ECI 2015
Buenos Aires

Fundamentos lógicos de bases de datos

(Logical foundations of databases)

Diego Figueira **Gabriele Puppis**

CNRS LaBRI

Trading expressiveness for efficiency

Alternation of quantifiers significantly affects complexity
(recall that evaluation of QBF is PSPACE-complete: $\forall x\ \exists y\ \forall z\ \exists w \ldots \phi$).

What happens if we disallow $\forall$ and $\neg$?

The class NP

$$\text{LOGSPACE} \subseteq \text{PTIME} \subseteq \text{PSPACE} \subseteq \text{EXPTIME}$$

The class NP

LOGSPACE $\subseteq$ PTIME $\subseteq$ **NP** $\subseteq$ PSPACE $\subseteq$ EXPTIME

NP = Problems whose solutions can be witnessed by a ***certificate***
to be guessed and checked in *polynomial time* (e.g. a colouring)

The class NP

LOGSPACE ⊆ PTIME ⊆ **NP** ⊆ PSPACE ⊆ EXPTIME

NP = Problems whose solutions can be witnessed by a ***certificate*** to be guessed and checked in *polynomial time* (e.g. a colouring)

Examples:

- **3-COLORABILITY**: Given a graph G, can we assign a colour from $\{R,G,B\}$ to each node so that adjacent nodes have always different colours ?
- **SAT**: Given a propositional formula, e.g. $(p \vee \neg q \vee r) \wedge (\neg p \vee s) \wedge (\neg s \vee \neg p)$, can we assign a truth value to each variable so that the formula becomes true ?
- **MONEY-CHANGE**: Given an amount of money A and a set of coins $\{B_1, \ldots, B_n\}$, can we find a subset $S \subseteq \{B_1, \ldots, B_n\}$ such that $\sum S = A$?

The class NP

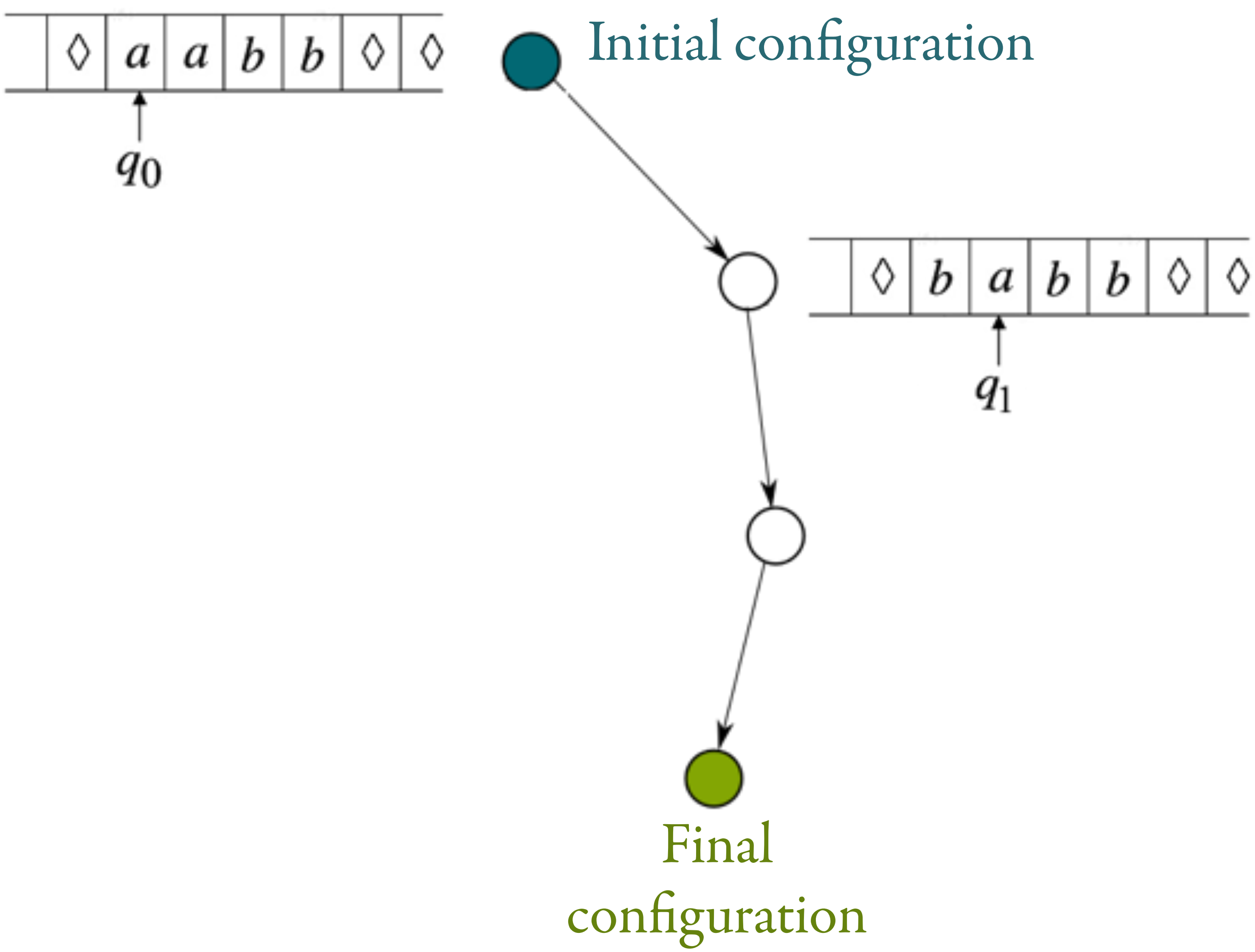
LOGSPACE $\subseteq$ PTIME $\subseteq$ **NP** $\subseteq$ PSPACE $\subseteq$ EXPTIME

NP = Problems whose solutions can be witnessed by a ***certificate*** to be guessed and checked in *polynomial time* (e.g. a colouring)

The class NP

LOGSPACE ⊆ PTIME ⊆ **NP** ⊆ PSPACE ⊆ EXPTIME

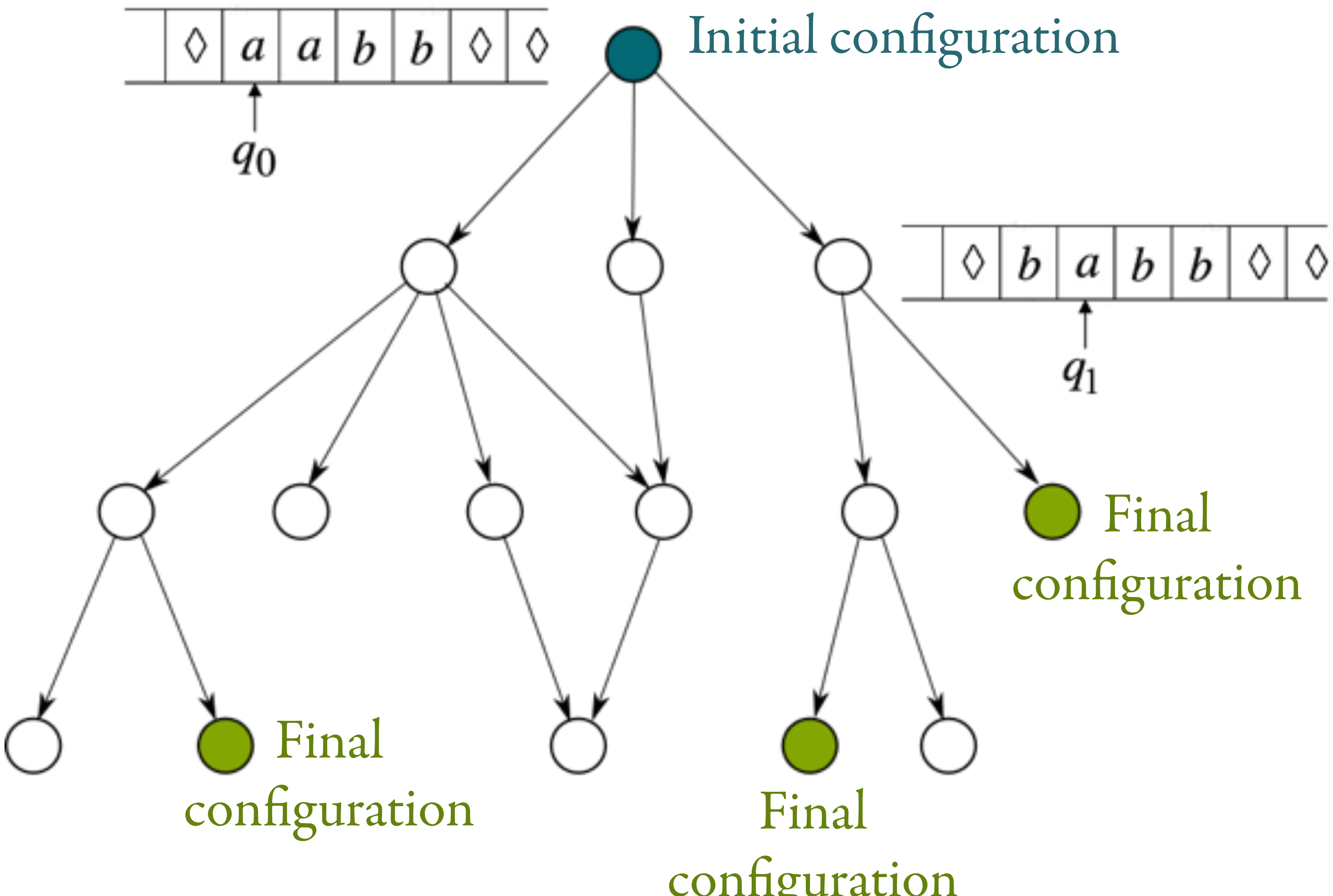
NP = Problems whose solutions can be witnessed by a ***certificate*** to be guessed and checked in *polynomial time* (e.g. a colouring)



