Modal logics and Semi-structured Databases

S. Cerrito, Lab. IBISC, Univ. d'Evry Val d'Essonne, FRANCE

july 2007

Contents

1 Introduction

1.1 Relational DB's and Classical Logic

Classical Many-Sorted First-order Logic : A foundation for Relational Databases.

- It can express: Data, Schema, Queries and Constraints.
- It is a foundation for query languages : Relational many-sorted "tuple" languages (on which SQL is founded).
- Classical problems for databases: Query Containment and Constraint Implication:
  QC : Is the set of the answers for query Q_1 included in the set for query Q_2?
  CI : Does the constraint C_1 implies the constraint C_2?
- These are deduction problems (here: in classical logic):
  $Q_1(x) \rightarrow Q_2(x)$ is valid?
  $C_1 \rightarrow C_2$ is valid?

Toy example of a Relational Database.

Film			
Title	MovieDirector	Actor	Producer
nf1	r1	a1	p1
nf1	r1	a2	p1
nf2	r2	a1	p2
nf3	r2	a1	p2

Projection		
Title	Cinema	Time
nf1	nc1	h1
nf1	nc2	h2
nf2	nc1	h3
nf3	nc2	h1

Loves	
Title	Spectator
nf1	s1
nf1	s2
nf2	s1
nf3	s3

- In the example, the signature Σ of a many-sorted language (with sorted equalities) $\mathcal{L}$ consists of:
  - *Basic Sorts* : Title, Cinema, Time, MovieDirector, Spectator Actor, Producer (The attributes).
  - *Sorted constants*: nf1:Title, r1:MovieDirector,etc.(the atomic values in attribute columns).
  - *Sorted Relational symbols* :
    Film, having sort: Title $\times$ MovieDirector $\times$ Producer
    Projection, having sort: Title $\times$Cinema $\times$ Time
    Love, having sort: Title $\times$ Spectator.
    (The names of the relations, with their profiles).
- Terms are either sorted variables ranging over tuples of atomic values or sorted constants.
  The sort of a term is a list of basic sorts. We write: $t : s_1 \cdots s_n$

- Here, Data : Closed Formulae of $\mathcal{L}$:
  Film(nf1, r1, a1, p1), where nf1:Title, r1:Movie Director, etc...
  A DB may be seen as a Herbrand interpretation of $\mathcal{L}$.
- The Schema: the set of sorted relations of the signature.
- The Integrity Constraints: closed formulae of $\mathcal{L}$.
  E.g.: the functional dependency for the table Projection :
  $Title\ Cinema \Rightarrow Time$ is expressed by
  $\forall t : Title\ Cinema\ Time\ \forall t' : Title\ Cinema\ Time$
  $(t.Title\ Cinema = t'.Title\ Cinema \rightarrow t.Time = t'.Time)$
- The Queries: formulae of $\mathcal{L}$.
  E.g.: *At 3p.m where one can see a film, and which one ?*

$$\{x : Title\ Cinema \mid Projection(x.Title\ x.Cinema\ 3.pm\}$$

1.2 Which logic(s) for Semi-structured Databases?

Semi-structured Databases: widely used to integrate data having different formats, e.g. biological data (mediator).

The Web can be seen as a giant Semi-structured Database.

Central Question of this tutorial: is there a logic able to provide a foundation for semi-structured Database ?

Several proposals using modal logics have been made in recent years.

