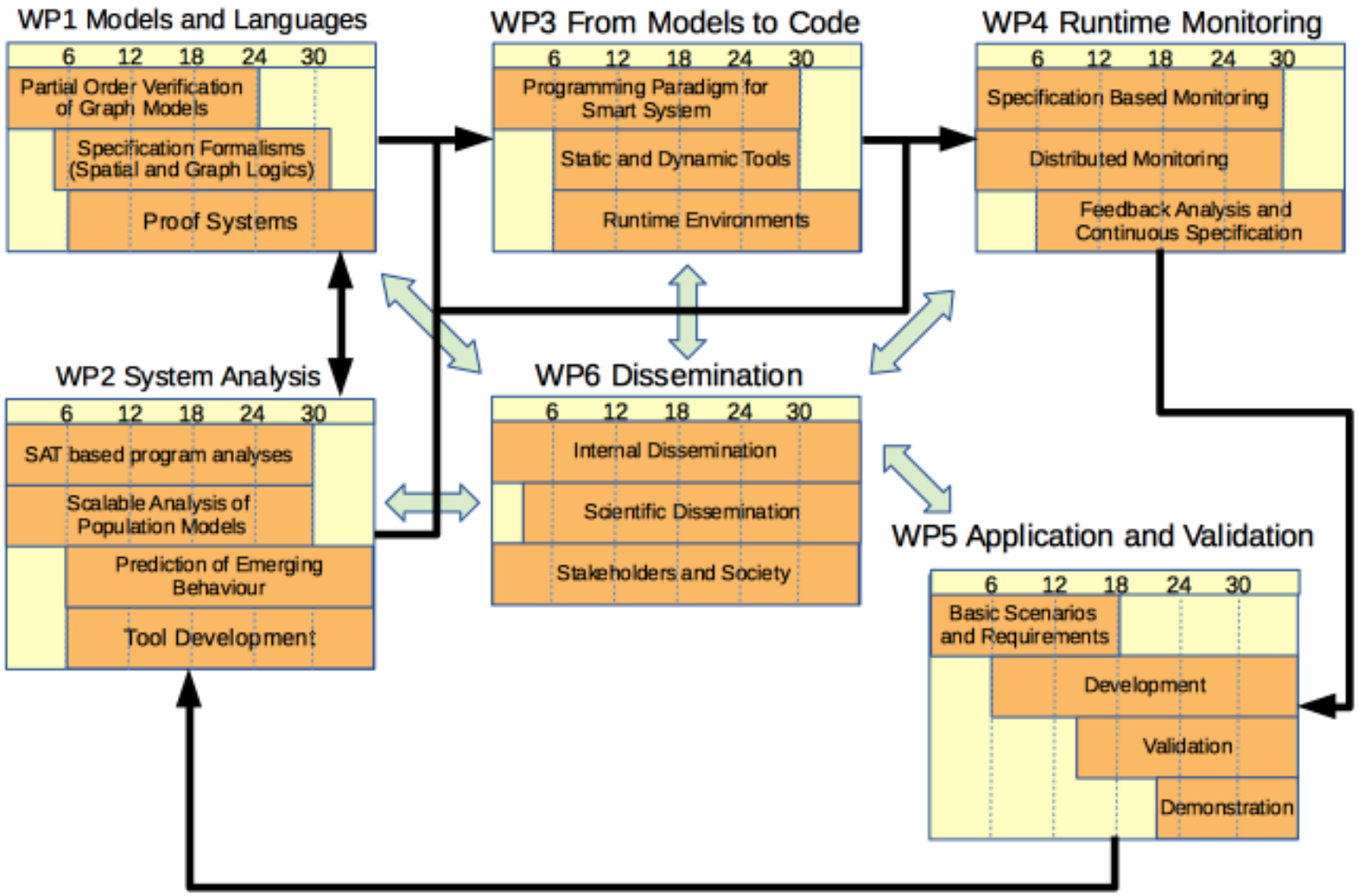
