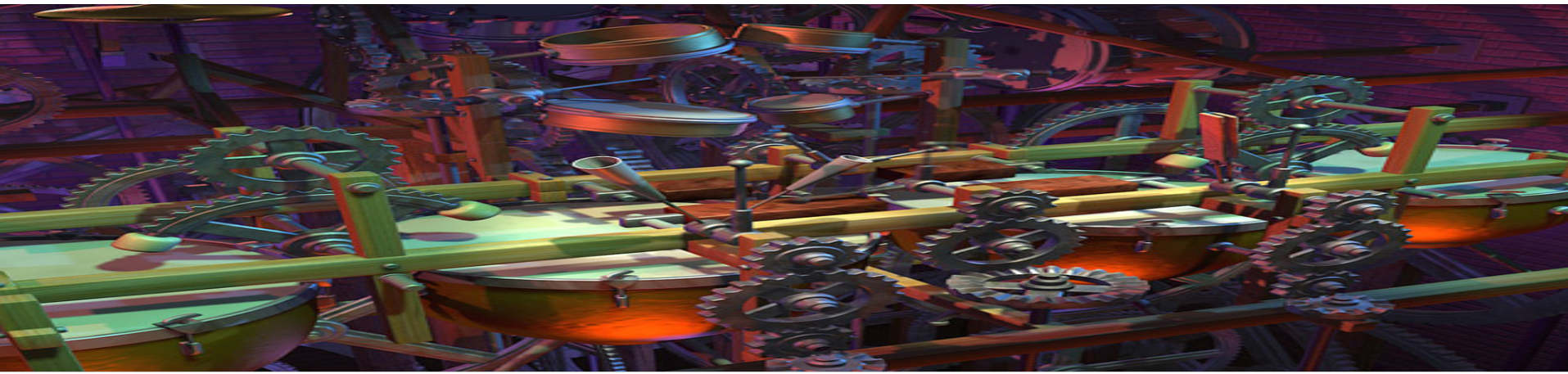
Automata As Abstractions

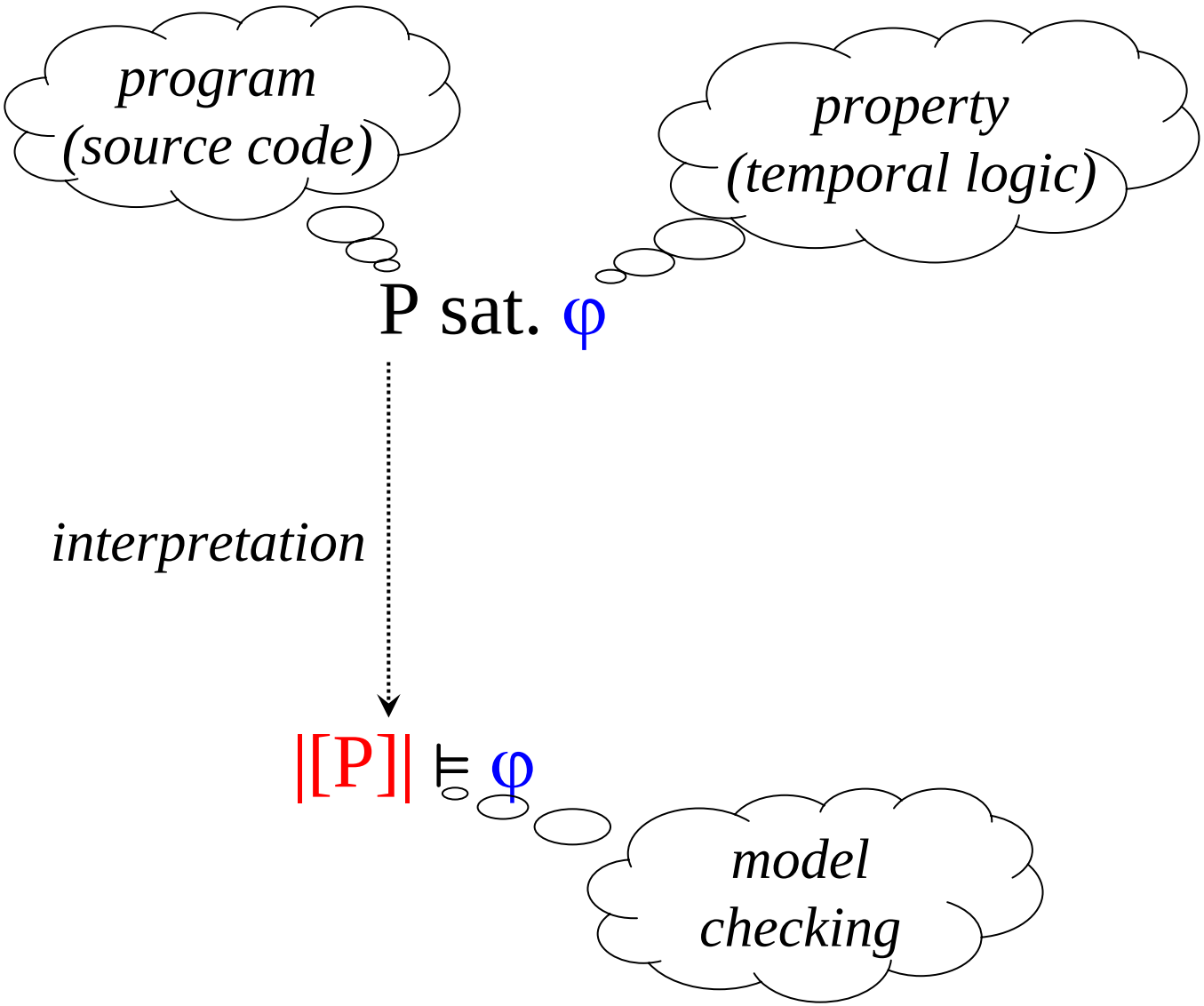


Dennis Dams

joint work with Kedar Namjoshi

Bell Laboratories (Murray Hill, NJ)

The Mission


program
(source code)
property
(temporal logic)
P sat. φ
interpretation
|[P]| ⊨ φ
model
checking

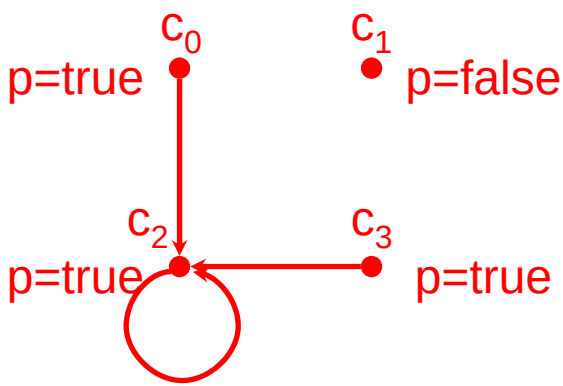
Kripke structures and Branching-time TL

P sat. φ

interpretation

$|[P]| \models \varphi$

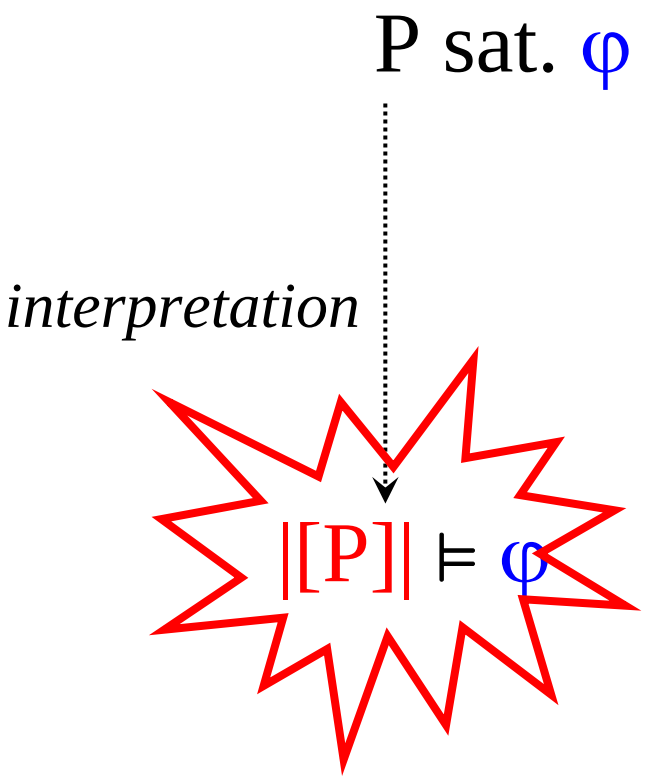
Example:



