The Dynamic Turn in Strategy Logics

Rustam Galimullin
University of Bergen, Norway

Maksim Gladyshev
Utrecht University

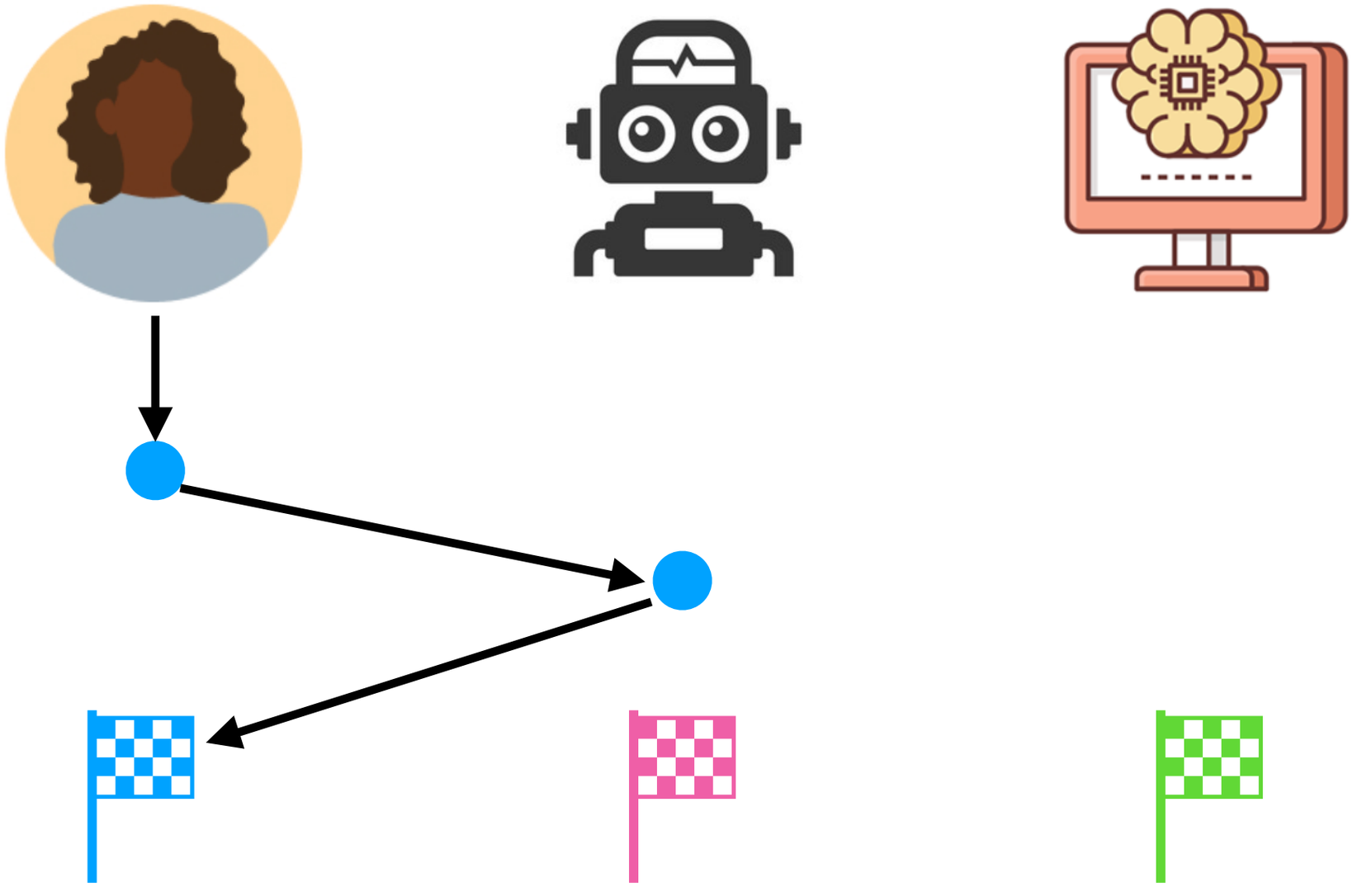
Munyque Mittelmann
CNRS, U. Sorbonne Paris Nord

