

Data integration is harder than you thought

Maurizio Lenzerini

Dipartimento di Informatica e Sistemistica

Università di Roma “La Sapienza”

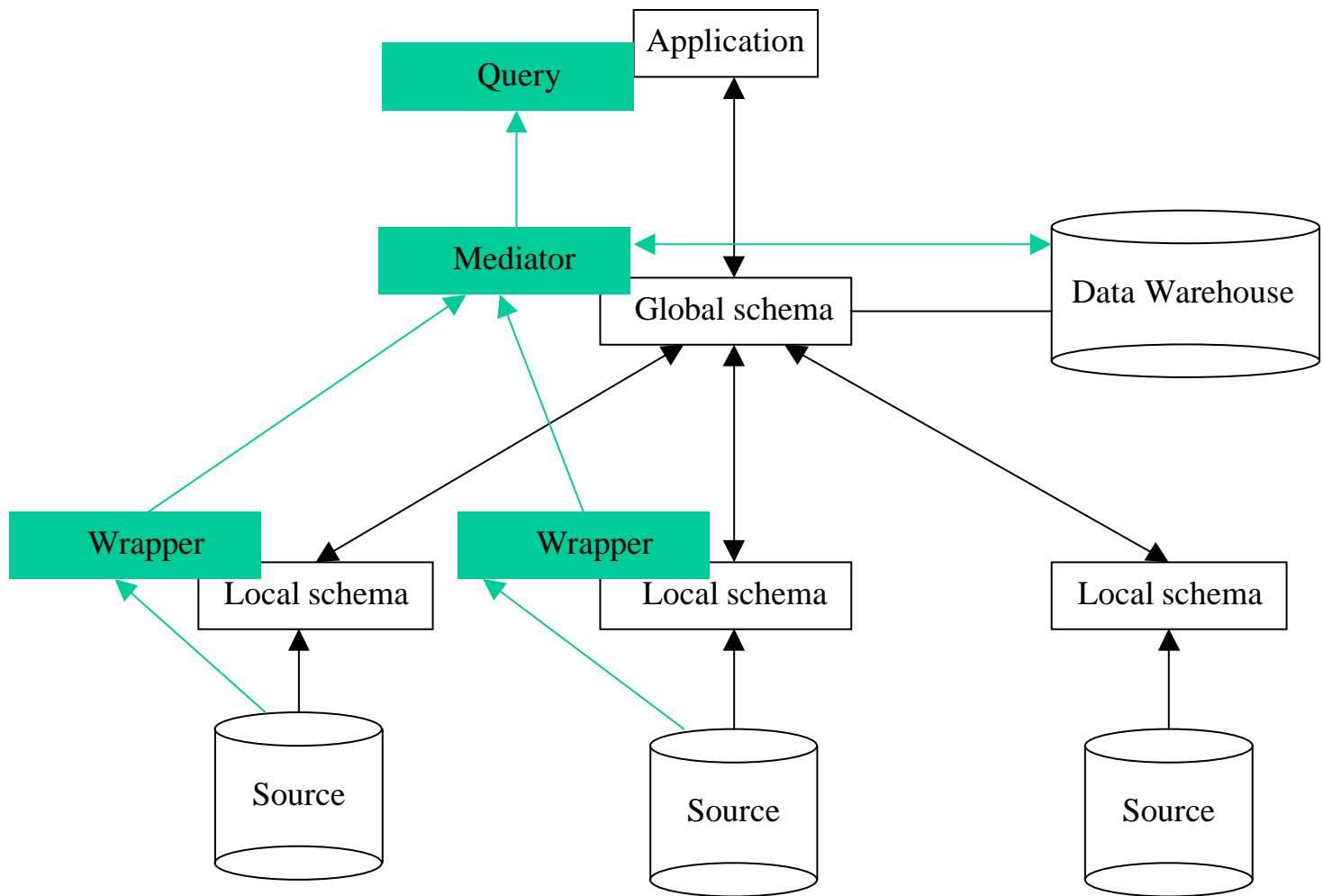
CoopIS 2001

September 5, 2001 — Trento, Italy

Outline

- Introduction to data integration
- Approaches to modeling and querying
- Case study in LAV: **hard**
- Case study in GAV: **harder than you thought**
- Beyond LAV and GAV: **even harder**
- Conclusions

Architecture for data integration



Main problems in data integration

1. Heterogeneity of sources (intensional and extensional level)
2. Limitations in the mechanisms for accessing the sources
3. Materialized vs virtual integration
4. Data extraction, cleaning and reconciliation
5. How to process updates expressed on the global schema, and updates expressed on the sources
6. **The querying problem:** How to answer queries expressed on the global schema
7. **The modeling problem:** How to model the global schema, the sources, and the relationships between the two

The querying problem

- Each query is expressed in terms of the global schema, and the associated mediator must **reformulate** the query in terms of a set of queries at the sources
- The crucial step is deciding the **query plan**, i.e., how to decompose the query into a set of subqueries to the sources
- The computed subqueries are then shipped to the sources, and the results are **assembled** into the final answer

Quality in query answering

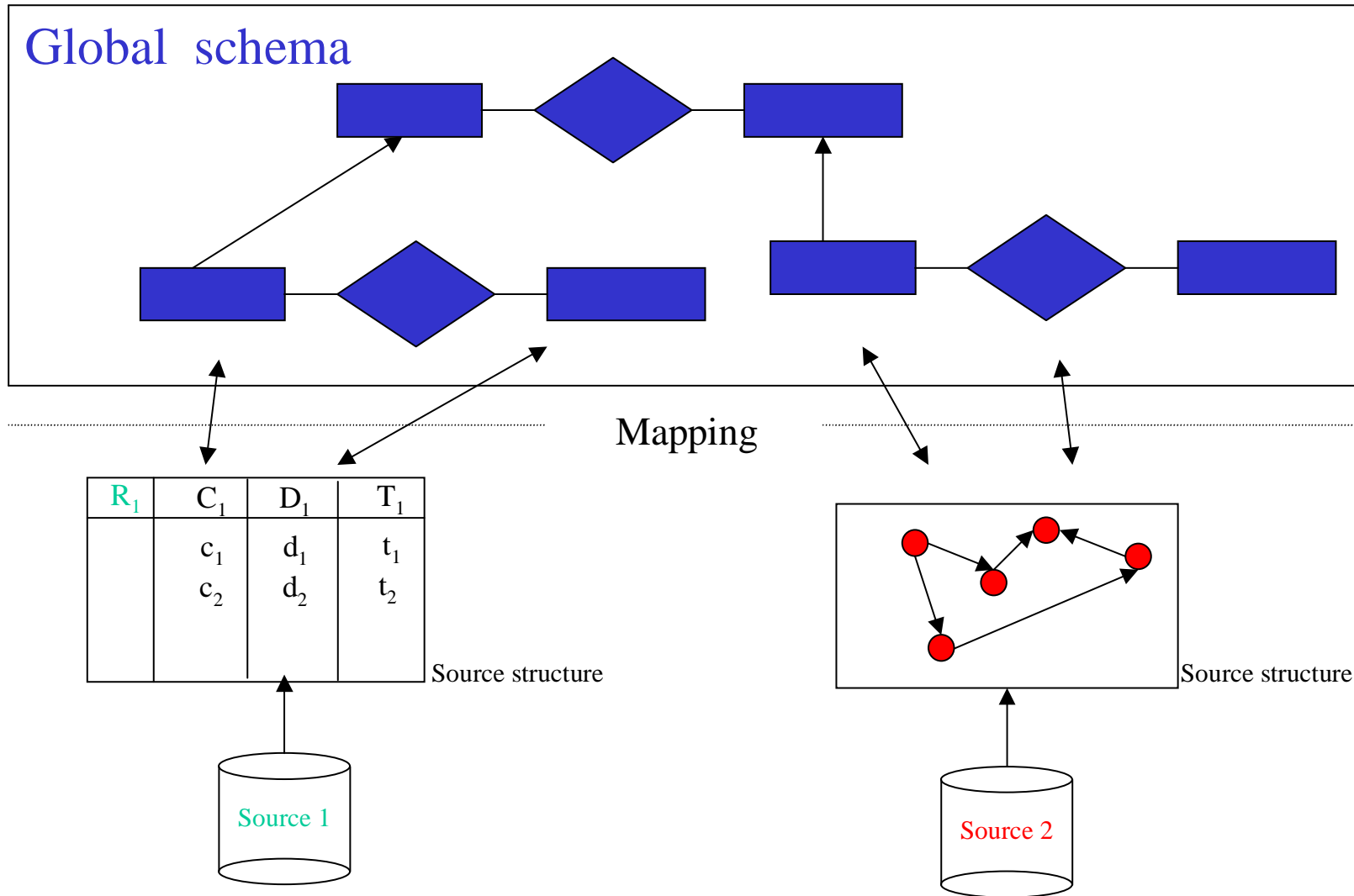
The data integration system should be designed in such a way that suitable **quality criteria** are met.

Here, we concentrate on:

- **Soundness**: the answer to queries includes **nothing but the truth**
- **Completeness**: the answer to queries includes **the whole truth**

We aim at the whole truth, and nothing but the truth. But, what the truth is depends on the approach adopted for **modeling**.

The modeling problem



Outline

- Introduction to data integration
- **Approaches to modeling and querying**
- Case study in LAV
- Case study in GAV
- Beyond LAV and GAV
- Conclusions

The modeling problem: fundamental questions

- How do we model the global schema (structured vs semistructured)
- How do we model the sources (conceptual and structural level)
- How do we model the relationship between the global schema and the sources
 - Are the sources defined in terms of the global schema (this approach is called **source-centric**, or **local-as-view**, or **LAV**)?
 - Is the global schema defined in terms of the sources (this approach is called **global-schema-centric**, or **global-as-view**, or **GAV**)?
 - A mixed approach ?

The modeling problem: formal framework

A data integration system \mathcal{D} is a triple $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where

- \mathcal{G} is the global schema (structure and constraints),
- \mathcal{S} is the source schema (structures and constraints), and
- \mathcal{M} is the mapping between \mathcal{G} and \mathcal{S} .

Semantics of \mathcal{D} : which data satisfy \mathcal{G} ?

We have to start with a source database \mathcal{C} (source data coherent with \mathcal{S}):

$$sem^{\mathcal{C}}(\mathcal{D}) = \{ \mathcal{B} \mid \mathcal{B} \text{ is a database that is legal for } \mathcal{D} \text{ wrt } \mathcal{C}, \\ \text{i.e., that satisfies both } \mathcal{G} \text{ and } \mathcal{M} \text{ wrt } \mathcal{C} \}$$

A query q to \mathcal{D} is expressed over \mathcal{G} . If q has arity n , then the answer to q wrt \mathcal{D} and \mathcal{C} is

$$q^{\mathcal{D}, \mathcal{C}} = \{(c_1, \dots, c_n) \mid (c_1, \dots, c_n) \in q^{\mathcal{B}} \ \forall \mathcal{B} \in sem^{\mathcal{C}}(\mathcal{D})\}$$

Global-as-view vs local-as-view – Example

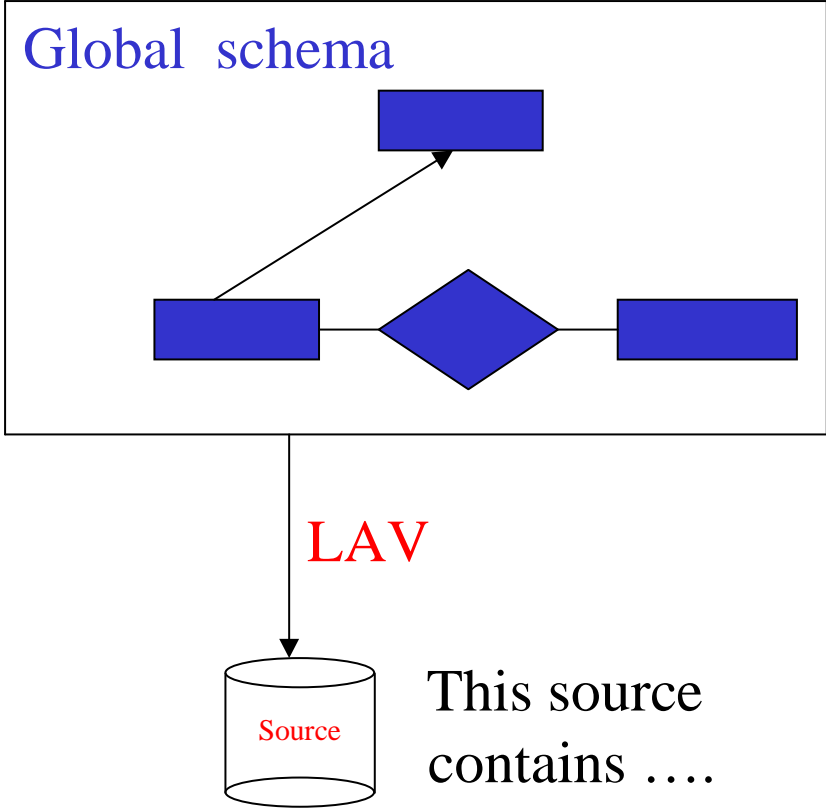
Global schema: $movie(Title, Year, Director)$
 $europaean(Director)$
 $review(Title, Critique)$

Source 1: $r_1(Title, Year, Director)$ since 1960, european directors

Source 2: $r_2(Title, Critique)$ since 1990

Query: Title and critique of movies in 1998
 $\{ (T, R) \mid \exists D. movie(T, 1998, D) \wedge review(T, R) \}$, written
 $\{ (T, R) \mid movie(T, 1998, D) \wedge review(T, R) \}$

Local-as-view



Formalization of LAV

In LAV, the mapping \mathcal{M} is constituted by a set of assertions:

$$s \rightsquigarrow \phi_{\mathcal{G}}$$

one for each source structure s in \mathcal{S} , where $\phi_{\mathcal{G}}$ is a **query** over \mathcal{G} . Given source data \mathcal{C} , a database \mathcal{B} satisfies \mathcal{M} wrt \mathcal{C} if for each source $s \in \mathcal{S}$:

$$s^{\mathcal{C}} \subseteq \phi_{\mathcal{G}}^{\mathcal{B}}$$

The mapping \mathcal{M} does **not** provide direct information about which data satisfies the global schema.

To answer a query q over \mathcal{G} , we have to **infer** how to use \mathcal{M} in order to access the source data \mathcal{C} .

Answering queries is an inference process, which is similar to answering queries with **incomplete information**.

