

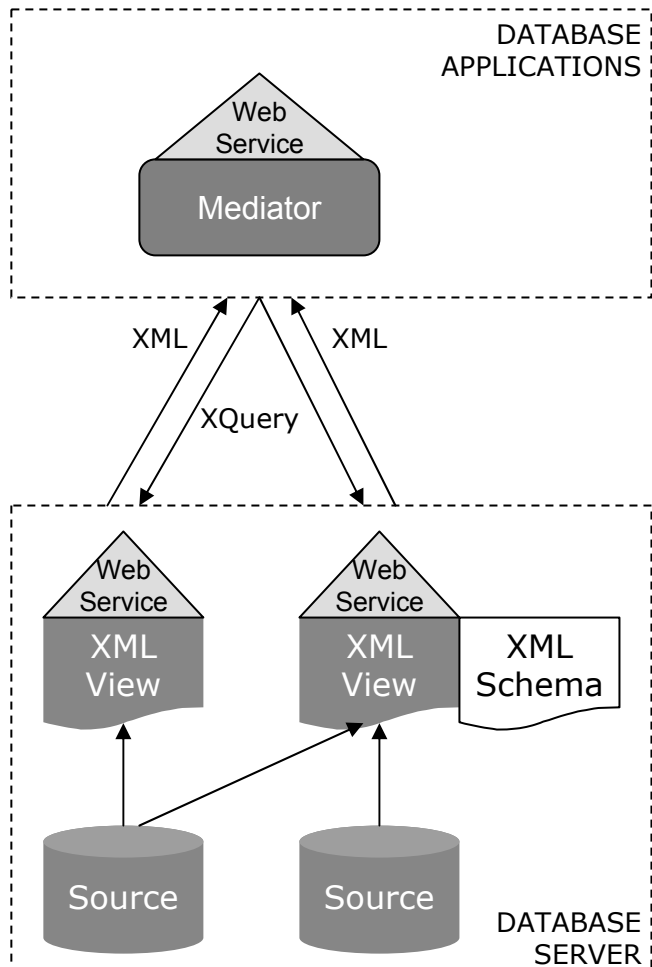
# **Query Set Specification Language (QSSL)**

---

**Michalis Petropoulos**  
**Alin Deutsch**  
**Yannis Papakonstantinou**  
UNIVERSITY OF CALIFORNIA, SAN DIEGO

June 2003

# Exporting DBMSs on the Web



- Exporting Query Capabilities on the Web
  - Web Services (Function Signatures)
- Integrating Web Applications
  - Use Web Services
  - Export Query Capabilities Themselves

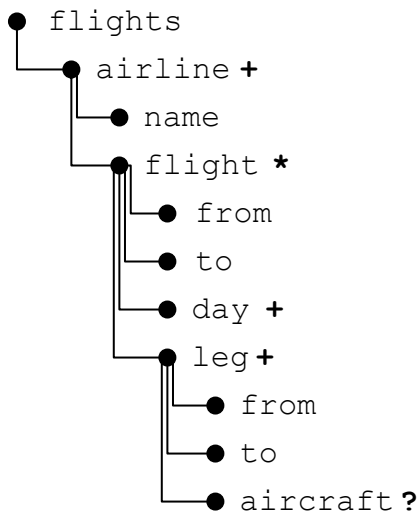
# Overview

---

- Query Set Specification Language (QSSL)
  - Describes Parameterized Tree Pattern (TP) Queries
- Data Services
  - Web Services for Query Capabilities
- Authoring Interface

# Motivation for QSSL (I)

---



- Any combination of the following conditions on:
  - the name of the airline company
  - the origin and destination of one or more flights (optional)
  - a day of the week
  - the origin of zero or more legs (optional)
  - the destination of zero or more legs (optional)
  - the aircraft used for zero or more legs (optional)
- The queries may return “airline” or “flight” elements