$c_i \models$ AXAG p $(\forall i)$

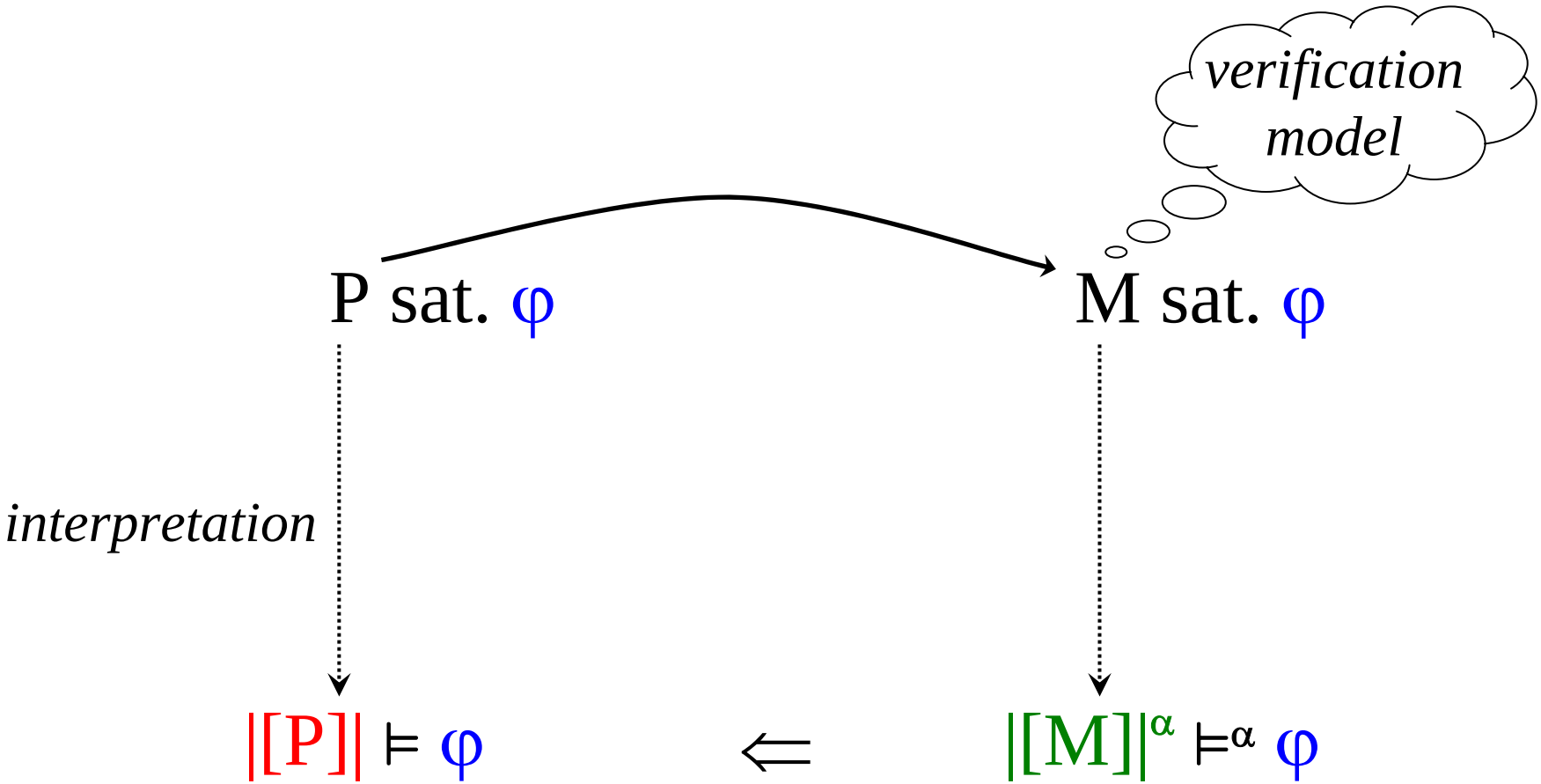
$c_1 \not\models$ EG true

The Problem

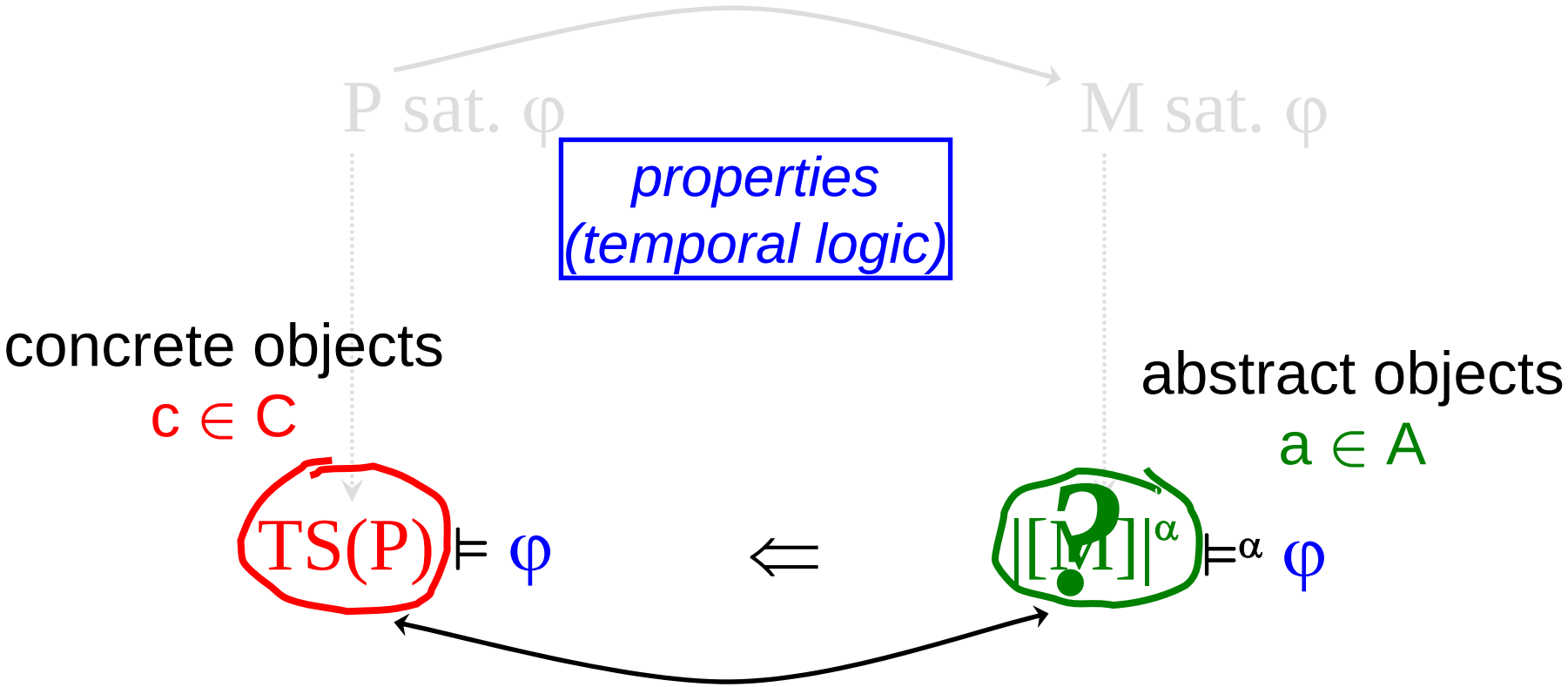




Abstraction to the Rescue



Abstraction Framework: Ingredients



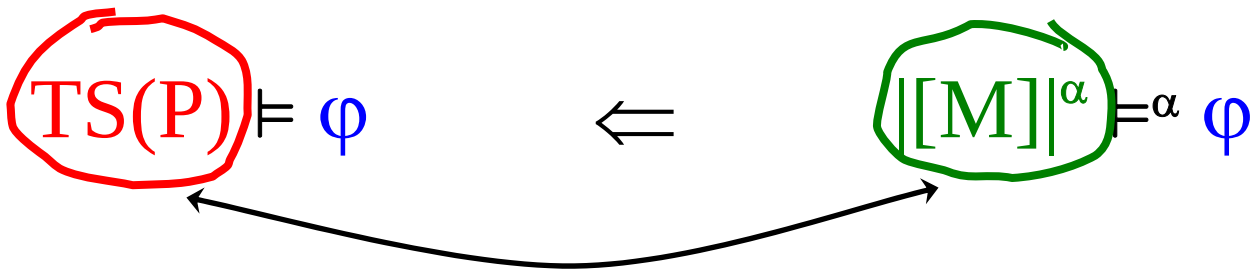
Abstraction Framework: Requirements

Soundness (preservation, safety):

$$\forall c\ \forall \varphi\ [\exists a\ \ \rho(c,a) \wedge a \models^{\alpha} \varphi] \quad \Rightarrow \quad c \models \varphi$$

