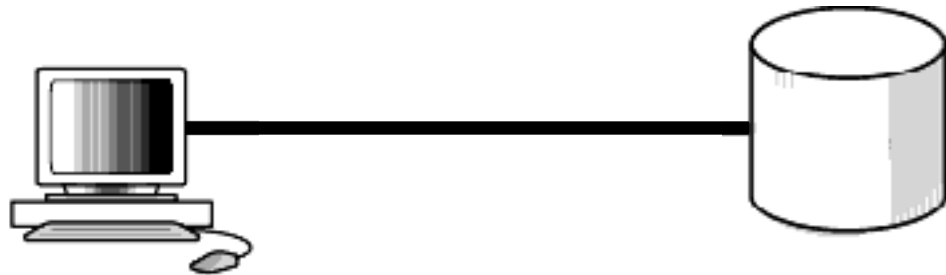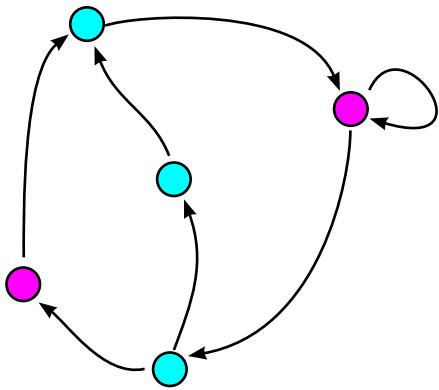# Verification of database-driven systems via amalgamation
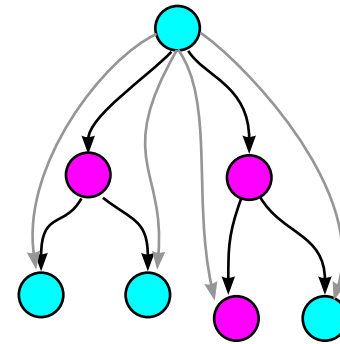
Mikołaj Bojańczyk, Luc Segoufin, <u>Szymon Toruńczyk</u>

Database =
finite relational structure
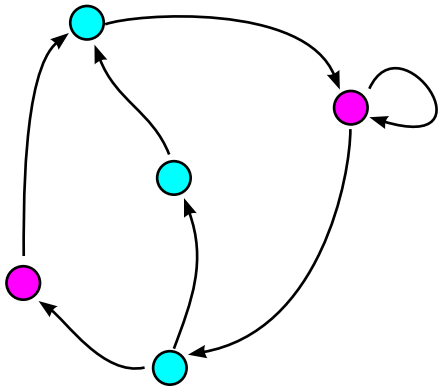

E(–,–), 🟣(–), 🔵(–)   E(–,–), ≤(–,–), 🟣(–), 🔵(–)

Database =
finite relational structure

E(−,−), ●(−), ●(−)

E(−,−), ≤(−,−), ●(−), ●(−)