# Motivation for QSSL (II)

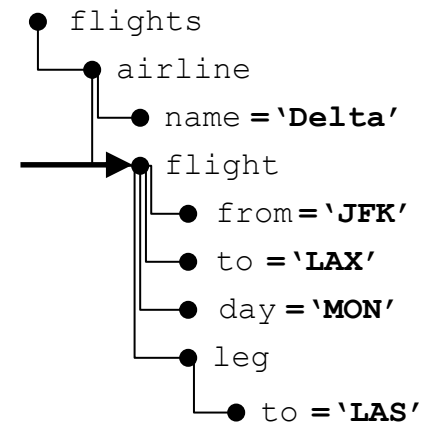
---

- Web Services published as function signatures:
  - Fixed number of input and output parameters
  - Do not capture the functionality of databases
  - Large number of web services needed
    - One function signature for every parameterized query
  - Do not capture the semantic connections the available functions have with each other and with the underlying databases
- JDBC Interfaces
  - All possible queries

# Query Set Specification Language

## Query Language

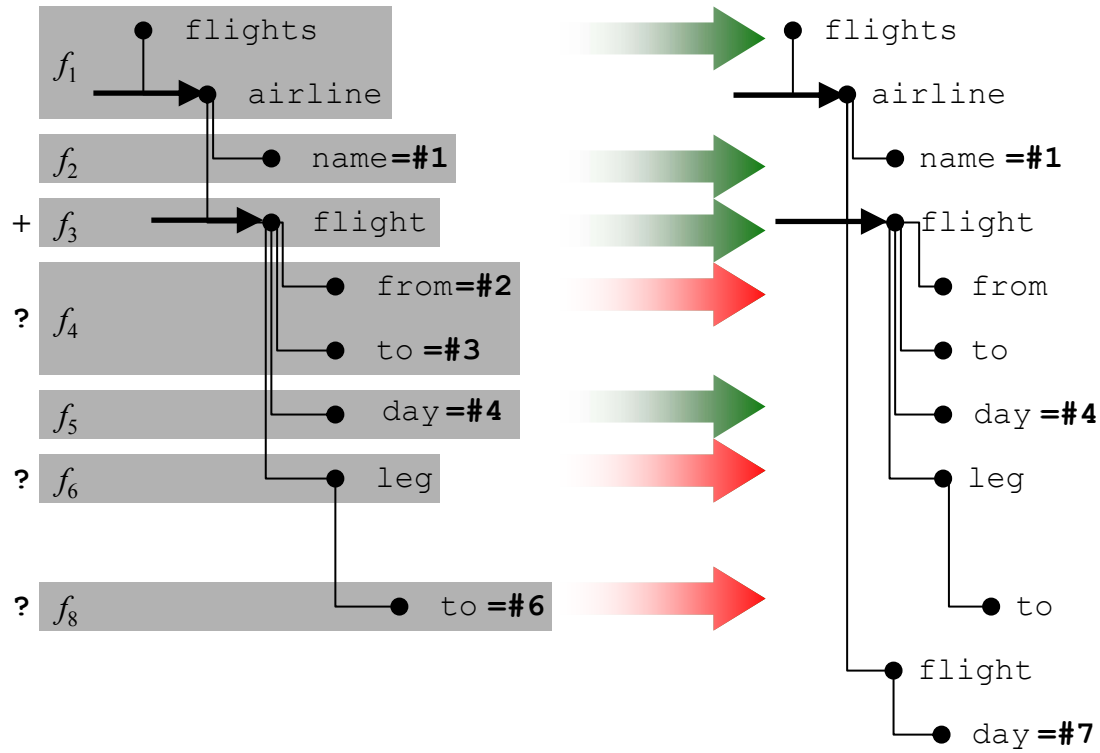
- Tree Pattern Queries:
  - Acyclic XPath expressions consisting of:
    - node tests
    - child axis '/'
    - descendant axis '//'
    - predicates '['']
  - Widely used in current applications
  - Building blocks of XQuery
  - Excellent visual paradigm for GUIs



```
flights/airline[name='Delta']/flight[from='JFK'][to='LAX'][day='MON'][leg[to='LAS']]
```

