



Logical foundations of databases

Diego Figueira

Gabriele Puppis

CNRS LaBRI



Recap

- Acyclic Conjunctive Queries
- Join Trees
- Evaluation of ACQ (LOGCFL-complete)
- Ears, GYO algorithm for testing acyclicity
- Tree decomposition, tree-width of CQ
- Evaluation of bounded tree-width CQs (LOGCFL-complete)
- Bounded variable fragment of FO, evaluation in PTIME
- Acyclic Conjunctive Queries

Ehrenfeucht-Fraïssé games

S_1 and S_2 are
 n -equivalent!



Duplicator

No they're
NOT!!!!



Spoiler

Ehrenfeucht-Fraïssé games

S_1 and S_2 are
 n -equivalent!



Duplicator

No they're
NOT!!!!



Spoiler

They play for n rounds on the board (S_1, S_2) .

At each round i : **Spoiler** chooses a node x_i from S_1 (resp. y_i from S_2)

Duplicator answers with a node y_i from S_2 (resp. x_i from S_1)
trying to maintain an isomorphism between $S_1 \upharpoonright \{x_i\}_i$ and $S_2 \upharpoonright \{y_i\}_i$

Ehrenfeucht-Fraïssé games

On non-isomorphic *finite* structures, Spoiler wins eventually... **Why?**

Ehrenfeucht-Fraïssé games

On non-isomorphic *finite* structures, Spoiler wins eventually... **Why?**

...and he often wins very quickly:



$2^n - 1$ nodes

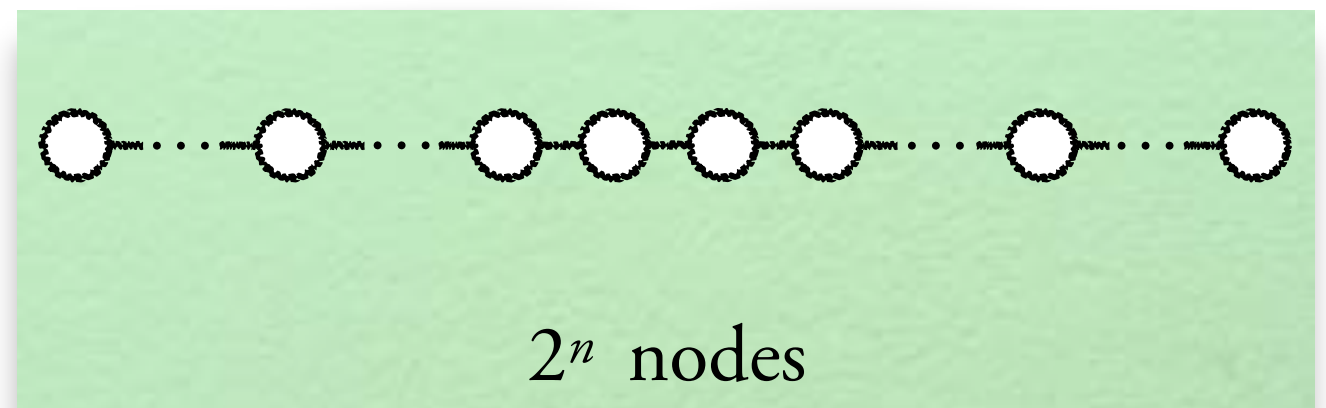
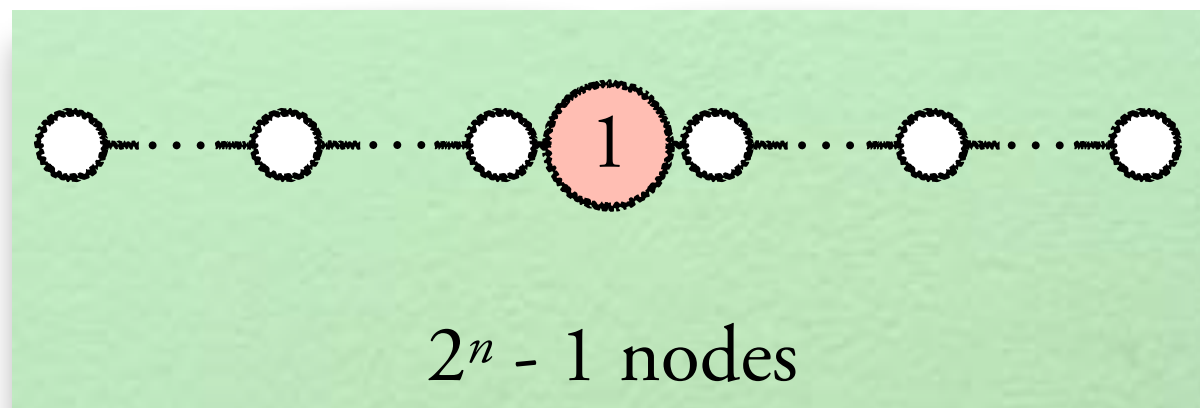


2^n nodes

Ehrenfeucht-Fraïssé games

On non-isomorphic *finite* structures, Spoiler wins eventually... **Why?**

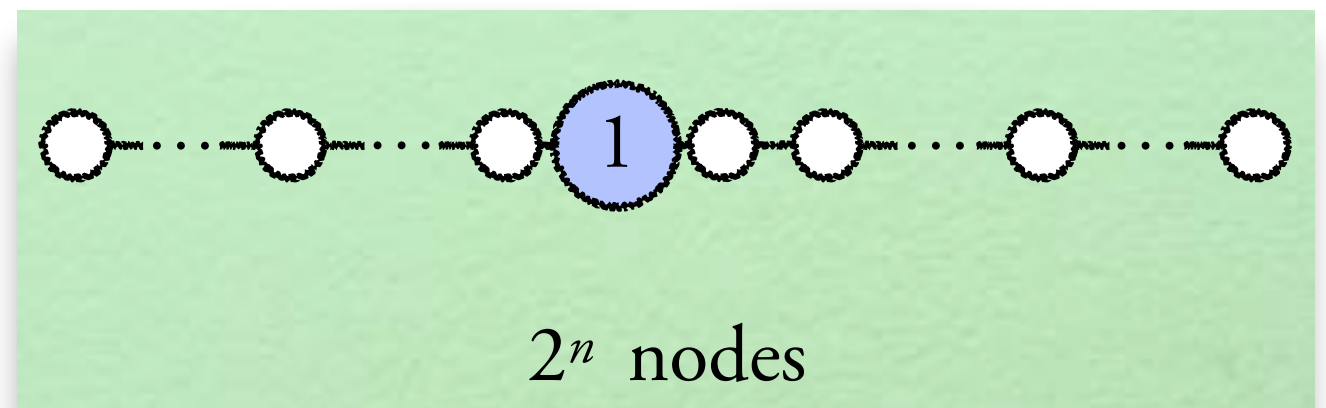
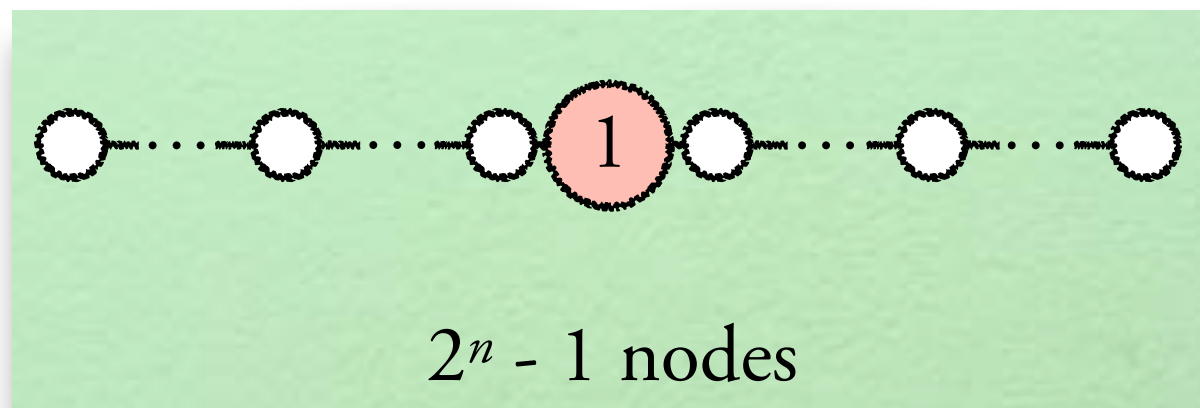
...and he often wins very quickly:



Ehrenfeucht-Fraïssé games

On non-isomorphic *finite* structures, Spoiler wins eventually... Why?

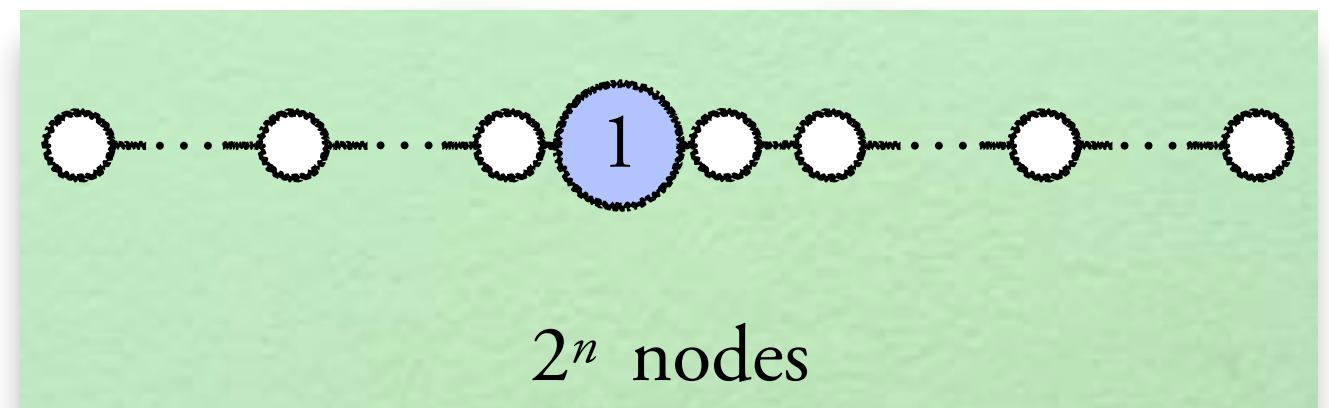
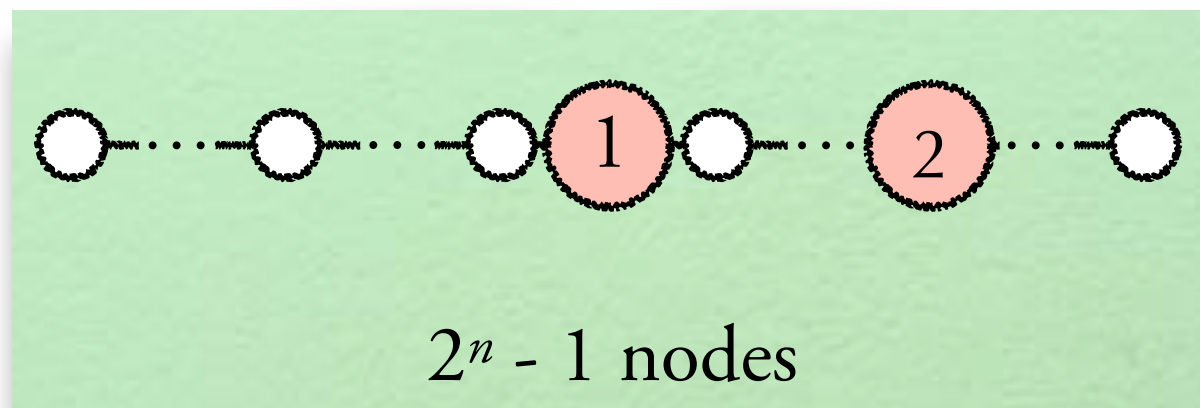
...and he often wins very quickly:



Ehrenfeucht-Fraïssé games

On non-isomorphic *finite* structures, Spoiler wins eventually... Why?

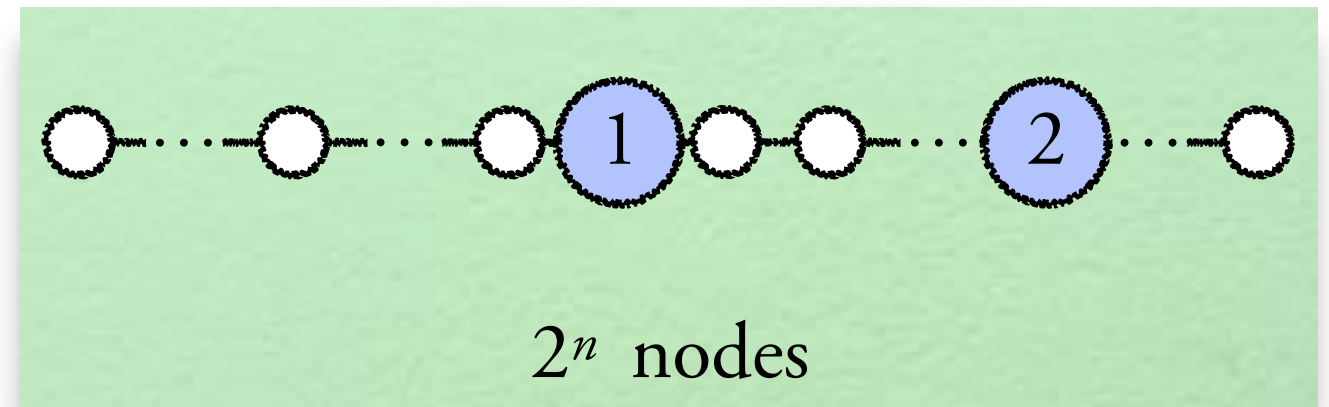
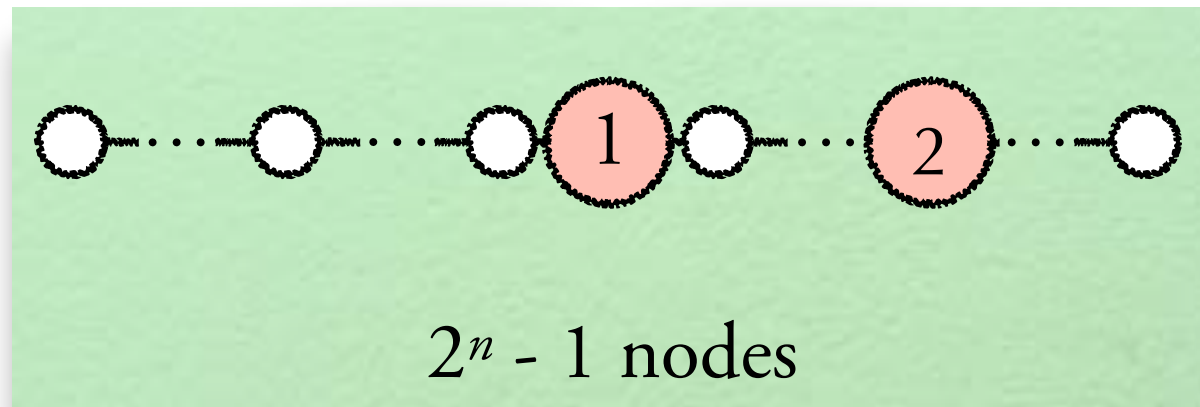
...and he often wins very quickly:



Ehrenfeucht-Fraïssé games

On non-isomorphic *finite* structures, Spoiler wins eventually... Why?

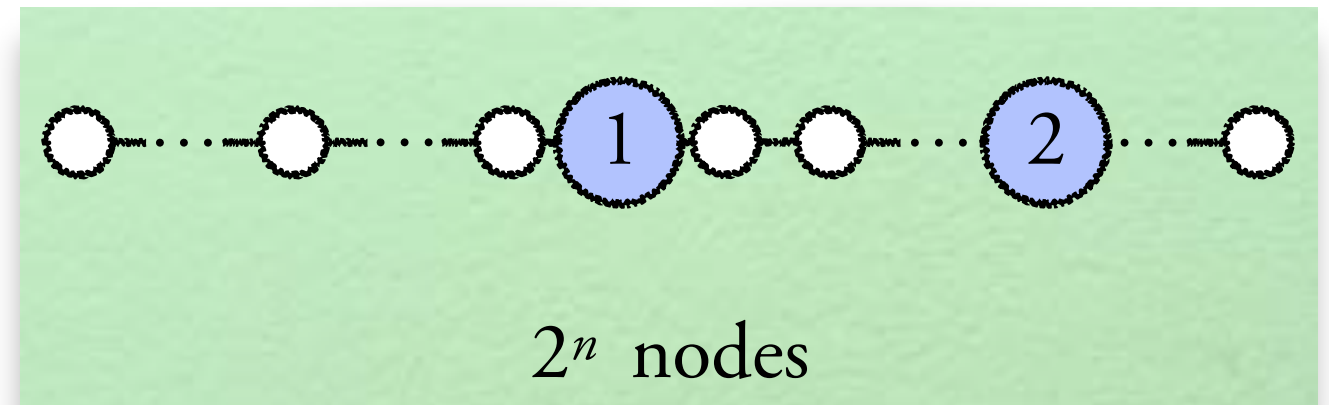
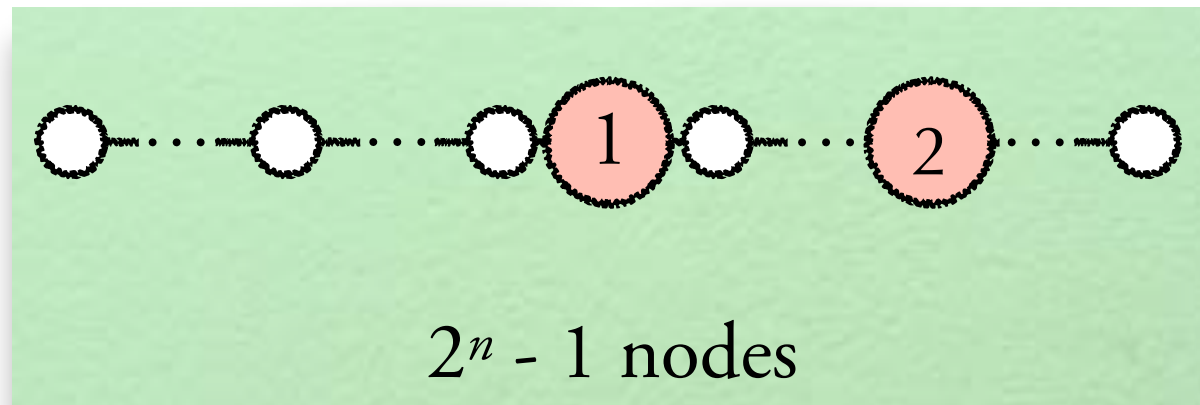
...and he often wins very quickly:



Ehrenfeucht-Fraïssé games

On non-isomorphic *finite* structures, Spoiler wins eventually... Why?

...and he often wins very quickly:



But there are non-isomorphic *infinite* structures where Duplicator can survive for *arbitrarily many rounds* (not necessarily forever!)

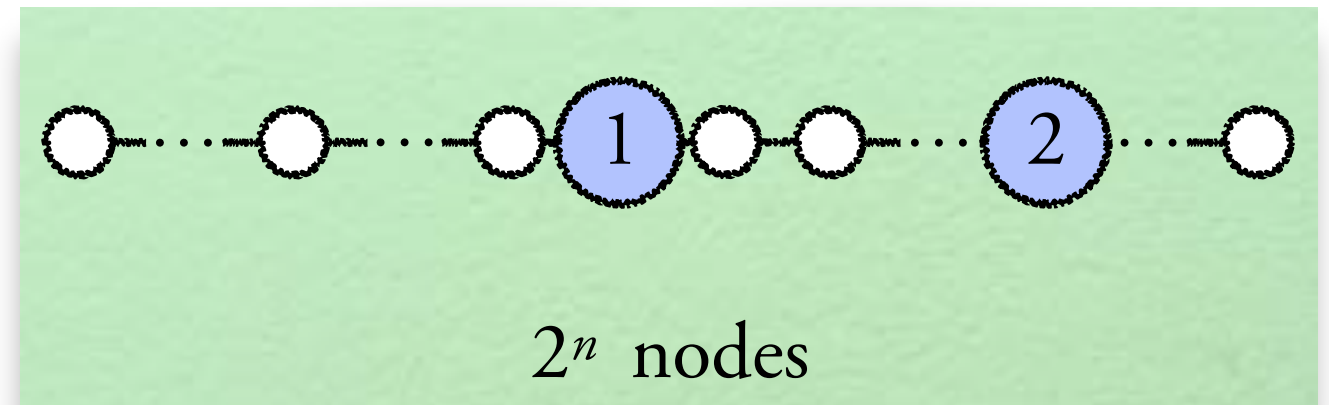
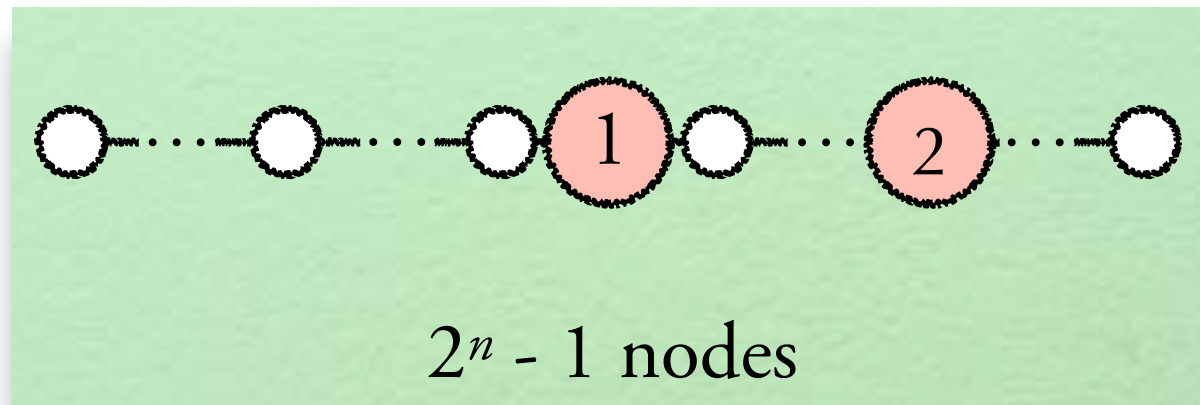
Any idea?

Ehrenfeucht-Fraïssé games

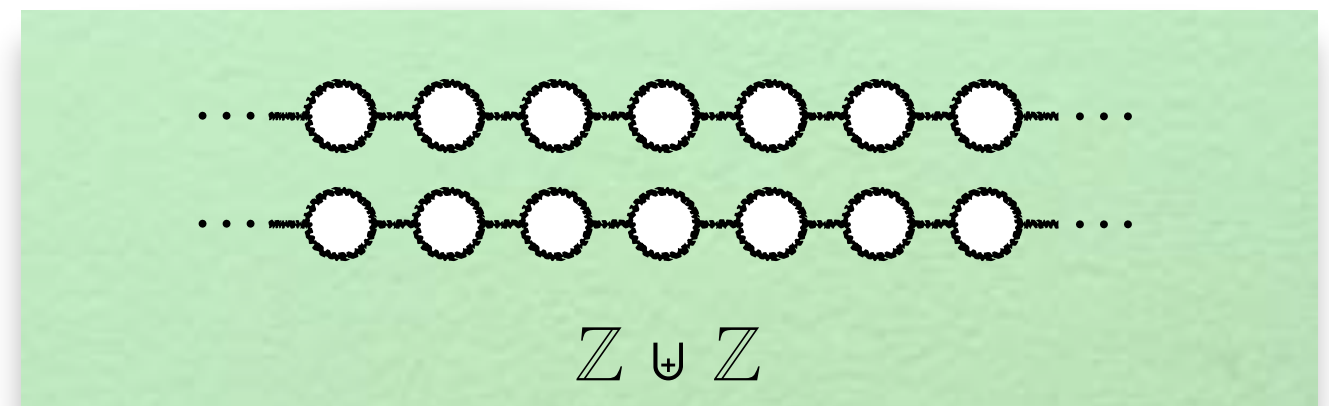
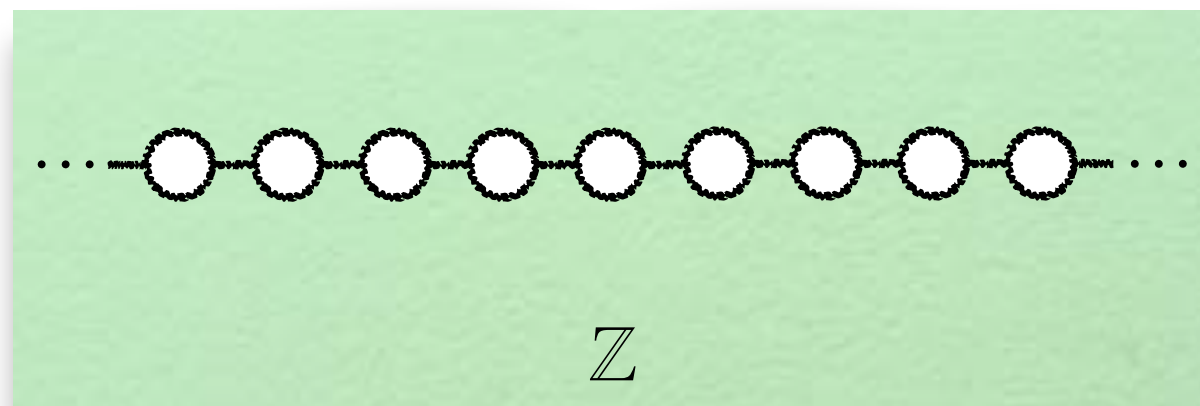
On non-isomorphic *finite* structures, Spoiler wins eventually...

Why?

...and he often wins very quickly:



But there are non-isomorphic *infinite* structures where Duplicator can survive for *arbitrarily many rounds* (not necessarily forever!)

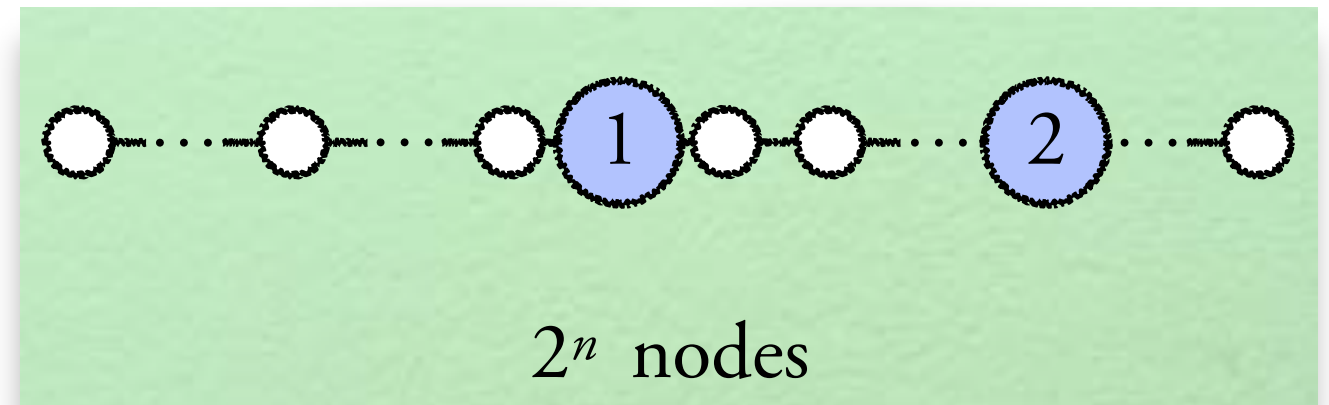
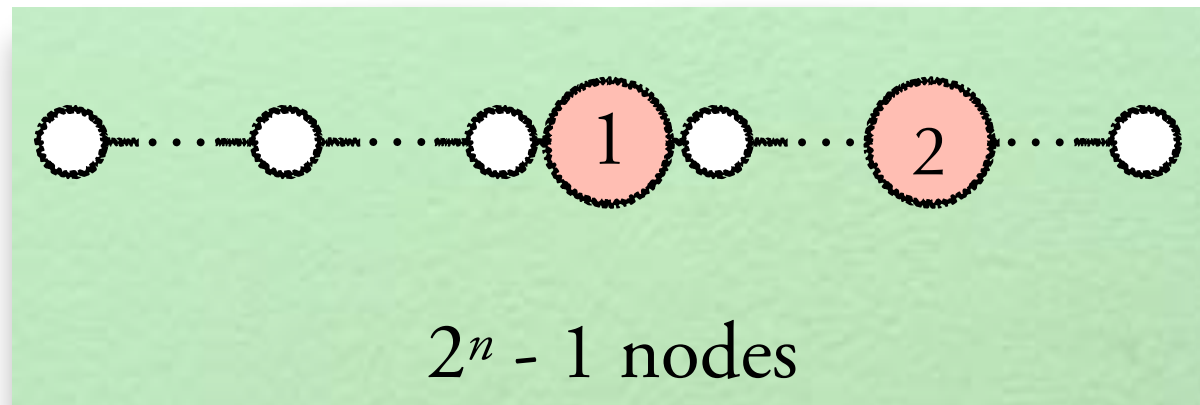


Ehrenfeucht-Fraïssé games

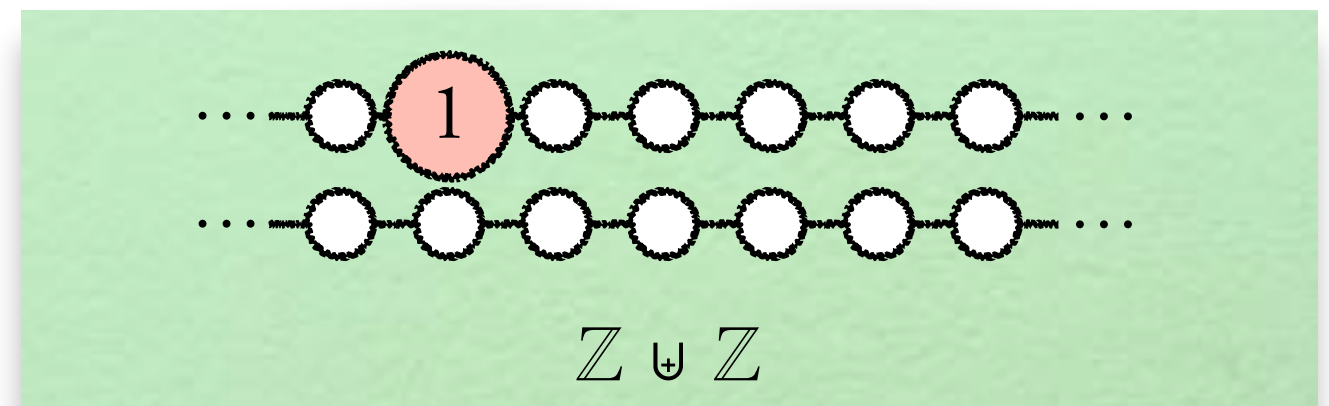
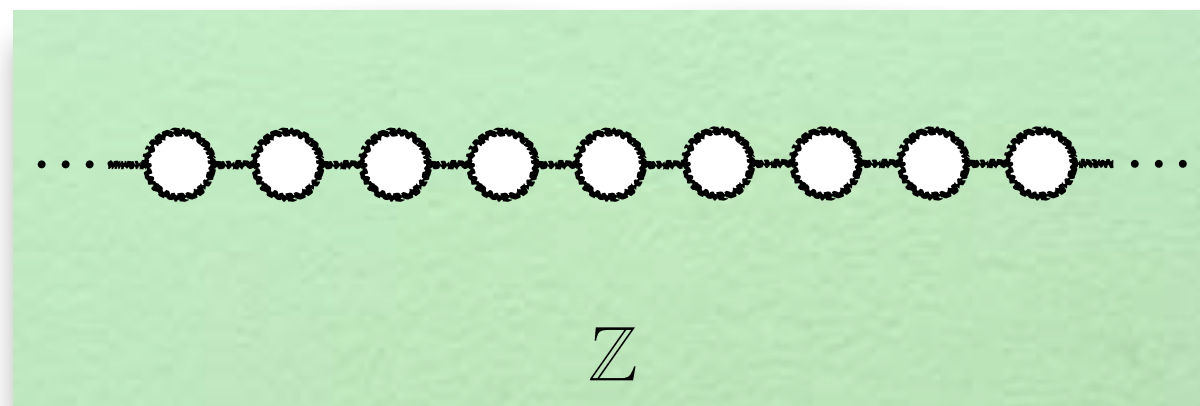
On non-isomorphic *finite* structures, Spoiler wins eventually...

Why?

...and he often wins very quickly:



But there are non-isomorphic *infinite* structures where Duplicator can survive for *arbitrarily many rounds* (not necessarily forever!)

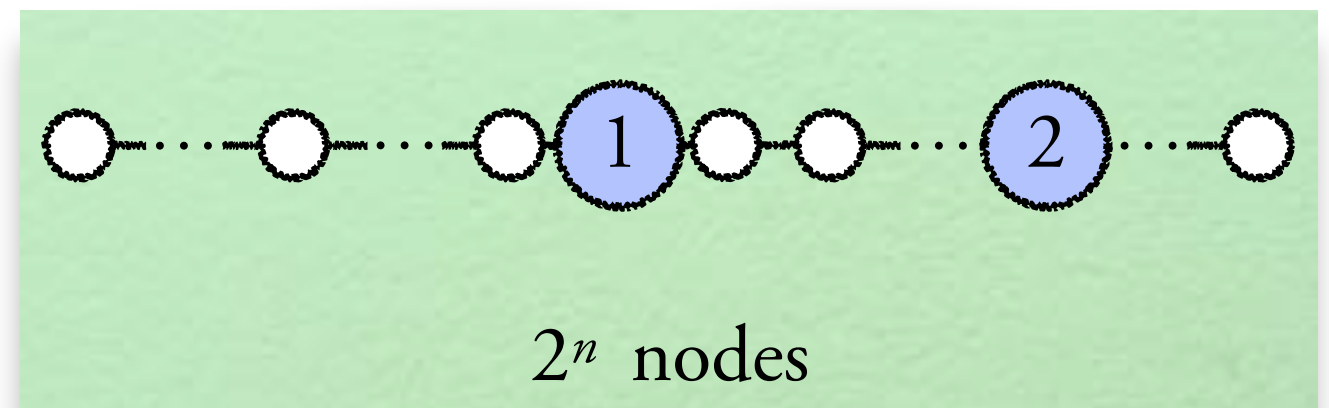
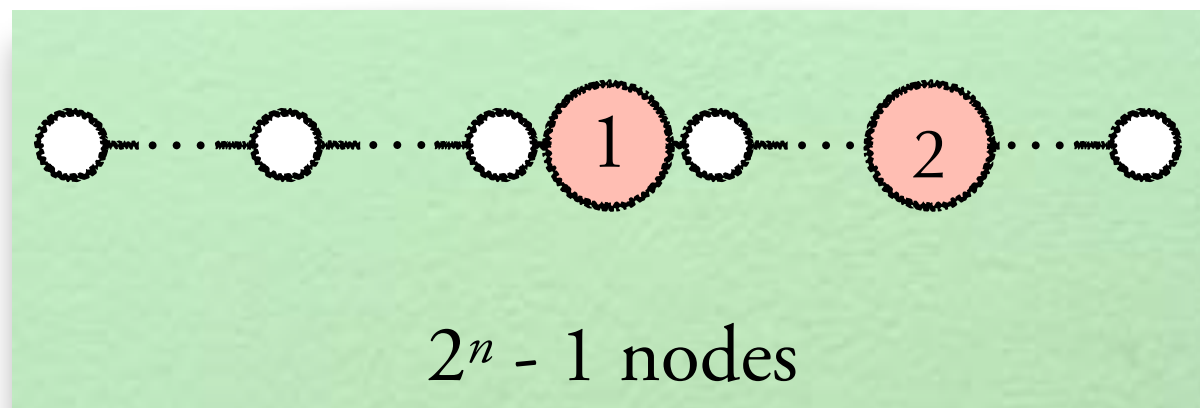


Ehrenfeucht-Fraïssé games

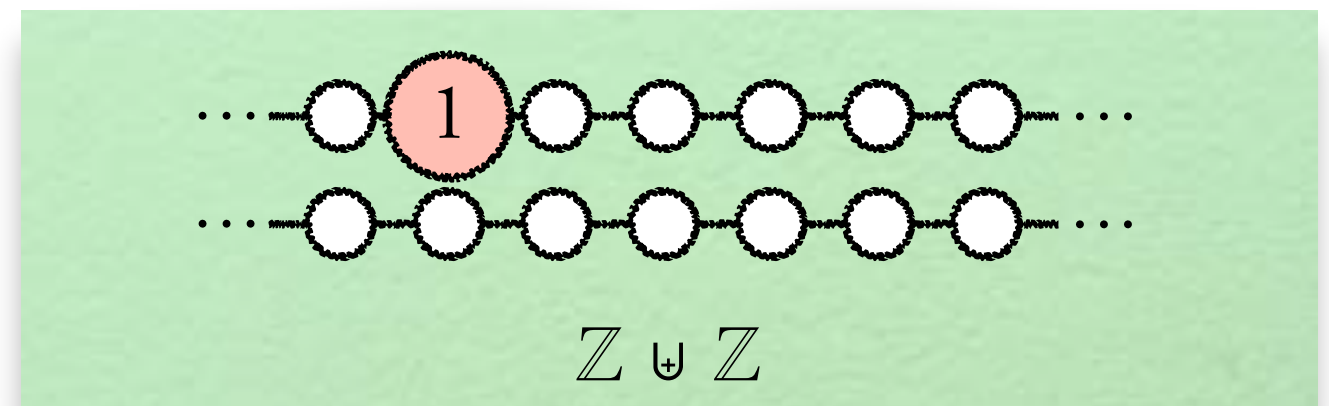
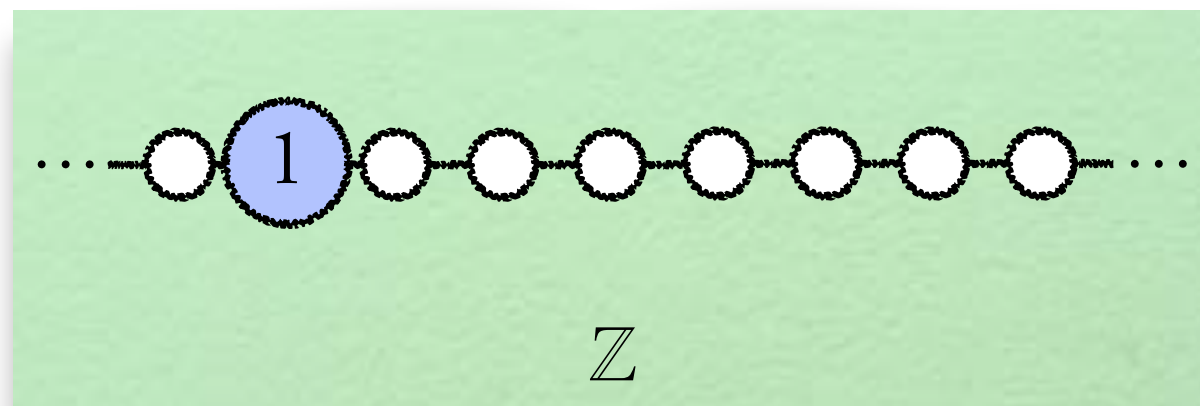
On non-isomorphic *finite* structures, Spoiler wins eventually...

