Cooperative Reactive Synthesis

Rüdiger Ehlers

University of Bremen & DFKI GmbH

Talk at TU Munich, 01/2016

Based on joint work with Robert Könighofer and Roderick Bloem, published at ATVA 2015 and IROS 2015

Synthesis of reactive systems

$$\mathsf{GF}(u \wedge v) \rightarrow \mathsf{G}(u \leftrightarrow \mathsf{X}\, v)$$

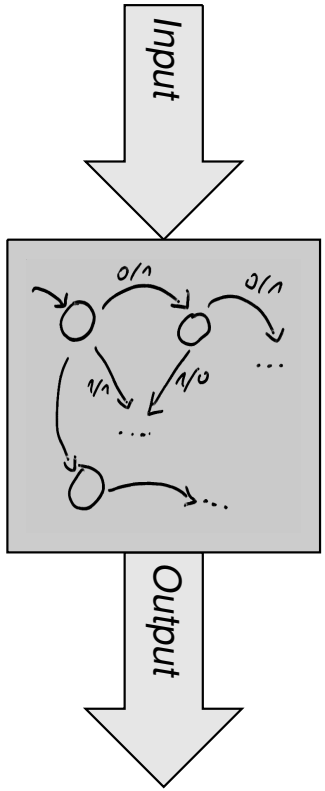
$$+$$

$\mathsf{Input} = \{u, \ldots\}$
$\mathsf{Output} = \{v, \ldots\}$

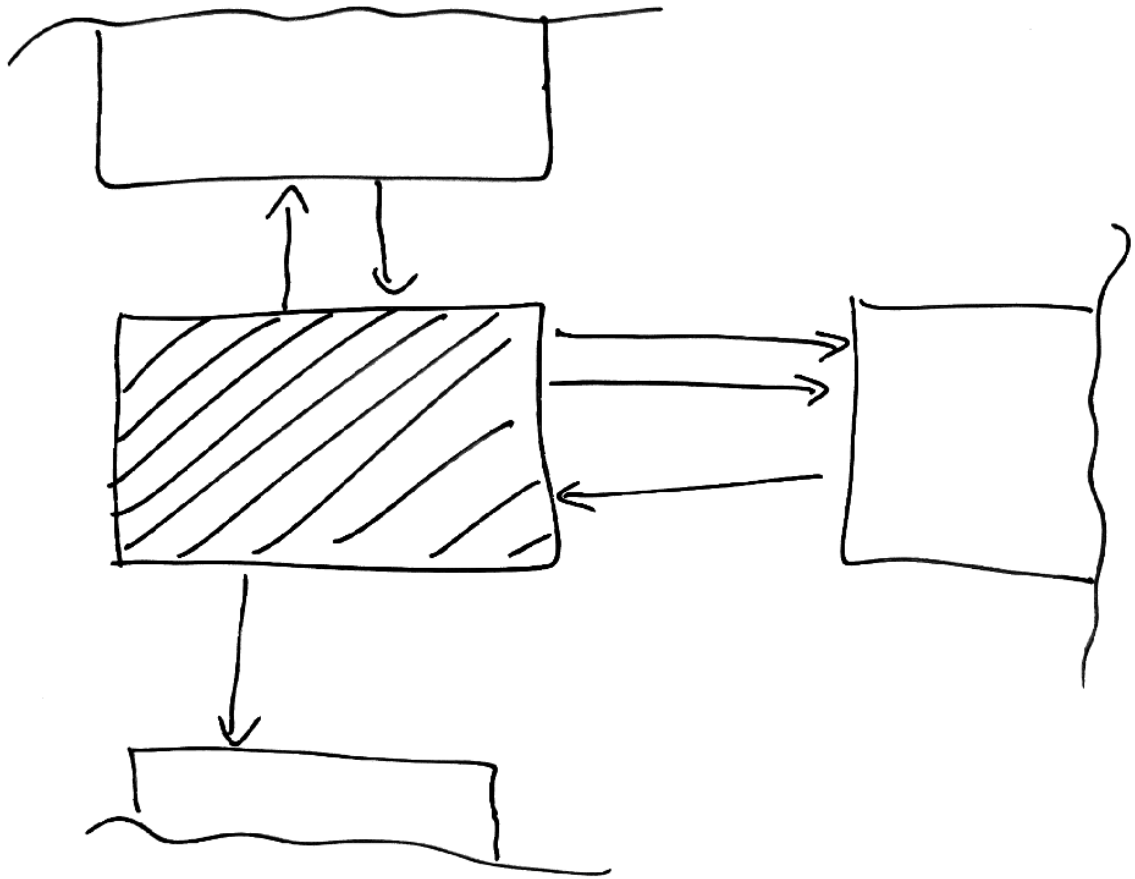
$\Longrightarrow$

Realizable $\longrightarrow$

Not realizable



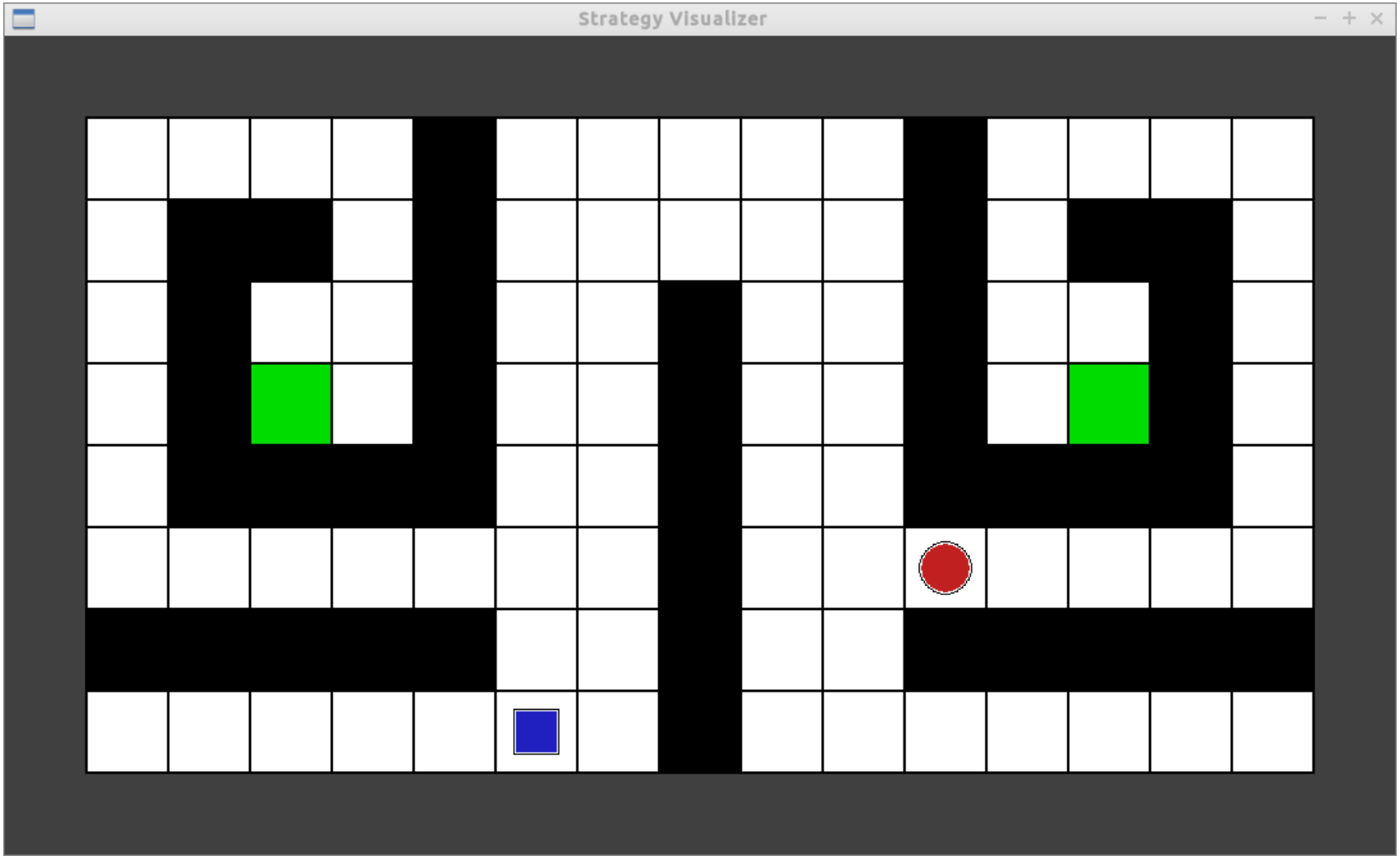
Assumptions and guarantees in specifications



Specification shape

$$\left(\bigwedge \text{Assumptions}\right) \rightarrow \left(\bigwedge \text{Guarantees}\right)$$

Demo



Cooperative implementations

Basic question

How can we synthesize implementations for the system that

- are **correct** (whenever possible) and
- are as **cooperative** with the environment as possible (in a given situation)?