Nima Motamed
Utrecht University

Strategy Logics

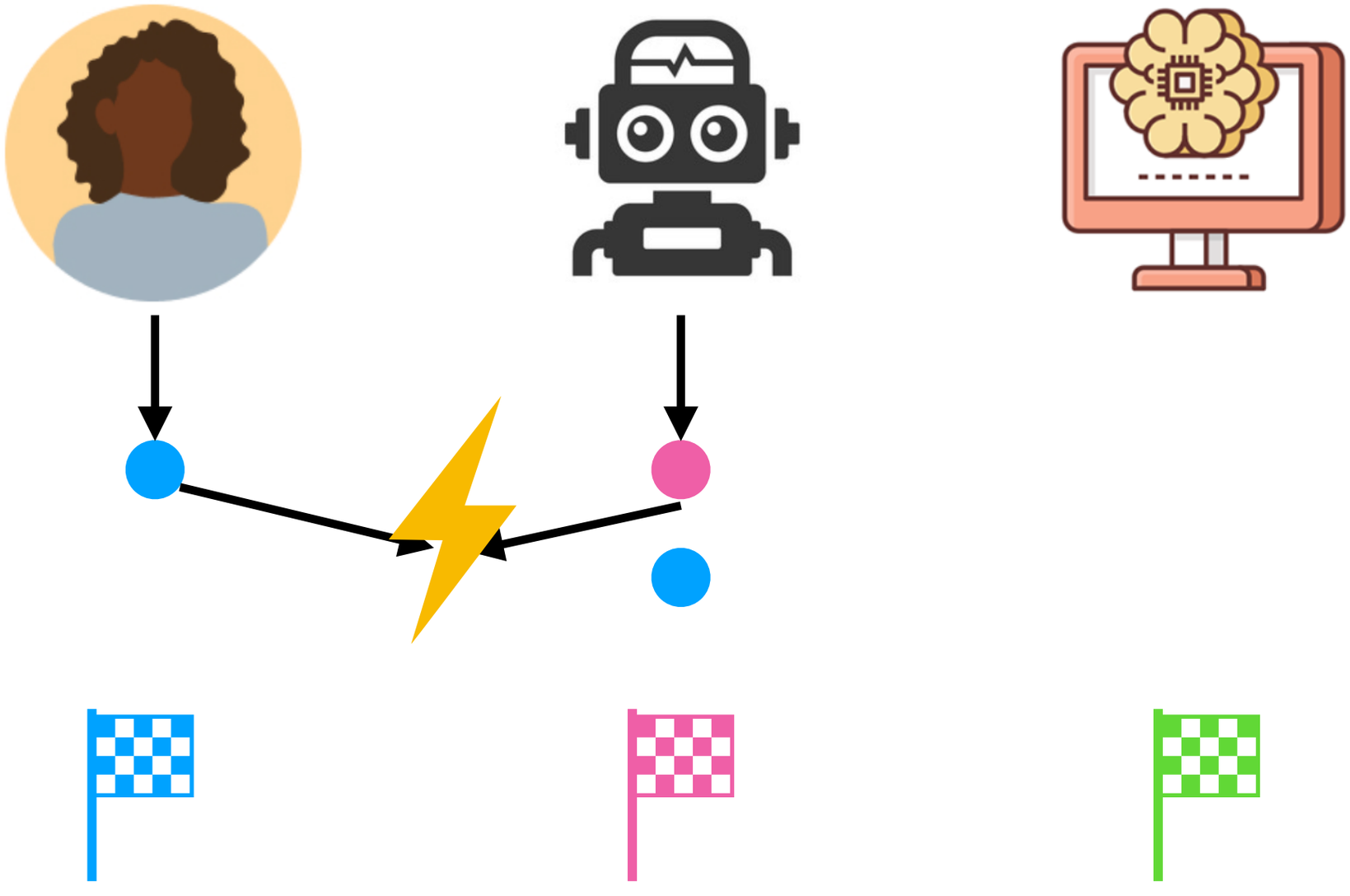
Strategy Logics

Strategy logics (SLs) is a family of formal frameworks devised for specification and verification of multi-agent systems