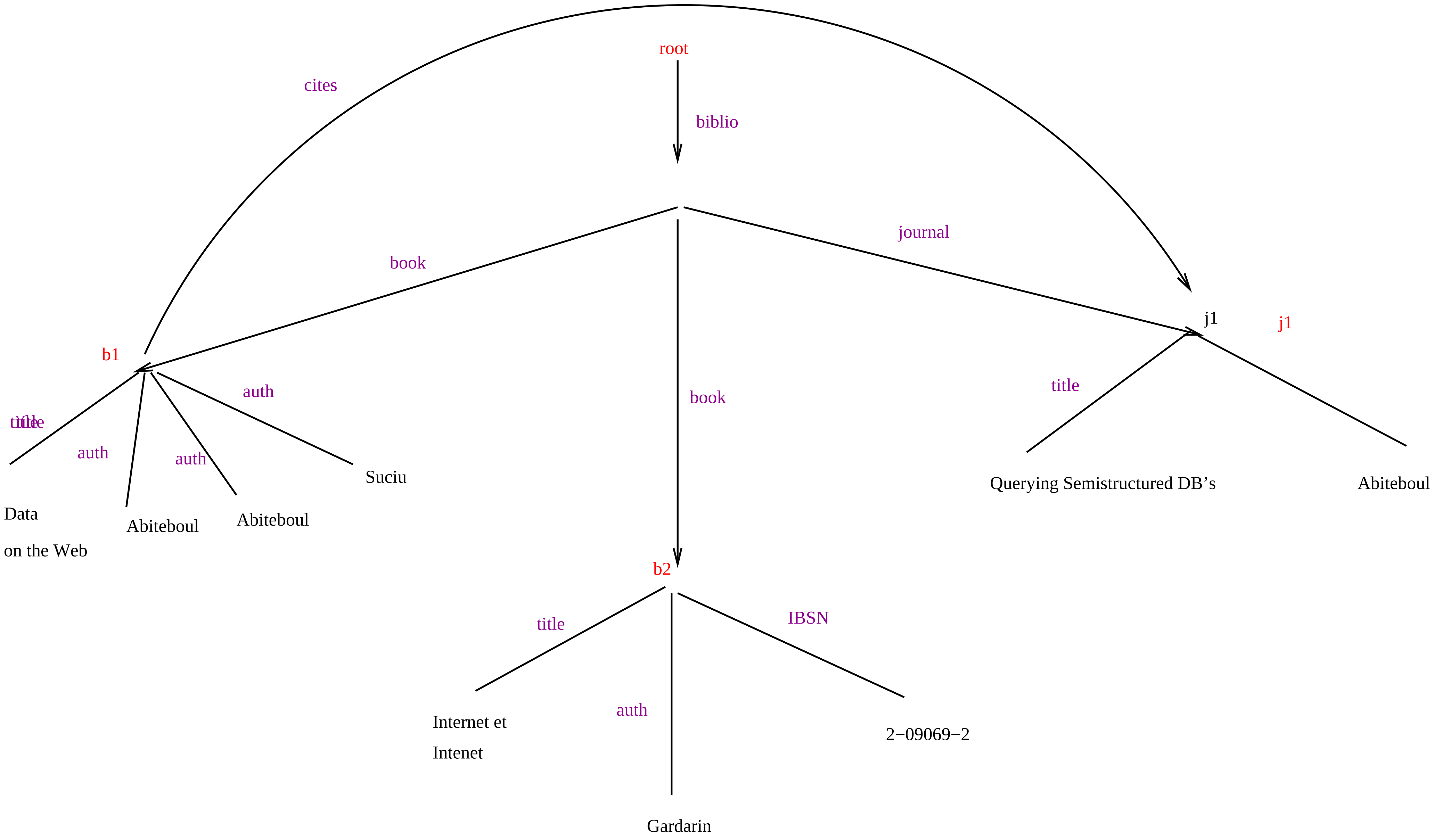
No claim to be exhaustive.

Accent on deductive problems.

A toy example of semi-structured database represented as a graph with labelled edges where one could ask the query:

Which are the authors of the book whose title is “Data on the Web”?

$$\Longrightarrow$$


root
biblio
cites
book
journal
b1
j1
j1
title
title
auth
auth
auth
Data
on the Web
Abiteboul
Abiteboul
Suciu
book
title
Querying Semistructured DB's
Abiteboul
b2
title
IBSN
auth
Internet et
Intenet
2−09069−2
Gardarin

In the corresponding XML-document:

- Each group of infos on a book is an *XML-element* whose *tag* is `book`.
- It is a complex element, having several sub-elements:
  one or more authors, a title, and, possibly, anXS IBSN number.
- Any element may have a *XML-attribute* describing a property: in particular, an attribute may identify an element, or point (reference) to another element. Here, `cites` is a referential attribute.
- Leaves correspond to atomic elements.

In the figure:

In red: node names (identifiers).

In magenta: arc labels (XML-tags, but for “cites” which is a reference).

In black: data.

A XML document for our example

```
<biblio>
<book key=''b1'' cite=''j1''>
<title> Data on the Web </title>
<author> Abiteboul </author>
<author> Bunemann </author>
<author> Suciu </Abiteboul>
</book>
<book key=''b2''>
<title> Iternet et Intranet </title>
<author> Gardarin </author>
<IBSN> 2-09069-2 </Abiteboul>
</book>
<journal key=j1>
<title> Querying Semi-structured Databases </title>
<author> Abiteboul </author>
</journal>
</biblio>
```

N.B.: `key` and `cite` are *XML-attributes* with different roles : the first allows one to name nodes, the second to point to a named node (reference links ⇒ cycles).

Two sorts of schemas for XML-DB's : **DTD**'s (*Data Type Definitions*) and **XML-Schema**s (richer typing system).