Our solution

- We define a **cooperation hierarchy** that allows us to specify how cooperative an implementation is with the environment.
- We synthesize controllers for some given **cooperation level** in the hierarchy, which consists of one or more *conjuncts*.
- We let the synthesized controller **move up in the hierarchy** whenever possible.

Cooperation levels

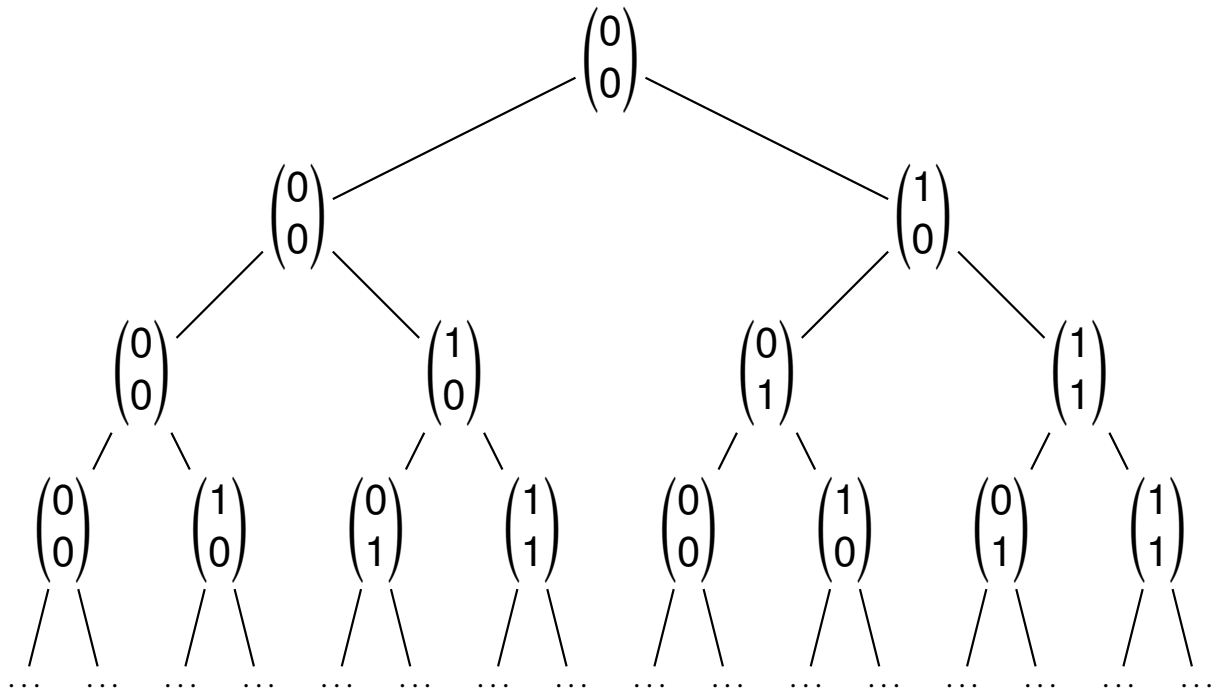
Example cooperation levels

- $\mathcal{A} \rightarrow \mathcal{G}$
- $(\mathcal{A} \rightarrow \mathcal{G}) \wedge \langle E \rangle \mathcal{A}$
- $(\mathcal{A} \rightarrow \mathcal{G}) \wedge \mathsf{G}\langle E \rangle \mathcal{A}$
- $(\mathcal{A} \rightarrow \mathcal{G}) \wedge \mathsf{G}\langle E \rangle \mathcal{G}$

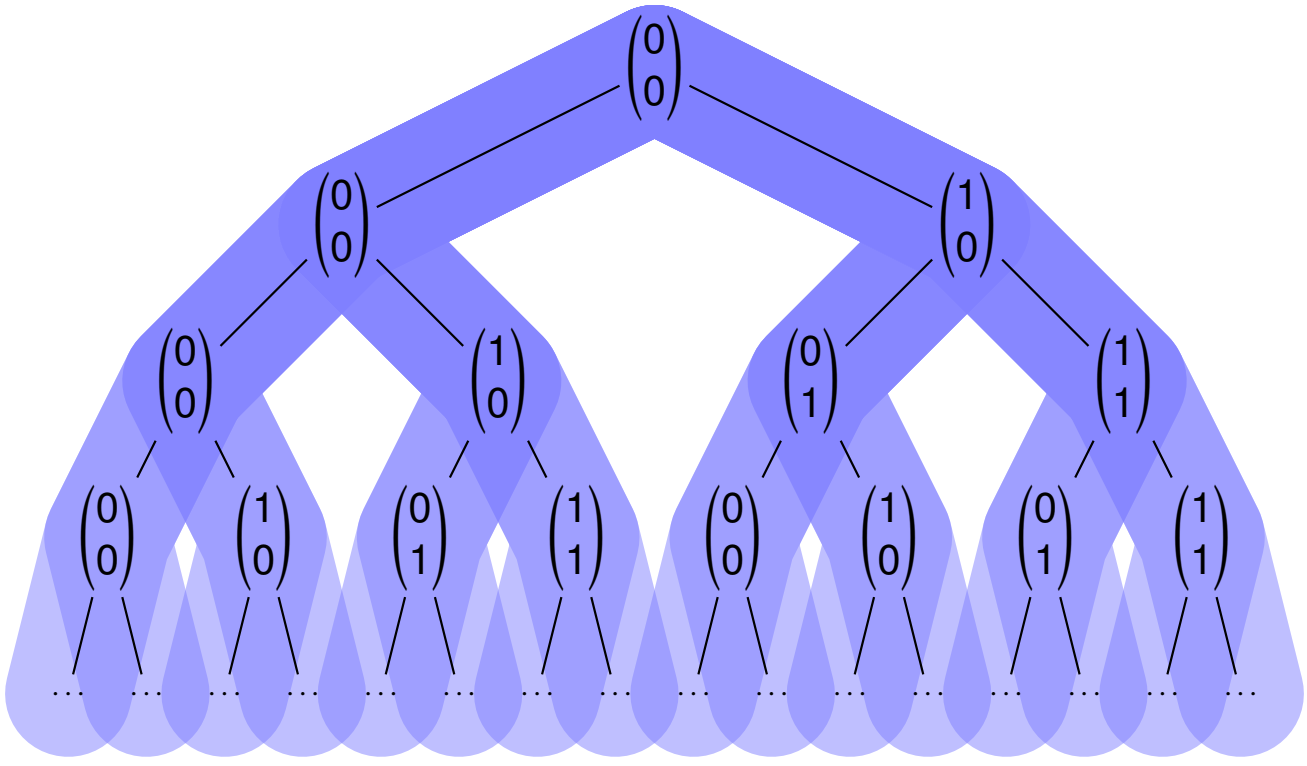
Cooperation levels without specification satisfaction

- $\mathcal{A} \wedge \mathsf{G}\langle E \rangle \mathcal{G}$
- $\mathsf{G}\langle E \rangle (\mathcal{A} \wedge \mathcal{G})$
- $\mathsf{G}\langle E \rangle \mathcal{A} \wedge \mathsf{G}\langle E \rangle \mathcal{G}$