Why?

...and he often wins very quickly:



But there are non-isomorphic *infinite* structures where Duplicator can survive for *arbitrarily many rounds* (not necessarily forever!)

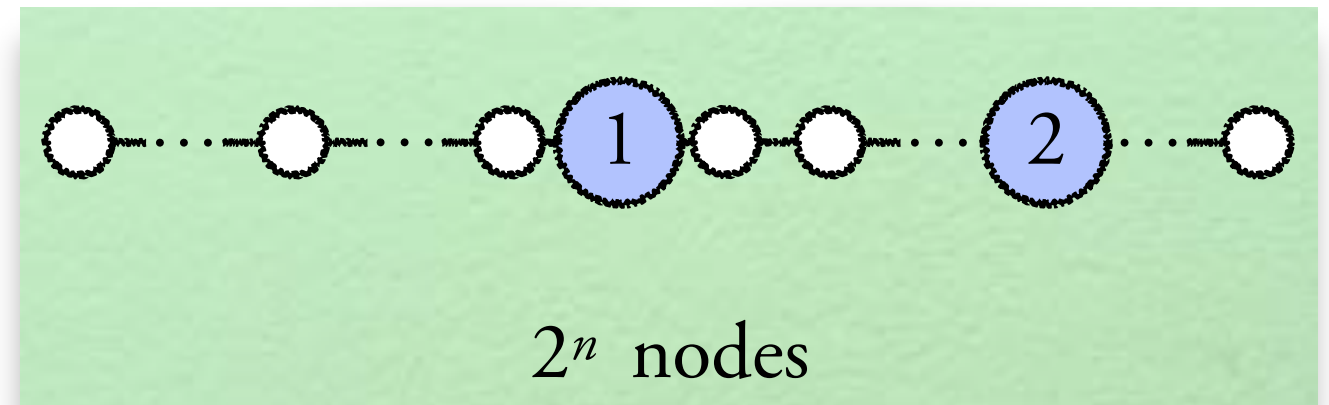
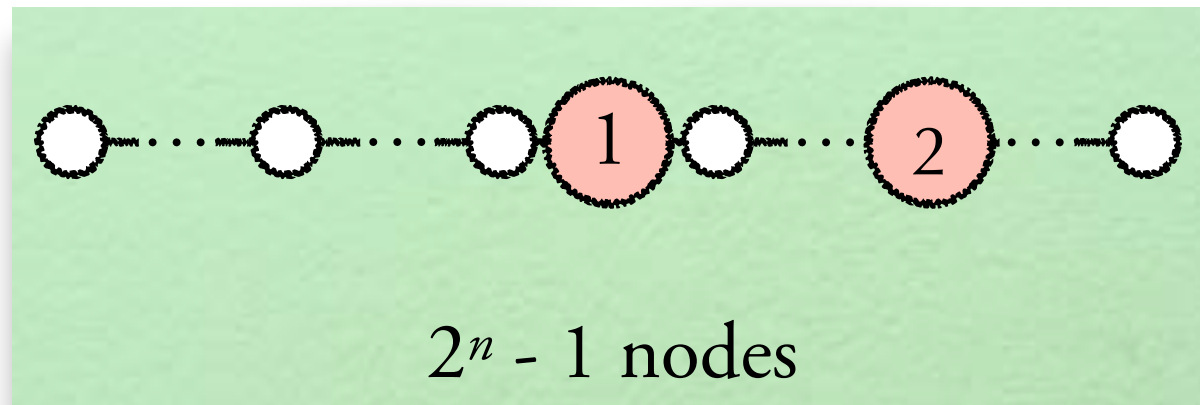


Ehrenfeucht-Fraïssé games

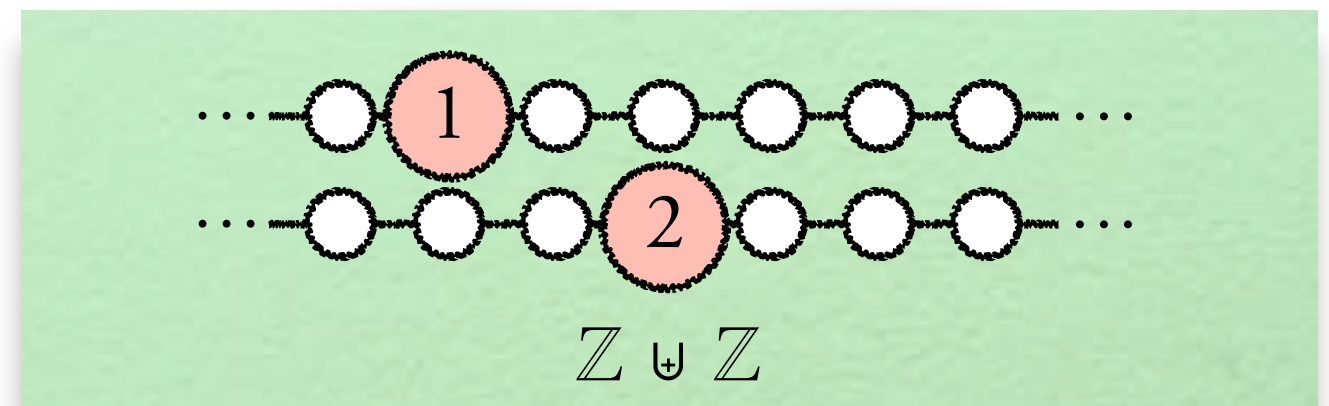
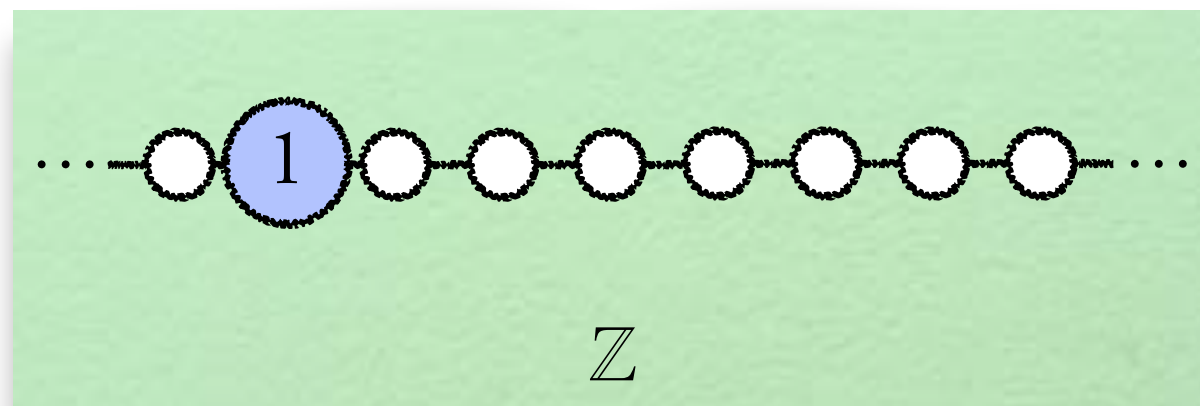
On non-isomorphic *finite* structures, Spoiler wins eventually...

Why?

...and he often wins very quickly:



But there are non-isomorphic *infinite* structures where Duplicator can survive for *arbitrarily many rounds* (not necessarily forever!)

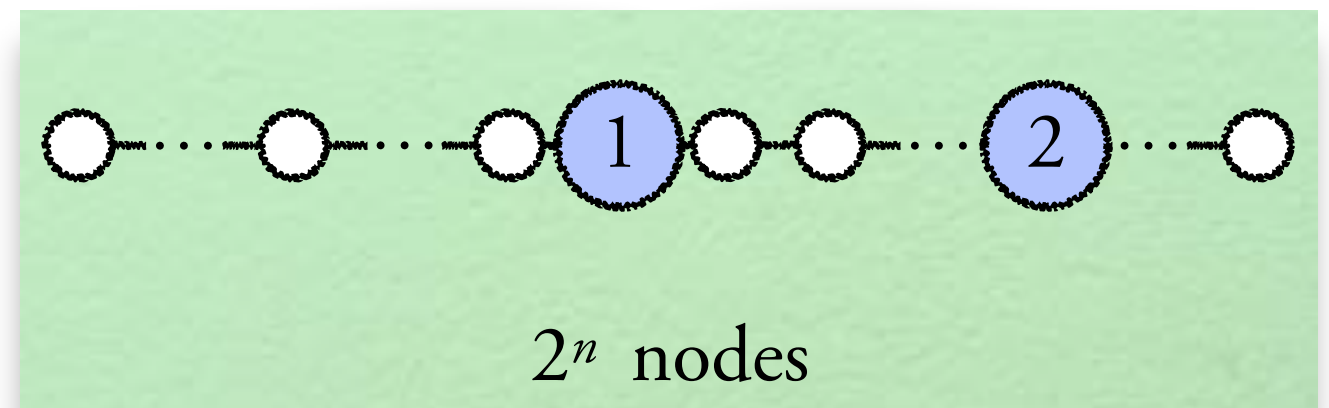
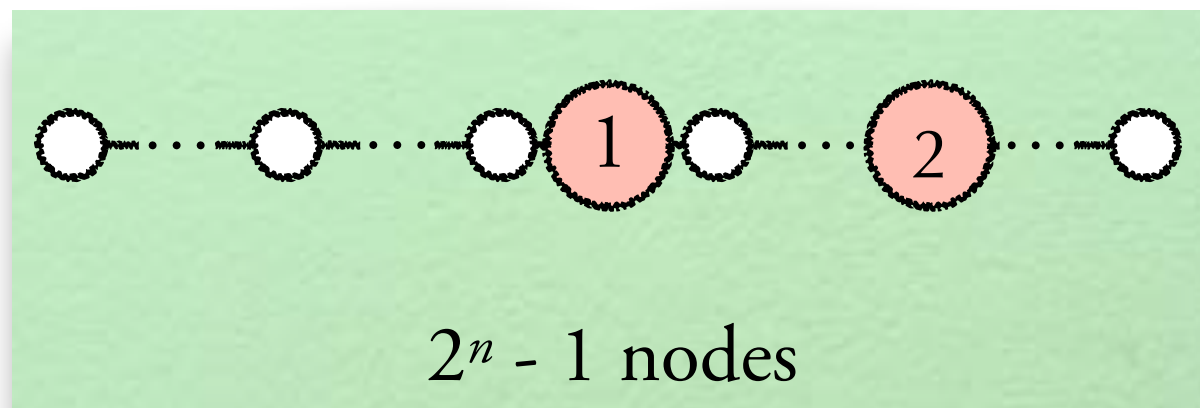


Ehrenfeucht-Fraïssé games

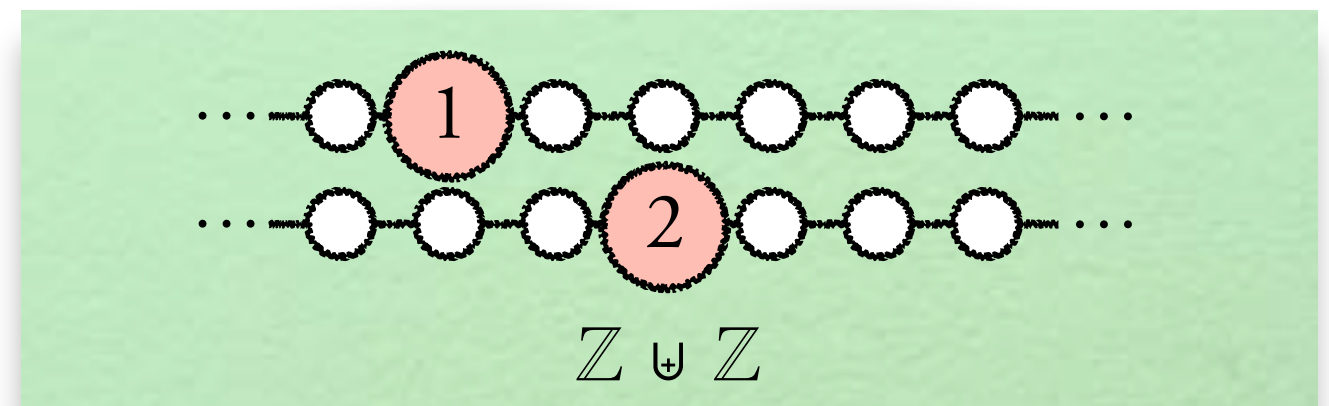
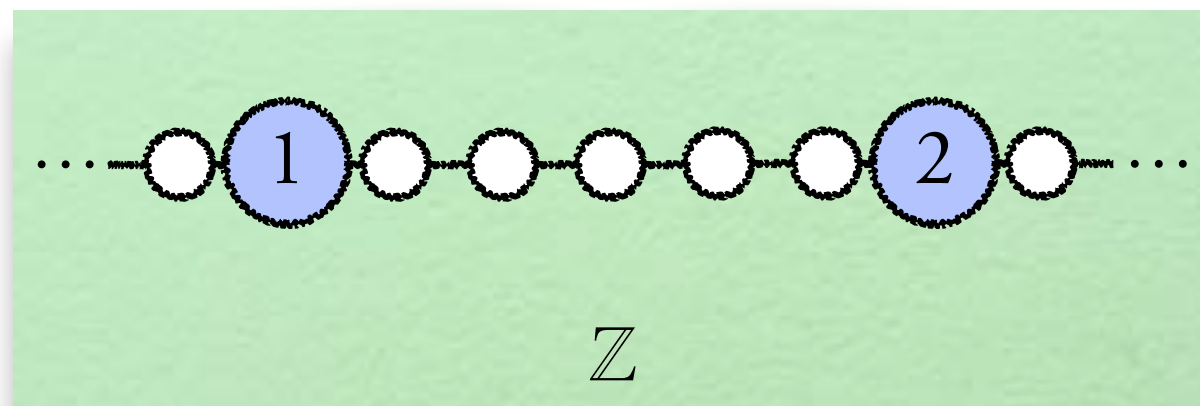
On non-isomorphic *finite* structures, Spoiler wins eventually...

Why?

...and he often wins very quickly:



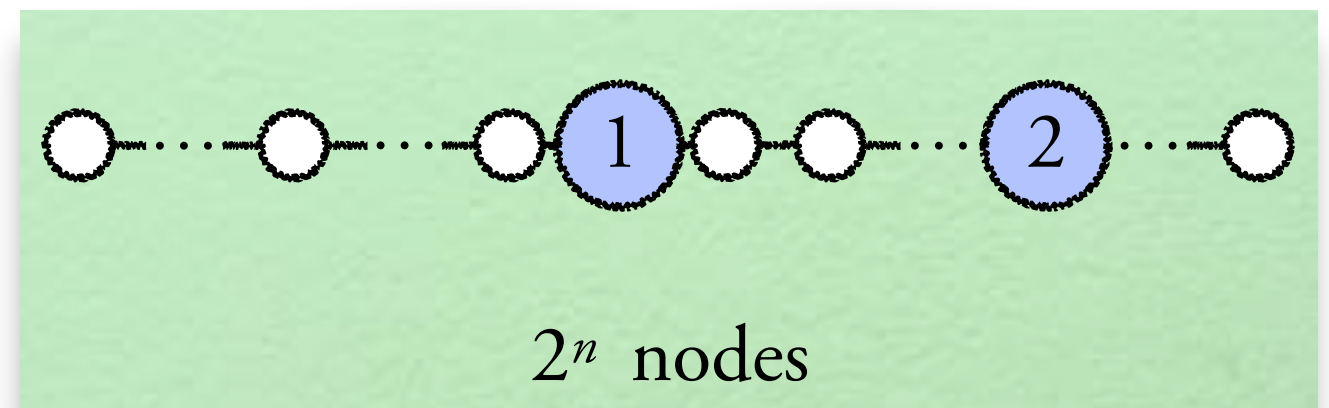
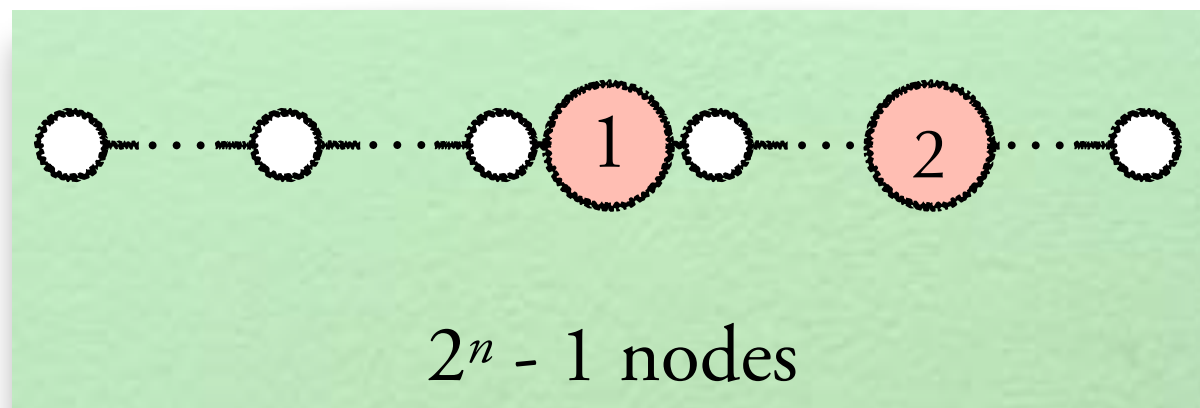
But there are non-isomorphic *infinite* structures where Duplicator can survive for *arbitrarily many rounds* (not necessarily forever!)



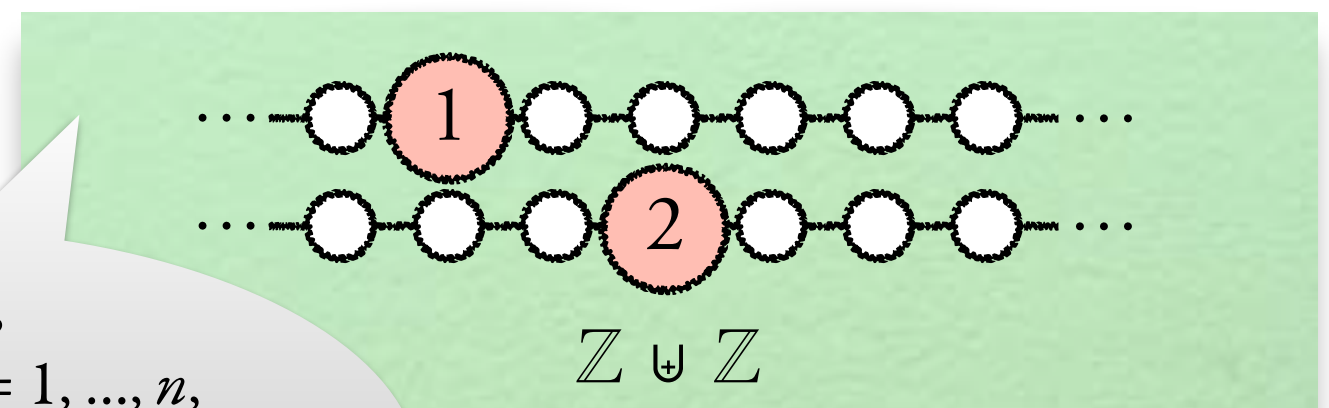
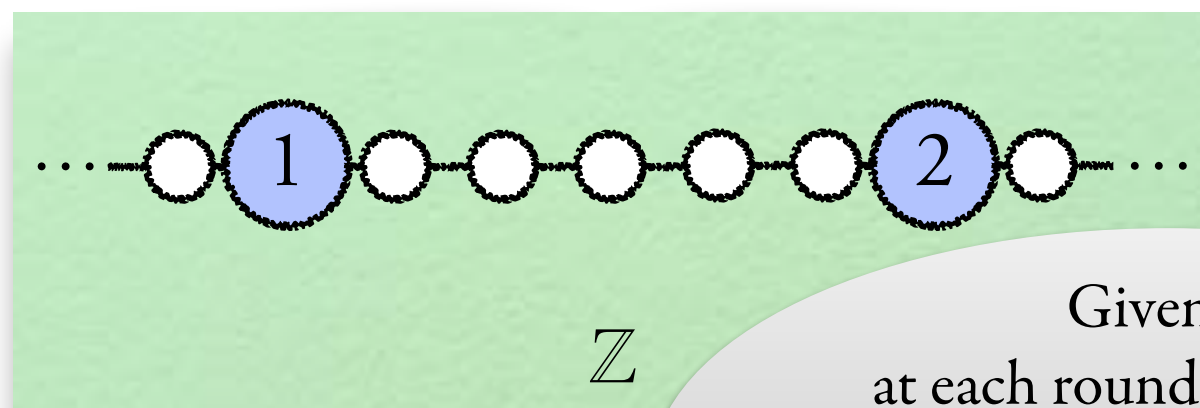
Ehrenfeucht-Fraïssé games

On non-isomorphic *finite* structures, Spoiler wins eventually... Why?

...and he often wins very quickly:



But there are non-isomorphic *infinite* structures where Duplicator can survive for *arbitrarily many rounds* (not necessarily forever!)



Given n ,
at each round $i = 1, \dots, n$,
pairs of marked nodes in S_1 and S_2
must be either at *equal distance*
or at *distance $\geq 2^{n-i}$*

Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.

Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



A new game to evaluate formulas....

The semantics game

Assume w.l.o.g. that ϕ is in **negation normal form**.

The semantics game

Assume w.l.o.g. that ϕ is in **negation normal form**.

push negations inside:

$$\neg \forall \phi \rightsquigarrow \exists \neg \phi$$

$$\neg \exists \phi \rightsquigarrow \forall \neg \phi$$

$$\neg(\phi \vee \psi) \rightsquigarrow \neg \phi \vee \neg \psi$$

...

The semantics game

Assume w.l.o.g. that ϕ is in **negation normal form**.

push negations inside:

$$\neg \forall \phi \rightsquigarrow \exists \neg \phi$$

$$\neg \exists \phi \rightsquigarrow \forall \neg \phi$$

$$\neg(\phi \wedge \psi) \rightsquigarrow \neg \phi \vee \neg \psi$$

...

Whether $S \models \phi$ can be decided by a *new game* between two players, **True** and **False**:

- $\phi = E(x, y)$ \rightarrow **True** wins if nodes marked x and y are connected by an edge, otherwise he loses
- $\phi = \exists x \phi'(x)$ \rightarrow **True** moves by marking a node x in S , the game continues with ϕ'
- $\phi = \forall y \phi'(y)$ \rightarrow **False** moves by marking a node y in S , the game continues with ϕ'
- $\phi = \phi_1 \vee \phi_2$ \rightarrow **True** moves by choosing ϕ_1 or ϕ_2 , the game continues with what he chose
- $\phi = \phi_1 \wedge \phi_2$ \rightarrow **False** moves by choosing ϕ_1 or ϕ_2 , the game continues with what he chose
- ...

The semantics game

Assume w.l.o.g. that ϕ is in **negation normal form**.

push negations inside:

$$\neg \forall \phi \rightsquigarrow \exists \neg \phi$$

$$\neg \exists \phi \rightsquigarrow \forall \neg \phi$$

$$\neg(\phi \wedge \psi) \rightsquigarrow \neg \phi \vee \neg \psi$$

...

Whether $S \models \phi$ can be decided by a *new game* between two players, **True** and **False**:

- $\phi = E(x, y)$ \rightarrow **True** wins if nodes marked x and y are connected by an edge, otherwise he loses
- $\phi = \exists x \phi'(x)$ \rightarrow **True** moves by marking a node x in S , the game continues with ϕ'
- $\phi = \forall y \phi'(y)$ \rightarrow **False** moves by marking a node y in S , the game continues with ϕ'
- $\phi = \phi_1 \vee \phi_2$ \rightarrow **True** moves by choosing ϕ_1 or ϕ_2 , the game continues with what he chose
- $\phi = \phi_1 \wedge \phi_2$ \rightarrow **False** moves by choosing ϕ_1 or ϕ_2 , the game continues with what he chose
- ...

Lemma. $S \models \phi$ iff **True** wins the semantics game.

Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.

Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



True wins the game on S_1



True wins the game on S_2

Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)



True wins the game on S_1

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



True wins the game on S_2



Turn winning strategy for **True** in S_1 into winning strategy for **True** in S_2

Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

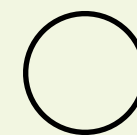
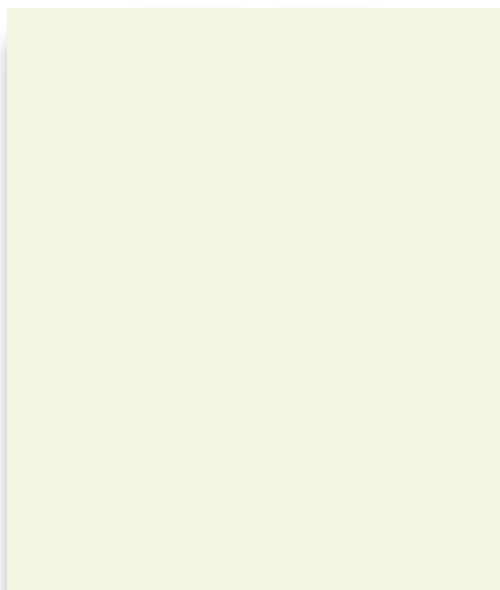
True wins the game on S_1

Consider ϕ with quantifier rank n .

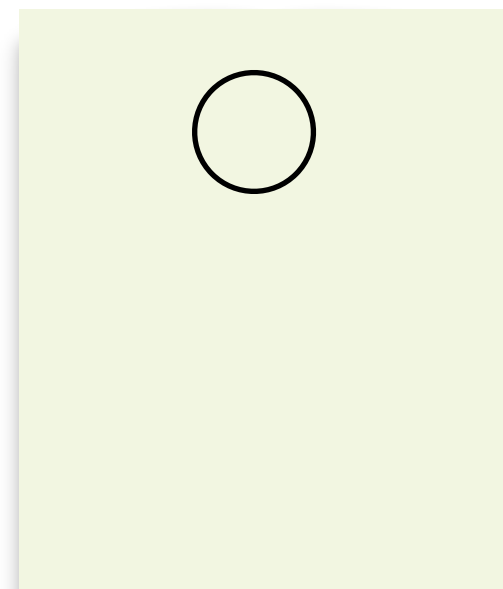
Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.

S_1



S_2



Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

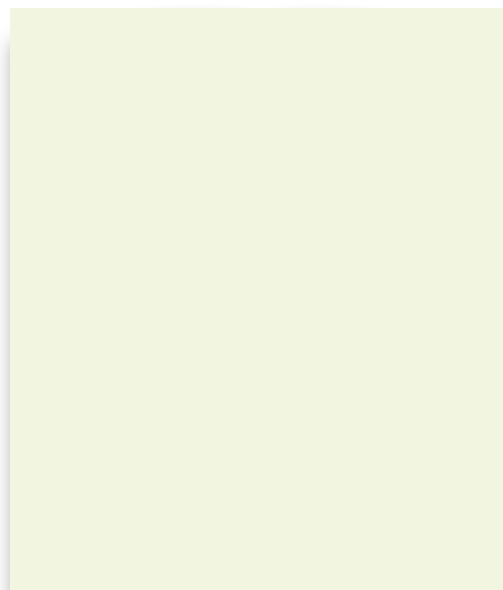
True wins the game on S_1

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

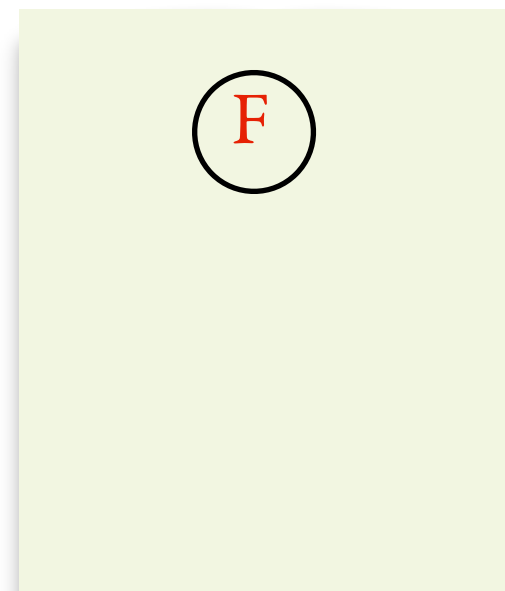
We need to prove that $S_2 \models \phi$.

S_1



F

S_2



Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

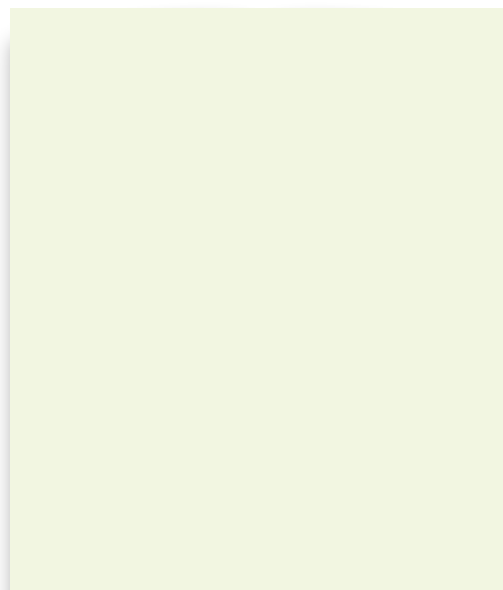
True wins the game on S_1

Consider ϕ with quantifier rank n .

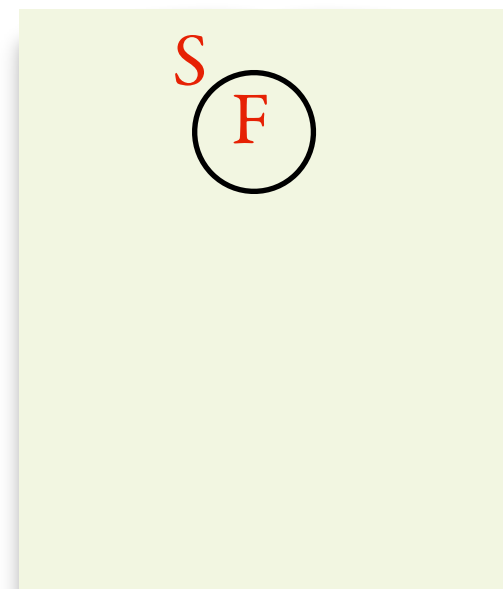
Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.