Configuration

state

register
valuation

database

Transitions

$\delta_{p,q}$
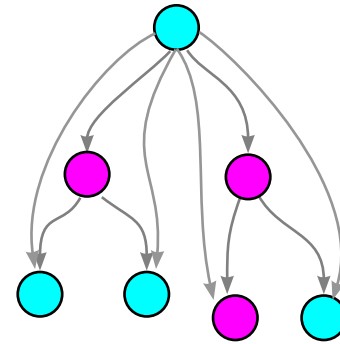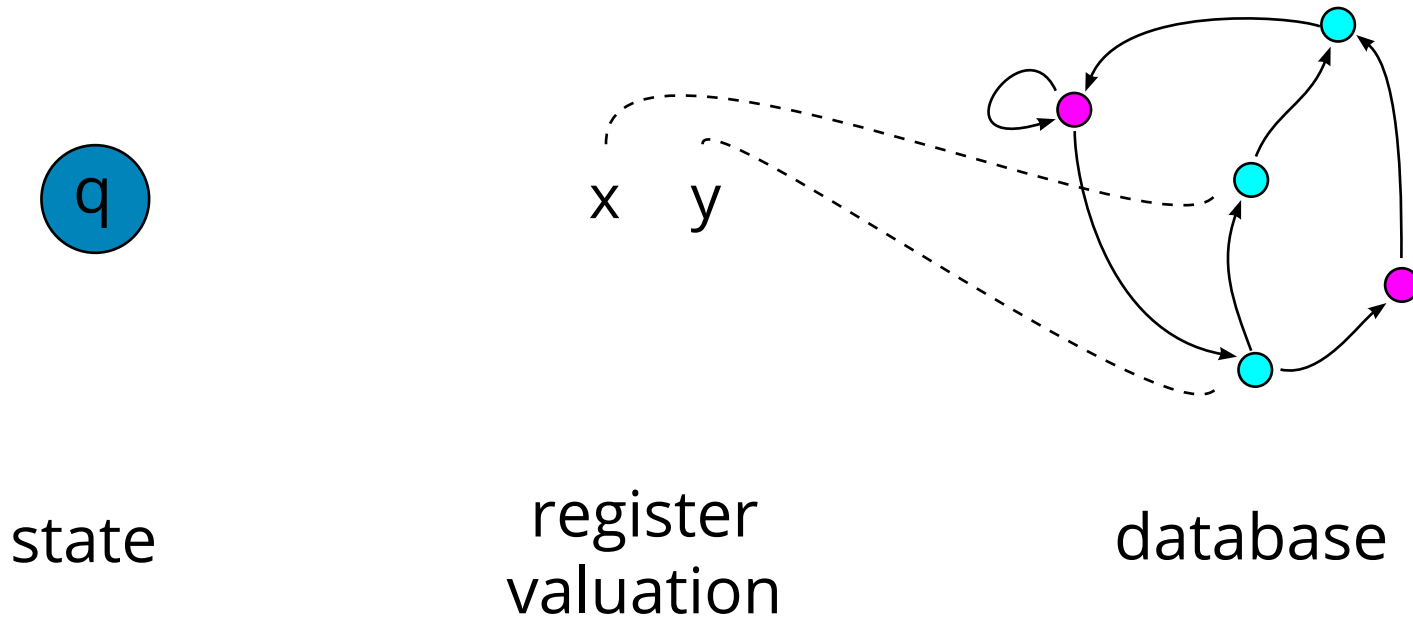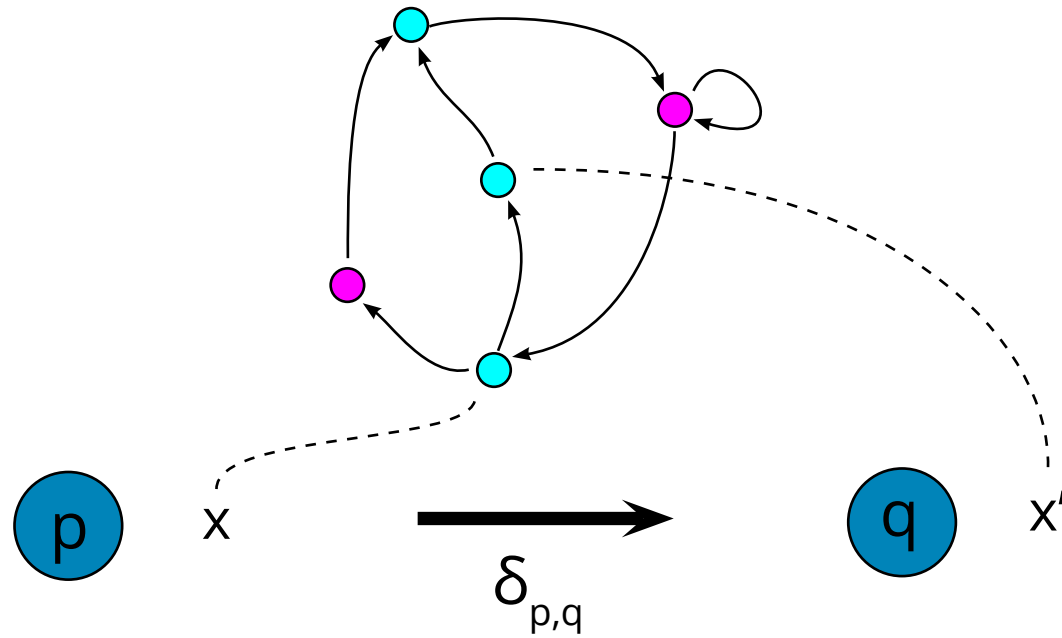
quantifier-free formula, e.g.