Cooperation level conjuncts



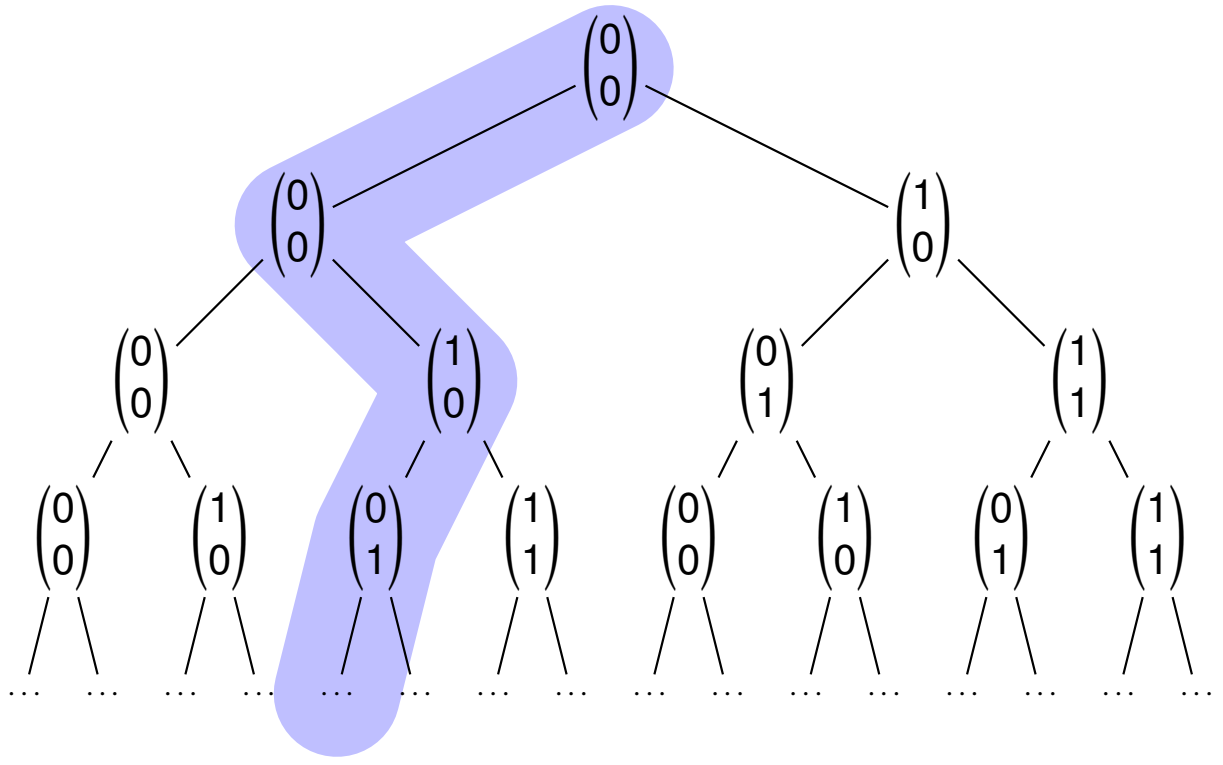
Cooperation level conjuncts



Linear-time properties

$\mathcal{A}$ $\mathcal{G}$ $\mathcal{A} \rightarrow \mathcal{G}$ $\mathcal{A} \vee \mathcal{G}$ $\mathcal{A} \wedge \mathcal{G}$

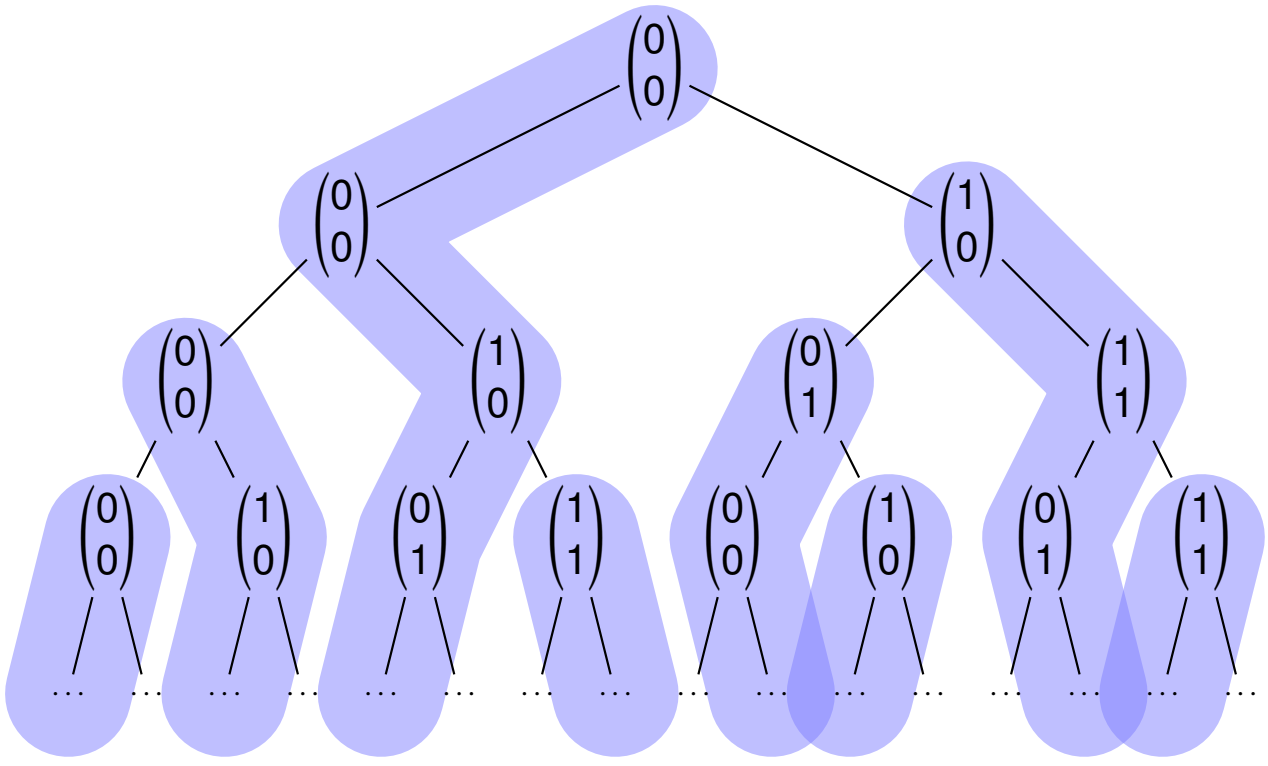
Cooperation level conjuncts



$\langle E \rangle$ properties

$\langle E \rangle \mathcal{A}$ $\quad \langle E \rangle \mathcal{G}$ $\quad \langle E \rangle (\mathcal{A} \to \mathcal{G})$ $\quad \langle E \rangle (\mathcal{A} \vee \mathcal{G})$ $\quad \langle E \rangle (\mathcal{A} \wedge \mathcal{G})$