The class NP

LOGSPACE ⊆ PTIME ⊆ **NP** ⊆ PSPACE ⊆ EXPTIME

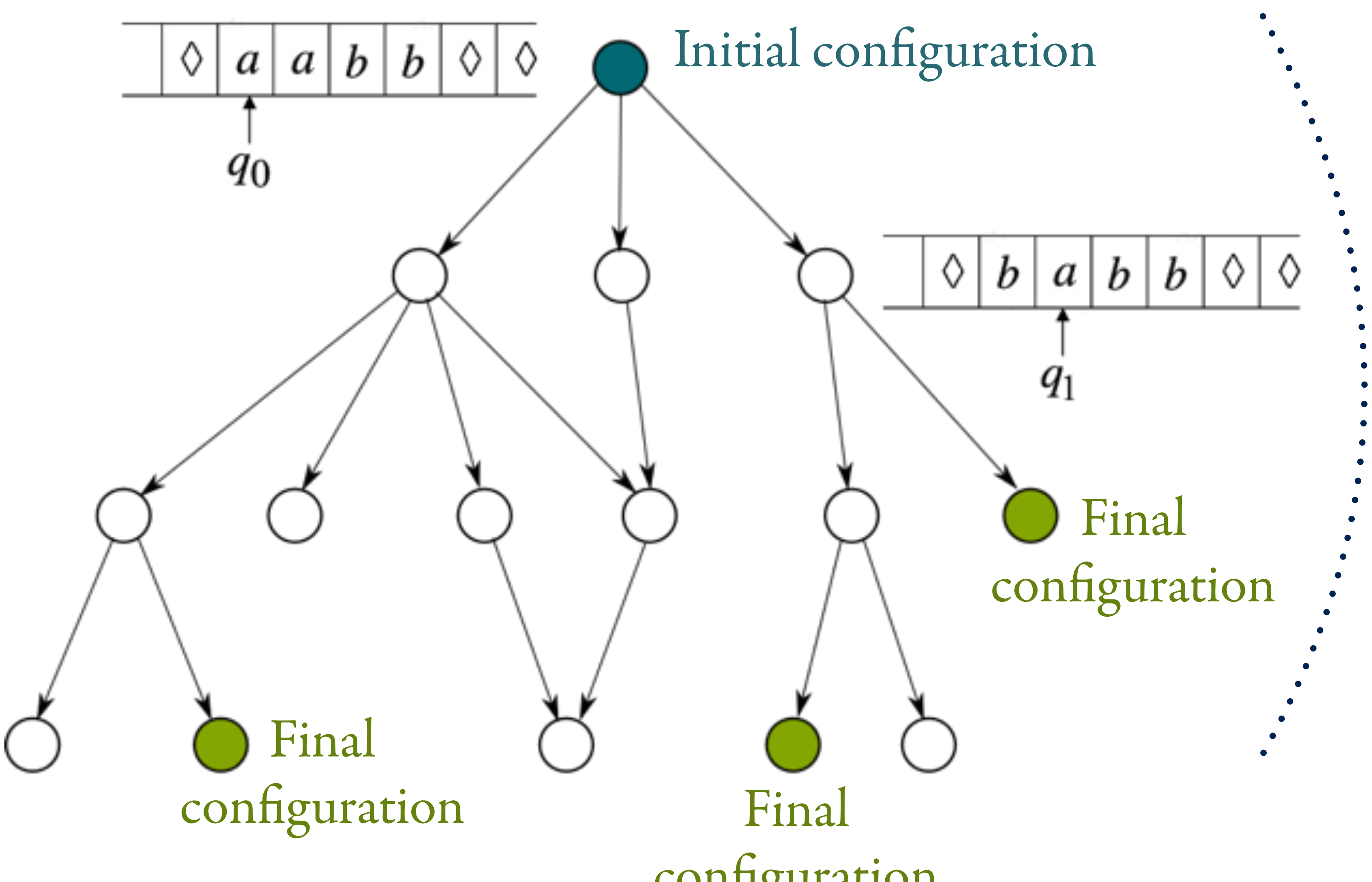
NP = Problems whose solutions can be witnessed by a ***certificate*** to be guessed and checked in *polynomial time* (e.g. a colouring)



The class NP

LOGSPACE ⊆ PTIME ⊆ **NP** ⊆ PSPACE ⊆ EXPTIME

NP = Problems whose solutions can be witnessed by a ***certificate*** to be guessed and checked in *polynomial time* (e.g. a colouring)

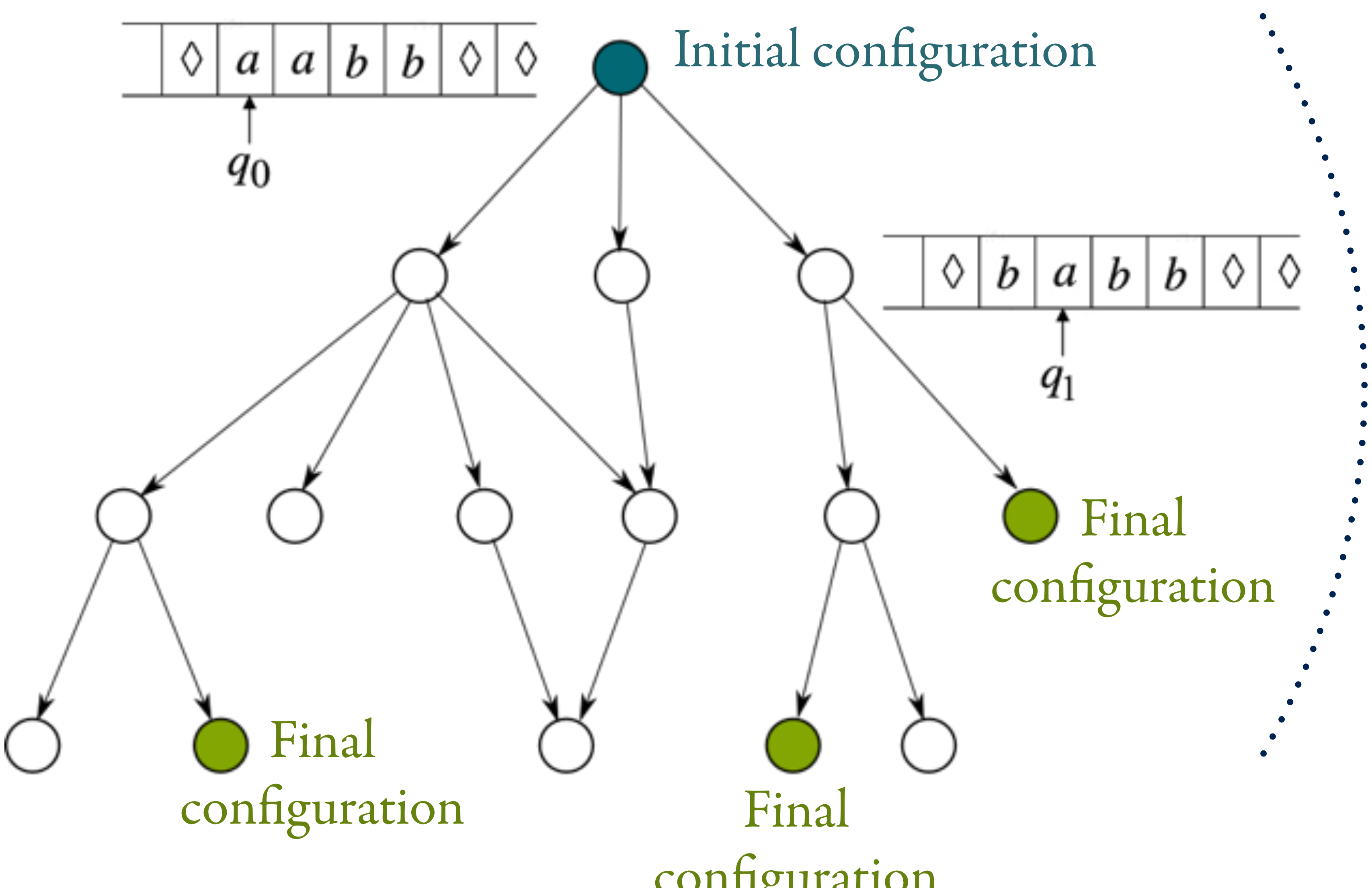


Non-deterministic transitions

The class NP

LOGSPACE ⊆ PTIME ⊆ **NP** ⊆ PSPACE ⊆ EXPTIME

NP = Problems whose solutions can be witnessed by a ***certificate*** to be guessed and checked in *polynomial time* (e.g. a colouring)