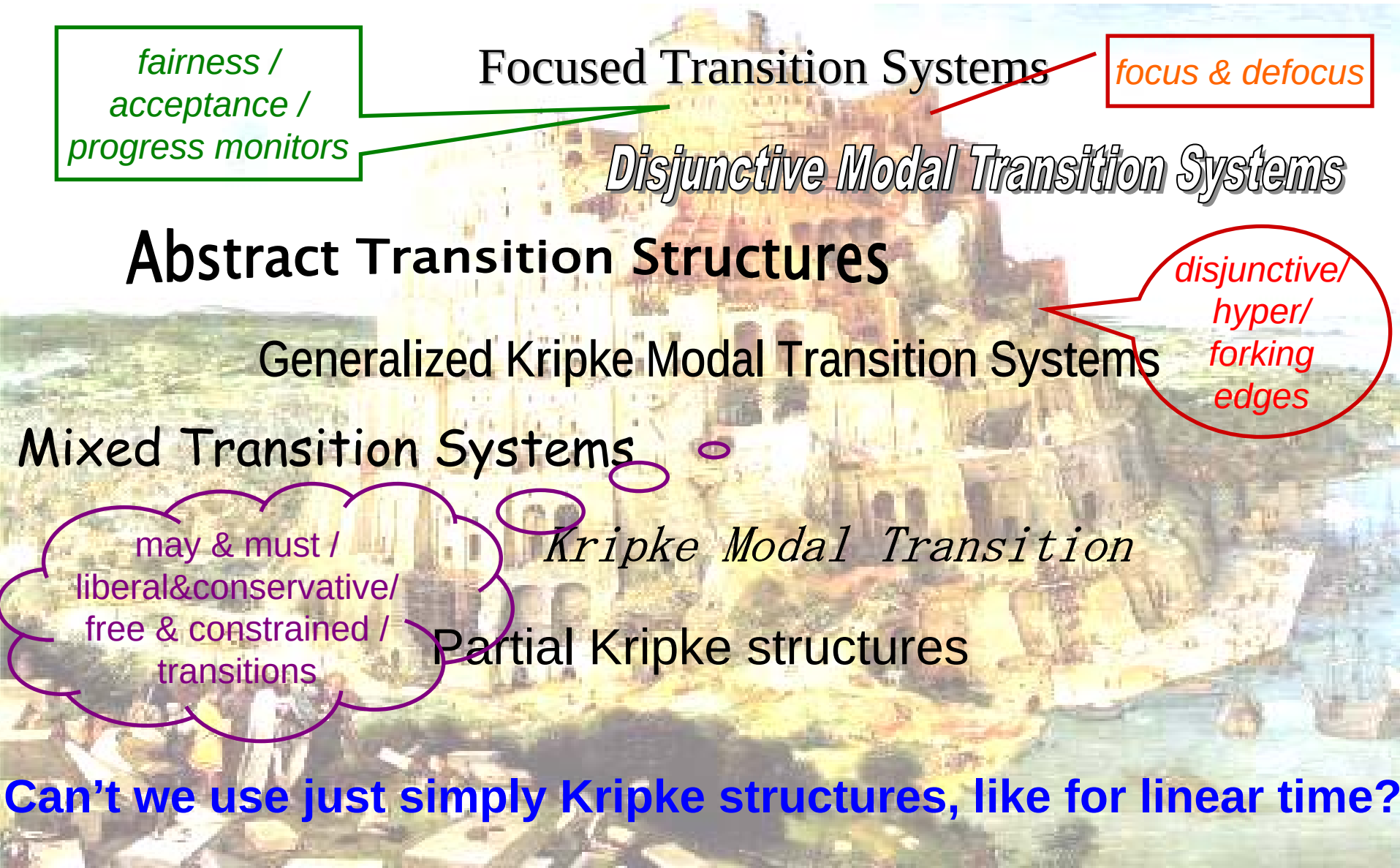
Completeness ("per property"):

$$\forall c\ \forall \varphi\ [\exists\ \textit{finite}\ a\ \ \rho(c,a) \wedge a \models^{\alpha} \varphi] \quad \Leftarrow \quad c \models \varphi$$



abstraction relation: $\rho \subseteq C \times A$

How to Abstract Kripke structures for Branching-time Logic...?



Can't we use just simply Kripke structures, like for linear time?

Use Automata As Abstractions!

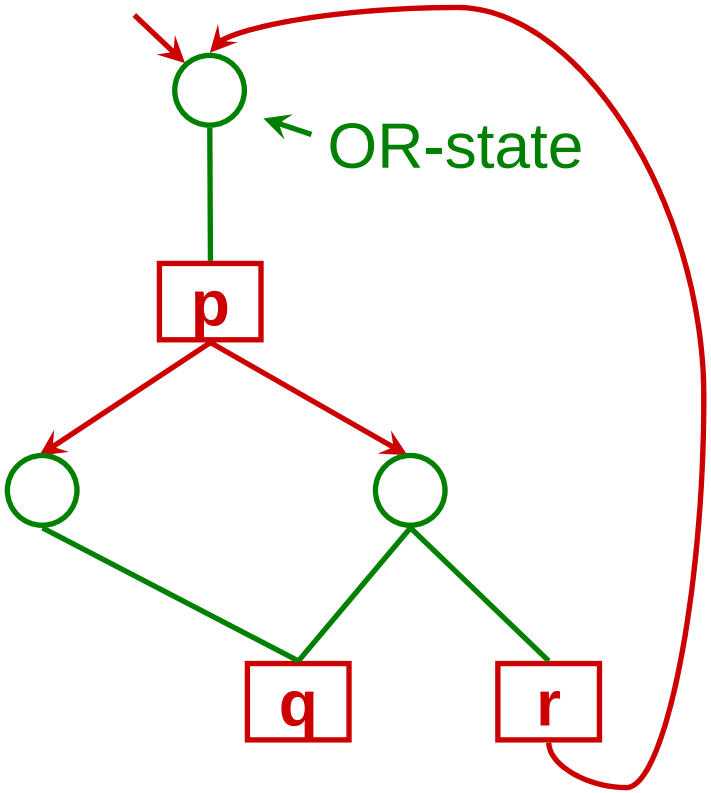
✓well-studied!
✓many results for free!
✓sound!
✓complete!
✓no concept clutter!



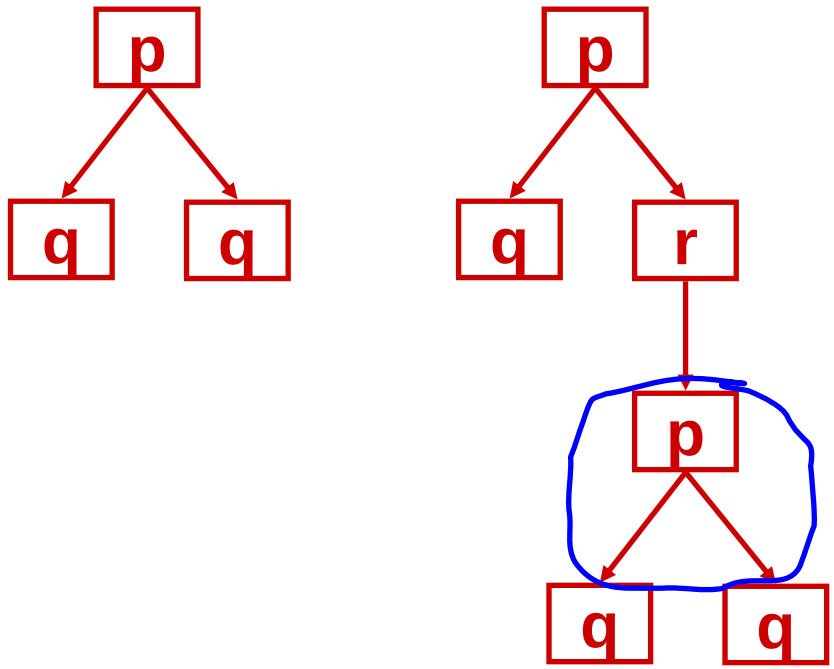
(Tree) Automata

Tree Automata are just Kripke structures with additional OR-states

tree automaton:

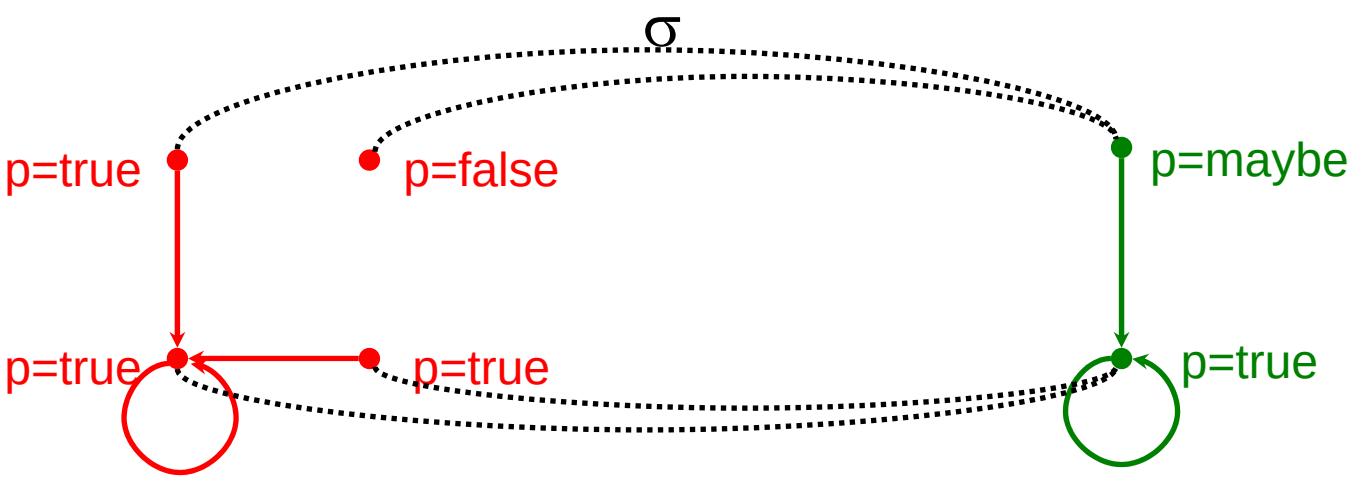


some generated Kripke structures (tree unfoldings):



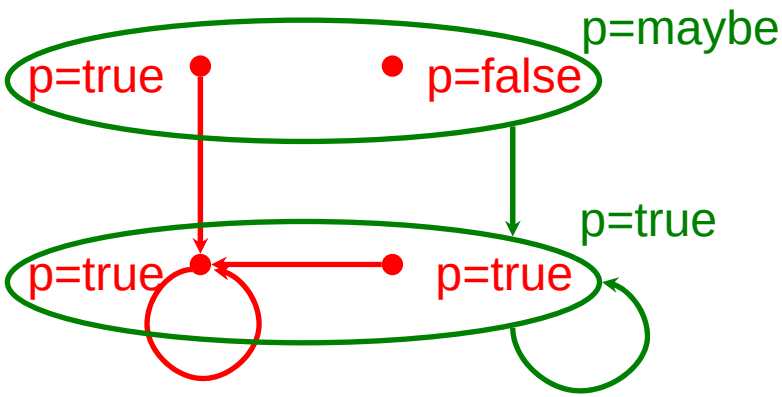
(Also need acceptance conditions, but ignored in this talk)

Framework 1: Simulation



View abstract states as sets

$\sigma(c,a)$: "$c \in a$"



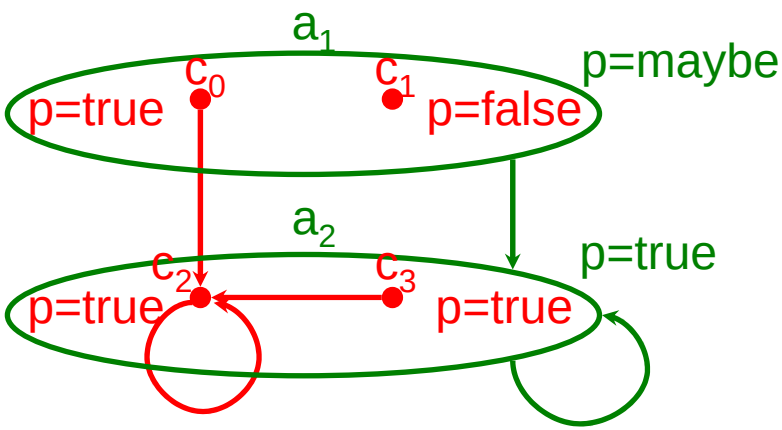
Partial Kripke structures

σ is a simulation:

- $p(a) = true \Rightarrow \forall c \in a \; p(c) = true$ (same for false)
- $\exists c \in a \; \exists d \in b \; c \rightarrow d \Rightarrow a \rightarrow b$

Partial Kripke structures preserve universal properties

$\sigma(c,a)$: "$c \in a$"



$c_i \models$ AXAG p $(\forall i)$

$c_1 \not\models$ EG true ☹

$a_i \models$ AXAG p $(\forall i)$

$a_1 \models$ EG true

σ is a simulation:

- p(a) = true $\Rightarrow$ $\forall\, c \in a$ p(c) = true (same for false)
- $\exists\, c \in a\ \exists\, d \in b$ $c \rightarrow d$ $\Rightarrow$ $a \rightarrow b$

How about existential properties?

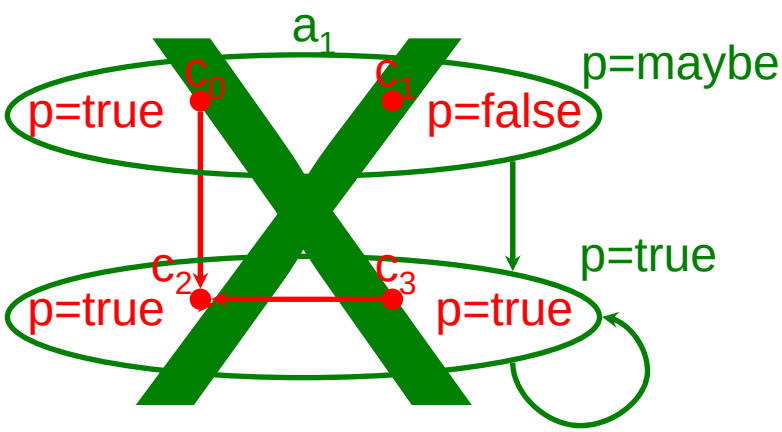
$$\exists\, c \in a \;\; \exists\, d \in b \;\; c \to d \quad \Rightarrow \quad a \to b$$

$$\forall\, c \in a \;\; \exists\, d \in b \;\; c \to d \quad \Leftarrow \quad a \to b$$

+

---

bisimulation... ☹



Framework 2: Modal Simulation

$$\exists\, c \in a \;\; \exists\, d \in b \;\; c \rightarrow d \quad \Rightarrow \quad a \dashrightarrow b \quad \textit{may}$$

$$\forall\, c \in a \;\; \exists\, d \in b \;\; c \rightarrow d \quad \Leftarrow \quad a \rightarrow b \quad \textit{must}$$

$$+$$

---

modal simulation

(note that *must* $\subseteq$ *may*)

Kripke Modal Transition Systems (KMTS)

$\models^{\alpha}$: interpret A quantifiers over *may* paths,
and E quantifiers over *must* paths

The Quest for Completeness

1. Simulation – Partial Kripke structs – *univ. props.*
2. Modal simulation – KMTS – *univ. + exist. props.*
3. Mixed simulation – XMTS
4. Disjunctive modal refinement – DMTS
5. Generalized mixed simulation – GKMTS
6. Alternating refinement – Abst. trans. structs.