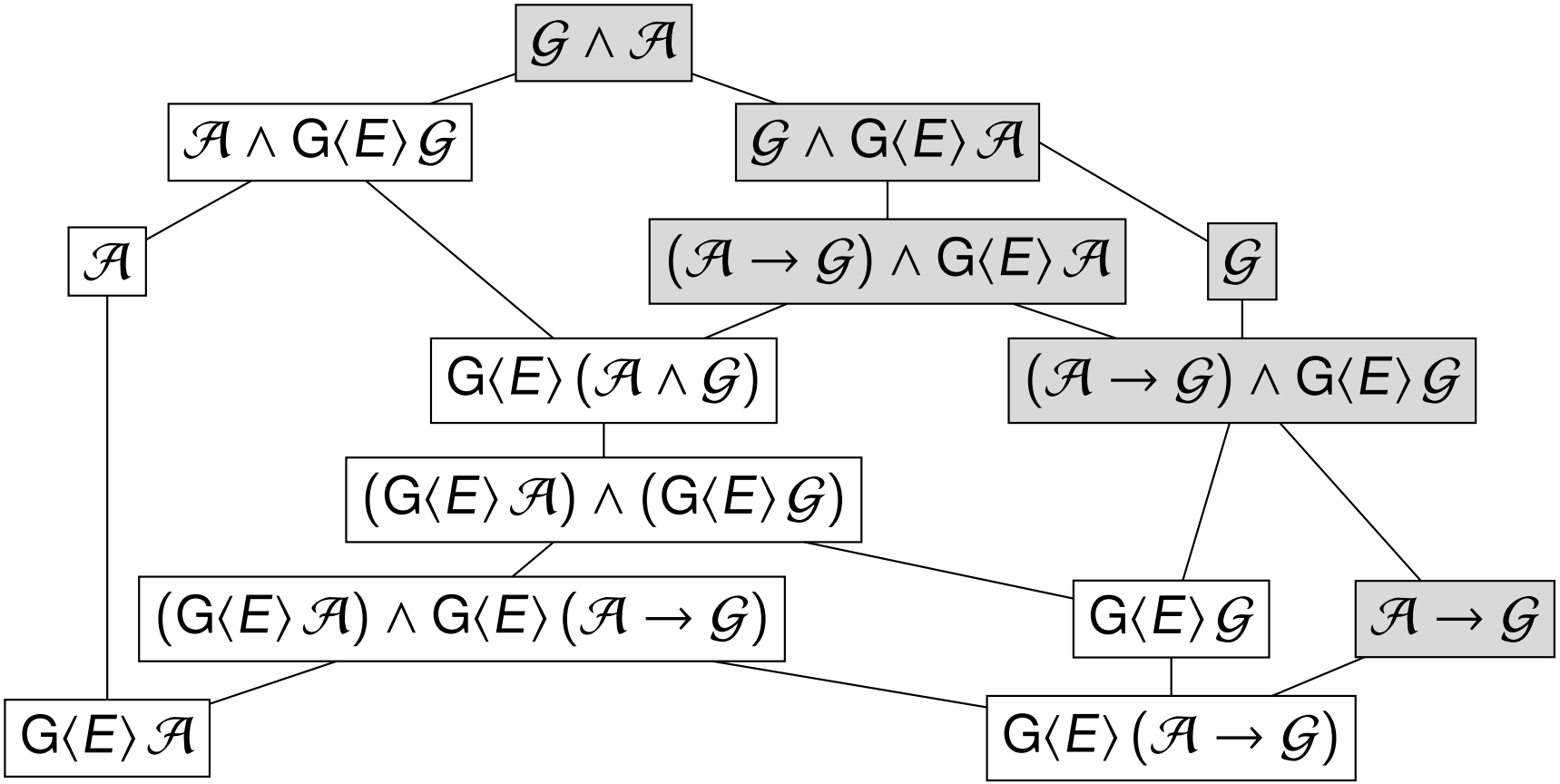
Cooperation level conjuncts



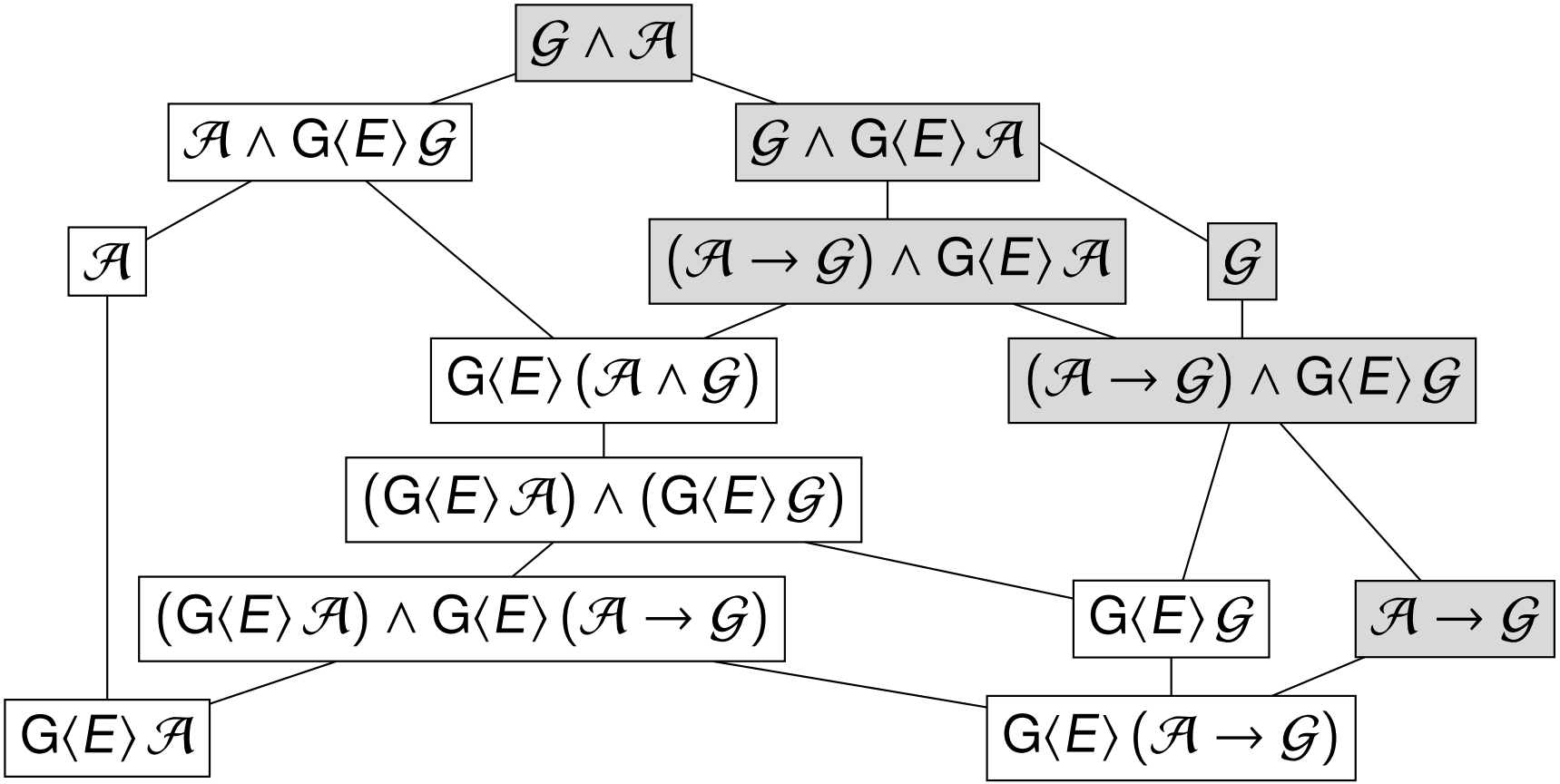
G⟨E⟩ properties

$$\mathsf{G}\langle E\rangle\mathcal{A} \quad \mathsf{G}\langle E\rangle\mathcal{G} \quad \mathsf{G}\langle E\rangle(\mathcal{A}\to\mathcal{G}) \quad \mathsf{G}\langle E\rangle(\mathcal{A}\vee\mathcal{G}) \quad \mathsf{G}\langle E\rangle(\mathcal{A}\wedge\mathcal{G})$$

Hierarchy of cooperation levels



Hierarchy of cooperation levels



Simplification rule example

$$(\mathcal{A} \to \mathcal{G}) \ \wedge \ \mathsf{G}\langle E\rangle \mathcal{A} \ \wedge \ \mathsf{G}\langle E\rangle \mathcal{G}$$

Synthesizing for a single cooperation level

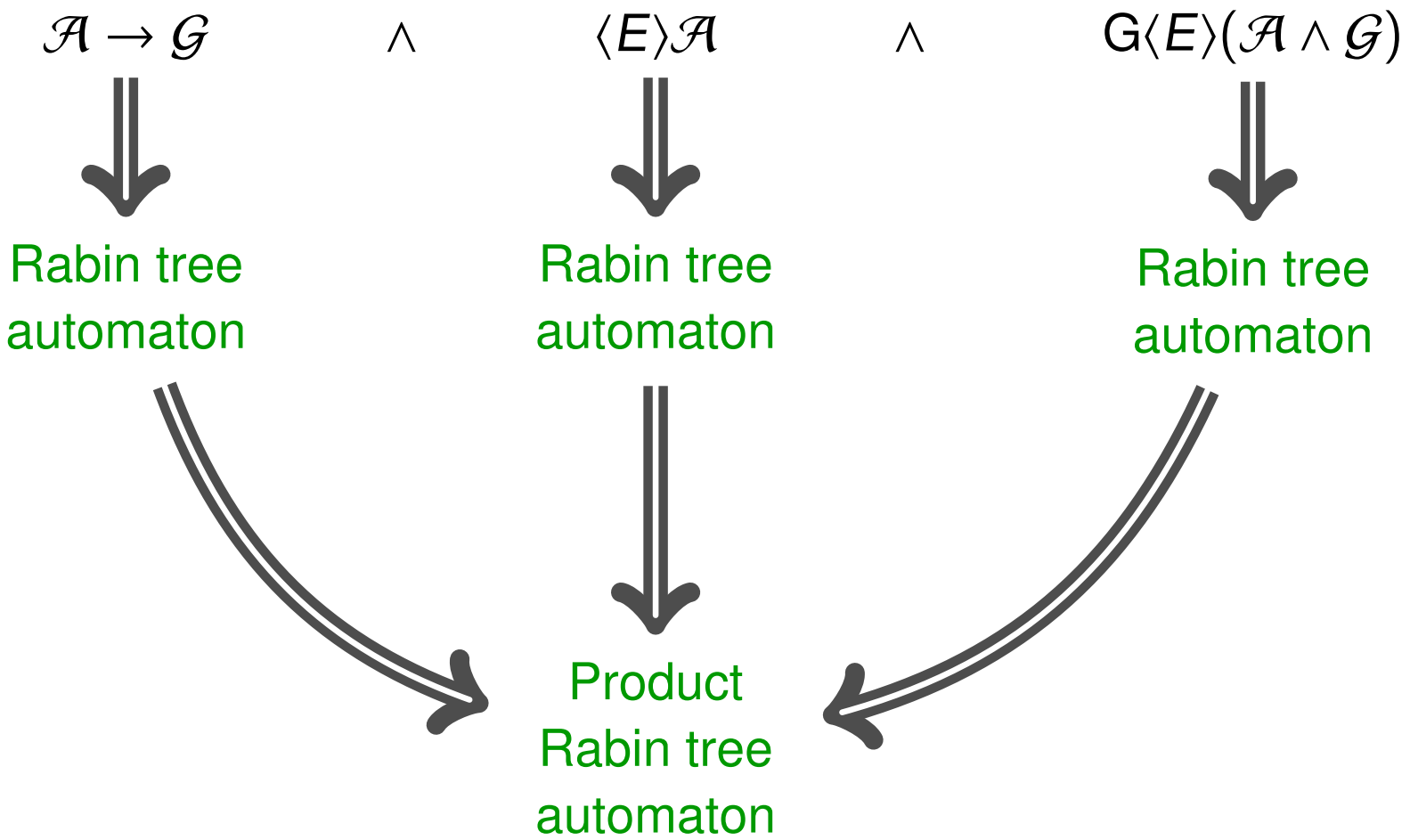
$\mathcal{A} \rightarrow \mathcal{G}$ $\wedge$ $\langle E \rangle \mathcal{A}$ $\wedge$ $\mathsf{G}\langle E \rangle(\mathcal{A} \wedge \mathcal{G})$