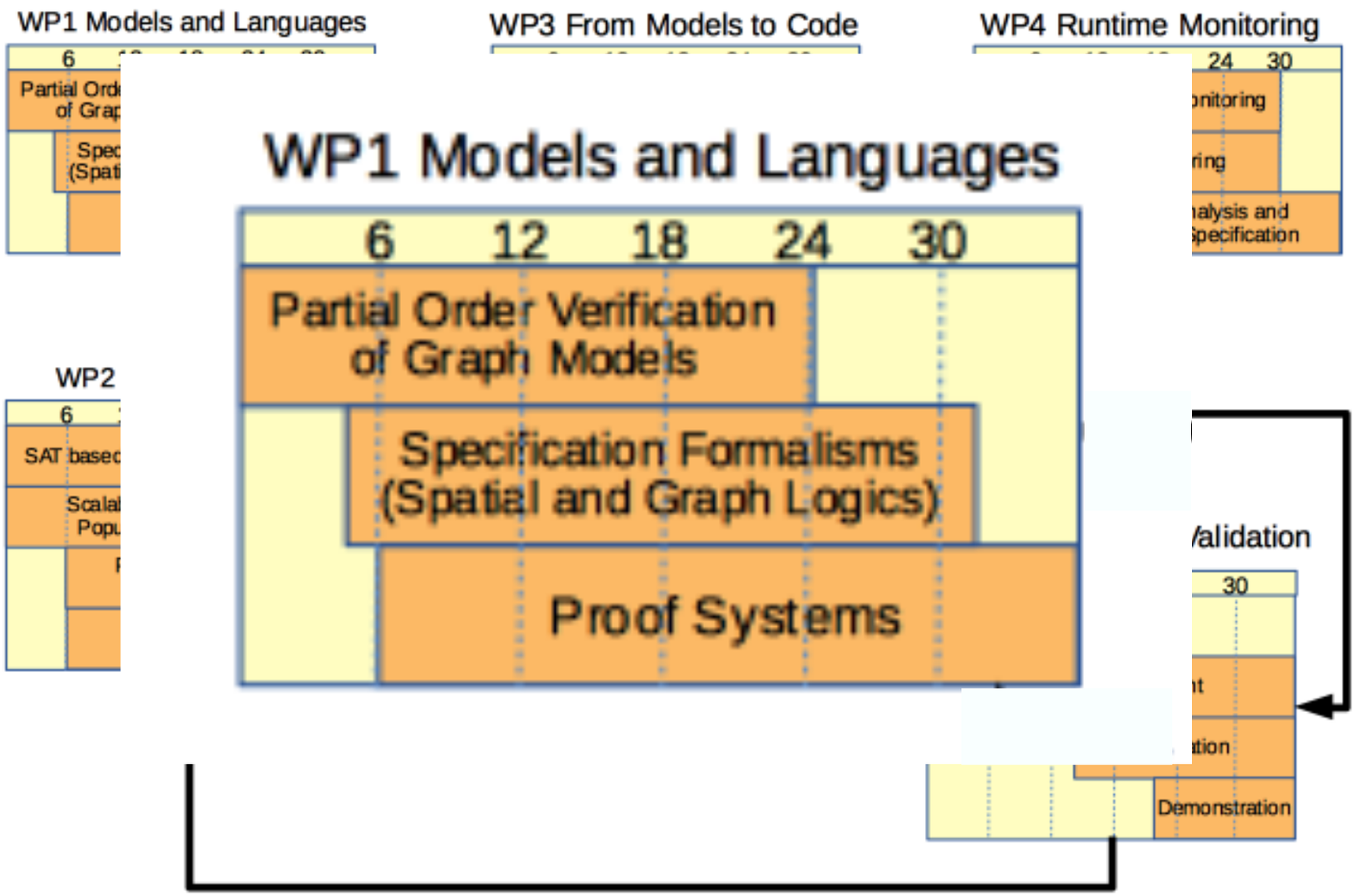
# An introduction to WP1