Local-as-view – Example

Global schema: $\text{movie}(Title, Year, Director)$
 $\text{european}(Director)$
 $\text{review}(Title, Critique)$

Local-as-view: associated to relations at the sources we have **views** over the global schema

$$\begin{aligned} r_1(T, Y, D) &\rightsquigarrow \{ (T, Y, D) \mid \text{movie}(T, Y, D) \wedge \text{european}(D) \wedge Y \geq 1960 \} \\ r_2(T, R) &\rightsquigarrow \{ (T, R) \mid \text{movie}(T, Y, D) \wedge \text{review}(T, R) \wedge Y \geq 1990 \} \end{aligned}$$

The query $\{ (T, R) \mid \text{movie}(T, 1998, D) \wedge \text{review}(T, R) \}$ is processed by means of an inference mechanism that aims at re-expressing the atoms of the global schema in terms of atoms at the sources. In this case:

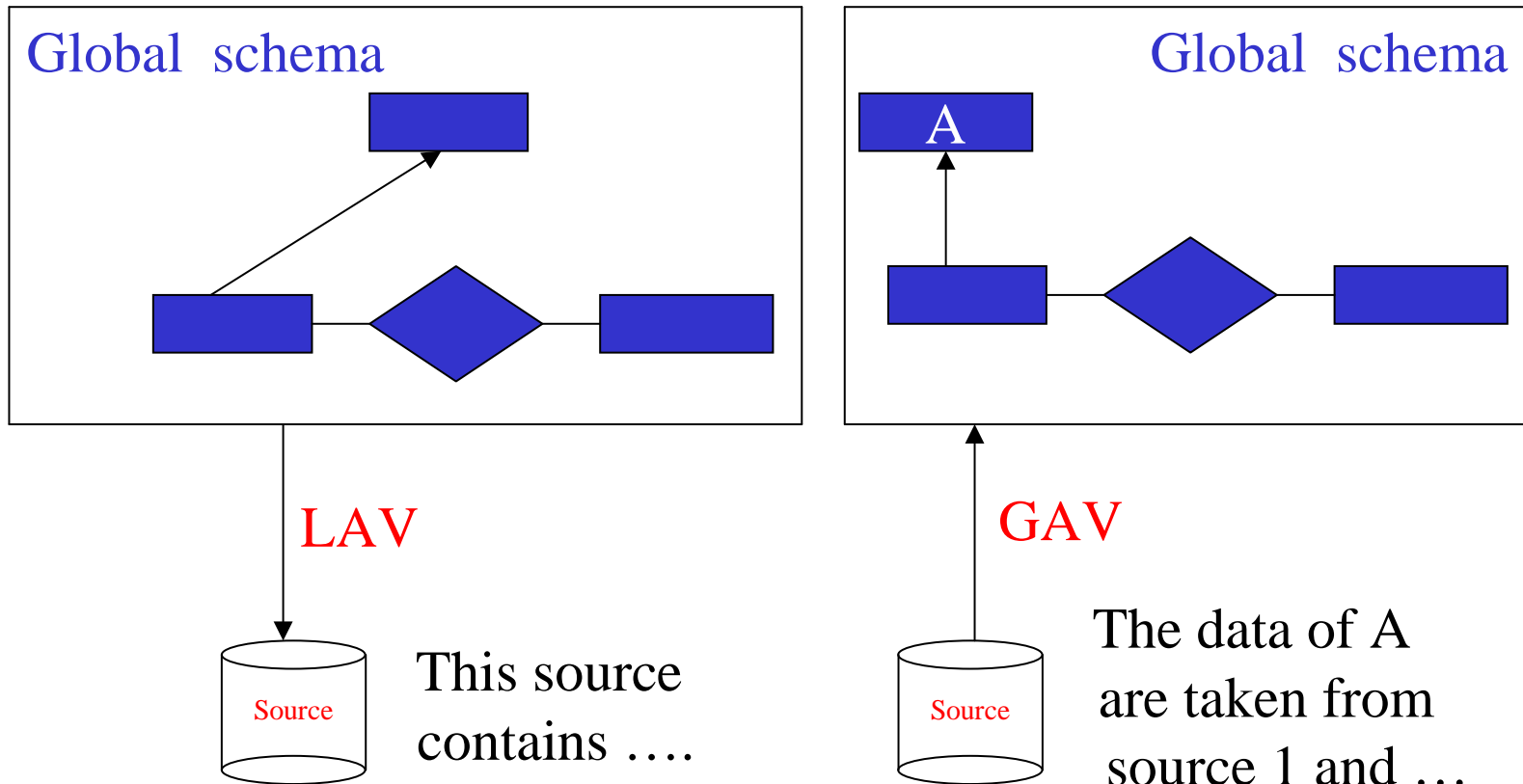
$$\{ (T, R) \mid r_2(T, R) \wedge r_1(T, 1998, D) \}$$

Query processing in LAV

Answering queries in LAV is like solving a mystery case:

- Sources represent reliable witnesses
- Witnesses know part of the story, and source data represent what they know
- We have an explicit representation of what the witnesses know
- We have to solve the case (answering queries) based on the information we are able to gather from the witnesses
- **Inference** is needed

Global-as-view



Formalization of GAV

In GAV, the mapping \mathcal{M} is constituted by a set of assertions:

$$g \rightsquigarrow \phi_{\mathcal{S}}$$

one for each structure g in \mathcal{G} , where $\phi_{\mathcal{S}}$ is a query over \mathcal{S} . Given source data \mathcal{C} , a database \mathcal{B} satisfies \mathcal{M} wrt \mathcal{C} if for each $g \in \mathcal{G}$:

$$\phi_{\mathcal{S}}^{\mathcal{C}} \subseteq g^{\mathcal{B}}$$

The mapping \mathcal{M} provides direct information about which data satisfies the global schema. Thus, given a query q over \mathcal{G} , it seems that we can simply evaluate the query over these data (as if we had a single database at hand).

More on this later....

Global-as-view – Example

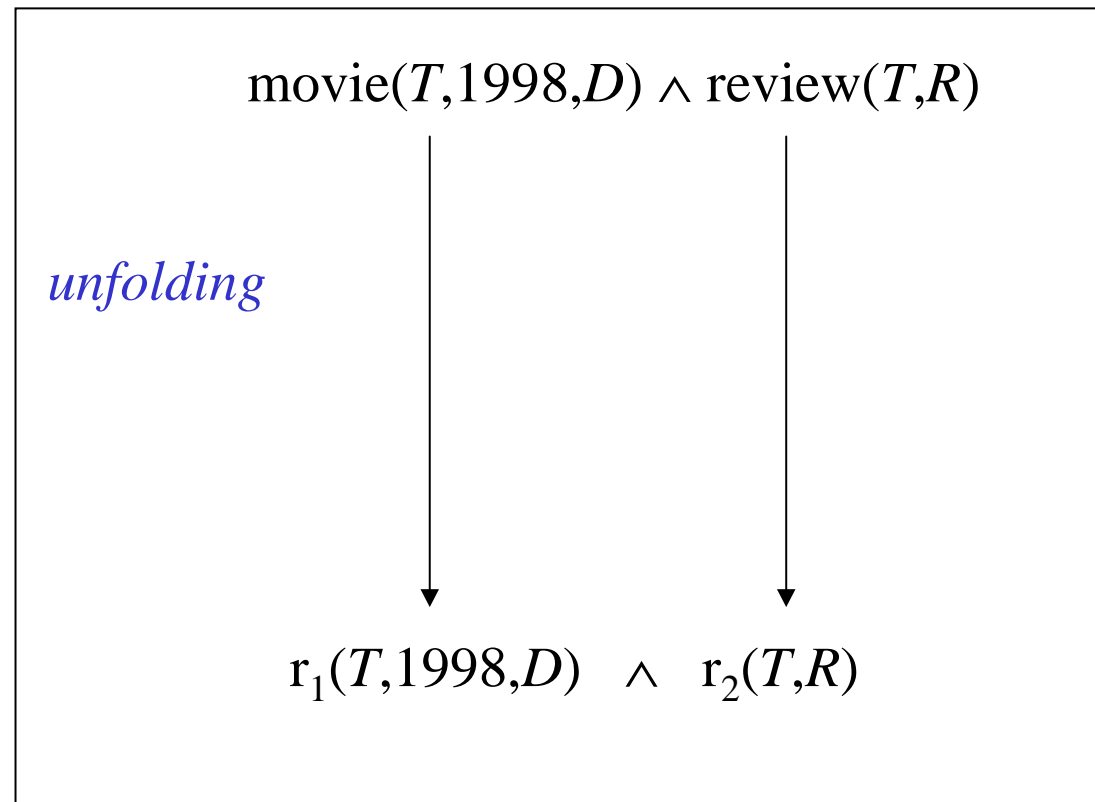
Global schema: $\text{movie}(Title, Year, Director)$
 $\text{european}(Director)$
 $\text{review}(Title, Critique)$

Global-as-view: associated to relations in the global schema we have **views** over the sources

$$\begin{aligned} \text{movie}(T, Y, D) &\rightsquigarrow \{ (T, Y, D) \mid r_1(T, Y, D) \} \\ \text{european}(D) &\rightsquigarrow \{ (D) \mid r_1(T, Y, D) \} \\ \text{review}(T, R) &\rightsquigarrow \{ (T, R) \mid r_2(T, R) \} \end{aligned}$$

Global-as-view – Example of query processing

The query $\{ (T, R) \mid \text{movie}(T, 1998, D) \wedge \text{review}(T, R) \}$ is processed by means of unfolding, i.e., by expanding the atoms according to their definitions, so as to come up with source relations. In this case:



Query processing in GAV

- We do not have any explicit representation of what the witnesses know
- All the information that the witnesses can provide have been compiled into an “investigation report” (the global schema, and the mapping)
- Solving the case (answering queries) means basically **looking at** the investigation report

Global-as-view and local-as-view – Comparison

Local-as-view: (Information Manifold, DWQ, Pictel)

- Quality depends on how well we have characterized the sources
- High modularity and reusability (if the global schema is well designed, when a source changes, only its definition is affected)
- Query processing needs reasoning (query reformulation complex)

Global-as-view: (Carnot, SIMS, Tsimmis, ...)

- Quality depends on how well we have compiled the sources into the global schema through the mapping
- Whenever a source changes or a new one is added, the global schema needs to be reconsidered
- Query processing can be based on some sort of unfolding (query reformulation looks easier)

For more details, see [Ullman, TCS 2000], [Halevy, SIGMOD 2000].

Outline

- Introduction to data integration
- Approaches to modeling and querying
- **Case study in LAV**
- Case study in GAV
- Beyond LAV and GAV
- Conclusions

A case study in LAV

We deal with the problem of answering queries to data integration systems of the form $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where

- the global schema \mathcal{G} is semi-structured
- the sources in \mathcal{S} are relational
- the mapping \mathcal{M} is of type LAV
- queries are typical of semi-structured data

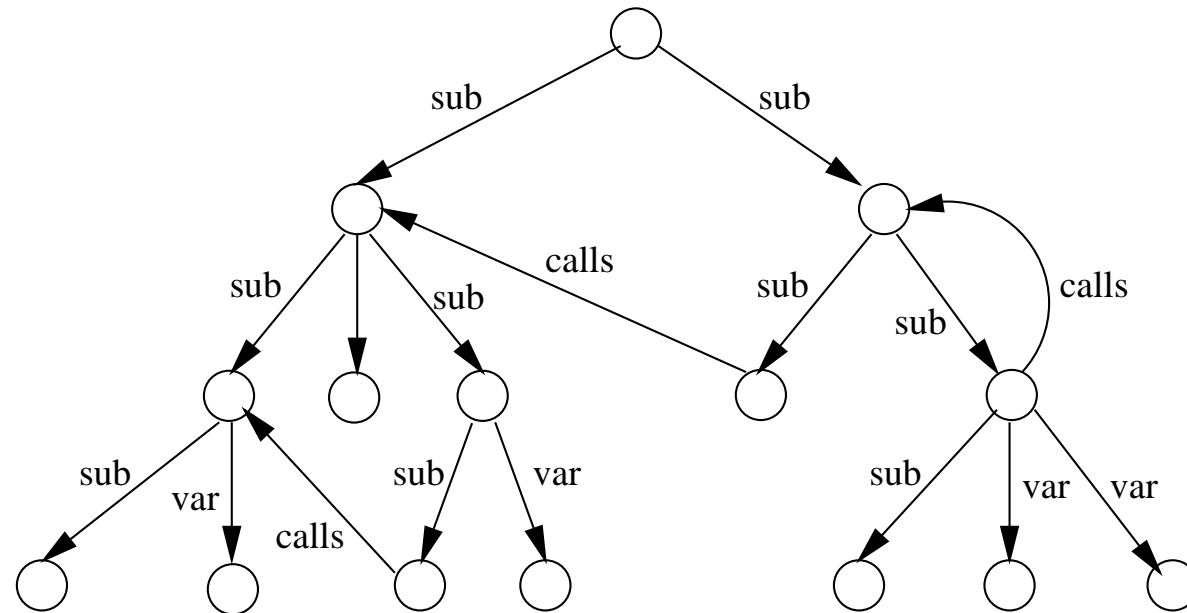
The query answering problem

Given data integration system $\mathcal{D} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, source database \mathcal{C} , query q , and tuple t , check whether $t \in q^{\mathcal{D}, \mathcal{C}}$ (i.e., whether $t \in q^{\mathcal{B}}$ for all $\mathcal{B} \in \text{sem}^{\mathcal{C}}(\mathcal{D})$).

Recent results:

- Complexity for several query and view languages [Abiteboul et al, PODS 98], [Grahne et al, ICDT 99]
- Schemas expressed in Description Logics [Calvanese et al, AAAI 2000]
- **Regular path queries** without inverse [Calvanese et al, ICDE 2000] and **with inverse** [Calvanese et al, PODS 2000]
- Conjunctive RPQIs [Calvanese et al, KR 2000], [Calvanese et al, LICS 2000], [Calvanese et al, DBPL 2001]

Global databases and queries



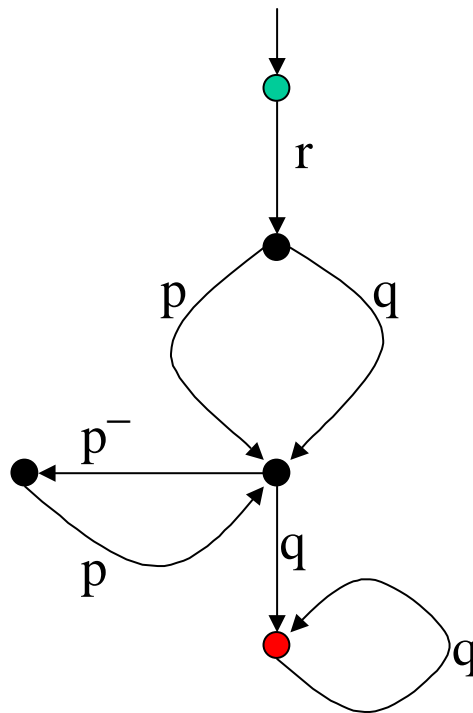
RPQ: $(sub)^* \cdot (sub \cdot (calls \cup sub))^* \cdot var$

RPQI: $(sub^-)^* \cdot (var \cup sub)$

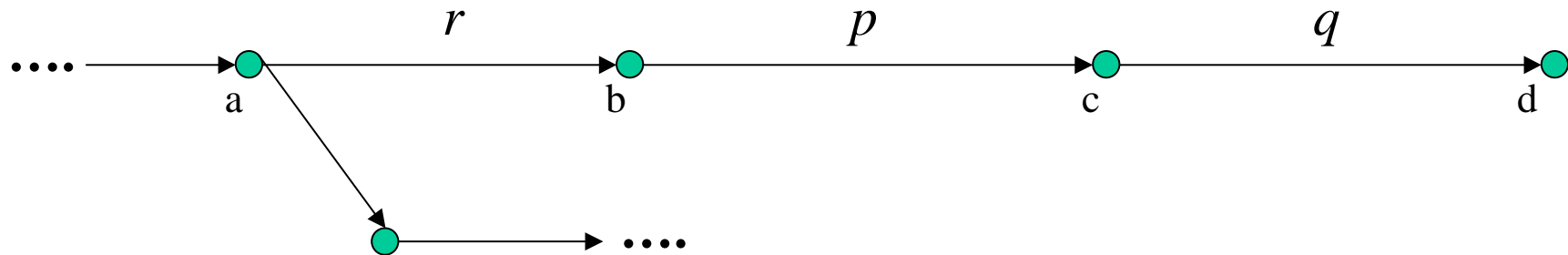
Regular path queries with inverse

Regular-path queries with inverse (RPQIs) are expressed by means of finite-state automata over $\Sigma = \Sigma' \cup \{p^- \mid p \in \Sigma'\}$ (p^- denotes the inverse of the binary relation p).

$$r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$



Finite state automata and RPQIs



Consider the query $Q = r \cdot (p \cup q) \cdot p^- \cdot p \cdot q \cdot q^*$

Automaton for Q $\left\{ \begin{array}{l} s_1 \in \delta(s_0, r), s_2 \in \delta(s_1, p), s_2 \in \delta(s_1, q), \\ s_3 \in \delta(s_2, p^-), s_4 \in \delta(s_3, p), s_5 \in \delta(s_4, q), s_5 \in \delta(s_5, q) \end{array} \right.$

The computation for RPQIs is not completely captured by finite state automata.