Rabin tree automaton

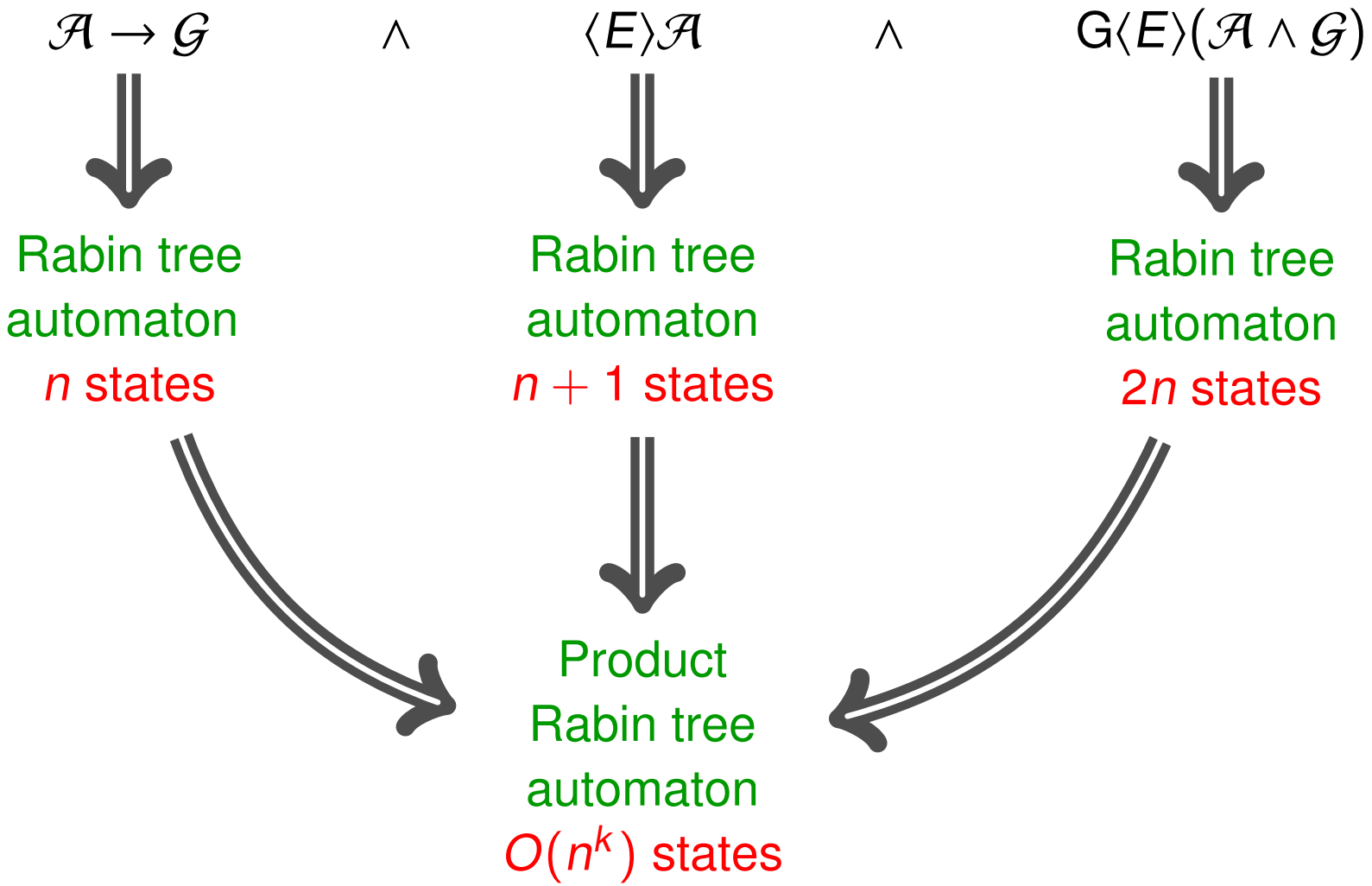
Rabin tree automaton

Rabin tree automaton

Synthesizing for a single cooperation level

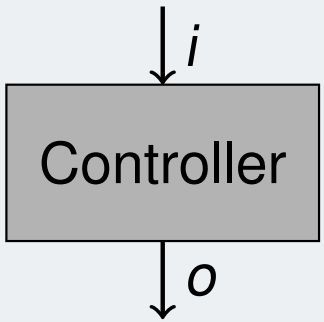


Synthesizing for a single cooperation level



Synthesis for multiple cooperation levels

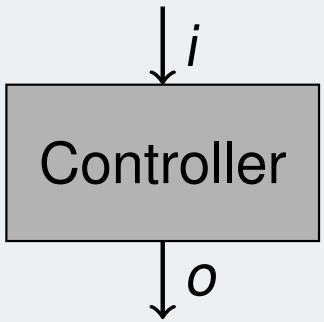
Example specification



$$\mathcal{A} = \mathsf{F}(i) \wedge \mathsf{GF}(o \vee \mathsf{X}i)$$
$$\mathcal{G} = \mathsf{G}(i \rightarrow o) \wedge (\neg o \,\mathcal{U}\, i)$$

Synthesis for multiple cooperation levels

Example specification



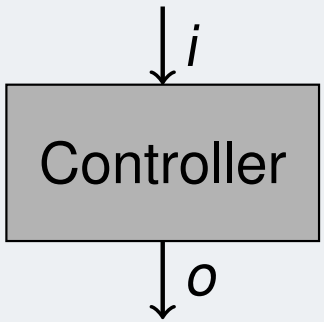
$$\mathcal{A} = \mathsf{F}(i) \wedge \mathsf{GF}(o \vee \mathsf{X}i)$$
$$\mathcal{G} = \mathsf{G}(i \rightarrow o) \wedge (\neg o \,\mathcal{U}\, i)$$

Example for switching between cooperation levels

$$(\mathcal{A} \rightarrow \mathcal{G}) \wedge \mathsf{G}\langle E \rangle \mathcal{A}$$

Synthesis for multiple cooperation levels

Example specification



$$\mathcal{A} = \mathsf{F}(i) \wedge \mathsf{GF}(o \vee \mathsf{X}i)$$
$$\mathcal{G} = \mathsf{G}(i \rightarrow o) \wedge (\neg o \,\mathcal{U}\, i)$$

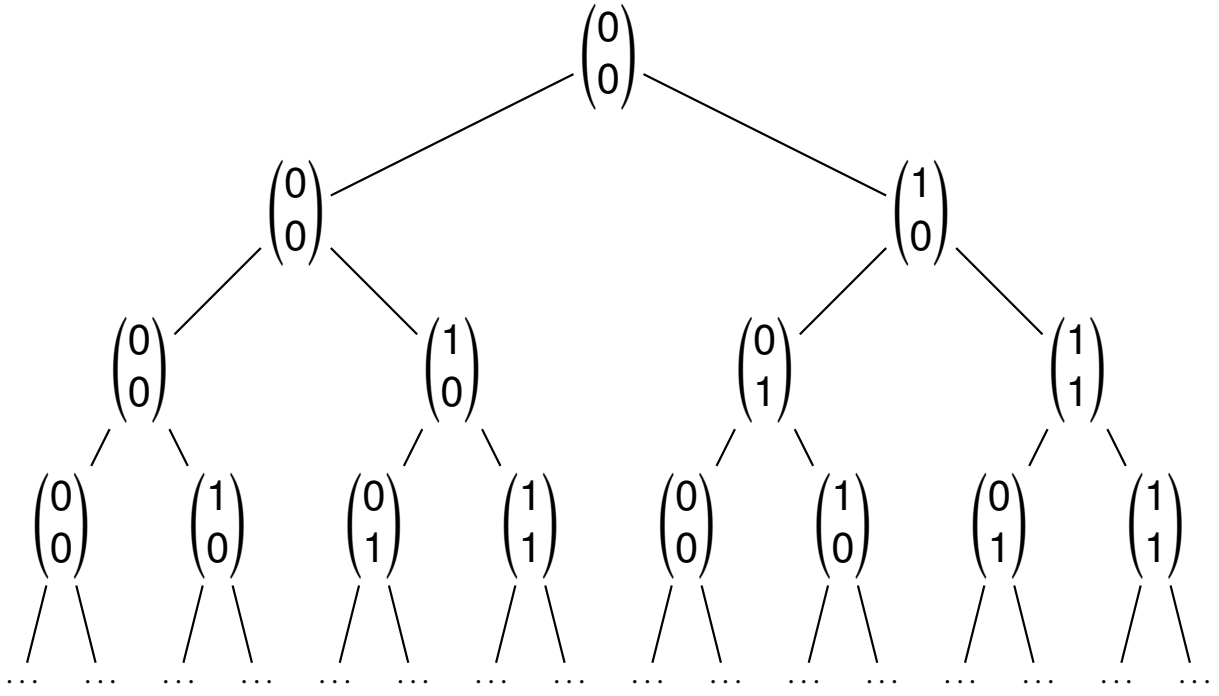
Example for switching between cooperation levels

$$(\mathcal{A} \rightarrow \mathcal{G}) \wedge \mathsf{G}\langle E\rangle\mathcal{A}$$