Fabio Gadducci — UniPi
(Michele Loreti — UniCam
Diego Latella — ISTI
Davide Castelnovo — UniUd)

# WP1 - Models and Languages



# WP1 - Models and Languages



# design-time analysis

- Graph-based specification and verification of spatial and temporal properties
  - Smart systems: sets of spatially distributed entities that evolve in time.
  - Complex requirements specified concisely by modal logics that predicate over the spatial and the temporal domain, exploiting topological concepts to keep descriptions close to the target.
  - Partial-order reduction and unfolding semantics used to face scalability problems in verification.
  - Extending graph-based formalisms to describe systems that can (possibly simultaneously) require the merging or the unbounded replication of components.
  - Most existing logics exhibiting both temporal and topological features and interpreted over graphs have a complex syntax and semantics, making them difficult to use for the programmer.
  - Developing proof systems for logics with temporal and topological operators is difficult. Besides being implementable, proof systems yield a better understanding of the logical operators.

# from the WP1 proposal

- To exploit expressiveness and naturalness of graphical models in the analysis and specification of smart systems, we need to address issues concerning the applicability of these formalisms to the scenarios of WP5. We will extend the concurrency theory for graph manipulation formalisms to support merging or replication of substructures and to apply the techniques based on unfolding semantics and partial order reduction to the verification of smart systems.
- […] we will identify fragments of existing logics (and their extensions to express separation, causality, and spatiality in distributed systems) interpreted over graphs that provide a good compromise between expressiveness and effectiveness, supporting efficient model checking algorithms and runtime monitoring.
- In order to support semi-automated verification of smart systems, we will develop proof systems for the logics considered in the project

# quantified modal logic

# quantified modal logic

- almost a classical idea [Barcan 1946, Carnap 1947]

# quantified modal logic

- almost a classical idea [Barcan 1946, Carnap 1947]
  - a lot of research by logicians, philosophers and linguists

# quantified modal logic

- almost a classical idea [Barcan 1946, Carnap 1947]
  - a lot of research by logicians, philosophers and linguists
- interplay between the modal and existential operators

# quantified modal logic

- almost a classical idea [Barcan 1946, Carnap 1947]
  - a lot of research by logicians, philosophers and linguists
- interplay between the modal and existential operators
  - degrees of interaction: some combinations may be forbidden

# quantified modal logic

- almost a classical idea [Barcan 1946, Carnap 1947]
  - a lot of research by logicians, philosophers and linguists
- interplay between the modal and existential operators
  - degrees of interaction: some combinations may be forbidden
- Kripke models: states are worlds and contain entities

# quantified modal logic

- almost a classical idea [Barcan 1946, Carnap 1947]
  - a lot of research by logicians, philosophers and linguists
- interplay between the modal and existential operators
  - degrees of interaction: some combinations may be forbidden
- Kripke models: states are worlds and contain entities
- transitions “express” how entities are related across worlds

# quantified modal logic

- almost a classical idea [Barcan 1946, Carnap 1947]
  - a lot of research by logicians, philosophers and linguists
- interplay between the modal and existential operators
  - degrees of interaction: some combinations may be forbidden
- Kripke models: states are worlds and contain entities
- transitions “express” how entities are related across worlds
  - familiar in e.g. recent works on automata

# monadic second order qml

# monadic second order qml

- second-order variables, yet no (second-order) predicates

# monadic second order qml

- second-order variables, yet no (second-order) predicates
  - except for “entity x belongs to set of entities X”

# monadic second order qml

- second-order variables, yet no (second-order) predicates
  - except for “entity x belongs to set of entities X”
  - equalities (and other properties) can be recovered

# monadic second order qml

- second-order variables, yet no (second-order) predicates
  - except for “entity x belongs to set of entities X”
  - equalities (and other properties) can be recovered
- also a (minimal) fixed point

# monadic second order qml

- second-order variables, yet no (second-order) predicates
  - except for “entity x belongs to set of entities X”
  - equalities (and other properties) can be recovered
- also a (minimal) fixed point
  - for reachability and birth&death of entities

