Forschungslinie: Mo, 8. April 2019

13:00 - 14:30 Maschinelle Verarbeitung natürlicher Sprachen
Prof. Heiko Vogler

14:50 - 16:20 Model Checking
Prof. Christel Baier / Dr. Sascha Klüppelholz

Model Checking

Technische Universität Dresden

Institut für Theoretische Informatik

Algebraische und Logische Grundlagen der Informatik

Christel Baier Philipp Chrszon Clemens Dubslaff
Florian Funke Simon Jantsch Sascha Klüppelholz
David Müller Jakob Piribauer Sascha Wunderlich

Verifikation

Verifikation:

Nachweis, dass ein System seine Anforderungen erfüllt

Verifikation

Verifikation:
Nachweis, dass ein System seine Anforderungen erfüllt

Anforderungen:

- Terminierung, partielle Korrektheit für sequentielle Programme
- temporale Eigenschaften der Interaktionen in parallelen Systemen
- quantitative Anforderungen an das Echtzeitverhalten

⋮

Warum Verifikation ?

Warum Verifikation ?

Therac-25 Radiation, 1985-87

SW-Fehler in Bestrahlungsmaschine

Verzögerte Eröffnung des Flughafen Denver, 1989

SW-Fehler im Gepäckabfertigungssystem

Ausfall des AT&T Telefonnetzwerks, 1990 in New York

falscher Schleifenabbruch in C-Programm

Intel Pentium-Prozessor, 1994

fehlerhafte Gleitkommadivision

Crash der Ariane 5 Rakete, 1996

SW-Fehler in Kontrollsoftware

Warum Verifikation ?

Therac-25 Radiation, 1985-87

SW-Fehler in Bestrahlungsmaschine
mindestens 3 Tote aufgrund 100-facher Bestrahlung

Verzögerte Eröffnung des Flughafen Denver, 1989

SW-Fehler im Gepäckabfertigungssystem

Ausfall des AT&T Telefonnetzwerks, 1990 in New York

falscher Schleifenabbruch in C-Programm

Intel Pentium-Prozessor, 1994

fehlerhafte Gleitkommadivision

Crash der Ariane 5 Rakete, 1996

SW-Fehler in Kontrollsoftware

Warum Verifikation ?

Therac-25 Radiation, 1985-87

SW-Fehler in Bestrahlungsmaschine
mindestens 3 Tote aufgrund 100-facher Bestrahlung

Verzögerte Eröffnung des Flughafen Denver, 1989

SW-Fehler im Gepäckabfertigungssystem ca. 500 Mio USD

Ausfall des AT&T Telefonnetzwerks, 1990 in New York

falscher Schleifenabbruch in C-Programm ca. 100 Mio USD

Intel Pentium-Prozessor, 1994

fehlerhafte Gleitkommadivision ca. 500 Mio USD

Crash der Ariane 5 Rakete, 1996

SW-Fehler in Kontrollsoftware ca. 370 Mio USD

Analyse- und Verifikationsmethoden

Analyse- und Verifikationsmethoden

- Peer reviewing: manuelle Code-Inspektion
- Testen, Simulation:

  Ausführung des Systems bzw. Modells in ausgewählten Szenarien

Analyse- und Verifikationsmethoden

- Peer reviewing: manuelle Code-Inspektion
- Testen, Simulation:

  Ausführung des Systems bzw. Modells in ausgewählten Szenarien
- formale Verifikation:

  mathematischer Nachweis ausgewählter Systemeigenschaften

Analyse- und Verifikationsmethoden

- Peer reviewing: manuelle Code-Inspektion
- Testen, Simulation:

  Ausführung des Systems bzw. Modells in ausgewählten Szenarien
- formale Verifikation:

  mathematischer Nachweis ausgewählter Systemeigenschaften

  * erfordert mathematische Modelle für das System und die Eigenschaften

Analyse- und Verifikationsmethoden

- Peer reviewing: manuelle Code-Inspektion
- Testen, Simulation:

  Ausführung des Systems bzw. Modells in ausgewählten Szenarien
- formale Verifikation:

  mathematischer Nachweis ausgewählter Systemeigenschaften

  * erfordert mathematische Modelle für das System und die Eigenschaften
  * interaktiv mit Hilfe eines Theorembeweisers
  * algorithmisch mit Hilfe eines Model Checkers

Model Checking

Model Checking:

algorithmische Verifikationsmethode
für parallele Systeme

- Automatenmodell für das System

  kompositionelle Modellkonstruktion mit Operatoren für Kommunikation, Synchronisation, etc.

- logische Formeln für die Anforderungen

  temporale und modale Logiken

ACM Turing Award 2007

Edmund M. Clarke	CMU, Pittsburgh
Allen E. Emerson	Texas at Austin University
Joseph Sifakis	IMAG Grenoble, Frankreich

Jury Entscheidung:

> ... *for their roles in developing model checking into a highly effective verification technology, widely adopted in the hardware and software industries ...*

Weitere Prominenz

weitere Turing Award Gewinner aus dem Gebiet "Formale Methoden":

Edsger Dijkstra	1972	Programmiersprachen
Donald Knuth	1974	Analyse von Algorithmen
Michael Rabin Dana Scott	1976	Automatentheorie
Tony Hoare	1980	Programmverifikation
Stephen Cook	1982	Komplexitätstheorie
Robin Milner	1991	Theorie paralleler Systeme
Amir Pnueli	1996	temporale Logik

Beispiel: paralleles Programm

paralleles Programm $\mathit{Inc} \parallel \mathit{Dec} \parallel \mathit{Reset}$

mit gemeinsamer Integer-Variable x

Beispiel: paralleles Programm

```
WHILE true DO
      IF x < 10 THEN x := x+1 FI
OD
```

Prozess *Inc*

```
WHILE true DO
      IF x > 0  THEN x := x−1 FI
OD
```

Prozess *Dec*

```
WHILE true DO
      IF x = 10 THEN x := 0 FI
OD
```

Prozess *Reset*

initial: $x = 1$

Beispiel: paralleles Programm

```
WHILE true DO
      IF x < 10 THEN x := x+1 FI
OD
```

Prozess *Inc*

```
WHILE true DO
      IF x > 0  THEN x := x−1 FI
OD
```

Prozess *Dec*

```
WHILE true DO
      IF x = 10 THEN x := 0 FI
OD
```

Prozess *Reset*

Gilt für $\mathit{Inc} \parallel \mathit{Dec} \parallel \mathit{Reset}$ immer $0 \leqslant x \leqslant 10$?

initial: $x = 1$

Beispiel: paralleles Programm

```
WHILE true DO
      IF x < 10 THEN x := x+1 FI
OD
```

Prozess *Inc*

```
WHILE true DO
      IF x > 0  THEN x := x−1 FI
OD
```

Prozess *Dec*

```
WHILE true DO
      IF x = 10 THEN x := 0 FI
OD
```

Prozess *Reset*

Gilt für $\mathit{Inc} \parallel \mathit{Dec} \parallel \mathit{Reset}$ immer $0 \leqslant x \leqslant 10$?

Antwort: nein initial: $x = 1$

Beispiel: paralleles Programm

initial:	$x = 1$	
Inc	$x < 10$?	ja
Inc	$x = 2$	
Inc	$x < 10$?	ja
Inc	$x = 3$	
$\vdots$	$\vdots$	
Inc	$x = 10$	

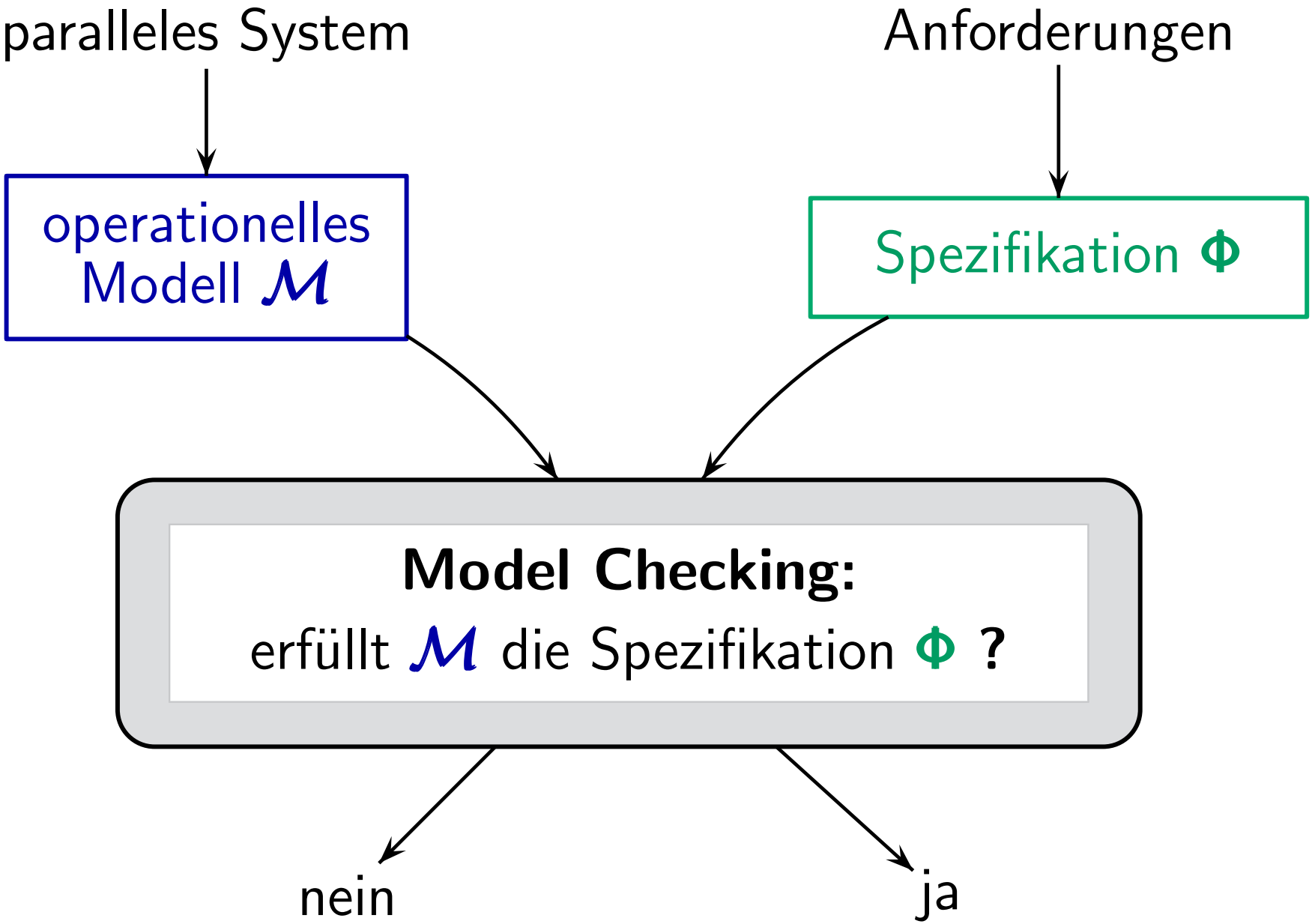
Beispiel: paralleles Programm

initial:	$x = 1$	
Inc	$x < 10$?	ja
Inc	$x = 2$	
Inc	$x < 10$?	ja
Inc	$x = 3$	
$\vdots$	$\vdots$	
Inc	$x = 10$	
Dec	$x > 0$?	ja
Reset	$x = 10$?	ja

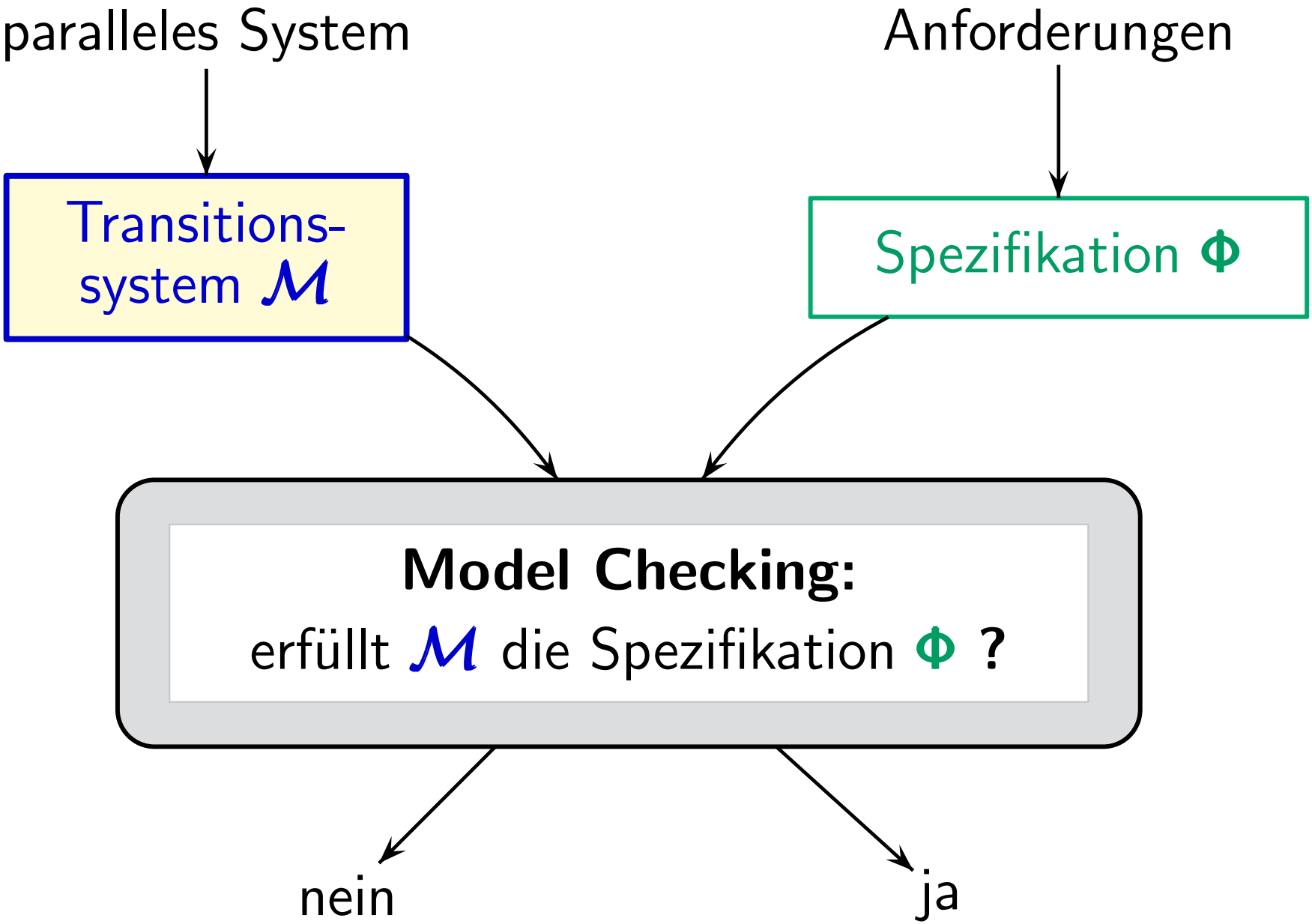
Beispiel: paralleles Programm

initial:	$x = 1$	
Inc	$x < 10$?	ja
Inc	$x = 2$	
Inc	$x < 10$?	ja
Inc	$x = 3$	
$\vdots$	$\vdots$	
Inc	$x = 10$	
Dec	$x > 0$?	ja
Reset	$x = 10$?	ja
Reset	$x = 0$	
Dec	$x = -1$	