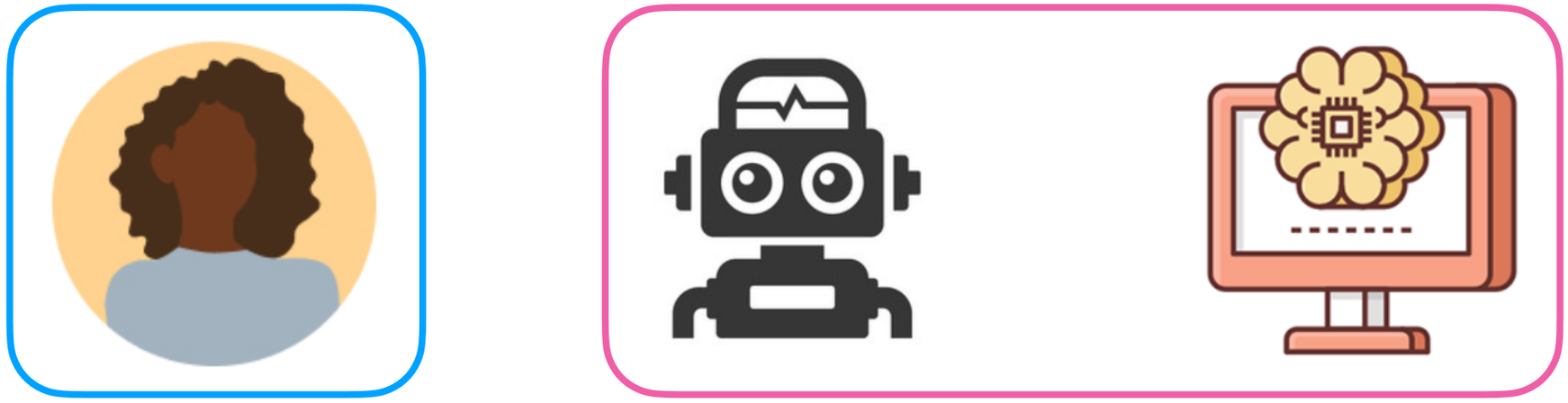
Strategy Logics

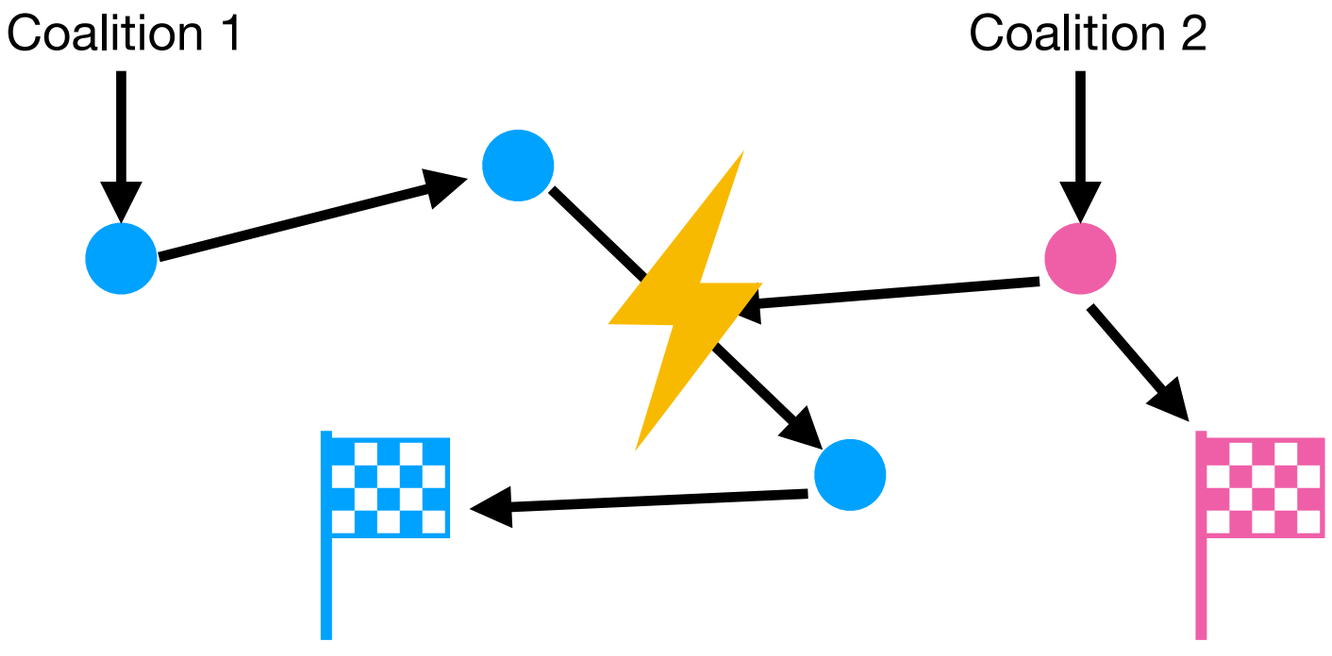
Strategy logics (SLs) is a family of formal frameworks devised for specification and verification of multi-agent systems