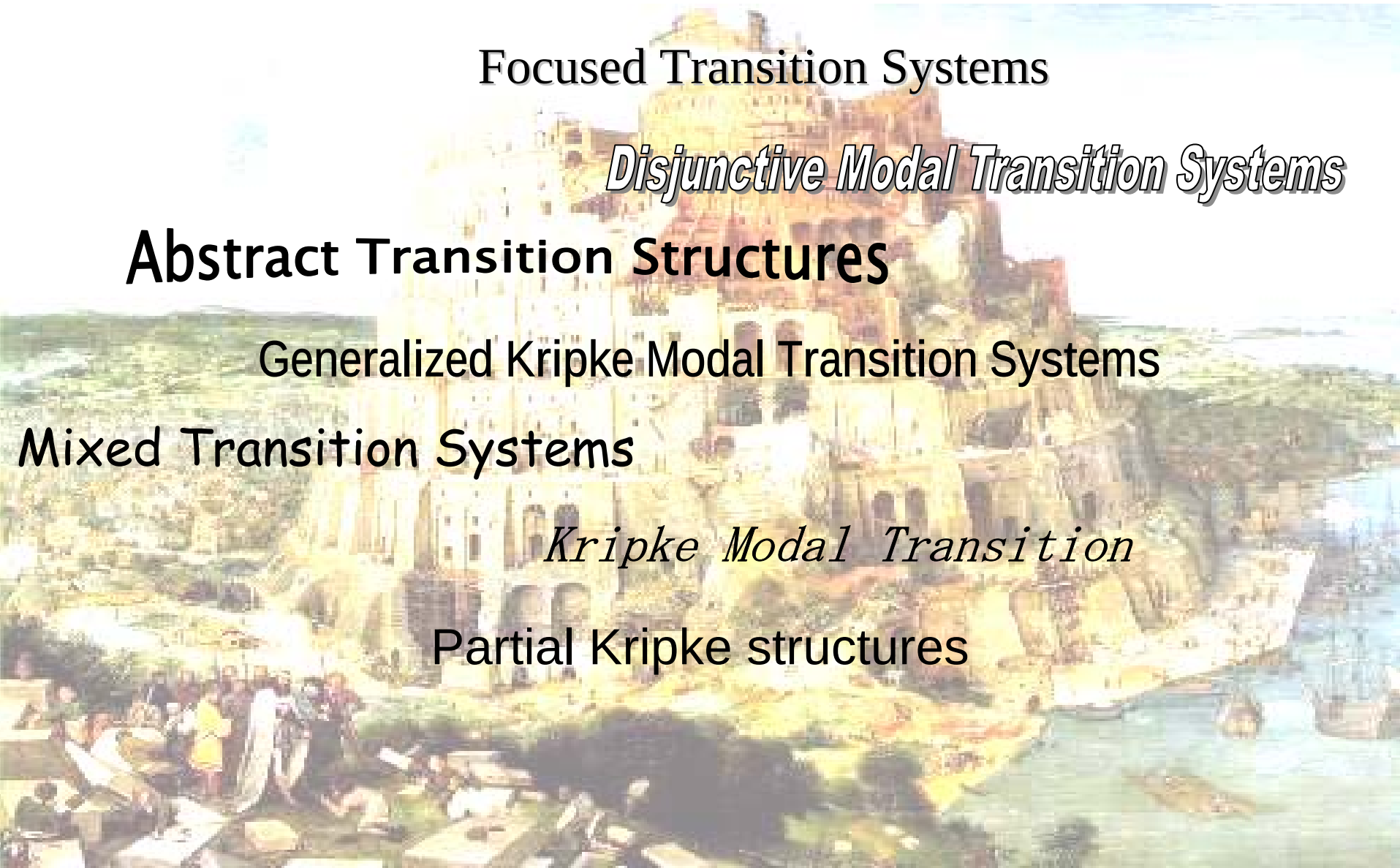
} *more exist. props.*

7. Focused simulation – FTS – *complete for all BTL*

   FTS: $(S, I, T^-, T^+, L^-, L^+, F^-, F^+, \Phi)$

   (closely related to alternating tree automata)

Is there a common underlying language?

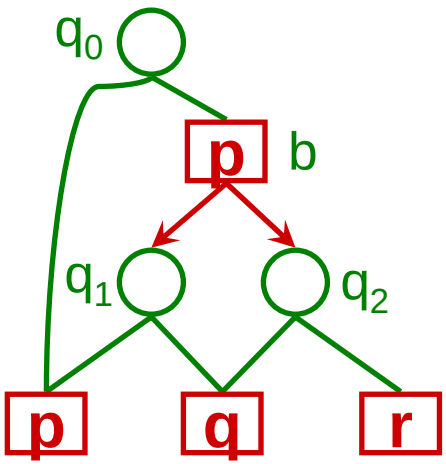


Tree Automata – Runs/Acceptance

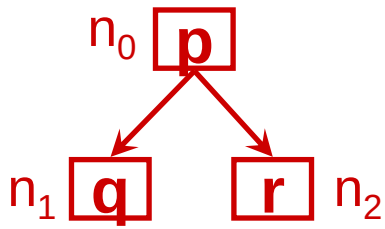
A run of A on T is a tree where

1. the root is (n_{init}, q_{init})
2. for nodes (n,q): $\exists\ b \in OR(q)$: (n,b) is a child
3. for nodes (n,b):
   - n and b have the same label
   - $\forall\ n' \in R(n)$: $\exists\ q' \in BR(b)$: (n',q') is a child
   - $\forall\ q' \in BR(b)$: $\exists\ n' \in R(n)$: (n',q') is a child

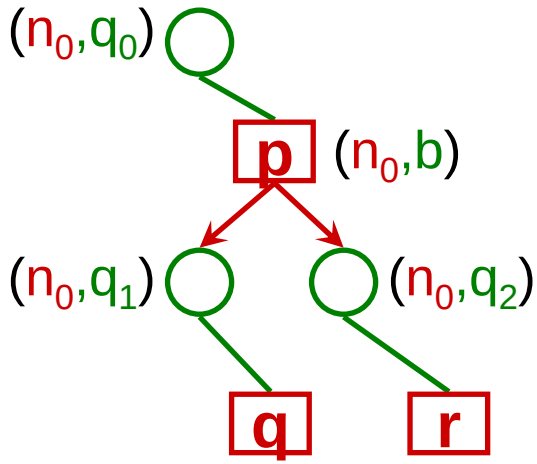
automaton A:



tree T:



run of A on T:



Automata as Abstractions (Framework 0)

Take $\rho(\mathcal{S},\mathcal{A}) \triangleq \mathcal{S} \in \mathcal{L}(\mathcal{A})$ ($\mathcal{S}$: Kripke str, $\mathcal{A}$: tree aut)

(i.e. as the *concretization* $\gamma(\mathcal{A})$, take the language of $\mathcal{A}$)

and $\mathcal{A} \models^{\alpha} \varphi \triangleq \mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_{\varphi})$

($\mathcal{A}_{\varphi}$: the automaton representation of φ)

Soundness:

$$\forall\mathcal{S}\,\forall\varphi\,[\exists\mathcal{A}\;\; \rho(\mathcal{S},\mathcal{A}) \wedge \mathcal{A} \models^{\alpha} \varphi] \quad \Rightarrow \quad \mathcal{S} \models \varphi$$

$$\mathcal{S} \in \mathcal{L}(\mathcal{A}) \wedge \mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_{\varphi}) \Rightarrow \mathcal{S} \in \mathcal{L}(\mathcal{A}_{\varphi})$$

Completeness:

$$\forall\mathcal{S}\,\forall\varphi\,[\exists \textit{ finite } \mathcal{A}\;\; \rho(\mathcal{S},\mathcal{A}) \wedge \mathcal{A} \models^{\alpha} \varphi] \quad \Leftarrow \quad \mathcal{S} \models \varphi$$

$$\mathcal{S} \in \mathcal{L}(\mathcal{B}) \wedge \mathcal{L}(\mathcal{B}) \subseteq \mathcal{L}(\mathcal{A}_{\varphi}) \Leftarrow \mathcal{S} \in \mathcal{L}(\mathcal{A}_{\varphi})$$

Automata Simulation (Framework 0'...)

- Language inclusion EXPTIME-hard
- Define simulation $\mathcal{A} \sqsubseteq \mathcal{B}$ (like in word case)
  - sufficient condition
  - in NP (polynomial if accept. conditions are Büchi)

- Redefine: $\mathcal{A} \models^{\alpha} \mathcal{B} \triangleq \mathcal{A} \sqsubseteq \mathcal{B}$
- Then soundness, completeness still trivial

must/may as a 3-valued transition relation

Since *must* ⊆ *may*, there are 3 possibilities:



an abstract transition with value *true*: R(a,b) = true



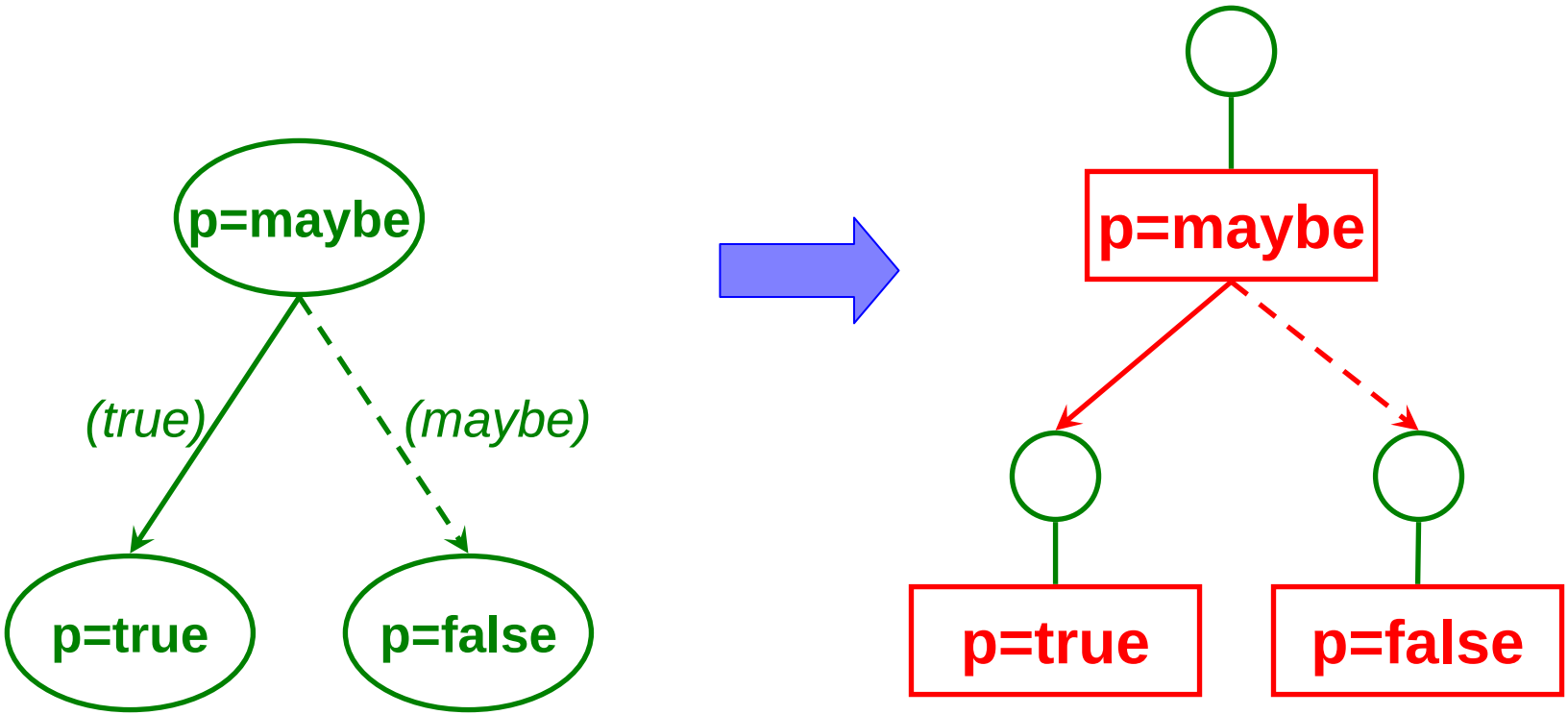
an abstract transition with value *maybe*: R(a,b) = maybe



an abstract transition with value *false*: R(a,b) = false

(cf. p(a) = true/maybe/false)

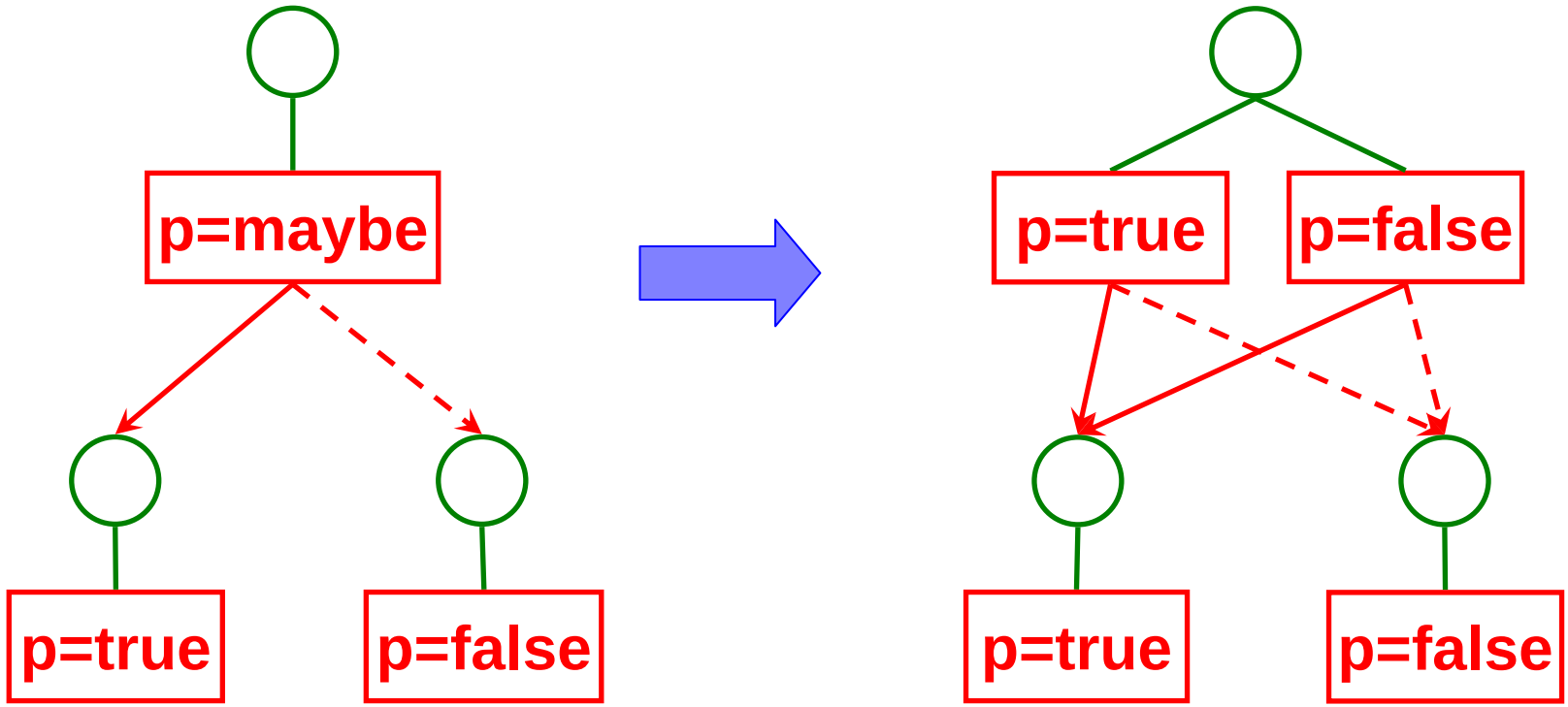
KMTS to Automata – 1: Insert OR states



KMTS

“Modal Tree Automaton”
(with 3-valued prop^s and trans^s)

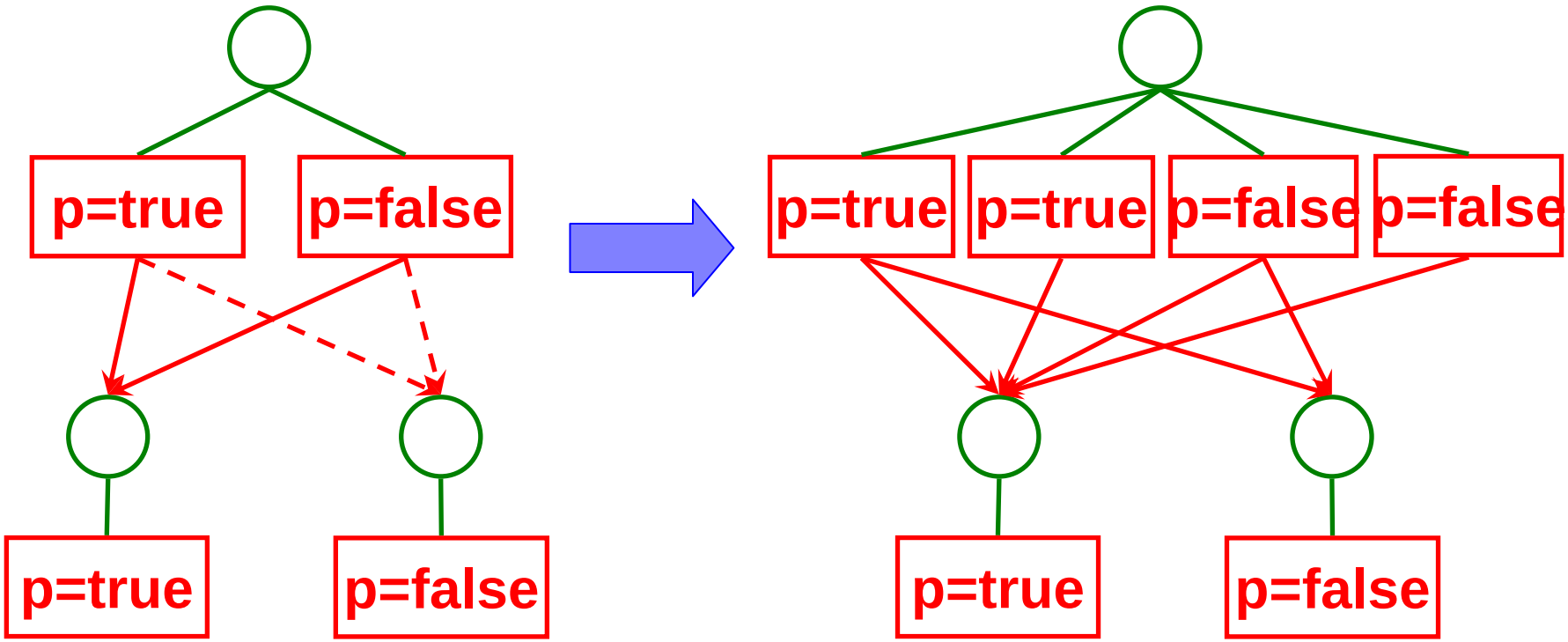
KMTS to Automata – 2: Remove *maybe* propositions