Two-way automata

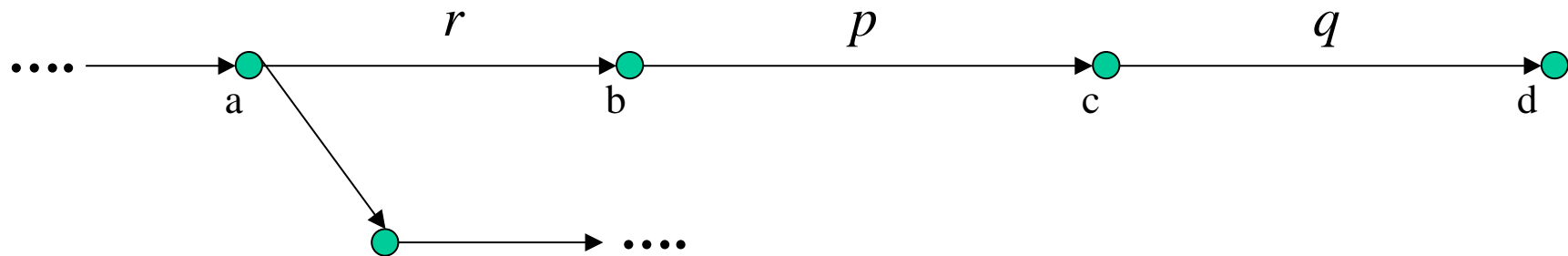
A *two-way automaton* $A = (\Gamma, S, S_0, \rho, F)$ consists of an alphabet Γ , a finite set of states S , a set of initial states $S_0 \subseteq S$, a transition function

$$\rho : S \times \Sigma \rightarrow 2^S \times \{-1, 0, 1\}$$

and a set of accepting states $F \subseteq S$.

Given a two-way automaton A with n states, one can construct a one-way automaton B_1 with $O(2^{n \log n})$ states such that $L(B_1) = L(A)$, and a one-way automaton B_2 with $O(2^n)$ states such that $L(B_2) = \Gamma^* - L(A)$.

Two-way automata and RPQIs



Consider the query $Q = r \cdot (p \cup q) \cdot p^- \cdot p \cdot q \cdot q^*$

Automaton for Q $\left\{ \begin{array}{l} s_1 \in \delta(s_0, r), s_2 \in \delta(s_1, p), s_2 \in \delta(s_1, q), \\ s_3 \in \delta(s_2, p^-), s_4 \in \delta(s_3, p), s_5 \in \delta(s_4, q), s_5 \in \delta(s_5, q) \end{array} \right.$

2way automaton $\left\{ \begin{array}{l} (s_1, 1) \in \delta_A(s_0, r), (s_2, 1) \in \delta_A(s_1, p), \\ (s_2^{\leftarrow}, -1) \in \delta_A(s_2, q), (s_3, 0) \in \delta_A(s_2^{\leftarrow}, p), \\ (s_4, 1) \in \delta_A(s_3, p), (s_5, 1) \in \delta_A(s_4, q), (s_f, 1) \in \delta_A(s_5, \$) \end{array} \right.$

Two-way automata and RPQIs

Given an RPQI $E = (\Sigma, S, I, \delta, F)$ over the alphabet Σ , the corresponding two-way automaton A_E is:

$$(\Sigma_A = \Sigma \cup \{\$, \}, S_A = S \cup \{s_f\} \cup \{s^{\leftarrow} \mid s \in S\}, I, \delta_A, \{s_f\})$$

where δ_A is defined as follows:

- $(s_2, 1) \in \delta_A(s_1, r)$, for each transition $s_2 \in \delta(s_1, r)$ of E
- **enter backward mode:**
 $(s^{\leftarrow}, -1) \in \delta_A(s, \ell)$, for each $s \in S$ and $\ell \in \Sigma_A$
- **exit backward mode:**
 $(s_2, 0) \in \delta_A(s_1^{\leftarrow}, r)$, for each transition $s_2 \in \delta(s_1, r^-)$ of E
- $(s_f, 1) \in \delta_A(s, \$)$, for each $s \in F$.

\implies **w satisfies E iff $w\$ \in L(A_E)$.**

Query answering: basic idea

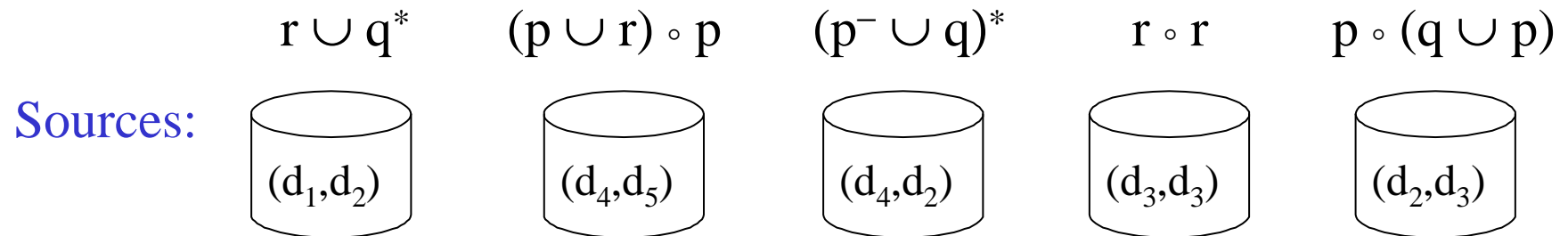
- Given $\mathcal{D} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, source database \mathcal{C} , query q , and tuple (c, d) , we search for a **counterexample** to $(c, d) \in q^{\mathcal{C}, \mathcal{D}}$, i.e., a database $\mathcal{B} \in \text{sem}^{\mathcal{C}}(\mathcal{D})$ such that $(c, d) \notin q^{\mathcal{B}}$.
- Each counterexample DB \mathcal{B} can be represented by a word $w_{\mathcal{B}}$ over the alphabet $\Sigma_A = \Sigma \cup \mathcal{C} \cup \{\$\}$, which has the form

$$\$ d_1 w_1 d_2 \$ d_3 w_2 d_4 \$ \cdots \$ d_{2m-1} w_m d_{2m} \$$$

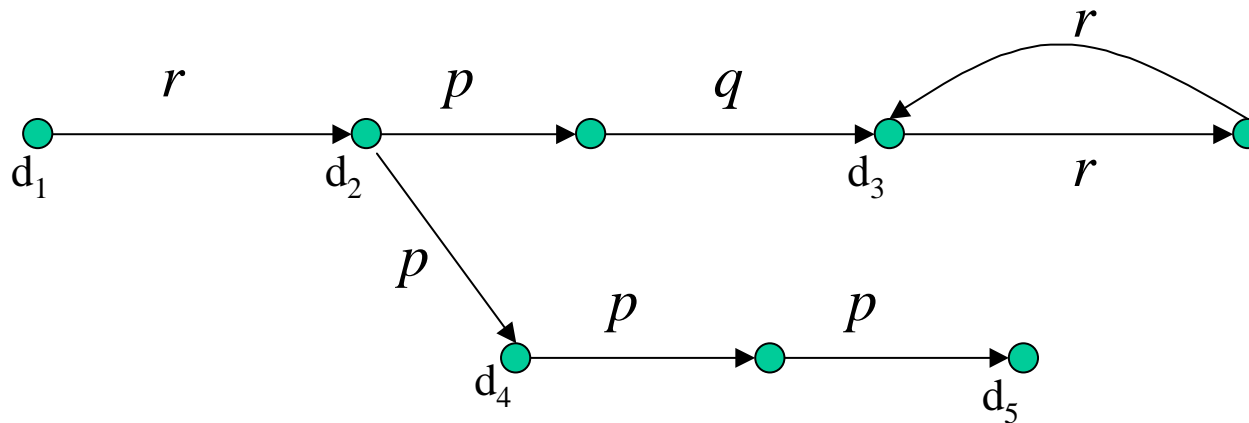
where d_1, \dots, d_{2m} range over data objects in \mathcal{C} (simply denoted by \mathcal{C}), $w_i \in \Sigma^+$, and the $\$$ acts as a separator.

Two-way automata and canonical DBs

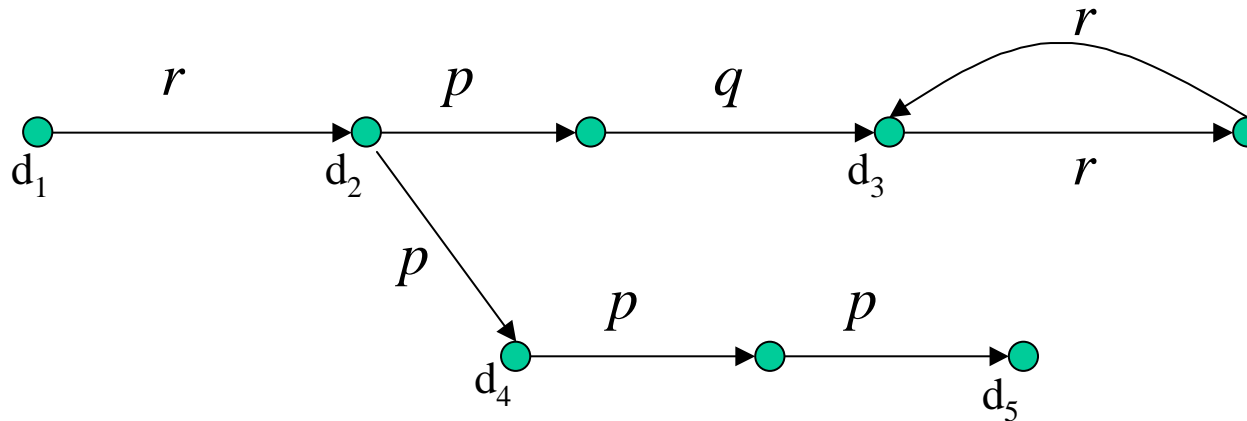
Global schema G : $(r \cup p \cup q \cup r^- \cup p^- \cup q^-)^*$



Database for G :



Two-way automata and canonical DBs



As a word: $\$d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

The above database \mathcal{B} is a counterexample to $(d_2, d_3) \in Q^{\mathcal{D}, \mathcal{C}}$. To verify that $(d_2, d_3) \notin Q^{\mathcal{B}}$, we exploit not only the ability of two-way automata to move on the word both **forward and backward**, but also the ability to **jump** from one position in the word representing a node to any other position (either preceding or succeeding) representing the same node.

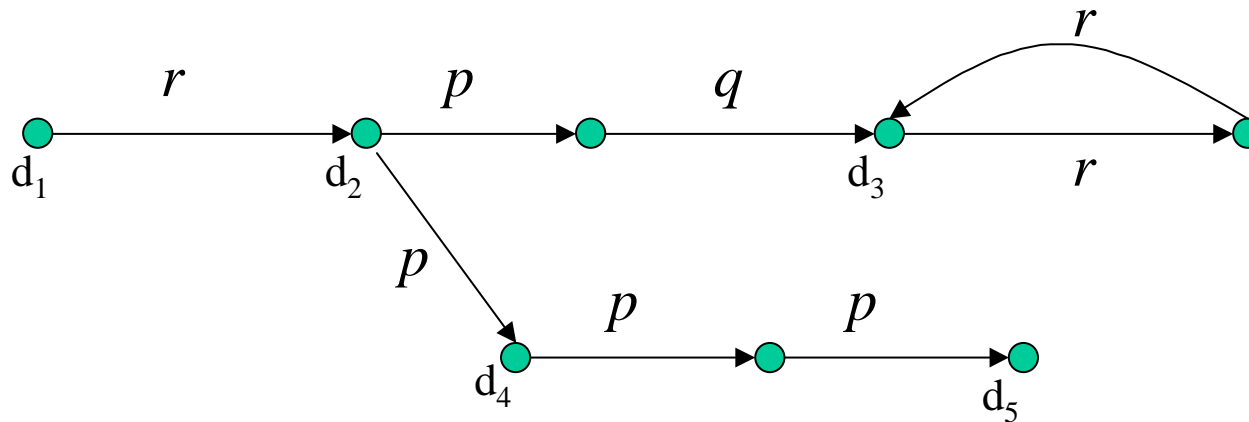
Query answering: Basic idea

If $Q = (\Sigma, S, I, \delta, F)$, then $A_{(Q,a,b)} = (\Sigma_A, S_A, \{s_0\}, \delta_A, \{s_f\})$, where $S_A = S \cup \{s_0, s_f\} \cup \{s^\leftarrow \mid s \in S\} \cup (S \times \mathcal{D})$, and

1. $(s^\leftarrow, -1) \in \delta_A(s, \ell)$, for each $s \in S$ and $\ell \in \Sigma \cup \mathcal{C}$
2. $(s_2, 1) \in \delta_A(s_1, r)$, for each $s_2 \in \delta(s_1, r)$
3. $(s_2, 0) \in \delta_A(s_1^\leftarrow, r)$, for each $s_2 \in \delta(s_1, r^-)$
4. $((s, d), 0) \in \delta_A(s, d)$, $((s, d), 0) \in \delta_A(s^\leftarrow, d)$
 $((s, d), 1) \in \delta_A((s, d), \ell)$, $((s, d), -1) \in \delta_A((s, d), \ell)$
 $(s, 0) \in \delta_A((s, d), d)$, $(s, 1) \in \delta_A(s, d)$
5. $(s_0, 1) \in \delta_A(s_0, \ell)$, for each $\ell \in \Sigma_A$, $(s, 0) \in \delta_A(s_0, a)$ for each $s \in I$
6. $(s_f, 0) \in \delta_A(s, b)$, for each $s \in F$, and $(s_f, 1) \in \delta_A(s_f, \ell)$ for each $\ell \in \Sigma_A$.

$A_{(Q,a,b)}$ **accepts** w_B **iff** $(a, b) \in Q^B$.