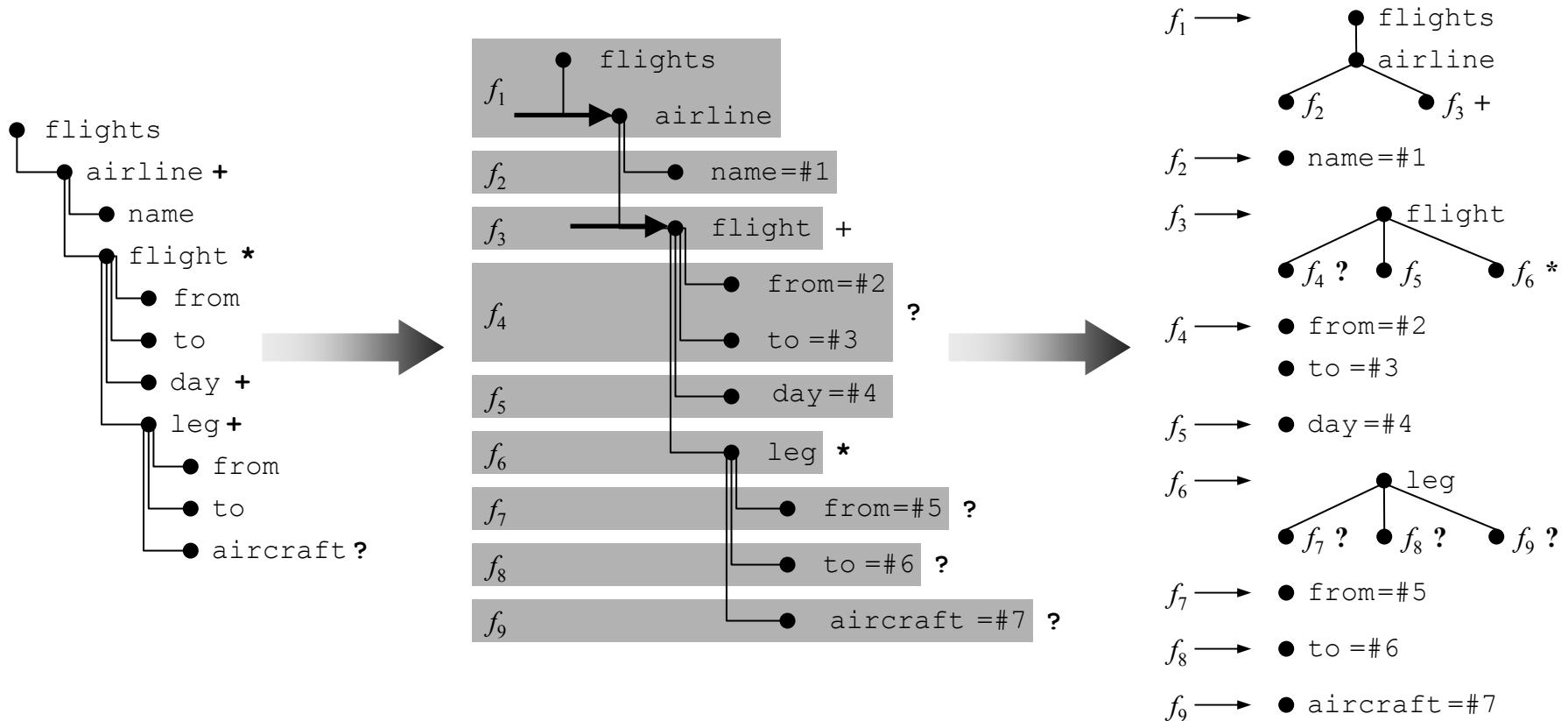
# Query Set Specification Language

## Query Set Specification



# Query Set Specification Language

## Query Set Specification

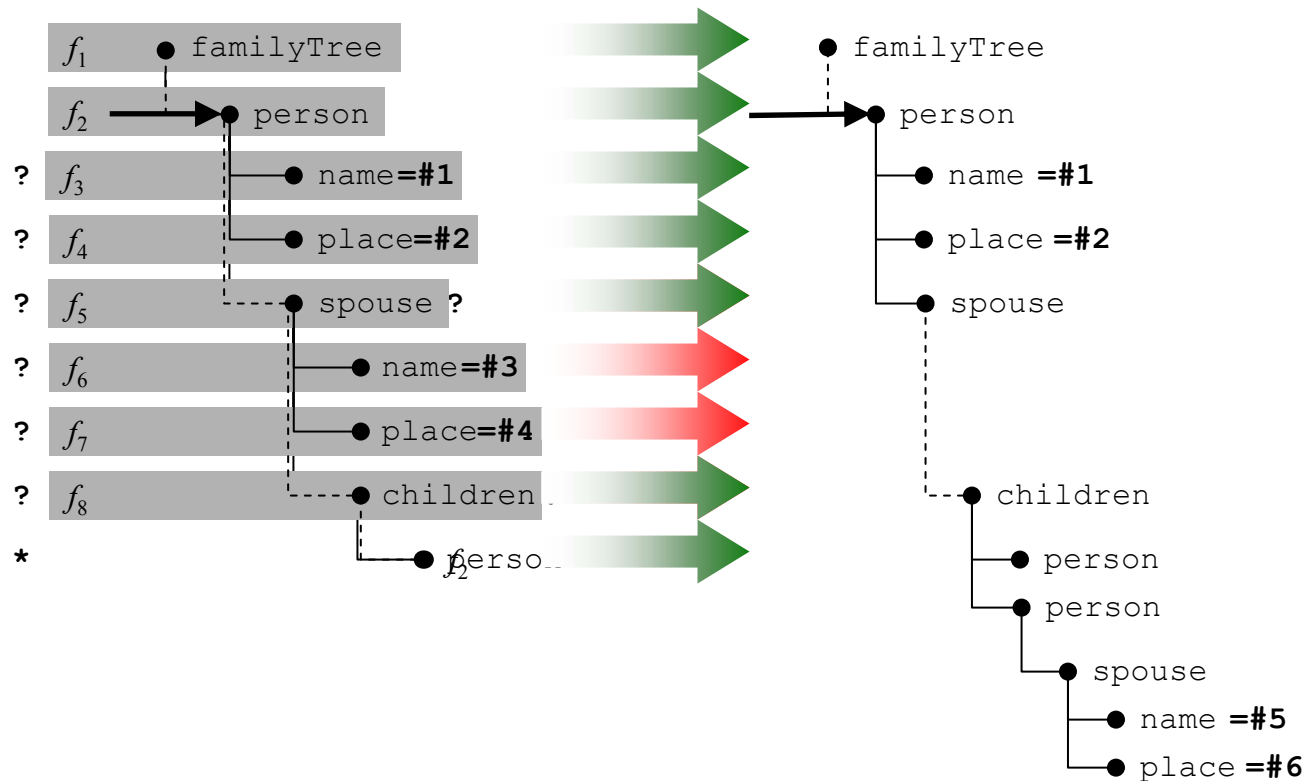