A DTD w.r.t. our example is "valid"

```
<!DOCTYPE biblio [
<!Element biblio (book*,journals*)>
<!Element book (title,author*,IBSN?)>
<!ATTLIST book key ID REQUIRED>
<!ATTLIST book cite IDREFS IMPLIED>
<!Element journal (title,author*)>
<!ATTLIST journal key ID REQUIRED>
<!Element title #PCDATA>
<!Element author #PCDATA>
<!Element IBSN #PCDATA>
]>
```

ID is the type of attributes identifying (naming) nodes, IDREFS the type of "pointer (reference) attributes".
REQUIRED means that the attribute is mandatory, IMPLIED that it is optional.

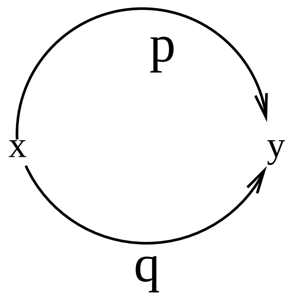
2 Using Propositional Dynamic Logics

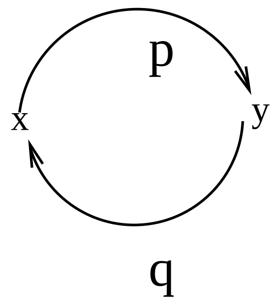
2.1 The logic PDL^{path}

- A pioneer work on modal logics and Path Constraints: [1](N. Alechina, S. Demri and M. de Rijk, 2003)
- It uses a version of PDL to model semi-structured DB's.
- Centered on path constraints.
- Knowledge about integrity constraints is essential to querying any DB!

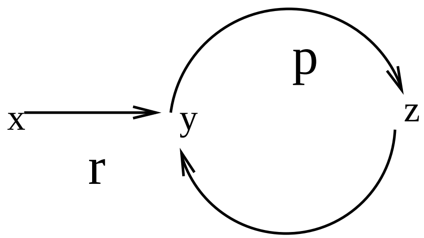
Focus on 3 classes of path-constraints:

Inclusion Constraint





Backward Constraint



Lollipop Constraint

Let L be a countable set of labels.

Syntax of transition expressions of PDL^{path}:

$$t ::= l \in L \mid \epsilon \mid \# \mid t;t \mid t+t \mid t^* \mid t^{-1}$$

Elements of L will be used to label arcs (but tags are not attached).

Semantics of transition expressions (a):

Transition expressions evaluated on L-structures. A L-structure G : a tuple of the form $\langle V, rt, (R_a)_{a \in L} \rangle$ such that V is a set of nodes, rt is a distinguished element of V (the root of G) and $(R_a)_{a \in L}$ is a family of binary relations on V.

Let's note $Cl(r)$ the reflexive transitive closure of a binary relation r.

Semantics of transition expressions (b):

Given an L-structure, the interpretation of a transition expression t is denoted by $tr(t)$ and is defined by:

$tr(a) = R_a$ for $a \in L$	$tr(\epsilon) = \{\langle v, v\rangle \mid v \in V\}$
$tr(\#) = \bigcup_{a \in L} R_a$	$tr(t^*) = Cl(tr(t))$
$tr(t_1; t_2) = \{\langle u, v\rangle \mid \exists z(tr(t_1)(u, z) \wedge tr(t_2)(z, v))\}$	$tr(t_1 + t_2) = tr(t_1) \cup tr(t_2)$
$tr({t_1}^{-1}) = \{\langle u, v\rangle \mid \langle v, u\rangle \in tr(t)\}$	

Syntax of path formulae of PDL^{path}:

$$\phi ::= T \mid \perp \mid root \mid \neg\phi \mid \phi \wedge \phi \mid \langle t \rangle \phi \mid [t]\phi$$

where t is any transition expression. NB: $T, \perp$ and $root$ are he unique atomic formulae.