A run of $A_{(Q,d_1,d_3)}$



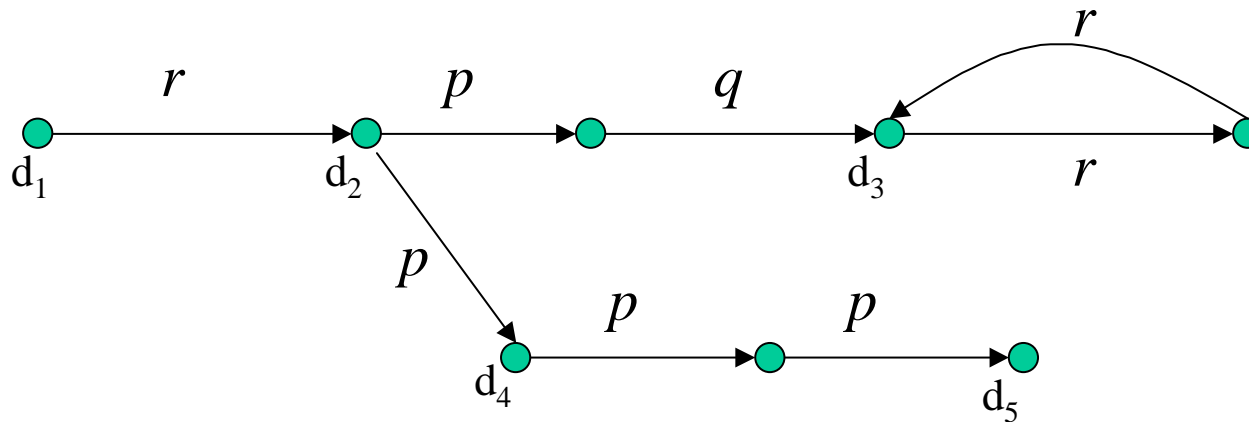
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_0

Transition: $(s_0, 1) \in \delta_A(s_0, \ell)$, for each $\ell \in \Sigma_A$

A run of $A_{(Q,d_1,d_3)}$



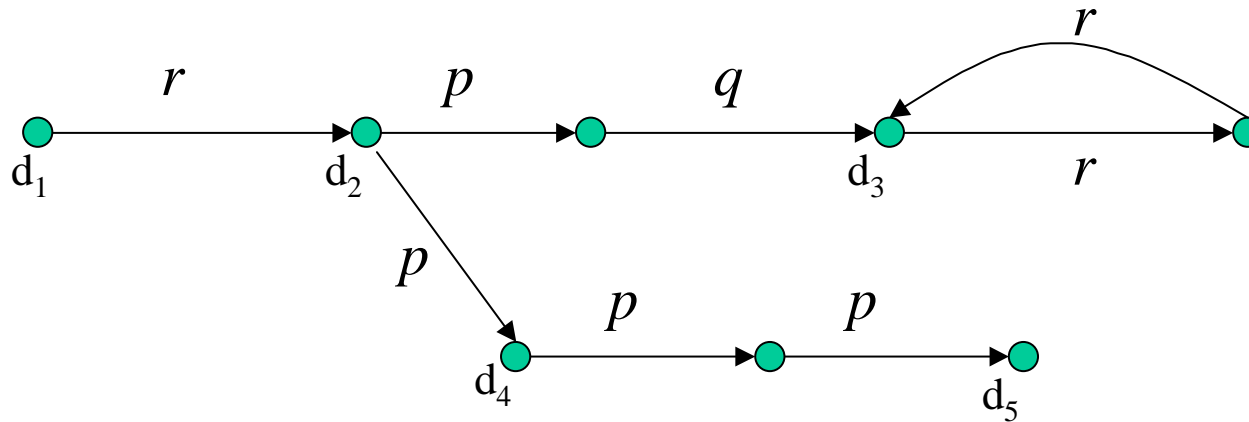
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_0

Transition: $(s_0, 1) \in \delta_A(s_0, \ell)$, for each $\ell \in \Sigma_A$

A run of $A_{(Q,d_1,d_3)}$



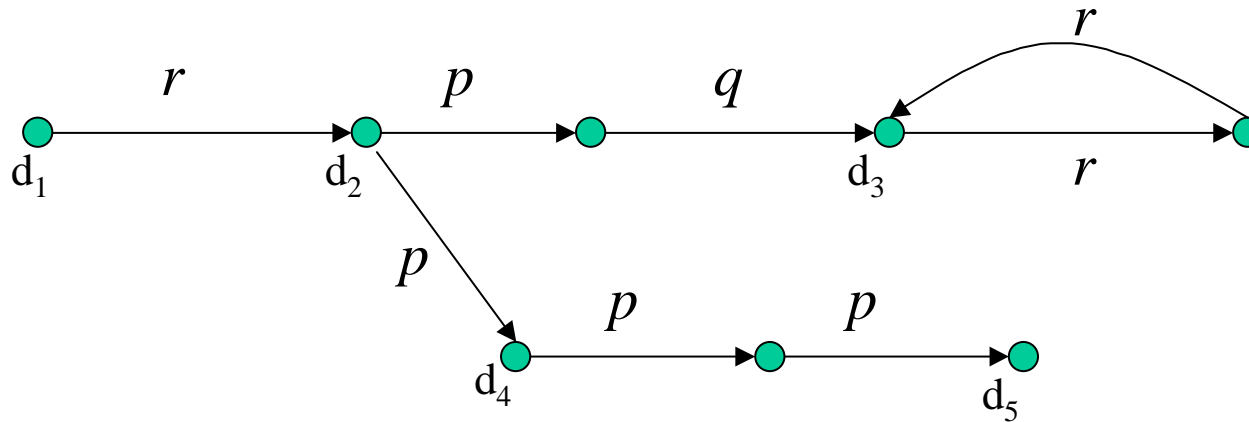
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_0

Transition: $(s_0, 1) \in \delta_A(s_0, \ell)$, for each $\ell \in \Sigma_A$

A run of $A_{(Q,d_1,d_3)}$



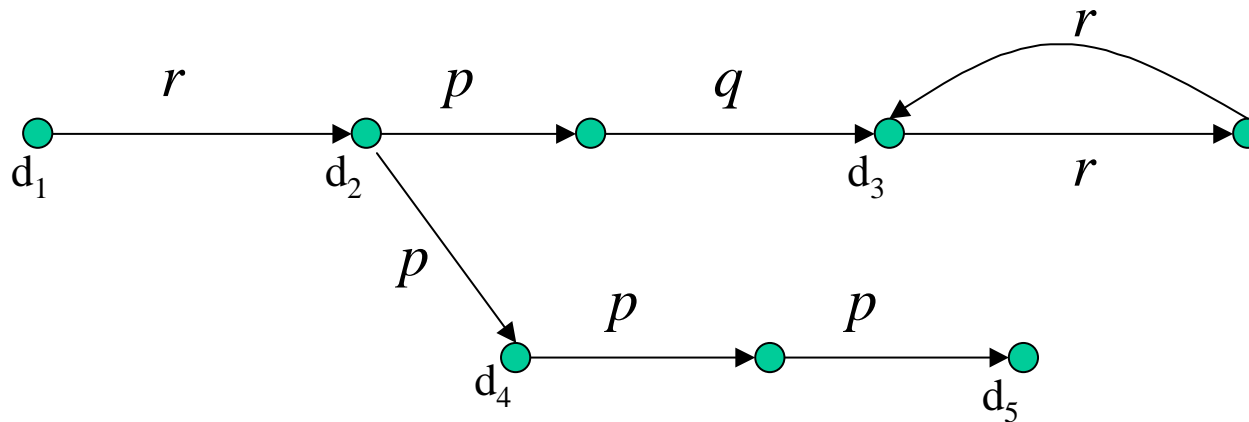
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_0

Transition: $(s_0, 1) \in \delta_A(s_0, \ell)$, for each $\ell \in \Sigma_A$

A run of $A_{(Q,d_1,d_3)}$



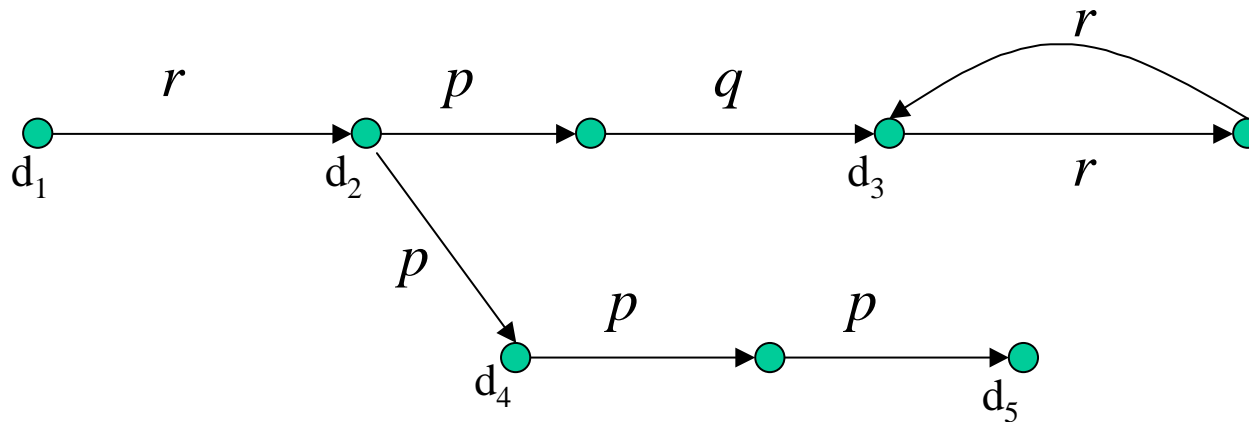
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_0

Transition: $(s_0, 1) \in \delta_A(s_0, \ell)$, for each $\ell \in \Sigma_A$

A run of $A_{(Q,d_1,d_3)}$



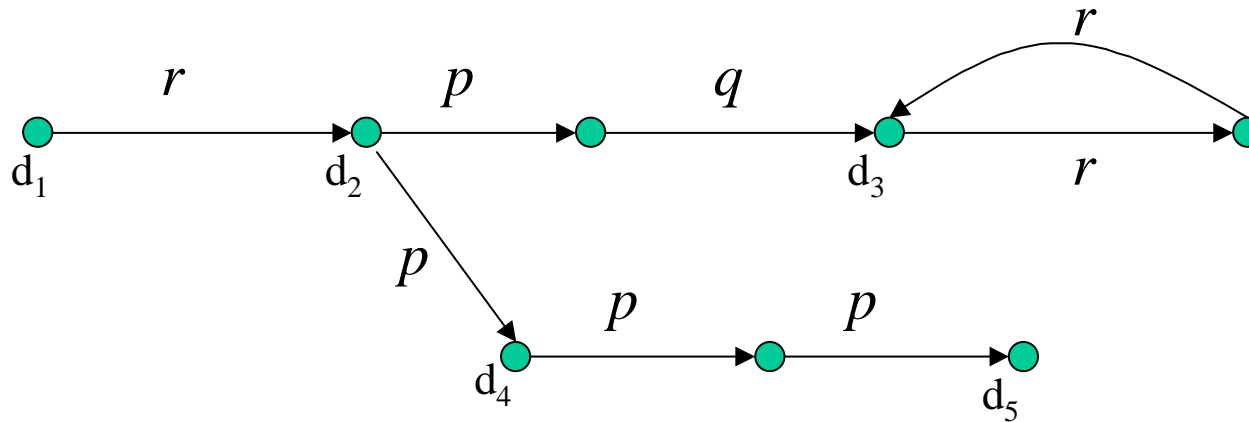
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_0

Transition: $(s_0, 1) \in \delta_A(s_0, \ell)$, for each $\ell \in \Sigma_A$

A run of $A_{(Q,d_1,d_3)}$



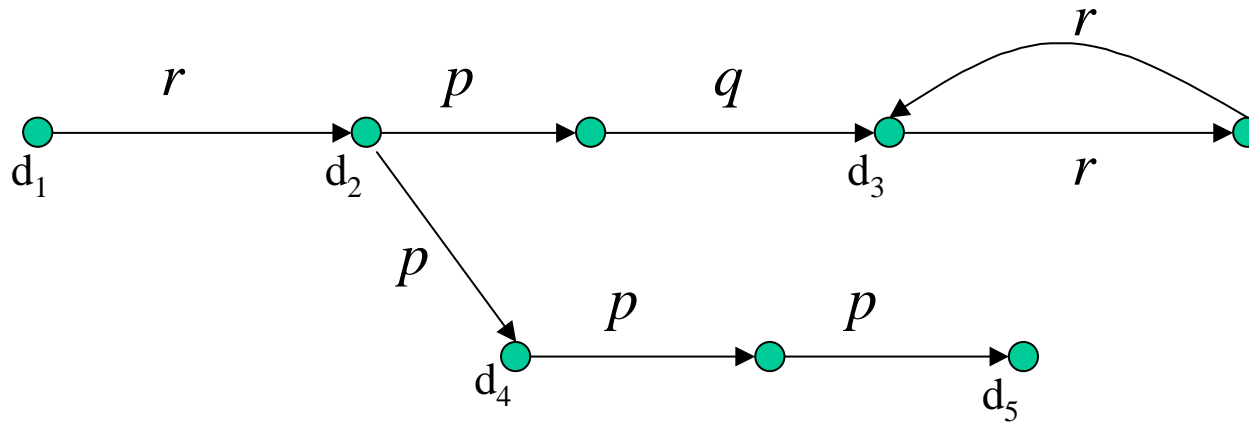
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_0

Transition: $(s_1, 0) \in \delta_A(s_0, d_1)$, s_1 initial state for Q

A run of $A_{(Q,d_1,d_3)}$



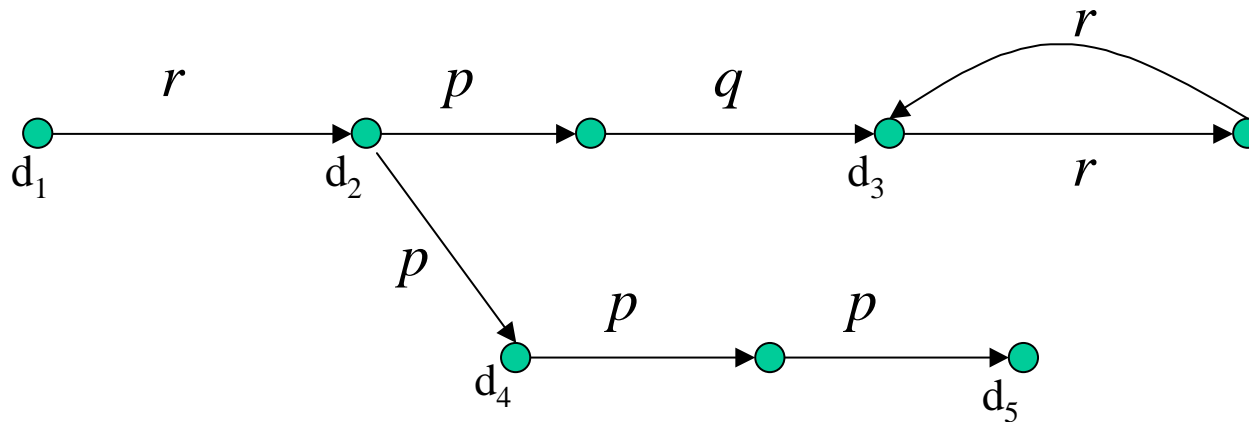
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_1

Transition: $(s_1, 1) \in \delta_A(s_1, d_1)$

A run of $A_{(Q,d_1,d_3)}$



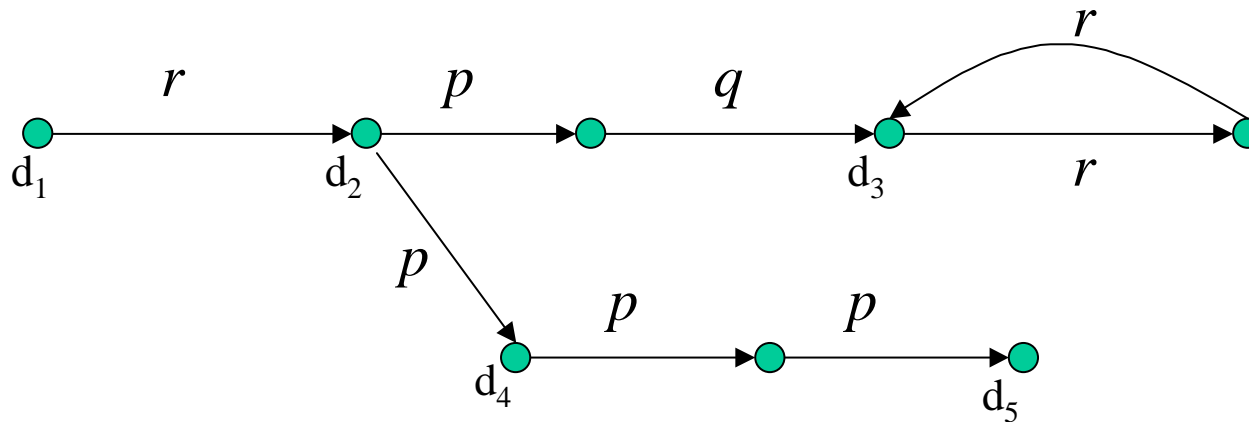
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_1