Strategy Logics

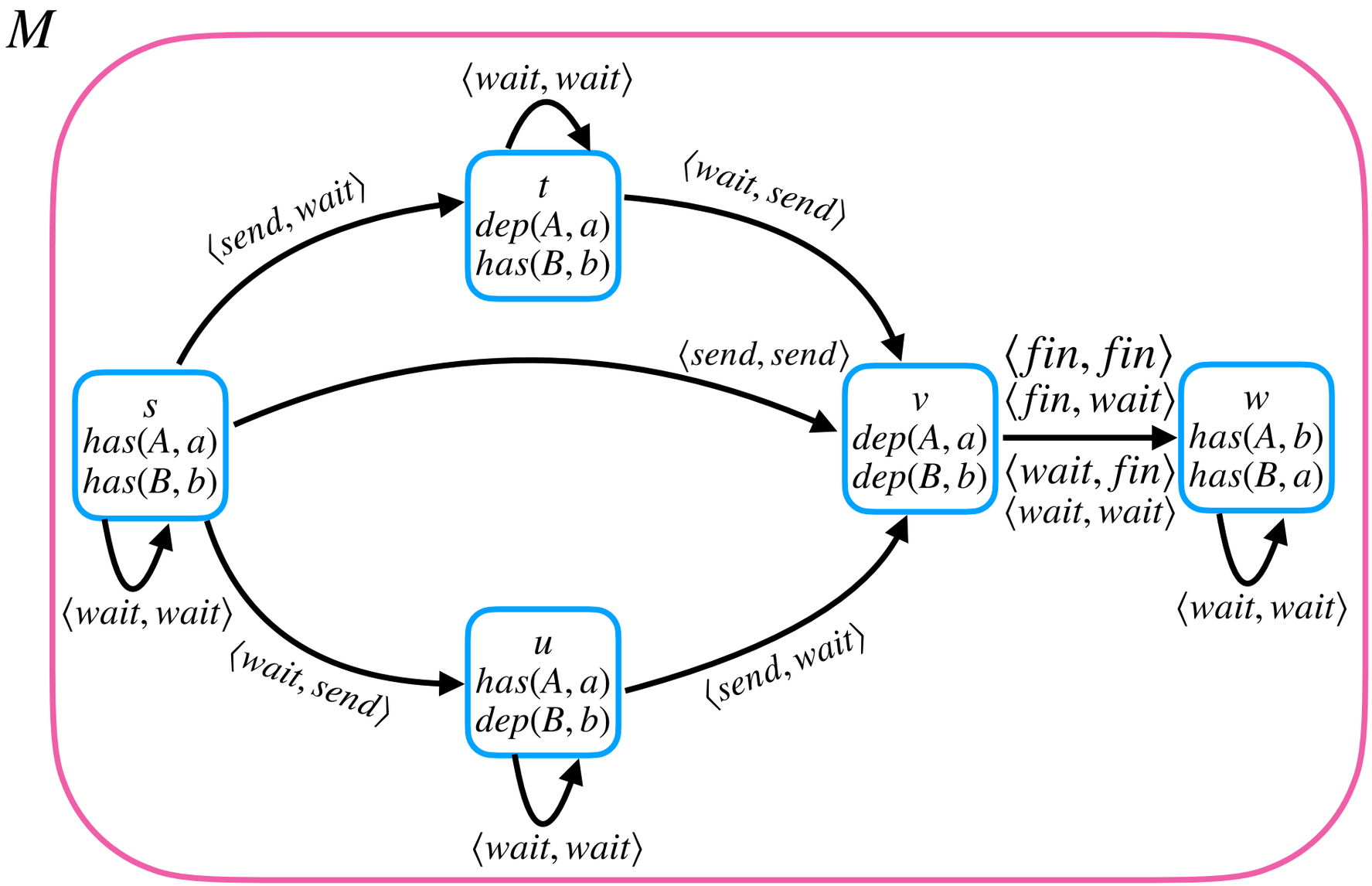
Strategy logics (SLs) is a family of formal frameworks devised for specification and verification of multi-agent systems





Strategy Logics

We focus on logics interpreted on Concurrent Game Models (CGMs) like Coalition Logic, ATL, etc



Strategy Logics

We focus on logics interpreted on Concurrent Game Models (CGMs) like Coalition Logic, ATL, etc

$\langle\langle C \rangle\rangle \varphi$: coalition C **has** a strategy to ensure φ **no matter what** agents outside of the coalition do

$[\![C]\!] \varphi$: **whatever** coalition C does, agents outside of the coalition **have** a strategy to ensure φ

Functionality. Authorised or honest agents can achieve their goals

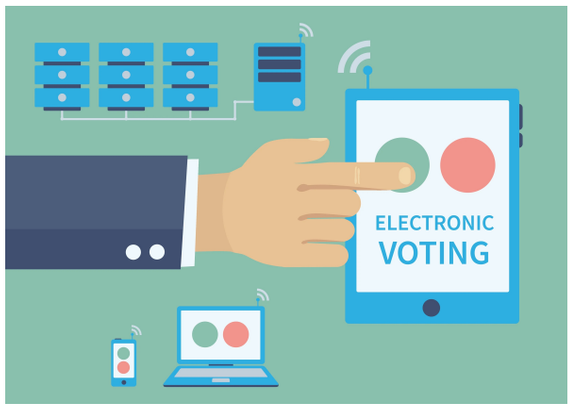
Security. Unauthorised or malicious agents can not achieve undesirable results