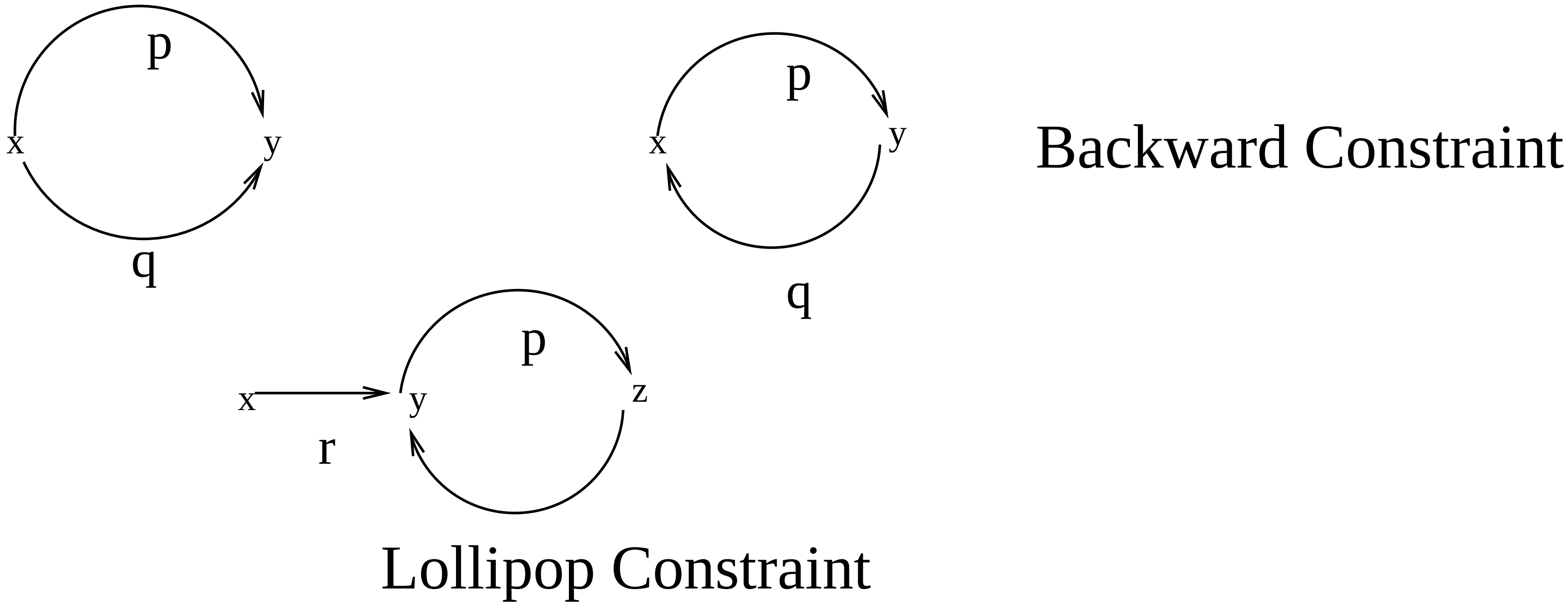
Semantics of path formulae of PDL^{path}:

The usual multi-modal one, but the nominal $root$ is true only at rt (the root of the L-structure G, i.e. the edge labeled graph G) and the accessibility relation associated to $\langle t \rangle$ and $[t]$ is $tr(t)$.

A PDL^{path} formula is true at G if it is true at its root. It is valid if it is true at each G.

Inclusion constraints and Backward Constraints can be expressed in PDL^{path}:

Inclusion Constraint



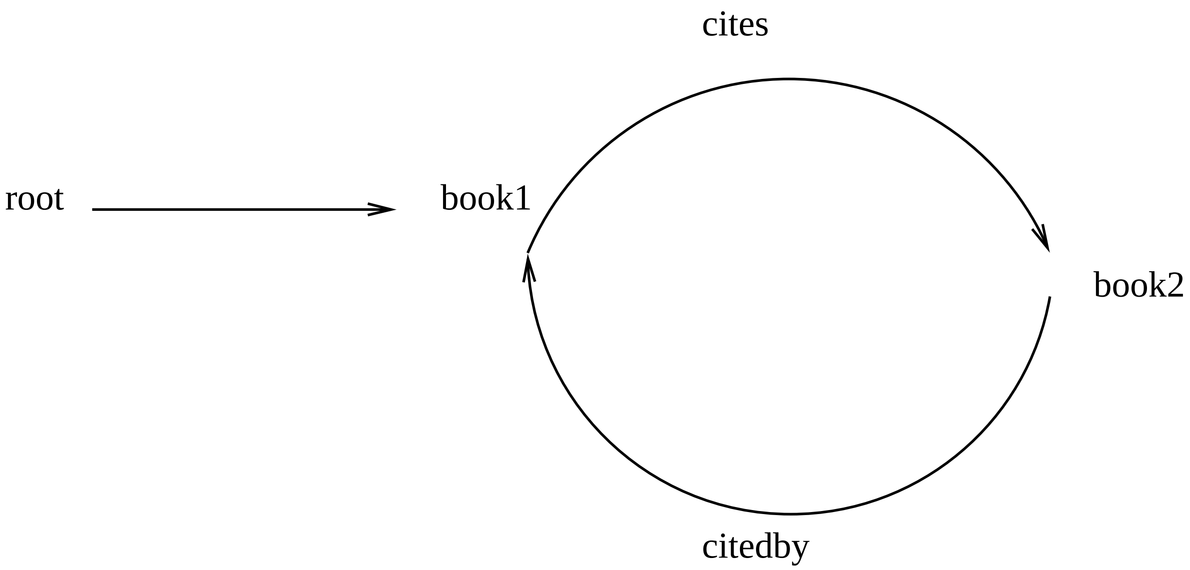
Inclusion: $[p] < (q)^{-}1 > root$ Backward:$[p] < q > root$

Lollipop constraint cannot. Why ?