S_1



S_2



Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

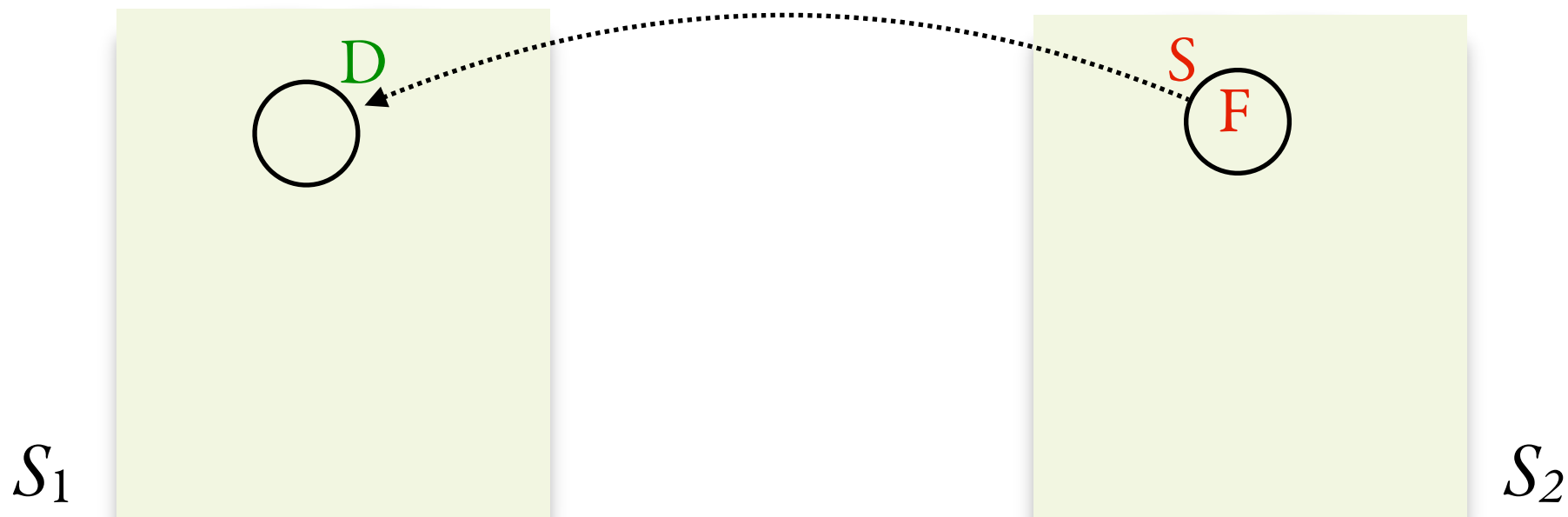
Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

True wins the game on S_1

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

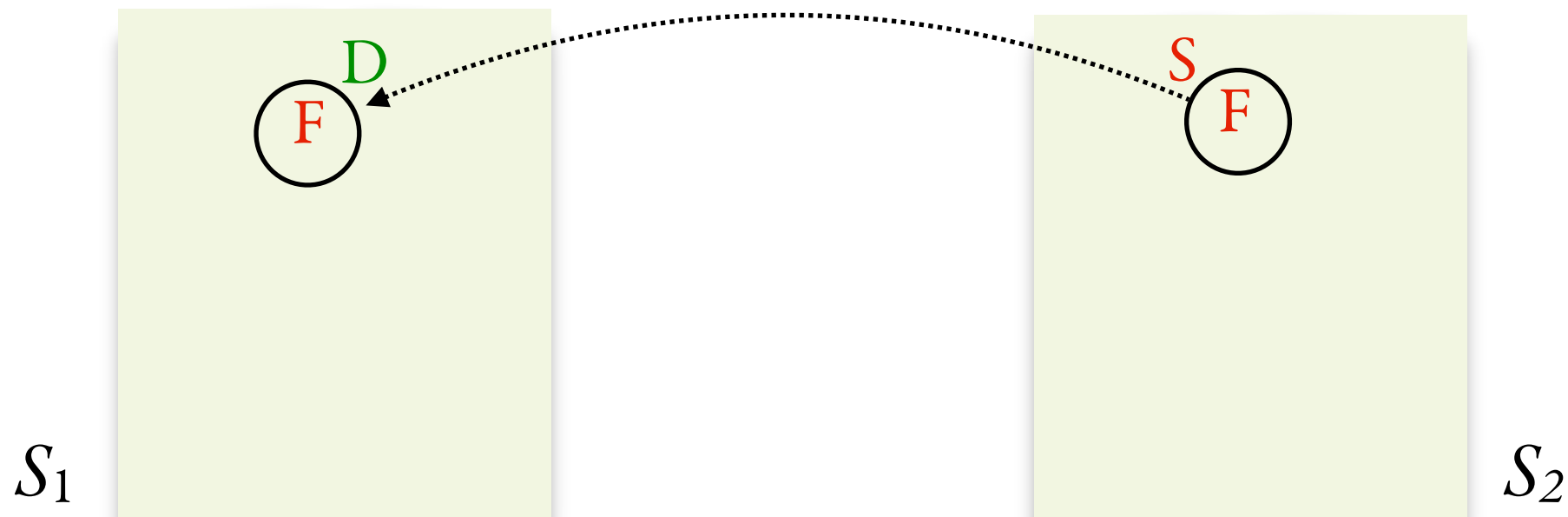
Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

True wins the game on S_1

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

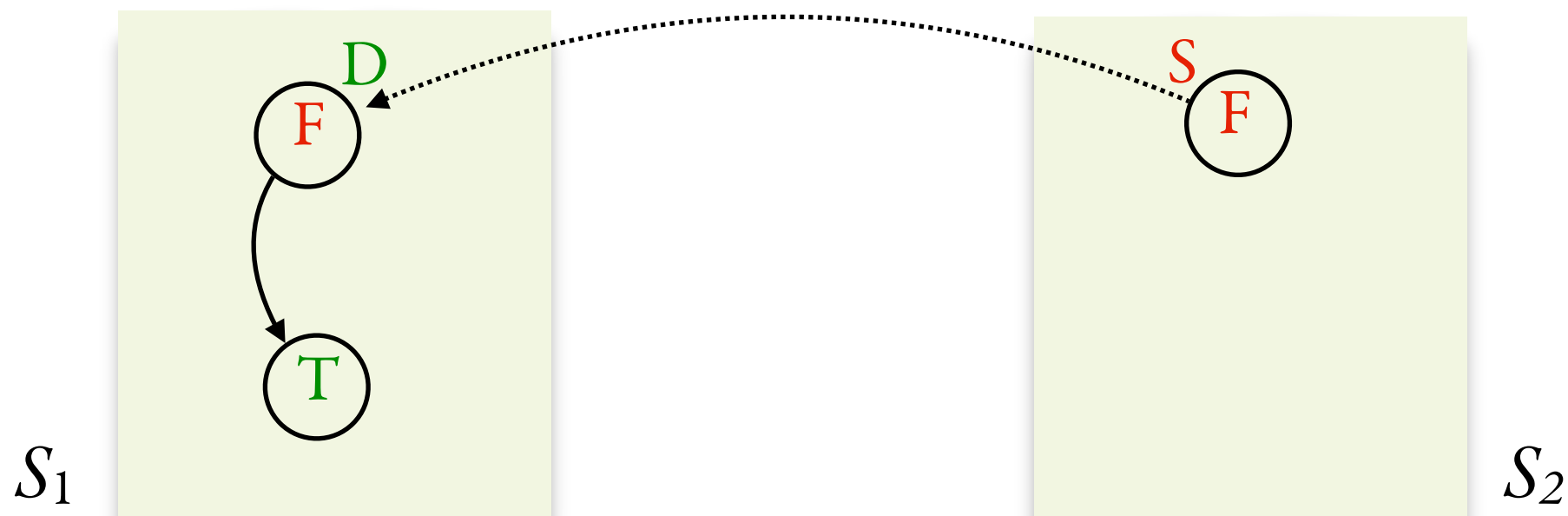
Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

True wins the game on S_1

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

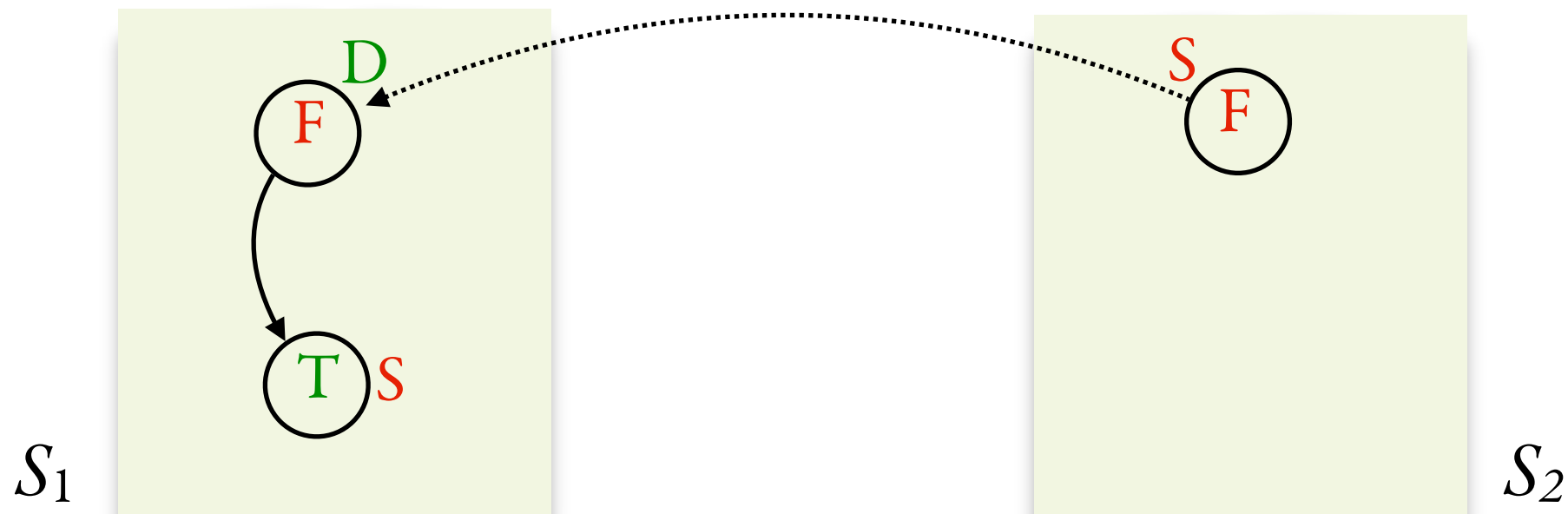
Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

True wins the game on S_1

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

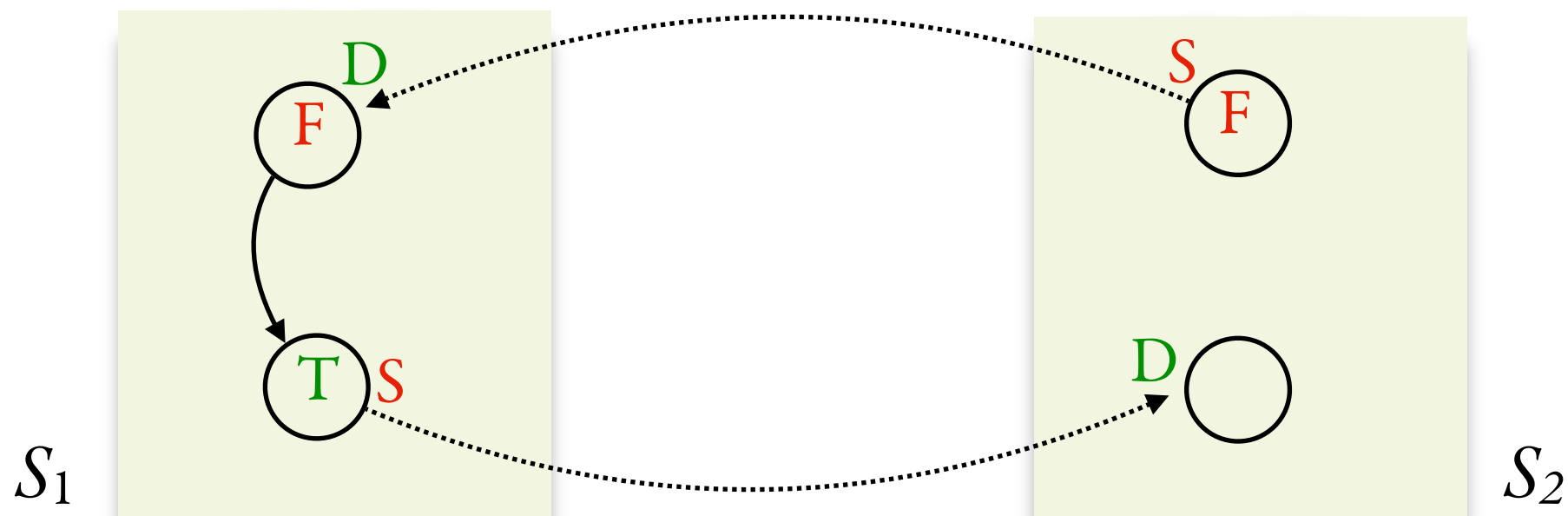
Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

True wins the game on S_1

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



Ehrenfeucht-Fraïssé games

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

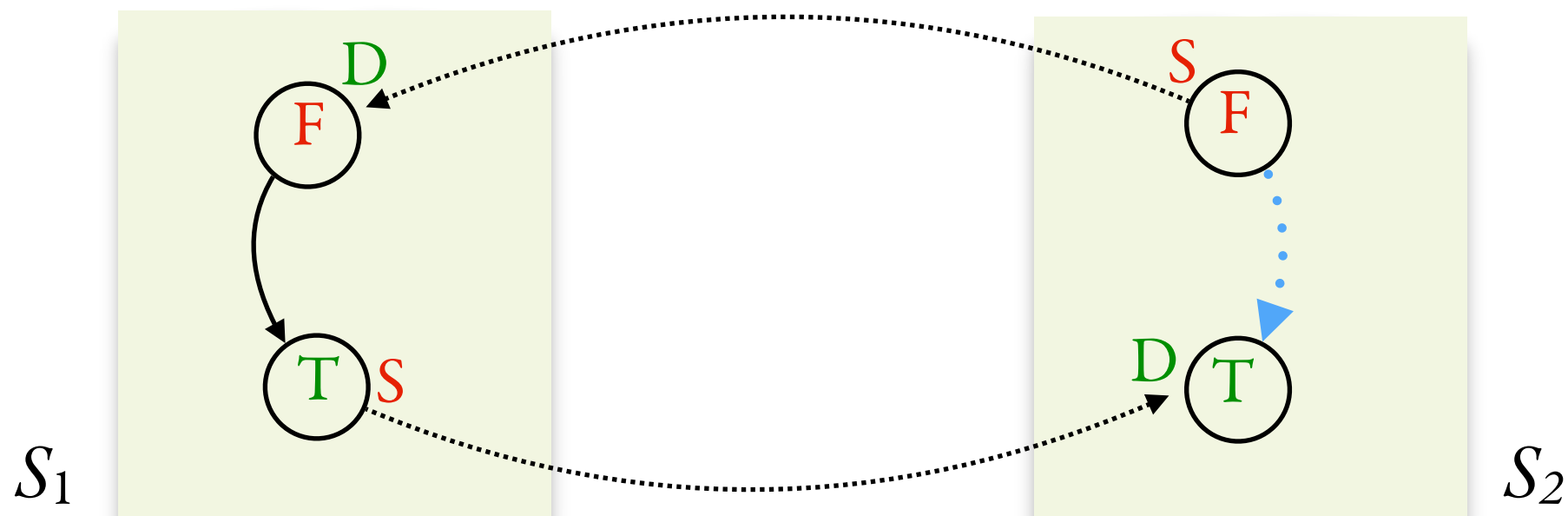
Proof ideas for the if-direction (from Duplicator's winning strategy to n -equivalence)

True wins the game on S_1

Consider ϕ with quantifier rank n .

Suppose $S_1 \models \phi$ and Duplicator survives n rounds on S_1, S_2 .

We need to prove that $S_2 \models \phi$.



Definability in FO

Theorem. S_1 and S_2 are n -equivalent

[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Corollary. A property P is *not definable in FO*

iff $\forall n \exists S_1 \in P \exists S_2 \notin P$ Duplicator can survive n rounds on S_1 and S_2 .

Definability in FO

Theorem. S_1 and S_2 are n -equivalent

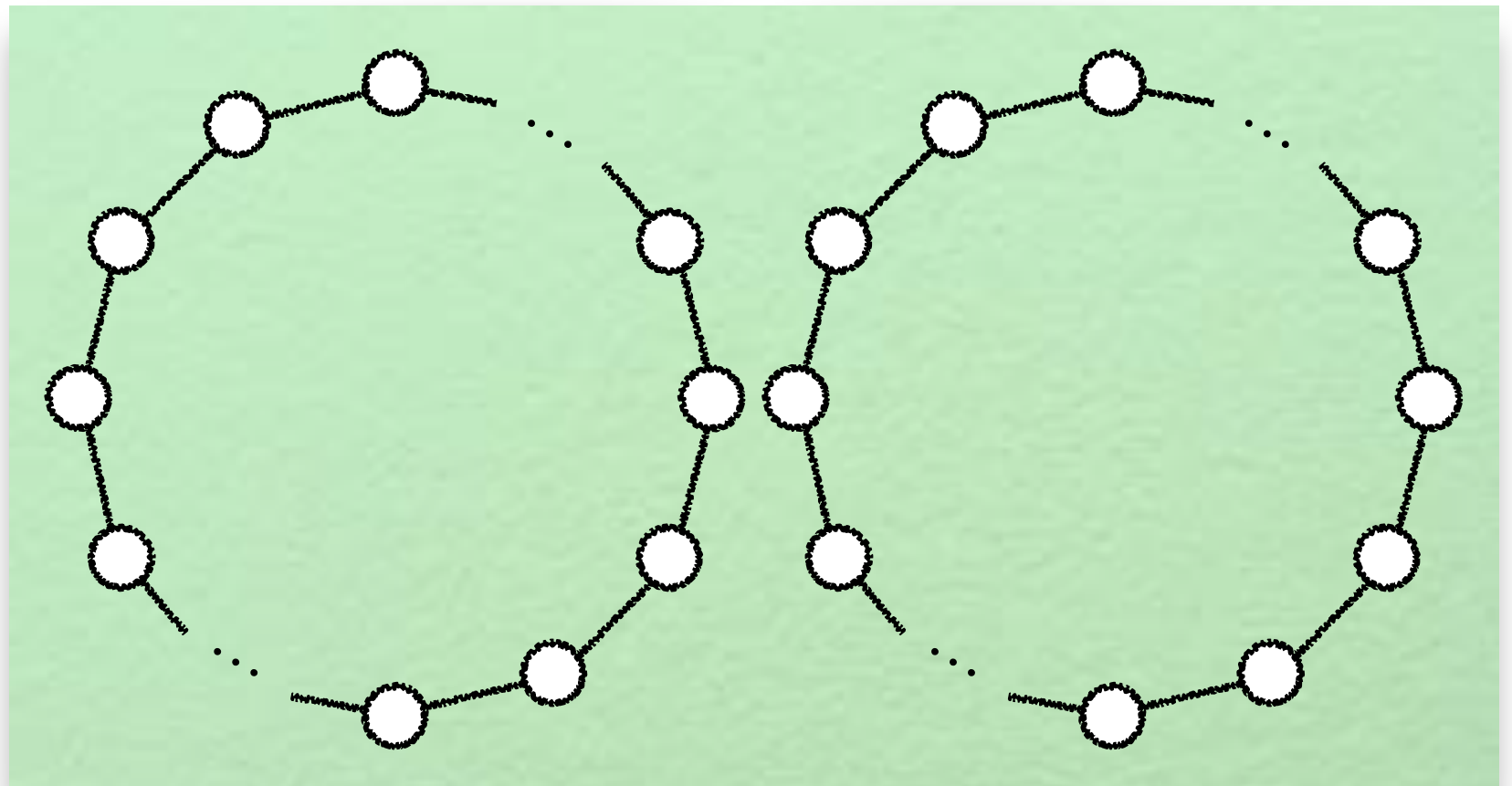
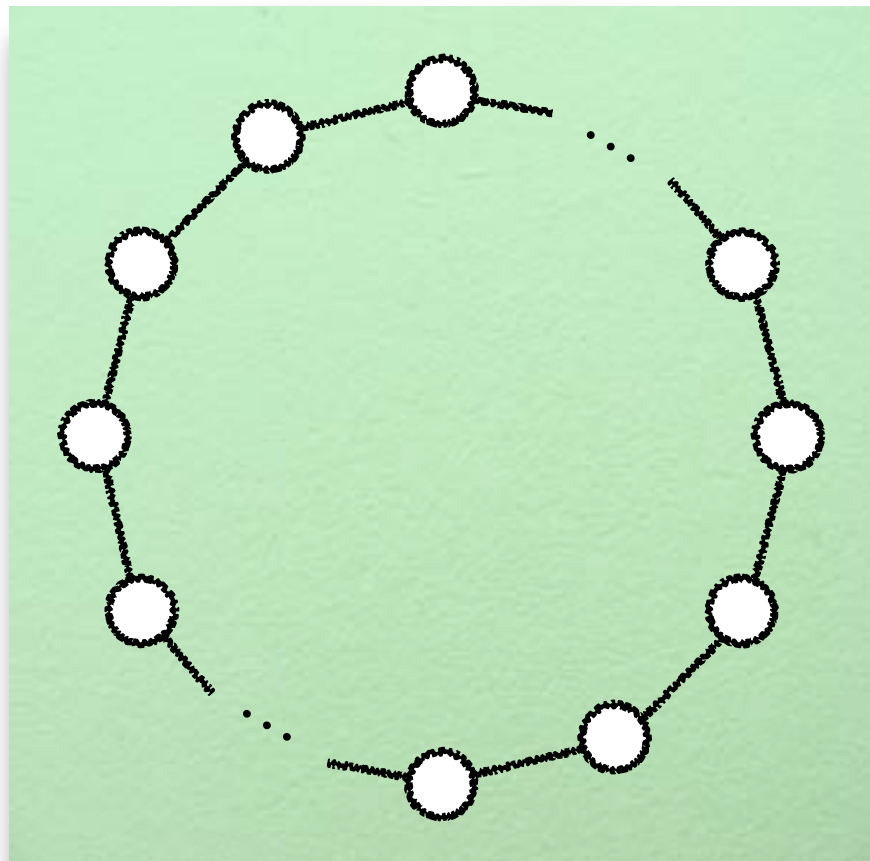
[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Corollary. A property P is *not definable in FO*

iff $\forall n \exists S_1 \in P \exists S_2 \notin P$ Duplicator can survive n rounds on S_1 and S_2 .

Example: $P = \{ \text{connected graphs} \}$. Given n , take $S_1 \in P$ large enough and $S_2 = S_1 \uplus S_1 \notin P$



Definability in FO

Theorem. S_1 and S_2 are n -equivalent

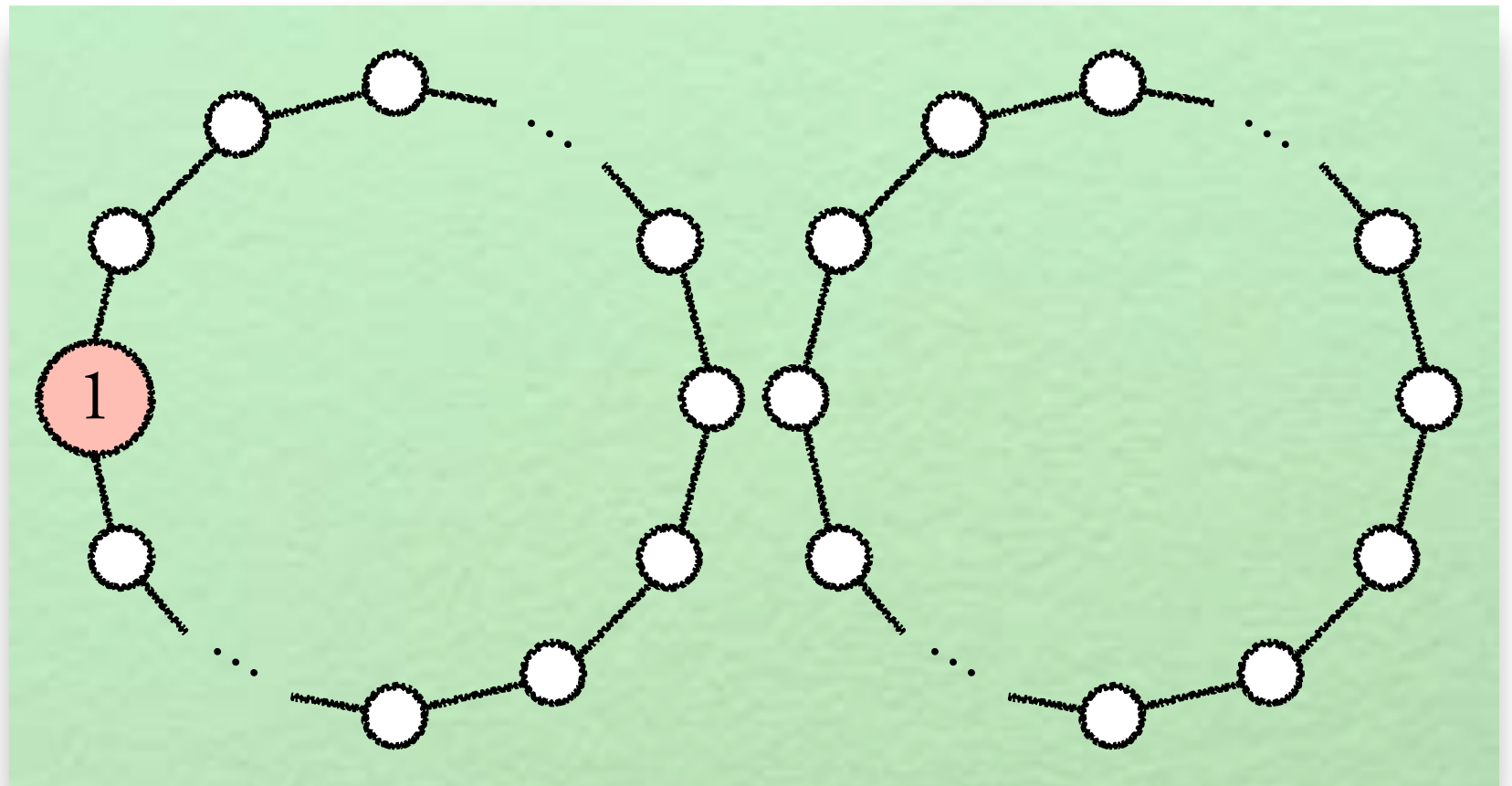
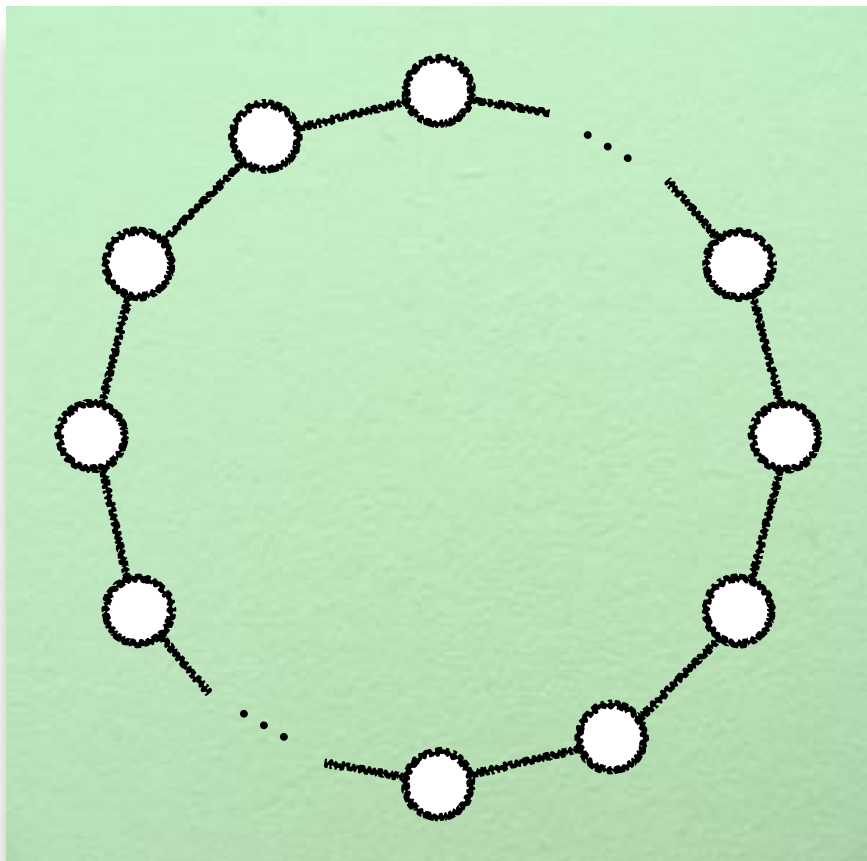
[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Corollary. A property P is *not definable in FO*

iff $\forall n \exists S_1 \in P \exists S_2 \notin P$ Duplicator can survive n rounds on S_1 and S_2 .

Example: $P = \{ \text{connected graphs} \}$. Given n , take $S_1 \in P$ large enough and $S_2 = S_1 \uplus S_1 \notin P$



Definability in FO

Theorem. S_1 and S_2 are n -equivalent

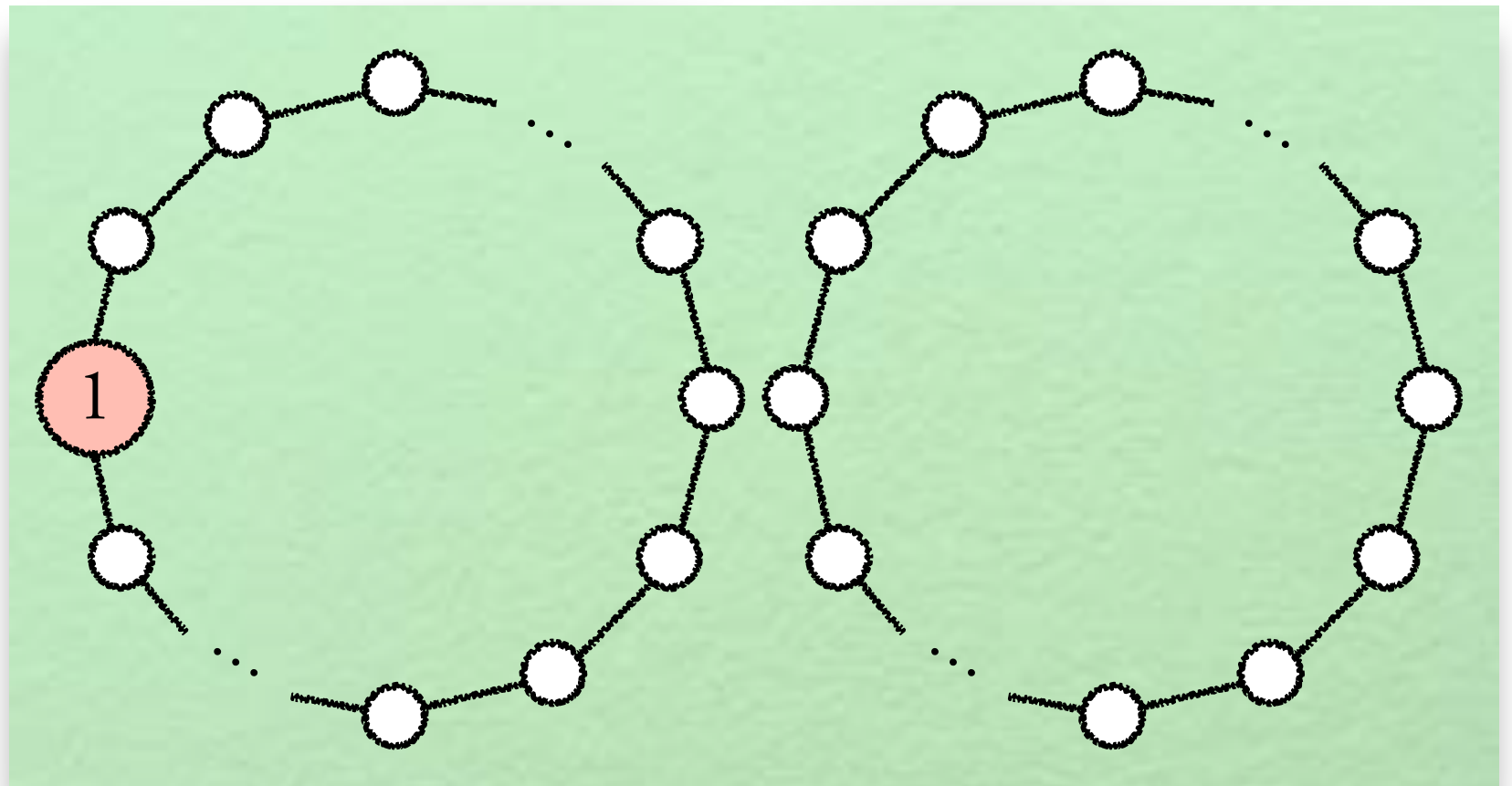
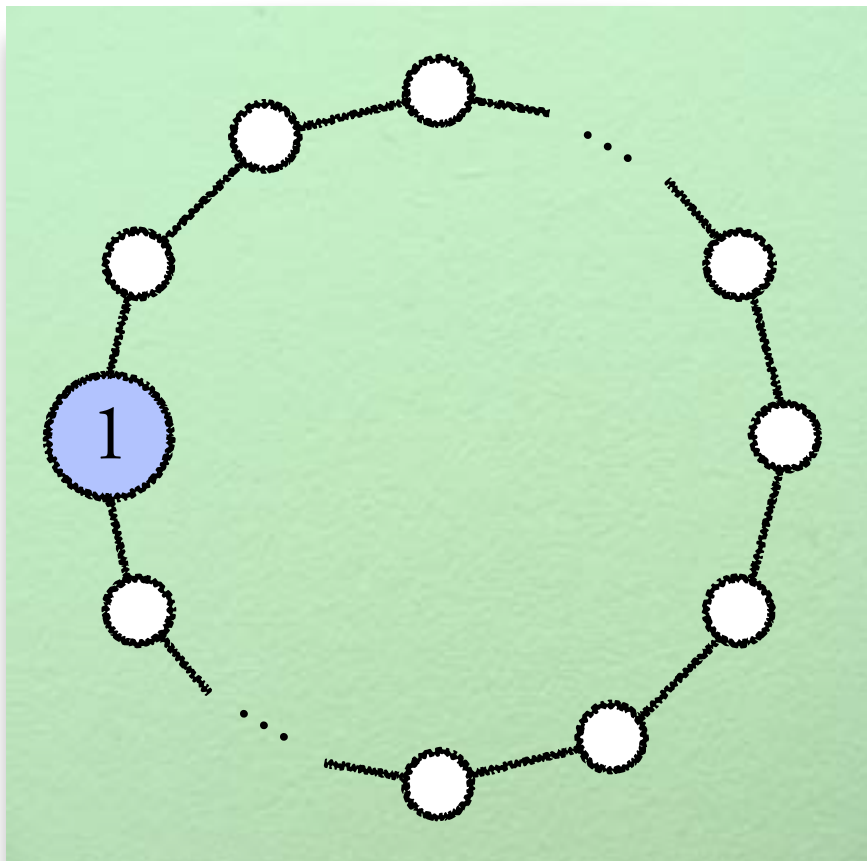
[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Corollary. A property P is *not definable in FO*

iff $\forall n \exists S_1 \in P \exists S_2 \notin P$ Duplicator can survive n rounds on S_1 and S_2 .

Example: $P = \{ \text{connected graphs} \}$. Given n , take $S_1 \in P$ large enough and $S_2 = S_1 \uplus S_1 \notin P$



Definability in FO

Theorem. S_1 and S_2 are n -equivalent

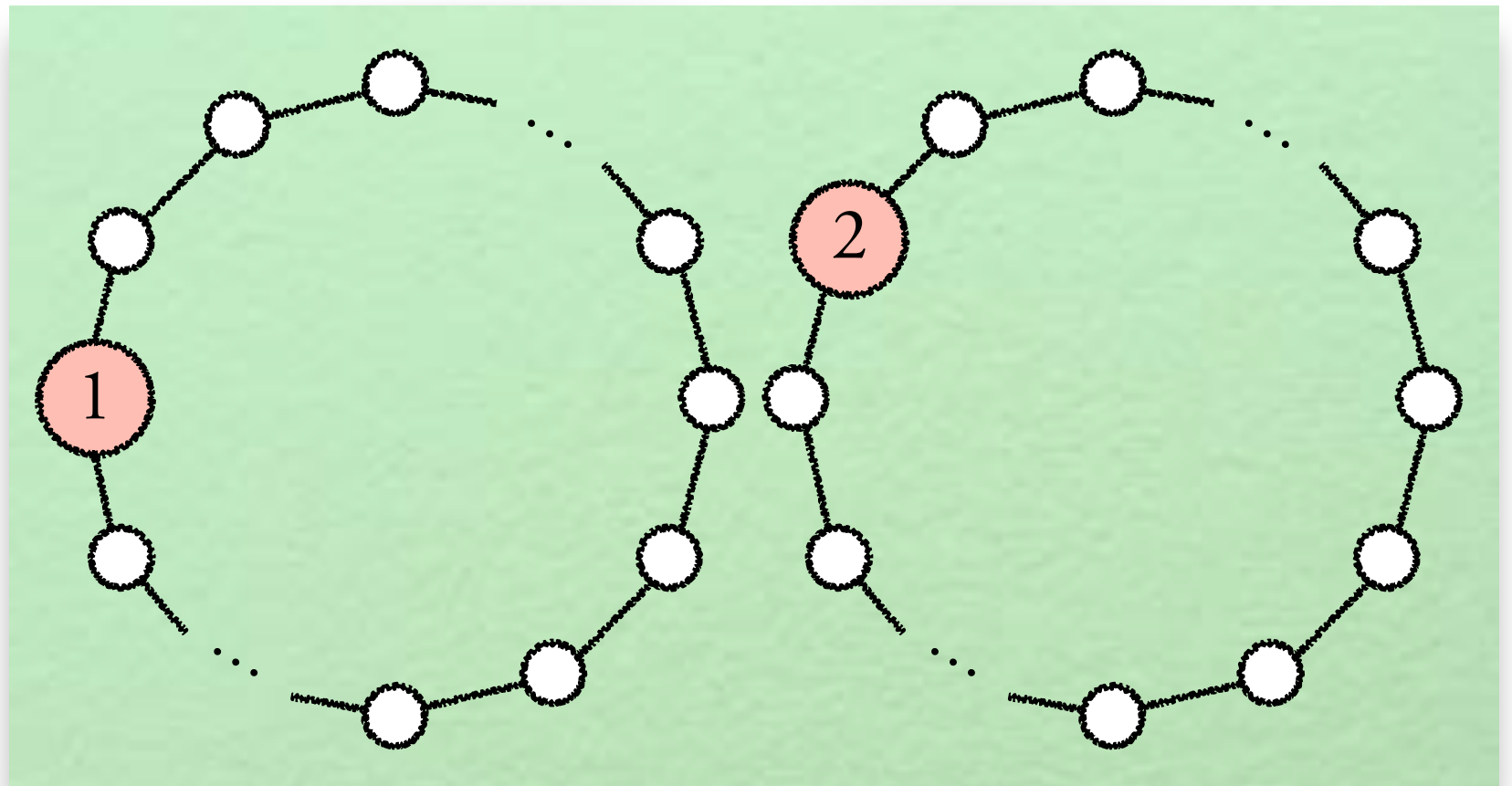
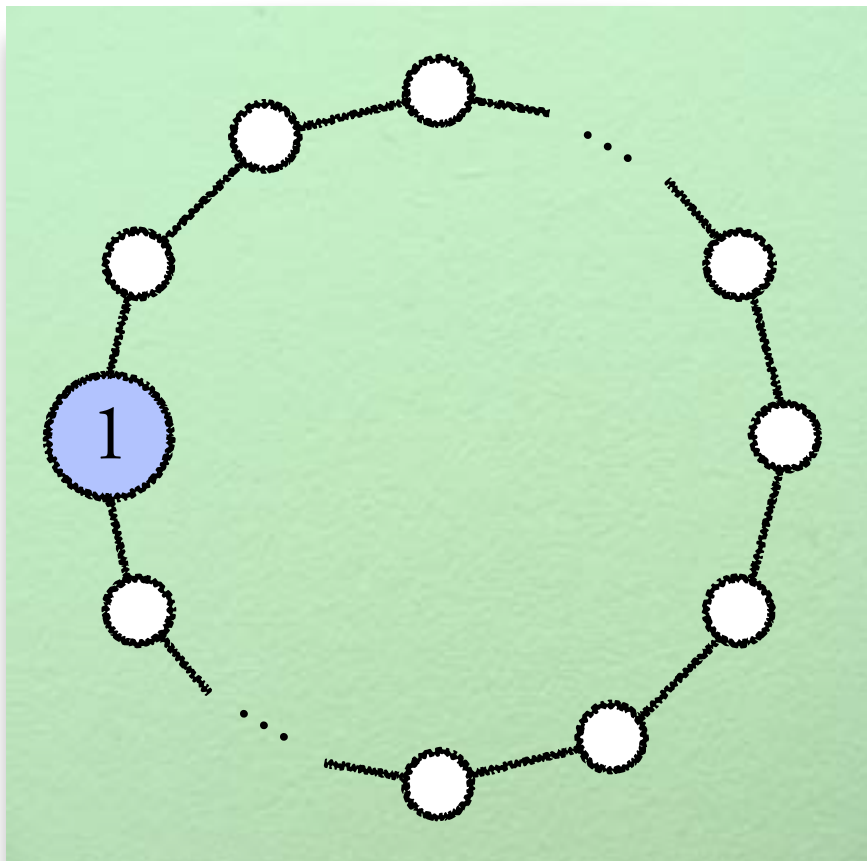
[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Corollary. A property P is *not definable in FO*

iff $\forall n \exists S_1 \in P \exists S_2 \notin P$ Duplicator can survive n rounds on S_1 and S_2 .