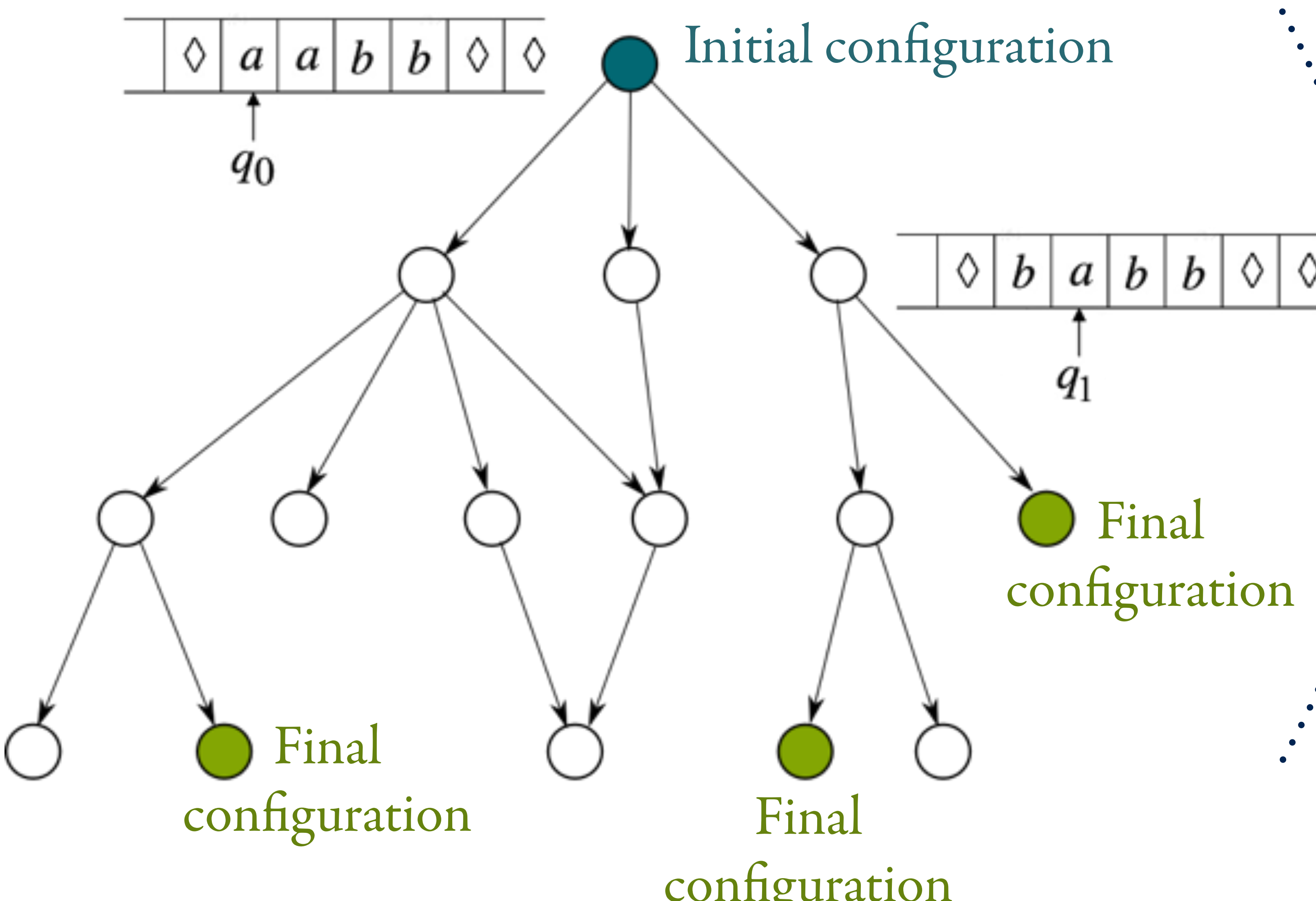
Non-deterministic transitions

Many paths, each has length bounded by a **polynomial**

The class NP

LOGSPACE ⊆ PTIME ⊆ **NP** ⊆ PSPACE ⊆ EXPTIME

NP = Problems whose solutions can be witnessed by a ***certificate*** to be guessed and checked in *polynomial time* (e.g. a colouring)



Non-deterministic transitions

Many paths, each has length bounded by a **polynomial**

A solution exists if there is at least a **successful** path.

Question

Consider: **Positive FO = FO without ∀,¬**

E.g. $\phi = \exists x \, \exists y \, \exists z \, . \, \big(E(x,y) \vee E(y,z)\big) \wedge \big(\, y{=}z \vee E(x,z)\big)$

What is the complexity of evaluating Positive FO on graphs ?

Question

Consider: **Positive FO = FO without $\forall, \neg$**

E.g. $\phi = \exists x \, \exists y \, \exists z \, . \, \big(E(x,y) \vee E(y,z)\big) \wedge \big(y{=}z \vee E(x,z)\big)$

What is the complexity of evaluating Positive FO on graphs ?

Solution

This is in NP: Given ϕ and $G{=}(V,E)$
it suffices to guess a binding $\alpha : \{ x, y, z, \dots \} \to V$
and then verify that the formula holds.

Conjunctive Queries

Def.

CQ = FO without ∀, ¬, ∨

Eg: $\phi(x, y) = \exists z\, .\, (\mathrm{Parent}(x, z) \wedge \mathrm{Parent}(z, y))$

Usual notation: "Grandparent(X,Y) : – Parent(X,Z), Parent(Z,Y)"

Conjunctive Queries

Def.

CQ = FO without $\forall, \neg, \vee$

Normal form: " $\exists x_1, \ldots, x_n \, . \, \phi(x_1, \ldots, x_n)$ "

quantifier-free and no equalities!

Eg: $\phi(x, y) = \exists z \, . \, (\text{Parent}(x, z) \wedge \text{Parent}(z, y))$

Usual notation: "Grandparent(X,Y) : – Parent(X,Z), Parent(Z,Y)"

Conjunctive Queries

Def.

CQ = FO without $\forall, \neg, \vee$

Normal form: " $\exists x_1, \ldots, x_n \,.\, \phi(x_1, \ldots, x_n)$ "

quantifier-free and no equalities!

Eg: $\phi(x, y) = \exists z \,.\, (\text{Parent}(x, z) \wedge \text{Parent}(z, y))$

Usual notation: "Grandparent(X,Y) : – Parent(X,Z), Parent(Z,Y)"

It corresponds to positive "SELECT-FROM-WHERE" SQL queries

```
Select ...
From ...
Where Z
```

no negation or disjunction

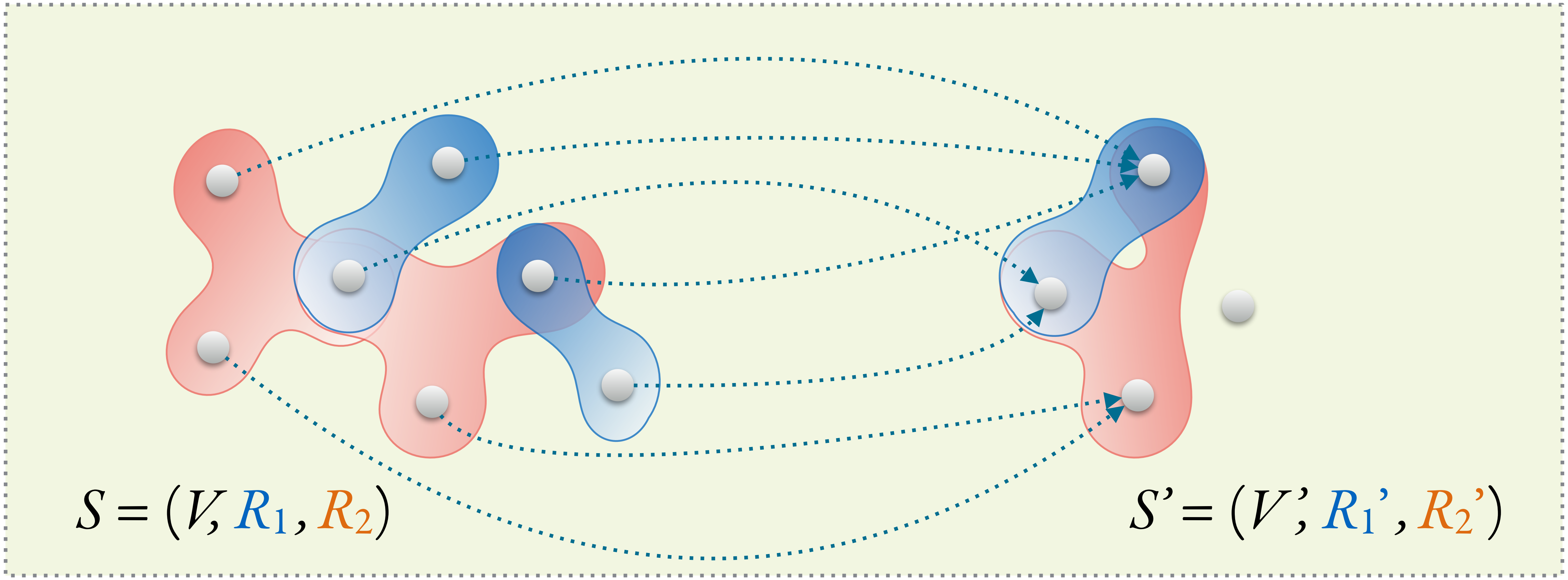
It corresponds to "π-σ-×" RA queries

$\pi_X(\sigma_Z(R_1 \times \cdots \times R_n))$

no negation

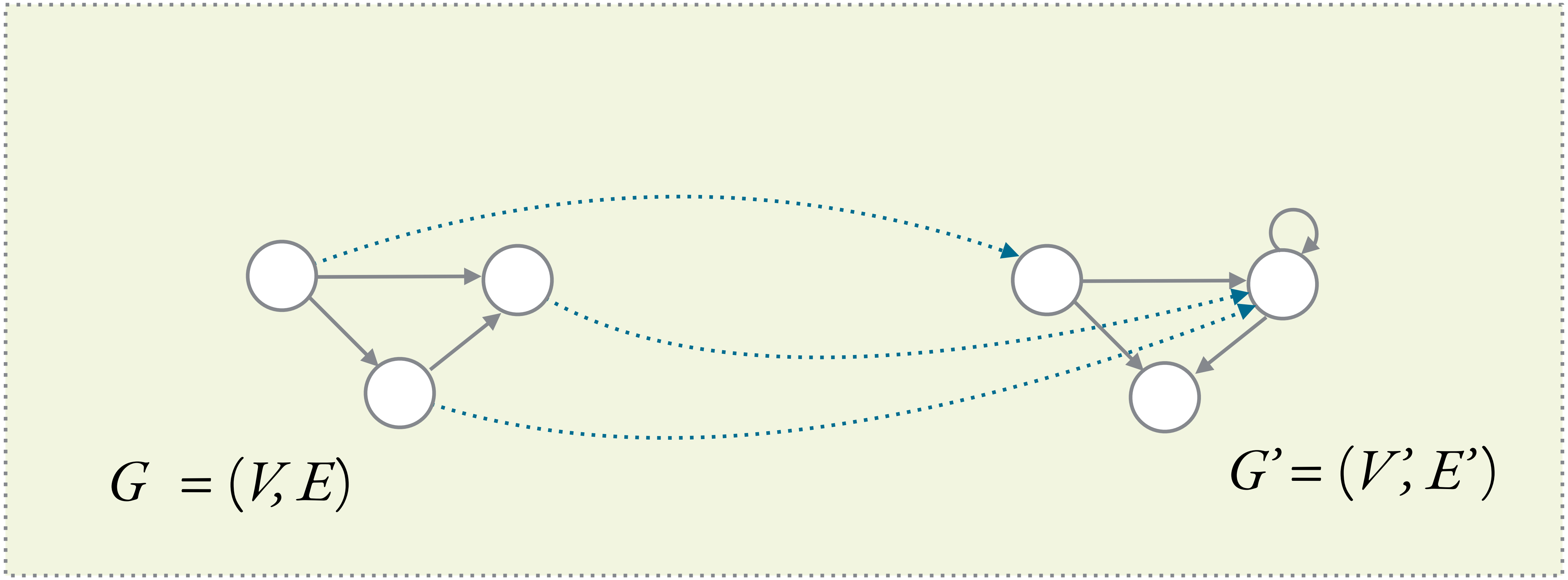
Homomorphisms

Homomorphism between structures $S=(V, R_1, \ldots, R_n)$ and $S'=(V', R_1', \ldots, R_n')$
is a function $h : V \longrightarrow V'$ such that
$(x_1, \ldots, x_n) \in R_i$ implies $(h(x_1), \ldots, h(x_n)) \in R_i'$