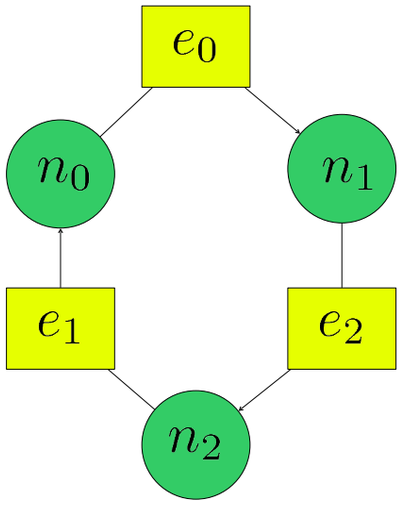
# monadic second order qml

- second-order variables, yet no (second-order) predicates
  - except for “entity x belongs to set of entities X”
  - equalities (and other properties) can be recovered
- also a (minimal) fixed point
  - for reachability and birth&death of entities
- an algebra of terms

# monadic second order qml

- second-order variables, yet no (second-order) predicates
  - except for “entity x belongs to set of entities X”
  - equalities (and other properties) can be recovered
- also a (minimal) fixed point
  - for reachability and birth&death of entities
- an algebra of terms
  - yet, always studied for (flavours of) graphs!!

# monadic second order qml



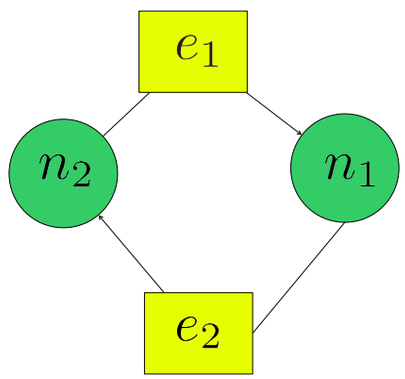

**states = graphs**

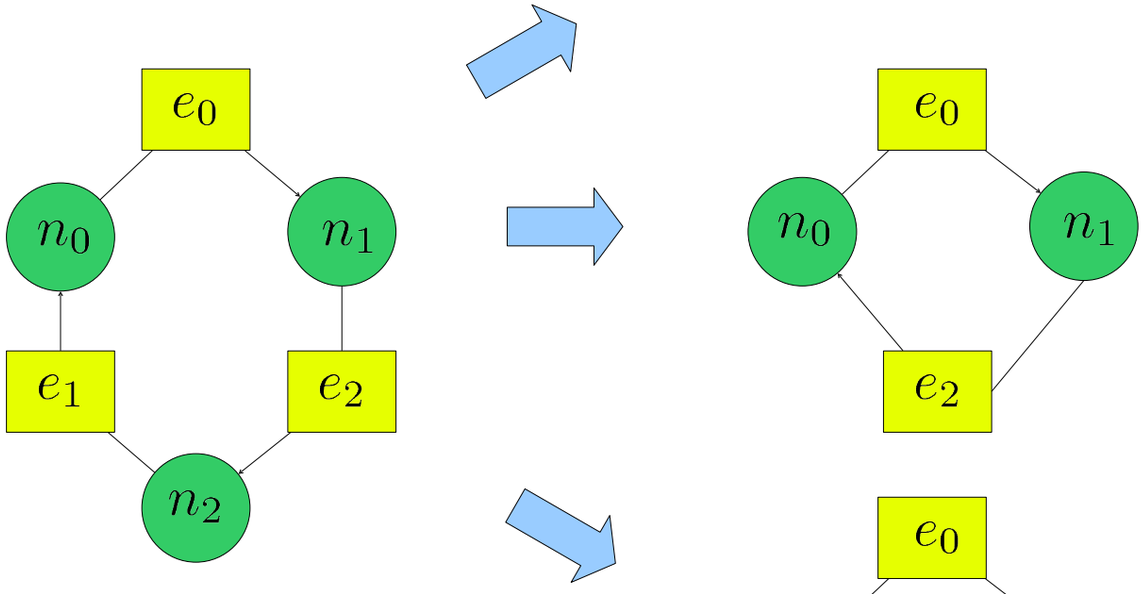
# monadic second order qml

**transitions**

**=**

**morphisms**



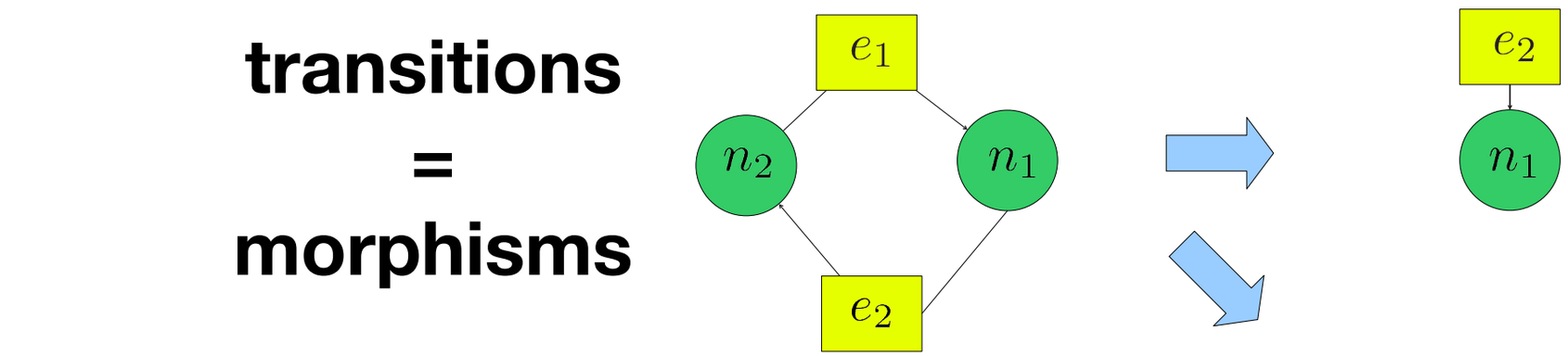

**states = graphs**

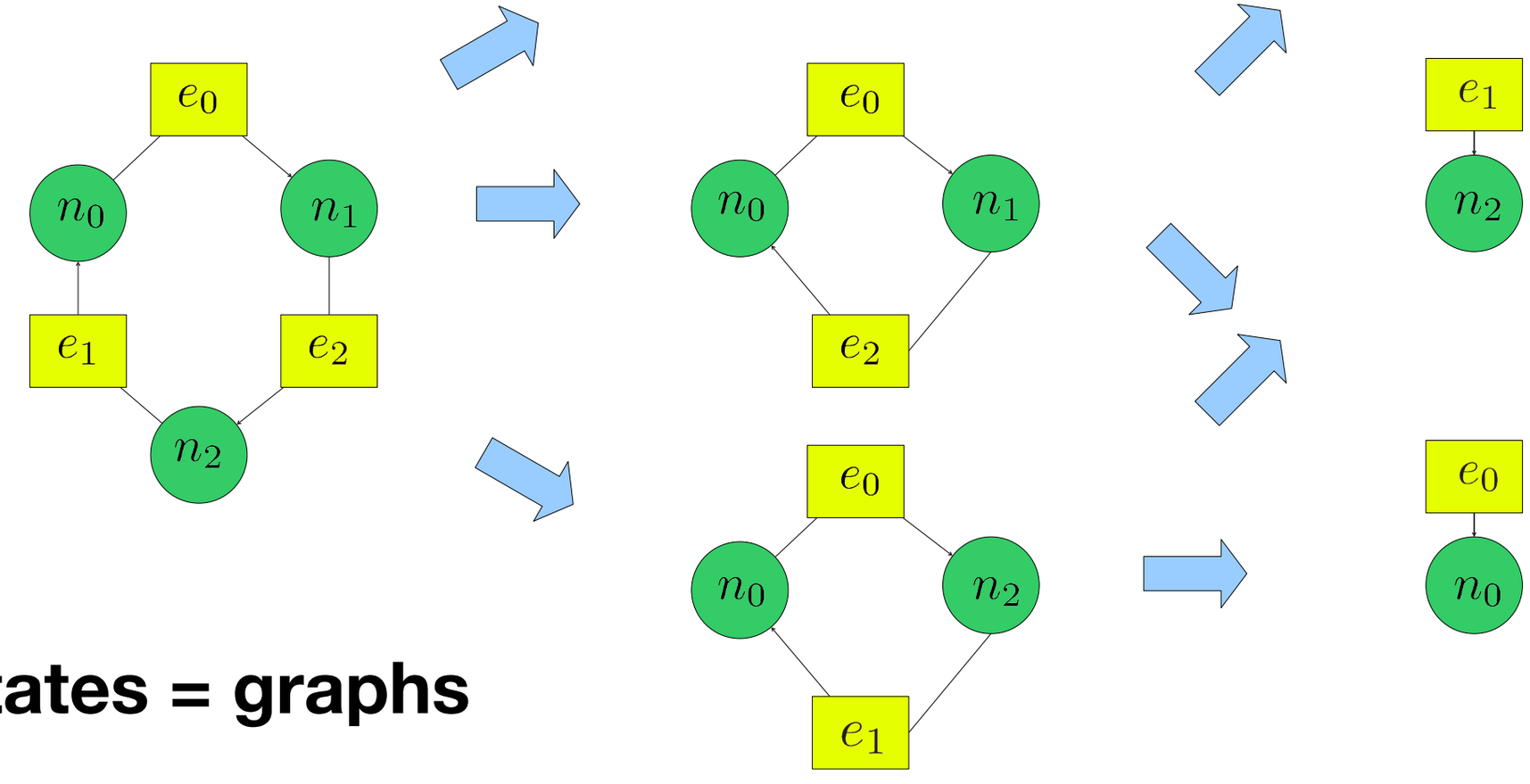