Example: $P = \{ \text{connected graphs} \}$. Given n , take $S_1 \in P$ large enough and $S_2 = S_1 \uplus S_1 \notin P$



Definability in FO

Theorem. S_1 and S_2 are n -equivalent

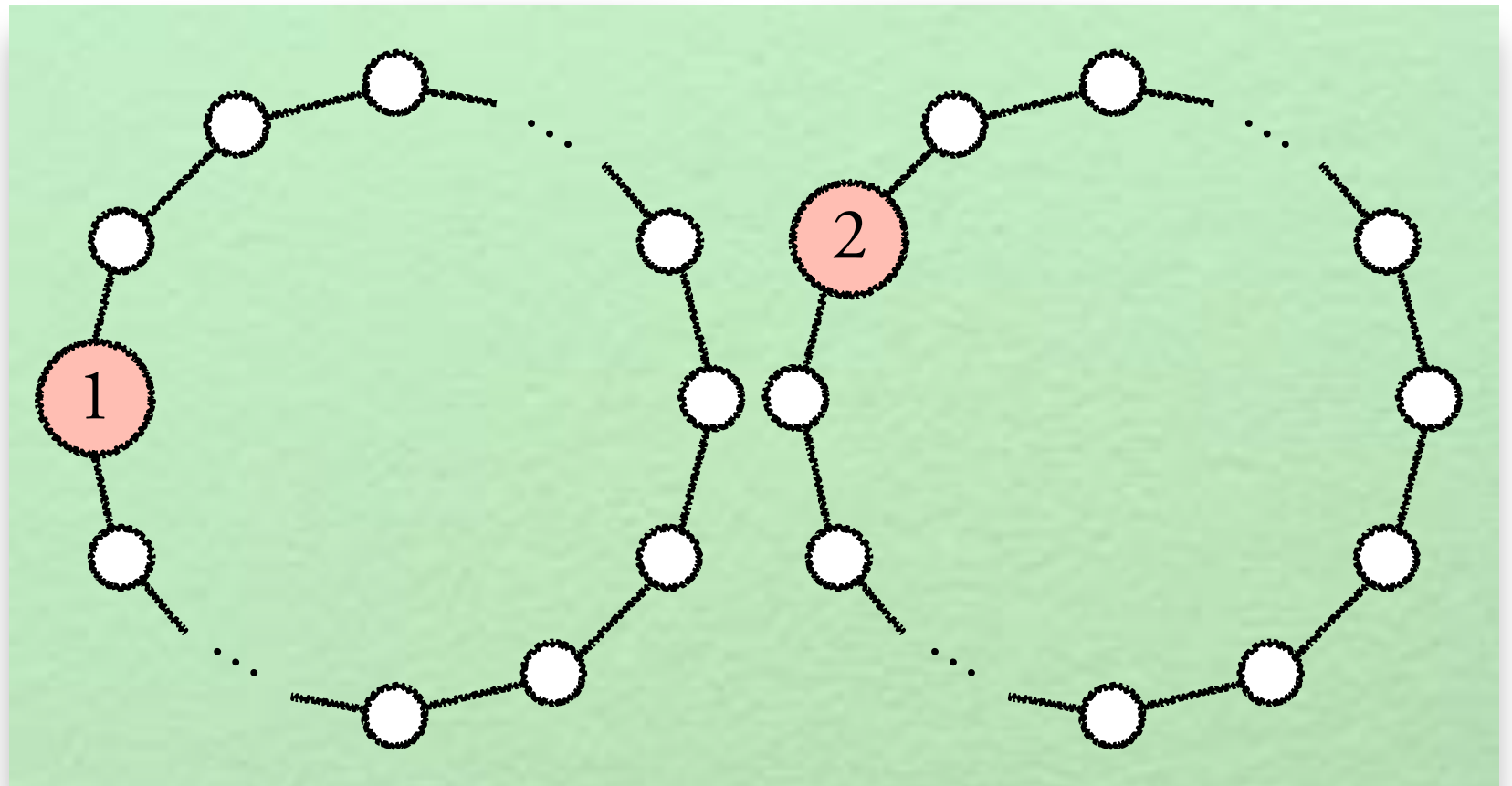
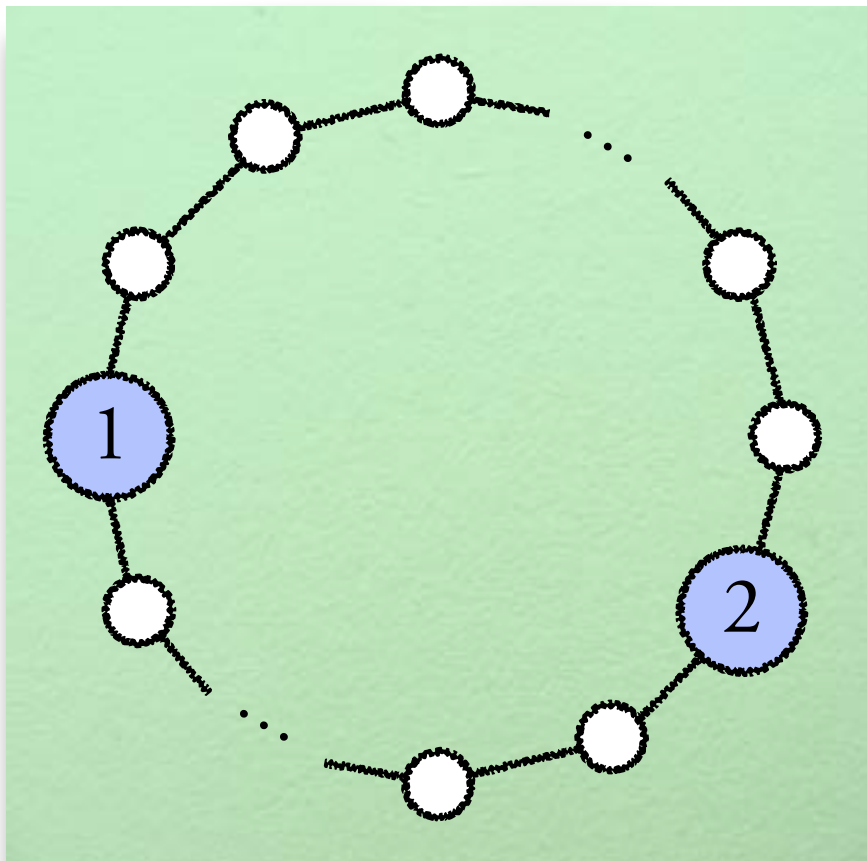
[Fraïssé '50, Ehrenfeucht '60]

iff Duplicator has a strategy to survive n rounds in the EF game on S_1 and S_2 .

Corollary. A property P is *not definable in FO*

iff $\forall n \exists S_1 \in P \exists S_2 \notin P$ Duplicator can survive n rounds on S_1 and S_2 .

Example: $P = \{ \text{connected graphs} \}$. Given n , take $S_1 \in P$ large enough and $S_2 = S_1 \uplus S_1 \notin P$



Ehrenfeucht-Fraïssé games

Several properties can be proved to be *not FO-definable*:

- connectivity (previous slide)

Ehrenfeucht-Fraïssé games

Several properties can be proved to be *not FO-definable*:

- connectivity (previous slide)

- even / odd size **Your turn now!** ...given n , take $S_1 =$ large even structure
 $S_2 =$ large odd structure...

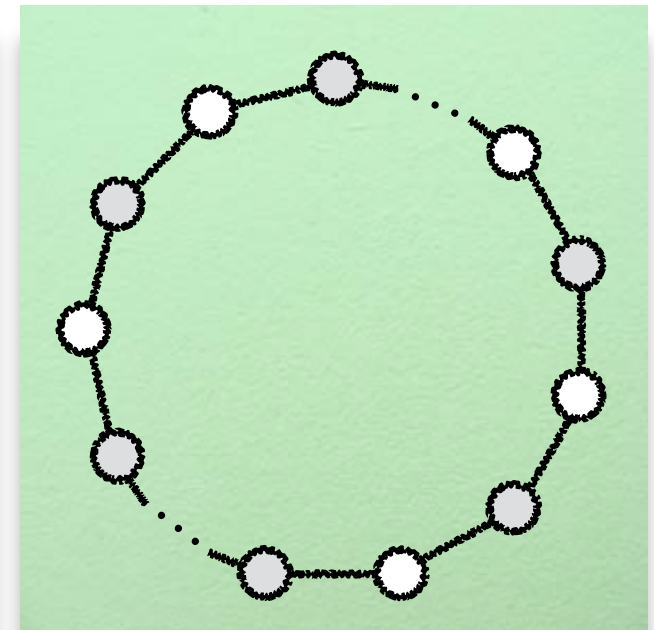
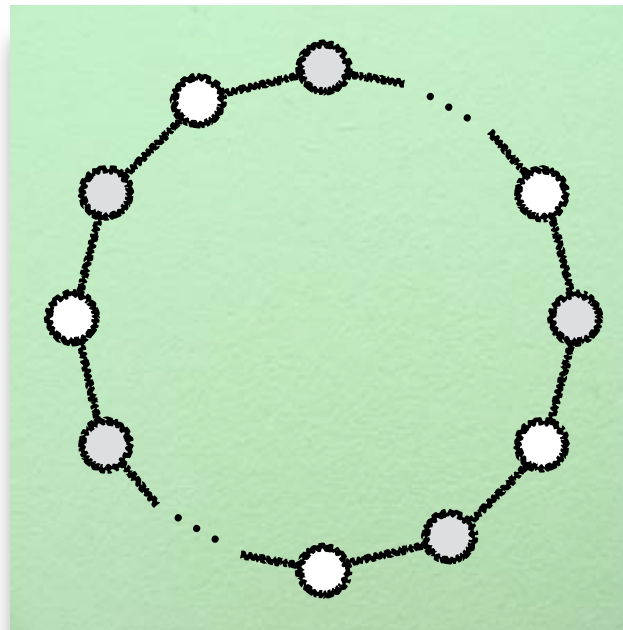
Ehrenfeucht-Fraïssé games

Several properties can be proved to be *not FO-definable*:

- connectivity (previous slide)

- even / odd size **Your turn now!** ...given n , take $S_1 =$ large even structure
 $S_2 =$ large odd structure...

- 2-colorability Given n , take $S_1 =$ large even cycle $S_2 =$ large odd cycle



Ehrenfeucht-Fraïssé games

Several properties can be proved to be *not FO-definable*:

- connectivity (previous slide)

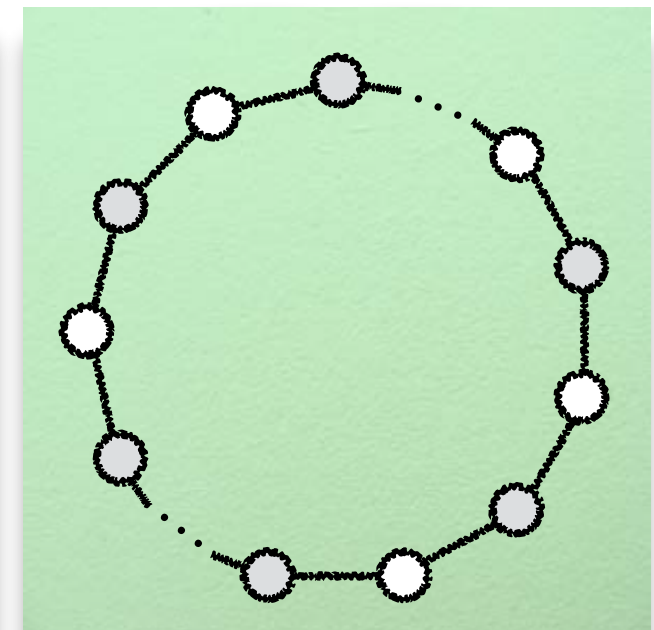
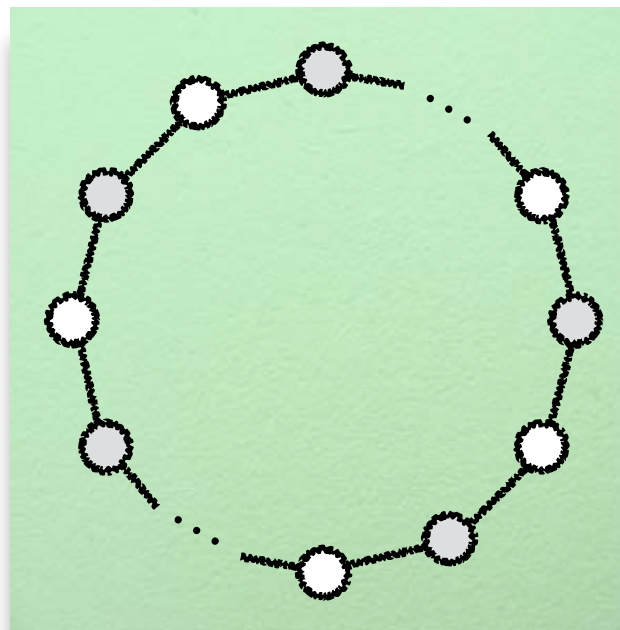
- even / odd size **Your turn now!** ...given n , take $S_1 =$ large even structure
 $S_2 =$ large odd structure...

- 2-colorability Given n , take $S_1 =$ large even cycle $S_2 =$ large odd cycle

- finiteness

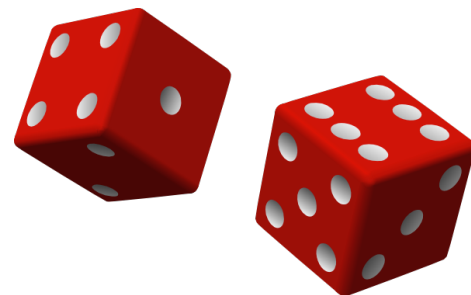
- acyclicity

...



0-1 Law

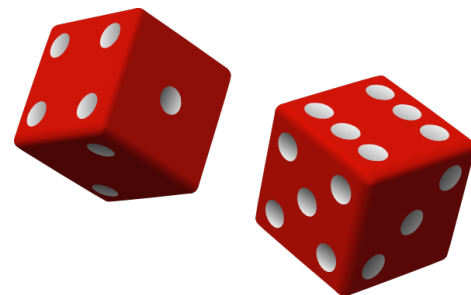
A different perspective: a coarser view on expressiveness...



0-1 Law

A different perspective: a coarser view on expressiveness...

What percentage of graphs verify a given FO sentence?

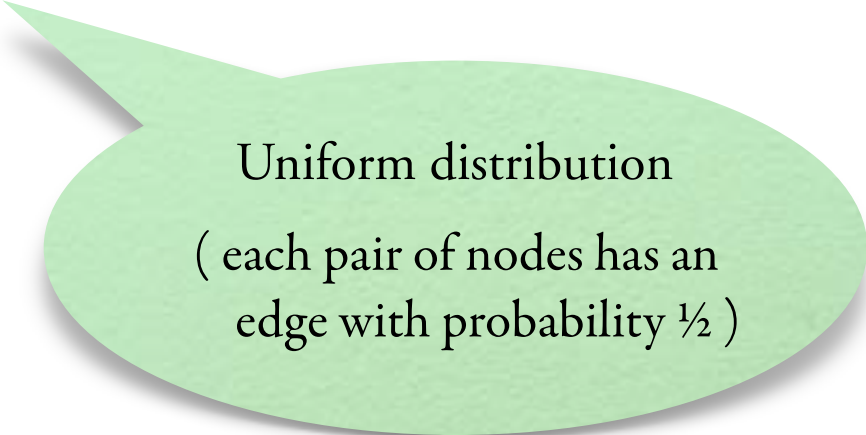


0-1 Law

$\mu_n(\mathbf{P})$ = “probability that property \mathbf{P} holds in a random graph with n nodes”

0-1 Law

$\mu_n(\mathbf{P})$ = “probability that property \mathbf{P} holds in a random graph with n nodes”



Uniform distribution
(each pair of nodes has an
edge with probability $\frac{1}{2}$)

0-1 Law

$\mu_n(\mathbf{P})$ = “probability that property \mathbf{P} holds in a random graph with n nodes”

$$\mathbf{C}_n = \{ \text{graphs with } n \text{ nodes} \}$$

$$\mu_n(\mathbf{P}) = \frac{|\{G \in \mathbf{C}_n \mid G \models \mathbf{P}\}|}{|\mathbf{C}_n|} = 2^{n^2}$$

Uniform distribution
(each pair of nodes has an
edge with probability $\frac{1}{2}$)

0-1 Law

$\mu_n(\mathbf{P})$ = “probability that property \mathbf{P} holds in a random graph with n nodes”

$$\mathbf{C}_n = \{ \text{graphs with } n \text{ nodes} \}$$

$$\mu_n(\mathbf{P}) = \frac{|\{G \in \mathbf{C}_n \mid G \models \mathbf{P}\}|}{|\mathbf{C}_n|} = \frac{|\{G \in \mathbf{C}_n \mid G \models \mathbf{P}\}|}{2^{n^2}}$$

Uniform distribution
(each pair of nodes has an
edge with probability $\frac{1}{2}$)

E.g. for \mathbf{P} = “the graph is complete”

$$\mu_3(\mathbf{P}) = \frac{1}{|\mathbf{C}_3|} = \frac{1}{2^{3^2}}$$

0-1 Law

$\mu_n(\mathbf{P})$ = “probability that property \mathbf{P} holds in a random graph with n nodes”

$$\mathbf{C}_n = \{ \text{graphs with } n \text{ nodes} \}$$

$$\mu_n(\mathbf{P}) = \frac{|\{G \in \mathbf{C}_n \mid G \models \mathbf{P}\}|}{|\mathbf{C}_n|} = \frac{|\{G \in \mathbf{C}_n \mid G \models \mathbf{P}\}|}{2^{n^2}}$$

Uniform distribution
(each pair of nodes has an
edge with probability $\frac{1}{2}$)

E.g. for \mathbf{P} = “the graph is complete”

$$\mu_3(\mathbf{P}) = \frac{1}{|\mathbf{C}_3|} = \frac{1}{2^{3^2}}$$

$$\mu_\infty(\mathbf{P}) = \lim_{n \rightarrow \infty} \mu_n(\mathbf{P})$$

0-1 Law

Theorem.

[Glebskii et al. '69, Fagin '76]

For every *FO sentence* ϕ , $\mu_{\infty}(\phi)$ is either 0 or 1.

0-1 Law

Theorem.

[Glebskii et al. '69, Fagin '76]

For every *FO sentence* ϕ , $\mu_{\infty}(\phi)$ is either 0 or 1.

Examples:

- $\phi = \text{“there is a triangle”}$

$$\mu_3(\phi) = 1/|C_3| \quad \mu_{3n}(\phi) \geq 1 - (1 - 1/|C_3|)^n \rightarrow 1$$

0-1 Law

Theorem.

[Glebskii et al. '69, Fagin '76]

For every *FO sentence* ϕ , $\mu_{\infty}(\phi)$ is either 0 or 1.

Examples:

- ϕ = “there is a triangle”
 $\mu_3(\phi) = 1/|C_3|$ $\mu_{3n}(\phi) \geq 1 - (1 - 1/|C_3|)^n \rightarrow 1$
- ϕ_H = “there is an occurrence of H as induced sub-graph”
 $\mu_{\infty}(\phi_H) = 1$

0-1 Law

Theorem.

[Glebskii et al. '69, Fagin '76]

For every *FO sentence* ϕ , $\mu_{\infty}(\phi)$ is either 0 or 1.

Examples:

- ϕ = “there is a triangle”
 $\mu_3(\phi) = 1/|C_3|$ $\mu_{3n}(\phi) \geq 1 - (1 - 1/|C_3|)^n \rightarrow 1$
- ϕ_H = “there is an occurrence of H as induced sub-graph”
 $\mu_{\infty}(\phi_H) = 1$
- ϕ = “there no 5-clique”
 $\mu_{\infty}(\phi) = 0$

0-1 Law

Theorem.

[Glebskii et al. '69, Fagin '76]

For every *FO sentence* ϕ , $\mu_{\infty}(\phi)$ is either 0 or 1.

Examples:

- ϕ = “there is a triangle”
 $\mu_3(\phi) = 1/|C_3|$ $\mu_{3n}(\phi) \geq 1 - (1 - 1/|C_3|)^n \rightarrow 1$
- ϕ_H = “there is an occurrence of H as induced sub-graph”
 $\mu_{\infty}(\phi_H) = 1$
- ϕ = “there no 5-clique”
 $\mu_{\infty}(\phi) = 0$
- ϕ = “even number of edges”
- ϕ = “even number of nodes”

Your turn!

0-1 Law

Theorem.

[Glebskii et al. '69, Fagin '76]

For every *FO sentence* ϕ , $\mu_{\infty}(\phi)$ is either 0 or 1.

Examples:

- $\phi = \text{"there is a triangle"}$ $\mu_3(\phi) = 1/|C_3|$ $\mu_{3n}(\phi) \geq 1 - (1 - 1/|C_3|)^n \rightarrow 1$
- $\phi_H = \text{"there is an occurrence of } H \text{ as induced sub-graph"}$ $\mu_{\infty}(\phi_H) = 1$
- $\phi = \text{"there no 5-clique"}$ $\mu_{\infty}(\phi) = 0$
- $\phi = \text{"even number of edges"}$
- $\phi = \text{"even number of nodes"}$

Your turn!

$$\mu_{\infty}(\phi) = 1/2$$

$\mu_{\infty}(\phi)$ not even defined

0-1 Law

Theorem.

[Glebskii et al. '69, Fagin '76]

For every *FO sentence* ϕ , $\mu_\infty(\phi)$ is either 0 or 1.

Examples:

- ϕ = “there is a triangle”
 $\mu_3(\phi) = 1/|C_3|$ $\mu_{3n}(\phi) \geq 1 - (1 - 1/|C_3|)^n \rightarrow 1$
- ϕ_H = “there is an occurrence of H as induced sub-graph”
 $\mu_\infty(\phi_H) = 1$
- ϕ = “there no 5-clique”
 $\mu_\infty(\phi) = 0$
- ϕ = “even number of edges”
 $\mu_\infty(\phi) = 1/2$
- ϕ = “even number of nodes”
 $\mu_\infty(\phi)$ not even defined
- ϕ = “more edges than nodes”
 $\mu_\infty(\phi) = 1$
(yet not FO-definable!)

Your turn!

0-1 Law

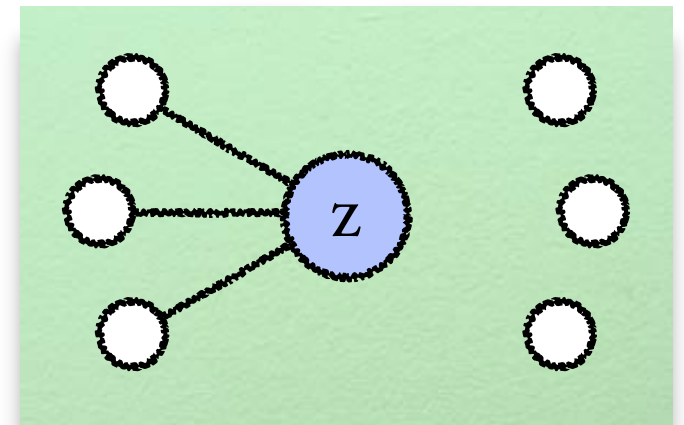


For every *FO sentence* ϕ , $\mu_\infty(\phi)$ is either 0 or 1.

Let k = quantifier rank of ϕ

$$\delta_k = \forall x_1, \dots, x_k \forall y_1, \dots, y_k \exists z \bigwedge_{i,j} x_i \neq y_j \wedge E(x_i, z) \wedge \neg E(y_j, z)$$

(Extension Formula/Axiom)



0-1 Law

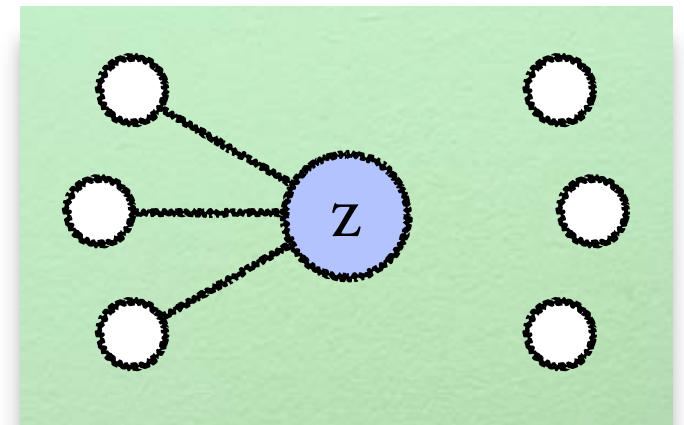


For every *FO sentence* ϕ , $\mu_\infty(\phi)$ is either 0 or 1.

Let k = quantifier rank of ϕ

$$\delta_k = \forall x_1, \dots, x_k \forall y_1, \dots, y_k \exists z \bigwedge_{i,j} x_i \neq y_j \wedge E(x_i, z) \wedge \neg E(y_j, z)$$

(Extension Formula/Axiom)



Fact 1: If $G \models \delta_k \wedge H \models \delta_k$ then
Duplicator survives k rounds on G, H

0-1 Law

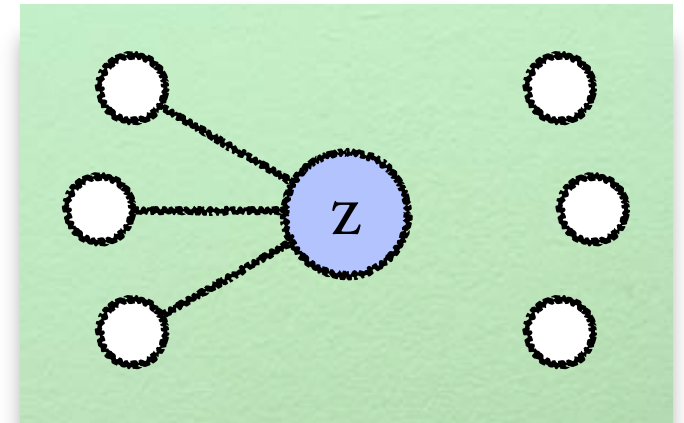


For every *FO sentence* ϕ , $\mu_\infty(\phi)$ is either 0 or 1.

Let k = quantifier rank of ϕ

$$\delta_k = \forall x_1, \dots, x_k \forall y_1, \dots, y_k \exists z \bigwedge_{i,j} x_i \neq y_j \wedge E(x_i, z) \wedge \neg E(y_j, z)$$

(Extension Formula/Axiom)



Fact 1: If $G \models \delta_k \wedge H \models \delta_k$ then
Duplicator survives k rounds on G, H

Fact 2: $\mu_\infty(\delta_k) = 1$
(δ_k is almost surely true)

0-1 Law

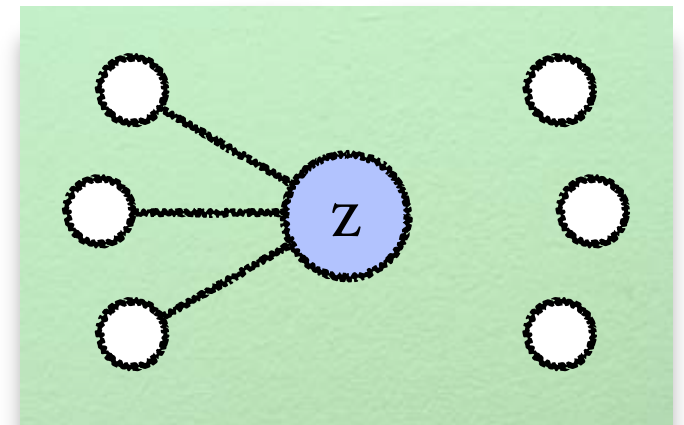


For every *FO sentence* ϕ , $\mu_\infty(\phi)$ is either 0 or 1.

Let k = quantifier rank of ϕ

$$\delta_k = \forall x_1, \dots, x_k \forall y_1, \dots, y_k \exists z \bigwedge_{i,j} x_i \neq y_j \wedge E(x_i, z) \wedge \neg E(y_j, z)$$

(Extension Formula/Axiom)



Fact 1: If $G \models \delta_k \wedge H \models \delta_k$ then
Duplicator survives k rounds on G, H

Fact 2: $\mu_\infty(\delta_k) = 1$
(δ_k is almost surely true)

2 cases



a) There is G $G \models \delta_k \wedge \phi \Rightarrow$ (by Fact 1) $\forall H$: If $H \models \delta_k$ then $H \models \phi$

Thus, $\mu_\infty(\delta_k) \leq \mu_\infty(\phi)$

\Rightarrow (by Fact 2) $\mu_\infty(\delta_k) = 1$, hence $\mu_\infty(\phi) = 1$

0-1 Law

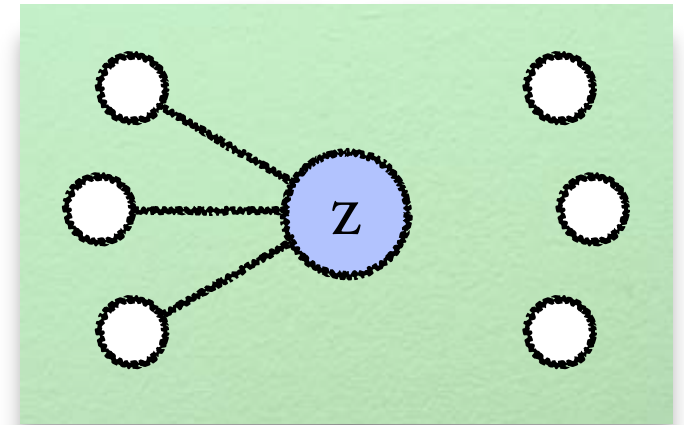


For every *FO sentence* ϕ , $\mu_\infty(\phi)$ is either 0 or 1.