The validity problem for PDL^{Path} is decidable.

The containment problem for lollipop constraints is undecidable [3]

Use of "references" may create lollipop constraints !



- The web can be assimilated to a giant semi-structured database (rooted edge labeled graph).
- An L edge labeled graph is *deterministic* if for every node u and label a there is at most one node v such that $\langle u, v \rangle \in tr(a)$.
  "In the case of the web it is reasonable to expect a graph to be deterministic".[1]
- The tables below present the complexity results of [1] w.r.t. to deterministic and non-deterministic graphs. By "constraint", *tout court*, one means any type of path constraint.

Constraint Evaluation Problem

	non-det. graphs	det-graps
Inclusion c.	NLOGSPACE-complete	NLOGSPACE-complete
Backward c.	NLOGSPACE-complete	NLOGSPACE-complete
constraints	NLOGSPACE-complete	NLOGSPACE-complete

Constraint Containment Problem

	non-det. graphs	det-graps
Inclusion c.	PSPACE-hard, in EXPTIME	open*
Backward c.	PSPACE-hard, in EXPTIME	in EXPTIME for finite L
constraints	undecidable	undecidable

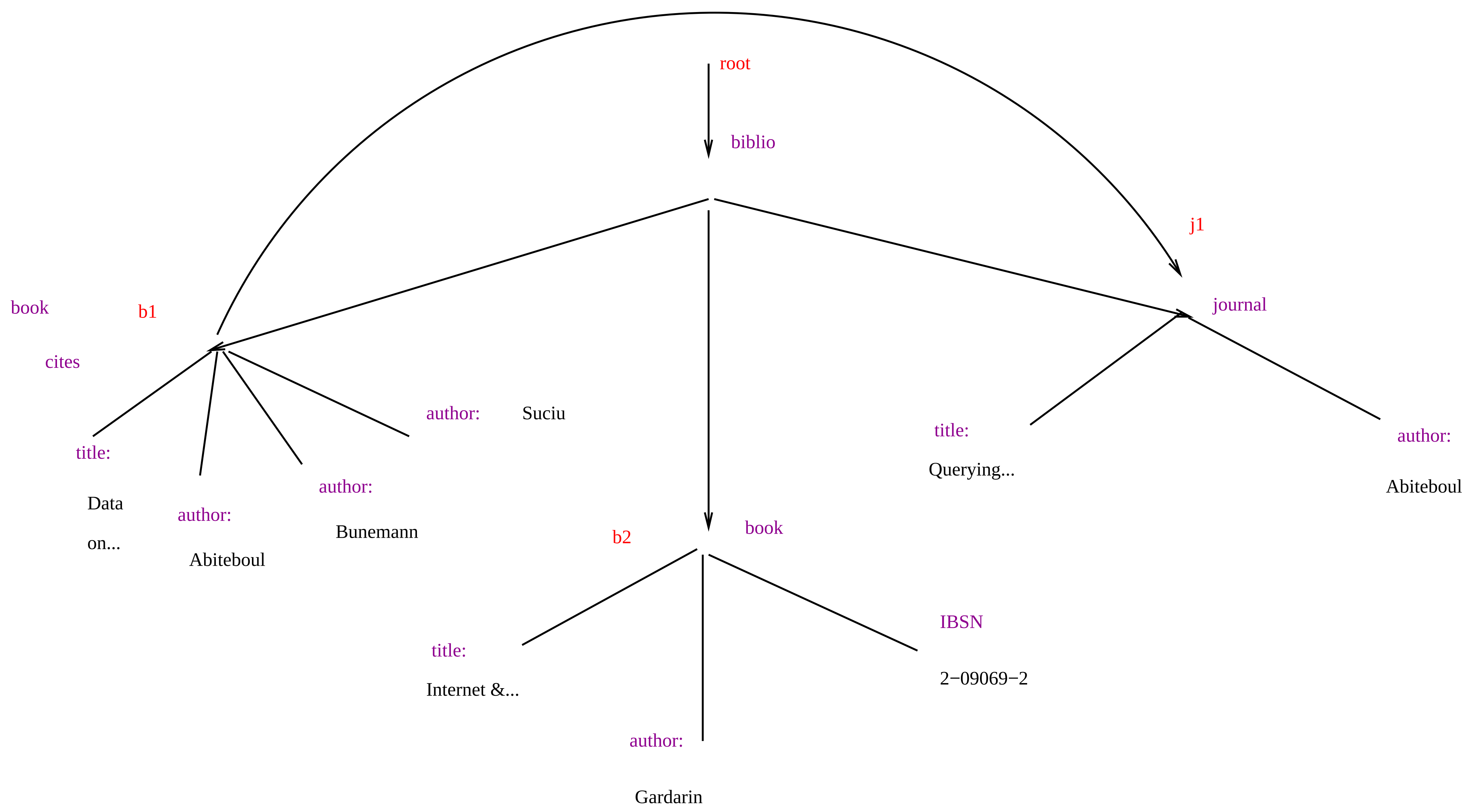
Remark

The use of T, $\perp$ and *root* as the only atomic formulae prevents one to model:

- information attached to an XML-internal node (values for XML-attributes).
- In particular, unique identifiers for internal nodes
- Data on leaves.

2.2 Using other variants of PDL

In [6] M. Marx (2003) proposes 2 variants of PDL logic to model XML-data. In both cases, *labels are not attached to edges but to nodes* (of the tree/graph).


root
biblio
j1
journal
book
b1
cites
author:
Suciu
title:
Querying...
author:
Abiteboul
title:
Data
on...
author:
Abiteboul
author:
Bunemann
b2
book
IBSN
2−09069−2
title:
Internet &...
author:
Gardarin

2.2.1 A modal logic for finite trees : $\mathcal{L}_K$

Here, references are not taken into account.

P: a non-empty, finite or countably finite set of atoms.

Transition expression

$$t ::= \rightarrow | \leftarrow | \uparrow | \downarrow | \ \pi ; \pi \ | \ \pi + \pi \ | \ \pi^* \ | ?\phi$$

where ϕ is a path formula.

Path formulae

$$\phi ::= p \in P \mid T \mid \neg\phi \mid \phi \wedge \phi \mid < t > \phi \mid [t]\phi$$