Homomorphisms

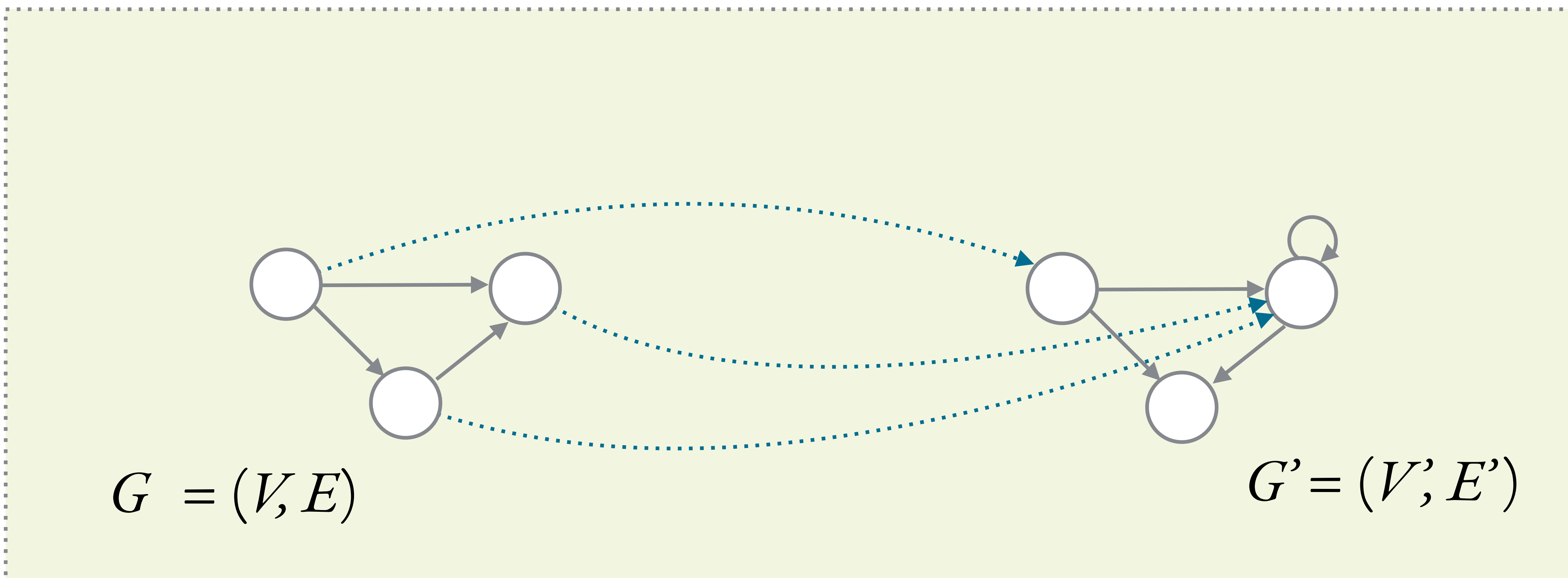
Homomorphism between structures $S=(V, R_1, \ldots, R_n)$ and $S'=(V', R_1', \ldots, R_n')$
is a function $h : V \longrightarrow V'$ such that
$(x_1, \ldots, x_n) \in R_i$ implies $(h(x_1), \ldots, h(x_n)) \in R_i'$



Canonical structures

Canonical structure S_ϕ of a Conjunctive Query ϕ has

- variables as nodes
- tuples $(x_1, \ldots, x_n) \in R_i$
  for all atomic sub-formulas $R_i(x_1, \ldots, x_n)$ of ϕ

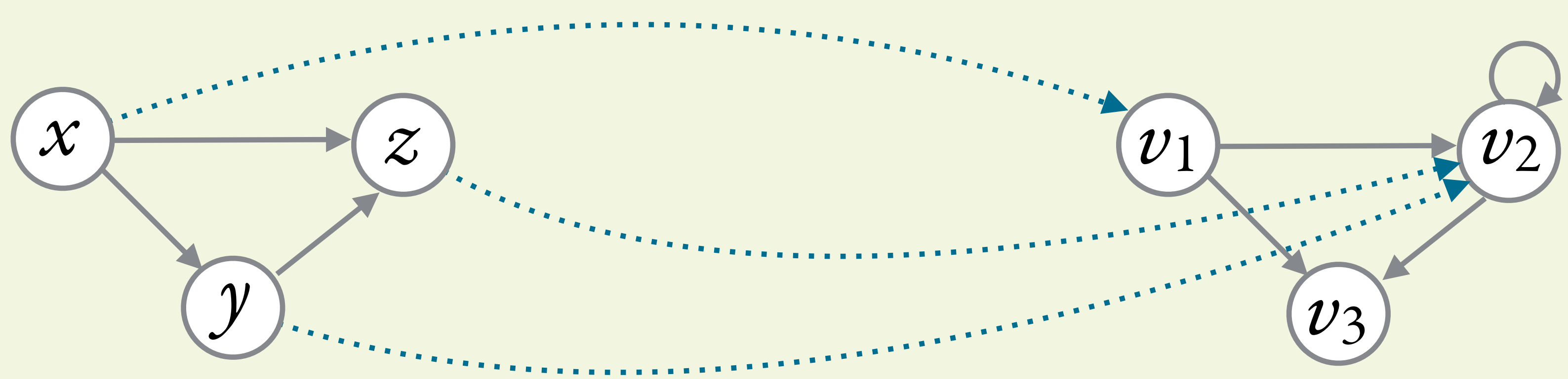


Canonical structures

Canonical structure S_ϕ of a Conjunctive Query ϕ has

- variables as nodes
- tuples $(x_1, \ldots, x_n) \in R_i$
  for all atomic sub-formulas $R_i(x_1, \ldots, x_n)$ of ϕ

E.g.: $\phi = \exists x \, \exists y \, \exists z \, . \, (E(x,y) \wedge E(y,z) \wedge E(x,z))$



$G_\phi = (V, E)$ $\quad$ $G' = (V', E')$

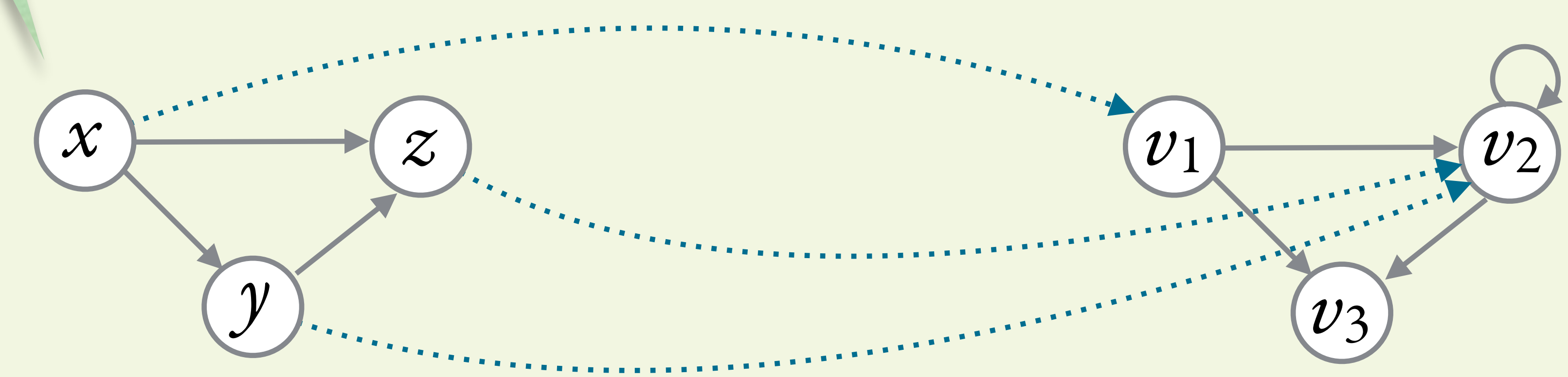
Canonical structures

Canonical structure S_ϕ of a Conjunctive Query ϕ has

- variables as nodes
- tuples $(x_1, \ldots, x_n) \in R_i$
  for all atomic sub-formulas $R_i(x_1, \ldots, x_n)$ of ϕ

Fact 1: $G_\phi \models \phi$

E.g.: $\phi = \exists x\, \exists y\, \exists z\, .\, (E(x,y) \wedge E(y,z) \wedge E(x,z))$



$G_\phi = (V, E)$

$G' = (V', E')$

Canonical structures

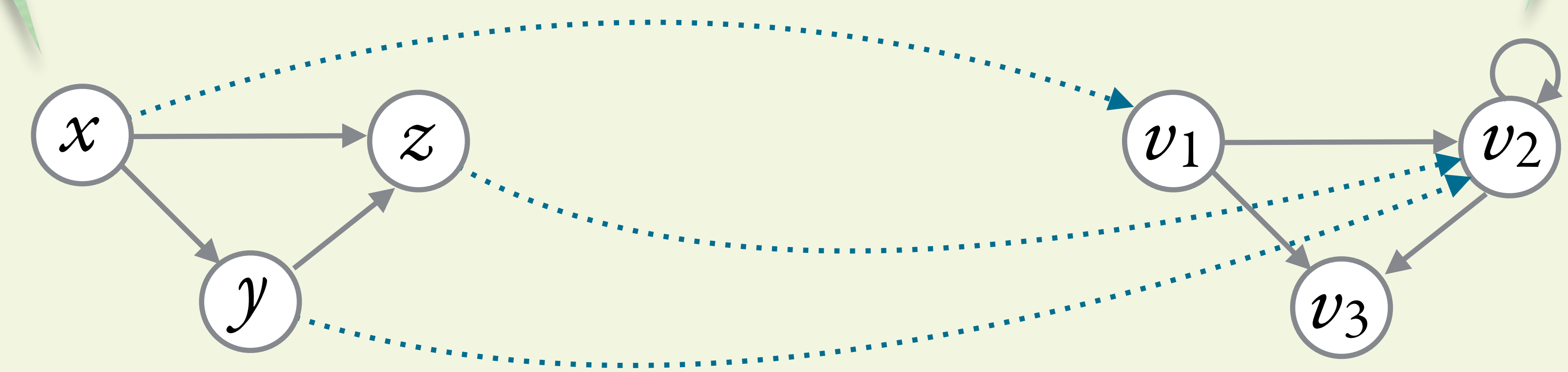
Canonical structure S_ϕ of a Conjunctive Query ϕ has