Strategy Logics

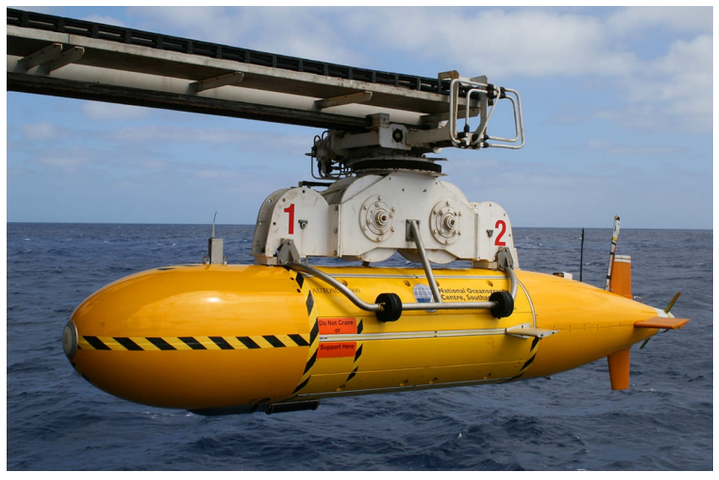
We focus on logics interpreted on Concurrent Game Models (CGMs) like Coalition Logic, ATL, etc

Functionality. Authorised or honest agents can achieve their goals

Security. Unauthorised or malicious agents can not achieve undesirable results



SELENE e-voting



National Oceanography Centre

Ezeikiel et al. *Verifying fault tolerance and self-diagnosability of an autonomous underwater vehicle*. 2011

Jamroga et al. *Model Checking the SELENE E-Voting Protocol in Multi-agent Logics*. 2018

The Dynamic Turn

(dynamics as in model transformations)

Dynamics in MAS

Normative reasoning: the set of actions is divided into desirable\undesirable

Obstruction logics: a special agent (Demon) can switch off some transitions to prevent an attacker reaching crucial points of a system

Our goal: unifying general approach to reasoning about dynamic phenomena in MAS

$$[U]\varphi$$

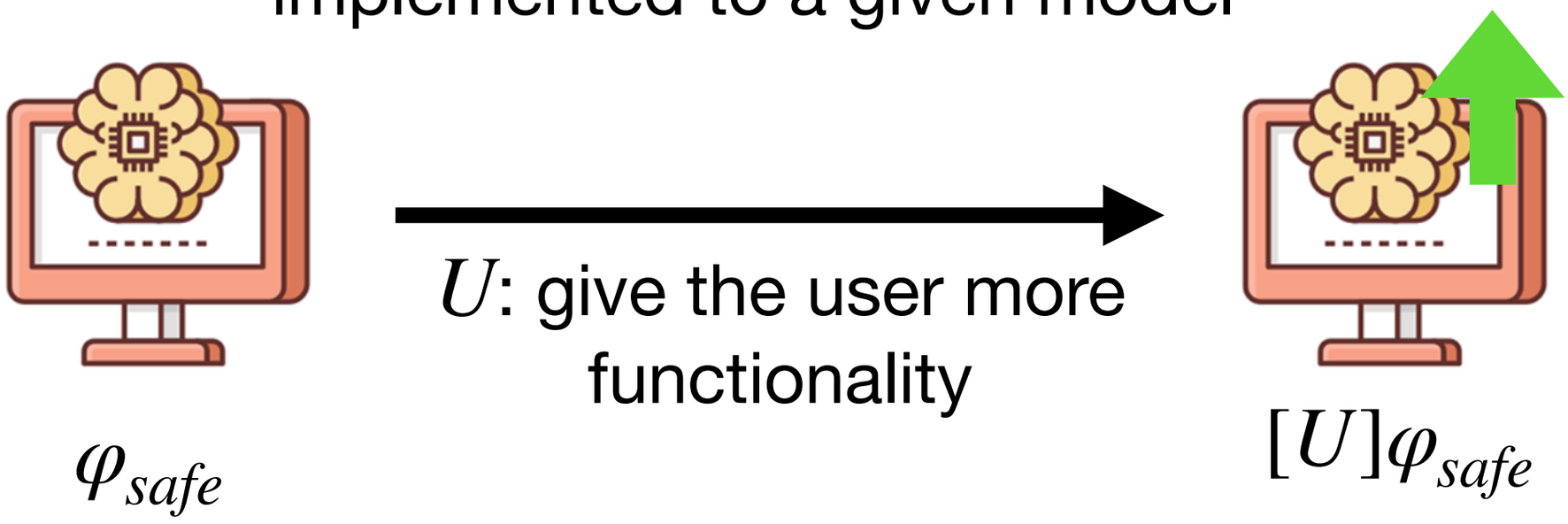
‘After implementing update\upgrade\change U, φ will hold in the resulting updated\upgraded\changed model’