Transition: $(s_2, 1) \in \delta_A(s_1, r)$, transition coming from Q

A run of $A_{(Q,d_1,d_3)}$



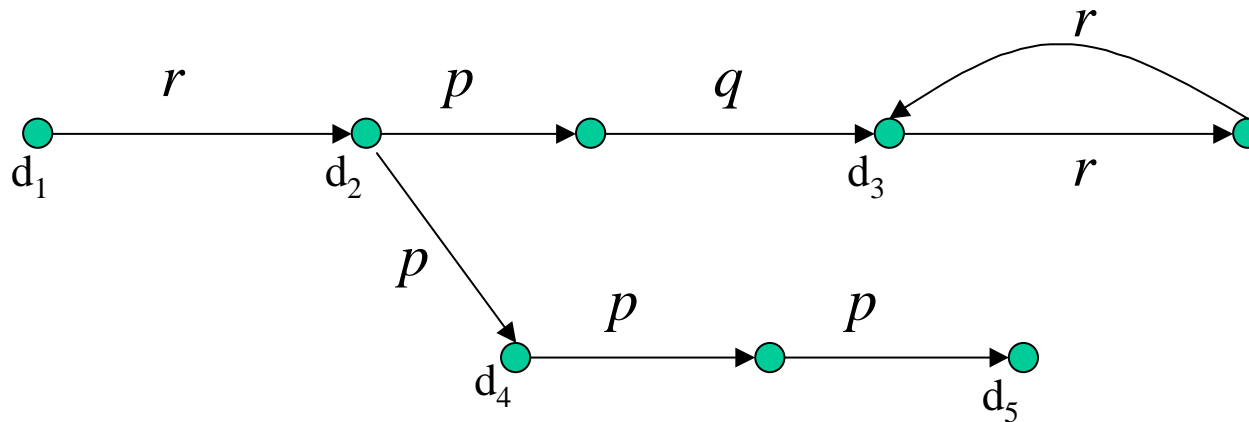
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_2

Transition: $((s_2, d_2), 1) \in \delta_A(s_2, d_2)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



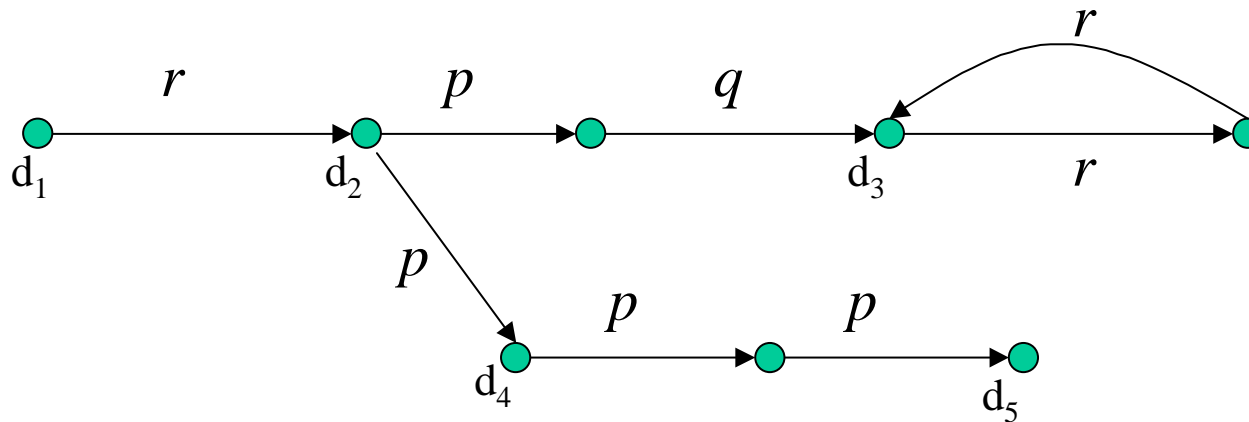
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_2, d_2)

Transition: $((s_2, d_2), 1) \in \delta_A((s_2, d_2), \$)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



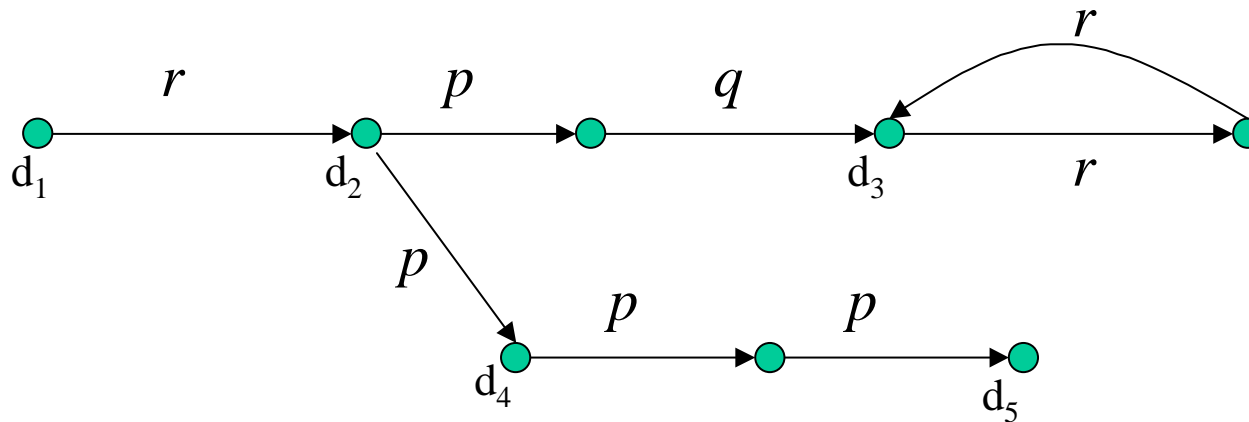
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_2, d_2)

Transition: $((s_2, d_2), 1) \in \delta_A((s_2, d_2), d_4)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



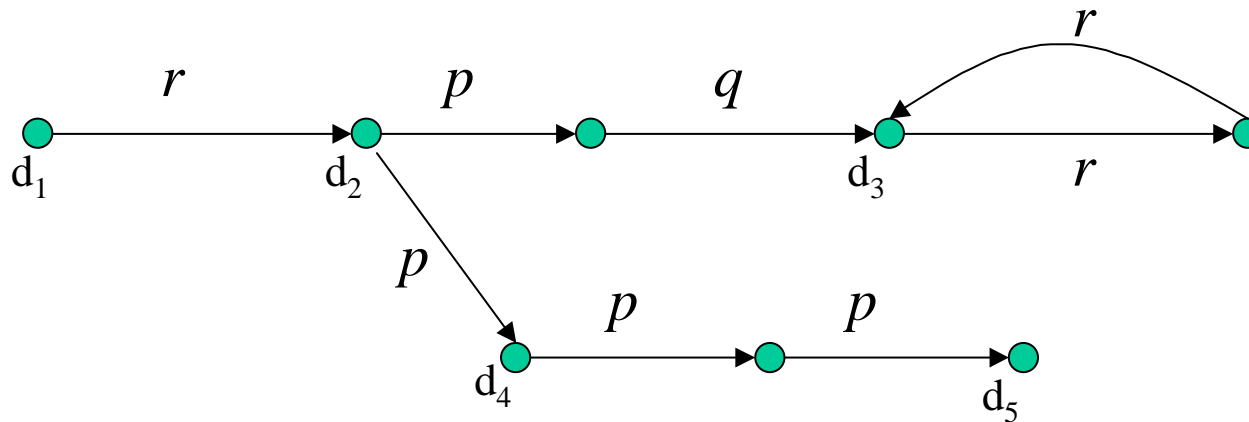
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_2, d_2)

Transition: $((s_2, d_2), 1) \in \delta_A((s_2, d_2), p^-)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



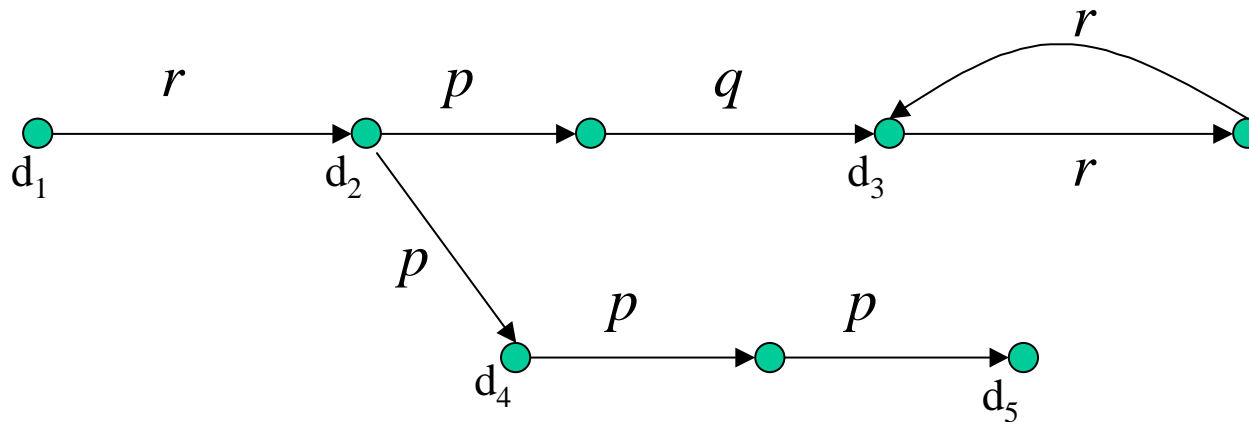
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_2, d_2)

Transition: $(s_2, 0) \in \delta_A((s_2, d_2), d_2)$, exit search mode

A run of $A_{(Q,d_1,d_3)}$



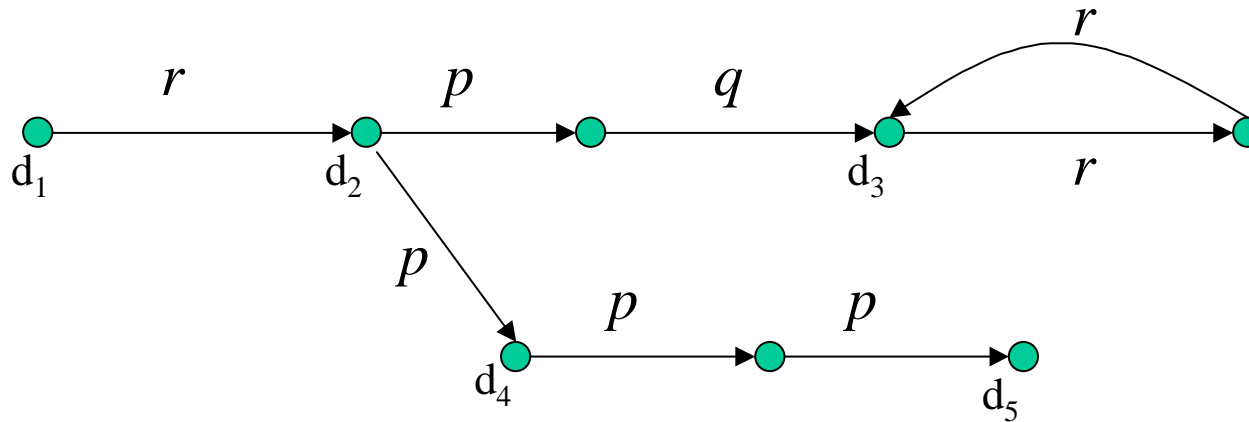
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_2

Transition: $(s_2^{\leftarrow}, -1) \in \delta_A(s_2, d_2)$, backward mode

A run of $A_{(Q,d_1,d_3)}$



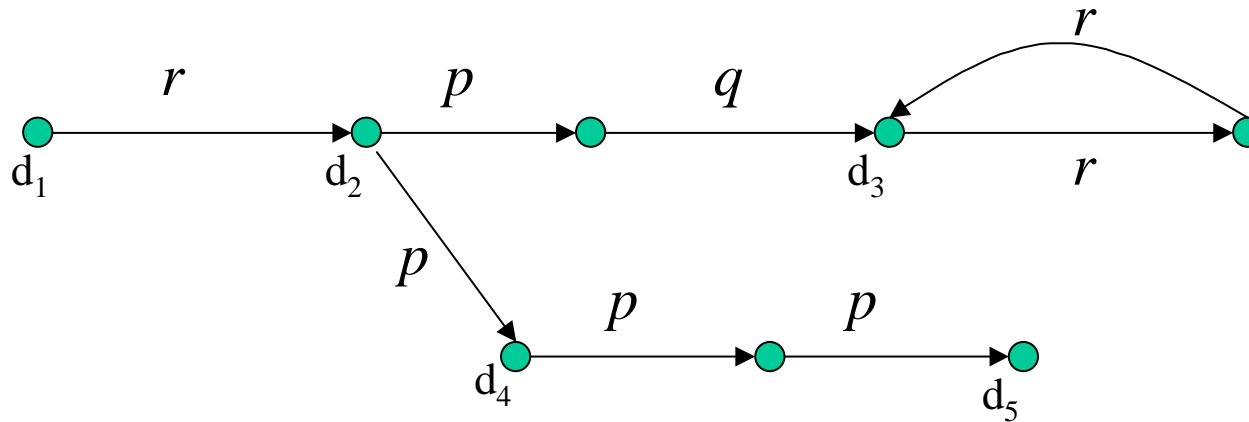
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_2^{\leftarrow}

Transition: $(s_3, 0) \in \delta_A(s_2^{\leftarrow}, p^-)$, transition coming from Q

A run of $A_{(Q,d_1,d_3)}$



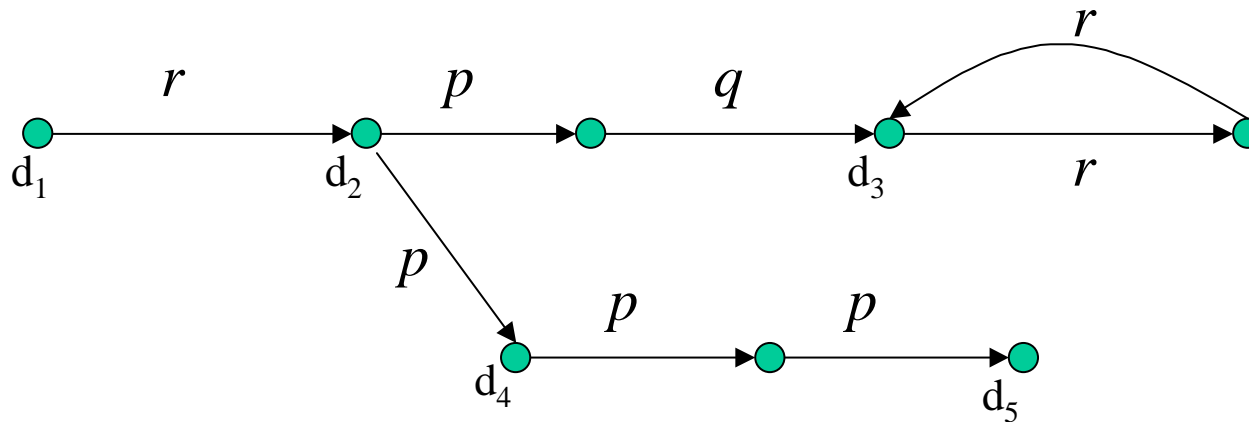
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_3

Transition: $(s_4, 1) \in \delta_A(s_3, p^-)$, transition coming from Q

A run of $A_{(Q,d_1,d_3)}$



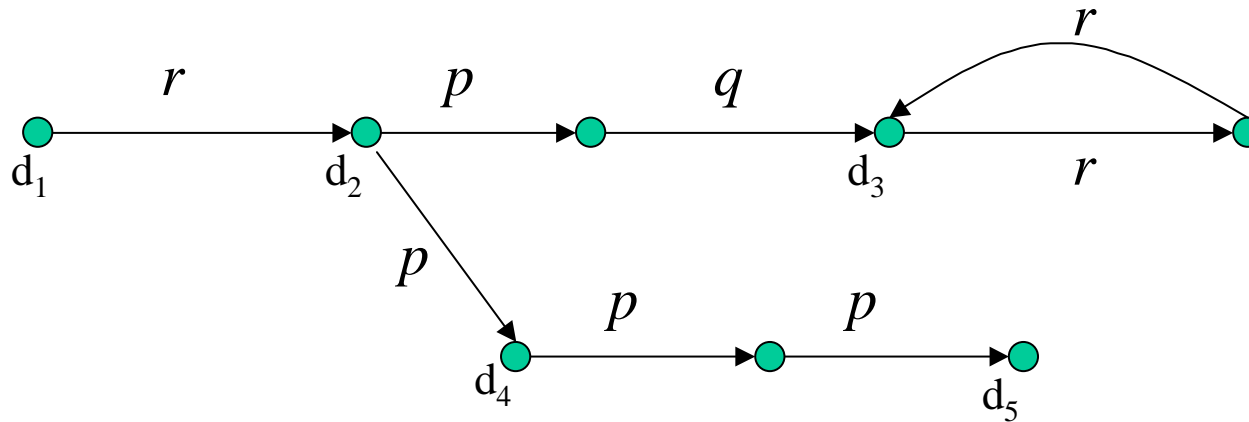
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_4