# monadic second order qml

**transitions**

**=**

**morphisms**




**states = graphs**

# the horizon…

# the horizon…

- streamline syntax and semantics

# the horizon…

- streamline syntax and semantics
  - fixed points vs. until operators [proposals in the literature]

# the horizon…

- streamline syntax and semantics
  - fixed points vs. until operators [proposals in the literature]
  - what about “freshness”? [annotated next? less proposals…]

# the horizon…

- streamline syntax and semantics
  - fixed points vs. until operators [proposals in the literature]
  - what about “freshness”? [annotated next? less proposals…]
- introduce (sound&complete) proof systems

# the horizon…

- streamline syntax and semantics
  - fixed points vs. until operators [proposals in the literature]
  - what about “freshness”? [annotated next? less proposals…]
- introduce (sound&complete) proof systems
  - along the whole S/K spectrum [proposals in the literature]

# the horizon…

- streamline syntax and semantics
  - fixed points vs. until operators [proposals in the literature]
  - what about “freshness”? [annotated next? less proposals…]
- introduce (sound&complete) proof systems
  - along the whole S/K spectrum [proposals in the literature]
- integrate with spatial properties [predicates *on a* world]

# the horizon…

- streamline syntax and semantics
  - fixed points vs. until operators [proposals in the literature]
  - what about “freshness”? [annotated next? less proposals…]
- introduce (sound&complete) proof systems
  - along the whole S/K spectrum [proposals in the literature]
- integrate with spatial properties [predicates *on a* world]
- find abstract characterisation of Kripke models [simpler semantics]

# the horizon…

- streamline syntax and semantics
  - fixed points vs. until operators [proposals in the literature]
  - what about “freshness”? [annotated next? less proposals…]
- introduce (sound&complete) proof systems
  - along the whole S/K spectrum [proposals in the literature]
- integrate with spatial properties [predicates *on a* world]
- find abstract characterisation of Kripke models [simpler semantics]
- check the effectiveness with the chosen scenarios