Dynamics in MAS

$$[U]\varphi$$

‘After implementing update\upgrade\change U, φ will hold in the resulting updated\upgraded\changed model’

U is a part of a language, and encapsulates changes to be implemented to a given model

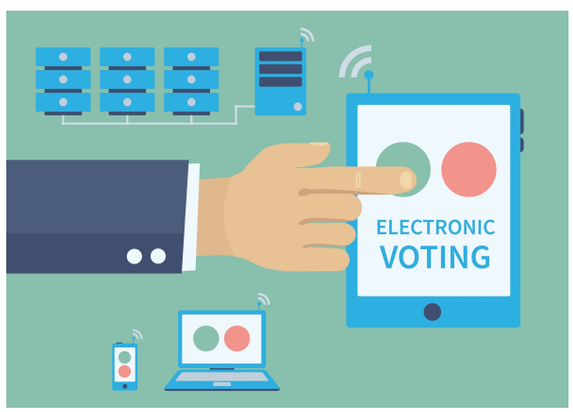


Dynamics in MAS

$$[U]\varphi$$

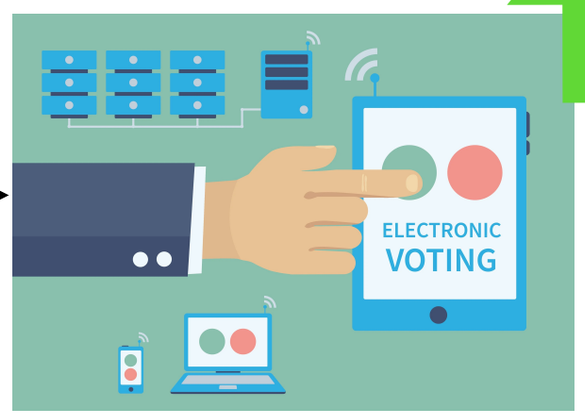
‘After implementing update\upgrade\change U, φ will hold in the resulting updated\upgraded\changed model’

U is a part of a language, and encapsulates changes to be implemented to a given model



$\varphi_{no\_coercion}$

U: switch from single-winner to multiple-winner



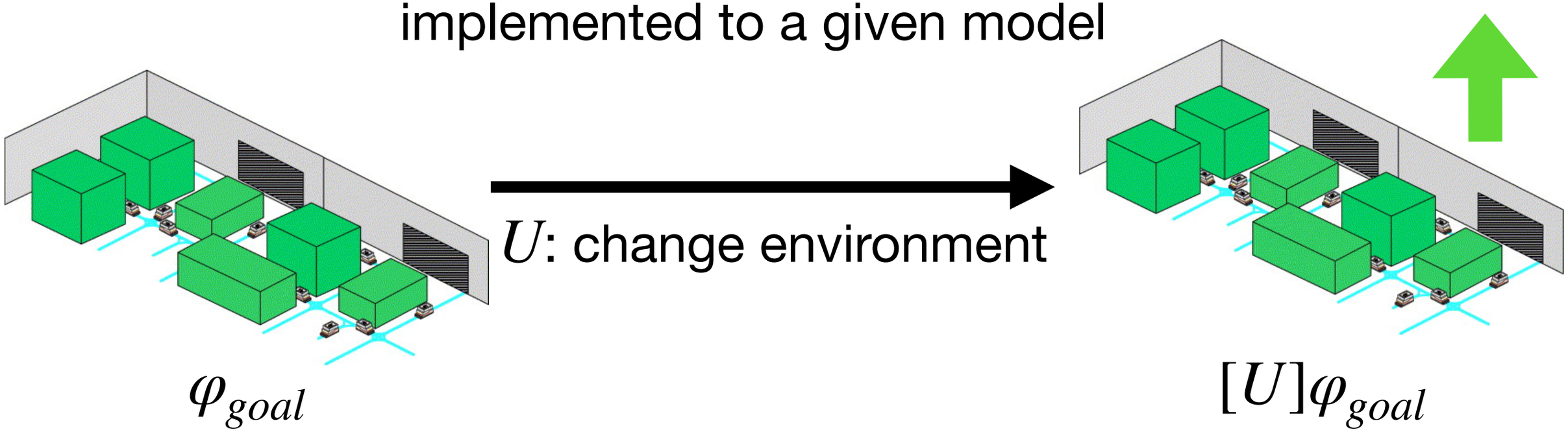
$[U]\varphi_{no\_coercion}$

Dynamics in MAS

$$[U]\varphi$$

‘After implementing update\upgrade\change U, φ will hold in the resulting updated\upgraded\changed model’