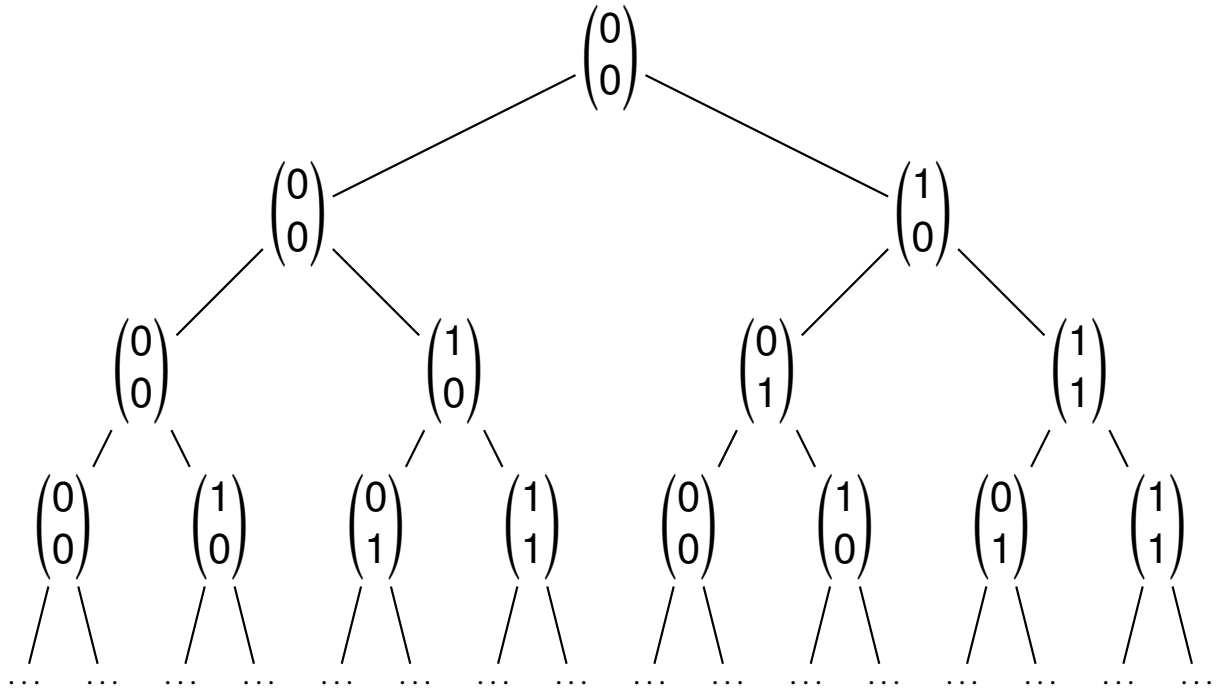
$$\Downarrow$$

$$\mathcal{A} \wedge \mathcal{G}$$

Semantics of mode switching



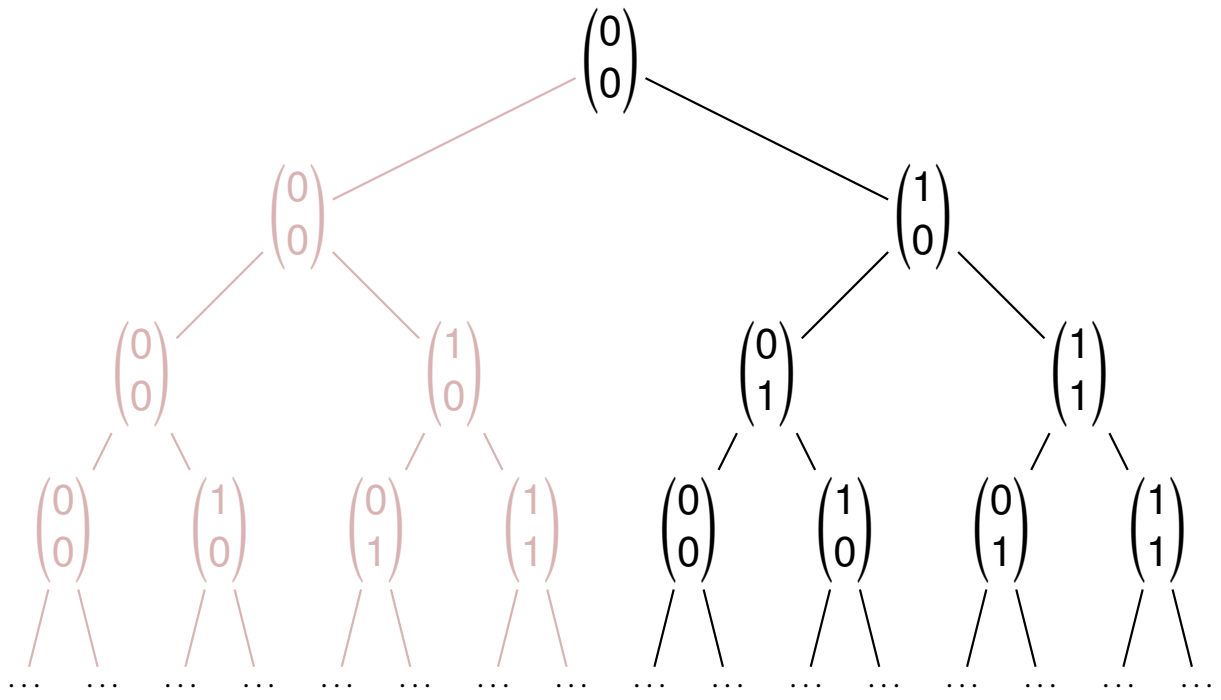
Semantics of mode switching



Example for switching between cooperation levels

$$(\mathcal{A} \rightarrow \mathcal{G}) \wedge \mathsf{G}\langle E \rangle \mathcal{A}$$

Semantics of mode switching



Example for switching between cooperation levels

$$(\mathcal{A} \to \mathcal{G}) \wedge \mathsf{G}\langle E\rangle\mathcal{A}$$

$$\Downarrow$$

$$\mathcal{A} \wedge \mathcal{G}$$

How does mode switching work? (1)

Simplifying assumption

- We only consider cooperation level conjuncts of the form $\mathrm{G}\langle E\rangle\psi$ and ψ.

How does mode switching work? (1)

Simplifying assumption

- We only consider cooperation level conjuncts of the form $\mathrm{G}\langle E\rangle\psi$ and ψ.

Properties of the single-level Rabin automata product

$$q = (\ q_1,\ b_1,\ q_2,\ q_3,\ \cdots,\ b_n,\ q_n\ )$$

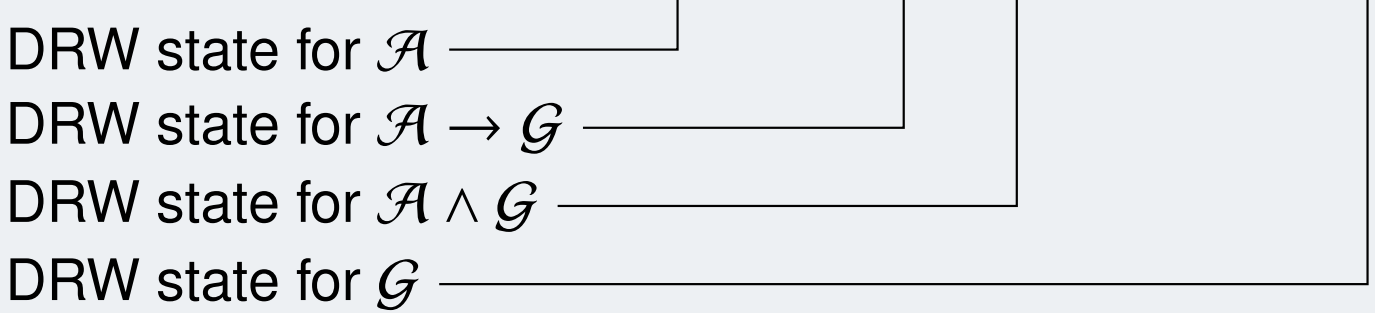
How does mode switching work? (1)

Simplifying assumption

- We only consider cooperation level conjuncts of the form $\mathrm{G}\langle E\rangle\psi$ and ψ.