- Similar to extended context-free grammars



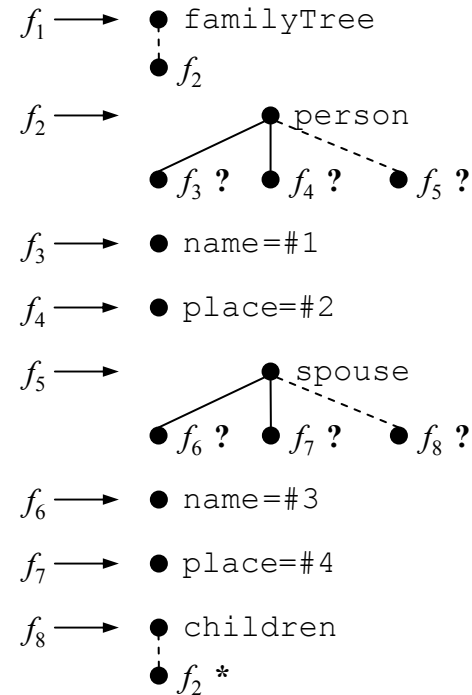
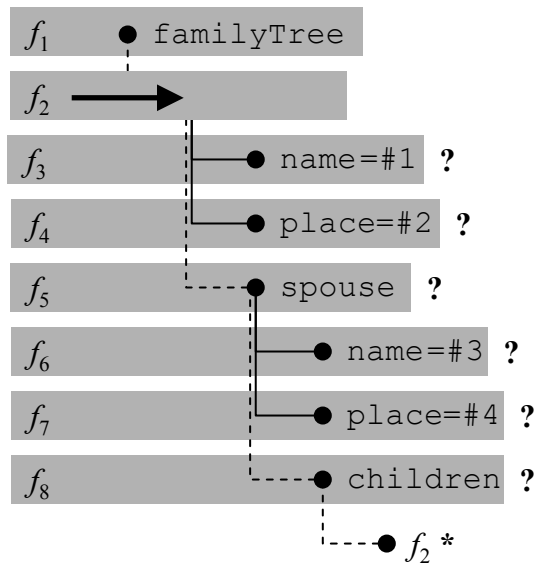
# Query Set Specification Language

