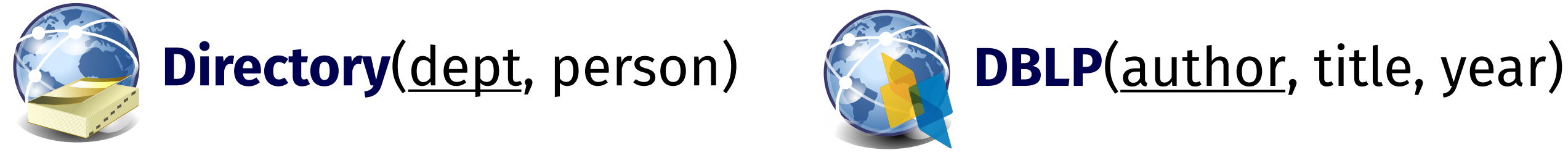
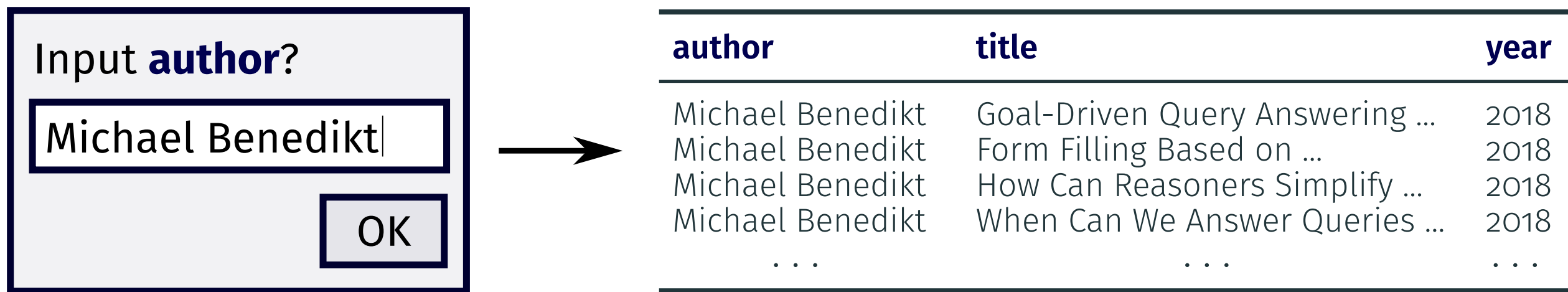


Problem Description

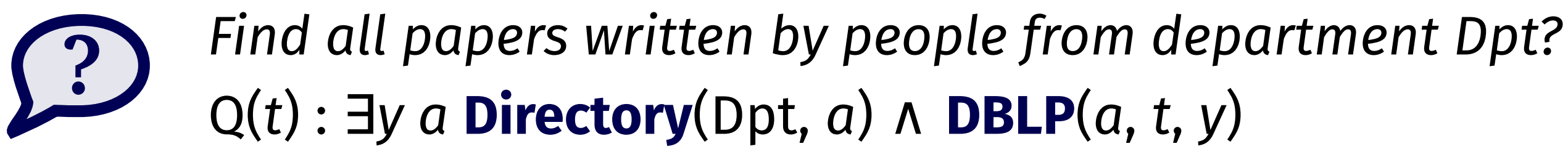
- We have several **Web services** represented as **relations**



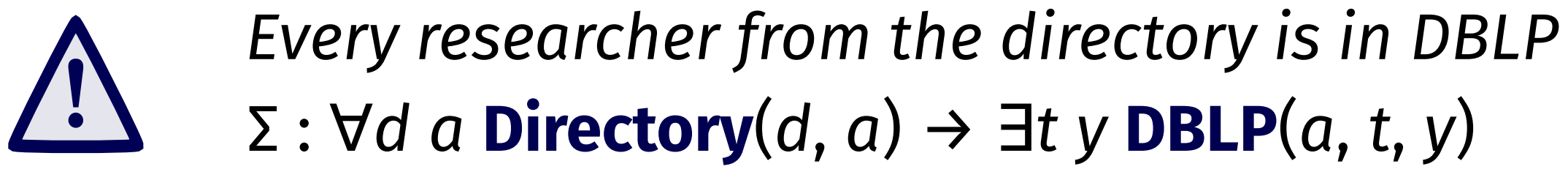
- We can access them by giving a **binding** for the **input attributes** and we obtain the **tuples** that match the **binding**