$\mathcal{L}_K$ is interpreted on finite ORDERED trees, i.e. tuples $\mathbf{T}{=}\langle T, R_{\rightarrow}, R_{\downarrow}\rangle$ where T is the set of nodes and $R_{\rightarrow}, R_{\downarrow}$, are respectively, the left-brother and the child relation. No references! To get an interpretation $\mathcal{M}$, one adds, as usual, valuations for atoms.

The interpretation of a transition expression t, denoted by $tr(t)$, is defined by:

$tr(\downarrow) = R_{\downarrow}$	$tr(\uparrow) = (R_{\downarrow})^{-}1$
$tr(\rightarrow) = R_{\rightarrow}$	$tr(\leftarrow) = (R_{\rightarrow})^{-}1$
$tr(t^*) = Cl(tr(t))$	$tr(t_1; t_2) =$ concatenation
$tr(t_1 + t_2) =$ union	$tr(?\phi) = (w, w) \mid w \in T \wedge \mathcal{M}, w \models \phi$

Given this def. of transition expressions, the interpretation of formulae is straightforward.

Abbreviations

- $root :\neg <\uparrow> T$
- $leaf : \neg <\downarrow> T$
- $first : \neg <\leftarrow> T$
- $last : \neg <\rightarrow> T$

Query Languages for XML : XPATH, XQUERY (extension of XPATH), Lorel,...

XPATH queries are easily translated into formulae of this logic.

Translation of a DTD into $\mathcal{L}_K$

```
<!ELEMENT Collection (Painter+)>
<!ELEMENT Painter (Name, Painting*)>
<!ELEMENT Name CDATA>
<!ELEMENT Painting CDATA>
```

Collection $\leadsto \langle\downarrow; ?first; ?Painter; (\rightarrow ?Painter)* > last$

Painter $\leadsto \langle\downarrow; ?first; Name; (\rightarrow ?Painting)* > last$

Name $\leadsto \langle\downarrow; ?first; ?CDATA > last$

Painting $\leadsto \langle\downarrow; ?first; ?CDATA > last$

2.2.2 A modal logic for finite DAG's : $\mathcal{L}_P$

So far, references have not been taken into account. A restriction on the syntax of $\mathcal{L}_P$ and a modification of the semantics allows one to do it, provided that references do not create cycles (e.g. b1 $\xrightarrow{cites}$ j1 $\xrightarrow{cites}$ b1 is forbidden).

- **Syntax**: the brotherhood axes $\rightarrow$ and $\rightarrow$ are removed, nominals are added.
- **Semantics**: interpretation not on *any* graph, but on finite DAG's.
- The hybrid @ operator is definable in $\mathcal{L}_P$:
  $@_i\phi =_{def} <\uparrow^*> (root \wedge <\downarrow^* (i \wedge \phi) >$
- An algorithm deciding the consequence problem of this logic is given.
  It uses *mosaic style* proof techniques (idea: existence of a model is equivalent to existence of a finite set of partial models).
  EXPTIME complexity.