Properties of the single-level Rabin automata product

$$q = (\ q_1,\ b_1,\ q_2,\ q_3,\ \cdots,\ b_n,\ q_n\ )$$



Notation: Let $\mathrm{unpack}(q) = \{q_1, q_2, q_3, \ldots, q_n\}$

How does mode switching work? (2)

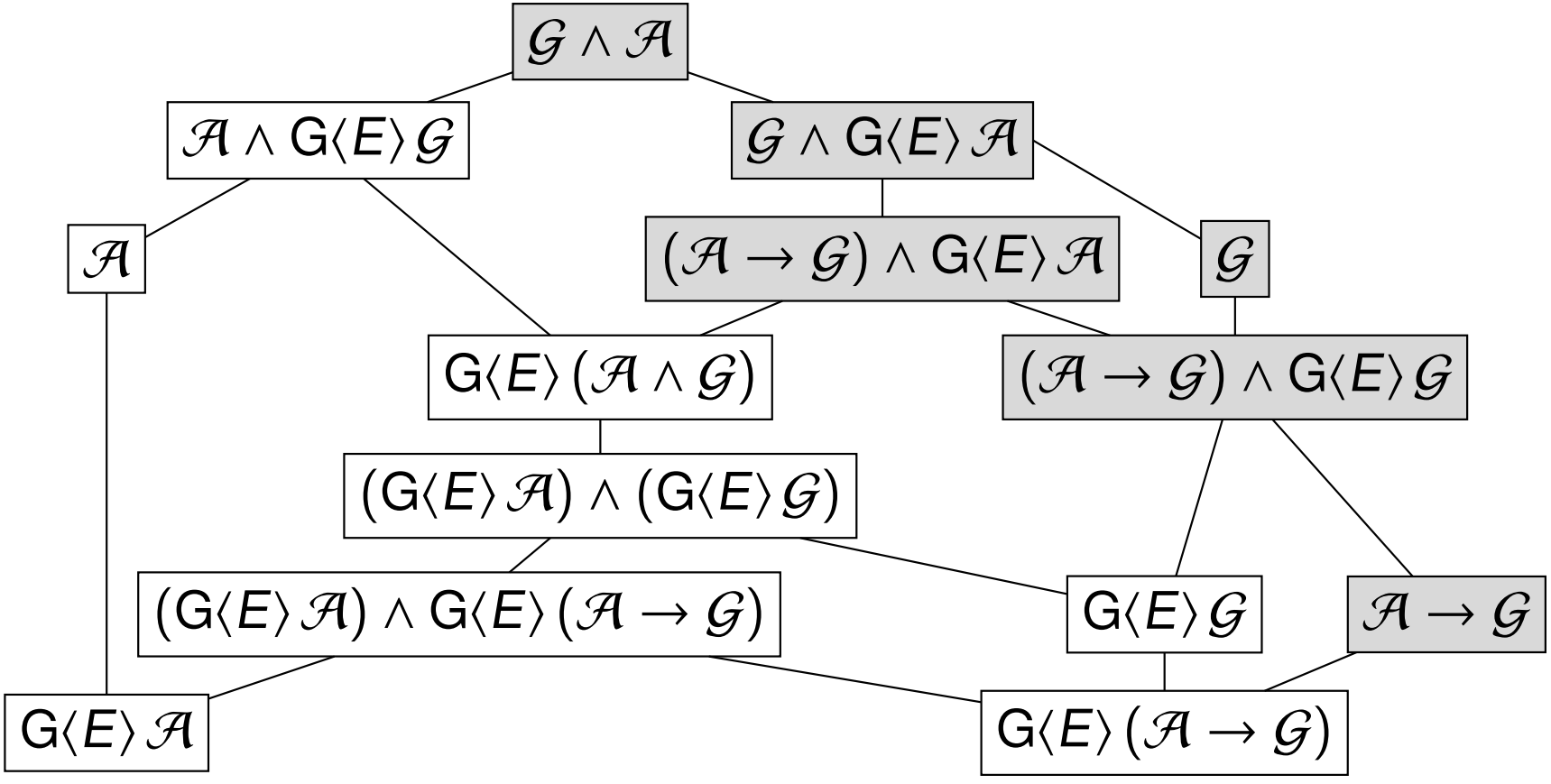
Properties of the single-level Rabin automaton product

- Any two Rabin tree automaton states q, q' with $\text{unpack}(q) = \text{unpack}(q')$ have the same language.
- The elements of $\text{unpack}(q)$ follow the state of the word automata on the trace seen so far.

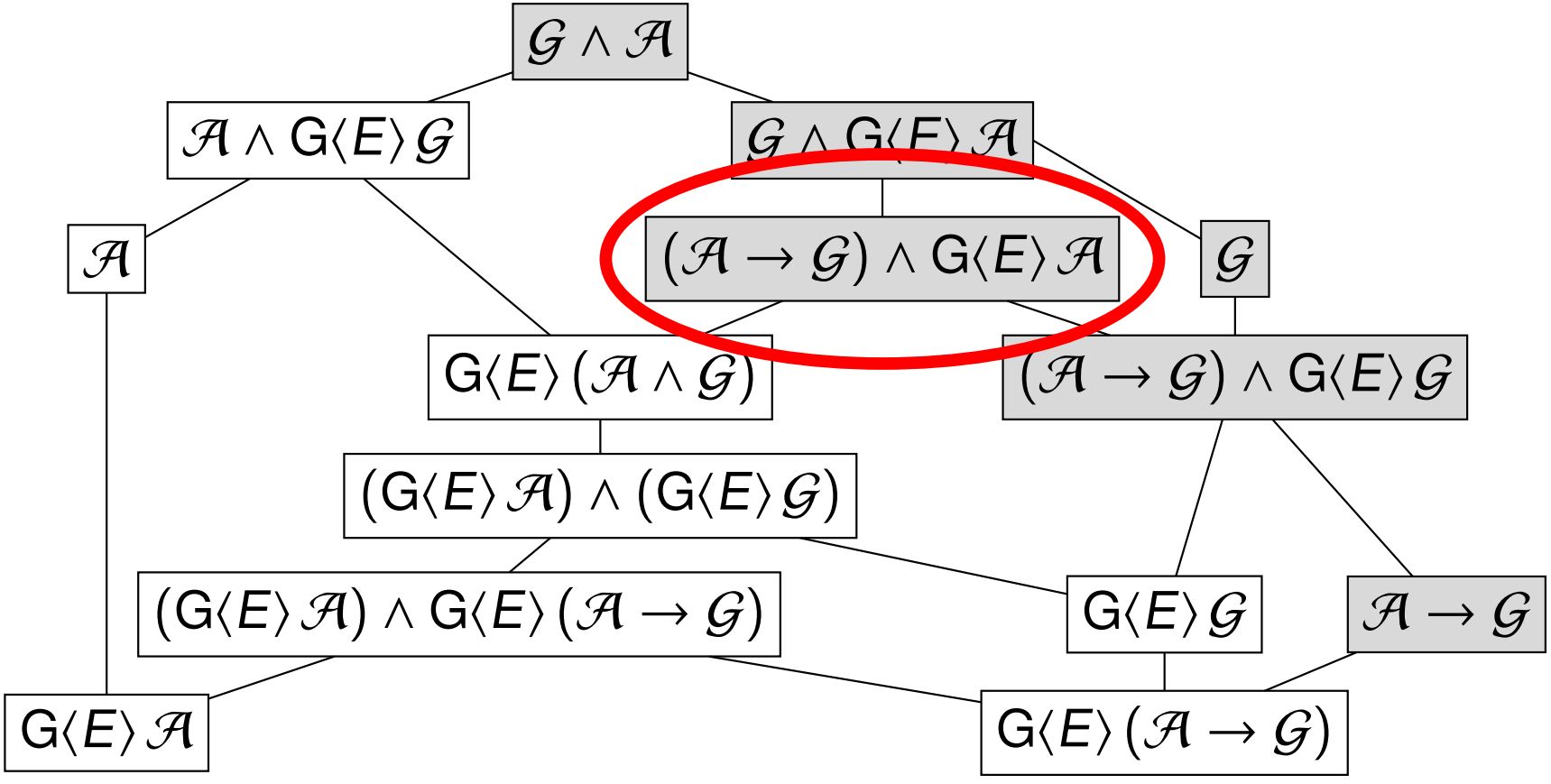
Switching to a higher cooperation level

Taking two Rabin tree automata $\mathcal{A}_1$ and $\mathcal{A}_2$ for two cooperation levels $l_1 < l_2$, we can reroute a transition to a state $q \in \mathcal{A}_1$ to a state $q' \in \mathcal{A}_2$ if $\text{unpack}(q) = \text{unpack}(q')$ and $L(q') \neq \emptyset$.