- variables as nodes
- tuples $(x_1, \dots, x_n) \in R_i$
  for all atomic sub-formulas $R_i(x_1, \dots, x_n)$ of ϕ

Fact 1: $G_\phi \vDash \phi$

Fact 2: $h(G_\phi) \vDash \phi$

E.g.: $\phi = \exists x \, \exists y \, \exists z \, . \, (E(x,y) \wedge E(y,z) \wedge E(x,z))$



$G_\phi = (V, E)$

$G' = (V', E')$

Canonical structures

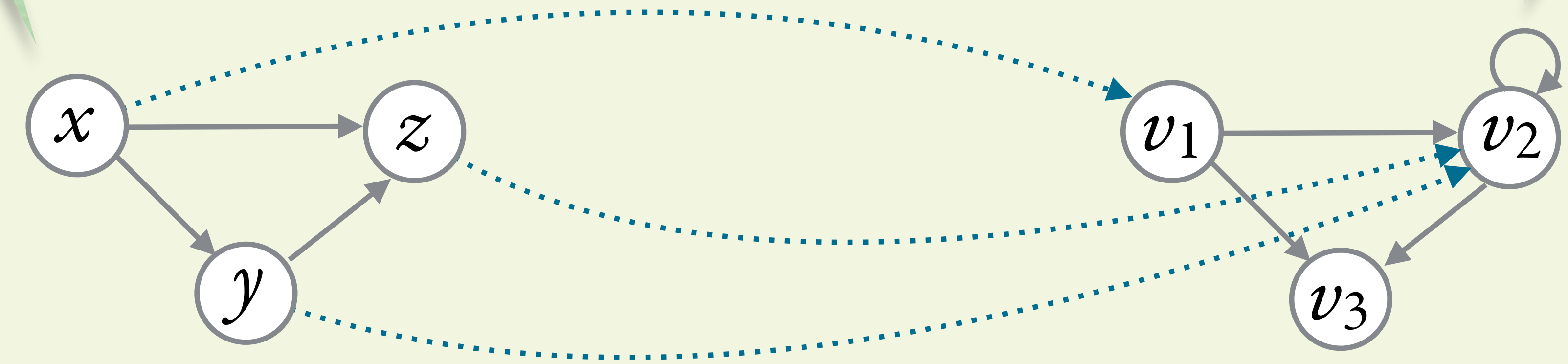
Canonical structure S_ϕ of a Conjunctive Query ϕ has
- variables as nodes
- tuples $(x_1, \ldots, x_n) \in R_i$ for all atomic sub-formulas $R_i(x_1, \ldots, x_n)$ of ϕ

Fact 1: $G_\phi \vDash \phi$

Fact 3: $G" \vDash \phi$ implies $\exists$ h: $G_\phi \rightarrow G"$

Fact 2: $h(G_\phi) \vDash \phi$

E.g.: $\phi = \exists x \, \exists y \, \exists z \, . \, (E(x,y) \wedge E(y,z) \wedge E(x,z))$



$G_\phi = (V, E)$

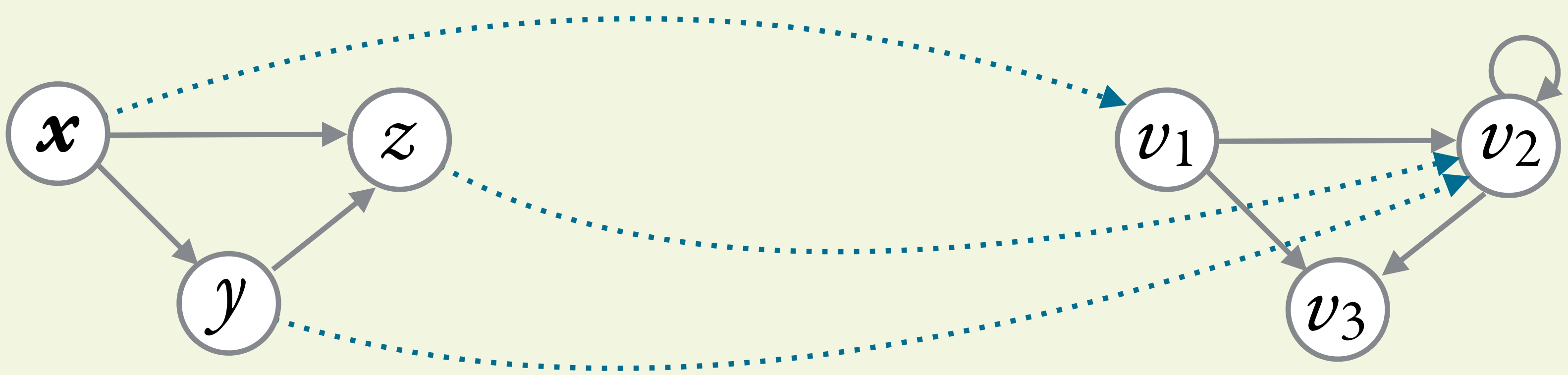
$G' = (V', E')$

Evaluation via homomorphisms

Lemma. The evaluation of a CQ $\phi(x_1, \ldots, x_n)$ on S' returns the set

$$\phi(S') = \{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S' \}$$

E.g.: $\phi(x) = \exists y \, \exists z \, . \, (E(x,y) \wedge E(y,z) \wedge E(x,z))$



$G_\phi = (V, E)$ $\qquad$ $G' = (V', E')$

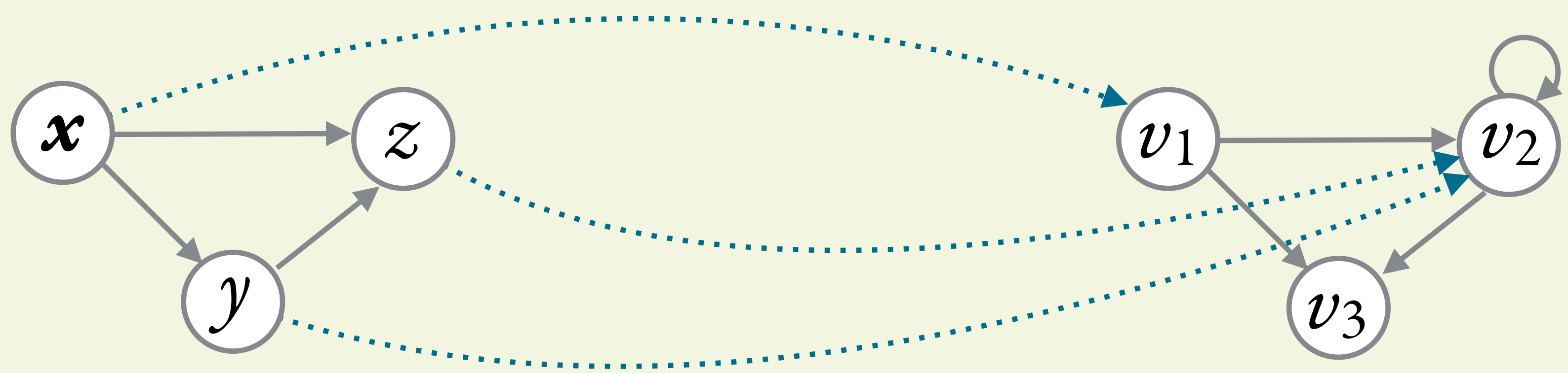
Evaluation via homomorphisms

Lemma. The evaluation of a CQ $\phi(x_1, \ldots, x_n)$ on S' returns the set

$$\phi(S') = \{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S' \}$$

Question: Which are the homomorphisms $G_\phi \longrightarrow G'$? What is the result of $\phi(G')$?

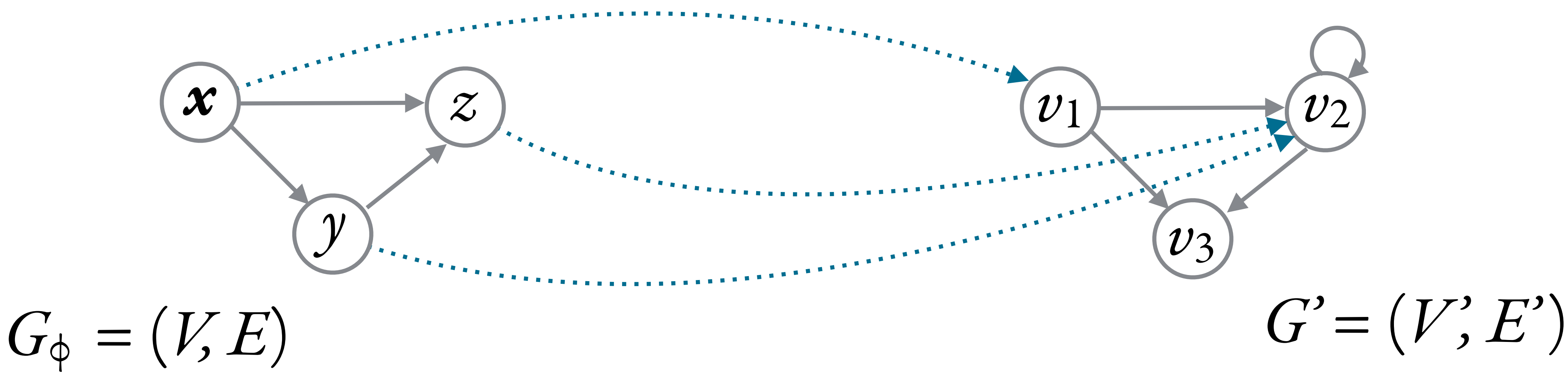
E.g.: $\phi(x) = \exists y \, \exists z \, . \, (E(x,y) \wedge E(y,z) \wedge E(x,z))$