Remark: In $\mathcal{L}_P$ DTD's cannot be always expressed:
`<ELEMENT! Collection(Painter,Painting)+>` cannot be formalized.

3 Using Hybrid Logics

3.1 The "standard" multi-modal hybrid logic $HL(@, \downarrow)$

Syntax

- A non-empty set L of labels.
- Nom = set of propositional letters called nominals. X = set of variables. $Nom \cup X$= state expressions.
- P = set of propositional letters disjoint from Nom. $A = Nom \cup P$= set of atoms.
- A grammar for the Formulae of $HL(@, \downarrow)$:

$$\phi := p \mid \neg\phi \mid \phi \wedge \phi \mid [l]\phi \mid \; < l > \phi \mid @_s\phi \mid \; \downarrow x.\phi$$

where $p \in A$, $l \in L$, s is a state expression and $x \in X$.

Semantics, preliminaries

- An interpretation $\mathcal{M}$ is any edge labeled graph (an L-structure having W as set of nodes), equipped with an evaluation function I assigning a set of states to each $p \in P$ and a function N assigning a unique state to each $n \in Nom$.
- A subset r_l of $W \times W$ is associated to each label $l \in L$ (labelled transition).
- Let $\mathcal{M}$ be an interpretation and g a variable-evaluation function $X \to W$. For each state expression e, its interpretation $\mathcal{M}_g(e)$ is $N(e)$ if $e \in Nom$ and is $g(e)$ if $e \in X$.
- Notation: if w is a state, ${g^x}_w$ denotes the x-variant of g whose value for the variable x is w.

Semantics, full definitions

1. $\mathcal{M}, w, g \models p$ if $w \in I(p)$, for $p \in P$.
2. $\mathcal{M}, w, g \models e$ if $\mathcal{M}_g(e) = w$, if e is a state expression.
3. $\mathcal{M}, w, g \models \neg\phi$ if $\mathcal{M}, w, g \not\models \phi$.
4. $\mathcal{M}, w, g \models \phi \wedge \psi$ if $\mathcal{M}, w, g \models \phi$ and $\mathcal{M}, w, g \models \psi$.
5. $\mathcal{M}, w, g \models [l]\phi$ if for each w' such that $w \; r_l \; w'$, $\mathcal{M}, w', g \models \phi$.
6. $\mathcal{M}, w, g \models < l > \phi$ if there exists w' such that $w \; r_l \; w'$, and $\mathcal{M}, w', g \models \phi$.
7. $\mathcal{M}, w, g \models @_e\phi$ if $\mathcal{M}, \mathcal{M}_g(e), g \models \phi$
8. $\mathcal{M}, w, g \models \downarrow x.\phi$ if $\mathcal{M}, w, {g^x}_w \models \phi$

A formula ϕ is *satisfiable* if there exist $\mathcal{M}$, w and g such that $\mathcal{M}, w, g \models \phi$.

3.2 An early attempt to model semi-structured-DB

V. Thion's PHd's thesis (2004) and [2] (Bidoit, Cerrito, Thion) constitute a first step towards using hybrid logic as a uniform framework to express data, schema, constraints an queries for semi-structured DB's. A DB is seen as an edge labeled graph s.t. when references are omitted a tree is left. Cycles created by references are allowed.

V. Thion's PHd's thesis:

* Nominals express node identifiers, propositional letters in P express data when they are attached to leaves and XML-attributes when they are attached to internal nodes. $root$: a special nominal.
* The set of labels L is partitioned in 2 classes : T which expresses "ordinary transitions" (DTD's tags) and REF, which expresses reference links.
* An additional modal operator $< F >$ is used: the associated transition is the transitive closure of $\bigcup_{r_t \ t \in T}$.
* An enriched notion of schema is defined which allows one to "well type" reference targets (e.g. a person cannot be a `father_of` a dog).
* Any schema is formalizable, via a general algorithm $\mathcal{A}$.

In Thion's thesis:

- Examples of constraints: (label $cites \in REF$)

  "if x cites y, then y cites x":
  $@_{root}(@_x[cites] < cites > x)$

  "if x cites y and y cites z, then x cites z:
  $@_{root}(@_x[cites][cites] \downarrow z.(@_x < cites > z)$

- Example of query: "Which are the authors of the book whose name is "Data on the Web" ? :
  one looks for values of x satisfying
  $@_{root} < biblio >< book > (< title > DataOnTheWeb \wedge < author > x)$

Positive features of Thion's work

:-) One of the first works trying to use HL to model semi-structured DB's.
:-) Uniformity of language
:-) Rather rich expressivity
:-) Good typing of references.