“Modal Tree Automaton”

Tree Automaton with 3-valued trans^{s}

KMTS to Automata – 3: Remove *maybe* transitions



Tree Automaton with 3-valued trans^{s}

Tree Automaton

Result

For every KMTS M, BTL formula ϕ,

and $\mathcal{A}$ such that M $\Rightarrow$ $\mathcal{A}$:

$$[M \models \varphi] \quad = \quad [\mathcal{A} \models^{\alpha} \varphi]$$

true	true
false	false
maybe	maybe

Conclusions

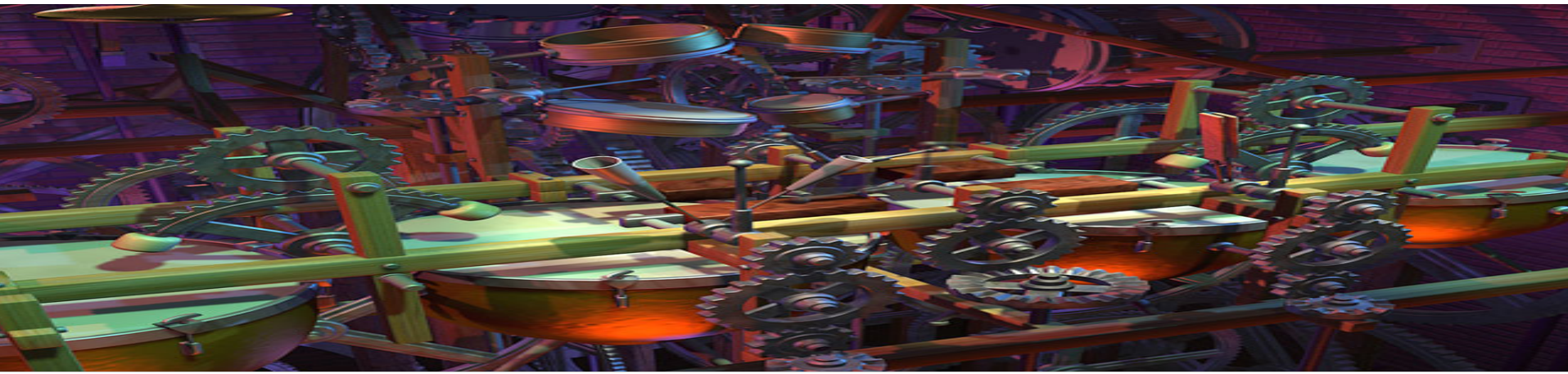
- Automata provide a unifying view of abstractions of transition systems (for branching *and* linear time [Uribe99, Kesten&Pnueli00])

  modal transition systems (etc.) *are* automata

  - Soundness and completeness of abs. frameworks: trivial
  - Using automata in Maximal Model Theorems
    - allows new solutions
    - simpler proofs of existing theorems

- Modal transition systems (etc.) provide a compact representation for certain automata
  - Modal tree automata: tree automata with 3-valued aspects
  - Novel notion: simulation between (ω-)tree automata

Thank You



Automata for system vs. for property

- *Automata-theoretic approach to BTL model checking:*
  - checking the concrete system:

    $M \models \varphi \quad \rightarrow \quad \text{tree}(M) \in \mathcal{L}(\mathcal{A}_\varphi)$

  - checking an abstraction:

    $\mathcal{A}_M \models \varphi \quad \rightarrow \quad \mathcal{L}(\mathcal{A}_M) \subseteq \mathcal{L}(\mathcal{A}_\varphi)$

- *Automata-theoretic approach for linear time:*
  - checking the concrete system:

    $M \models \varphi \quad \rightarrow \quad \mathcal{L}(M) \subseteq \mathcal{L}(\mathcal{A}_\varphi)$

  - checking an abstraction:

    $\mathcal{A}_M \models \varphi \quad \rightarrow \quad \mathcal{L}(\mathcal{A}_M) \subseteq \mathcal{L}(\mathcal{A}_\varphi)$

Some Related Work

- Completeness:
  - P. Cousot, Partial Completeness of Abstract Fixpoint Checking, SARA 2000
  - A. Mycroft, Completeness and predicate-based abstract interpretation, PEPM 1993
  - R. Giacobazzi, F. Ranzato, F. Scozzari, Making abstract interpretations complete, J. ACM, 2000
  - P. Cousot, Abstract Interpretation Based Formal Methods and Future Challenges, LNCS #2000
- Sets of Trees:
  - Laurent Mauborgne, Representation of Sets of Trees for Abstract Interpretation, PhD thesis, 1999

Notions of Completeness

- Soundness:

  $\forall c\ \forall \phi\ [\exists a\ \rho(c,a) \wedge a \vDash \phi] \Rightarrow c \vDash \phi$

- Completeness “for one property”:

  $\forall c\ \forall \phi\ [\exists a\ \rho(c,a) \wedge a \vDash \phi] \Leftrightarrow c \vDash \phi$

- Completeness “for all properties” (cf. Giacobazzi et. al.):

  $\forall c\ \exists a\ \forall \phi\ [\rho(c,a) \wedge a \vDash \phi] \Leftrightarrow c \vDash \phi$

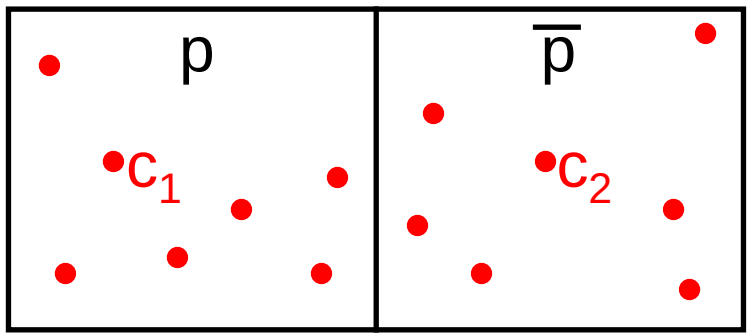
- Strong Preservation:

  $\forall c\ \forall a\ \rho(c,a) \Rightarrow [\forall \phi\ a \vDash \phi \Leftrightarrow c \vDash \phi]$

3-Valuedness Rules!

$c_1 \in p$ so $p(c_1)$ = true

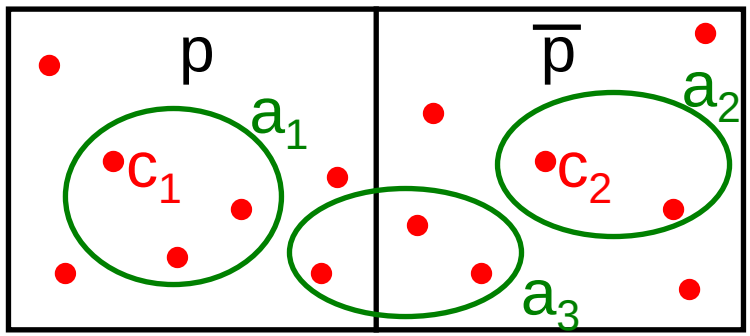
$c_2 \in \overline{p}$ so $p(c_2)$ = false



3-Valuedness Rules...

$c_1 \in p$ so $p(c_1)$ = true

$c_2 \in \overline{p}$ so $p(c_2)$ = false

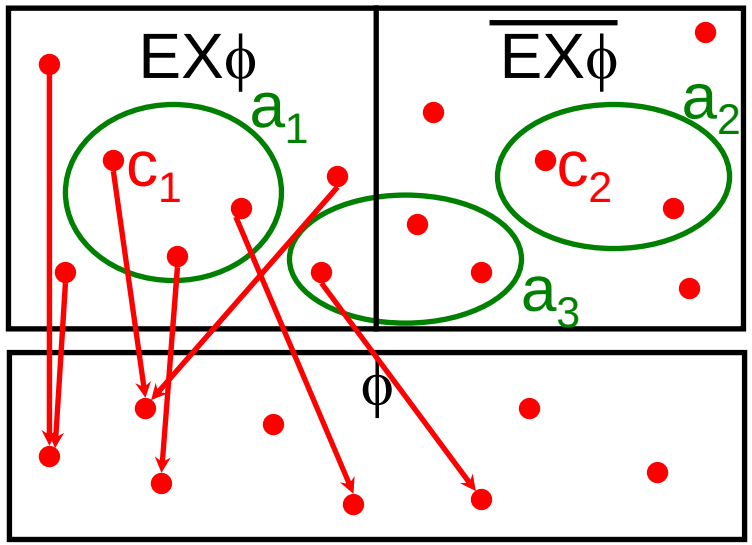


$a_1 \subseteq p$ so $p(a_1)$ = true

$a_2 \subseteq \overline{p}$ so $p(a_2)$ = false

$a_3 \not\subseteq p$, $a_3 \not\subseteq \overline{p}$ so $p(a_3)$ = maybe

3-Valuedness Rules...