Transition: $((s_4, d_2), 1) \in \delta_A(s_4, d_2)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



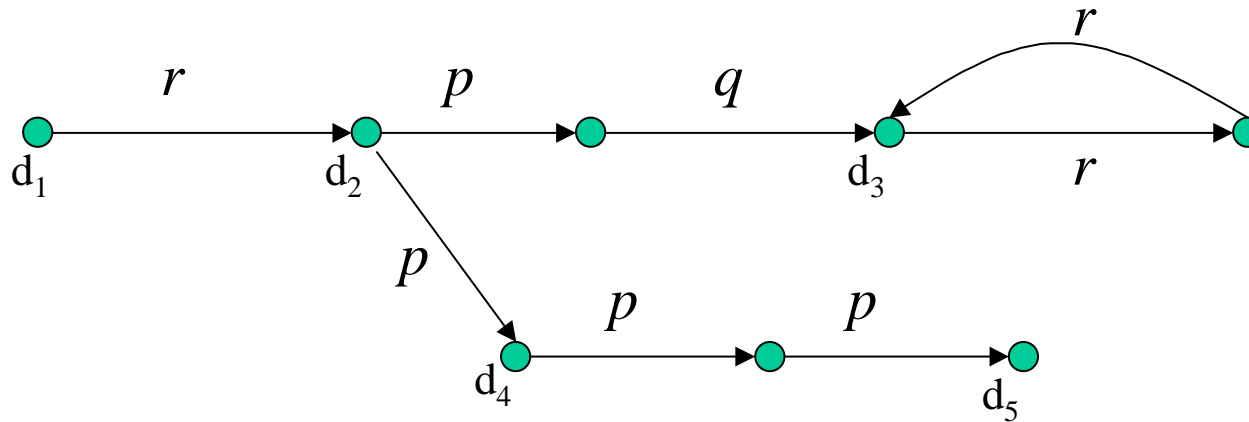
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_4, d_2)

Transition: $((s_4, d_2), 1) \in \delta_A((s_4, d_2), \$)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



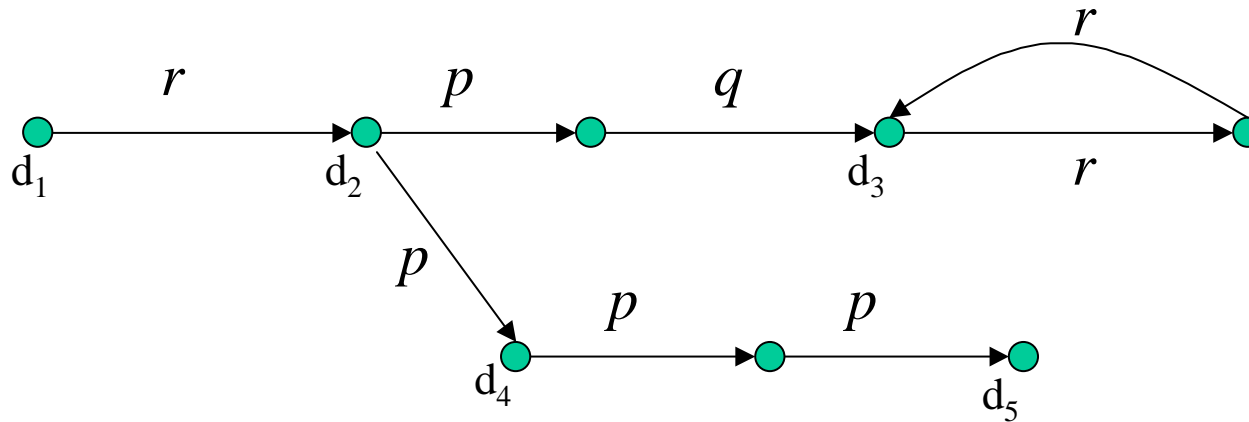
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_4, d_2)

Transition: $((s_4, d_2), 1) \in \delta_A((s_4, d_2), d_3)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



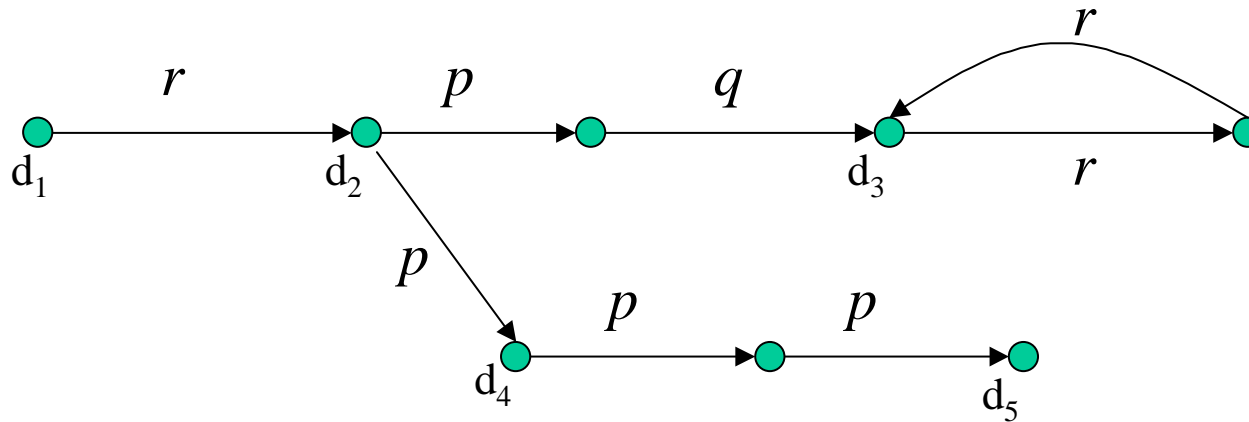
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_4, d_2)

Transition: $((s_4, d_2), 1) \in \delta_A((s_4, d_2), r)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



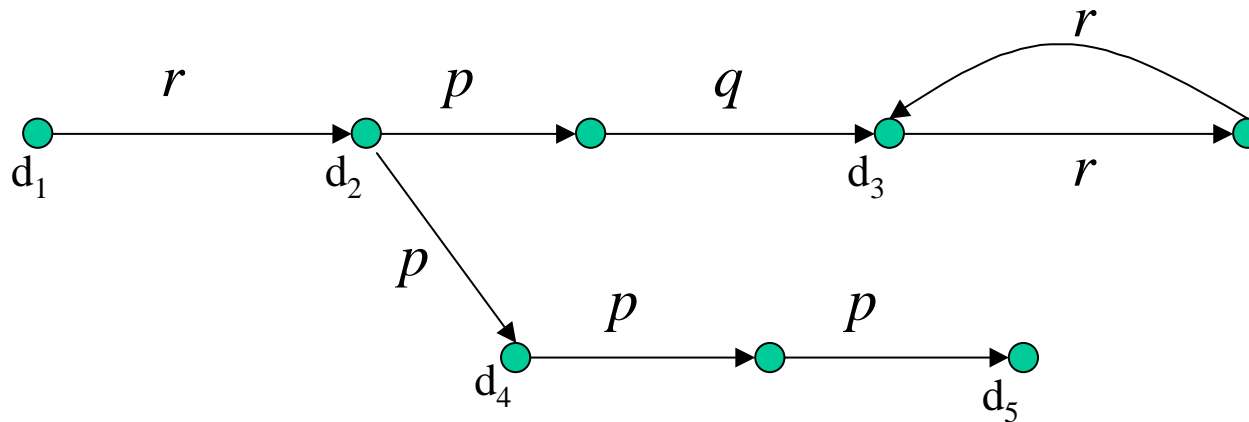
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_4, d_2)

Transition: $((s_4, d_2), 1) \in \delta_A((s_4, d_2), r)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



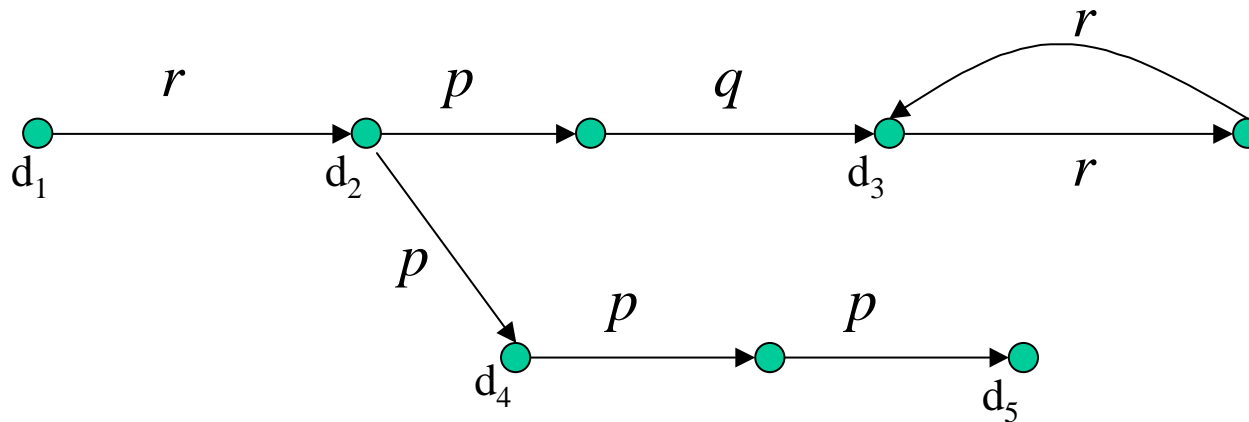
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_4, d_2)

Transition: $((s_4, d_2), 1) \in \delta_A((s_4, d_2), d_3)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



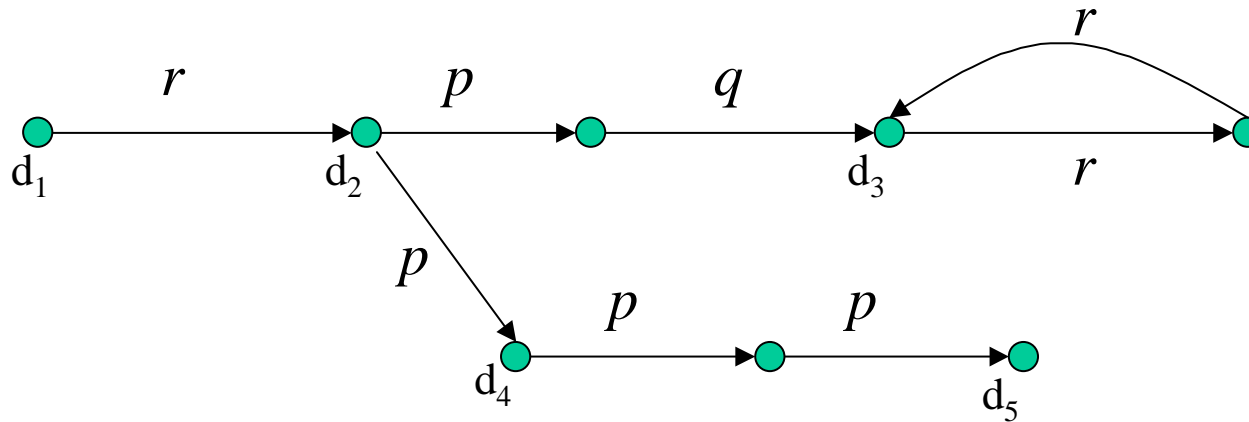
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_4, d_2)

Transition: $((s_4, d_2), 1) \in \delta_A((s_4, d_2), \$)$, search for d_2

A run of $A_{(Q,d_1,d_3)}$



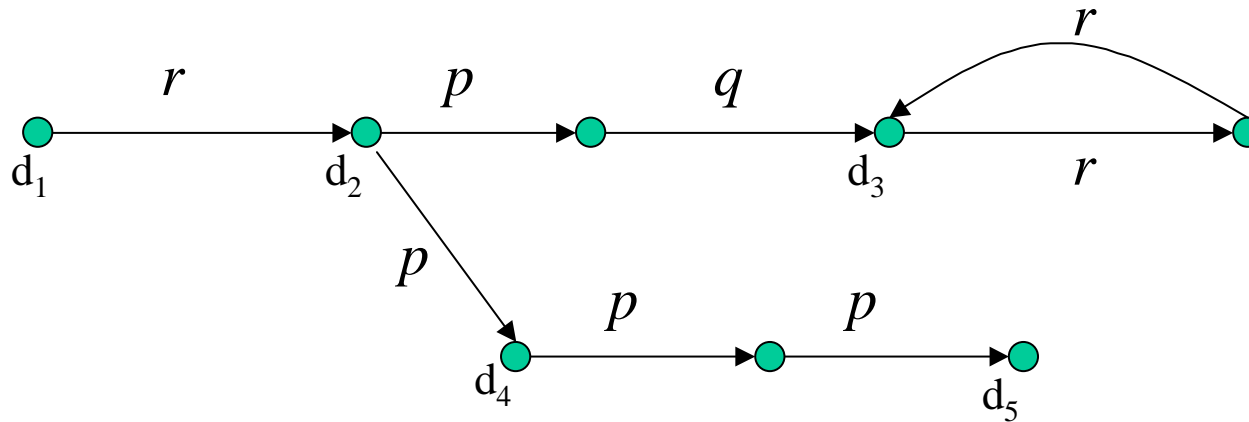
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: (s_4, d_2)

Transition: $(s_4, 0) \in \delta_A((s_4, d_2), d_2)$, exit search mode

A run of $A_{(Q,d_1,d_3)}$



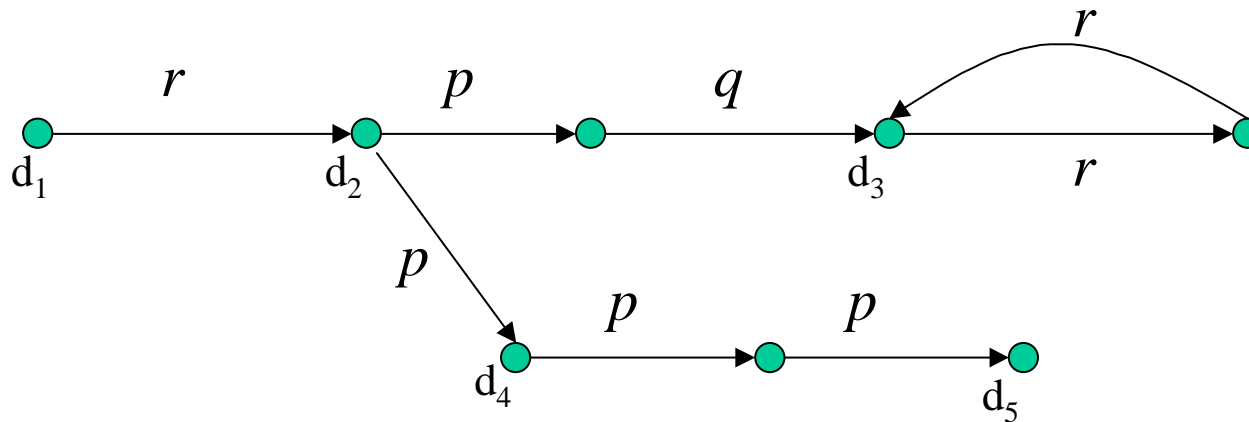
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_4