## Recursive XML Schemas



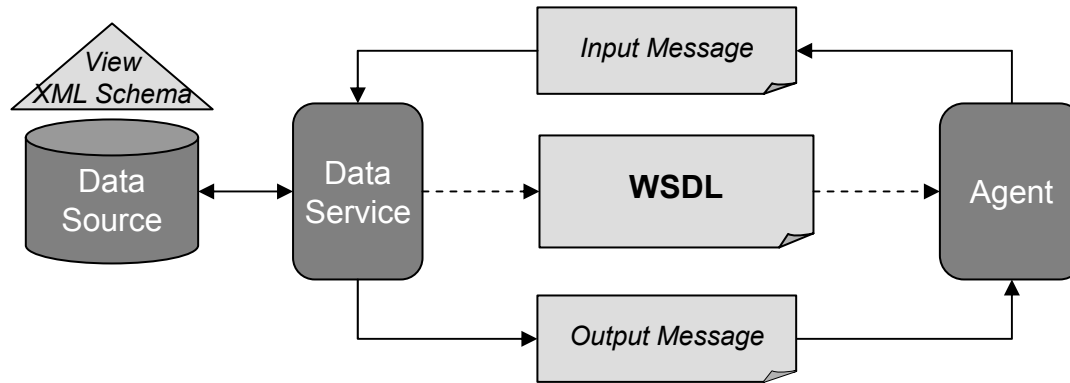
# Query Set Specification Language

## Recursive XML Schemas



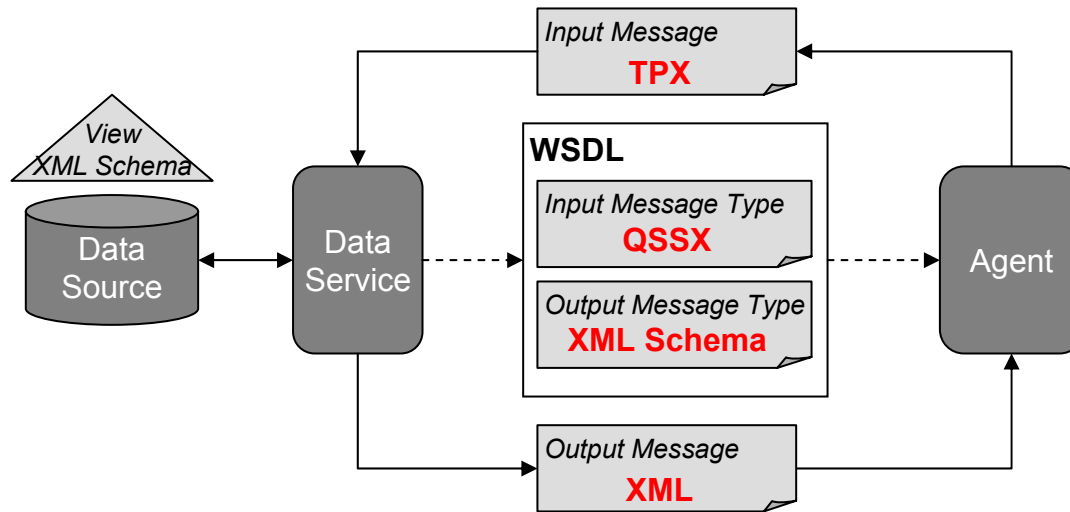
- QSS of fixed size

# Data Services



- **Data Service** = WSDL + QSS
- A QSS deployed as a web service
- Exports the XML Schema of an XML view
- Connects the WSDL calls with the underlying database
- Receives queries that are encoded by QSS
- Explicit relationship between input and output

# Data Services



- QSS is translated to XML Schema (QSSX)
- TP queries that are encoded as XML (TPX)
- Query result is described by an XML Schema

# Data Services

---

## Reasoning

1. Membership of a query in a data service
  2. Subsumption of data services
  3. Totality of a data service
  4. Overlap of data services
- QSS can be translated to an equivalent top-down nondeterministic unranked tree automaton
  - Problems are decidable