Cooperative GR(1) Synthesis



Cooperative GR(1) Synthesis



GR(1) Specifications

Components

- Initialization Assumptions
- Initialization Guarantees
- Basic Safety Assumptions
- Basic Safety Guarantees
- A liveness specification of the form:

$$\big(\mathsf{GF}(\ldots) \wedge \mathsf{GF}(\ldots) \wedge \ldots\big) \rightarrow \big(\mathsf{GF}(\ldots) \wedge \mathsf{GF}(\ldots) \wedge \ldots\big)$$

Standard GR(1) Fixpoint

$$\nu Z.\ \bigwedge_{j\in\{1,\ldots,n\}}\ \mu Y.\ \bigvee_{i\in\{1,\ldots,m\}}\ \nu X.\ \mathsf{EnfPre}\Big(\ (Z' \wedge \psi_j^g)\ \vee\ Y'\ \vee\ (\neg\psi_i^a \wedge X')\ \Big)$$

Standard GR(1) Fixpoint

$$\nu Z.\ \bigwedge_{j\in\{1,\ldots,n\}}\ \mu Y.\ \bigvee_{i\in\{1,\ldots,m\}}\ \nu X.\ \mathsf{EnfPre}\Big(\ (Z' \wedge \psi_j^g)\ \vee\ Y'\ \vee\ (\neg\psi_i^a \wedge X')\ \Big)$$

Reaching the next system goal

Standard GR(1) Fixpoint

$$\nu Z.\ \bigwedge_{j\in\{1,\ldots,n\}}\ \mu Y.\ \bigvee_{i\in\{1,\ldots,m\}}\ \nu X.\ \mathsf{EnfPre}\Big(\ (Z' \wedge \psi_j^g)\ \vee\ Y'\ \vee\ (\neg\psi_i^a \wedge X')\ \Big)$$

Reaching the next system goal

Getting closer to the next system goal

Standard GR(1) Fixpoint

$$\nu Z.\ \bigwedge_{j\in\{1,\ldots,n\}}\ \mu Y.\ \bigvee_{i\in\{1,\ldots,m\}}\ \nu X.\ \mathrm{EnfPre}\Big(\big(Z' \wedge \psi_j^g\big) \vee Y' \vee \big(\neg\psi_i^a \wedge X'\big)\Big)$$

Reaching the next system goal

Getting closer to the next system goal

Waiting for some environment goal

Standard GR(1) Fixpoint

$$\nu Z.\ \bigwedge_{j\in\{1,\ldots,n\}}\ \mu Y.\ \bigvee_{i\in\{1,\ldots,m\}}\ \nu X.\ \mathrm{EnfPre}\Big((Z' \wedge \psi_j^g) \vee Y' \vee (\neg\psi_i^a \wedge X')\Big)$$

Reaching the next system goal

Getting closer to the next system goal

Waiting for some environment goal

Give semantics to X, Y, and Z

Cooperative GR(1) Synthesis

$$\nu Z. \bigwedge_{j\in\{1,\ldots,n\}} \mu Y. \bigvee_{i\in\{1,\ldots,m\}} \nu X.\mathsf{EnfPre}\Big((Z' \wedge \psi_j^g) \vee Y' \vee (\neg\psi_i^a \wedge X')\Big)$$

$$\wedge\ \mu R.\mathsf{Reach}\Big((\psi_j^g \vee Y' \vee R') \wedge X\Big)$$

$$\wedge \bigwedge_{k\in\{1,\ldots,m\}} \mu R.\mathsf{Reach}\Big((\psi_k^a \vee R') \wedge Z\Big)$$

Cooperative GR(1) Synthesis

$$\nu Z.\ \bigwedge_{j\in\{1,\ldots,n\}} \mu Y.\ \bigvee_{i\in\{1,\ldots,m\}} \nu X.\mathrm{EnfPre}\Big((Z' \wedge \psi_j^g) \vee Y' \vee (\neg\psi_i^a \wedge X')\Big)$$

$$\wedge\ \mu R.\mathrm{Reach}\Big((\psi_j^g \vee Y' \vee R') \wedge X\Big)$$

$$\wedge \bigwedge_{k\in\{1,\ldots,m\}} \mu R.\mathrm{Reach}\Big((\psi_k^a \vee R') \wedge Z\Big)$$

System must be able to get closer to the goal in every waiting period

Cooperative GR(1) Synthesis

$$\nu Z. \bigwedge_{j\in\{1,\ldots,n\}} \mu Y. \bigvee_{i\in\{1,\ldots,m\}} \nu X.\mathrm{EnfPre}\Big((Z' \wedge \psi_j^g) \vee Y' \vee (\neg\psi_i^a \wedge X')\Big)$$

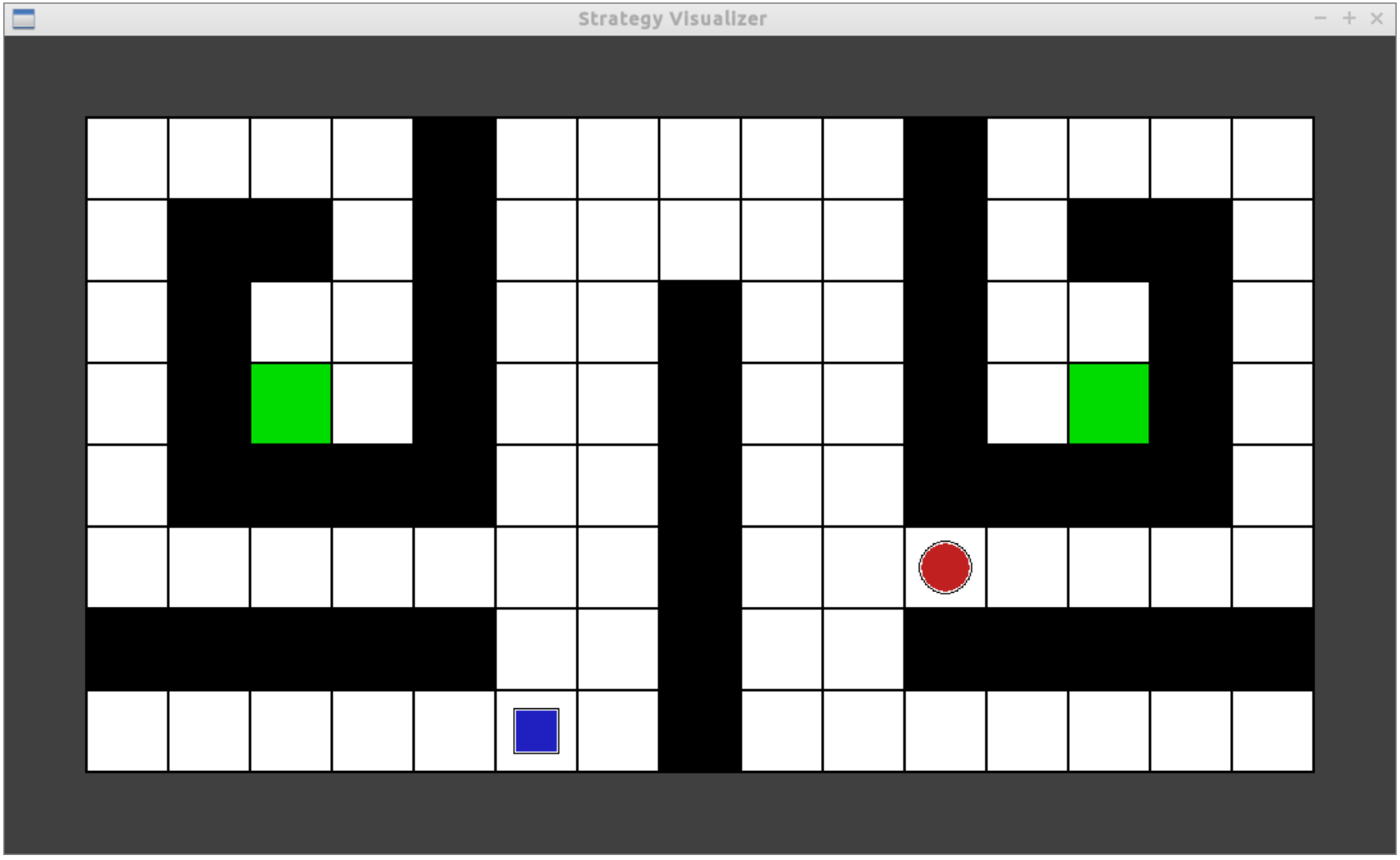
$$\wedge\ \mu R.\mathrm{Reach}\Big((\psi_j^g \vee Y' \vee R') \wedge X\Big)$$

$$\wedge \bigwedge_{k\in\{1,\ldots,m\}} \mu R.\mathrm{Reach}\Big((\psi_k^a \vee R') \wedge Z\Big)$$

System must be able to get closer to the goal in every waiting period

Every environment goal can be reached

Demo



Related work

Strategy logic (Chatterjee et al., 2010)

- can reason about the strategic capabilities of players
- does not allow to *rewind* the computation → cannot express $\mathrm{G}\langle E\rangle\psi$ in our semantics

CTL* with linear past (Bozzelli, 2008)

- allows to encode synthesis for a single cooperation level
- more complicated than our approach
- translations to automata not direct enough to allow switching between cooperation levels

Conclusion

Cooperative synthesis

- **improves the quality of synthesized solutions** without the need to state additional specification parts
- helps to bring **reactive synthesis into practice**
- analyzes the limits of a system's interaction with its environment by pointing out the feasible level(s) of our **fine-grained cooperation hierarchy**

References I

Laura Bozzelli. The complexity of CTL* + linear past. In *FOSSACS*, pages 186–200, 2008.

Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. *Inf. Comput.*, 208(6):677–693, 2010.