Let k = quantifier rank of ϕ

$$\delta_k = \forall x_1, \dots, x_k \forall y_1, \dots, y_k \exists z \bigwedge_{i,j} x_i \neq y_j \wedge E(x_i, z) \wedge \neg E(y_j, z)$$

(Extension Formula/Axiom)



Fact 1: If $G \models \delta_k \wedge H \models \delta_k$ then
Duplicator survives k rounds on G, H

Fact 2: $\mu_\infty(\delta_k) = 1$
(δ_k is almost surely true)

2 cases

a) There is G $G \models \delta_k \wedge \phi \Rightarrow$ (by Fact 1) $\forall H$: If $H \models \delta_k$ then $H \models \phi$

Thus, $\mu_\infty(\delta_k) \leq \mu_\infty(\phi)$

\Rightarrow (by Fact 2) $\mu_\infty(\delta_k) = 1$, hence $\mu_\infty(\phi) = 1$

b) There is no $G \models \delta_k \wedge \phi \Rightarrow$ (by Fact 2) there is $G \models \delta_k$,

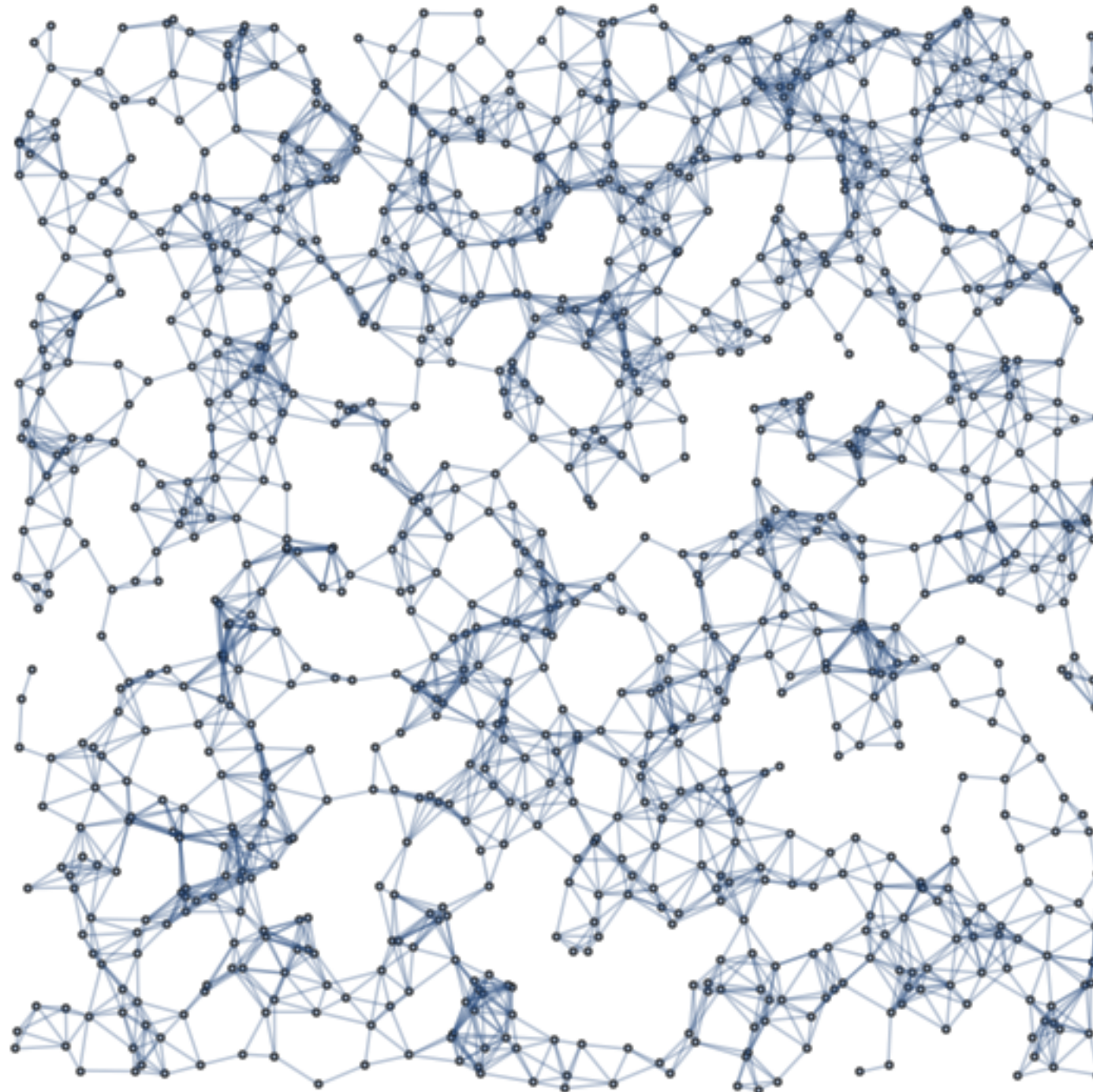
$\Rightarrow G \models \delta_k \wedge \neg\phi \Rightarrow$ (by case a) $\mu_\infty(\neg\phi) = 1$

0-1 Law



For every *FO* sentence ϕ , $\mu_\infty(\phi)$ is either 0 or 1, *and this depends on whether* $\text{RADO} \models \phi$

RADO =



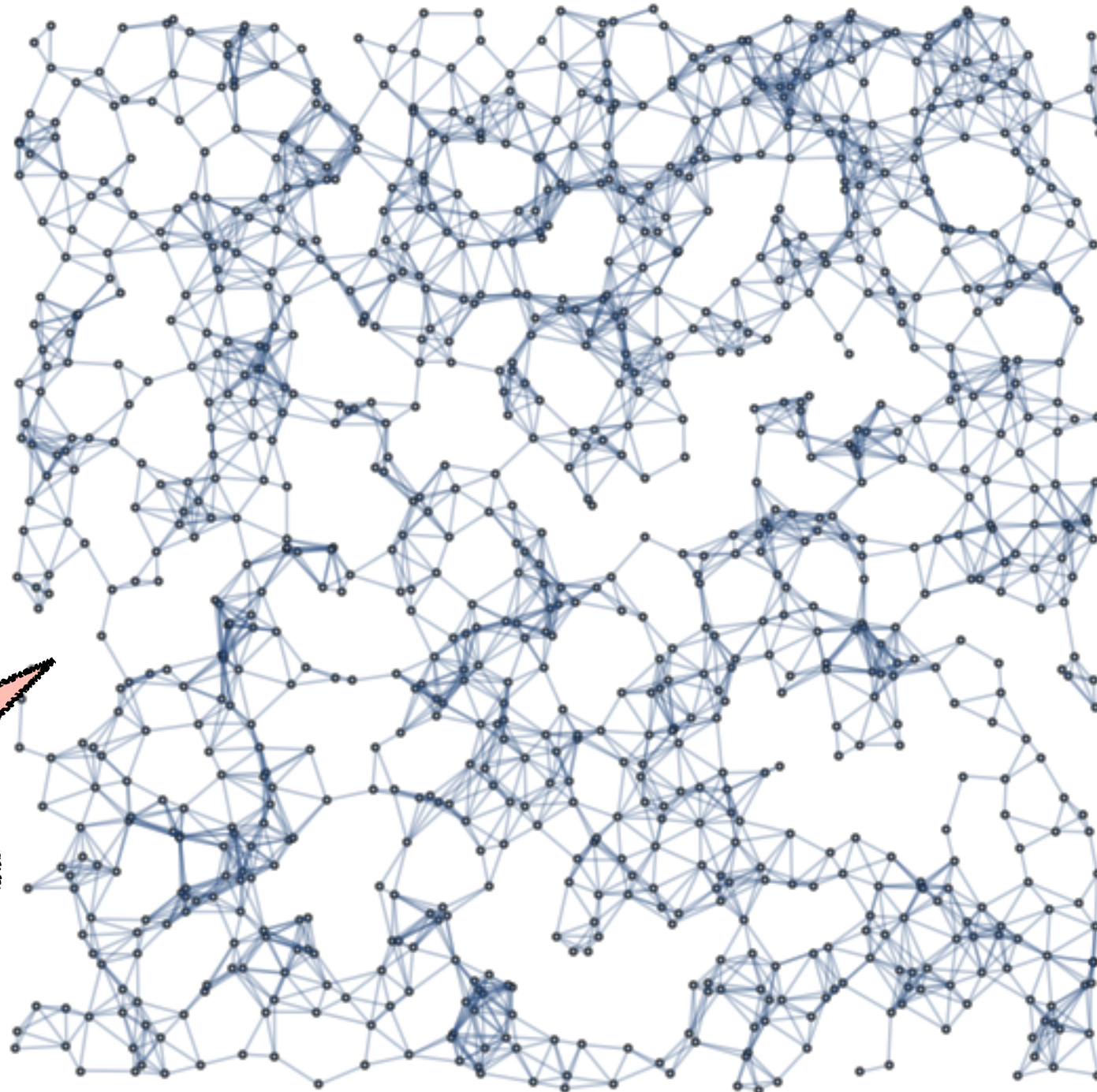
0-1 Law



For every *FO* sentence ϕ , $\mu_\infty(\phi)$ is either 0 or 1, *and this depends on whether* $\text{RADO} \models \phi$

RADO =

each pair of nodes i, j
is connected with
probability $1/2$



0-1 Law

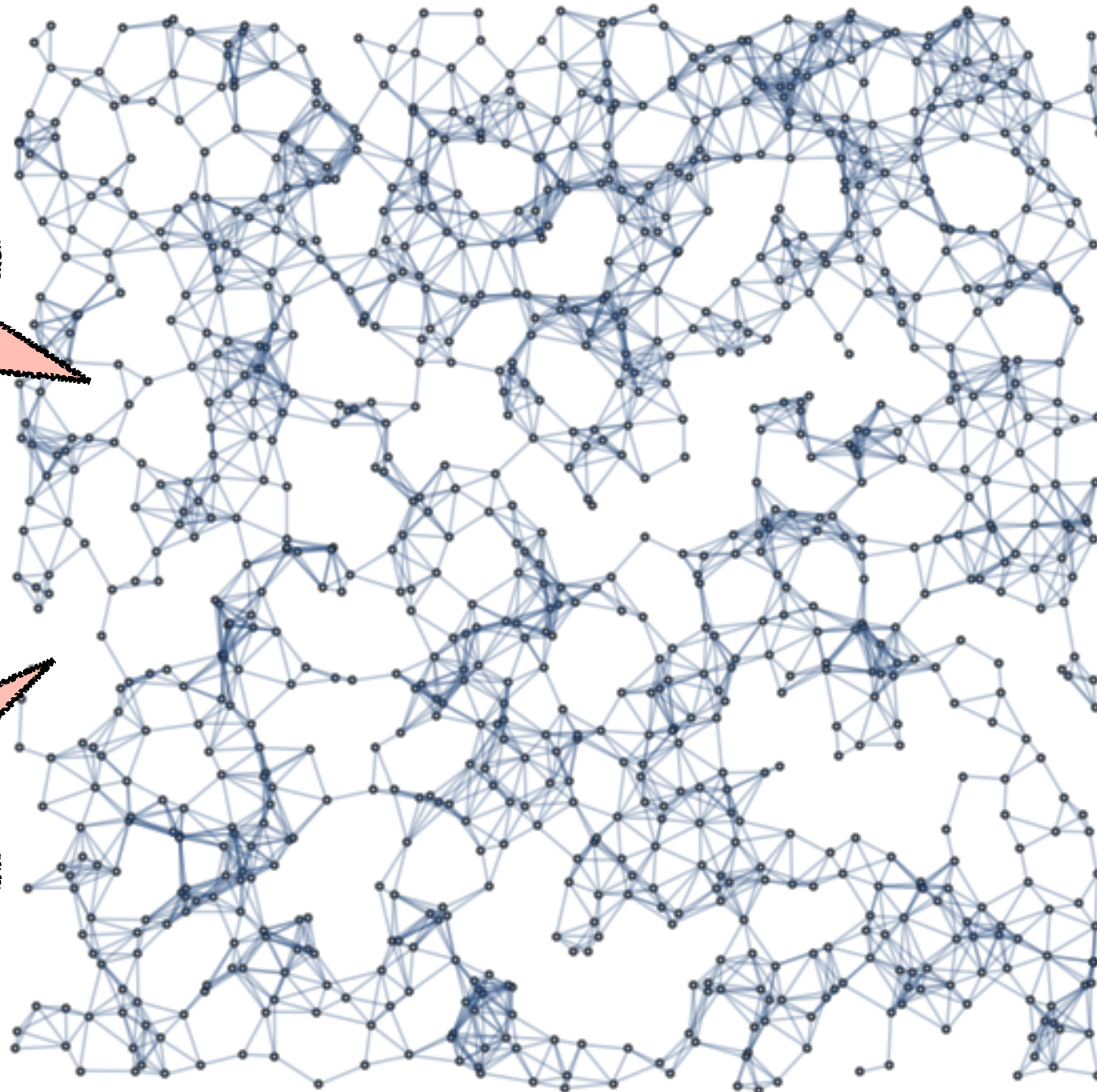


For every *FO* sentence ϕ , $\mu_\infty(\phi)$ is either 0 or 1, *and this depends on whether* $\text{RADO} \models \phi$

each pair of nodes i, j
is connected if
 i -th bit of j is 1

RADO =

each pair of nodes i, j
is connected with
probability $1/2$



0-1 Law



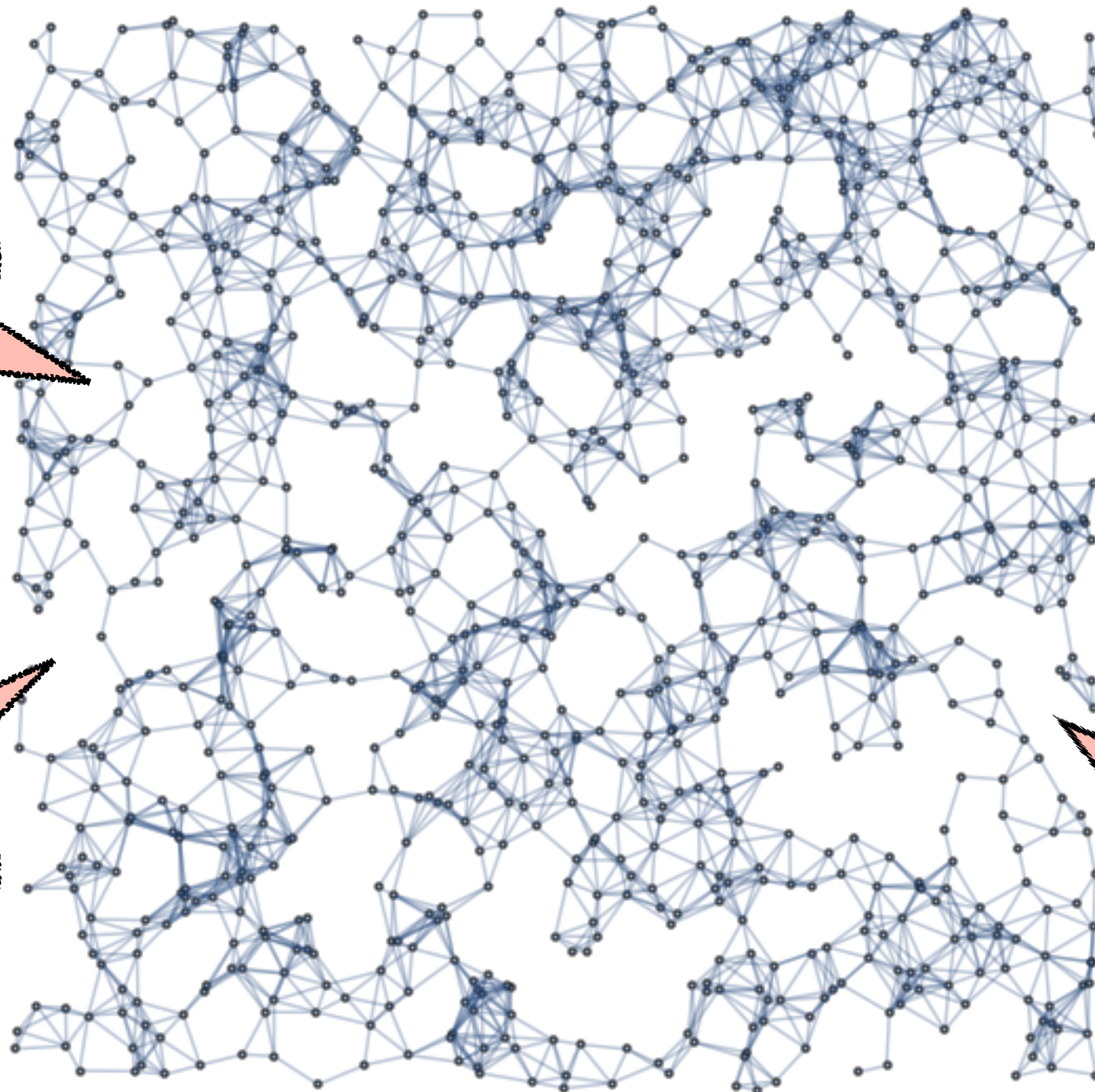
For every *FO* sentence ϕ , $\mu_\infty(\phi)$ is either 0 or 1, *and this depends on whether* $\text{RADO} \models \phi$

each pair of nodes i, j
is connected if
 i -th bit of j is 1

RADO =

each pair of nodes i, j
is connected with
probability $1/2$

the unique
graph that
satisfies
 δ_k for all k

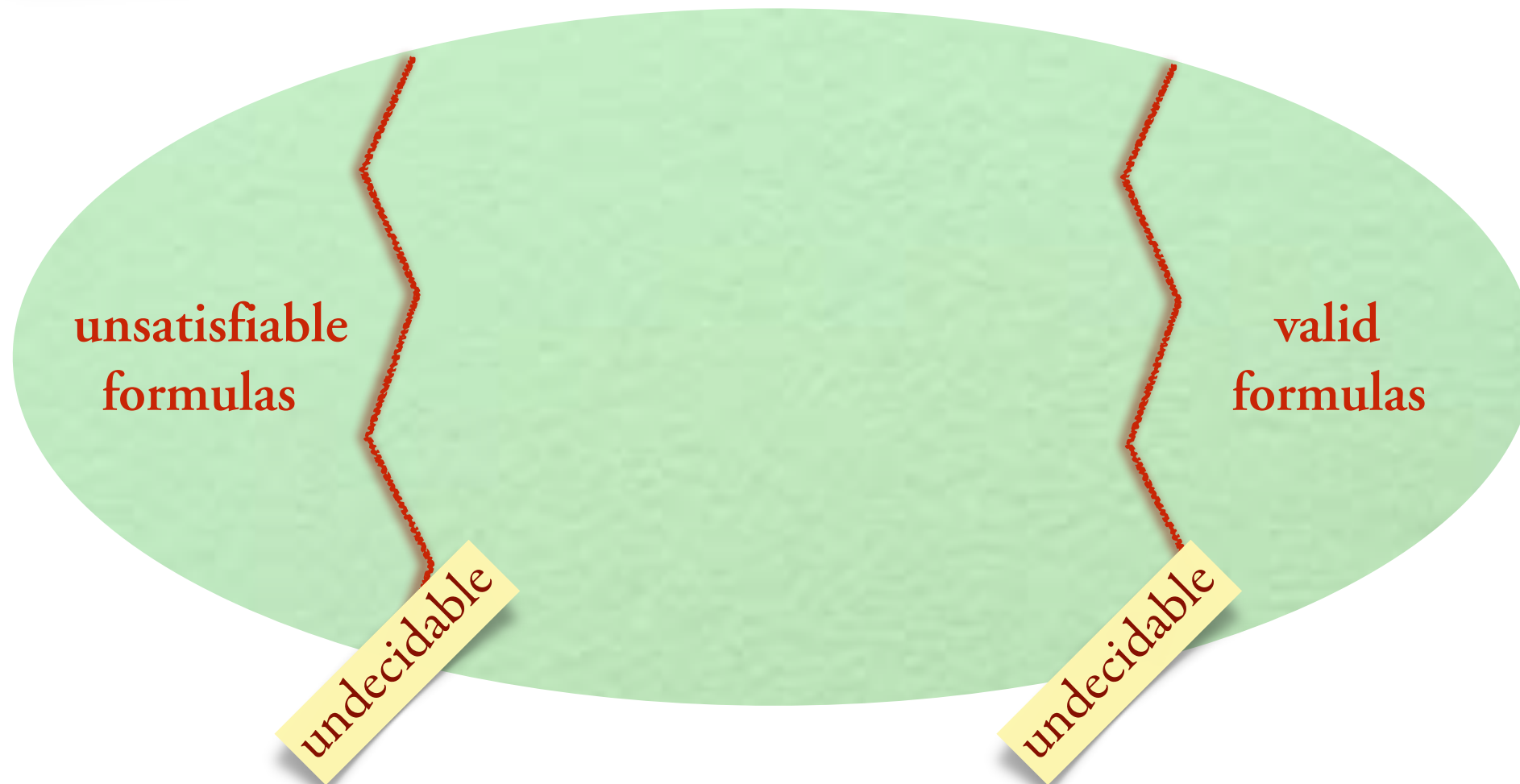


0-1 Law

Theorem. The problem of deciding whether an FO sentence is *almost surely true* ($\mu_\infty = 1$) is PSPACE-complete. [Grandjean '83]

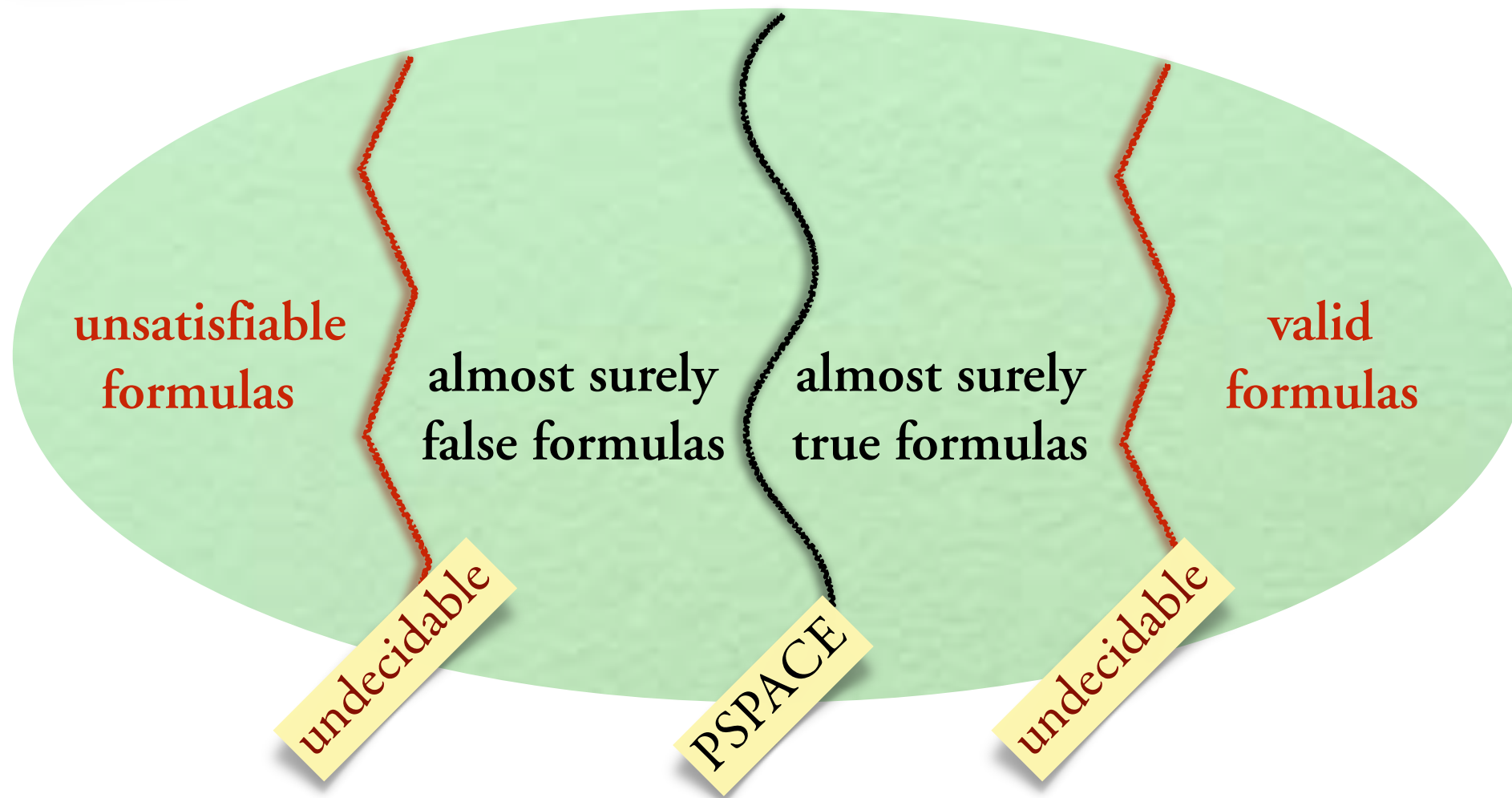
0-1 Law

Theorem. The problem of deciding whether an FO sentence is *almost surely true* ($\mu_\infty = 1$) is PSPACE-complete. [Grandjean '83]



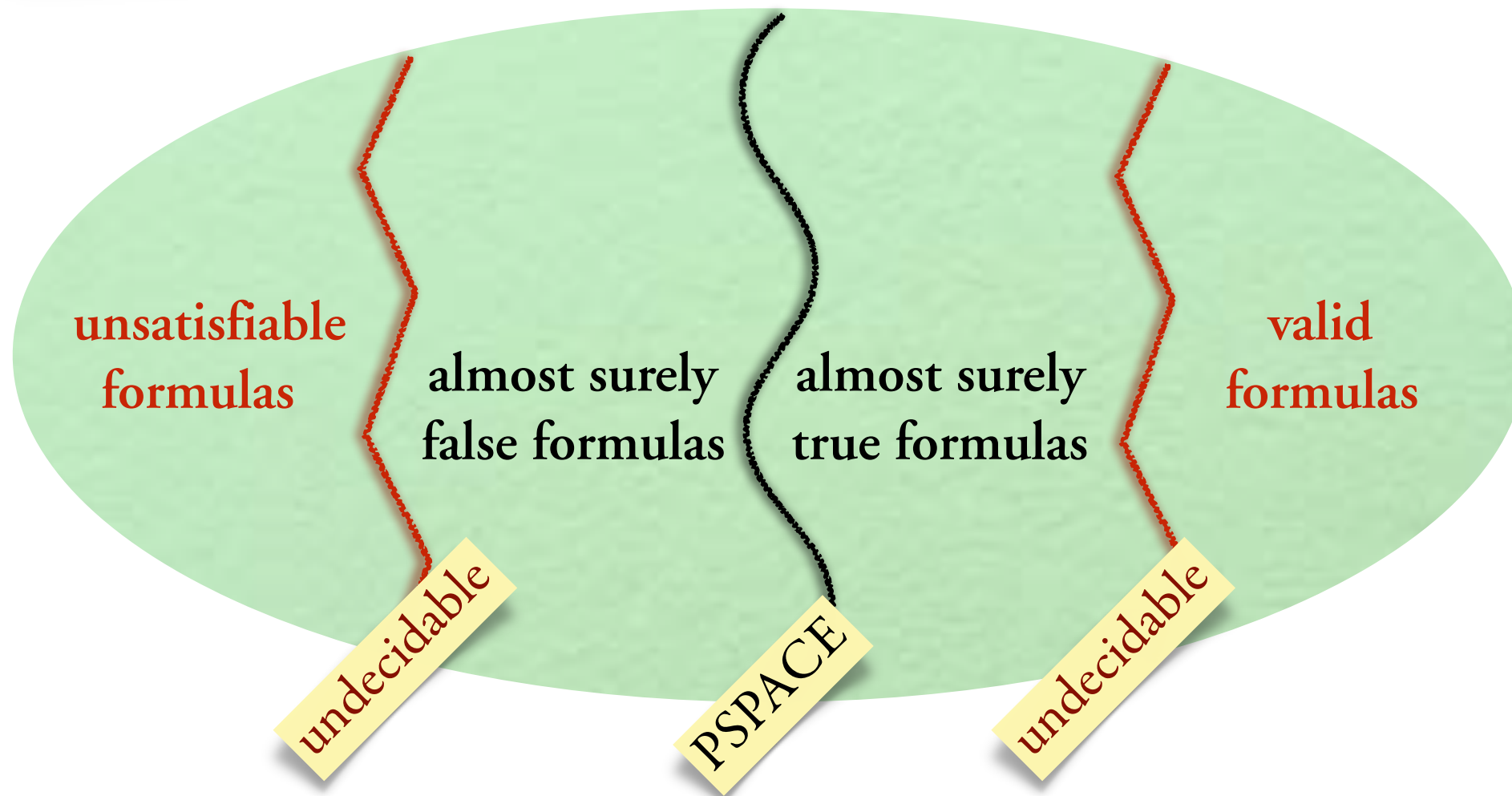
0-1 Law

Theorem. The problem of deciding whether an FO sentence is *almost surely true* ($\mu_\infty = 1$) is PSPACE-complete. [Grandjean '83]



0-1 Law

Theorem. The problem of deciding whether an FO sentence is *almost surely true* ($\mu_\infty = 1$) is PSPACE-complete. [Grandjean '83]



Query evaluation on large databases:

Don't bother evaluating an FO query,
it's either *almost surely true* or *almost surely false*!



0-1 Law

Does the 0-1 Law apply to real-life databases?

Not quite: database *constraints* easily spoil Extension Axiom.

0-1 Law

Does the 0-1 Law apply to real-life databases?

Not quite: database *constraints* easily spoil Extension Axiom.

Consider:

- functional constraint $\forall x, x', y, y' \left(E(x, y) \wedge E(x, y') \Rightarrow y = y' \right) \wedge$
 $\left(E(x, y) \wedge E(x', y) \Rightarrow x = x' \right)$ (E is a permutation)
- FO query $\phi = \neg \exists x E(x, x)$

0-1 Law

Does the 0-1 Law apply to real-life databases?

Not quite: database *constraints* easily spoil Extension Axiom.

Consider:

- functional constraint $\forall x, x', y, y' \left(E(x, y) \wedge E(x, y') \Rightarrow y = y' \right) \wedge$
 $\left(E(x, y) \wedge E(x', y) \Rightarrow x = x' \right)$ (E is a permutation)
- FO query $\phi = \neg \exists x E(x, x)$

Probability that a permutation E satisfies $\phi = n!/n! \rightarrow e^{-1} = 0.3679\dots$

0-1 Law

Does the 0-1 Law apply to real-life databases?

Not quite: database *constraints* easily spoil Extension Axiom.

Consider:

- functional constraint $\forall x, x', y, y' \left(E(x, y) \wedge E(x, y') \Rightarrow y = y' \right) \wedge$
 $\left(E(x, y) \wedge E(x', y) \Rightarrow x = x' \right)$ (E is a permutation)
- FO query $\phi = \neg \exists x E(x, x)$

Probability that a permutation E satisfies $\phi = n! / n! \rightarrow e^{-1} = 0.3679...$

0-1 Law only applies to **unconstrained** databases...

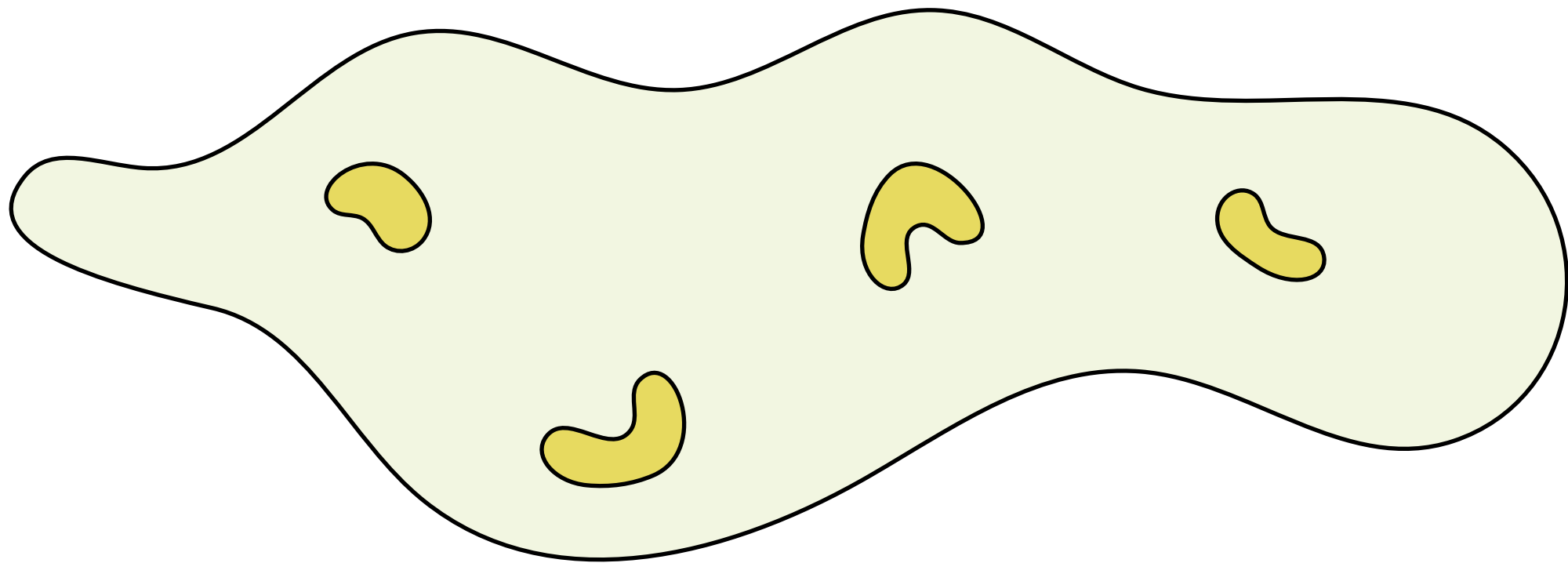
Another technique: Locality

Idea: First order logic can only express “local” properties

Another technique: Locality

Idea: First order logic can only express “local” properties

Local = properties of nodes which are close to one another



Hanf locality

Definition. The **Gaifman graph** of a structure $S = (V, R_1, \dots, R_m)$ is the **undirected** graph

$$G_S = (V, E) \text{ where } E = \{ (u, v) \mid \exists (\dots, u, \dots, v, \dots) \in R_i \text{ for some } i \}$$

Hanf locality

Definition. The **Gaifman graph** of a structure $S = (V, R_1, \dots, R_m)$ is the **undirected** graph $G_S = (V, E)$ where $E = \{ (u, v) \mid \exists (..., u, ..., v, ...) \in R_i \text{ for some } i \}$

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

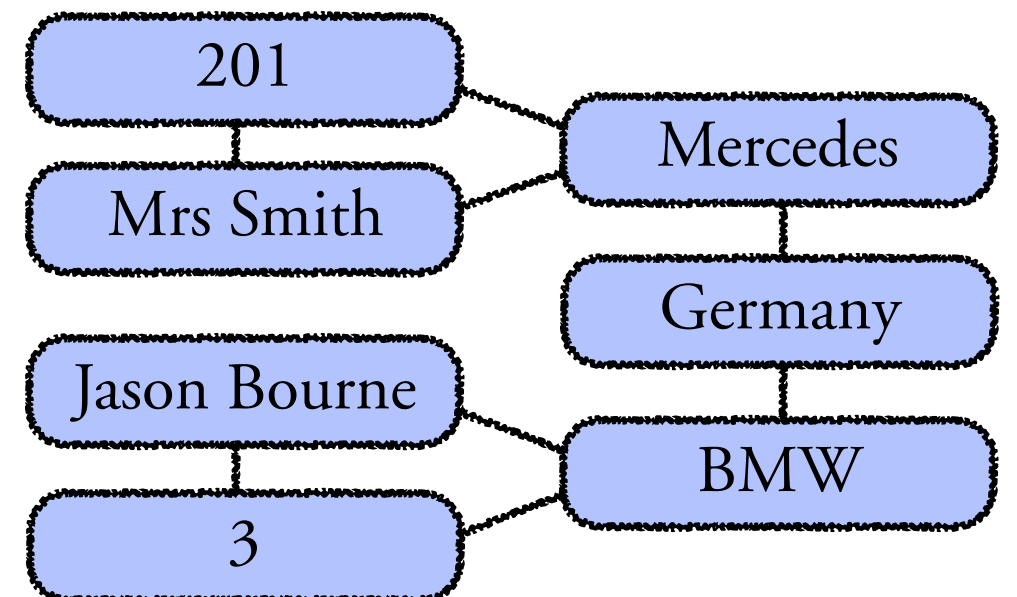
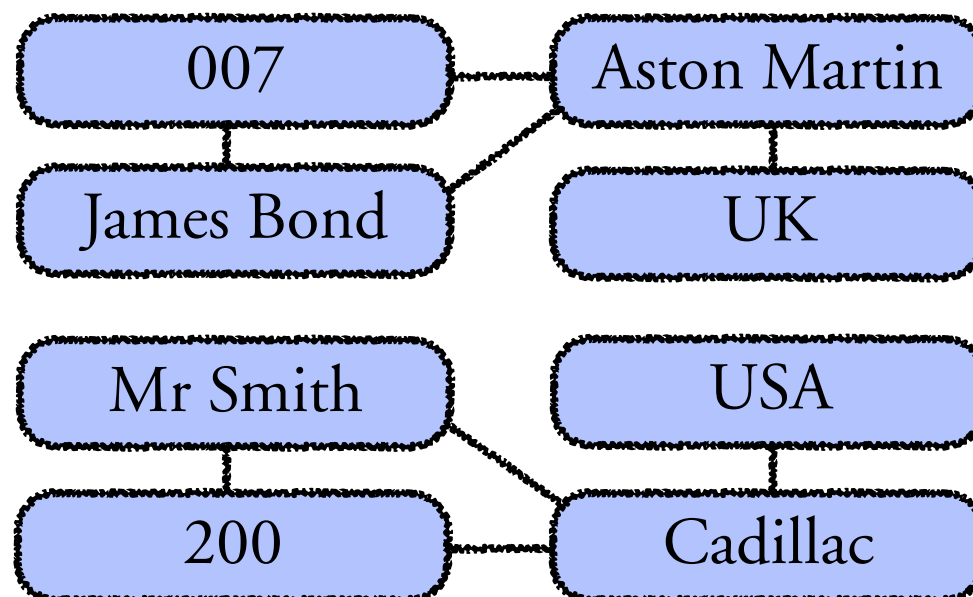
Car	Country
Aston Martin	UK
Cadillac	USA
Mercedes	Germany
BMW	Germany

Hanf locality

Definition. The **Gaifman graph** of a structure $S = (V, R_1, \dots, R_m)$ is the **undirected** graph $G_S = (V, E)$ where $E = \{ (u, v) \mid \exists (..., u, ..., v, ...) \in R_i \text{ for some } i \}$

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

Car	Country
Aston Martin	UK
Cadillac	USA
Mercedes	Germany
BMW	Germany

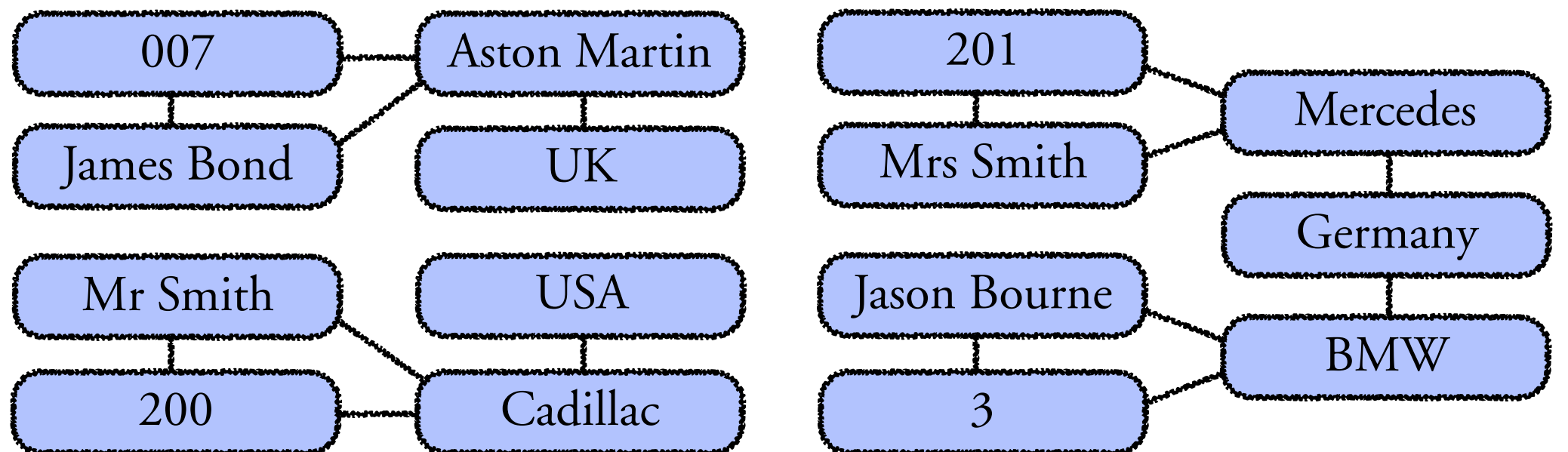


Hanf locality

Definition. The **Gaifman graph** of a structure $S = (V, R_1, \dots, R_m)$ is the **undirected** graph $G_S = (V, E)$ where $E = \{ (u, v) \mid \exists (\dots, u, \dots, v, \dots) \in R_i \text{ for some } i \}$

Agent	Name	Drives	Car	Country
007	James Bond	Aston		UK
200	Mr Smith	Cadill		USA
201	Mrs Smith	Mercedes	Mercedes	Germany
3	Jason Bourne	BMW	BMW	Germany

The Gaifman graph of a graph G is the underlying undirected graph.