# Data Services

## Authoring Interface

The screenshot shows the QSSX Editor interface for editing flight data. The main window displays a tree view of the data structure, with the following elements:

- EST Root
  - flights
    - airline
      - name = #0
      - flight
        - to = #1
        - from = #2
        - day = #3
        - leg
          - to = #4
          - from = #5
          - aircraft = #6

### Selected Element Information:

Name:  
Occ:  
Type:

### Fragments:

Name	Multiplicity	Fragment Color
f1	1	Yellow
f2	1	Light Blue
f3	+	Light Green
f4	?	Pink
f5	1	Yellow
f6	*	Magenta
f7	?	Cyan
f8	?	Light Green

New Fragment

Edit Fragment

Delete Fragment

### Fragment Elements:

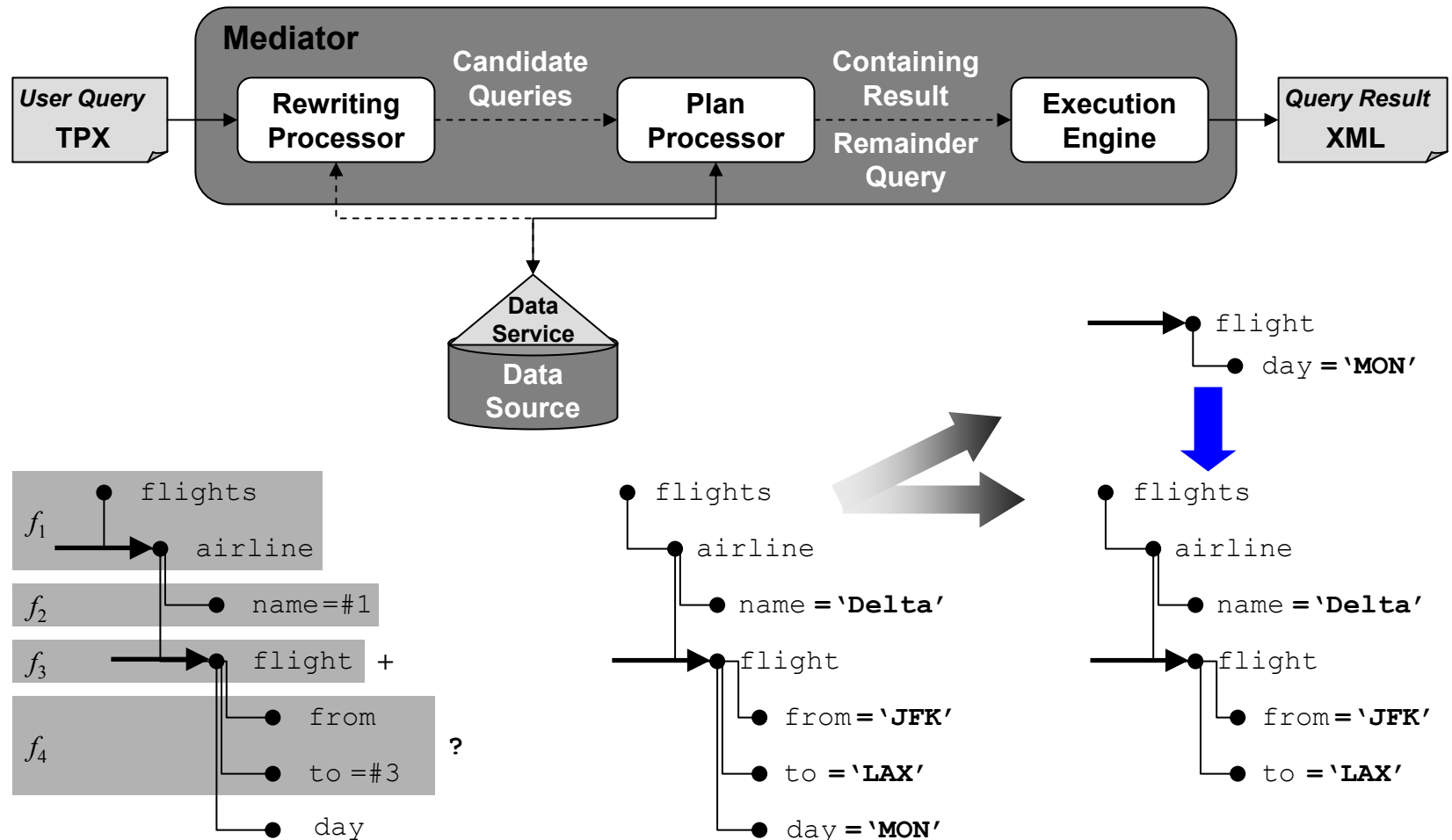
Name	Operator	Value
aircraft	=	#6

Bind Value to Element

Delete Fragment Element

# Future Work

## Capability-Based Rewriting



# Questions and Answers

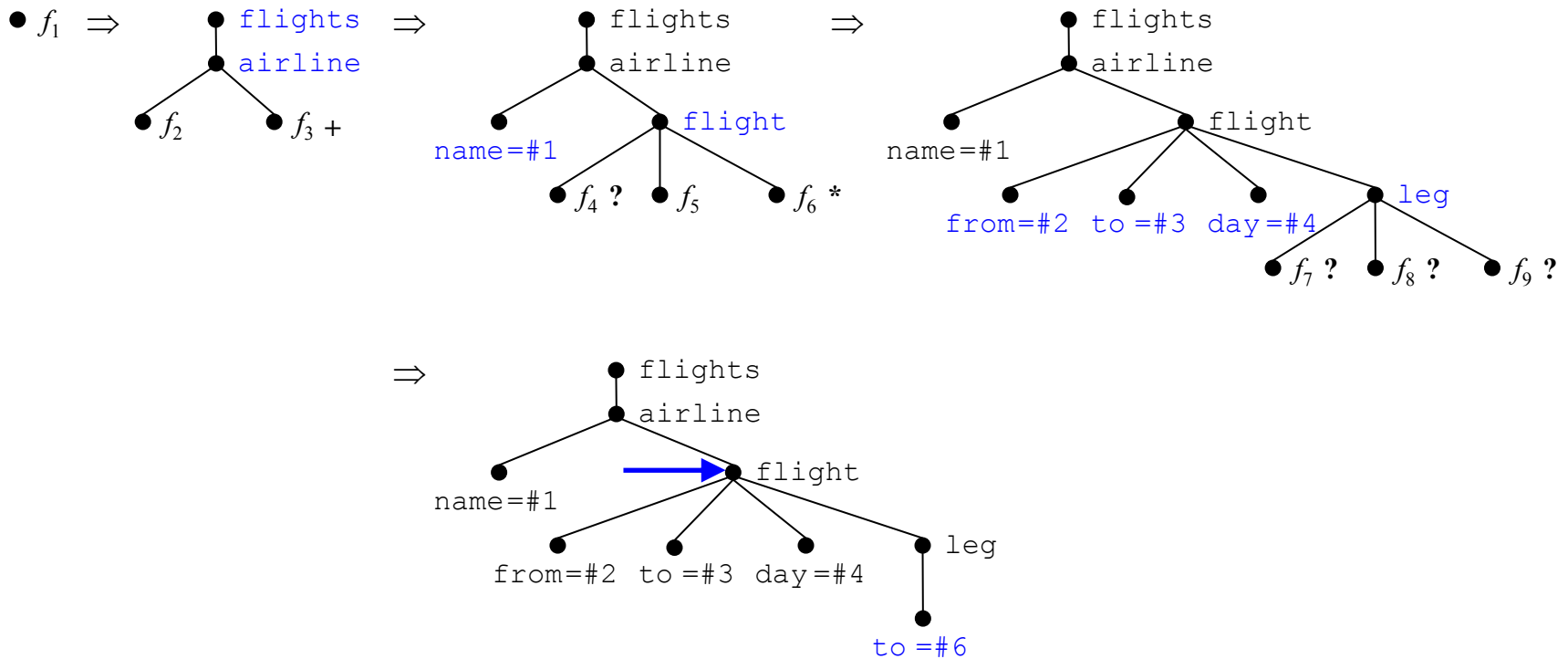
---

?



# Query Set Specification Language

## Example Derivation



# Related Work

---

## Capability-Based Rewriting

- Capabilities described as binding patterns
  - Adornments on view attributes
  - Negative approach
- Expansions of Datalog programs
  - Recursive programs → Infinite queries
  - Positive approach