$G_\phi = (V, E)$ $\qquad$ $G' = (V', E')$

Evaluation via homomorphisms

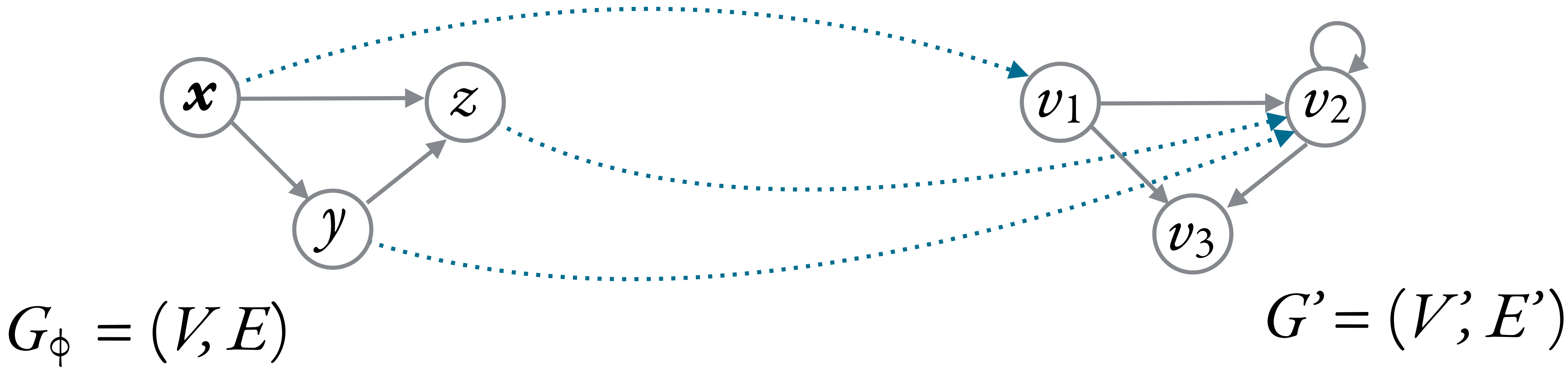
Theorem. Evaluation of CQ is in NP (combined complexity)



Input: A CQ $\phi(x_1, \ldots, x_n)$, a graph G, a tuple $(a_1, \ldots, a_n)$
Output: Is $(a_1, \ldots, a_n) \in \phi(G)$?

Evaluation via homomorphisms

Theorem. Evaluation of CQ is in NP (combined complexity)



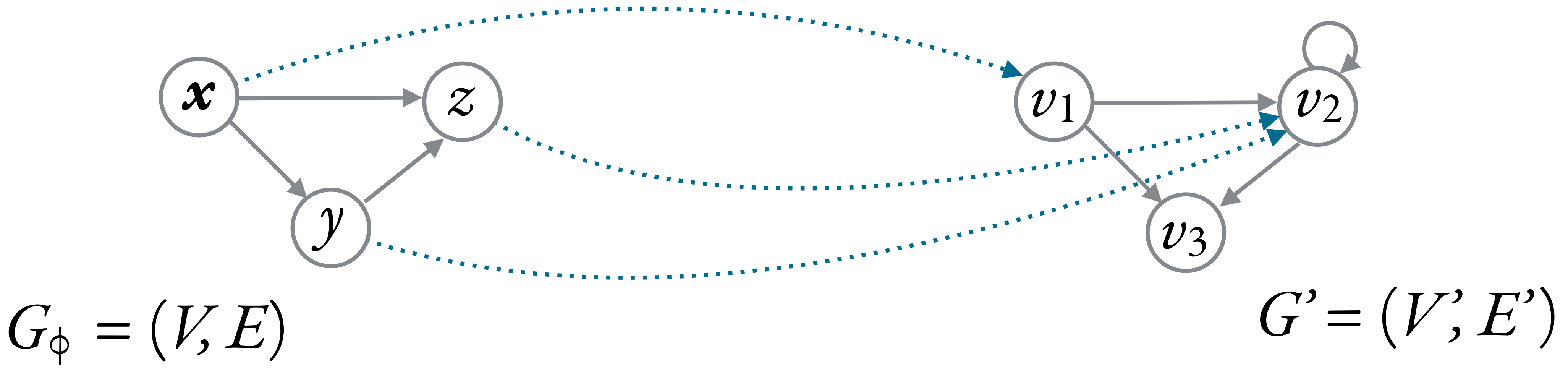
Input: A CQ $\phi(x_1, \ldots, x_n)$, a graph G, a tuple $(a_1, \ldots, a_n)$

Output: Is $(a_1, \ldots, a_n) \in \phi(G)$?

Ideas?

Evaluation via homomorphisms

Theorem. Evaluation of CQ is in NP (combined complexity)



Input: A CQ $\phi(x_1, \ldots, x_n)$, a graph G, a tuple $(a_1, \ldots, a_n)$

Output: Is $(a_1, \ldots, a_n) \in \phi(G)$?

Ideas?

1. Guess h: $G_\phi \longrightarrow G$
2. Check that it is a homomorphism
3. Output YES if $(h(x_1), \ldots, h(x_n)) = (a_1, \ldots, a_n)$; NO otherwise.

Evaluation via homomorphisms

Theorem. Evaluation of CQ is NP-complete (combined complexity)

Evaluation via homomorphisms

Theorem. Evaluation of CQ is NP-complete (combined complexity)

NP-complete problem: **3-COLORABILITY**

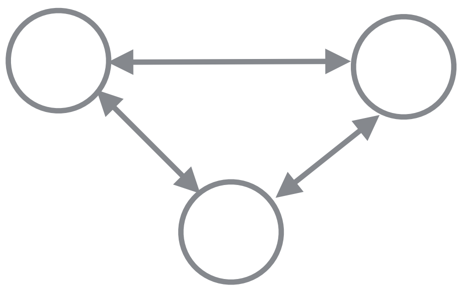
Input: A graph G

Output: Can we assign a colour from $\{R,G,B\}$ to each node so that adjacent nodes have always different colours ?

=

Is there a *homomorphism* from G to K_3 ?

K_3



Evaluation via homomorphisms

Theorem. Evaluation of CQ is NP-complete (combined complexity)

NP-complete problem: **3-COLORABILITY**

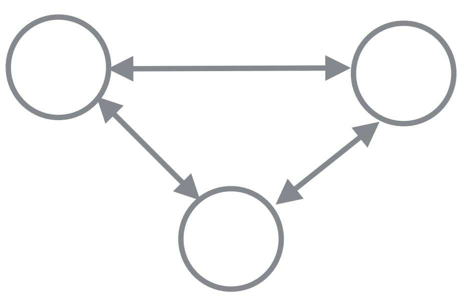
Input: A graph G

Output: Can we assign a colour from $\{R,G,B\}$ to each node so that adjacent nodes have always different colours ?

=

Is there a *homomorphism* from G to K_3 ?

K_3



Reduction 3COL ⇝ CQ-EVAL: 1. Given G, build a CQ ϕ such that $G_\phi = G$.
2. Test if $() \in \phi(G)$.

Monotonicity and preservation theorems

Lemma. Every CQ is **monotone**:

$S \subseteq S'$ implies $\phi(S) \subseteq \phi(S')$

Monotonicity and preservation theorems

Lemma. Every CQ is **monotone**:

$S \subseteq S'$ implies $\phi(S) \subseteq \phi(S')$

Proof: by closure under homomorphisms.

Monotonicity and preservation theorems

Lemma. Every CQ is **monotone**:

$S \subseteq S'$ implies $\phi(S) \subseteq \phi(S')$

Proof: by closure under homomorphisms.

- "The relation R has at most 2 elements" $\notin$ CQ
- "There is a node connected to every other node" $\notin$ CQ
- "The radius of the graph is 5" $\notin$ CQ

Monotonicity and preservation theorems

Theorem. If an FO query ϕ is **monotone**
then $\phi \in$ UCQ [Rossman '08]

Monotonicity and preservation theorems

Theorem. If an FO query ϕ is **monotone**

then $\phi \in \mathrm{UCQ}$ [Rossman '08]

Finite unions of CQs

$\approx$ $\exists \wedge \vee$ fragment of FO

Monotonicity and preservation theorems

Theorem. If an FO query ϕ is **monotone**

then $\phi \in$ UCQ [Rossman '08]

Finite unions of CQs

$\approx$ $\exists \wedge \vee$ fragment of FO

Equally expressive, but
UCQ are less succinct

Monotonicity and preservation theorems

Theorem. If an FO query ϕ is **monotone** then $\phi \in$ UCQ [Rossman '08]

Finite unions of CQs

$\approx$ $\exists \wedge \vee$ fragment of FO

Equally expressive, but
UCQ are less succinct

- One example of the few properties which still hold on finite structures.
- Proof in the finite is difficult and independent.

Static analysis with CQs

The satisfiability problem for CQ is decidable…

Question: What is the algorithm for CQ-SAT? What is the complexity?

Static analysis with CQs

The satisfiability problem for CQ is decidable…

Question: What is the algorithm for CQ-SAT? What is the complexity?

Answer: CQs are always satisfiable by their canonical structure!

$$G_\phi \models \phi$$

Static analysis with CQs

problem: **CQ-CONTAINMENT**

Input: Two CQs ϕ, ψ

Output: Does $\phi(S) \subseteq \psi(S)$ holds for every structure S ?

Static analysis with CQs

problem: **CQ-CONTAINMENT**

Input: Two CQs ϕ, ψ

Output: Does $\phi(S) \subseteq \psi(S)$ holds for every structure S ?

Theorem. The containment problem for CQ is NP-complete

Static analysis with CQs

problem: **CQ-CONTAINMENT**

Input: Two CQs ϕ, ψ
Output: Does $\phi(S) \subseteq \psi(S)$ holds for every structure S ?

Theorem. The containment problem for CQ is NP-complete

Question: Is this combined or data complexity?

Static analysis with CQs

problem: **CQ-CONTAINMENT**

Input: Two CQs ϕ, ψ

Output: Does $\phi(S) \subseteq \psi(S)$ holds for every structure S ?