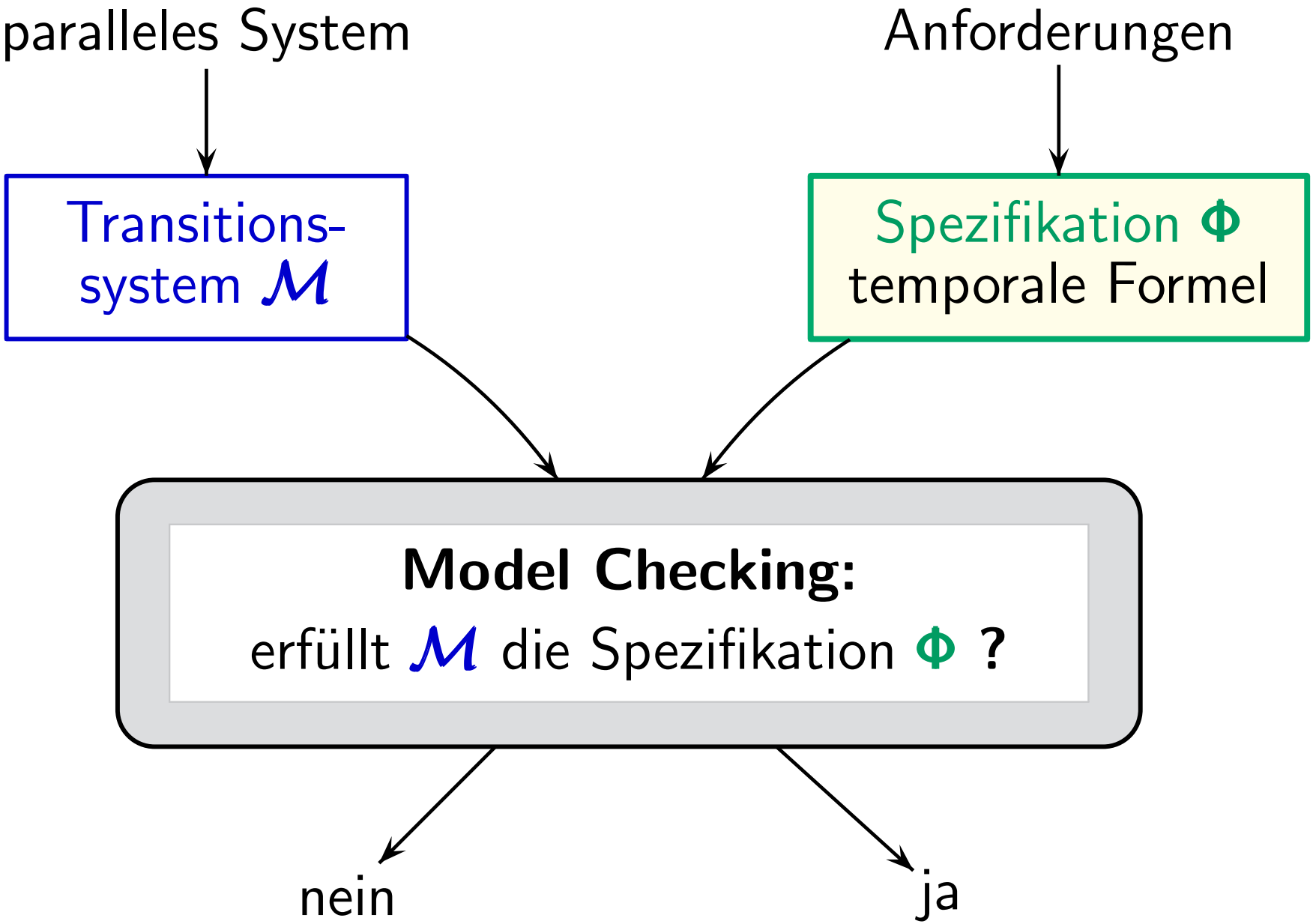
Model Checking



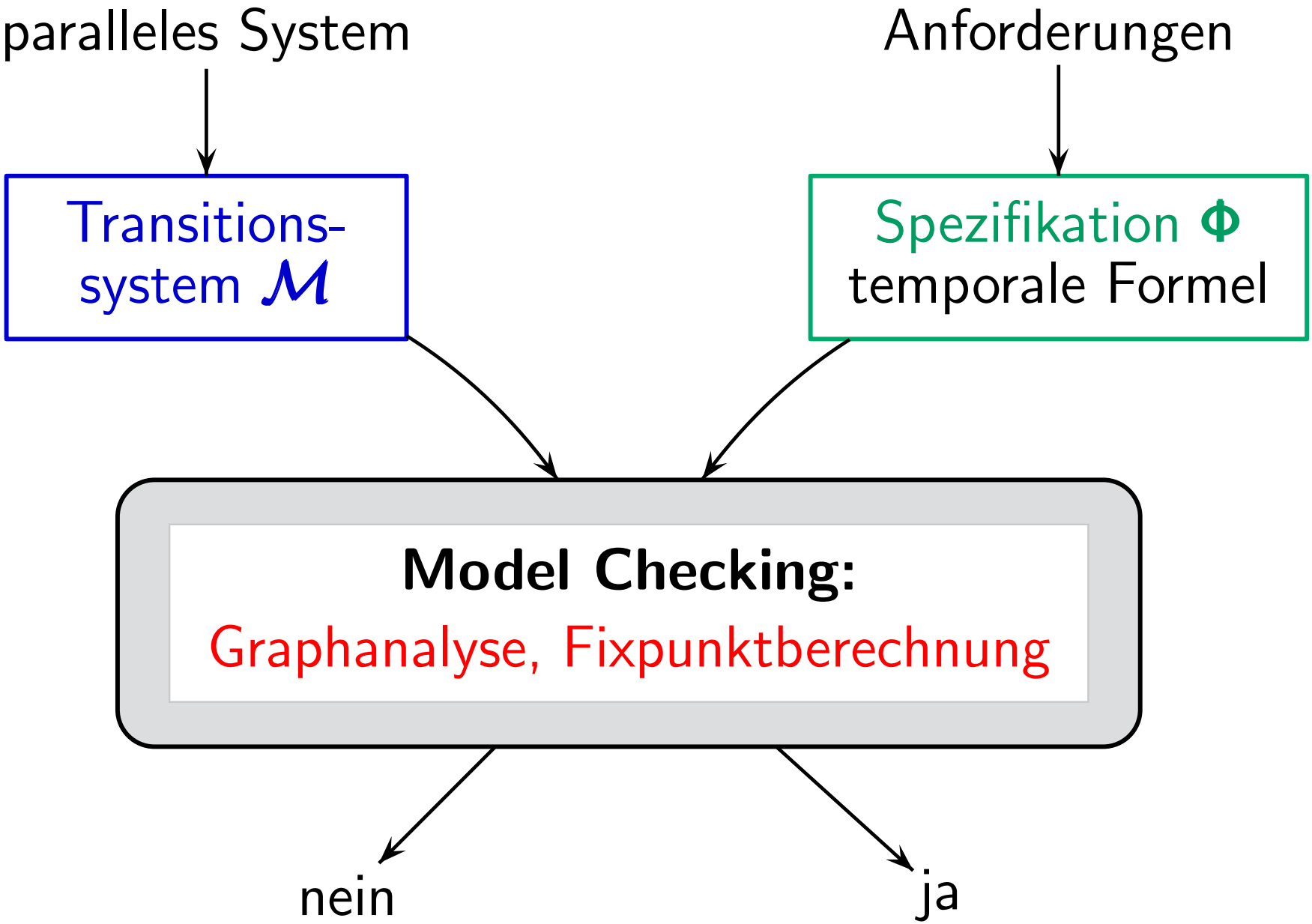
Model Checking



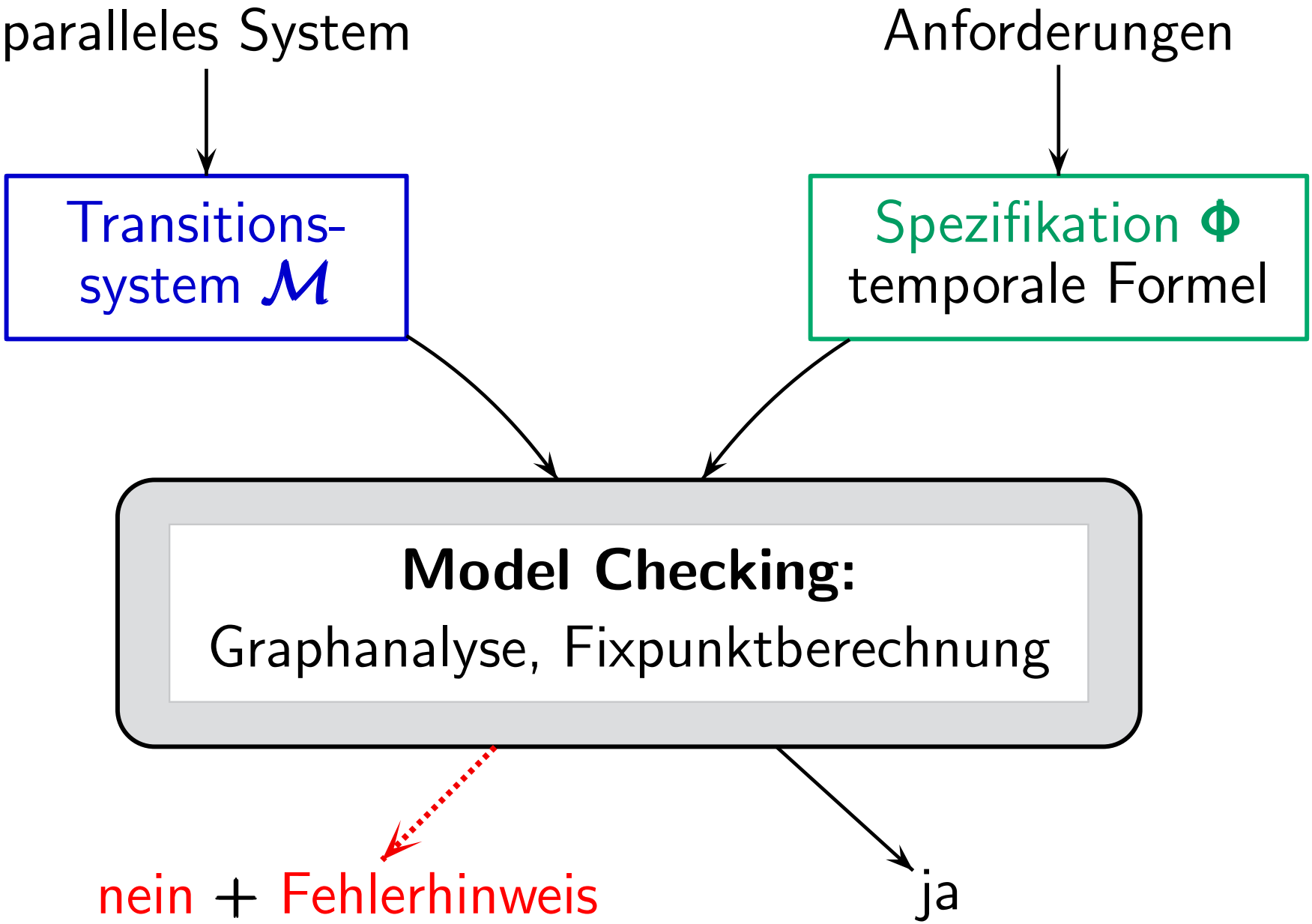
Model Checking



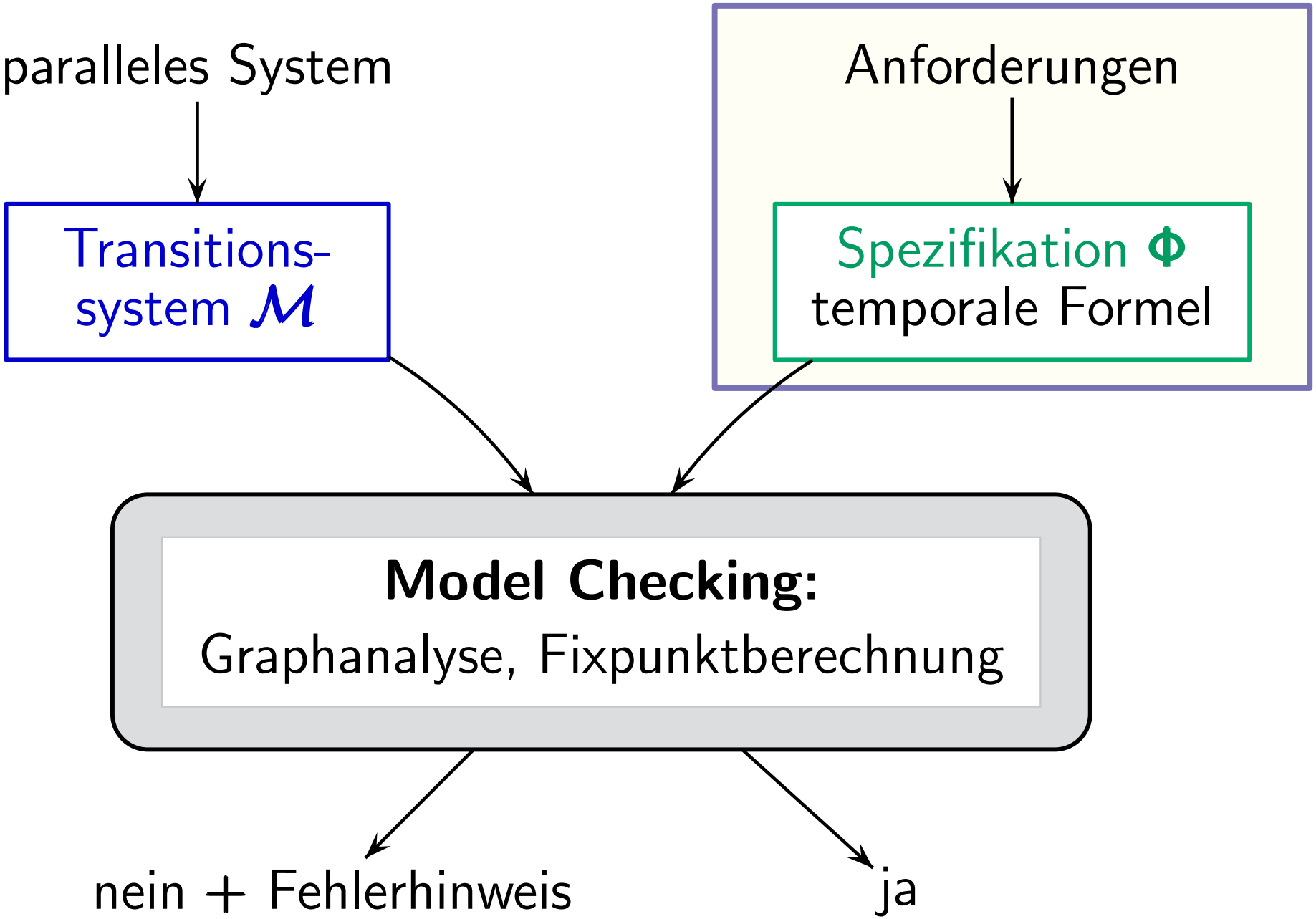
Model Checking



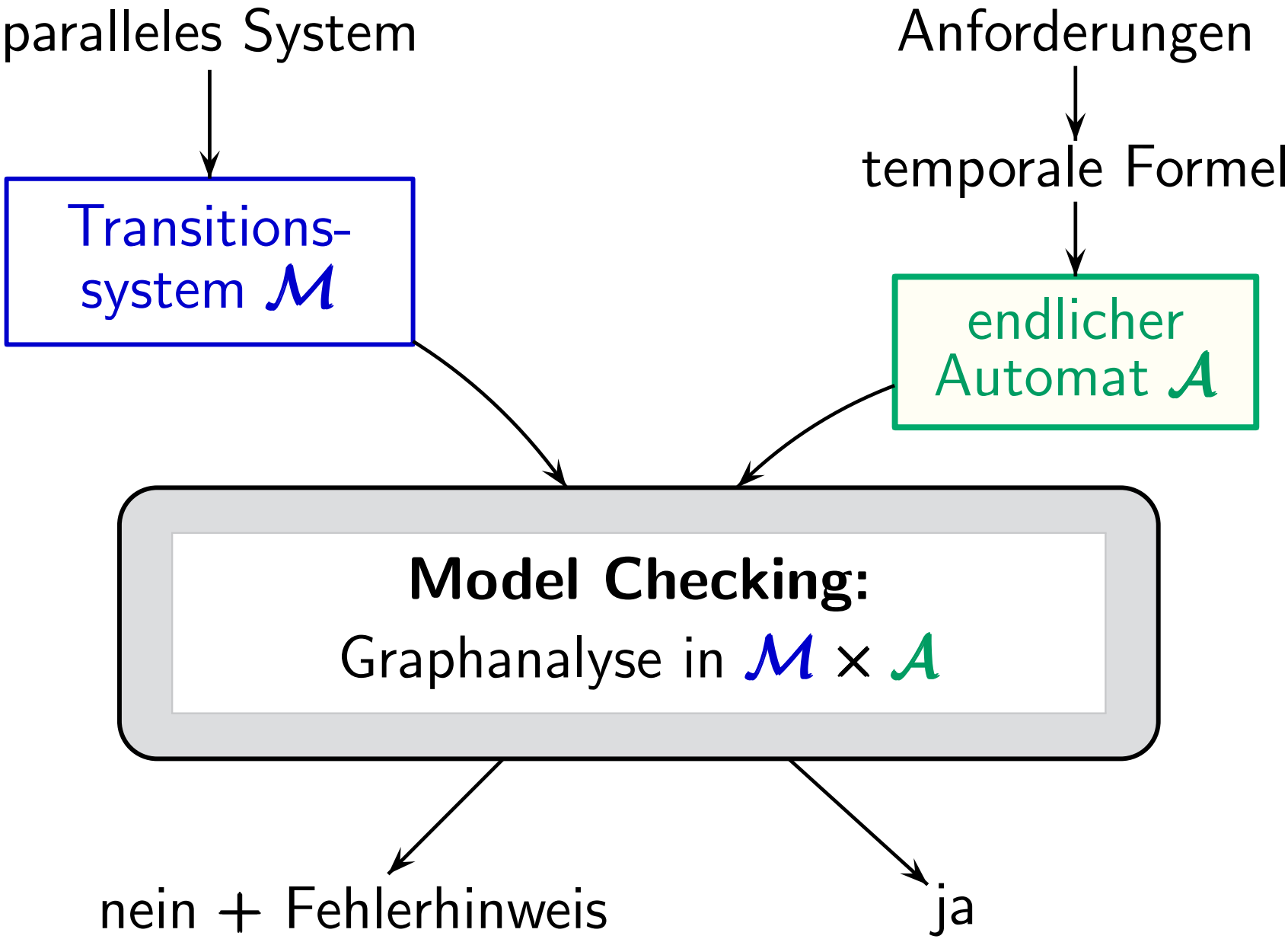
Model Checking



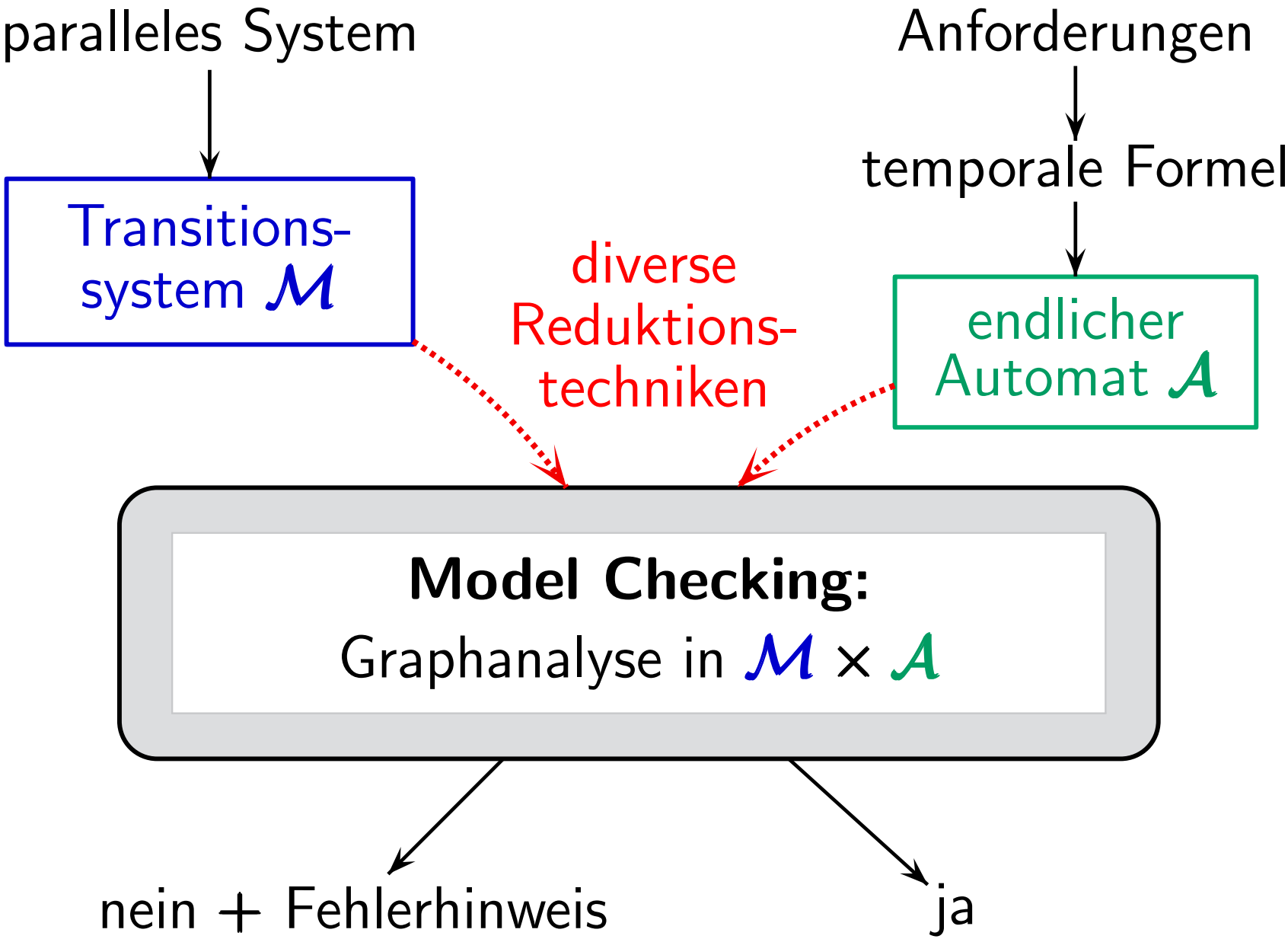
Model Checking