- We want to answer a **conjunctive query** on the relations



- We can relate the services with **constraints** in a logical fragment



Existing Solutions and New Challenge

- Existing work studied how to **reformulate** the query Q to a **plan**

Example:

- Access the services by giving **bindings**
- Store results in **temporary tables**
- Evaluate monotone **relational algebra**
- Must return **exactly** the output of Q on **all databases** that satisfy Σ

$T_1 \Leftarrow \text{Directory} \Leftarrow \text{Dpt};$
 $T_2 \Leftarrow \text{DBLP} \Leftarrow \pi_{\text{person}}(T_1);$
 $T_3 \Leftarrow \pi_{\text{title}}(T_2);$
 Return T_3

- Problem:** services sometimes do not return **all** matching tuples!

Currently the following URL query parameters are recognized:

Parameter	Description
q	The query string to search for.
h	Maximum number of search results (hits) to return. For bandwidth reasons, this number is capped at 1000.

- The service **DBLP** has a **result bound** of 1000, meaning:
 - If an access matches ≤ 1000 tuples then **all** are returned
 - If an access matches > 1000 tuples then **only 1000** are returned
- **How can we reformulate queries with result-bounded services?**

Main Results

Input: • Service schema S • Query Q • Constraints Σ

Output: Does there exist a plan for Q on S under Σ ?

- What is the **complexity** of this problem for various fragments?
- In what ways are result-bounded methods **useful** for plans?

- We give **schema simplification** results that show when result bounds can be **removed**
- We use them to derive **complexity results**

Fragment	Simplification	Complexity
Inclusion dependencies (IDs)	Existence-check	EXPTIME-complete
Bounded-width IDs	Existence-check	NP-complete
Functional dependencies (FDs)	FD	NP-complete
FDs and UIDs	Choice	NP-hard, in EXPTIME
Equality-free FO	Choice	Undecidable
Frontier-guarded TGDs	Choice	2EXPTIME-complete

Existence-Check Simplification

Idea: even if **DBLP**(author, title, year) has a result bound, we can use it to answer Q: "Has Michael Benedikt published something?"

Def: a schema S with constraints Σ is **existence-check simplifiable** if any query Q that has a plan still does on its **e.-c. approximation**:

- For each relation **DBLP**(author, title, year) with a **result bound** create a new relation **DBLP_{check}**(author)
- Add two new **IDs** in Σ to relate **DBLP_{check}** and **DBLP**:

$$\forall a \text{DBLP}_{\text{check}}(a) \leftrightarrow \exists t y \text{DBLP}(a, t, y)$$

- Forbid direct accesses to **DBLP** (so the result bound is irrelevant)

Thm: schemas with **ID** constraints are existence-check simplifiable

FD Simplification

Idea: if **Dir2**(name, address, phone) has a result bound but Σ has an **FD** name→address, we can answer Q: "Find the address of M.B."

Def: a schema S with constraints Σ is **FD simplifiable** if any query Q that has a plan still does on its **FD approximation**:

- For each relation **Dir2**(name, address, phone) with a **result bound** create **Dir2_{FD}**(name, address) with the **FD-determined** attributes
- Add two **IDs** between **Dir2_{FD}** and **Dir2** and forbid accesses to **Dir2**

Thm: schemas with **FD** constraints are FD simplifiable

Choice Simplification

Idea: sometimes the **value** of the bounds does not matter

Def: a schema S with constraints Σ is **choice simplifiable** if any query Q that has a plan still does if **all bounds are set to 1**

Thm: choice simplifiability holds for **=-free FO**, and for **UID+FDs**

Complexity Techniques and Other Results

- Some complexity bounds shown via a **linearization technique** for query containment under **IDs + side information**
- Results for **expressive arity-two constraints** (GC_2)
- Results for **non-monotone plans** (can use **relational difference**)
- Results when assuming **finiteness** of the underlying database
- Example of FO constraints that are **not choice simplifiable**