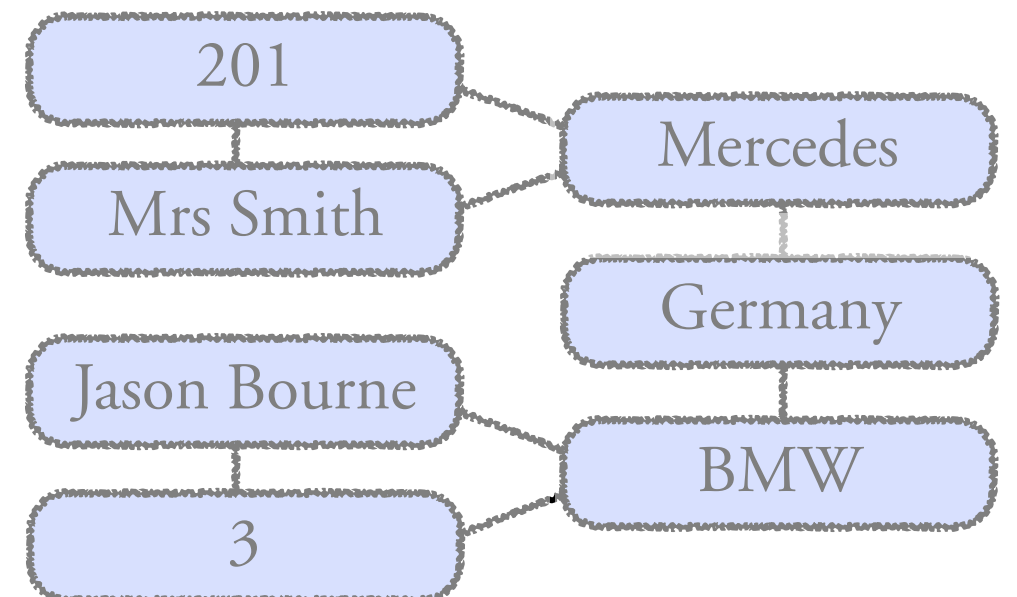
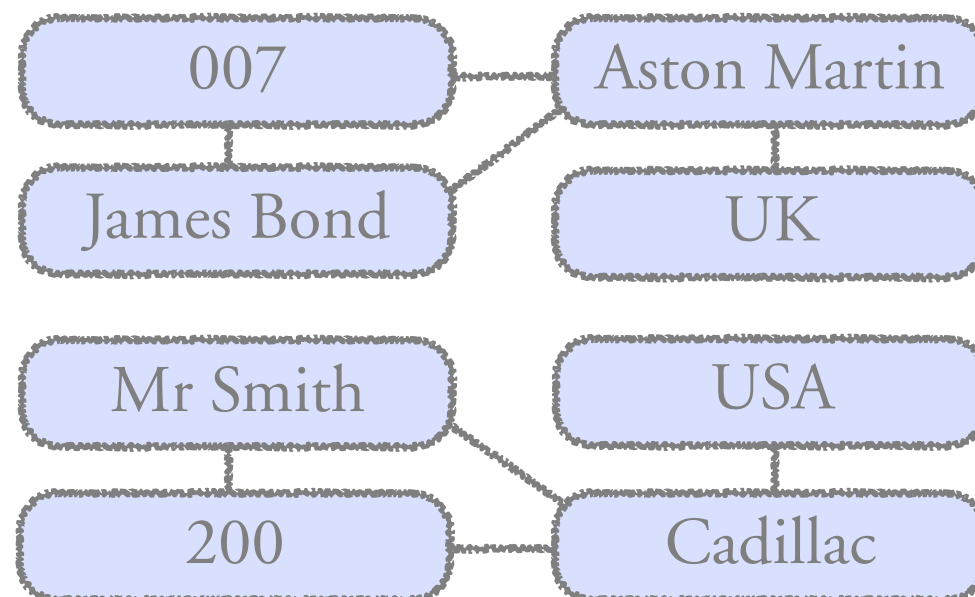


Hanf locality

- $\text{dist}(u, v)$ = **distance** between u and v in the Gaifman graph
- $S[u, r]$ = sub-structure induced by $\{v \mid \text{dist}(u, v) \leq r\}$ = **ball** around u of radius r

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

Car	Country
Aston Martin	UK
Cadillac	USA
Mercedes	Germany
BMW	Germany

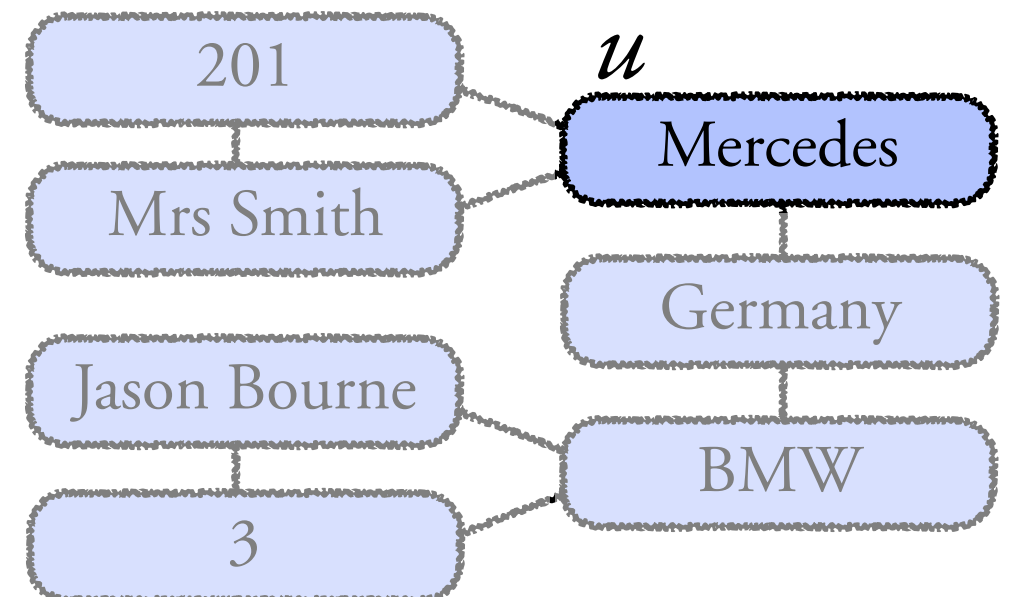
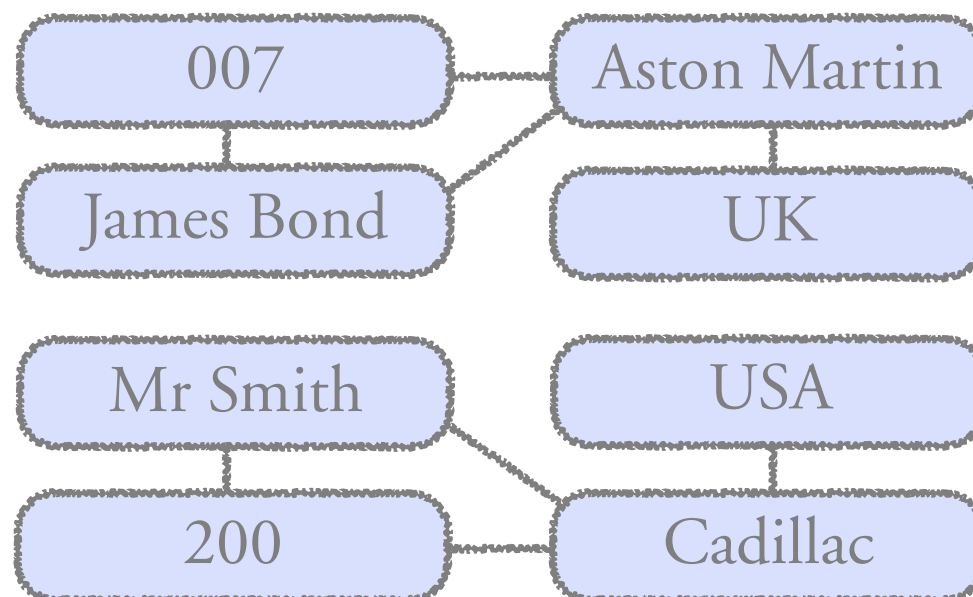


Hanf locality

- $\text{dist}(u, v)$ = **distance** between u and v in the Gaifman graph
- $S[u, r]$ = sub-structure induced by $\{v \mid \text{dist}(u, v) \leq r\}$ = **ball** around u of radius r

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

Car	Country
Aston Martin	UK
Cadillac	USA
u Mercedes	Germany
BMW	Germany

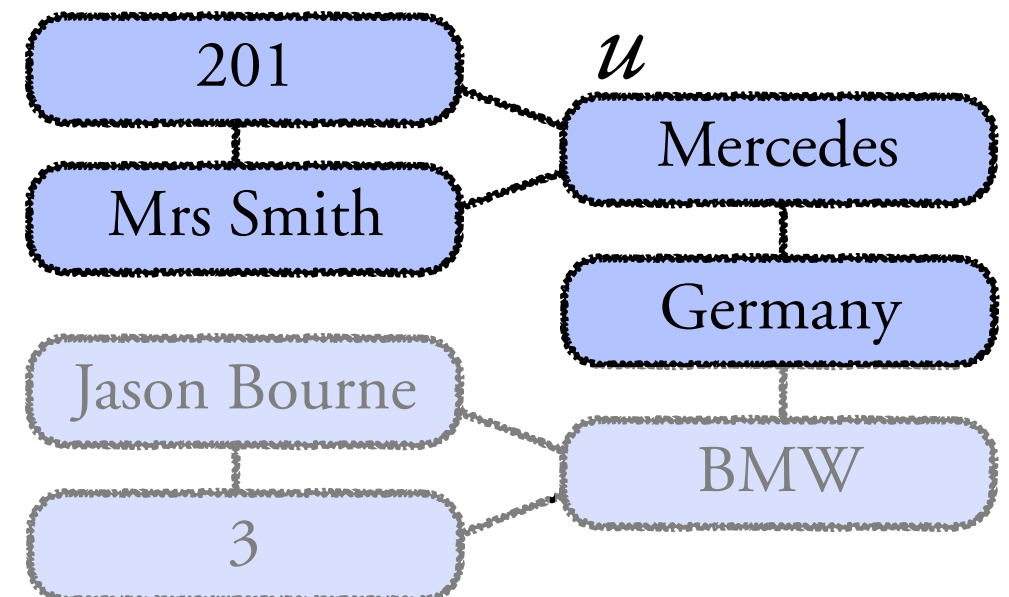
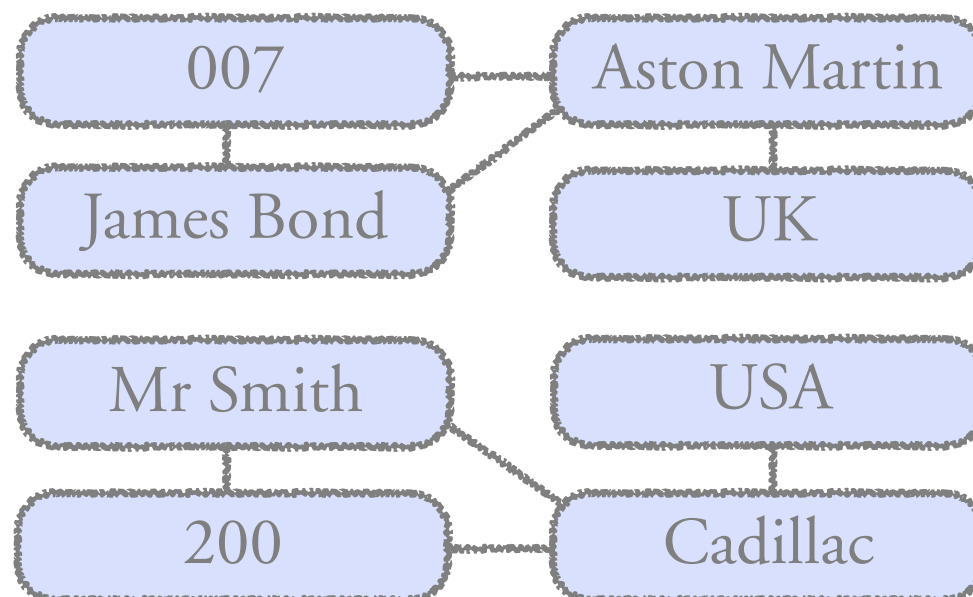


Hanf locality

- $\text{dist}(u, v)$ = **distance** between u and v in the Gaifman graph
- $S[u, r]$ = sub-structure induced by $\{v \mid \text{dist}(u, v) \leq r\}$ = **ball** around u of radius r

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

Car	Country
Aston Martin	UK
Cadillac	USA
u Mercedes	Germany
BMW	Germany

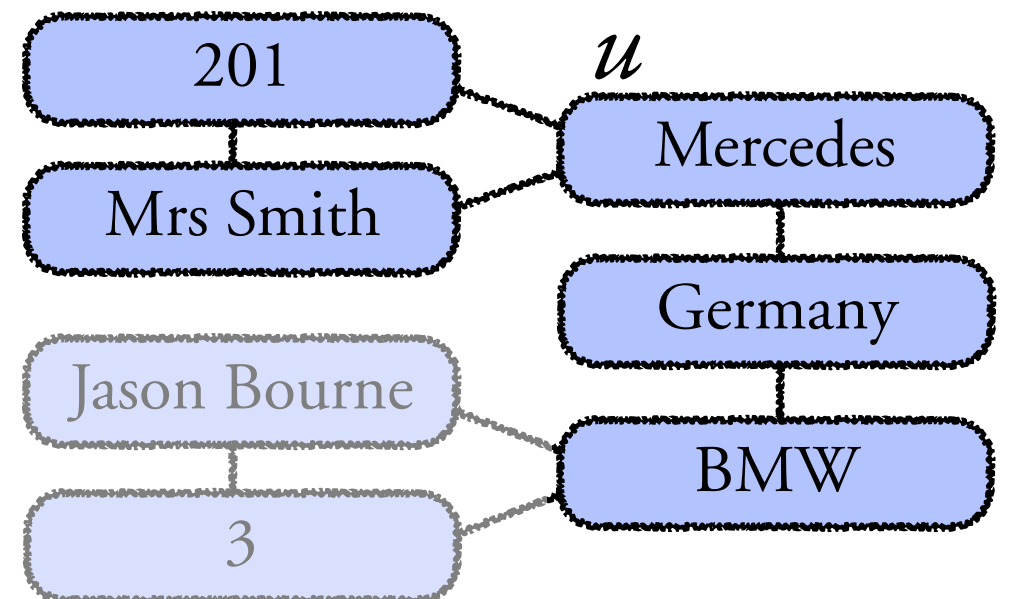
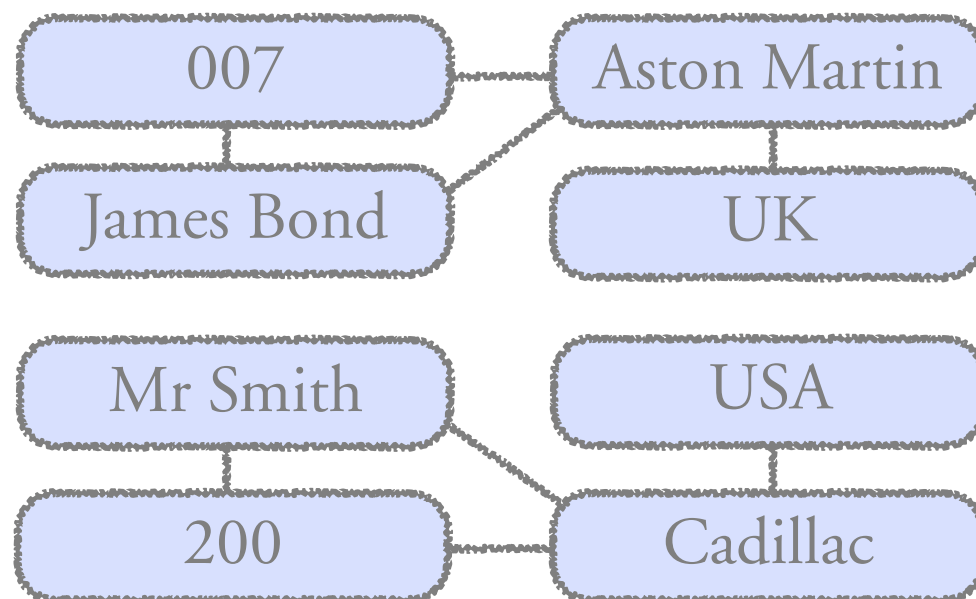


Hanf locality

- $\text{dist}(u, v)$ = **distance** between u and v in the Gaifman graph
- $S[u, r]$ = sub-structure induced by $\{v \mid \text{dist}(u, v) \leq r\}$ = **ball** around u of radius r

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

Car	Country
Aston Martin	UK
Cadillac	USA
u Mercedes	Germany
BMW	Germany

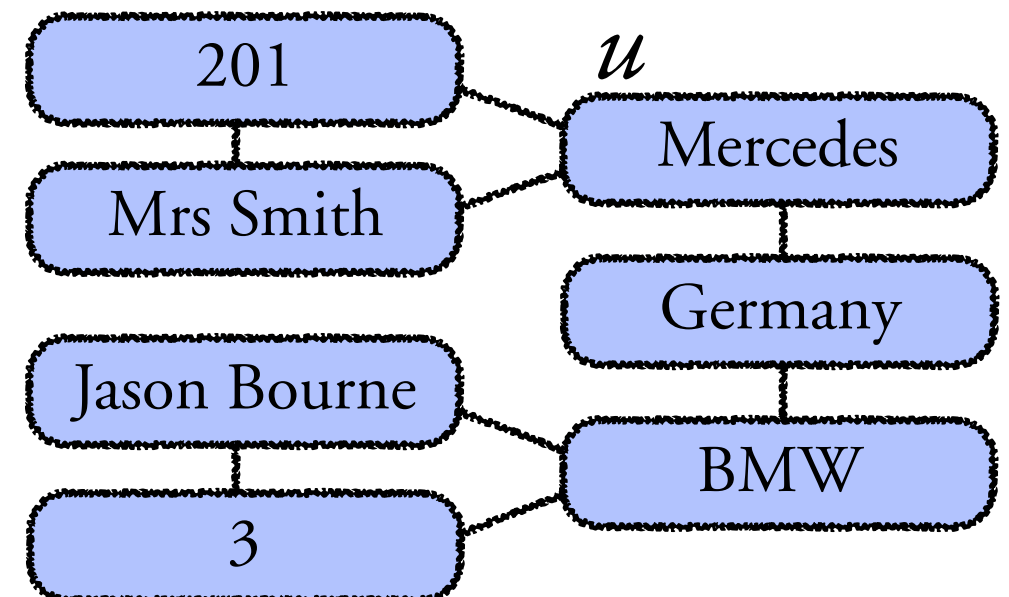
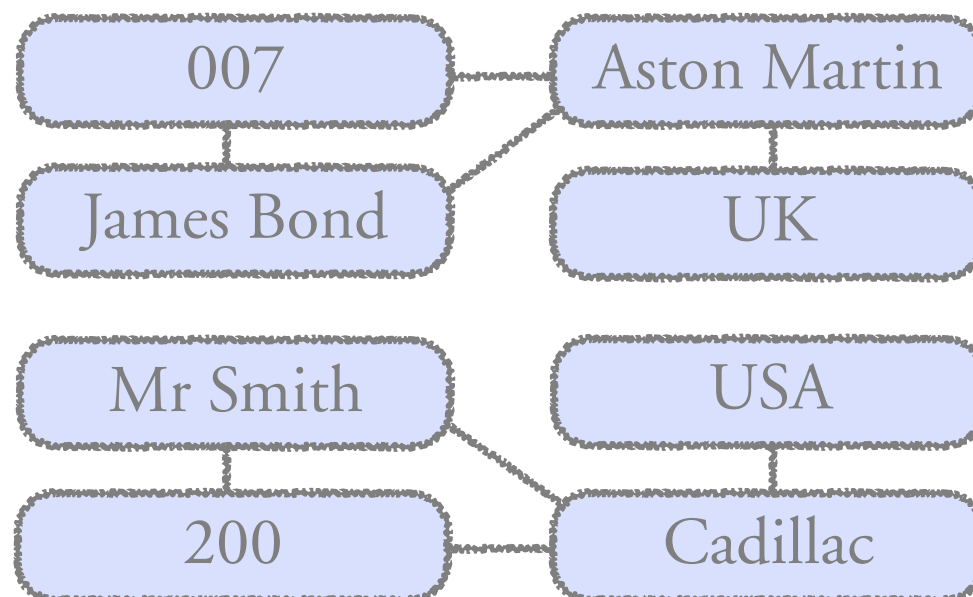


Hanf locality

- $\text{dist}(u, v)$ = **distance** between u and v in the Gaifman graph
- $S[u, r]$ = sub-structure induced by $\{v \mid \text{dist}(u, v) \leq r\}$ = **ball** around u of radius r

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

Car	Country
Aston Martin	UK
Cadillac	USA
u Mercedes	Germany
BMW	Germany



Hanf locality

Definition. Two structures S_1 and S_2 are **Hanf(r, t) - equivalent**

iff for each structure B , the two numbers

$$\#u \text{ s.t. } S_1[u, r] \cong B \qquad \#v \text{ s.t. } S_2[v, r] \cong B$$

are *either the same* or *both* $\geq t$.

Hanf locality

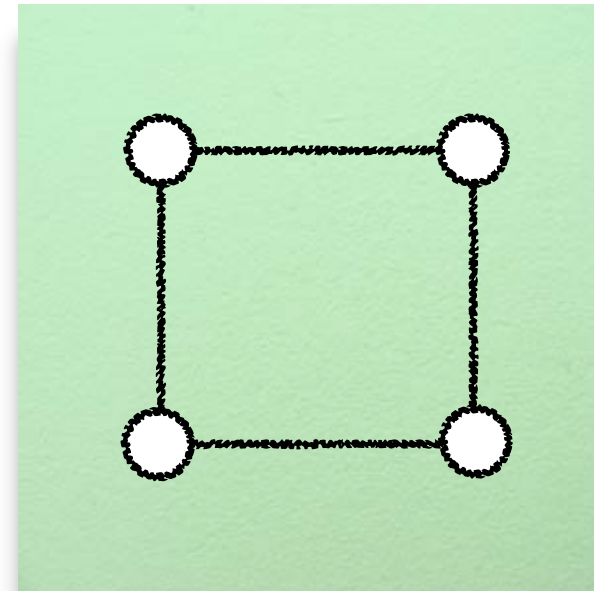
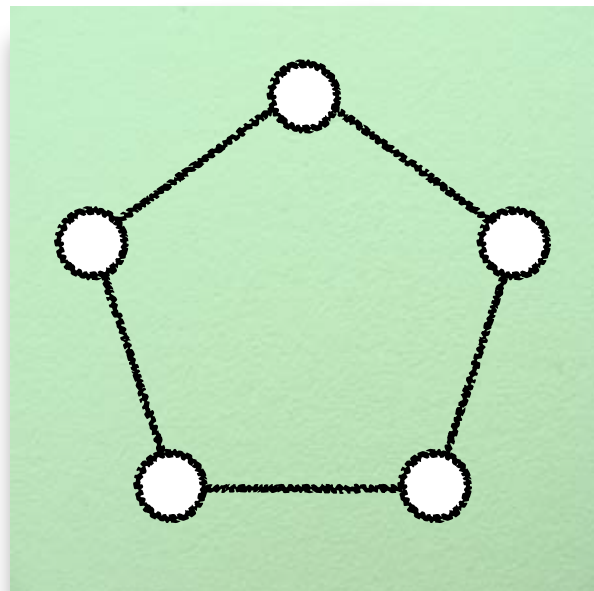
Definition. Two structures S_1 and S_2 are **Hanf(r, t) - equivalent**

iff for each structure B , the two numbers

$$\#u \text{ s.t. } S_1[u, r] \cong B \quad \#v \text{ s.t. } S_2[v, r] \cong B$$

are *either the same* or *both* $\geq t$.

Example. S_1, S_2 are Hanf(1, 1) - equivalent iff they have the *same balls* of radius 1



Hanf locality

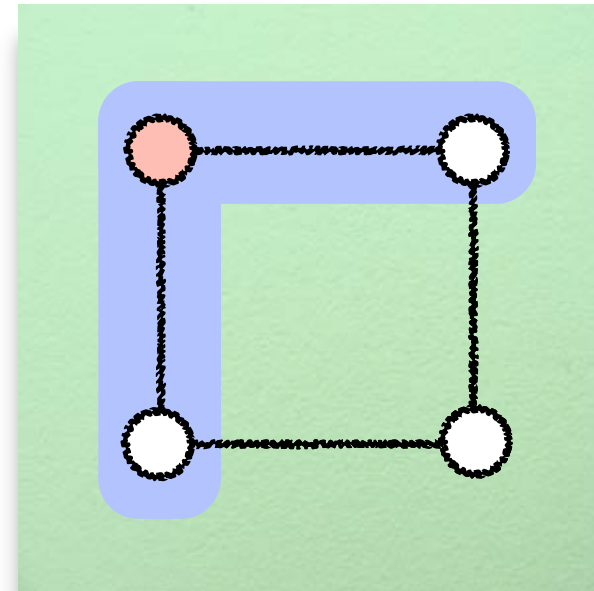
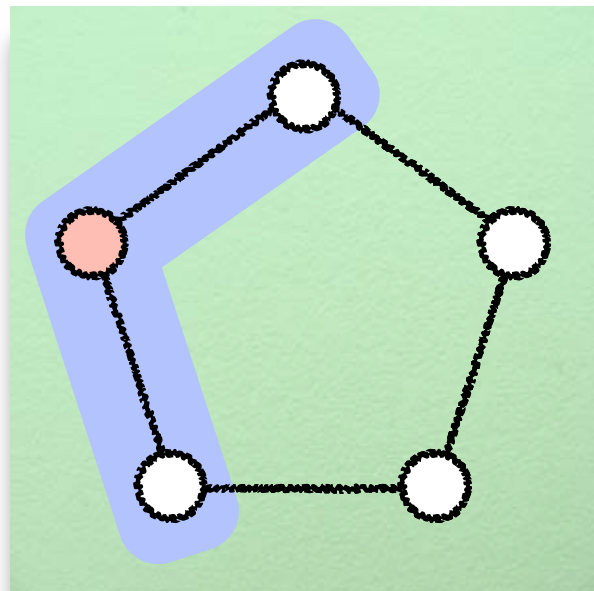
Definition. Two structures S_1 and S_2 are **Hanf(r, t) - equivalent**

iff for each structure B , the two numbers

$$\#u \text{ s.t. } S_1[u, r] \cong B \qquad \#v \text{ s.t. } S_2[v, r] \cong B$$

are *either the same* or *both* $\geq t$.

Example. S_1, S_2 are Hanf(1, 1) - equivalent iff they have the *same balls* of radius 1



Hanf locality

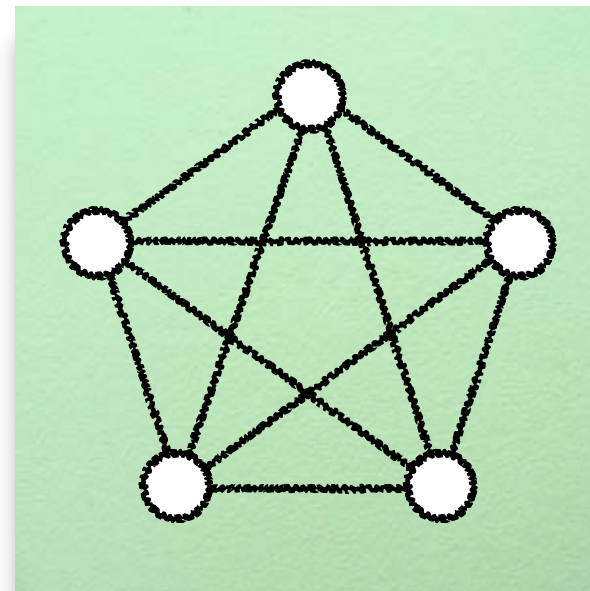
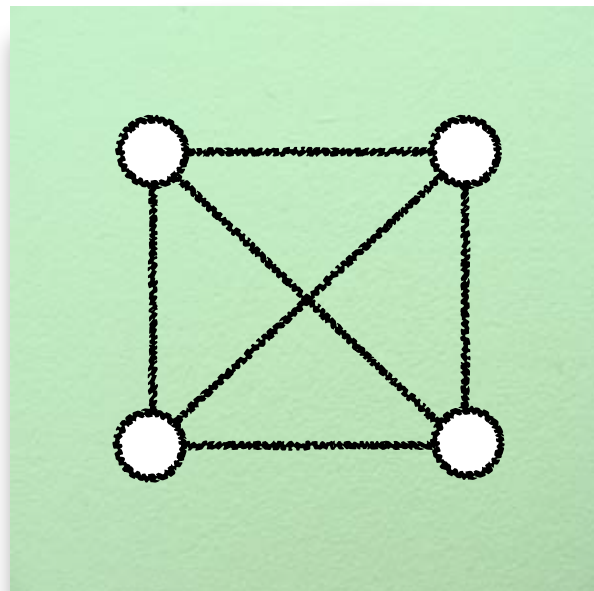
Definition. Two structures S_1 and S_2 are **Hanf(r, t) - equivalent**

iff for each structure B , the two numbers

$$\#u \text{ s.t. } S_1[u, r] \cong B \qquad \#v \text{ s.t. } S_2[v, r] \cong B$$

are *either the same* or *both* $\geq t$.

Example. K_n, K_{n+1} are **not** Hanf(1, 1) - equivalent



Hanf locality

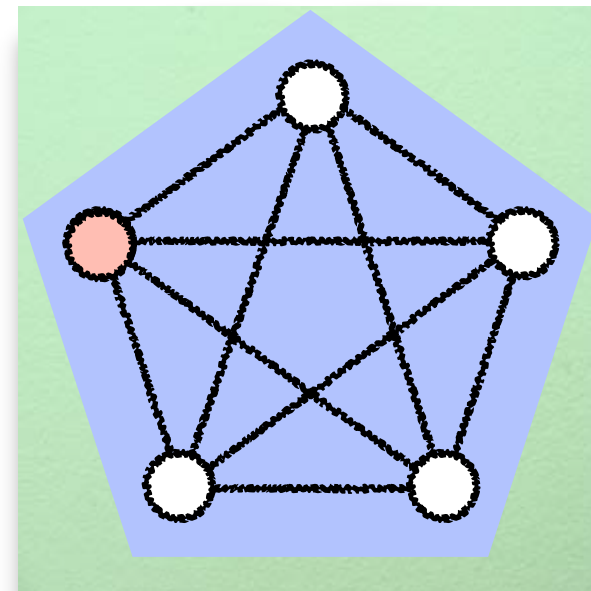
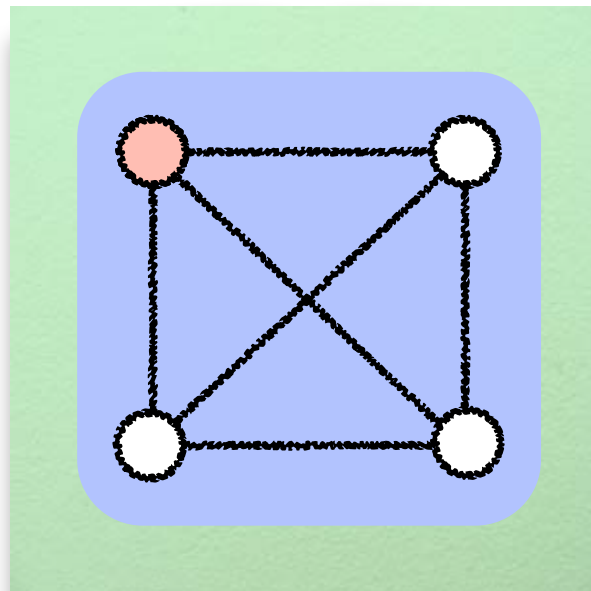
Definition. Two structures S_1 and S_2 are **Hanf(r, t) - equivalent**

iff for each structure B , the two numbers

$$\#u \text{ s.t. } S_1[u, r] \cong B \quad \#v \text{ s.t. } S_2[v, r] \cong B$$

are *either the same* or *both* $\geq t$.