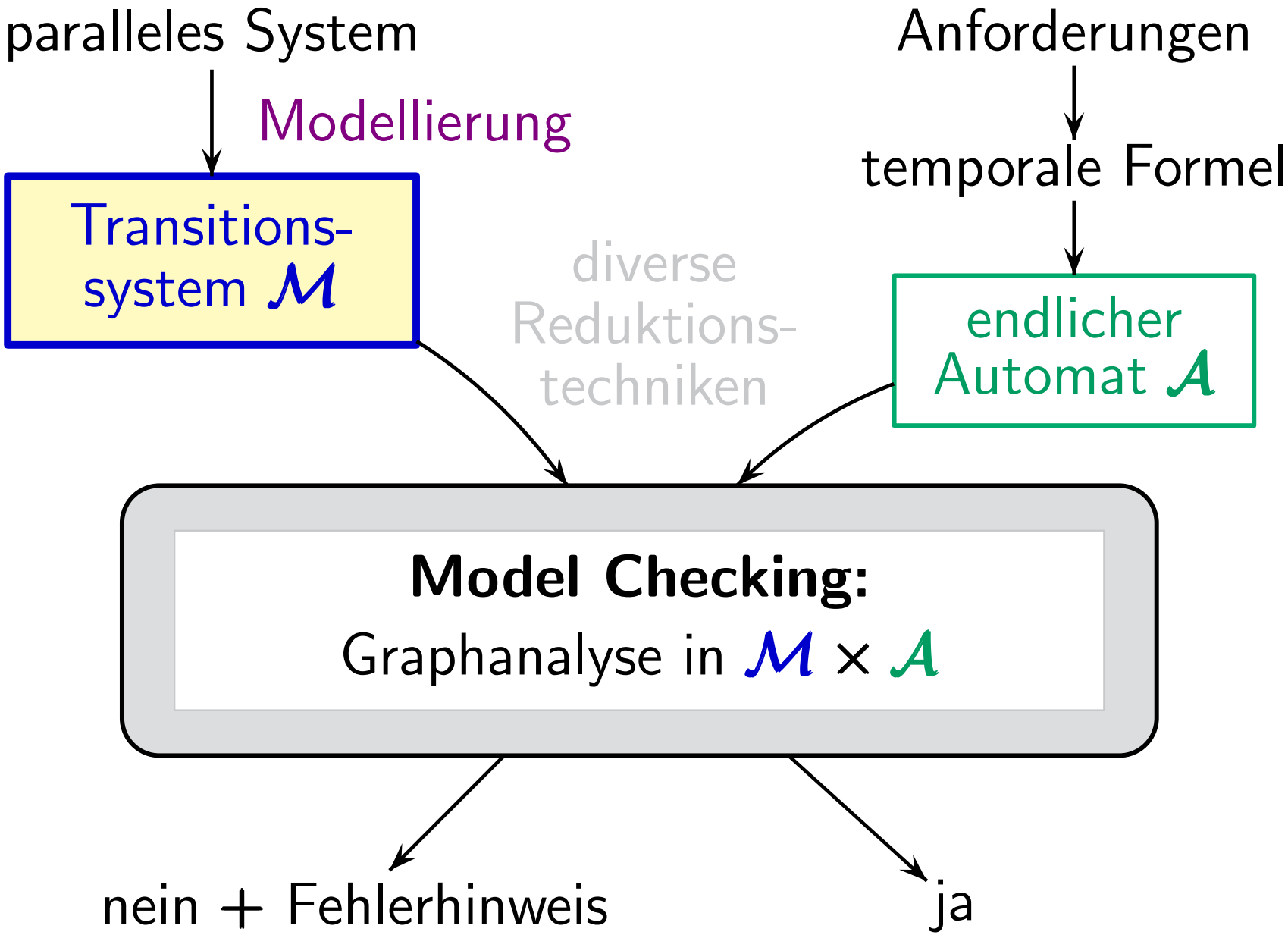
Model Checking



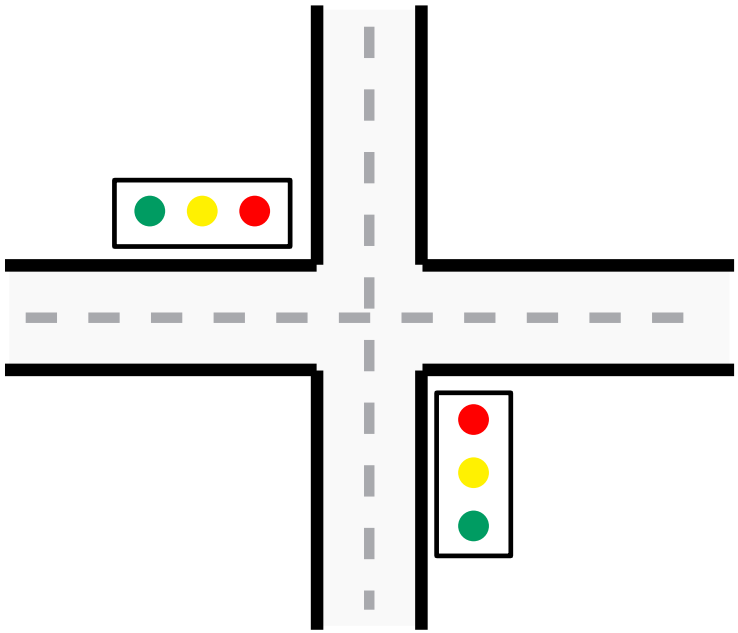
Model Checking



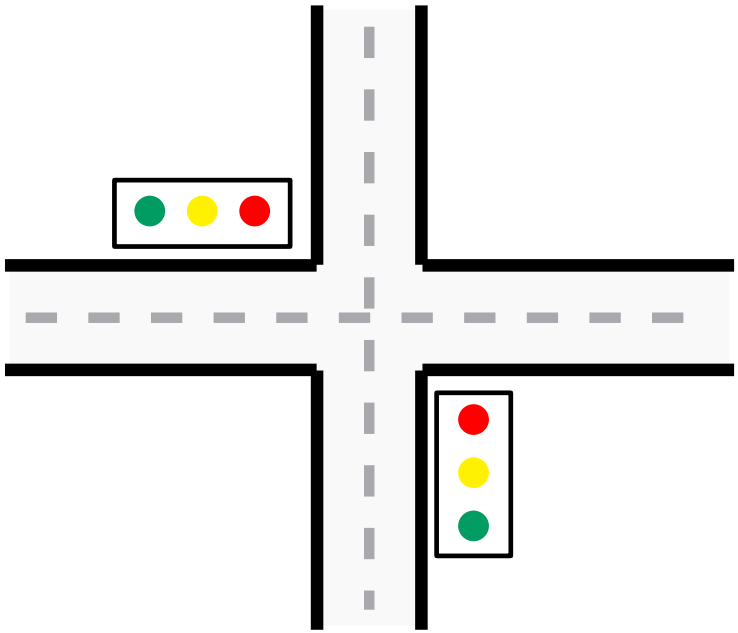
Model Checking



Beispiel: Ampelsystem



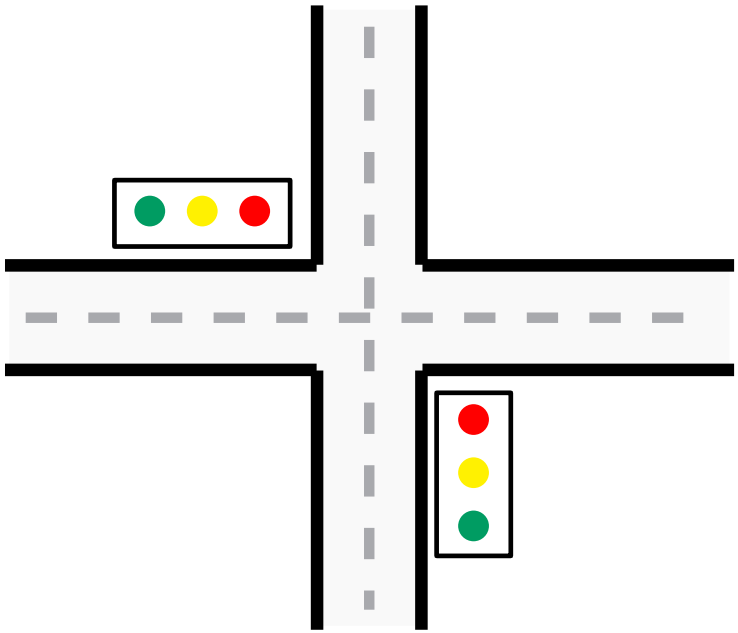
Beispiel: Ampelsystem



paralleles System:

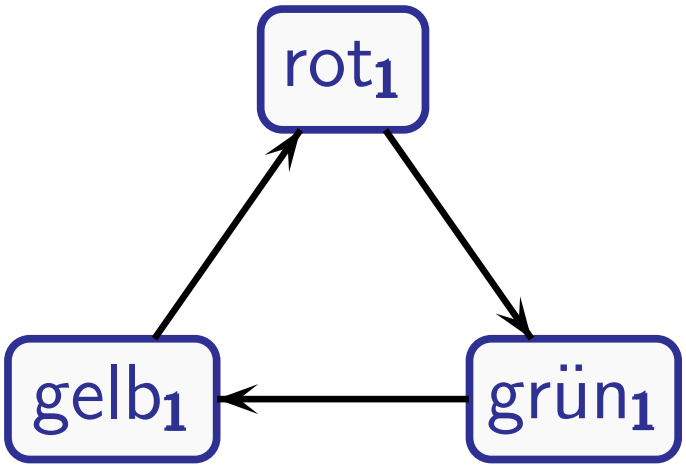
$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