Weakenesses

:-(The formalized notion of schema is not really DTD, but a bit more awkward one ("pattern grammar") which should be simplified.
:-(Semantics is given by unordered graps $\rightsquigarrow$ proposed notion of schema does not allows one to distinguish between `collection (painter,painting)+` and `collection(painting,painter)+`.
:-(Logical foundations of query evaluation and optimization are not studied.
:-(Semantically interesting classes of queries/constraints such that query containment and constraint implication problems are decidable are not yet studied.

3.3 Hybrid Model Checking and Evaluation

- Problem of global model checking for hybrid languages: given an interpretation $\mathcal{M}$, find the states where ϕ is satisfied. If only 1 free variable appears in ϕ, the output is a set of nodes.
- In [5] (Massimo Franceschet and Maarten de Rijke 2005) hybrid operators are added to several modal logics (temporal, dynamic..). Moreover, a large set of hybrid operators besides @ et $\downarrow$ are considered.
- A variety of model checkers are considered, and their complexity is studied, w.r.t.: number of nodes, number of the (labeled) edges of the graph, length of ϕ, nesting degree of binders in ϕ. One goes from linear complexity to PSPACE-complexity (unrestricted use of binders).
- The model checkers are used to evaluate (translations of) classes of queries expressed in Lorel.

An example in [5]

Which are the papers which have at least two authors?

In Lorel:

```
select X
from biblio.paper X, X. author Y, X.author Z
where Y !=Z
```

(In Xquery:

```
for $X in biblio.paper, let $N:= X.author let $M:= X.author
where $N!=$M
return $X
```

)

Formalization into a variant of hybrid dynamic logic with converse operator:

$$\downarrow x. < biblio.paper >^{-1} root \wedge < author > \downarrow y.@ \downarrow z.\ y \neq z$$

which can be model-checked.

An other example in [5]

Which are the papers which cite themselves?

In Lorel:

```
select X
from biblio.paper X, X.cite Y where X=Y
```

Formalization into a variant of hybrid dynamic logic with converse operator:

$$\downarrow x. < biblio.paper >^{-1} root \wedge < cite > \downarrow y.\ x = y$$

which can be model-checked.

3.4 Containment/implication problems in hybrid logic formalization.

A study of the tractability of these deductive problems can benefit from the knowledge of the following results for $HL(@, \downarrow)$ proven in [4] (B. ten Cate and M. Franceshet 2005). (The multi-modality does not play any role.)

1. Satisfiability of $HL(\downarrow)$ is undecidable.
2. Satisfiability of $HL(@)$ is decidable.
3. Satisfiability of $HL(@, \downarrow) \setminus (\Box \downarrow)$ is decidable.
4. Satisfiability of $HL(@, \downarrow) \setminus (\downarrow \Box)$ is decidable.
5. Satisfiability and validity of $HL(@, \downarrow) \setminus (\Box \downarrow \Box, \Diamond \downarrow \Diamond)$ are decidable.

4 Concluding Remarks

Two questions:

1. Some of the presented works attempt to formalize DTD's. What about XML-schemas, which have a richer typing structure ?

2. Several ideas come from different works and/or different modal logics to provide a formal semantics to semistructured-databases:
   non-atomic edge labels, converse operator, regular expressions for paths from dynamic logics, nominals as node identifiers, possibility to use binders from hybrid logics, etc.

   Each work presented here has advantages and drawbacks. Could one bring together all these ideas so as to define a really unified modal logic capable to found semi-structured databases as well as possible ?

References

[1] N. Alechina, S. Demri, and M. de Rijke. A modal perspective on path constraints, 2003.

[2] N. Bidoit, S. Cerrito, and V. Thion. A first step toward modelling semistructured data in hybrid multi-modal logic. *JANCL*, 14(4), 2004.

[3] Peter Buneman, Wenfei Fan, and Scott Weinstein. Path constraints on semistructured and structured data. pages 129–138, 1998.

[4] M. Franceschet and B. ten Cate. On the complexity of hybrid logics with binders. In *CSL*, pages 339–354, 2005.

[5] Massimo Franceschet and Maarten de Rijke. Model checking for hybrid logics. *Journal of Applied Logic*, 2005.

[6] M. Marx. Xpath and modal logics of finite dags. In M. Cialdea Mayer and F. Pirri, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2003, Rome, Italy,*

September 9-12, 2003. Proceedings, volume 2796 of *Lecture Notes in Computer Science*. Springer, 2003.