$a_1 \subseteq EX\phi$ so $R(a_1,\phi)$ = true

$a_2 \subseteq \overline{EX\phi}$ so $R(a_2,\phi)$ = false

$a_3 \not\subseteq EX\phi$, $a_3 \not\subseteq \overline{EX\phi}$ so $R(a_3,\phi)$ = maybe

From Precision to Completeness

- *completeness*:

  $\forall$ c, ϕ: if c $\models \phi$ then $\exists$ *finite* a: ρ(c,a) and a $\models^{\alpha} \phi$

- Trivial solution:

  A $\triangleq$ L (the property logic)

  ρ(c, ϕ) $\triangleq$ c $\models \phi$

  $\phi \models^{\alpha} \psi \triangleq \psi = \phi$

- objects in A should be some sort of transition systems (like C)
- $\models^{\alpha}$ should be like $\models$

Abstraction Frameworks

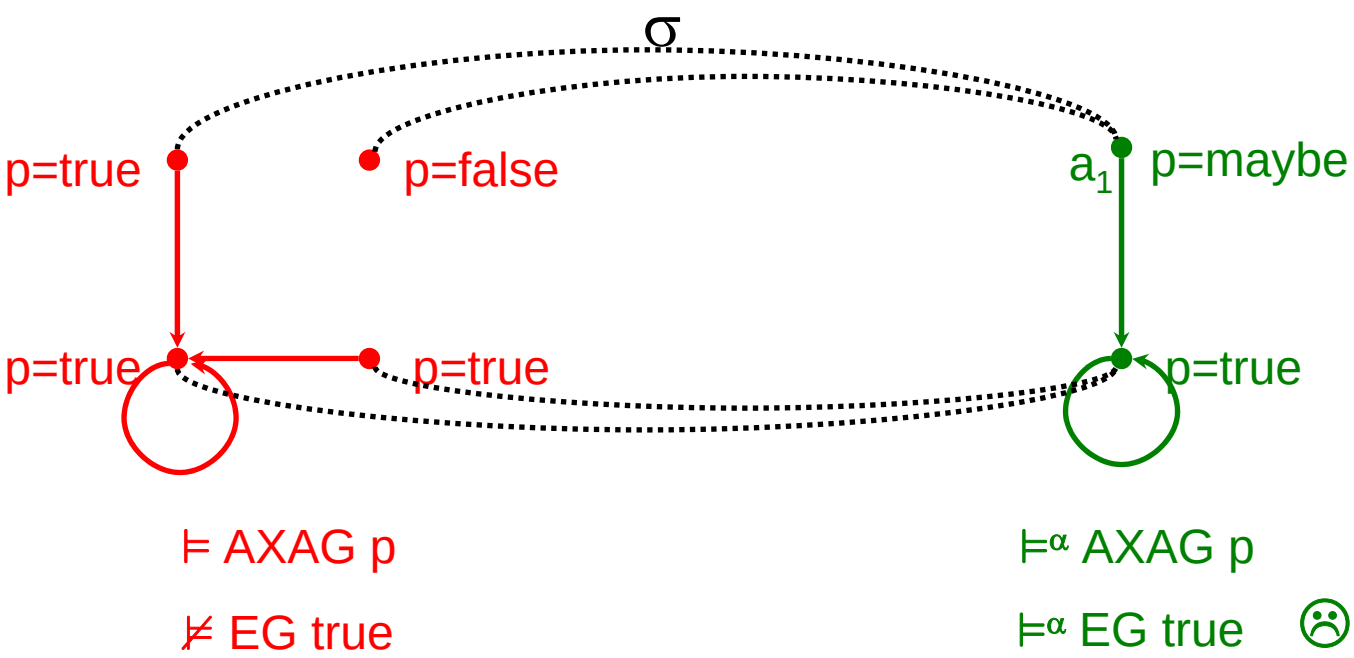
Ingredients:

- concrete objects $c \in C$
- abstract objects (aka abstractions) $a \in A$
- abstraction relation: $\rho \subseteq C \times A$
- properties: $\phi \in L$
- interpretations: $\models \subseteq C \times L$ $\models^{\alpha} \subseteq A \times L$

Requirements:

- soundness (preservation, safety):
  $\forall c\ \forall \phi\ [\exists a\ \rho(c,a) \wedge a \models \phi] \Rightarrow c \models \phi$
- completeness ("per property"):
  $\forall c\ \forall \phi\ [\exists$ *finite* $a\ \rho(c,a) \wedge a \models \phi] \Leftarrow c \models \phi$

Framework 1: Simulation

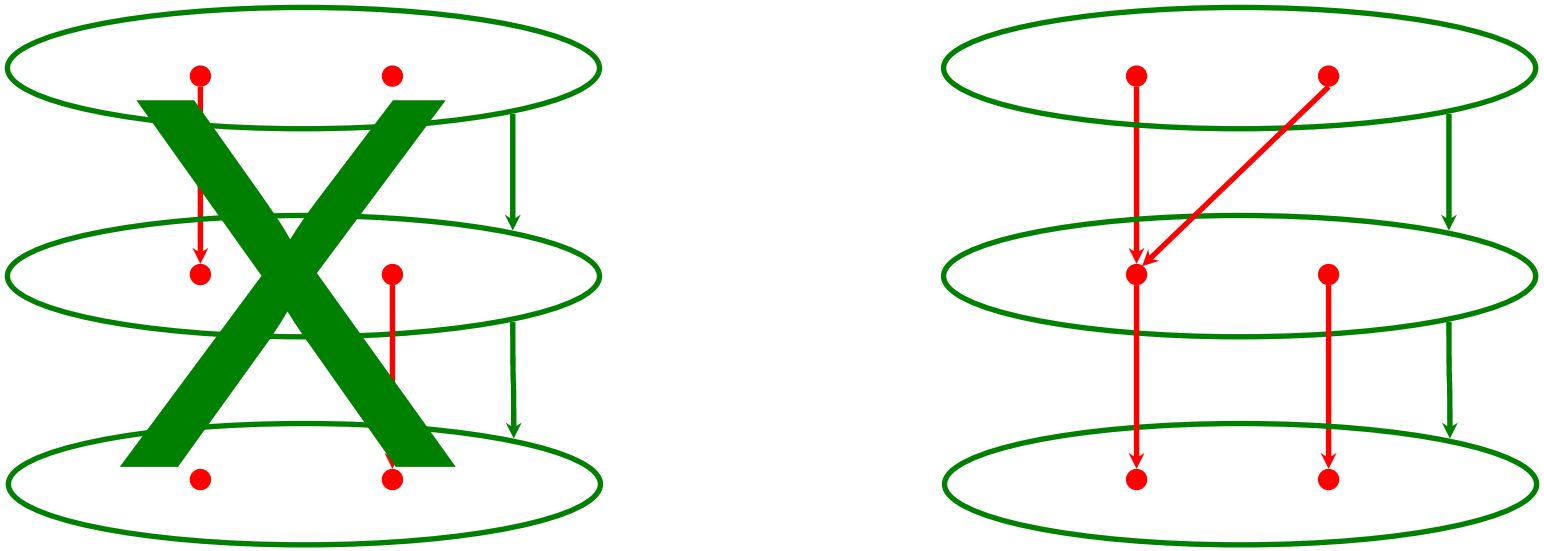


A : Partial Kripke structures (3-valued propositions)

L : universal fragment of BTL (via positive normal form)

$\models^{\alpha}$: 3-valued (e.g. $[a_1 \models^{\alpha} p] = \text{maybe}$)

Preserving universal + existential properties



$$\exists\, c \in a \;\; \exists\, d \in b \;\; c \to d \quad \Rightarrow \quad a \to b$$

$$\forall\, c \in a \;\; \exists\, d \in b \;\; c \to d \quad \Leftarrow \quad a \to b$$

\+

bisimulation... ☹

Framework 3: Mixed Simulation

A : Mixed Transition Systems (XMTS)

- replace $\alpha : C \to A$ by Galois Connection $(C, \subseteq) \underset{\alpha}{\overset{}{\leftrightarrows}} (A, \preceq)$
- revised definition of *must* and *may* (consequence: *must* $\not\subseteq$ *may*)
- this allows for an *optimality* result

(rest as before:)

L : BTL

$\models^{\alpha}$: 3-valued; interpret A quantifiers over *may* paths, and E quantifiers over *must* paths