Theorem. The containment problem for CQ is NP-complete

Question: Is this combined or data complexity? **Answer:** None!

Static analysis with CQs

problem: **CQ-CONTAINMENT**

Input: Two CQs ϕ, ψ
Output: Does $\phi(S) \subseteq \psi(S)$ holds for every structure S ?

Theorem. The containment problem for CQ is NP-complete

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

Question: Is this combined or data complexity? **Answer:** None!

Static analysis with CQs

problem: **CQ-CONTAINMENT**

Input: Two CQs ϕ, ψ

Output: Does $\phi(S) \subseteq \psi(S)$ holds for every structure S ?

Theorem. The containment problem for CQ is NP-complete

$\phi(x_1, \dots, x_n)$ is contained in $\psi(y_1, \dots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

Why?

Question: Is this combined or data complexity? **Answer:** None!

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g \colon S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g \colon S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

[$\Longrightarrow$] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

$\{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S \}$

[$\Longrightarrow$] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

$\{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S \}$

[⟹] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

$\{ (g(y_1), \ldots, g(y_n)) \mid g : S_\psi \longrightarrow S \}$

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

$\{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S \}$

[⟹] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

$\{ (g(y_1), \ldots, g(y_n)) \mid g : S_\psi \longrightarrow S \}$

If there is $h: S_\phi \longrightarrow S$

Then there is $g: S_\psi \longrightarrow S$ such that $h(x_1, \ldots, x_n) = g(y_1, \ldots, y_m)$

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

$\{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S \}$

[⟹] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

$\{ (g(y_1), \ldots, g(y_n)) \mid g : S_\psi \longrightarrow S \}$

If there is $h: S_\phi \longrightarrow S$

Then there is $g: S_\psi \longrightarrow S$ such that $h(x_1, \ldots, x_n) = g(y_1, \ldots, y_m)$

Take $S = S_\phi$ and $h = \textit{identity}$.

Static analysis with CQs

$\phi(x_1, \dots, x_n)$ is contained in $\psi(y_1, \dots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

$\{\, (h(x_1), \dots, h(x_n)) \mid h : S_\phi \longrightarrow S \,\}$

[⟹] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

$\{\, (g(y_1), \dots, g(y_n)) \mid g : S_\psi \longrightarrow S \,\}$

If there is $h: S_\phi \longrightarrow S$

Then there is $g: S_\psi \longrightarrow S$ such that $h(x_1, \dots, x_n) = g(y_1, \dots, y_m)$

Take $S = S_\phi$ and $h = \textit{identity}$.

[⟸] Consider $(v_1,\dots,v_m) \in \phi(S)$

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

$\{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S \}$

[⟹] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

$\{ (g(y_1), \ldots, g(y_n)) \mid g : S_\psi \longrightarrow S \}$

If there is $h: S_\phi \longrightarrow S$

Then there is $g: S_\psi \longrightarrow S$ such that $h(x_1, \ldots, x_n) = g(y_1, \ldots, y_m)$

Take $S = S_\phi$ and $h = \textit{identity}$.

$(v_1, \ldots, v_m) = (h(x_1), \ldots, h(x_n))$
for some $h : S_\phi \longrightarrow S$

[⟸] Consider $(v_1, \ldots, v_m) \in \phi(S)$

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

$\{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S \}$

[⟹] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

$\{ (g(y_1), \ldots, g(y_n)) \mid g : S_\psi \longrightarrow S \}$

If there is $h: S_\phi \longrightarrow S$

Then there is $g: S_\psi \longrightarrow S$ such that $h(x_1, \ldots, x_n) = g(y_1, \ldots, y_m)$

Take $S = S_\phi$ and $h = \textit{identity}$.

$(v_1, \ldots, v_m) = (h(x_1), \ldots, h(x_n))$

for some $h: S_\phi \longrightarrow S$

[⟸] Consider $(v_1, \ldots, v_m) \in \phi(S)$

Since $g(y_1, \ldots, y_m) = (x_1, \ldots, x_n)$, then $(v_1, \ldots, v_m) = h(x_1, \ldots, x_n) = h(g(y_1, \ldots, y_m))$

Static analysis with CQs

$\phi(x_1, \ldots, x_n)$ is contained in $\psi(y_1, \ldots, y_m)$ iff

1. $n = m$
2. There is $g: S_\psi \longrightarrow S_\phi$
3. $g(y_i) = x_i$ for all i

$\{ (h(x_1), \ldots, h(x_n)) \mid h : S_\phi \longrightarrow S \}$

[$\Longrightarrow$] Suppose $\forall S \quad \phi(S) \subseteq \psi(S)$

$\{ (g(y_1), \ldots, g(y_n)) \mid g : S_\psi \longrightarrow S \}$

If there is $h: S_\phi \longrightarrow S$

Then there is $g: S_\psi \longrightarrow S$ such that $h(x_1, \ldots, x_n) = g(y_1, \ldots, y_m)$

Take $S = S_\phi$ and $h = \mathit{identity}$.

$(v_1, \ldots, v_m) = (h(x_1), \ldots, h(x_n))$ for some $h : S_\phi \longrightarrow S$

[$\Longleftarrow$] Consider $(v_1, \ldots, v_m) \in \phi(S)$

Since $g(y_1, \ldots, y_m) = (x_1, \ldots, x_n)$, then $(v_1, \ldots, v_m) = h(x_1, \ldots, x_n) = h(g(y_1, \ldots, y_m))$

$h\,g$ is a homomorphism from S_ψ to S. Hence, $(v_1, \ldots, v_m) \in \psi(G)$.

Static analysis with CQs

problem: **CQ-EQUIVALENCE**

Input: Two CQs ϕ, ψ

Output: Does $\phi(S) = \psi(S)$ holds for every S ? (we write "$\phi \equiv \psi$")

Static analysis with CQs

problem: **CQ-EQUIVALENCE**

Input: Two CQs ϕ, ψ
Output: Does $\phi(S) = \psi(S)$ holds for every S ? (we write "$\phi \equiv \psi$")

Theorem. The equivalence problem for CQ is NP-complete

Static analysis with CQs

problem: **CQ-EQUIVALENCE**

Input: Two CQs ϕ, ψ
Output: Does $\phi(S) = \psi(S)$ holds for every S ? (we write "$\phi \equiv \psi$")

Theorem. The equivalence problem for CQ is NP-complete

Amounts to testing if G_ϕ and G_ψ are **hom-equivalent**
(i.e. there are homomorphisms in both senses)

Static analysis with CQs

problem: **CQ-MINIMIZATION**

Input: A CQ ϕ

Output: Is there a "smaller" CQ ψ such that $\psi \equiv \phi$?

Static analysis with CQs

problem: **CQ-MINIMIZATION**

Input: A CQ ϕ

Output: Is there a "smaller" CQ ψ such that $\psi \equiv \phi$?

Theorem. The minimization problem for CQ is NP-complete

Static analysis with CQs

problem: **CQ-MINIMIZATION**

Input: A CQ ϕ
Output: Is there a "smaller" CQ ψ such that $\psi \equiv \phi$?

Theorem. The minimization problem for CQ is NP-complete

Amounts to testing if there is a **non-injective endomorphism**

$g\colon G_\phi \longrightarrow G_\phi$

or, equally, if the **smallest graph hom-equivalent** to G_ϕ is G_ϕ itself
(we say that G_ϕ is a **core**)

Adding functional dependencies

A **functional dependency** is a sentence of the form

$$\boldsymbol{\gamma} = \forall \ldots\ R(x_1, \ldots, x_n) \wedge R(x_1', \ldots, x_n') \wedge \bigwedge_j (x_{ij} = x_{ij}') \Rightarrow (x_i = x_i')$$

Adding functional dependencies

A **functional dependency** is a sentence of the form

$$\boldsymbol{\gamma} = \forall \ldots \ R(x_1, \ldots, x_n) \wedge R(x_1', \ldots, x_n') \wedge \bigwedge_j (x_{ij} = x_{ij}') \Rightarrow (x_i = x_i')$$

Example: In the following relation we may enforce the functional dependency

$$\boldsymbol{\gamma} = \forall\, x, y, z, x', y', z' \ R(x, y, z) \wedge R(x', y', z') \wedge (x = x') \Rightarrow (y = y')$$

Agent	Name	Drives
007	James Bond	Aston Martin
200	Mr Smith	Cadillac
201	Mrs Smith	Mercedes
3	Jason Bourne	BMW

Adding functional dependencies

A **functional dependency** is a sentence of the form

$$\boldsymbol{\gamma} \;=\; \forall \ldots\; R(x_1, \ldots, x_n) \wedge R(x_1', \ldots, x_n') \wedge \bigwedge_j (x_{ij} = x_{ij}') \Rightarrow (x_i = x_i')$$

Example: In the following relation we may enforce the functional dependency

$$\boldsymbol{\gamma} \;=\; \forall\, x, y, z, x', y', z'\;\; R(x, y, z) \wedge R(x', y', z') \wedge (x = x') \Rightarrow (y = y')$$

Agent	Name	Drives
007	James Bond	Aston Martin
200	Mr Smith	Cadillac
201	Mrs Smith	Mercedes
3	Jason Bourne	BMW

We often abbreviate this with

R: $1 \rightarrow 2$

Adding functional dependencies

A **functional dependency** (FD) is a sentence of the form

$$F = \forall \ldots \; R(x_1, \ldots, x_n) \wedge R(x_1', \ldots, x_n') \wedge \bigwedge_j (x_{ij} = x_{ij}') \Rightarrow (x_i = x_i')$$

All the previous problems:

- CQ-CONTAINMENT
- CQ-EQUIVALENCE
- CQ-MINIMIZATION

remain in NP if we further restrict finite structures so as to satisfy any set of functional dependencies

Adding functional dependencies

A **functional dependency** (FD) is a sentence of the form

$$F = \forall \ldots\ R(x_1, \ldots, x_n) \wedge R(x_1', \ldots, x_n') \wedge \bigwedge_j (x_{ij} = x_{ij}') \Rightarrow (x_i = x_i')$$

All the previous problems:

Modify the canonical structure S_ϕ ...