Transition: $(s_4, 1) \in \delta_A(s_4, d_2)$

A run of $A_{(Q,d_1,d_3)}$



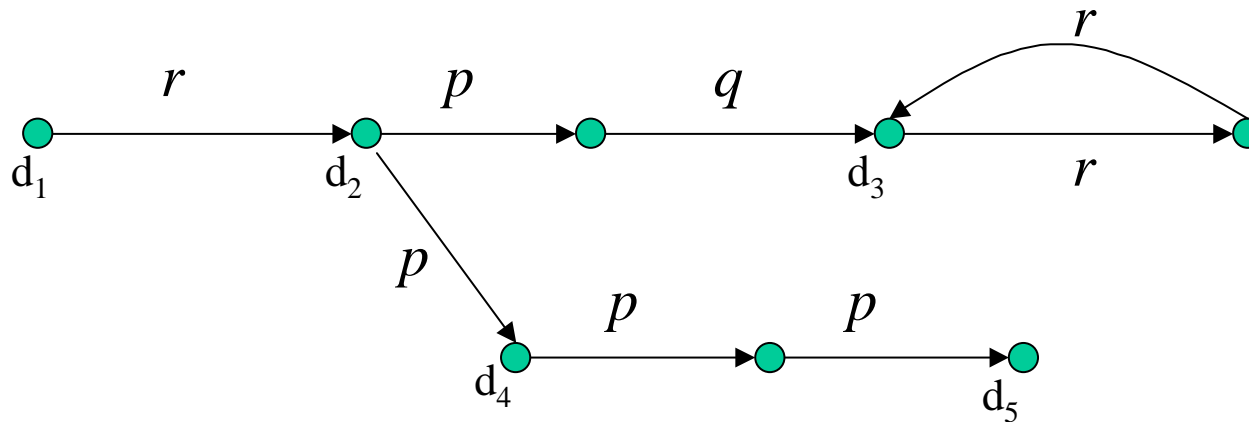
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_4

Transition: $(s_5, 1) \in \delta_A(s_4, p)$, transition coming from Q

A run of $A_{(Q,d_1,d_3)}$



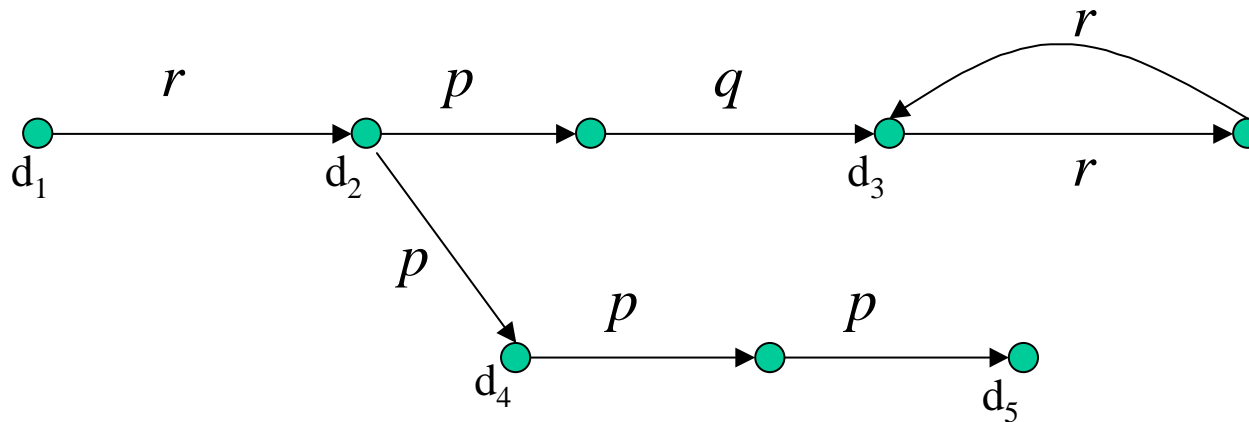
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_5

Transition: $(s_6, 1) \in \delta_A(s_5, q)$, transition coming from Q

A run of $A_{(Q,d_1,d_3)}$



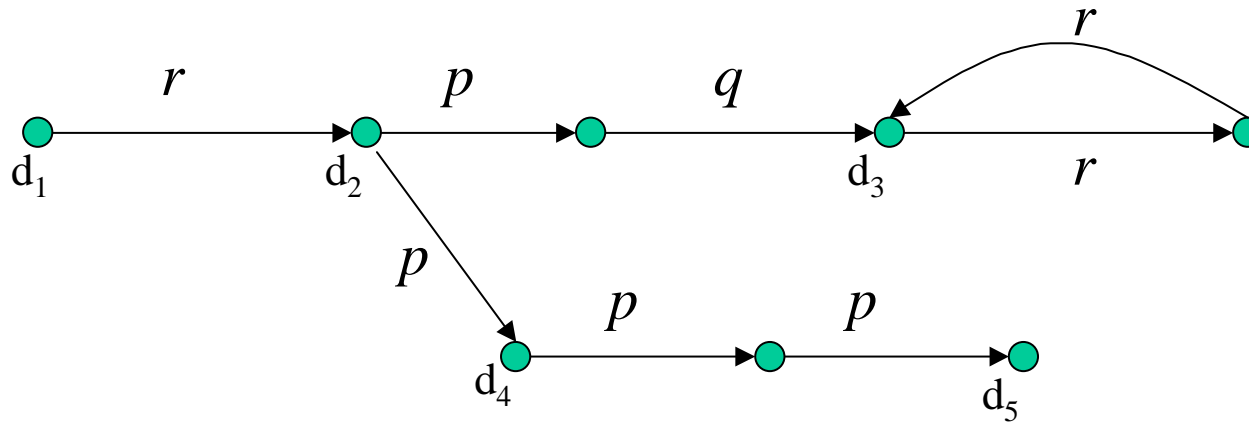
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_6

Transition: $(s_7, 0) \in \delta_A(s_6, d_3)$, s_7 final state

A run of $A_{(Q,d_1,d_3)}$



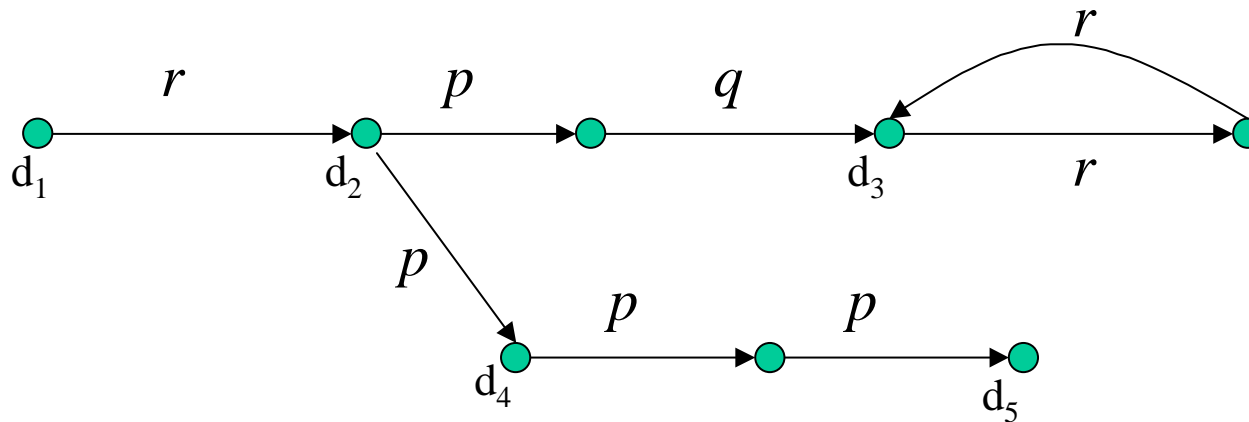
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_7

Transition: $(s_7, 1) \in \delta_A(s_7, d_3)$, s_7 final state

A run of $A_{(Q,d_1,d_3)}$



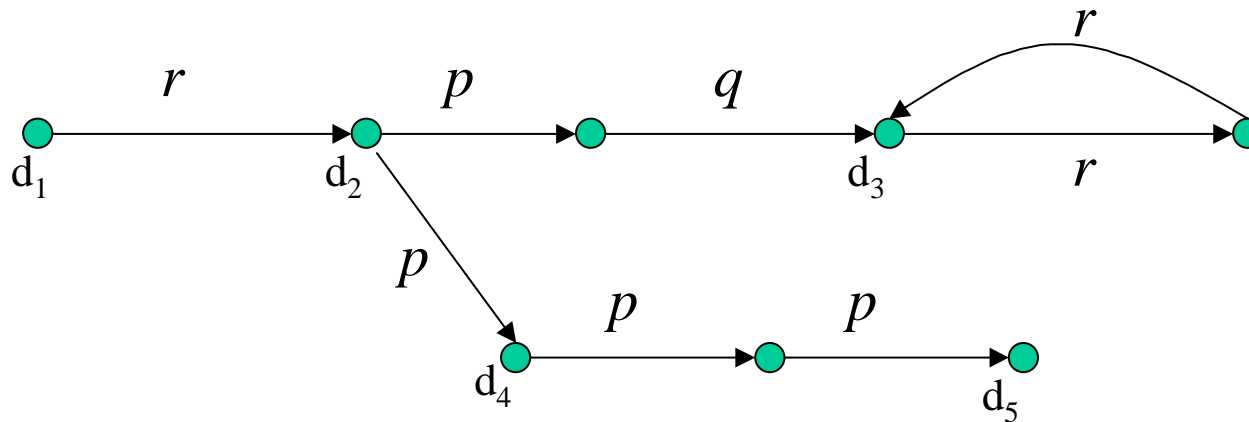
Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_7

Transition: $(s_7, 1) \in \delta_A(s_7, \$)$, s_7 final state

A run of $A_{(Q,d_1,d_3)}$



Word: $\$ d_4 p p d_5 \$ d_1 r d_2 \$ d_4 p^- d_2 \$ d_3 r r d_3 \$ d_2 p q d_3 \$$

$$Q = r \cdot (p \cup q) \cdot (p^- \cdot p)^* \cdot q \cdot q^*$$

State: s_7 final state

Word accepted by $A_{(Q,d_1,d_3)}$!

Query answering: Technique

To check whether $(c, d) \notin Q^{\mathcal{B}}$ for some $\mathcal{B} \in \text{sem}^{\mathcal{C}}(\mathcal{D})$, we check for nonemptiness of A , that is the **intersection** of

- the one-way automaton A_0 that accepts words that represent databases, i.e., words of the form $(\$ \cdot \mathcal{C} \cdot \Sigma^+ \cdot \mathcal{C})^* \cdot \$$
- the one-way automata corresponding to the various $A_{(S_i, a, b)}$ (for each source S_i and for each pair $(a, b) \in S_i^{\mathcal{C}}$)
- the one-way automaton corresponding to the complement of $A_{(Q, c, d)}$

Indeed, any word accepted by such intersection automaton represents a counterexample to $(c, d) \in Q^{\mathcal{C}, \mathcal{D}}$, i.e., a database $\mathcal{B} \in \text{sem}^{\mathcal{C}}(\mathcal{D})$ such that $(c, d) \notin Q^{\mathcal{B}}$.

Query answering: Complexity

- All two-way automata constructed above are of linear size in the size of Q , $def(S_1), \dots, def(S_k)$, and $S_1^{\mathcal{C}}, \dots, S_k^{\mathcal{C}}$. Hence, the corresponding one-way automata would be exponential.
- However, we do not need to construct A explicitly. Instead, we can construct it **on the fly** while checking for nonemptiness.

Query answering for RPQIs is PSPACE-complete (coNP-complete if complexity is measured wrt to the size of source data \mathcal{C} only).

Query answering: the complete picture

Different assumptions:

1. Database *domain* may be:

- completely known (closed domain assumption – CDA)
- **partially known (open domain assumption – ODA)**

2. Each *source* may be:

- **exact**: provides exactly the data specified in the associated view
- **sound: provides a subset of the data specified in the associated view**
- **complete**: provides a superset of the data specified in the associated view

Polynomial intractability: RPQ

Given a graph $G = (N, E)$, we define $\mathcal{D} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, and source database \mathcal{C} :

$$\begin{aligned} V_s &\rightsquigarrow R_s \\ V_e &\rightsquigarrow R_e \\ V_G &\rightsquigarrow R_{rg} \vee R_{gr} \vee R_{rb} \vee R_{br} \vee R_{gb} \vee R_{bg} \\ V_s^{\mathcal{C}} &= \{(c, a) \mid a \in N, c \notin N\} \\ V_e^{\mathcal{C}} &= \{(a, d) \mid a \in N, d \notin N\} \\ V_G^{\mathcal{C}} &= \{(a, b), (b, a) \mid (a, b) \in E\} \\ Q &\leftarrow R_s \cdot M \cdot R_e \end{aligned}$$

where M describes all mismatched edge pairs (e.g., $R_{rg} \cdot R_{rb}$).

If G is 3-colorable, then $\exists \text{db}$ where M (and Q) is empty, i.e. $(c, d) \notin Q^{\mathcal{D}, \mathcal{C}}$.

If G is not 3-colorable, then M is nonempty $\forall \text{db}$, i.e. $(c, d) \in Q^{\mathcal{D}, \mathcal{C}}$.

\implies **coNP-hard wrt data complexity**

Complexity of query answering: the complete picture

Assumption on domain	Assumption on views	Complexity		
		<i>data</i>	<i>expression</i>	<i>combined</i>
closed	all sound	<i>coNP</i>	<i>coNP</i>	<i>coNP</i>
	all exact	<i>coNP</i>	<i>coNP</i>	<i>coNP</i>
	arbitrary	<i>coNP</i>	<i>coNP</i>	<i>coNP</i>
open	all sound	<i>coNP</i>	<i>PSPACE</i>	<i>PSPACE</i>
	all exact	<i>coNP</i>	<i>PSPACE</i>	<i>PSPACE</i>
	arbitrary	<i>coNP</i>	<i>PSPACE</i>	<i>PSPACE</i>

Outline

- Introduction to data integration
- Approaches to modeling and querying
- Case study in LAV
- **Case study in GAV**
- Beyond LAV and GAV
- Conclusions

Coming back to GAV

In GAV, the mapping \mathcal{M} is constituted by a set of assertions:

$$g \rightsquigarrow \phi_{\mathcal{S}}$$

one for each structure g in \mathcal{G} , where $\phi_{\mathcal{S}}$ is a query over \mathcal{S} . Given source database \mathcal{C} , a database \mathcal{B} satisfies \mathcal{M} wrt \mathcal{C} if for each $g \in \mathcal{G}$:

$$\phi_{\mathcal{S}}^{\mathcal{C}} \subseteq g^{\mathcal{B}}$$

If \mathcal{G} **does not have constraints**, we can simply limit our attention to **one** model of the information integration system, and answering queries reduces to

- using \mathcal{M} for computing from \mathcal{C} **the virtual global database**, i.e., tuples satisfying the various $\phi_{\mathcal{S}}$ associated to each structure g of \mathcal{G} ,
- evaluating the query q over the data obtained for the various g 's.