U is a part of a language, and encapsulates changes to be implemented to a given model



Why?

$$[U]\varphi$$

High-level description of changes: as updates U are part of a language, they represent a succinct high-level representation of an update, as opposed to a low-level description of particular changes in a model

Universality: updates, as logical formulas, are model agnostic (for the same class of models)

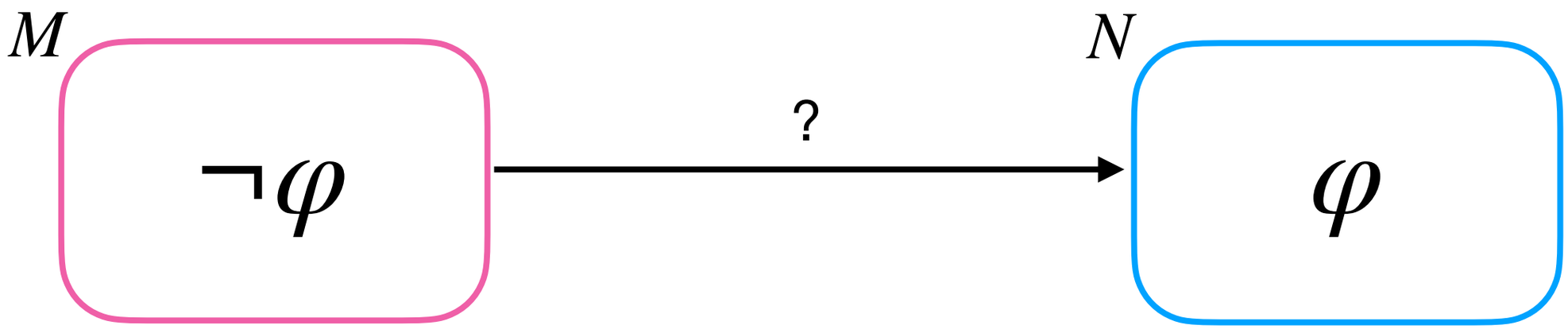
Automated checking: checking whether φ holds after U can be done automatically on a computer, instead of implementing all changes in a model by hand

Challenges

Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates

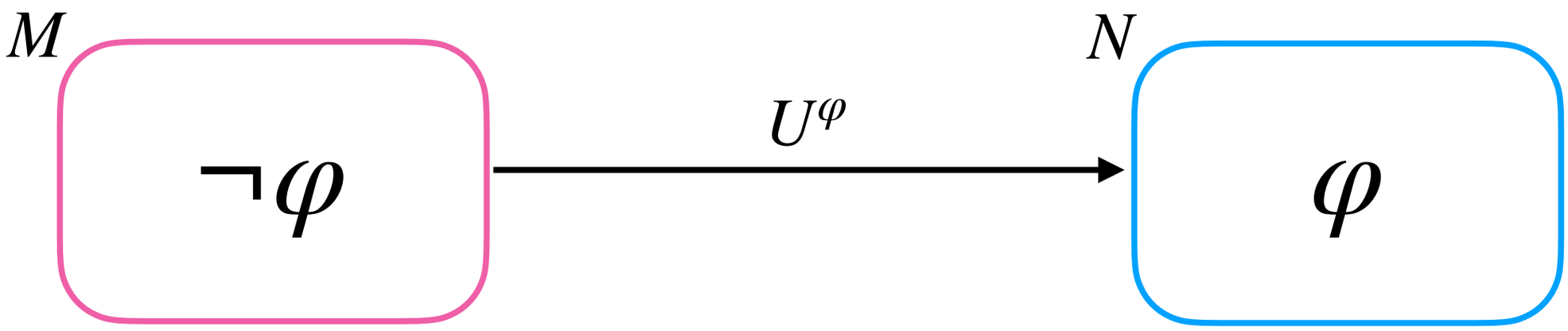
Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates

Automated synthesis\Model repair: given a starting model and a target property φ, construct an update that transforms the model so that the target formula is satisfied; provide minimal synthesis disrupting a model as little as possible



Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates

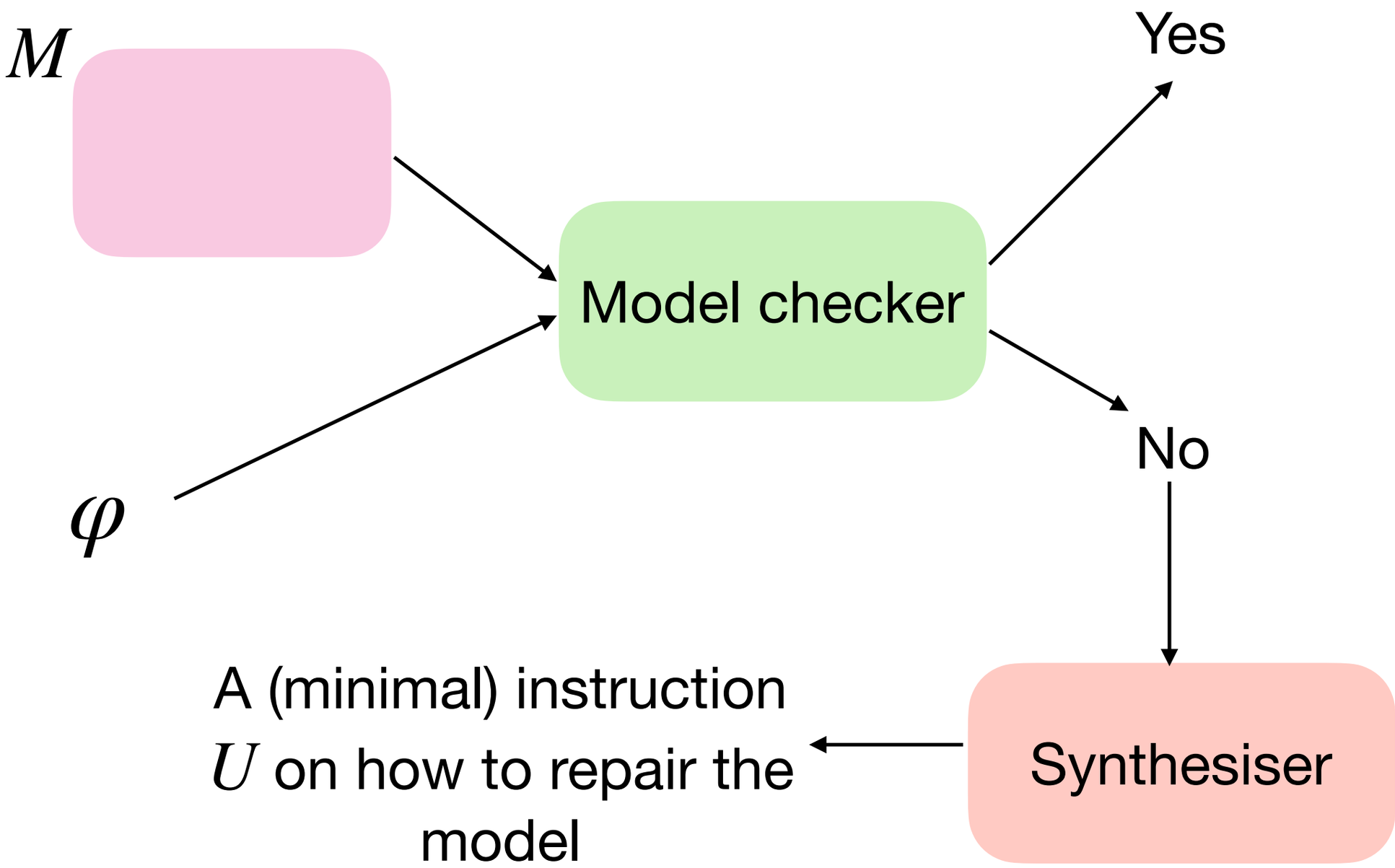
Automated synthesis\Model repair: given a starting model and a target property φ, construct an update that transforms the model so that the target formula is satisfied; provide minimal synthesis disrupting a model as little as possible



Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates

Automated synthesis\Model repair: given a starting model and a target property φ, construct an update that transforms the model so that the target formula is satisfied; provide minimal synthesis disrupting a model as little as possible

Tools for Verification and Synthesis: create extensions of established model checkers (MCMAS, STV, VITAMIN) for new logics; incorporate automated synthesis



Tools for Verification and Synthesis: create extensions of established model checkers (MCMAS, STV, VITAMIN) for new logics; incorporate automated synthesis

The Dynamic Turn in Strategy Logics

Our goal: unifying general approach to reasoning about dynamic phenomena in MAS