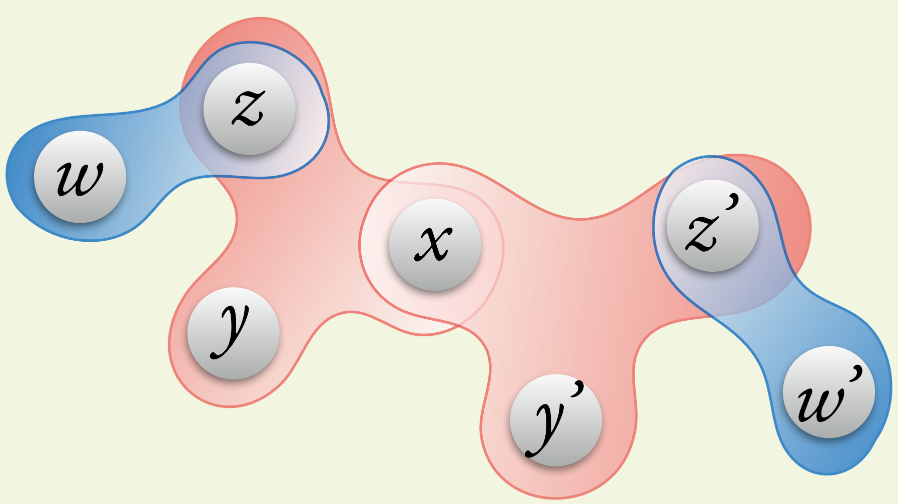
- CQ-CONTAINMENT
- CQ-EQUIVALENCE
- CQ-MINIMIZATION

remain in NP if we further restrict finite structures so as to satisfy any set of functional dependencies

Adding functional dependencies

CQ $\phi = R_2(x, y, z) \wedge R_2(x, y', z') \wedge R_1(z, w) \wedge R_1(z', w')$

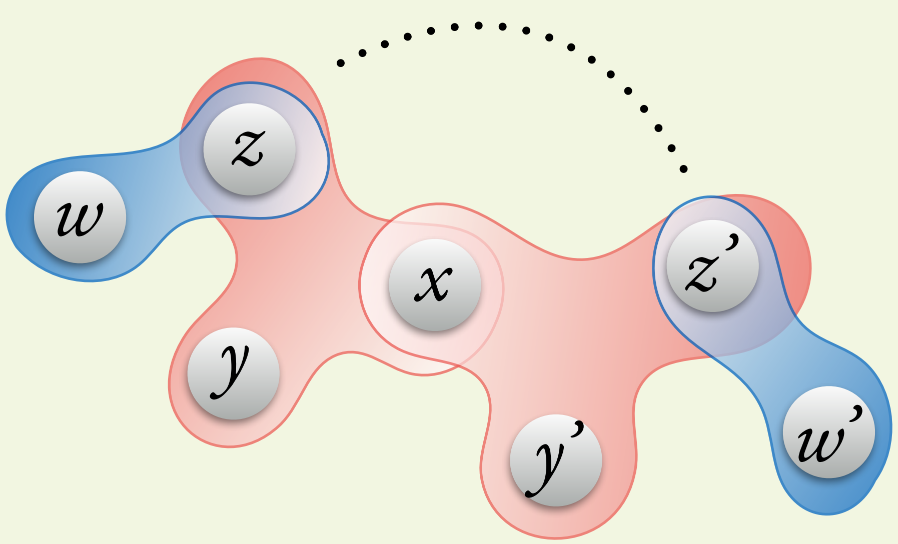
under functional dependencies $F = \{ R_1: 1 \rightarrow 2,\ R_2: 1 \rightarrow 3 \}$



Adding functional dependencies

CQ $\phi = R_2(x, y, z) \wedge R_2(x, y', z') \wedge R_1(z, w) \wedge R_1(z', w')$

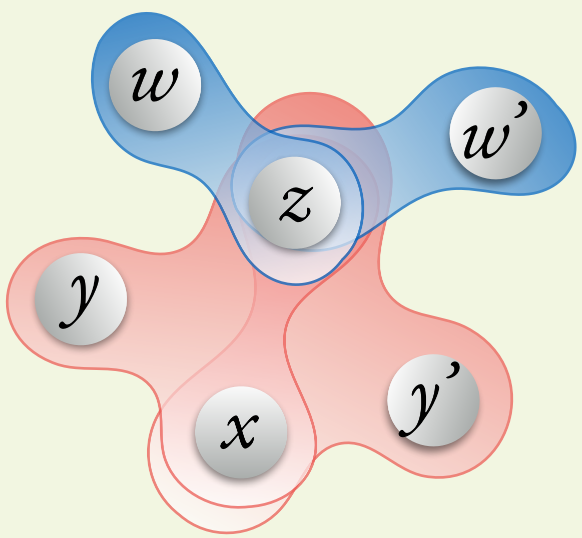
under functional dependencies $F = \{ R_1 \colon 1 \rightarrow 2,\ R_2 \colon 1 \rightarrow 3 \}$



Adding functional dependencies

CQ $\phi = R_2(x, y, z) \wedge R_2(x, y', z') \wedge R_1(z, w) \wedge R_1(z', w')$

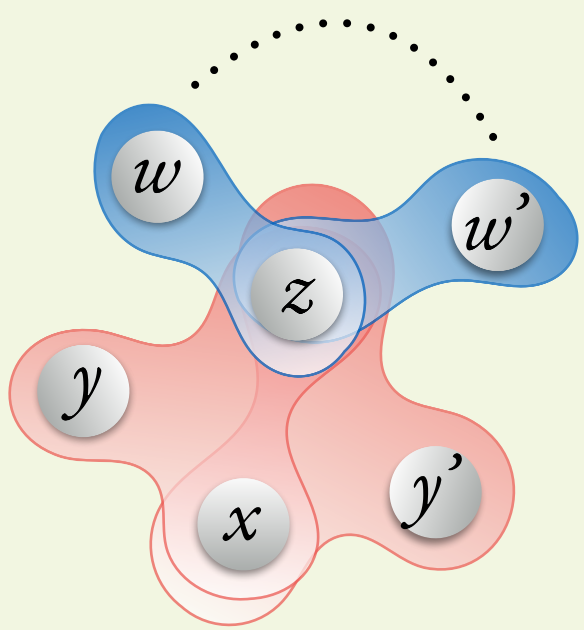
under functional dependencies $F = \{ R_1 \colon 1 \rightarrow 2,\ R_2 \colon 1 \rightarrow 3 \}$



Adding functional dependencies

CQ $\phi = R_2(x, y, z) \wedge R_2(x, y', z') \wedge R_1(z, w) \wedge R_1(z', w')$

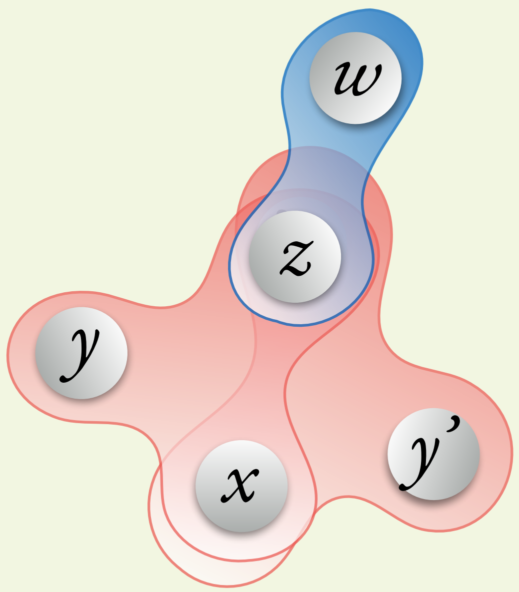
under functional dependencies $F = \{\, R_1\!: 1 \rightarrow 2,\ R_2\!: 1 \rightarrow 3 \,\}$



Adding functional dependencies

CQ $\phi = R_2(x, y, z) \wedge R_2(x, y', z') \wedge R_1(z, w) \wedge R_1(z', w')$

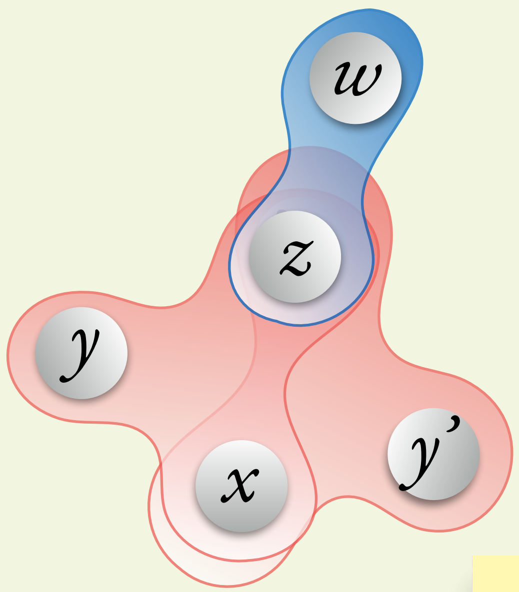
under functional dependencies $F = \{ R_1\colon 1 \rightarrow 2,\ R_2\colon 1 \rightarrow 3 \}$



Adding functional dependencies

CQ $\phi = R_2(x, y, z) \wedge R_2(x, y', z') \wedge R_1(z, w) \wedge R_1(z', w')$

under functional dependencies $F = \{ R_1\!: 1 \to 2,\ R_2\!: 1 \to 3 \}$



$= S_\phi$ (the **chased canonical structure**)

- S_ϕ is unique and can be constructed in polynomial time
- It is the "most general" constrained model of ϕ : $\phi(S) = h(\phi(S_\phi))$ for all S satisfying the funct. depend.

Adding functional dependencies

$\phi \in CQ$
FD's $F=\{fd_1, \ldots, fd_n\}$

chase →

$chase_F(\phi) \in CQ$

Adding functional dependencies

$\phi \in CQ$
FD's $F=\{fd_1, \ldots, fd_n\}$ —chase→ $chase_F(\phi) \in CQ$

The static analysis problems restricted to FD's can now be also shown in NP

- CQ-Containment $\phi \subseteq_F \psi$ iff $chase_F(\phi) \subseteq chase_F(\psi)$
- CQ-Equivalence $\phi \equiv_F \psi$ iff $chase_F(\phi) \equiv chase_F(\psi)$
- CQ-Minimization ϕ is minimal wrt structures verifying F iff $chase_F(\phi)$ is minimal