Example. K_n, K_{n+1} are **not** Hanf(1, 1) - equivalent



Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Exercise: prove that *acyclicity* is not FO-definable (on finite structures)

Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Exercise: prove that *acyclicity* is not FO-definable (on finite structures)

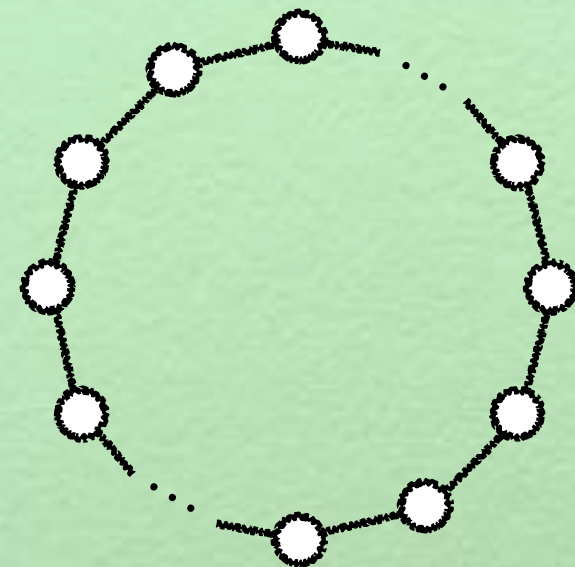


Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Exercise: prove that *acyclicity* is not FO-definable (on finite structures)

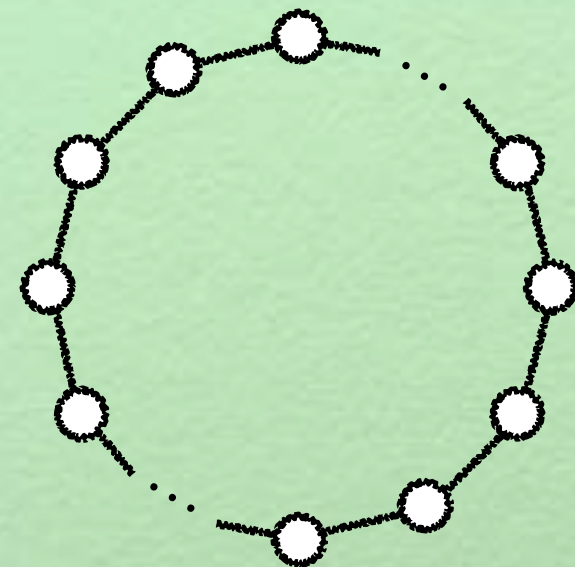


Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Exercise: prove that *acyclicity* is not FO-definable (on finite structures)

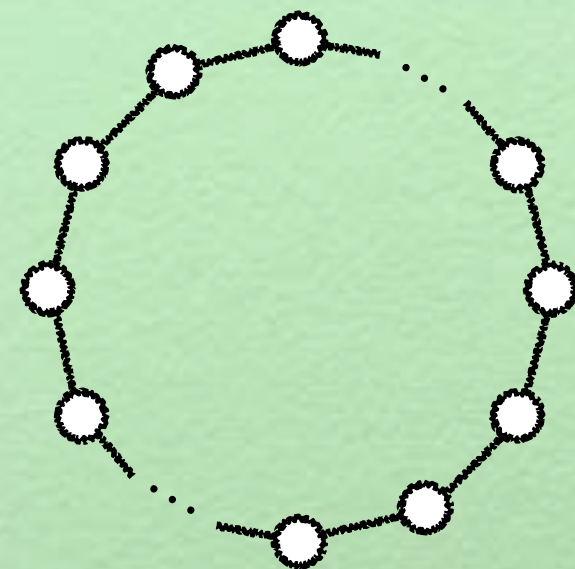


Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Exercise: prove that *acyclicity* is not FO-definable (on finite structures)

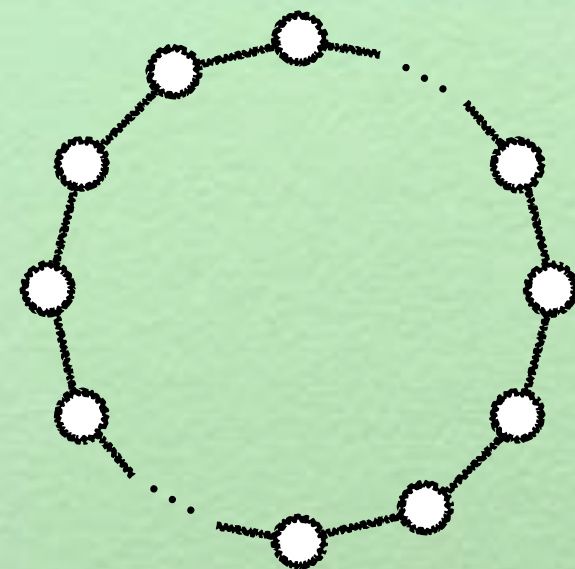


Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Exercise: prove that *acyclicity* is not FO-definable (on finite structures)

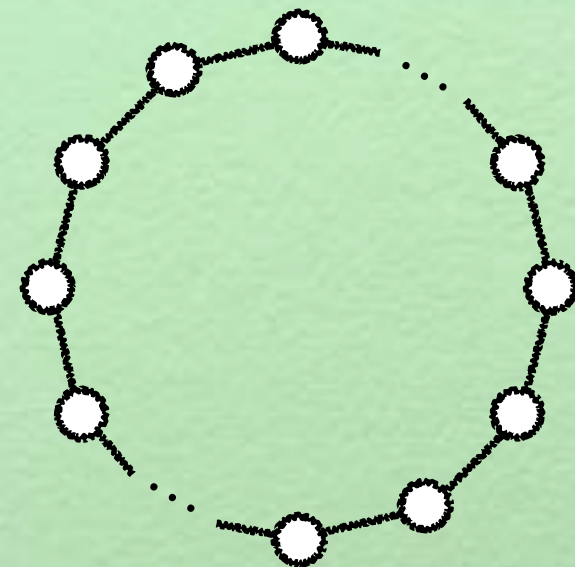


Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Exercise: prove that *acyclicity* is not FO-definable (on finite structures)

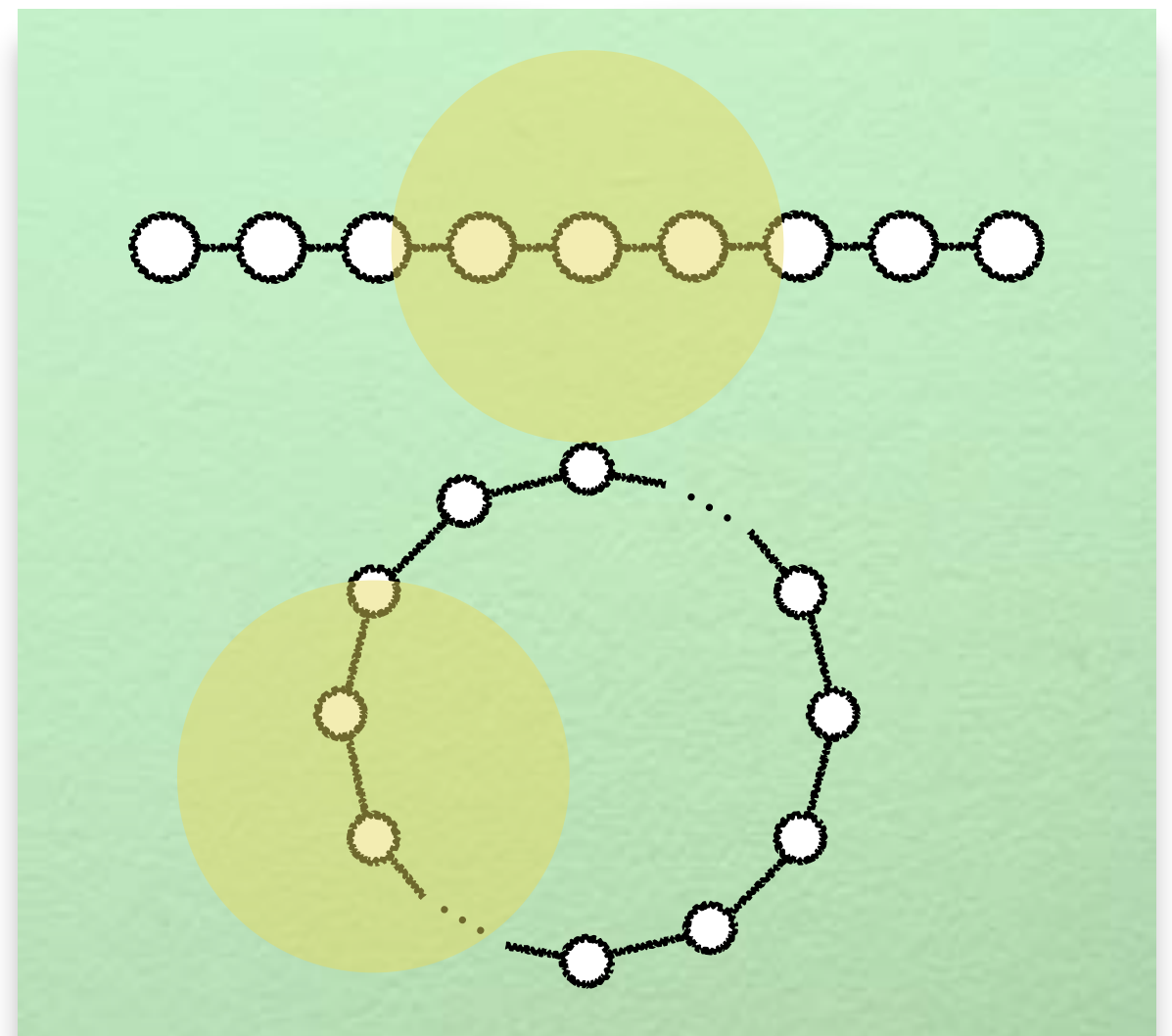
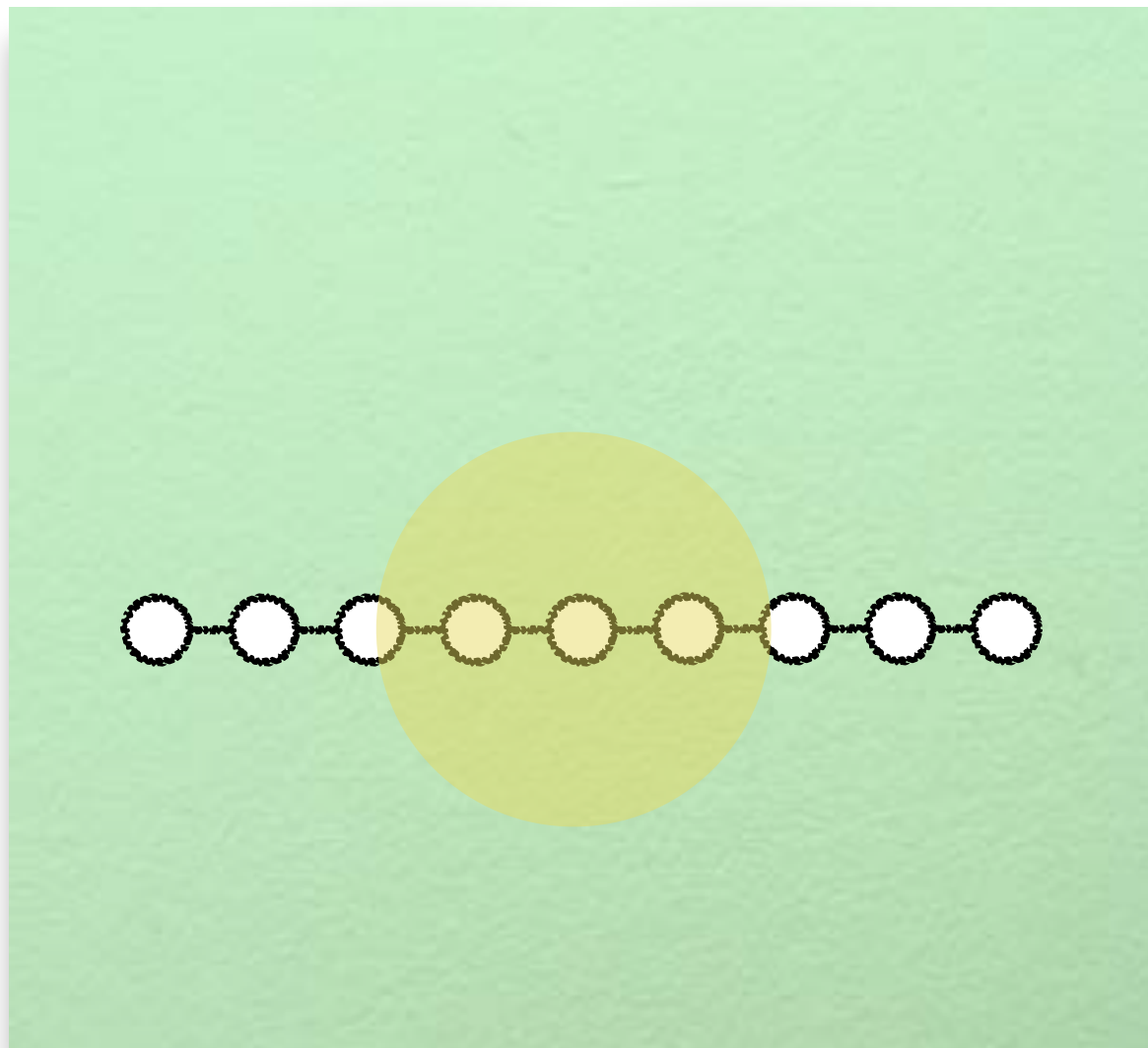


Hanf locality

Theorem. If S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$
then S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)

[Hanf '60]

Exercise: prove that *acyclicity* is not FO-definable (on finite structures)



Hanf locality

Theorem. S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)
whenever S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$. [Hanf '60]

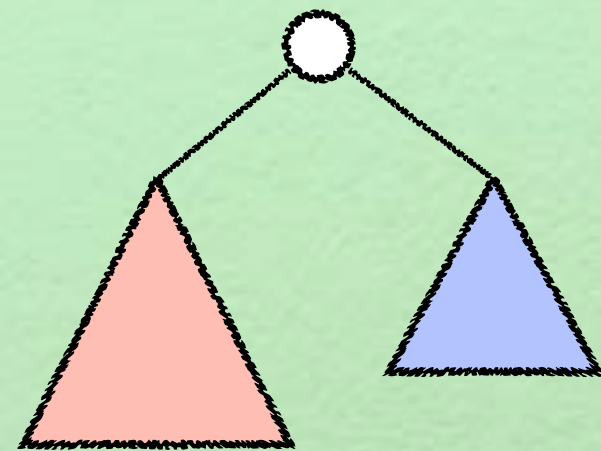
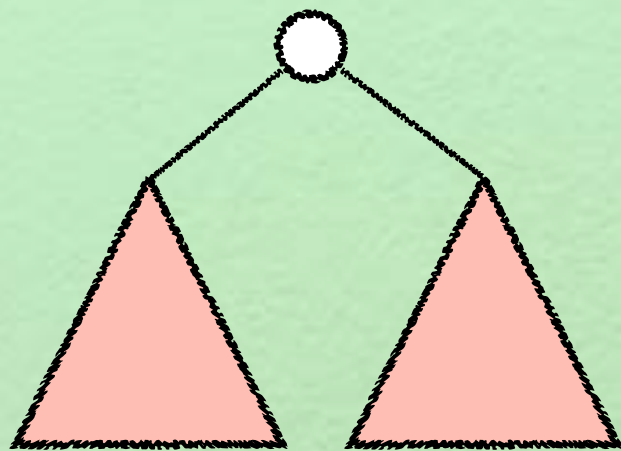
Exercise: prove that testing whether a binary tree is *complete* is not FO-definable

Hanf locality

Theorem. S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)
whenever S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$.

[Hanf '60]

Exercise: prove that testing whether a binary tree is *complete* is not FO-definable



Hanf locality

Theorem. S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)
whenever S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$. [Hanf '60]

Hanf locality

Theorem. S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)
whenever S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$. [Hanf '60]

Why so **BIG**?

Hanf locality

Theorem. S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)
whenever S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$. [Hanf '60]

Why so **BIG**?

Remember $\phi_k(x, y)$ = “there is a path of length 2^k from x to y ”

Hanf locality

Theorem. S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)
whenever S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$. [Hanf '60]

Why so **BIG**?

Remember $\phi_k(x, y)$ = “there is a path of length 2^k from x to y ”

$$\phi_0(x, y) = E(x, y), \text{ and}$$

$$\phi_k(x, y) = \exists z (\phi_{k-1}(x, z) \wedge \phi_{k-1}(z, y))$$

$$\text{qr}(\phi_k) = k$$

Hanf locality

Theorem. S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)
whenever S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$.

[Hanf '60]

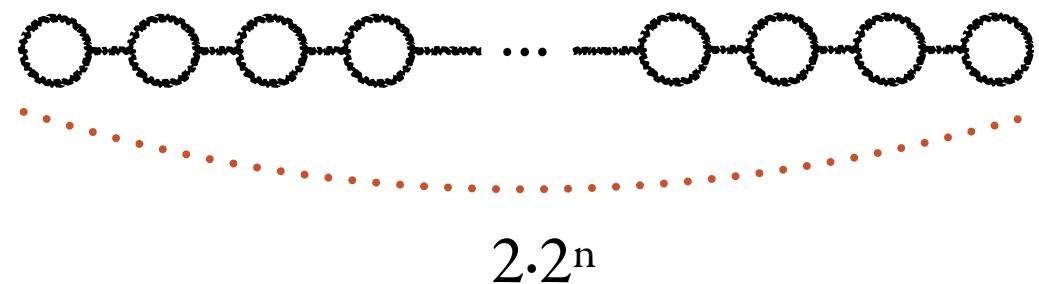
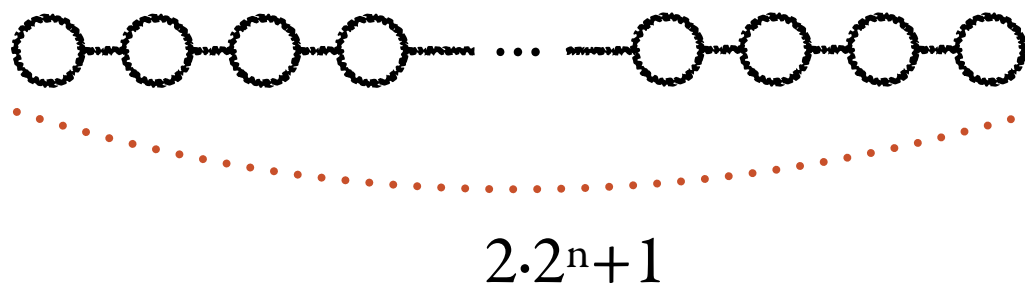
Why so **BIG**?

Remember $\phi_k(x, y)$ = “there is a path of length 2^k from x to y ”

$$\phi_0(x, y) = E(x, y), \text{ and}$$

$$\phi_k(x, y) = \exists z (\phi_{k-1}(x, z) \wedge \phi_{k-1}(z, y))$$

$$\text{qr}(\phi_k) = k$$



Hanf locality

Theorem. S_1, S_2 are n -equivalent (they satisfy the same sentences with quantifier rank n)
whenever S_1, S_2 are $\text{Hanf}(r, t)$ -equivalent, with $r = 3^n$ and $t = n$. [Hanf '60]

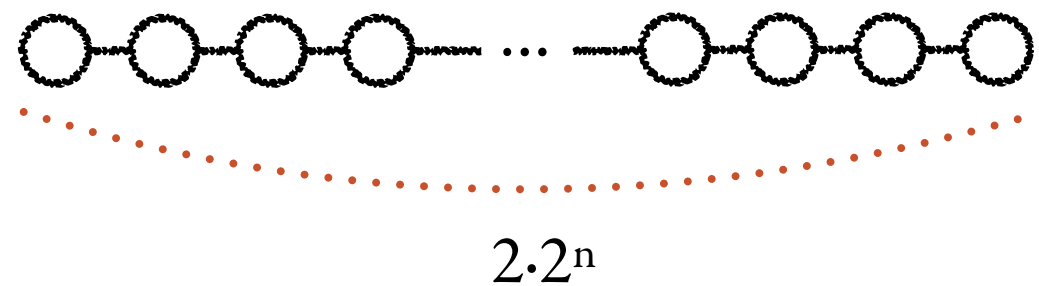
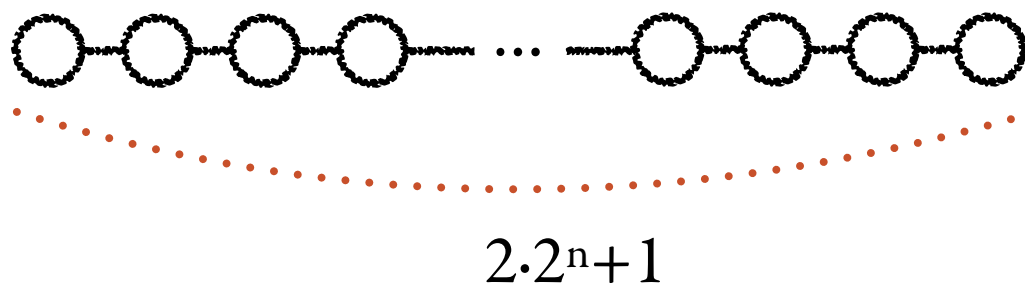
Why so **BIG**?

Remember $\phi_k(x, y)$ = “there is a path of length 2^k from x to y ”

$$\phi_0(x, y) = E(x, y), \text{ and}$$

$$\phi_k(x, y) = \exists z (\phi_{k-1}(x, z) \wedge \phi_{k-1}(z, y))$$

$$\text{qr}(\phi_k) = k$$



Not $(n+2)$ -equivalent yet they have the same $2^n - 1$ balls.

Gaifman locality

What about queries?

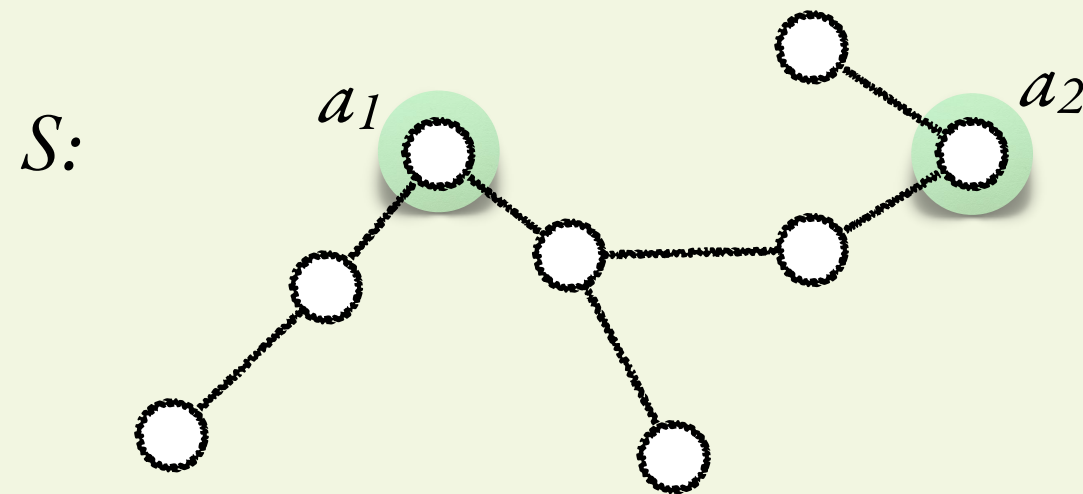
Eg: Is reachability expressible in FO?

What about equivalence on the same structure?

When are two points indistinguishable?

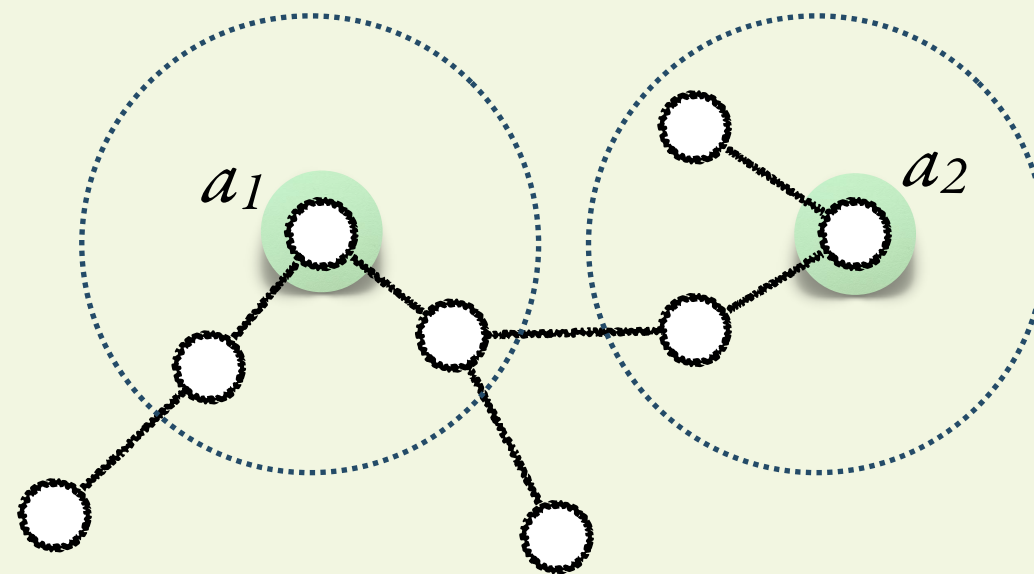
Gaifman locality

$S[(a_1, \dots, a_n), r]$ = induced substructure of S
of elements at distance $\leq r$ of some a_i in the Gaifman graph.



Gaifman locality

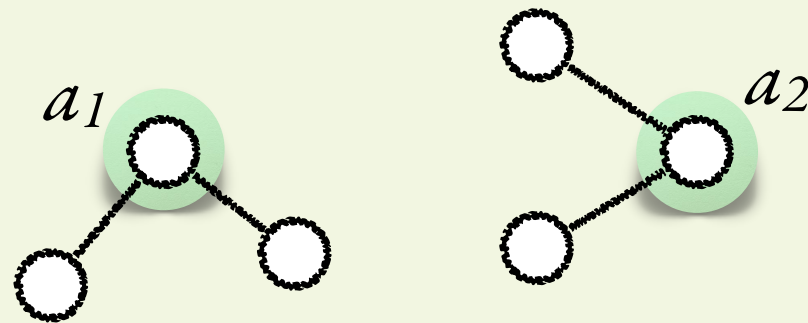
$S[(a_1, \dots, a_n), r]$ = induced substructure of S
of elements at distance $\leq r$ of some a_i in the Gaifman graph.



$S[(a_1, a_2), 1]$

Gaifman locality

$S[(a_1, \dots, a_n), r]$ = induced substructure of S
of elements at distance $\leq r$ of some a_i in the Gaifman graph.



$S[(a_1, a_2), 1]$

Gaifman locality

$S[(a_1, \dots, a_n), r]$ = induced substructure of S
of elements at distance $\leq r$ of some a_i in the Gaifman graph.

Gaifman locality

For any $\phi \in \text{FO}$ of quantifier rank k and structure S ,

$$S[(a_1, \dots, a_n), r] \cong S[(b_1, \dots, b_n), r] \text{ for } r = 3^{k+1}$$

implies

$$(a_1, \dots, a_n) \in \phi(S) \text{ iff } (b_1, \dots, b_n) \in \phi(S)$$

Idea: If the neighbourhoods of two tuples are the same,
the formula cannot distinguish them.

Gaifman locality vs Hanf locality

Difference between Hanf- and Gaifman-locality:

Hanf-locality relates **two different structures**,

Gaifman-locality talks about definability in **one structure**

Gaifman locality vs Hanf locality

Difference between Hanf- and Gaifman-locality:

Hanf-locality relates **two different structures**,

Gaifman-locality talks about definability in **one structure**

Inside S ,

3^{k+1} -balls of $(a_1, \dots, a_n) = 3^{k+1}$ -balls of (b_1, \dots, b_n)

\Downarrow

(a_1, \dots, a_n) indistinguishable from (b_1, \dots, b_n)
through **formulas of $qr \leq k$**

Gaifman locality vs Hanf locality

Difference between Hanf- and Gaifman-locality:

Hanf-locality relates **two different structures**,

S_1 and S_2 have the same # of balls of radius 3^k , **up to threshold k**



They verify the same sentences of $qr \leq k$

Gaifman-locality talks about definability in **one structure**

Inside S ,
 3^{k+1} -balls of $(a_1, \dots, a_n) = 3^{k+1}$ -balls of (b_1, \dots, b_n)



(a_1, \dots, a_n) indistinguishable from (b_1, \dots, b_n) through formulas of $qr \leq k$

Gaifman locality

Schema to show non-expressibility results is, as usual:

A query $Q(x_1, \dots, x_n)$ is not FO-definable if:

for every k there is a structure S_k and $(a_1, \dots, a_n), (b_1, \dots, b_n)$ such that

- $S_k[(a_1, \dots, a_n), 3^{k+1}] \cong S_k[(b_1, \dots, b_n), 3^{k+1}]$
- $(a_1, \dots, a_n) \in Q(S_k), (b_1, \dots, b_n) \notin Q(S_k)$

Proof: If Q were expressible with a formula of quantifier rank k ,

then $(a_1, \dots, a_n) \in Q(S_k)$ iff $(b_1, \dots, b_n) \in Q(S_k)$. Absurd!

Gaifman locality

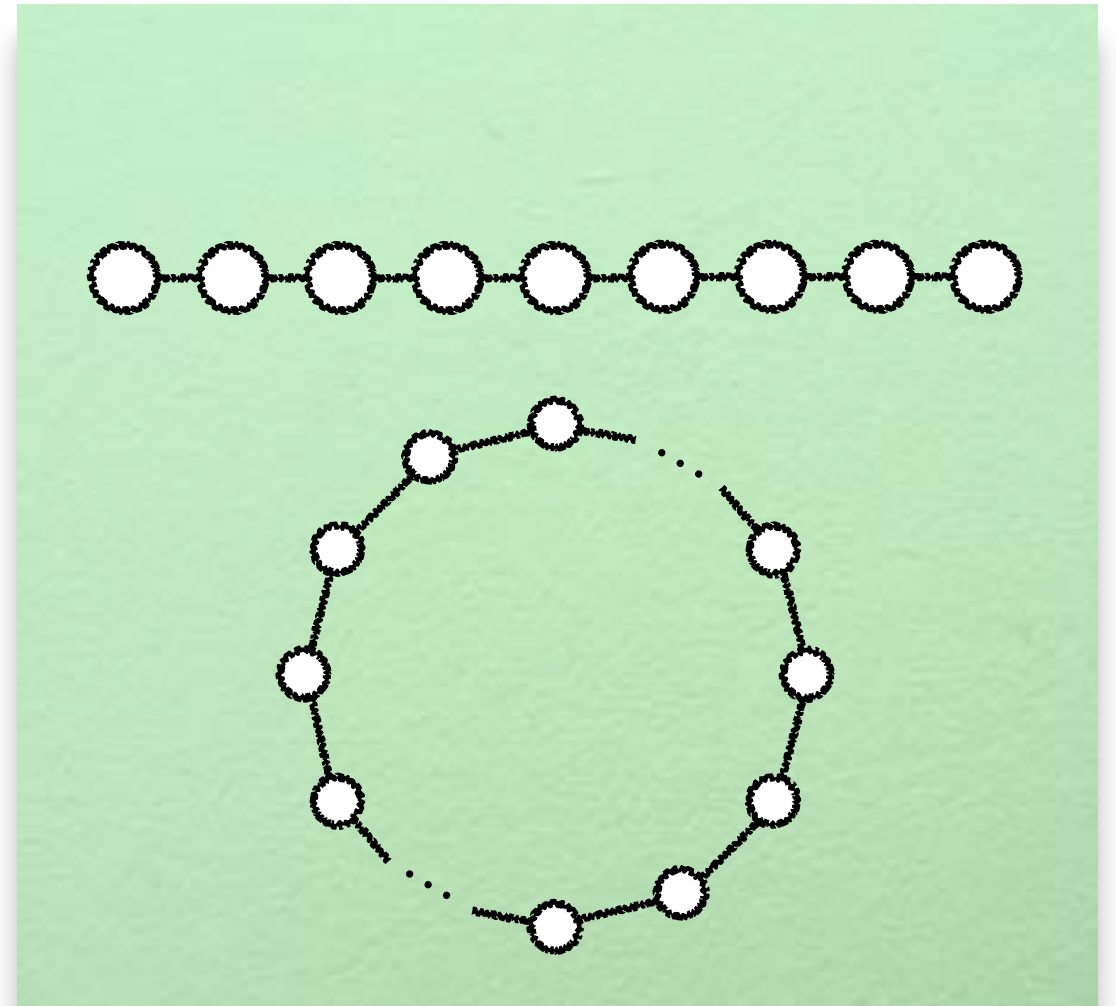
Reachability is not FO definable.

For every k , we build S_k :

Gaifman locality

Reachability is not FO definable.

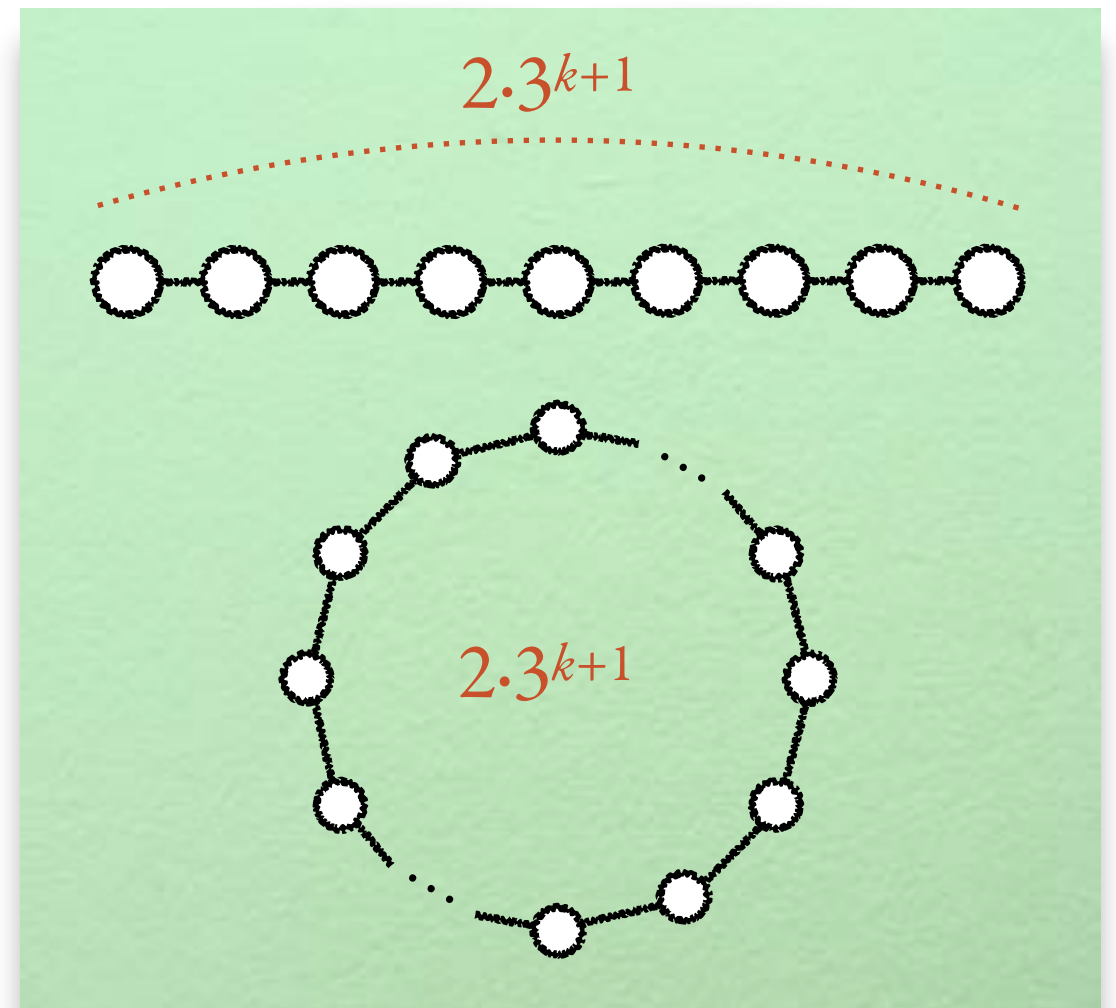
For every k , we build S_k :



Gaifman locality

Reachability is not FO definable.

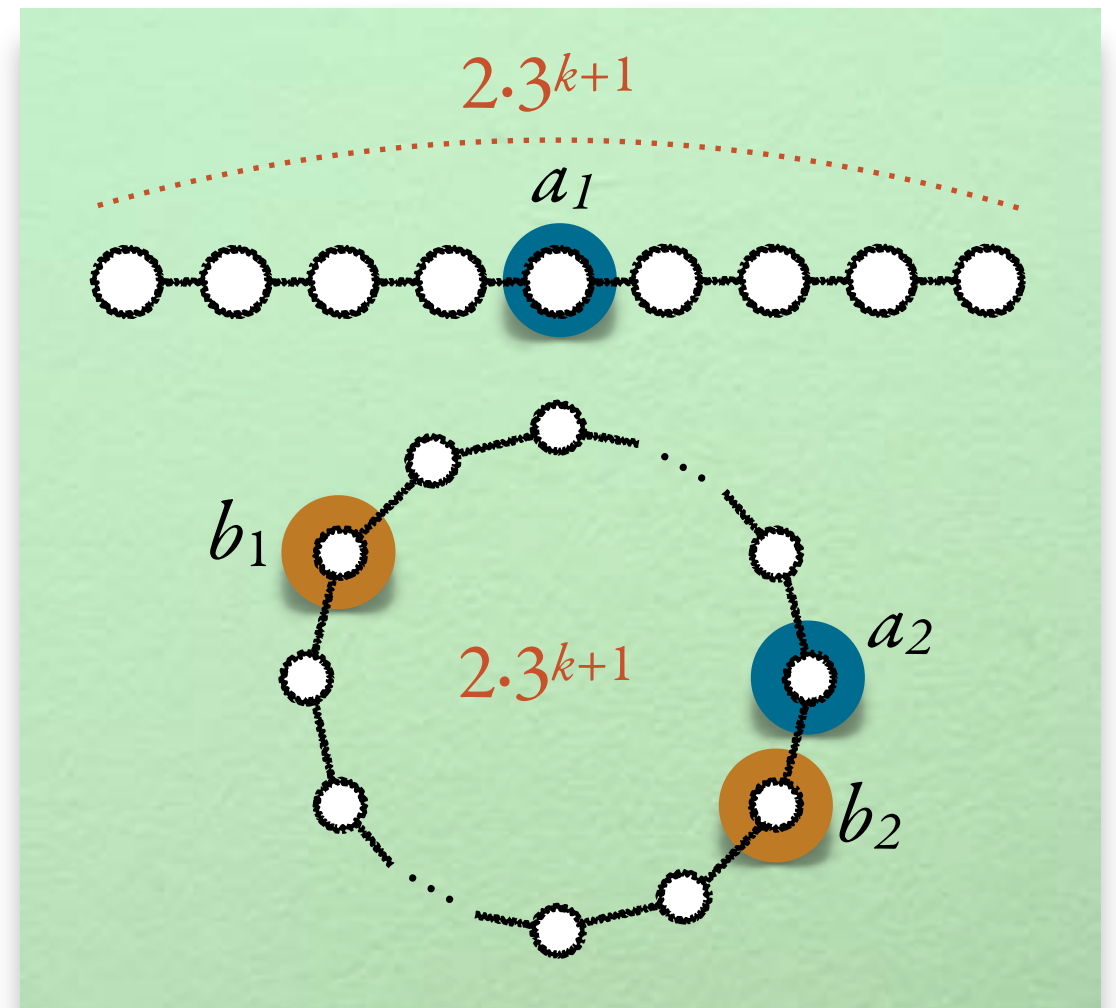
For every k , we build S_k :



Gaifman locality

Reachability is not FO definable.

