conf. on Data Engineering*, pp. 251-260, March 1995.

[24] D. Suciu. "Query decomposition and view maintenance for query languages for unstructured data". In *Proc. VLDB*, 1996.