$E(\ x\ ,x')\ \land\ \bigcirc(\ x\ )$

Reachability

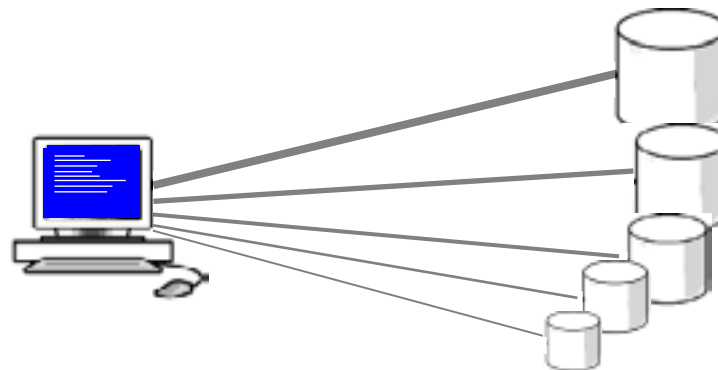**Parameter:** a class of databases C

Input: a database-driven system

states, initial and final states, registers, formulas $\delta_{p,q}$

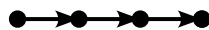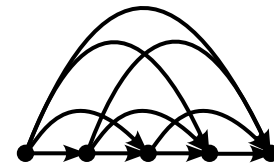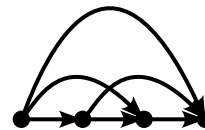**Decide:** does the system have some run to a final state for *some* database in the class C?

The database is unknown!
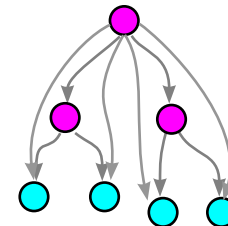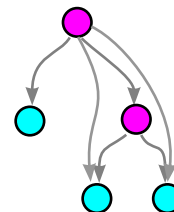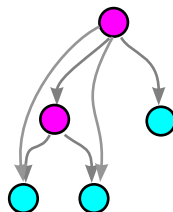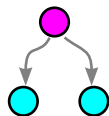
Classes of databases

- all labeled graphs – decidable

- all relational structures over given signature – decidable

- all strings with successor relation – undecidable (via Minsky)

- all strings with linear order – decidable

- a given regular tree language – ?   (XML motivation)

Contribution

Theorem 1. Reachability is decidable if C has *amalgamation*.

Fundamental notion from model theory

Decidability via emptiness of automata with suitable atoms

Theorem 2. Reachability is decidable if C is a regular tree language.
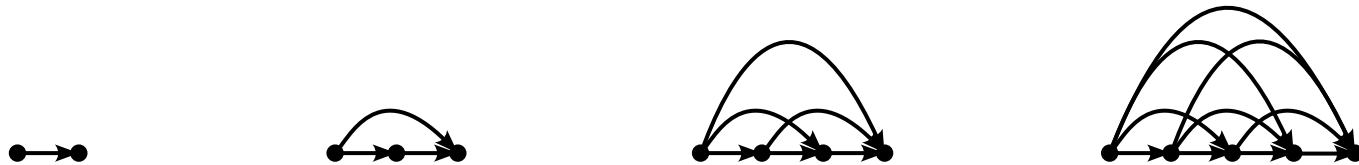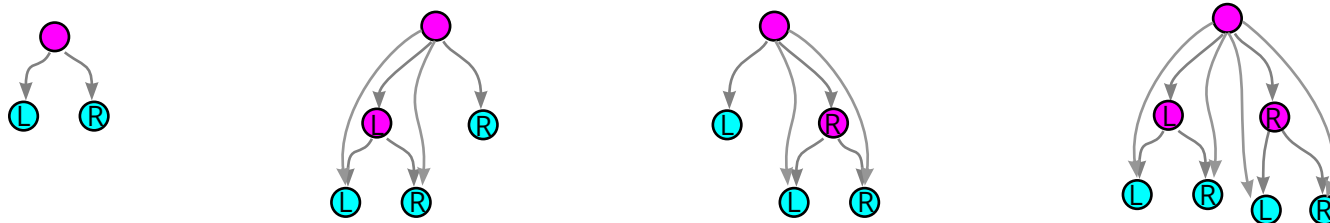
Classes of databases

- all labeled graphs – amalgamation

- all relational structures over given signature – amalgamation

- all strings with successor relation – no amalgamation

- all strings with linear order – amalgamation

- any regular tree language – extends to amalgamation class

Thank you!