For every k , we build S_k :

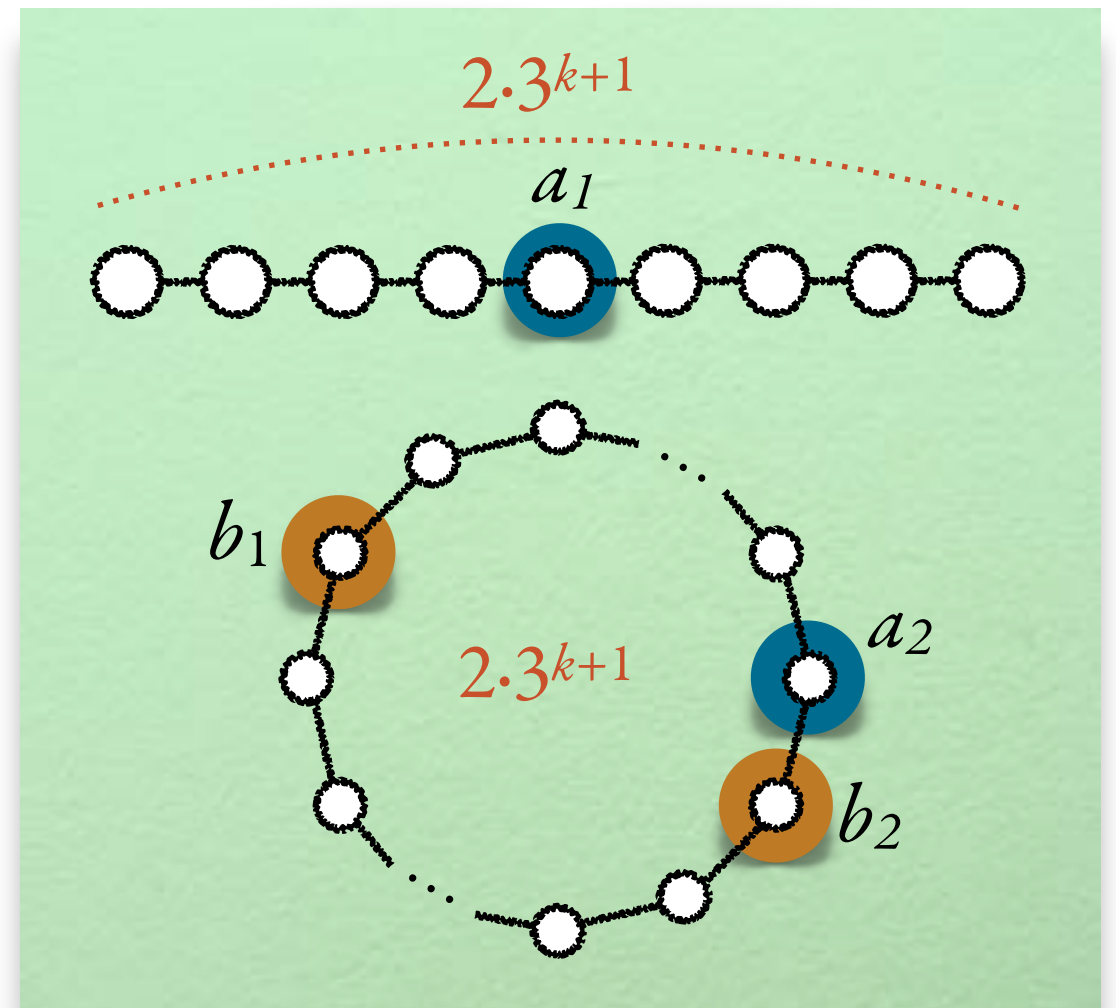


Gaifman locality

Reachability is not FO definable.

For every k , we build S_k :

And $S_k[(a_1, a_2), 3^{k+1}] \cong S_k[(b_1, b_2), 3^{k+1}]$



Gaifman locality

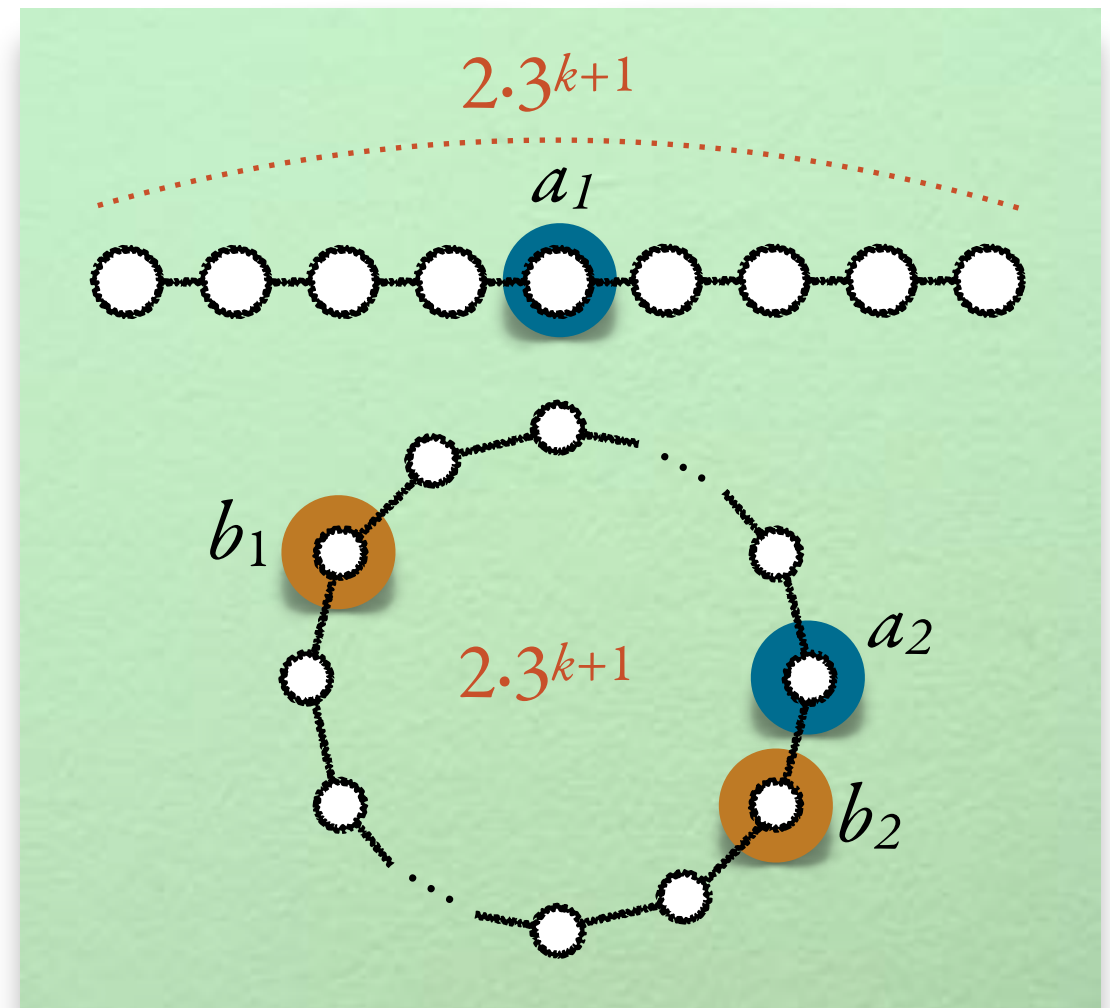
Reachability is not FO definable.

For every k , we build S_k :

And $S_k[(a_1, a_2), 3^{k+1}] \cong S_k[(b_1, b_2), 3^{k+1}]$

However,

- b_2 is reachable from b_1 ,
- a_2 is **not** reachable from a_1 .



Gaifman locality

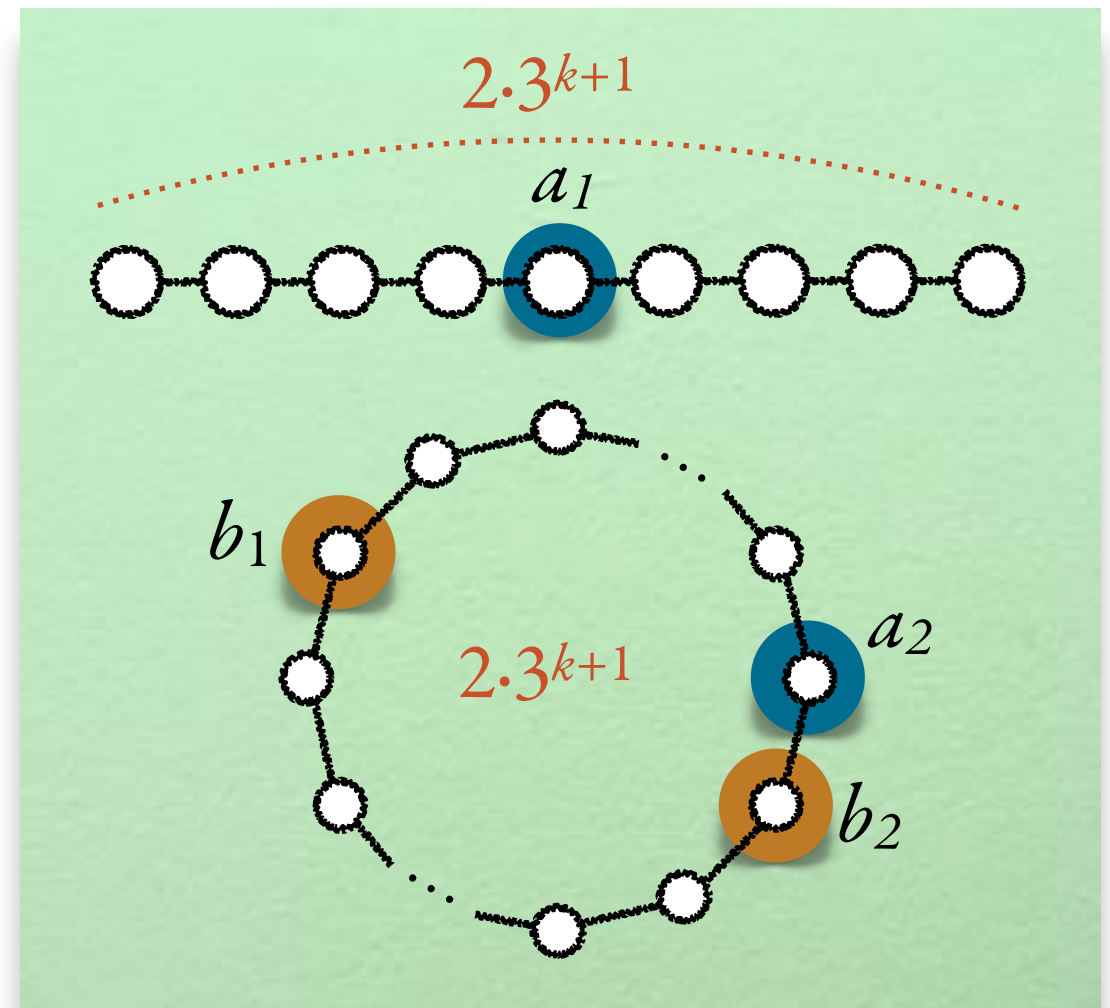
Reachability is not FO definable.

For every k , we build S_k :

And $S_k[(a_1, a_2), 3^{k+1}] \cong S_k[(b_1, b_2), 3^{k+1}]$

However,

- b_2 is reachable from b_1 ,
- a_2 is **not** reachable from a_1 .

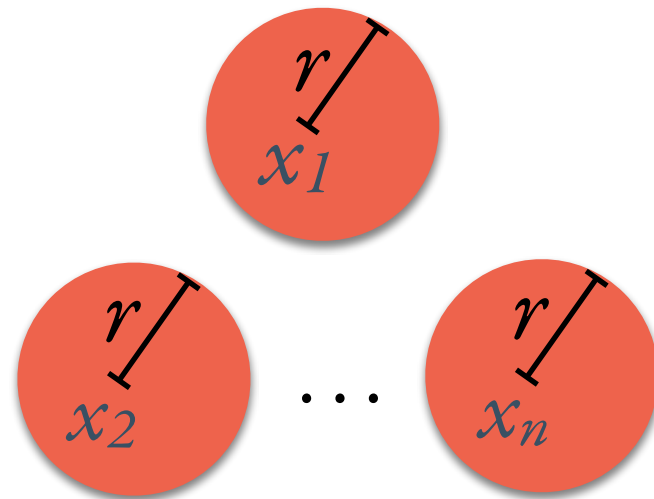


Your turn! $Q(x) = "x \text{ is a vertex separator}"$

Gaifman Theorem

Basic local sentence:

$\exists x_1, \dots, x_n$

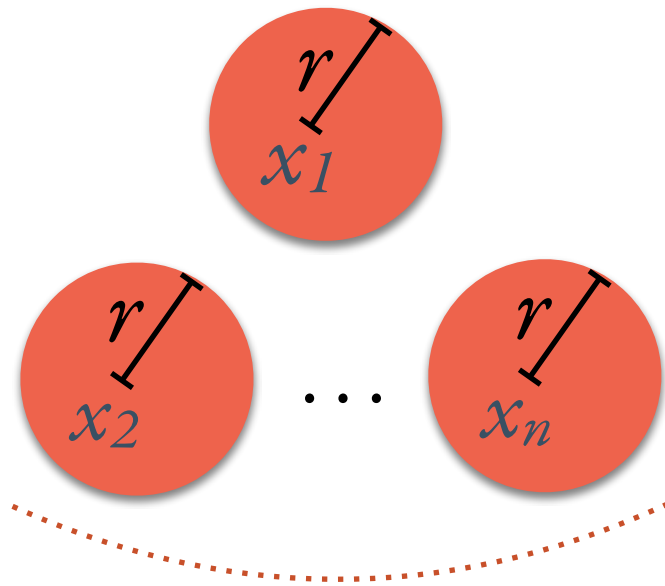


$\wedge \psi_1(x_1) \wedge \dots \wedge \psi_n(x_n)$

Gaifman Theorem

Basic local sentence:

$$\exists x_1, \dots, x_n$$



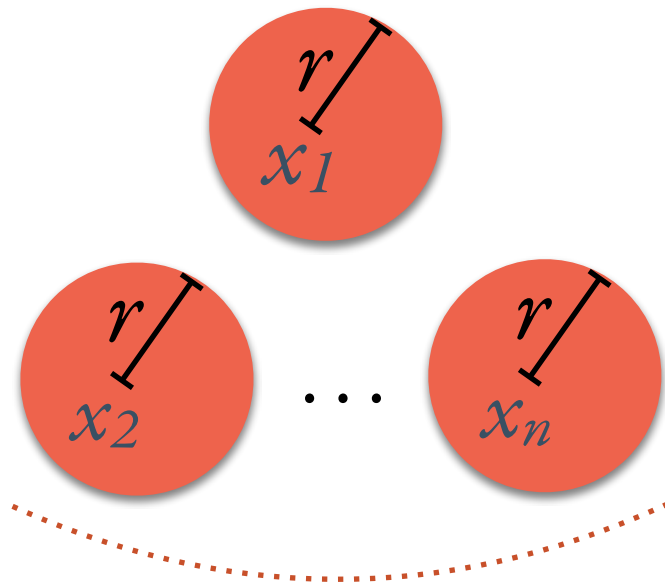
disjoint r -balls around x_1, \dots, x_n

$$\wedge \psi_1(x_1) \wedge \dots \wedge \psi_n(x_n)$$

Gaifman Theorem

Basic local sentence:

$\exists x_1, \dots, x_n$



disjoint r -balls around x_1, \dots, x_n

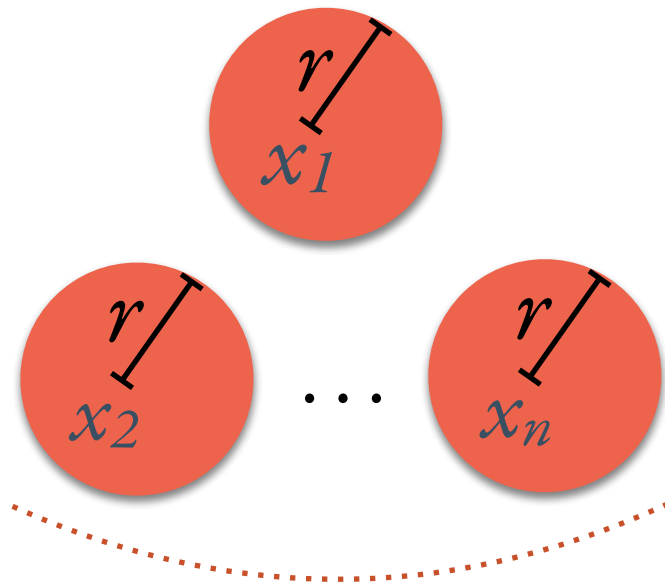
$\wedge \psi_1(x_1) \wedge \dots \wedge \psi_n(x_n)$

r -local formulas

Gaifman Theorem

Basic local sentence:

$\exists x_1, \dots, x_n$



disjoint r -balls around x_1, \dots, x_n

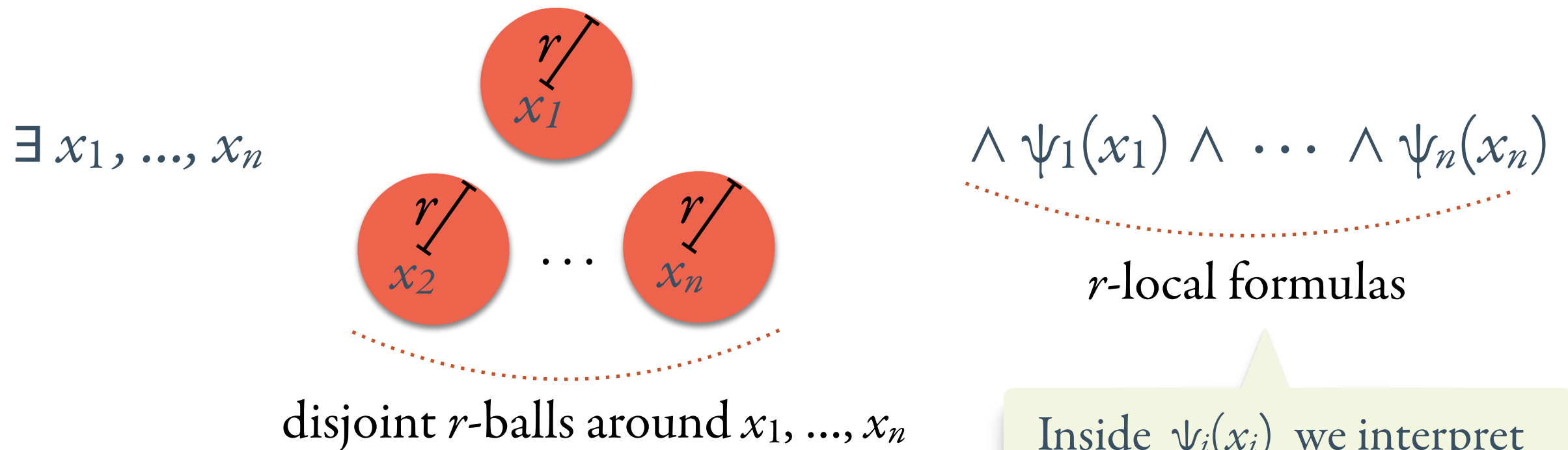
$\wedge \psi_1(x_1) \wedge \dots \wedge \psi_n(x_n)$

r -local formulas

Inside $\psi_i(x_i)$ we interpret
 $\exists y . \phi$ as $\exists y . d(x_i, y) \leq r \wedge \phi$

Gaifman Theorem

Basic local sentence:



Inside $\psi_i(x_i)$ we interpret
 $\exists y . \phi$ as $\exists y . d(x_i, y) \leq r \wedge \phi$

Gaifman Theorem: Every FO sentence is equivalent to
a boolean combination of **basic local sentences**.

Recap

Recap

EF games

FO sentences with quantifier rank n
=
winning strategies for Spoiler in the n -round EF game

Recap

EF games

FO sentences with quantifier rank n
=
winning strategies for Spoiler in the n -round EF game

0-1 Law

FO sentences are almost always true or almost always false

Recap

EF games

FO sentences with quantifier rank n
=
winning strategies for Spoiler in the n -round EF game

0-1 Law

FO sentences are almost always true or almost always false

Hanf locality

FO sentences with quantifier rank n
=
counting 3^n sized balls up to n

Recap

EF games

FO sentences with quantifier rank n
=
winning strategies for Spoiler in the n -round EF game

0-1 Law

FO sentences are almost always true or almost always false

Hanf locality

FO sentences with quantifier rank n
=
counting 3^n sized balls up to n

Gaifman locality

Queries of quantifier rank n output tuples closed under 3^{n+1} balls.

Recap

EF games

FO sentences with quantifier rank n
=
winning strategies for Spoiler in the n -round EF game

0-1 Law

FO sentences are almost always true or almost always false

Hanf locality

FO sentences with quantifier rank n
=
counting 3^n sized balls up to n

Gaifman locality

Queries of quantifier rank n output tuples closed under 3^{n+1} balls.

Gaifman Theorem

An FO sentence can only say
“there are some points at distance $\geq 2r$
whose r -balls are isomorphic to certain structures”
or a boolean combination of that.

Some more cool stuff...

Descriptive complexity

What properties can be checked efficiently? E.g. 3COL can be tested in NP

Metatheorem

“A property can be expressed in [insert some logic here]
iff
it can be checked in [some complexity class here]”

Some more cool stuff...

Descriptive complexity

What properties can be checked efficiently? E.g. 3COL can be tested in NP

Metatheorem

“A property can be expressed in [insert some logic here]
iff
it can be checked in [some complexity class here]”

↪ “A property is FO-definable iff it can be tested in AC^0 ”

↪ “A property is $\exists SO$ -definable iff it can be tested in NP” [Fagin 73]

↪ Open problem: which logic captures PTIME?

Some more cool stuff...

Recursion

Can we enhance query languages with recursion ? E.g. express reachability properties

Some more cool stuff...

Recursion

Can we enhance query languages with recursion ? E.g. express reachability properties

Datalog (semantics based on least fixpoint)

```
Ancestor(X,Y) :- Parent(X,Z), Ancestor(Z,Y)
Ancestor(X,X) :- .
?- Ancestor("Louis XIV",Y)
```

Some more cool stuff...

Recursion

Can we enhance query languages with recursion ? E.g. express reachability properties

Datalog (semantics based on least fixpoint)

```
Ancestor(X,Y) :- Parent(X,Z), Ancestor(Z,Y)
Ancestor(X,X) :- .
?- Ancestor("Louis XIV",Y)
```

↗ Incomparable with FO (has recursion, but is monotone)

↗ Evaluation is in PTIME (for data complexity, but also for bounded arity)

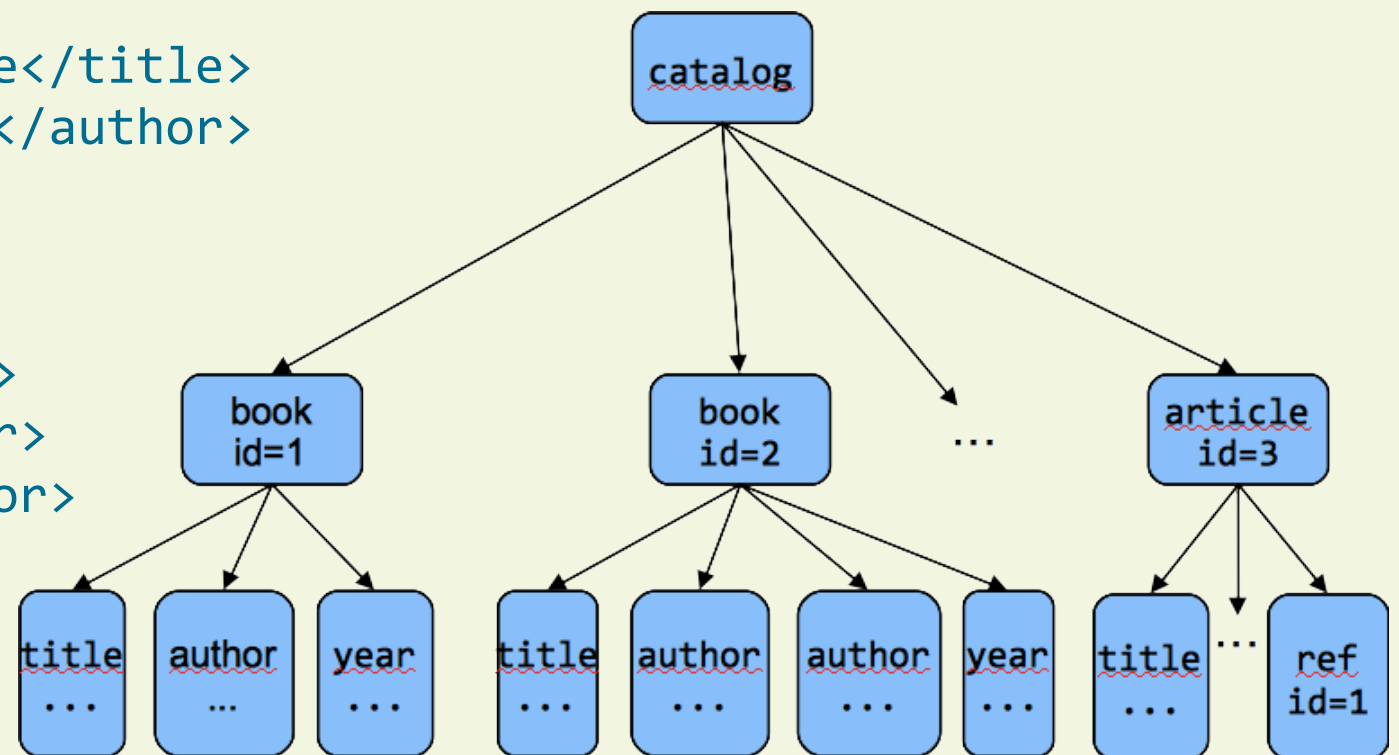
Some more cool stuff...

Semi-structured data

Tree-structured or graph-structures dbs in place of relational dbs.

XML, XPath, Stream processing, ...

```
<catalog>
  <book id="1">
    <title>XML Developer's Guide</title>
    <author>Matthew Gambardella</author>
    <year>2000</year>
  </book>
  <book id="2">
    <title>Beginning XML</title>
    <author>David Hunter</author>
    <author>David Gibbons</author>
    <year>2007</year>
  </book>
  ...
</catalog>
```



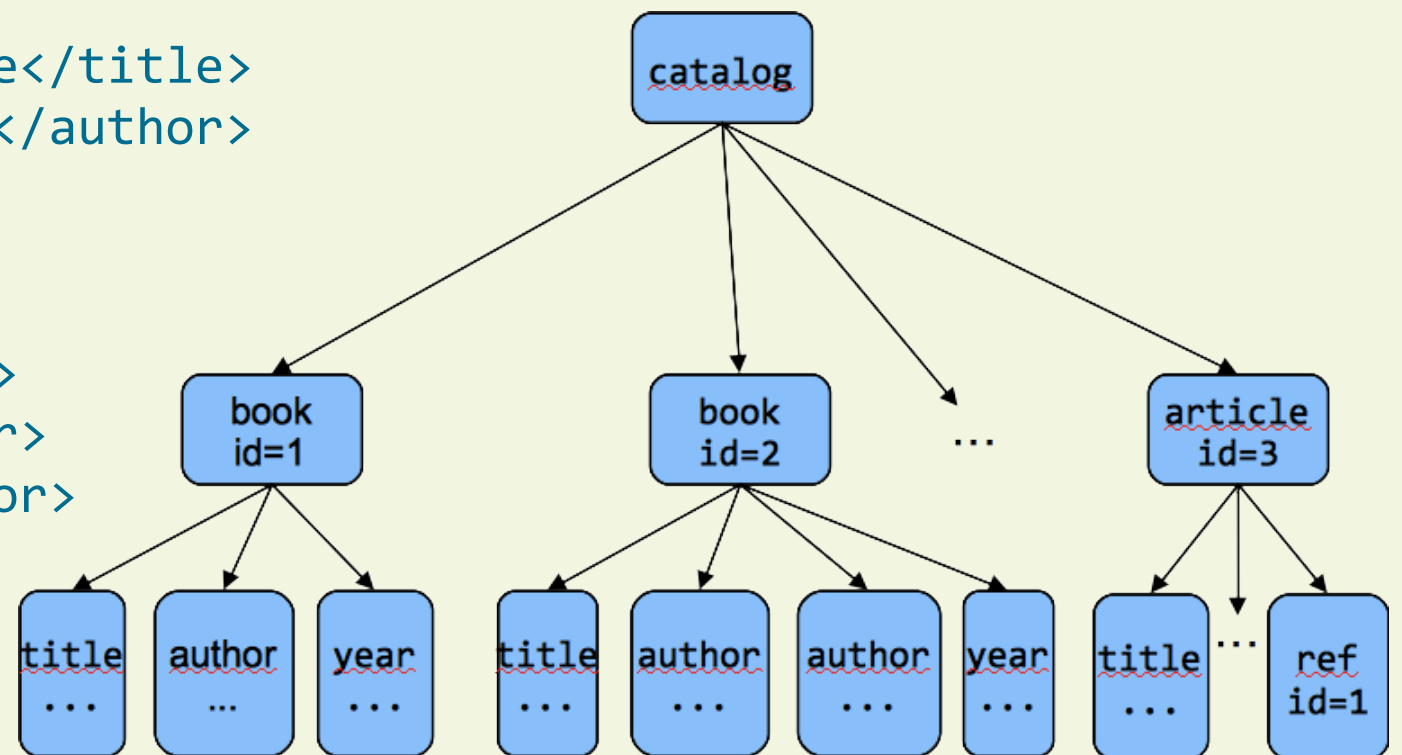
Some more cool stuff...

Semi-structured data

Tree-structured or graph-structures dbs in place of relational dbs.

XML, XPath, Stream processing, ...

```
<catalog>
  <book id="1">
    <title>XML Developer's Guide</title>
    <author>Matthew Gambardella</author>
    <year>2000</year>
  </book>
  <book id="2">
    <title>Beginning XML</title>
    <author>David Hunter</author>
    <author>David Gibbons</author>
    <year>2007</year>
  </book>
  ...
</catalog>
```



⇒ Evaluation of XPath is in linear time (data complexity)

[Bojanczyk, Parys 08]

⇒ Satisfiability for $\text{FO}^2[\downarrow, \sim]$ is decidable

[Bojanczyk, Muscholl, Schwentick, Segoufin 09]

Some more cool stuff...

Incomplete information

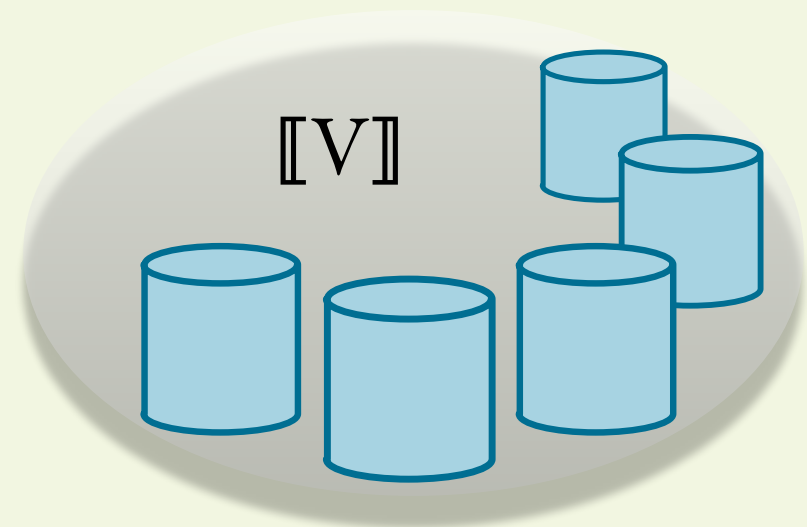
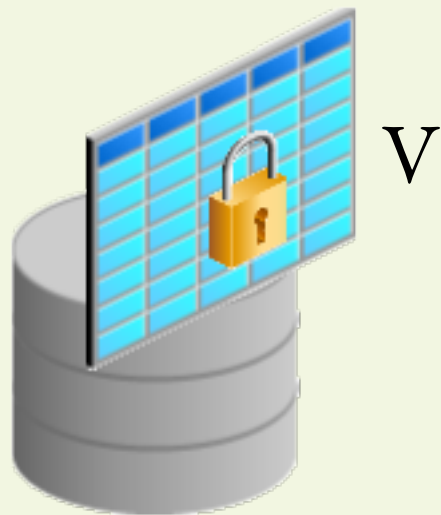
How to correctly reason when information is hidden/missing/noisy/... ?

Some more cool stuff...

Incomplete information

How to correctly reason when information is hidden/missing/noisy/... ?

Certain Query Answers (CQA)



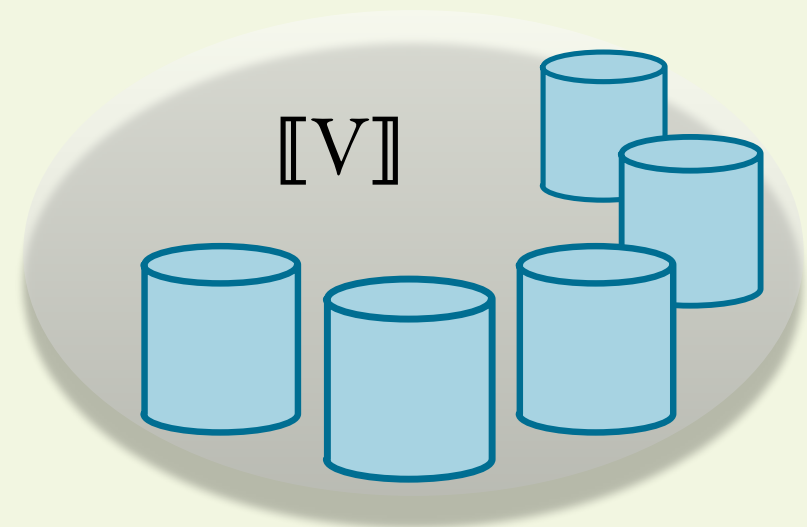
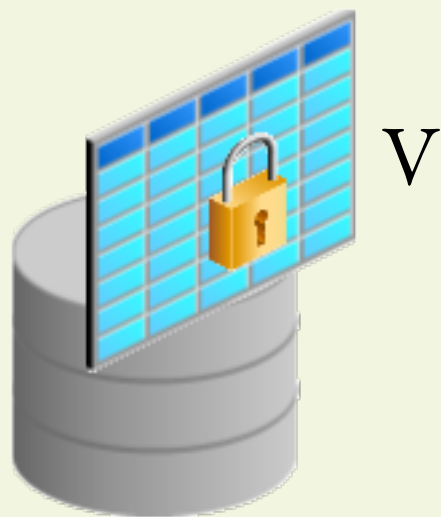
$$\phi[V] = \bigcap_{D \in [V]} \phi(D)$$

Some more cool stuff...

Incomplete information

How to correctly reason when information is hidden/missing/noisy/... ?

Certain Query Answers (CQA)



$$\phi \llbracket V \rrbracket = \bigcap_{D \in \llbracket V \rrbracket} \phi(D)$$

\rightsquigarrow CQA computable in PTIME w.r.t. view size. [Abiteboul, Kanellakis, Grahne 91]

Bibliography

- Abiteboul, Hull, Vianu, “Foundations of Databases”, Addison-Wesley, 1995.
(available at <http://webdam.inria.fr/Alice/>)
- Libkin, “Elements of Finite Model Theory”, Springer, 2004.
- Immerman, “Descriptive Complexity”, Springer, 1999.
- Otto, “Finite Model Theory”, Springer, 2005
(available at www.mathematik.tu-darmstadt.de/~otto/LEHRE/FMT0809.ps)
- Väänänen, “A Short course on Finite Model Theory”, 1994.
(available at www.math.helsinki.fi/logic/people/jouko.vaanananen/shortcourse.pdf)