GAV with constraints in the global schema: example

Consider $\mathcal{D} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, with

Global schema \mathcal{G} :

student($Scode, Sname, Scity$), $key\{Scode\}$

university($Ucode, Uname$), $key\{Ucode\}$

enrolled($Scode, Ucode$), $key\{Scode, Ucode\}$

enrolled[$Scode$] \subseteq student[$Scode$]

enrolled[$Ucode$] \subseteq university[$Ucode$]

Sources \mathcal{S} : s_1, s_2, s_3

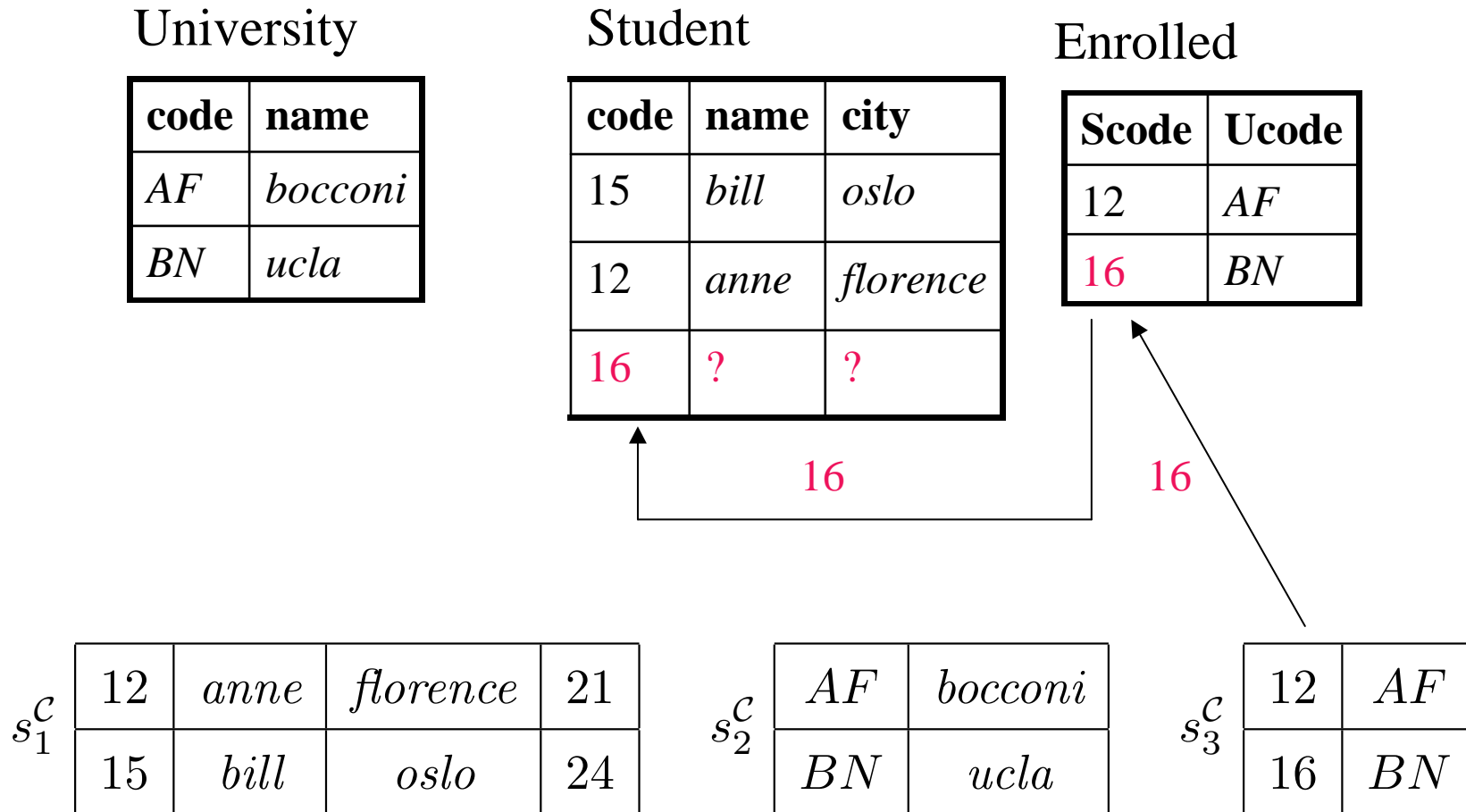
Mapping \mathcal{M} :

student \rightsquigarrow { (X, Y, Z) | $s_1(X, Y, Z, W)$ }

university \rightsquigarrow { (X, Y) | $s_2(X, Y)$ }

enrolled \rightsquigarrow { (X, W) | $s_3(X, W)$ }

Constraints in GAV: example



Constraints in GAV: example

Source database \mathcal{C} :

 $s_1^{\mathcal{C}}$

12	<i>anne</i>	<i>florence</i>	21
15	<i>bill</i>	<i>oslo</i>	24

 $s_2^{\mathcal{C}}$

<i>AF</i>	<i>bocconi</i>
<i>BN</i>	<i>ucla</i>

 $s_3^{\mathcal{C}}$

12	<i>AF</i>
16	<i>BN</i>

$s_3^{\mathcal{C}}(16, BN)$ implies $\text{enrolled}^{\mathcal{B}}(16, BN)$, for all $\mathcal{B} \in \text{sem}^{\mathcal{C}}(\mathcal{D})$.

Due to the integrity constraints in the global schema, **16 is the code of some student** in all $\mathcal{B} \in \text{sem}^{\mathcal{C}}(\mathcal{D})$.

Since \mathcal{C} says nothing about the name and the city of such student, we must accept as legal for \mathcal{D} all virtual global databases that differ in such attributes.

GAV revisited

If \mathcal{G} does have constraints, then several situations are possible, given the source data \mathcal{C} :

- **no model** exists for the data integration system,
- the data integration system has **one model**,
- **several models** exist for the information integration system.

In GAV too, answering queries is an inference process coping with **incomplete information**

Coming back to the analogy with the mystery case, constraints in the global schema can make the investigation report incomplete/incoherent, so that answering queries may require **reasoning** on the investigation report.

A case study in GAV

We deal with the problem of answering queries to data integration systems of the form $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where

- the global schema \mathcal{G} is relational, with both key and foreign key constraints
- the sources in \mathcal{S} are relational
- the mapping \mathcal{M} is of type GAV
- queries are conjunctive queries

Unfolding is not sufficient in our context

Mapping \mathcal{M} :

student $\rightsquigarrow \{ (X, Y, Z) \mid s_1(X, Y, Z, W) \}$
 university $\rightsquigarrow \{ (X, Y) \mid s_2(X, Y) \}$
 enrolled $\rightsquigarrow \{ (X, W) \mid s_3(X, W) \}$

 $s_1^{\mathcal{C}}$

12	<i>anne</i>	<i>florence</i>	21
15	<i>bill</i>	<i>oslo</i>	24

 $s_2^{\mathcal{C}}$

<i>AF</i>	<i>bocconi</i>
<i>BN</i>	<i>ucla</i>

 $s_3^{\mathcal{C}}$

12	<i>AF</i>
16	<i>BN</i>

Query: $\{ (X) \mid \text{student}(X, Y, Z), \text{enrolled}(X, W) \}$

Unfolding wrt \mathcal{M} : $\{ (X) \mid s_1(X, Y, Z, V), s_3(X, W) \}$

retrieves only the answer $\{12\}$ from \mathcal{C} , although $\{12, 16\}$ is the correct answer. The simple unfolding strategy is **not sufficient** in our context.

Most GAV systems use the simple unfolding strategy!

Processing queries in GAV: technique

Techniques for automated reasoning on incomplete information are needed. In our context, we have developed the following technique for processing queries:

- Given query q , we compute another query $exp_{\mathcal{G}}(q)$, called the expansion of q wrt the constraints of \mathcal{G} (partial evaluation)
- We unfold $exp_{\mathcal{G}}(q)$ wrt \mathcal{M} , and obtain a query $unf_{\mathcal{M}}(exp_{\mathcal{G}}(q))$ over the sources
- We evaluate $unf_{\mathcal{M}}(exp_{\mathcal{G}}(q))$ over the source database \mathcal{C}

$exp_{\mathcal{G}}(q)$ can be of exponential size wrt \mathcal{G} , but the whole process has polynomial time complexity wrt the size of \mathcal{C} (see [Calvanese et al, 2001] for details).

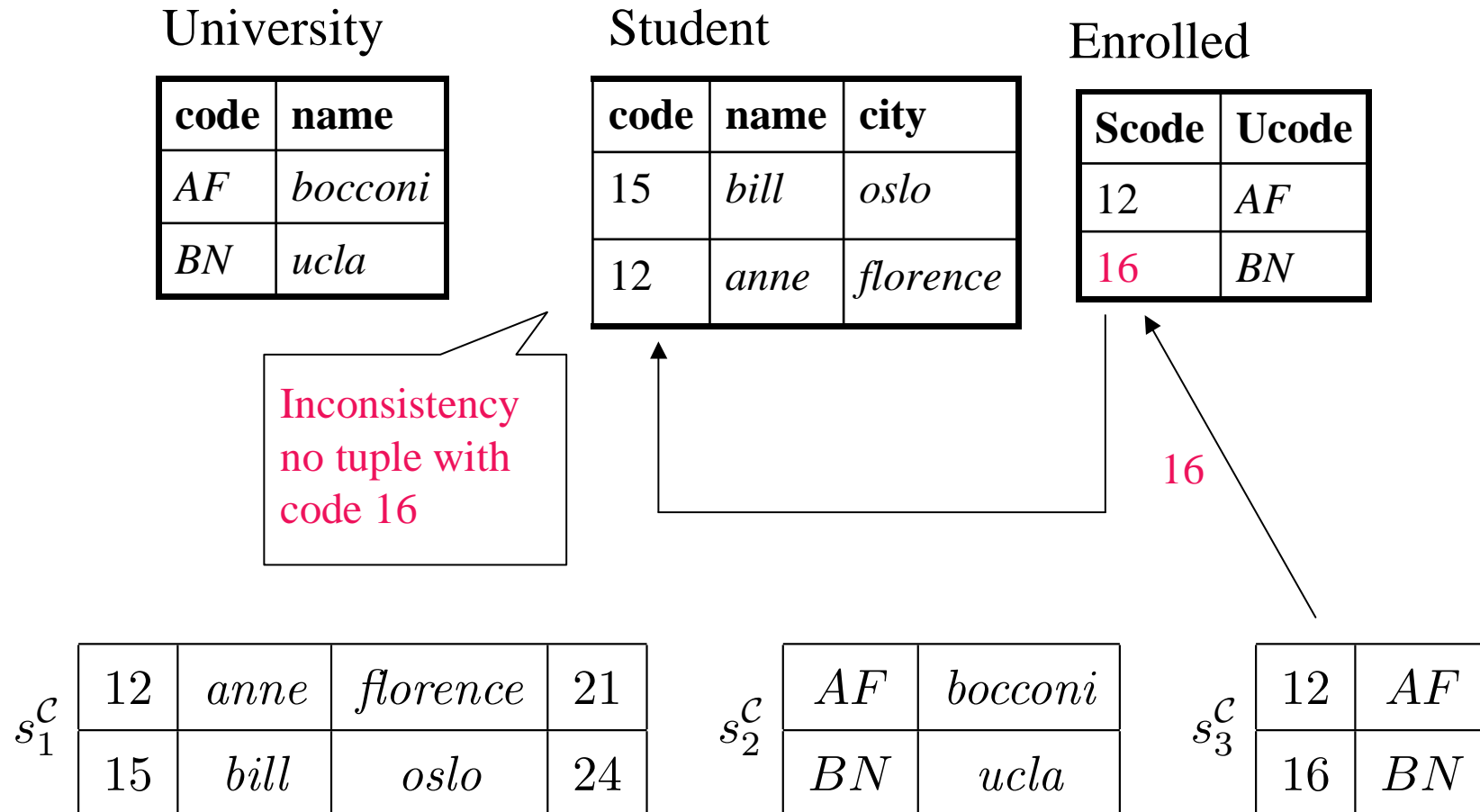
Processing queries in GAV: technique

The problems mentioned above also hold when:

- The global schema is expressed in terms of a **conceptual data model** – see [Calvanese et al, ER 2001]
- An **ontology** is used as global schema – see [Calvanese et al, SWWS 2001]
- The global schema is expressed in terms of a **semistructured data model** (e.g., XML)
- The mapping \mathcal{M} has the following different semantics (**exact sources**): Given source database \mathcal{C} , a database \mathcal{B} satisfies $g \rightsquigarrow \phi_S$ wrt \mathcal{C} if

$$g^{\mathcal{B}} = \phi_S^{\mathcal{C}}$$

The case of exact sources in GAV with constraints



Outline

- Introduction to data integration
- Approaches to modeling and querying
- Case study in LAV
- Case study in GAV
- **Beyond LAV and GAV**
- Conclusions

Beyond LAV and GAV

Global schema: $Work(Researcher, Project)$, $Area(Project, Field)$

Source 1: $Interest(Person, Field)$

Source 2: $Get(Researcher, Grant)$, $For(Grant, Project)$

Mapping:

- r being interested in field f maps to there exists a project p such that r works for p and the area of p is f .
- r getting grant g for project p , maps to r working for p .

This situation **cannot** be represented in GAV or LAV.

The modeling problem: GLAV = GAV + LAV

A **more general method** for specifying the mapping between the global schema and the sources is based on assertions of the forms:

$$\begin{aligned}\phi_{\mathcal{S}} &\rightsquigarrow_s \phi_{\mathcal{G}} \text{ (sound source)} \\ \phi_{\mathcal{S}} &\rightsquigarrow_c \phi_{\mathcal{G}} \text{ (complete source)}\end{aligned}$$

where $\phi_{\mathcal{S}}$ is a query on \mathcal{S} and $\phi_{\mathcal{G}}$ is a query on \mathcal{G} .

Given source database \mathcal{C} , a database \mathcal{B} for \mathcal{G} satisfies \mathcal{M} wrt \mathcal{C} , i.e., if

- for each assertion $\phi_{\mathcal{S}} \rightsquigarrow_s \phi_{\mathcal{G}}$ in \mathcal{M} , we have that $\phi_{\mathcal{S}}^{\mathcal{C}} \subseteq \phi_{\mathcal{G}}^{\mathcal{B}}$,
- for each assertion $\phi_{\mathcal{S}} \rightsquigarrow_c \phi_{\mathcal{G}}$ in \mathcal{M} , we have that $\phi_{\mathcal{G}}^{\mathcal{B}} \subseteq \phi_{\mathcal{S}}^{\mathcal{C}}$

Example of GLAV

Global schema: $Work(Researcher, Project)$, $Area(Project, Field)$

Source 1: $Interest(Person, Field)$

Source 2: $Get(Researcher, Grant)$, $For(Grant, Project)$

GLAV mapping:

$$\{ (r, f) \mid Interest(r, f) \} \rightsquigarrow \{ (r, p) \mid Work(r, p) \wedge Area(p, f) \}$$
$$\{ (r, p) \mid Get(r, g) \wedge For(g, p) \} \rightsquigarrow \{ (r, p) \mid Work(r, p) \}$$

Technique for GLAV

The mapping assertion

$$\phi_S \rightsquigarrow_s \phi_G$$

can be seen as $\phi_S \subseteq g' \subseteq \phi_G$, where g' is a new symbol added to \mathcal{G} .

Therefore, we can translate $\phi_S \rightsquigarrow_s \phi_G$ into:

- the GAV mapping rule $g' \rightsquigarrow \phi_S$
- the constraint $g' \subseteq \phi_G$

thus obtaining a GAV system with constraints, that can be dealt with a variant of the above described technique [Calì et al, FMII 2001].

Outline

- Introduction to data integration
- Approaches to modeling and querying
- Case study in LAV
- Case study in GAV
- Beyond LAV and GAV
- **Conclusions**

Conclusions

- Data integration applications have to cope with **incomplete information**, no matter which is the modeling approach
- Some techniques already developed, but several open problems still remain (in LAV, GAV, and GLAV)
- Many other problems not addressed here are relevant in data integration (e.g., how to construct the global schema, how to deal with inconsistencies, how to cope with updates, ...)
- In particular, given the complexity of sound and complete query answering, it is interesting to look at methods that accept **less quality answers**, trading efficiency for accuracy

Acknowledgements

Many thanks to

- **Andrea Calí**
- **Diego Calvanese**
- **Giuseppe De Giacomo**
- **Domenico Lembo**
- **Moshe Vardi**