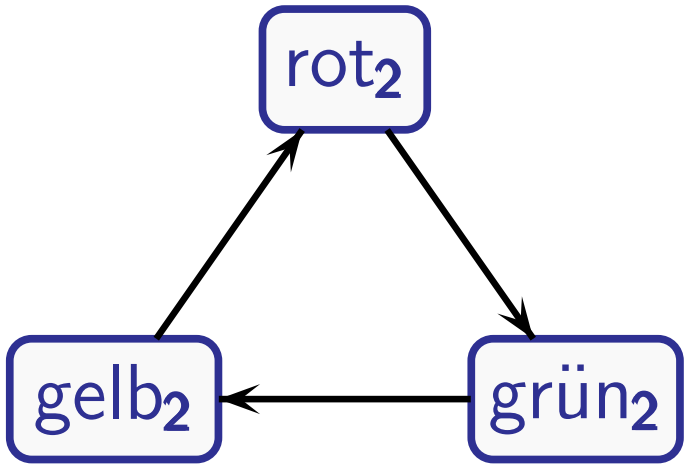
Beispiel: Ampelsystem



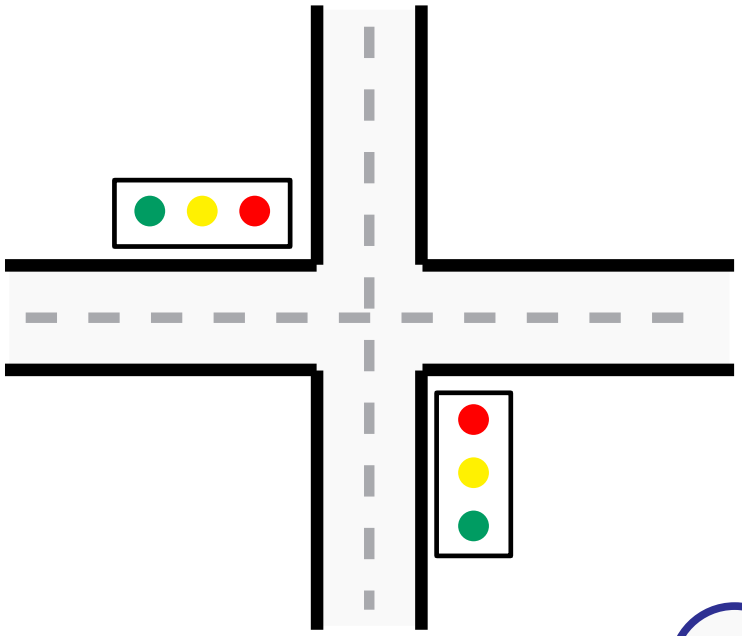
paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$



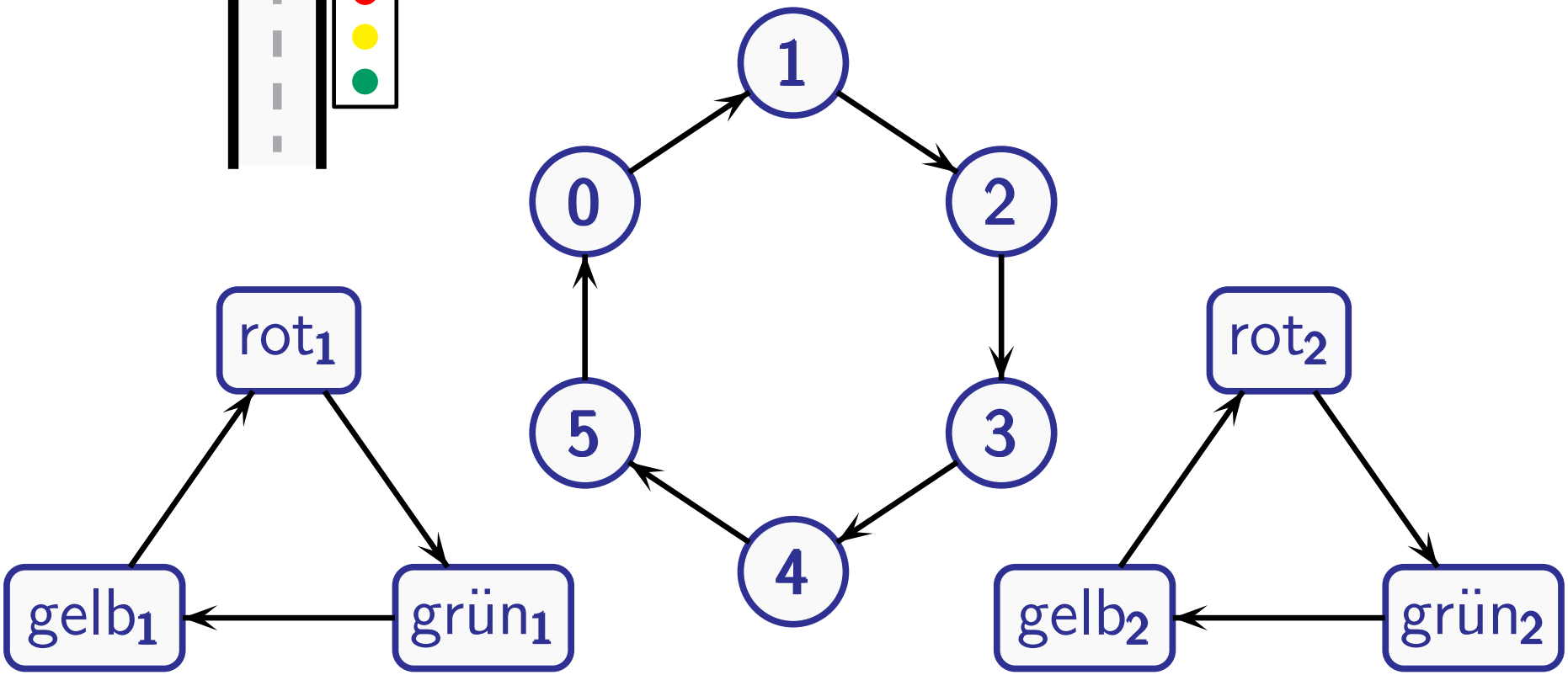


Beispiel: Ampelsystem

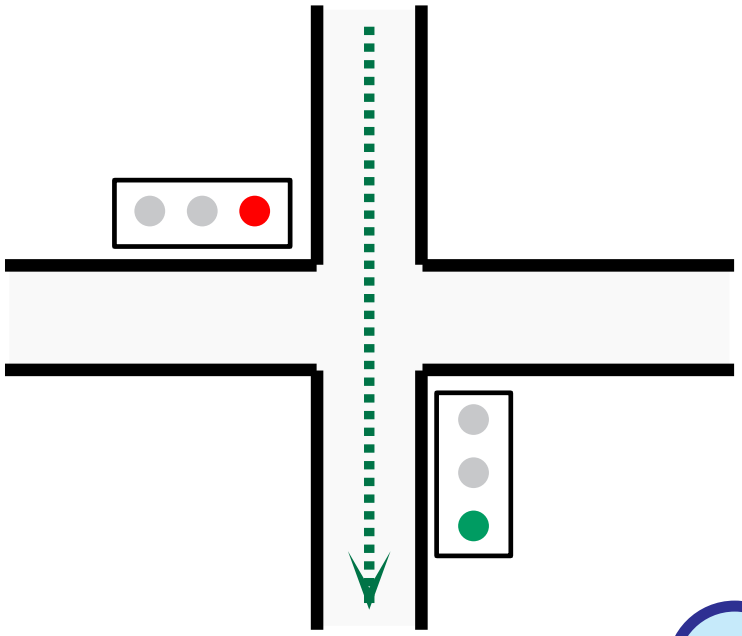


paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

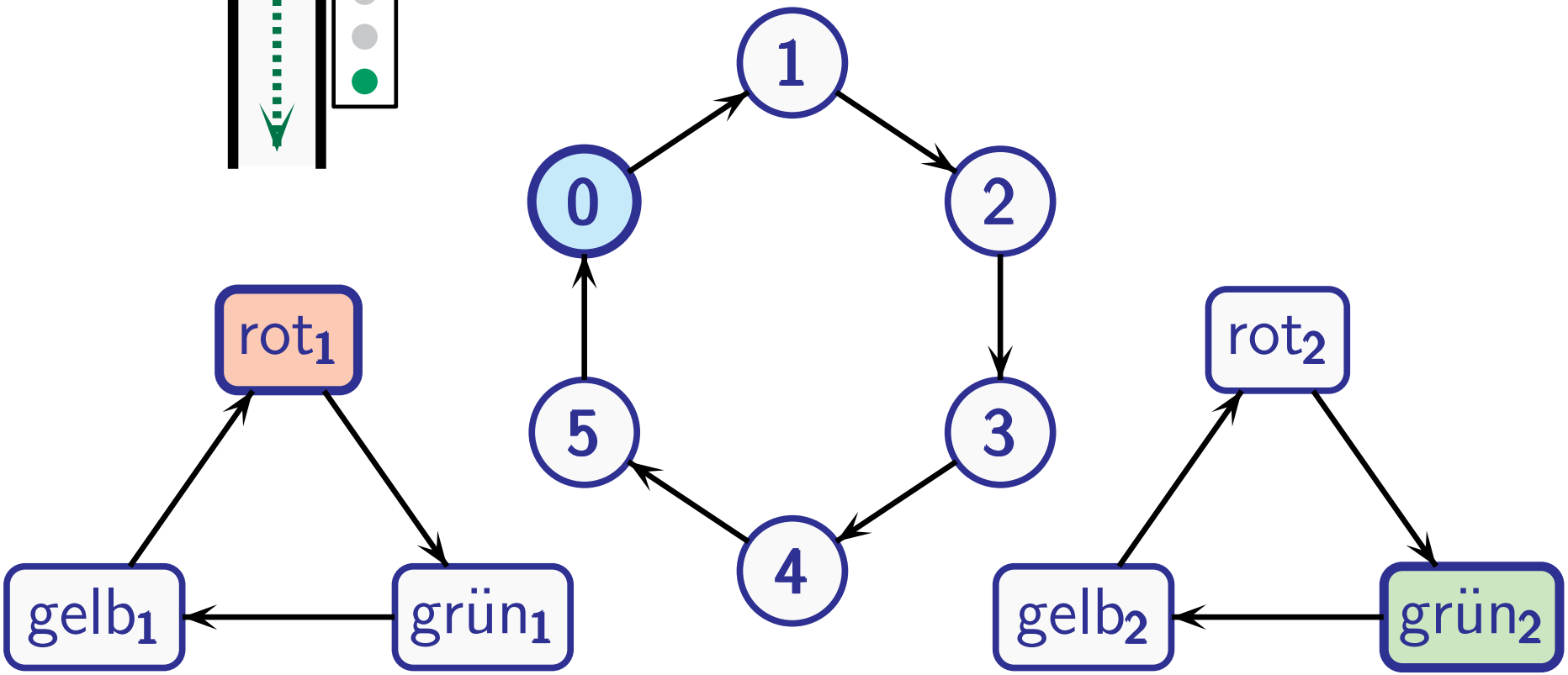


Beispiel: Ampelsystem

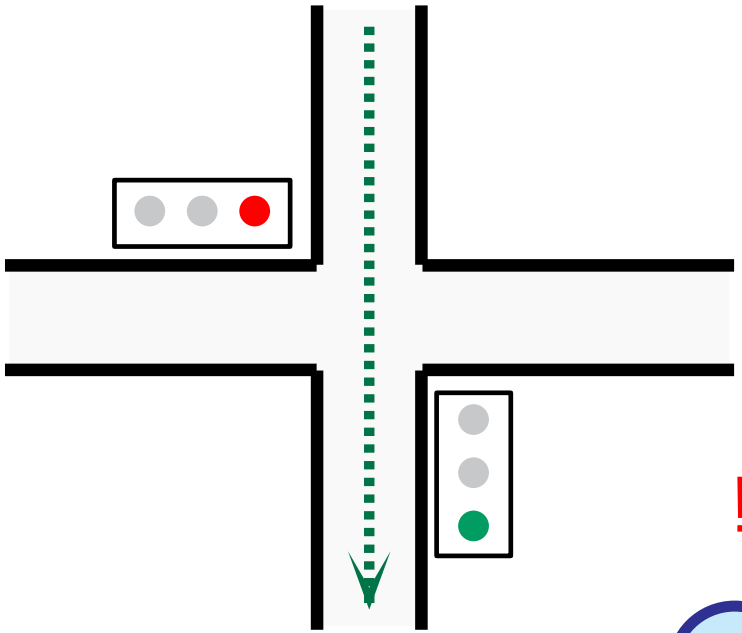


paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

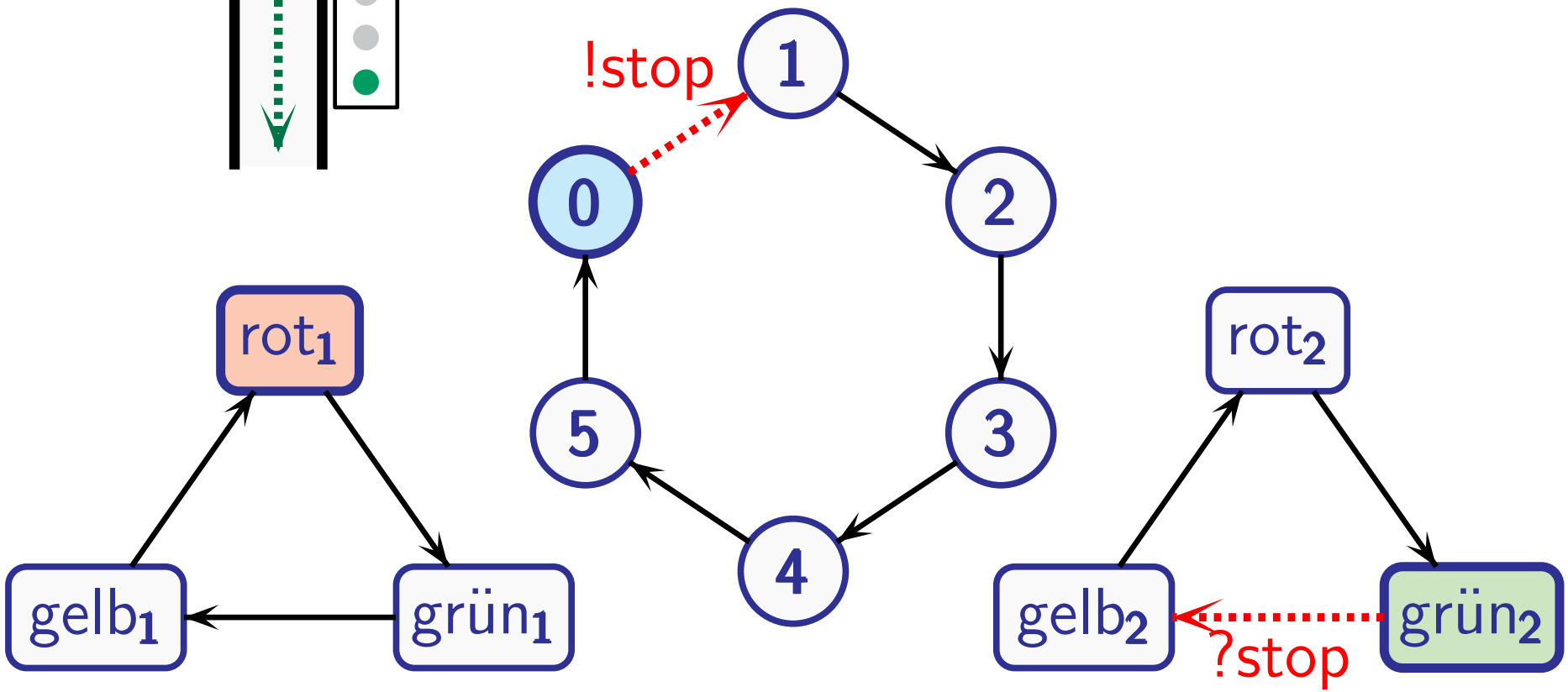


Beispiel: Ampelsystem

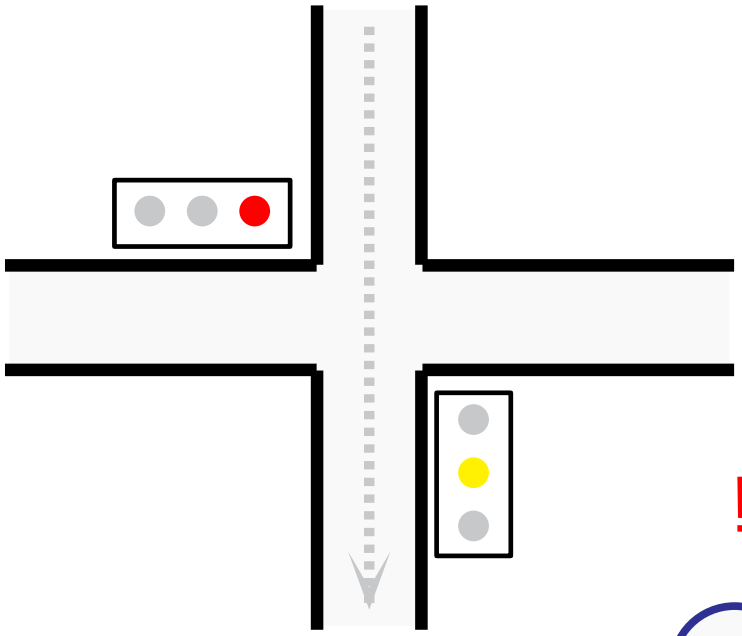


paralleles System:

$$Ampel_1 \parallel Controller \parallel Ampel_2$$

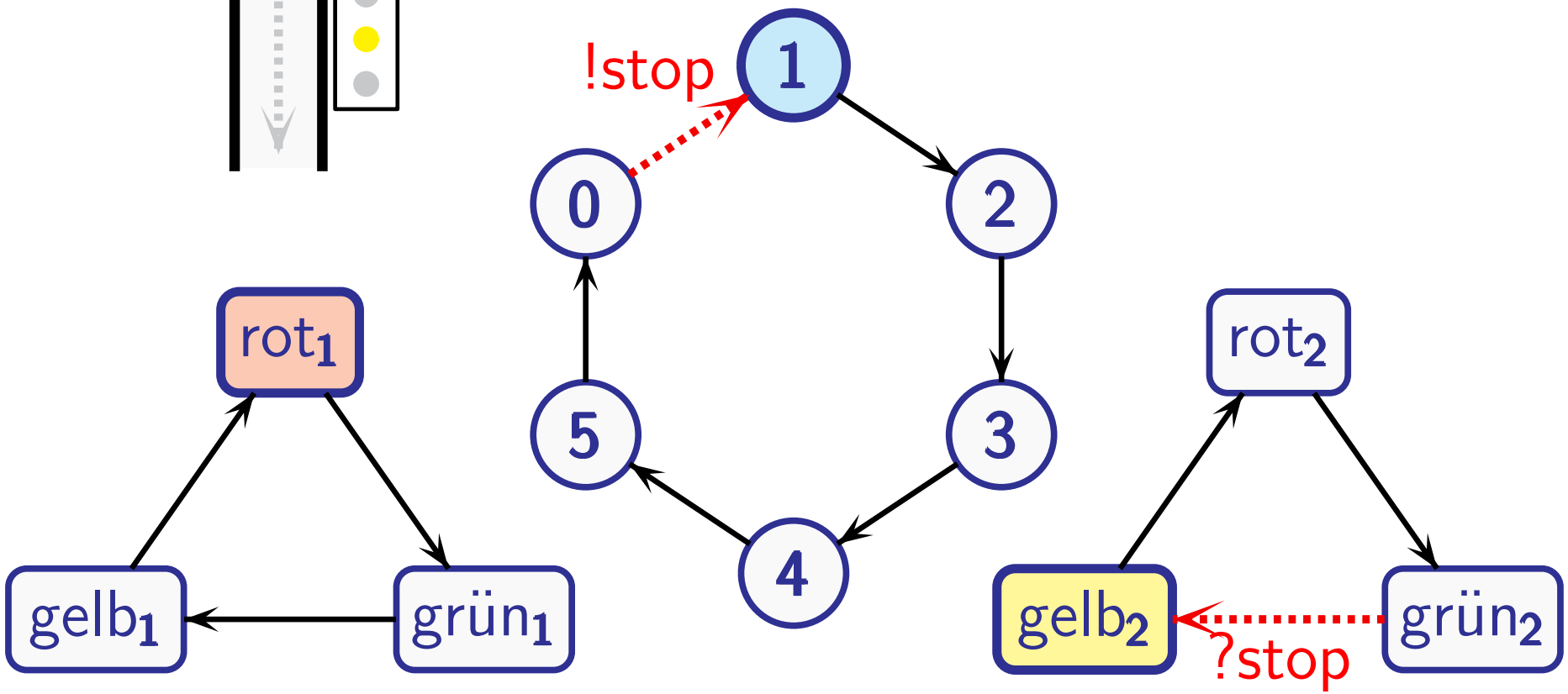


Beispiel: Ampelsystem

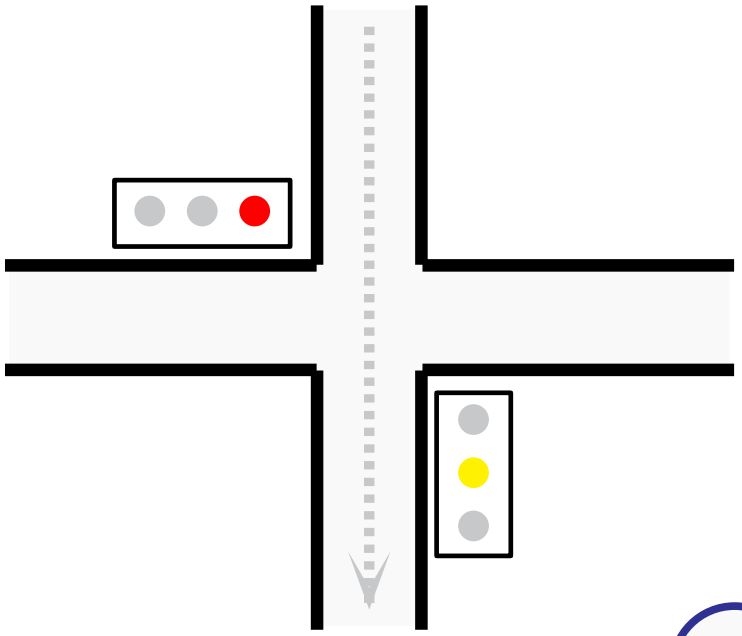


paralleles System:

$$Ampel_1 \parallel Controller \parallel Ampel_2$$

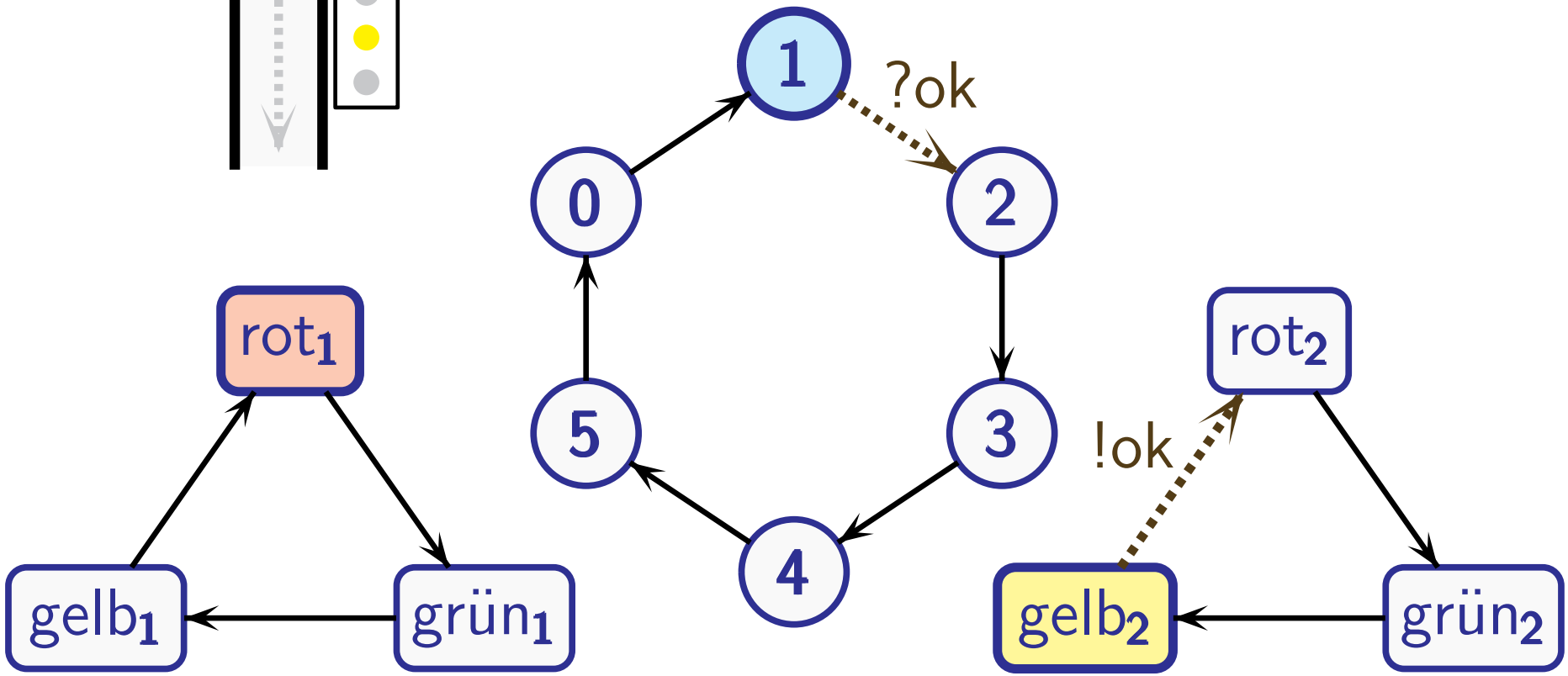


Beispiel: Ampelsystem

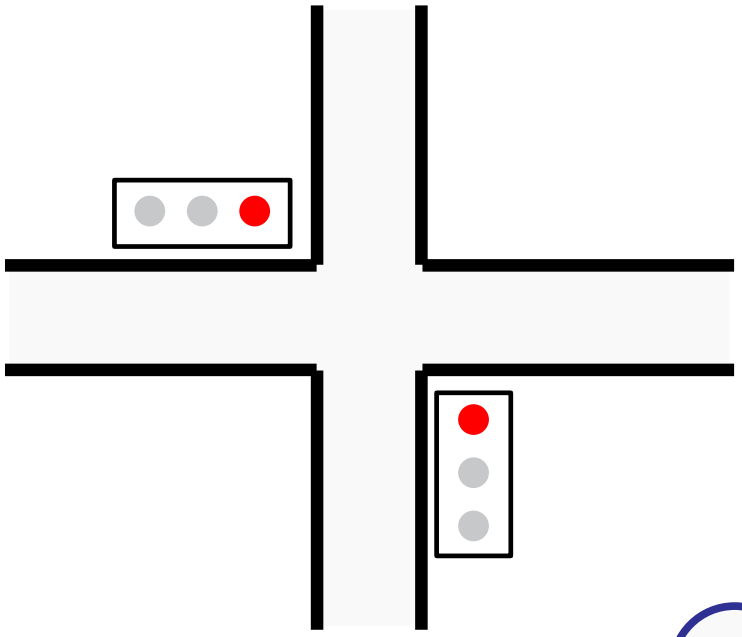


paralleles System:

$$Ampel_1 \parallel Controller \parallel Ampel_2$$

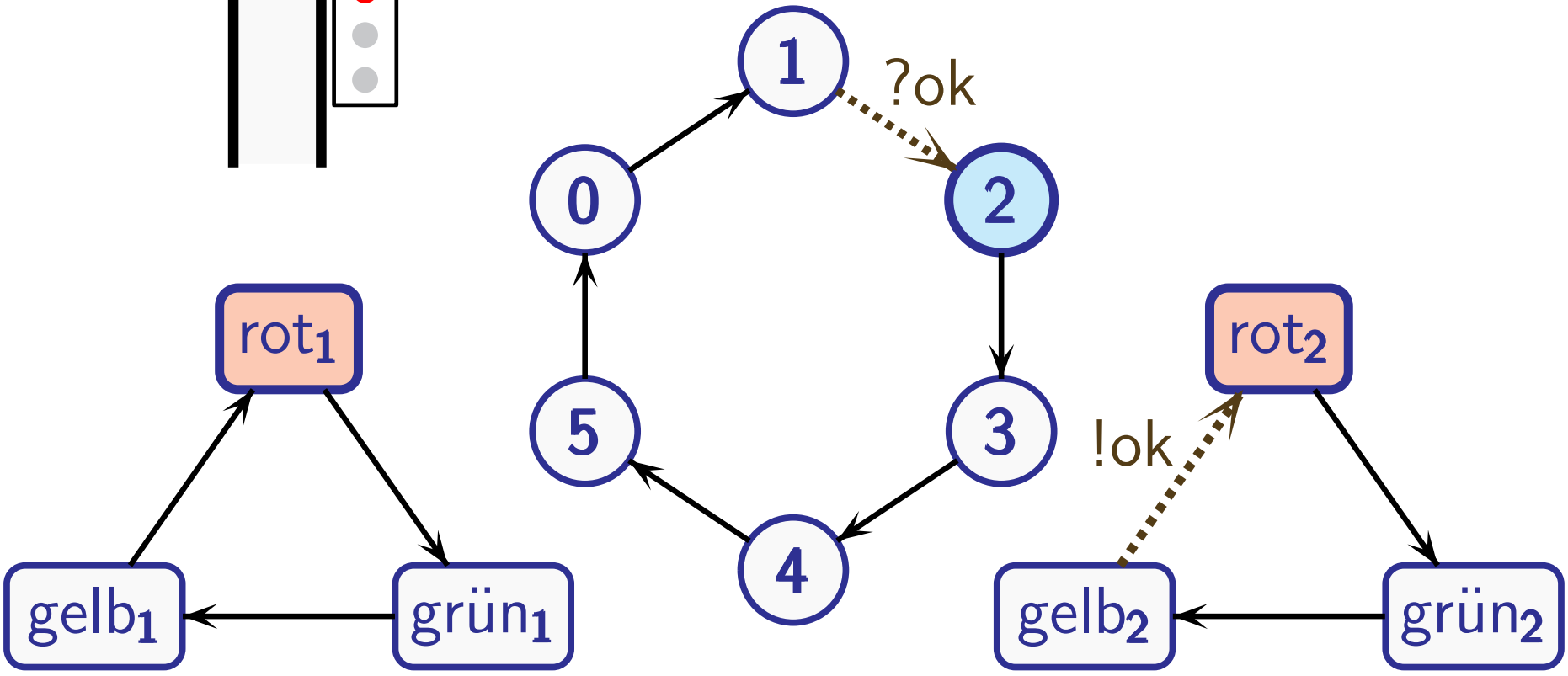


Beispiel: Ampelsystem

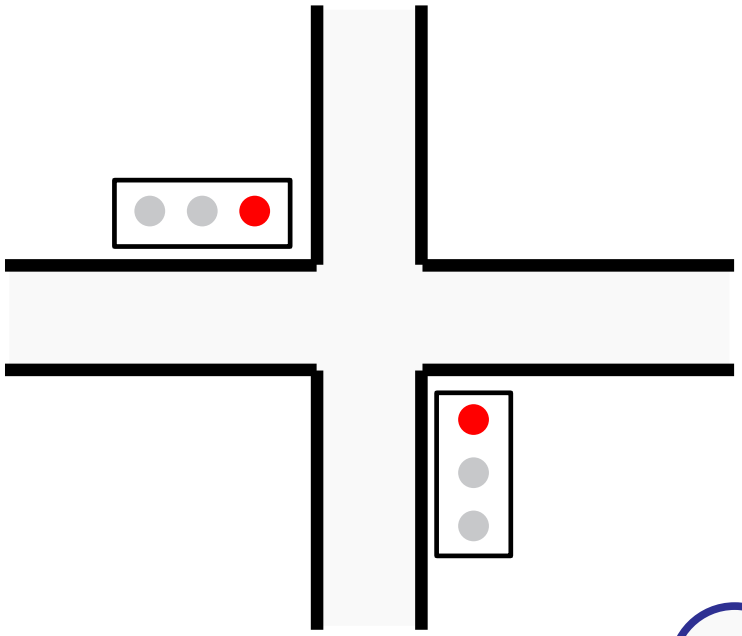


paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

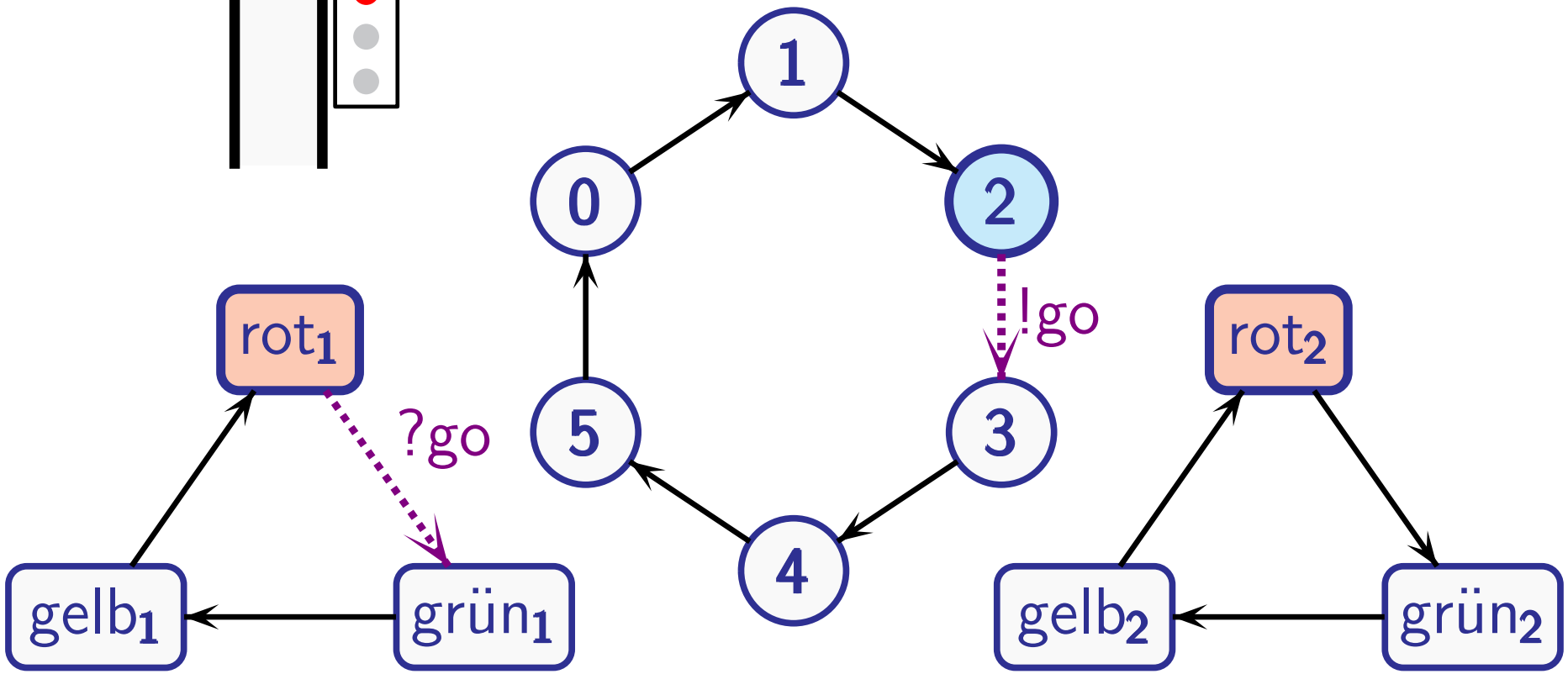


Beispiel: Ampelsystem

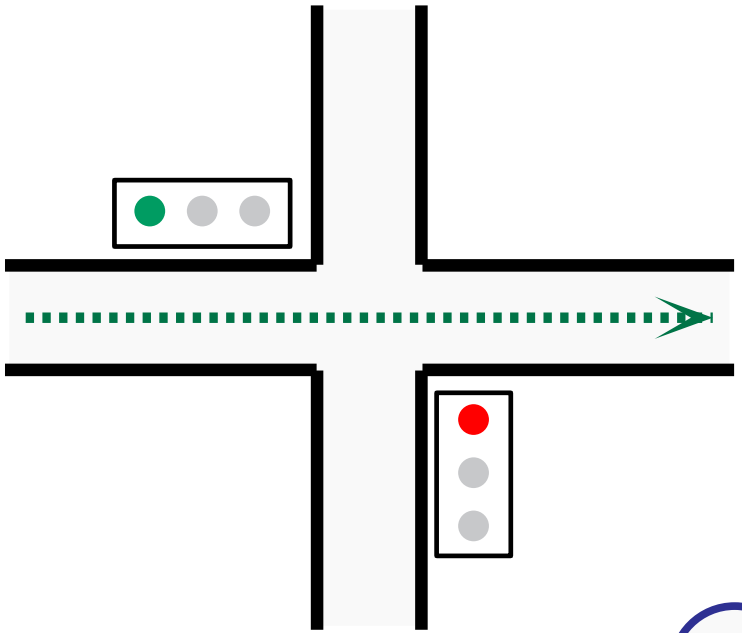


paralleles System:

$$Ampel_1 \parallel Controller \parallel Ampel_2$$

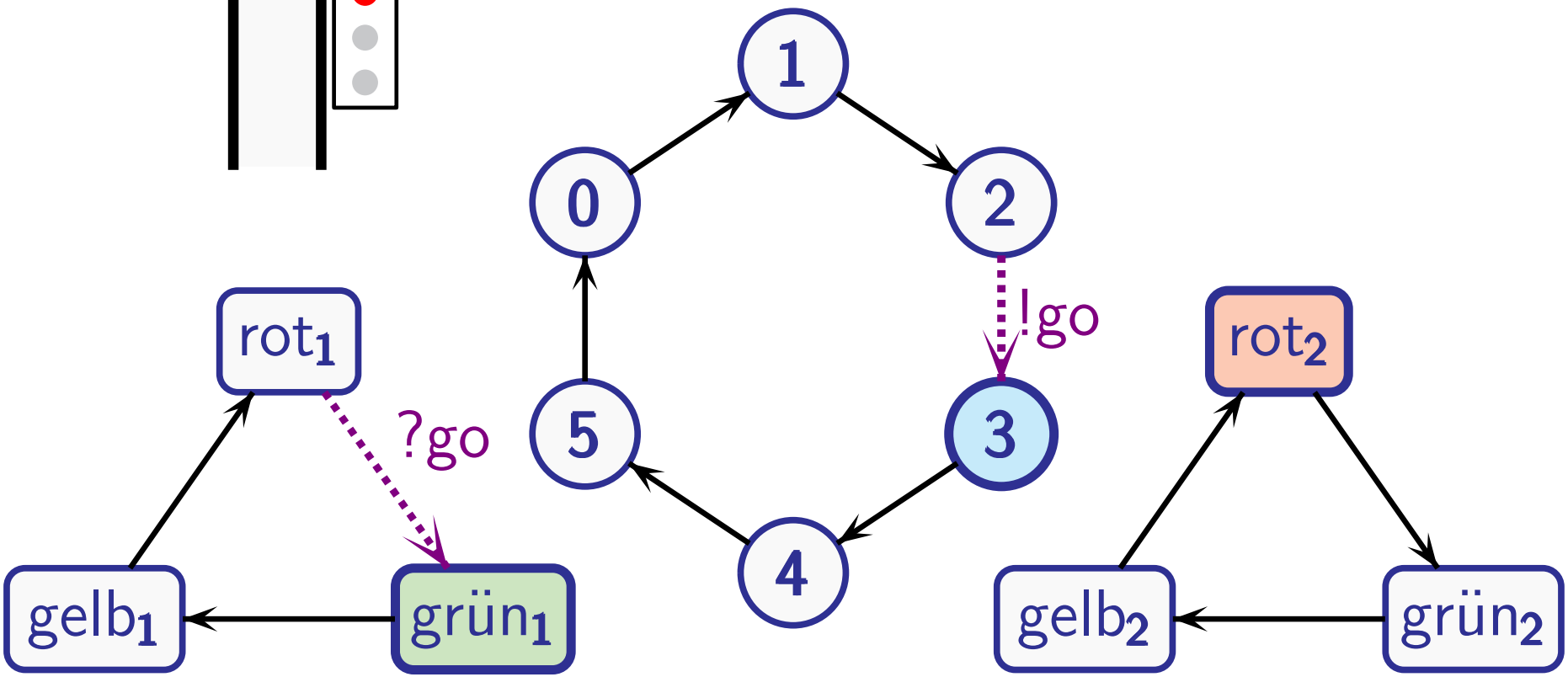


Beispiel: Ampelsystem

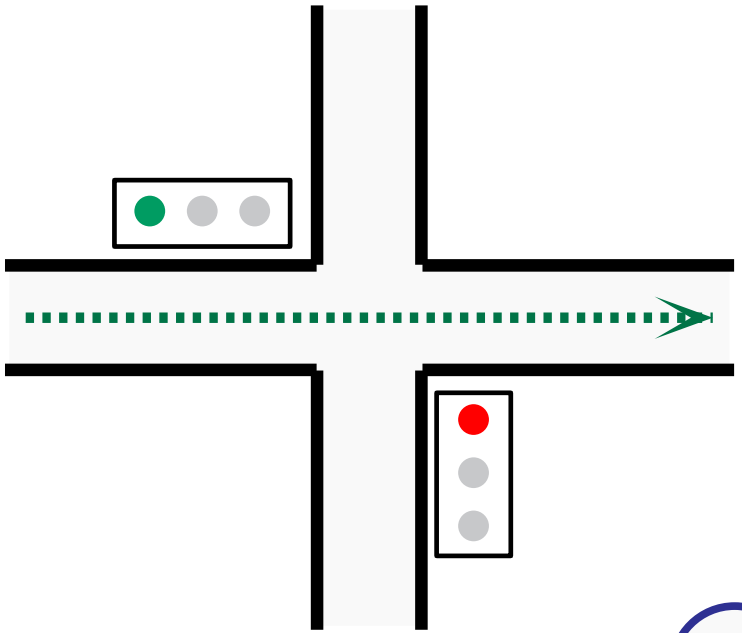


paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

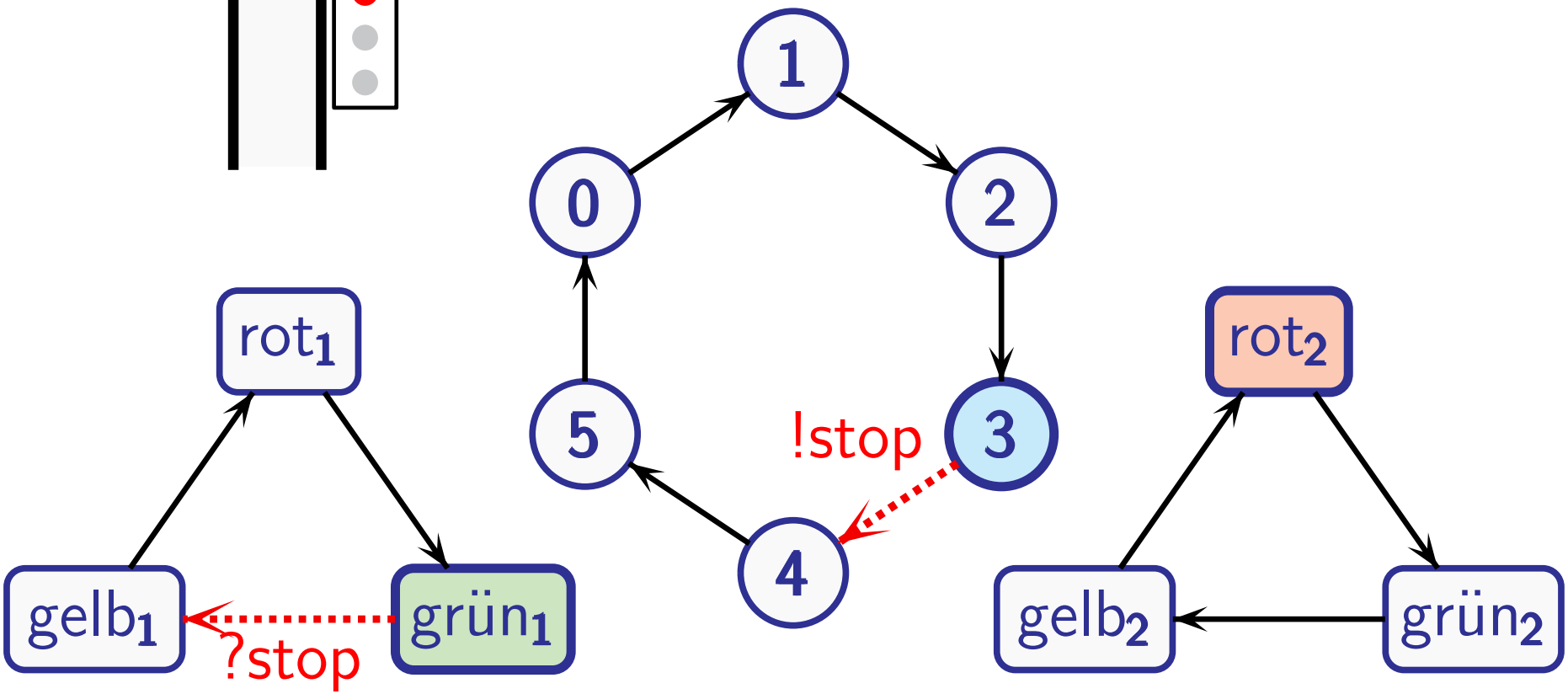


Beispiel: Ampelsystem

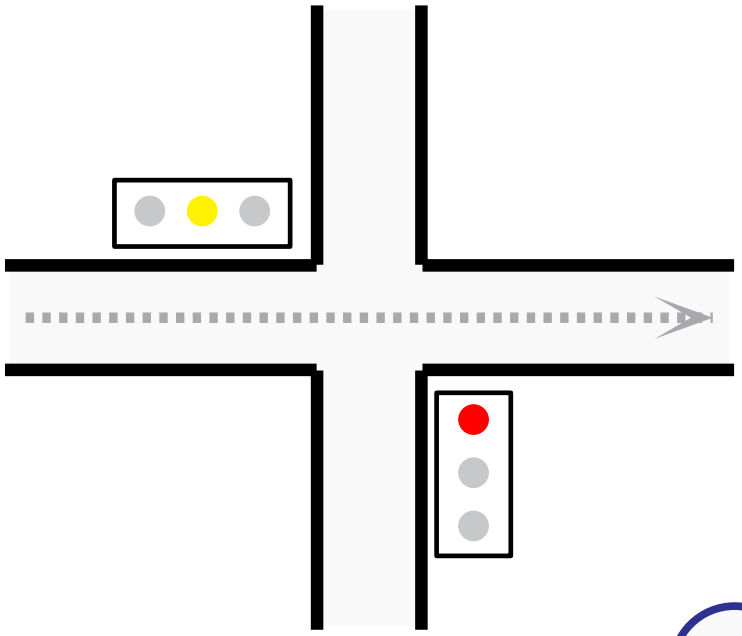


paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

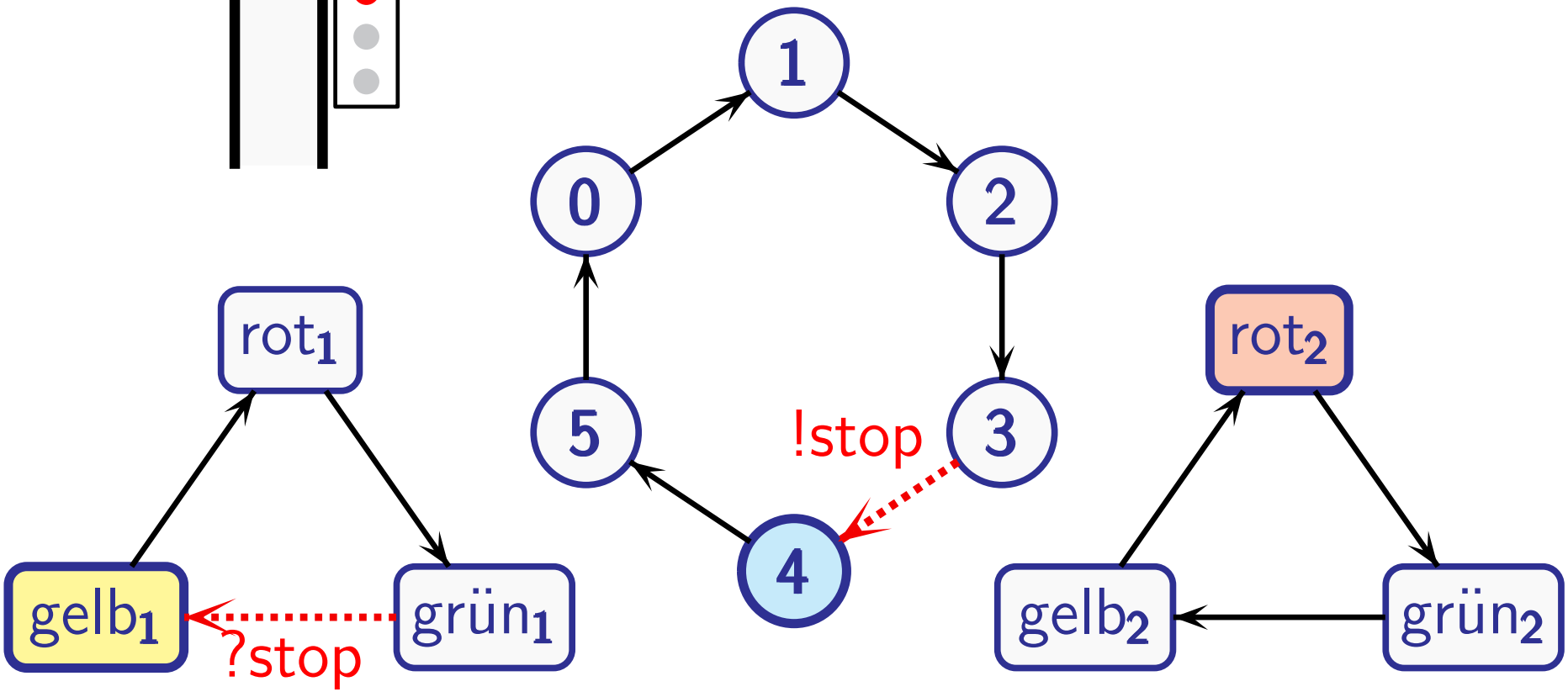


Beispiel: Ampelsystem

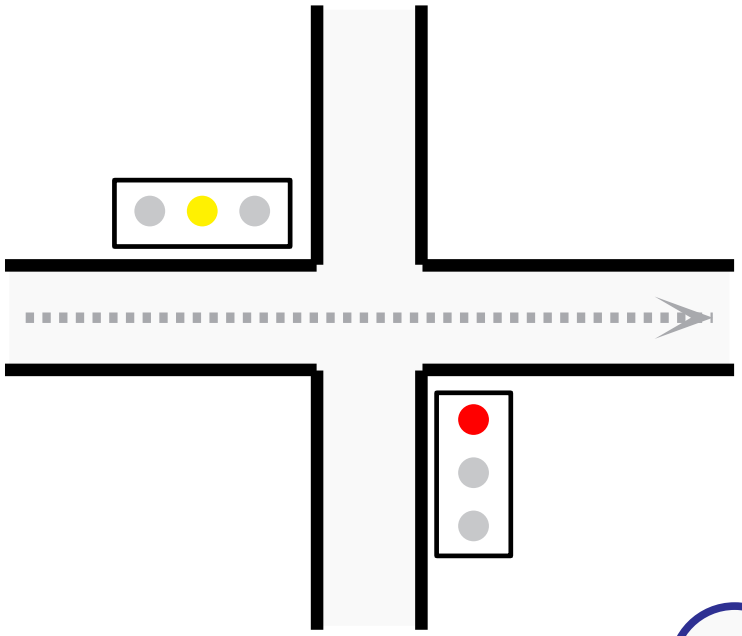


paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

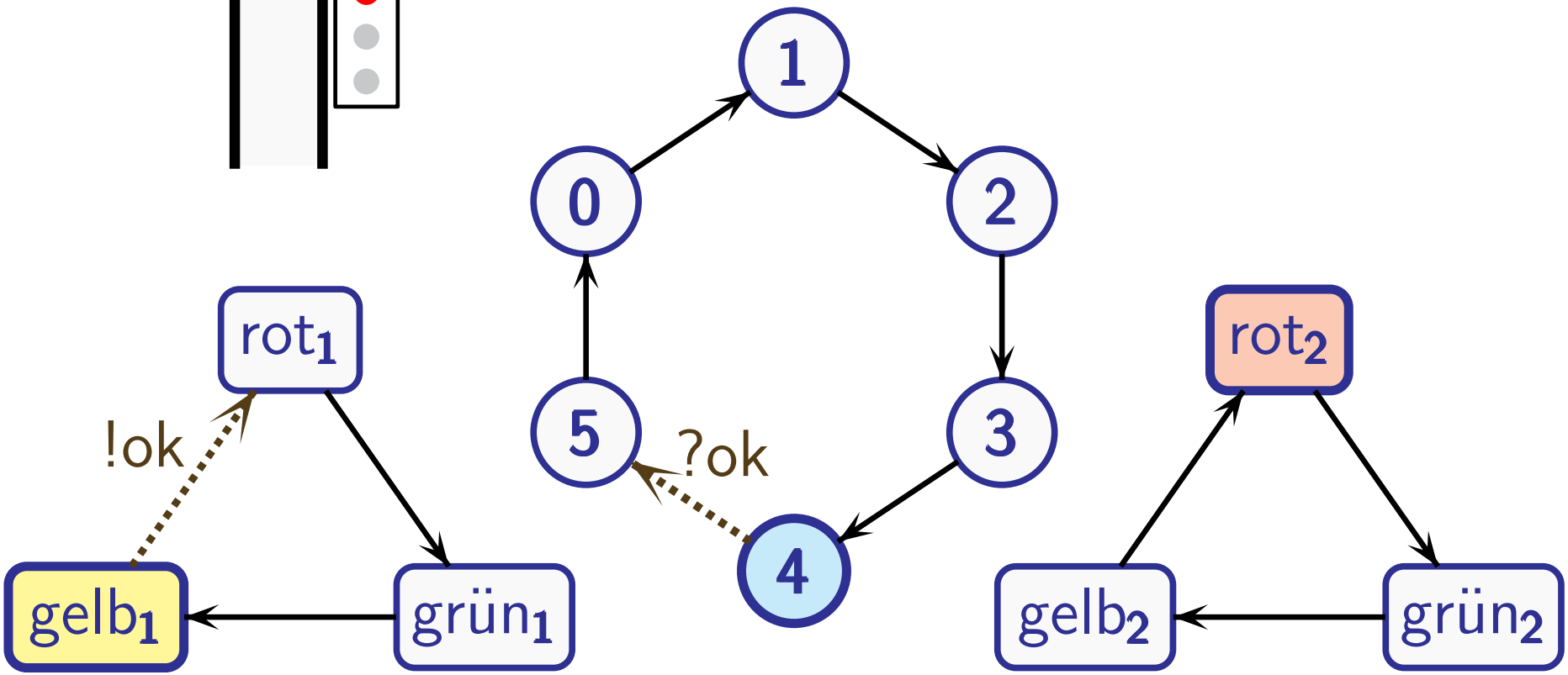


Beispiel: Ampelsystem

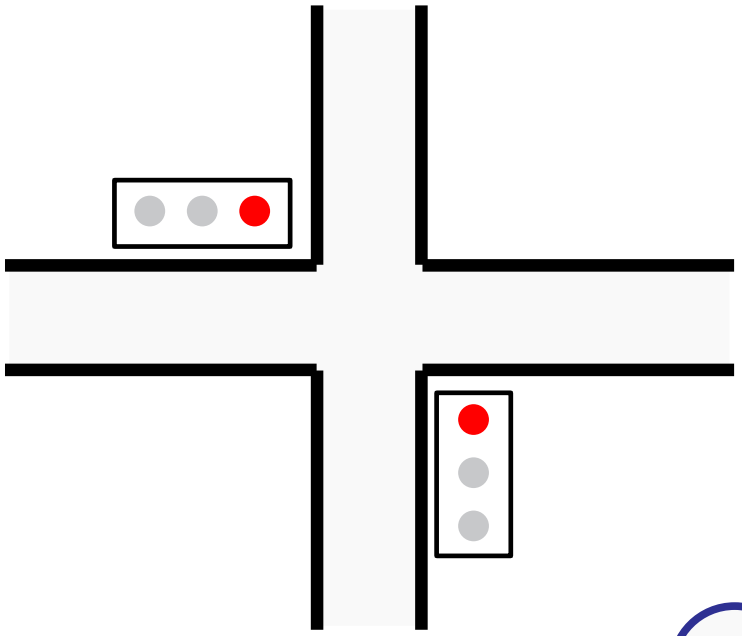


paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

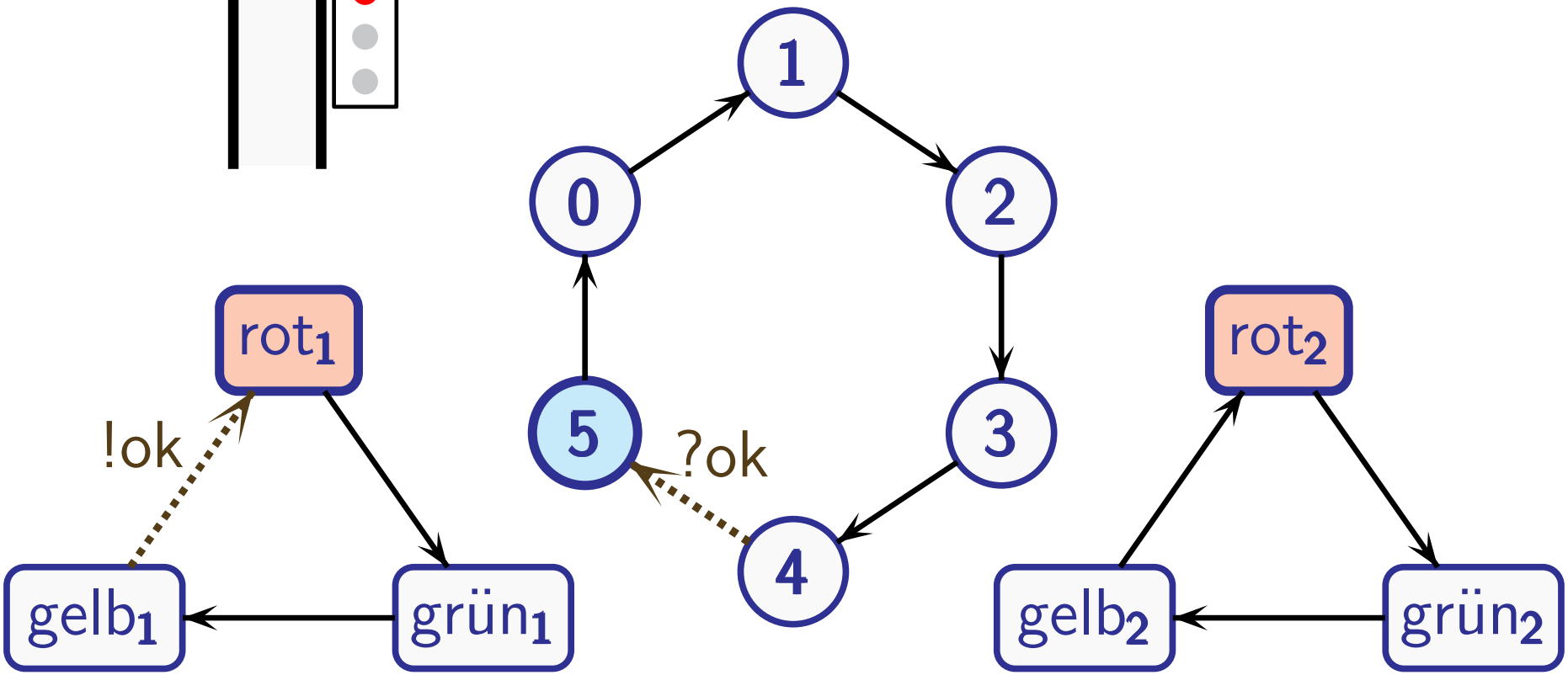


Beispiel: Ampelsystem

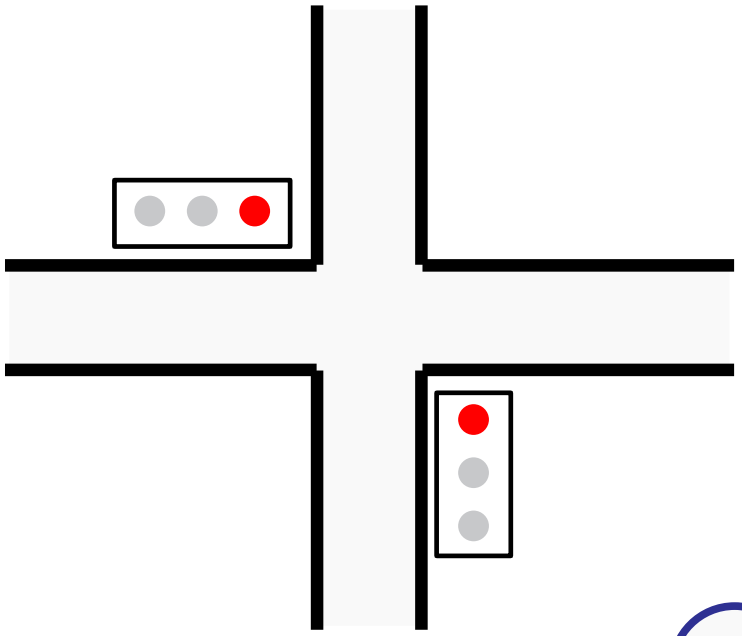


paralleles System:

$$Ampel_1 \parallel Controller \parallel Ampel_2$$

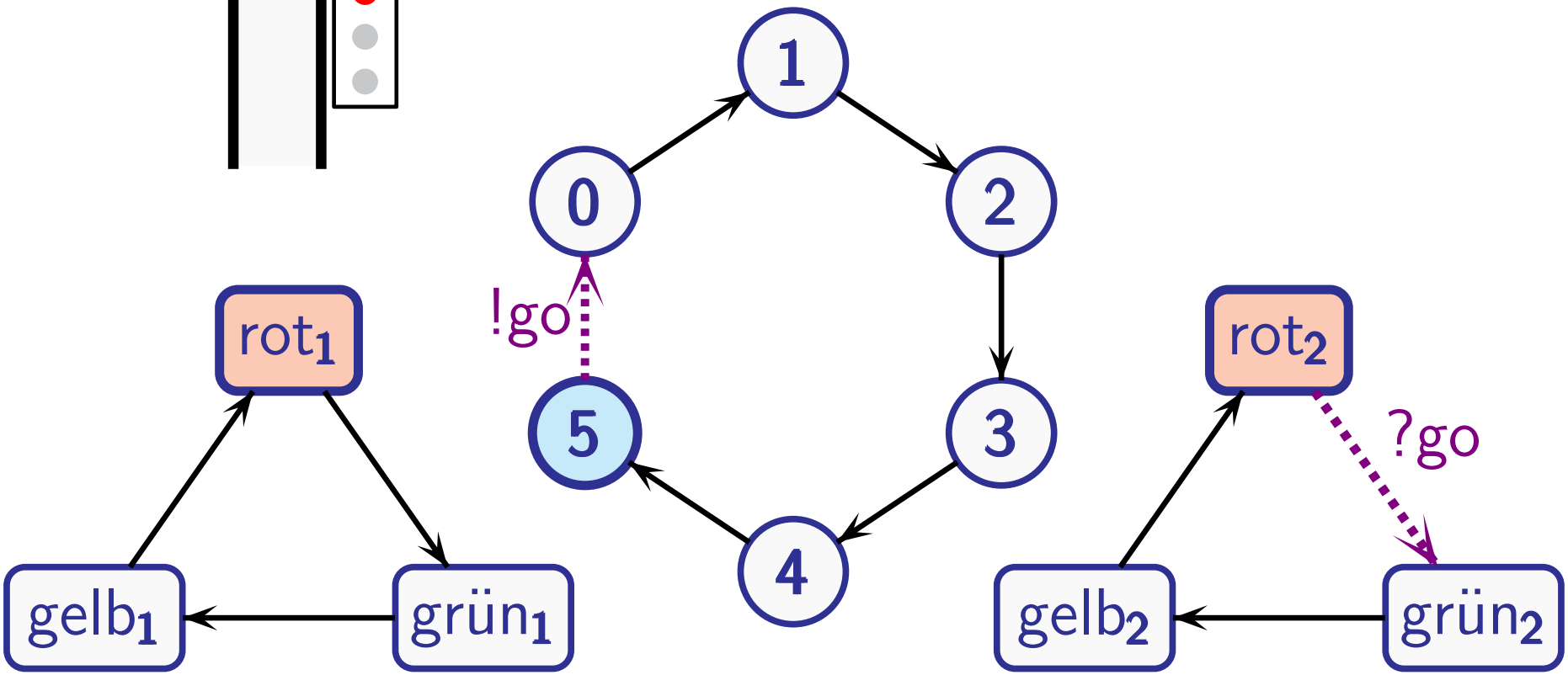


Beispiel: Ampelsystem

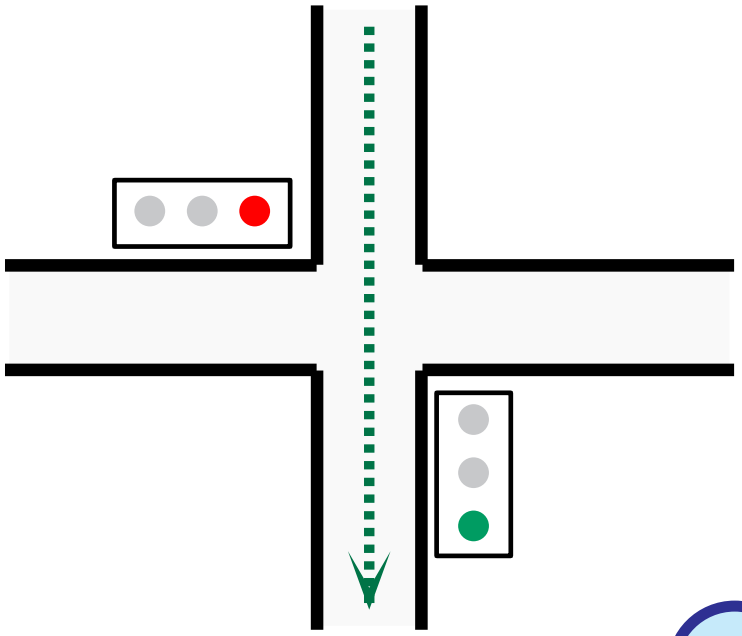


paralleles System:

$$Ampel_1 \parallel Controller \parallel Ampel_2$$

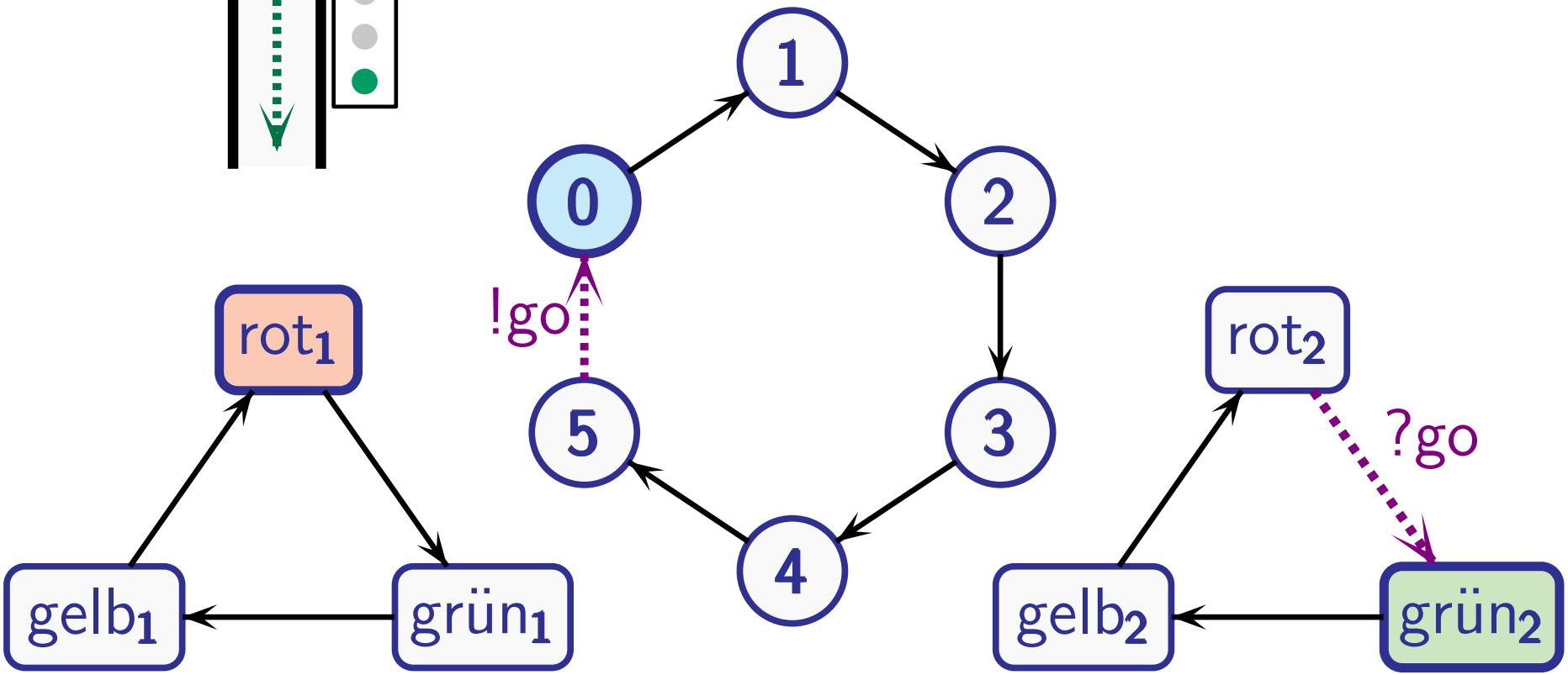


Beispiel: Ampelsystem

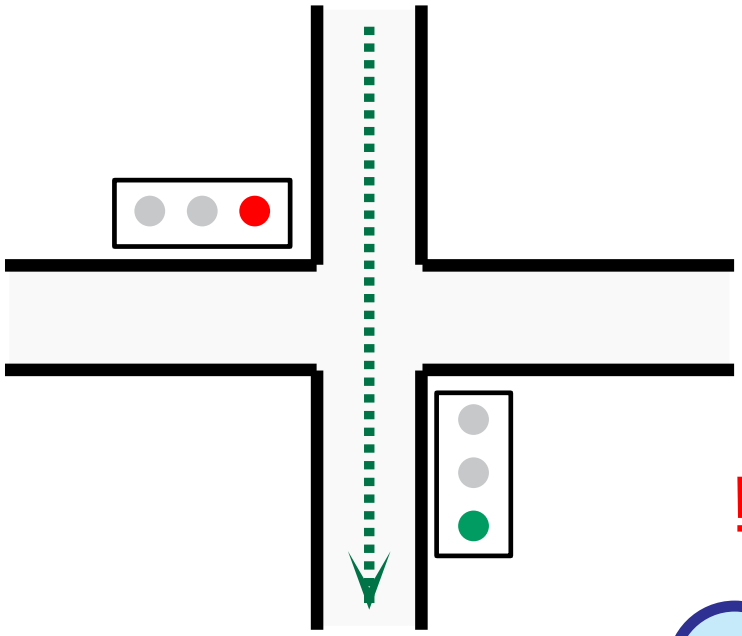


paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

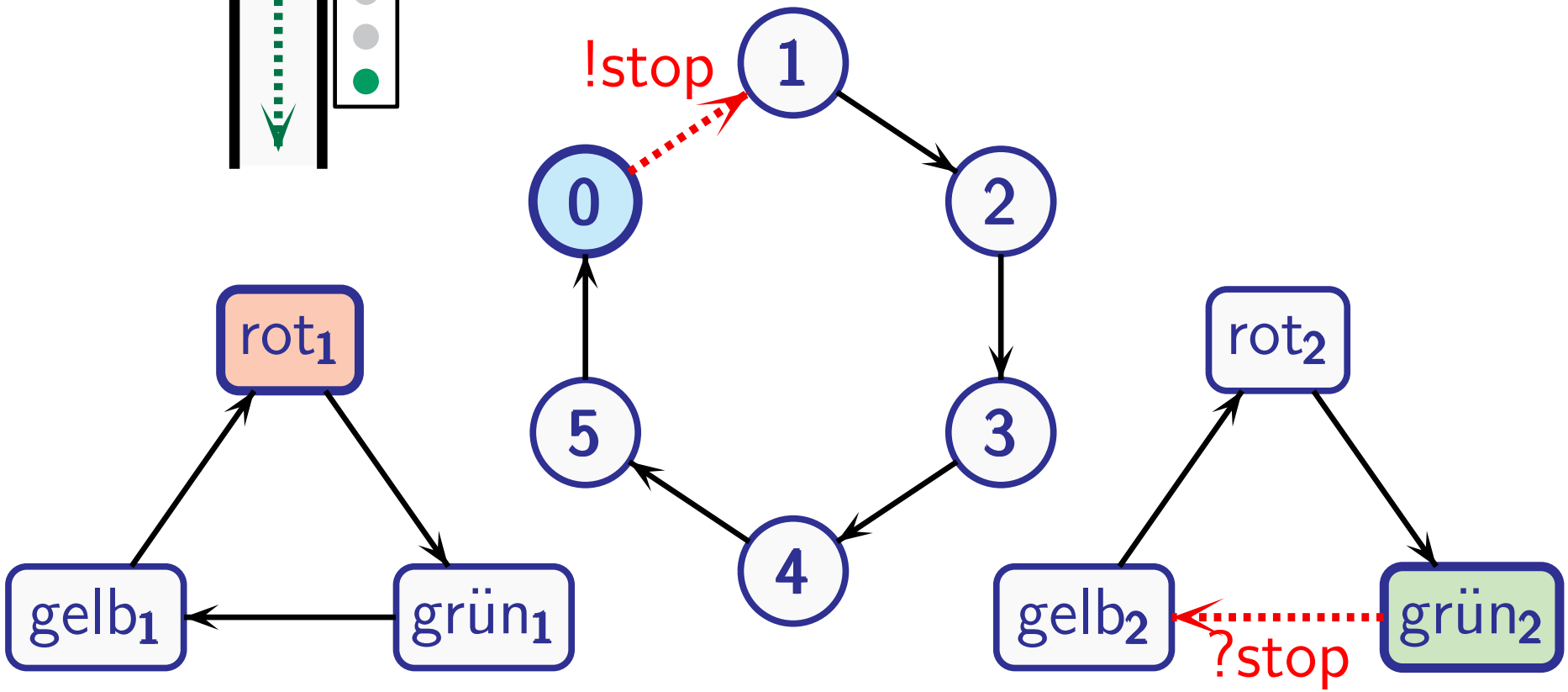


Beispiel: Ampelsystem

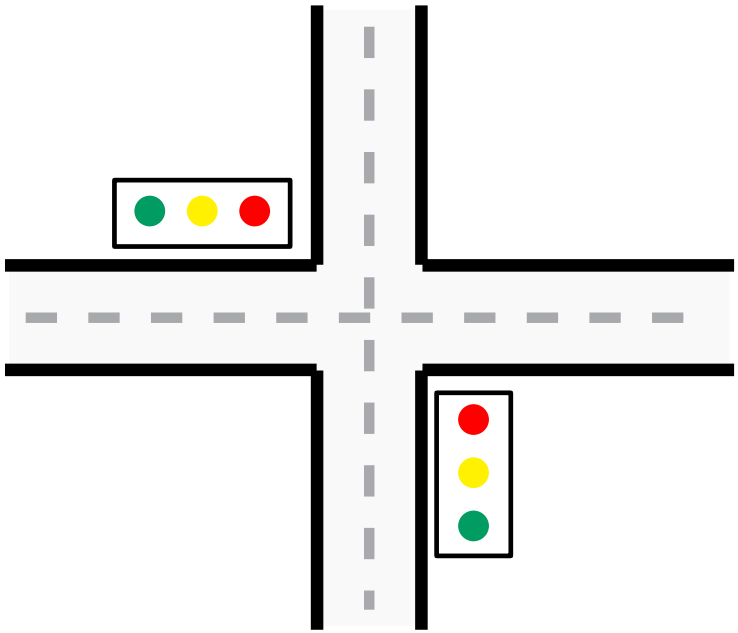


paralleles System:

$$Ampel_1 \parallel Controller \parallel Ampel_2$$



Beispiel: Ampelsystem



paralleles System:

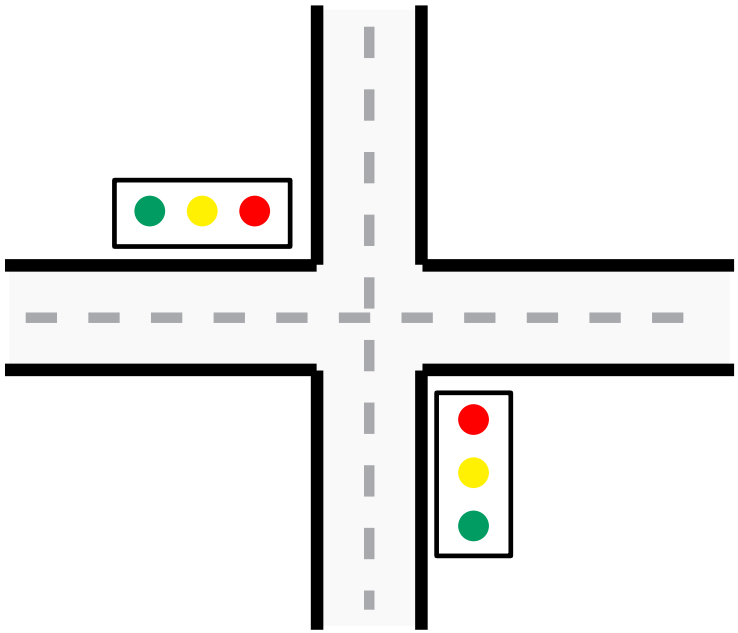
$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

gewünschte Systemeigenschaften:

Sicherheit: die beiden Ampeln sind niemals gleichzeitig grün

Lebendigkeit: jede Ampel wird irgendwann grün

Beispiel: Ampelsystem



paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

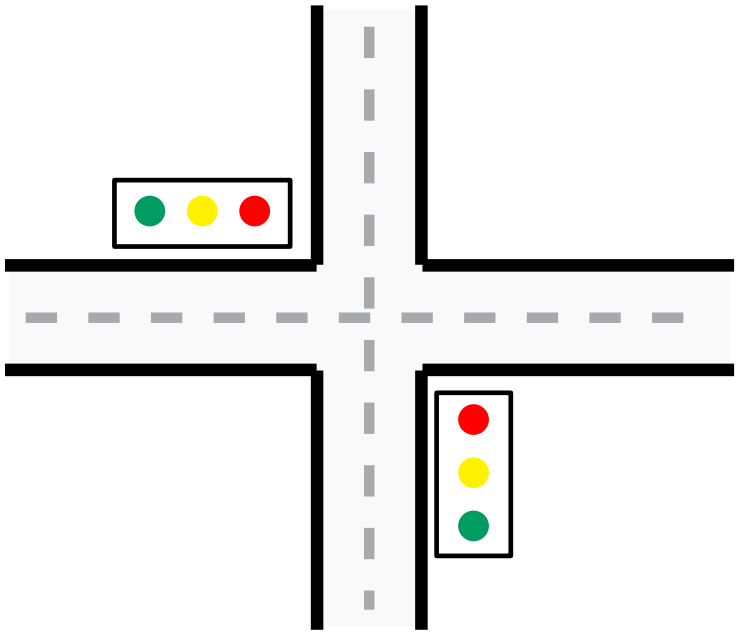
gewünschte Systemeigenschaften:

Sicherheit: die beiden Ampeln sind niemals gleichzeitig grün

Lebendigkeit: jede Ampel wird irgendwann grün

... wann ? was ist die durchschnittliche Wartezeit ?

Echtzeitvariante der Ampel

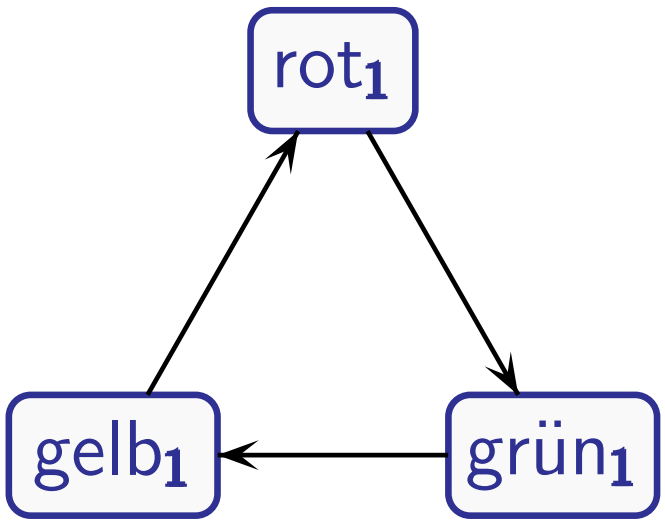


paralleles System:

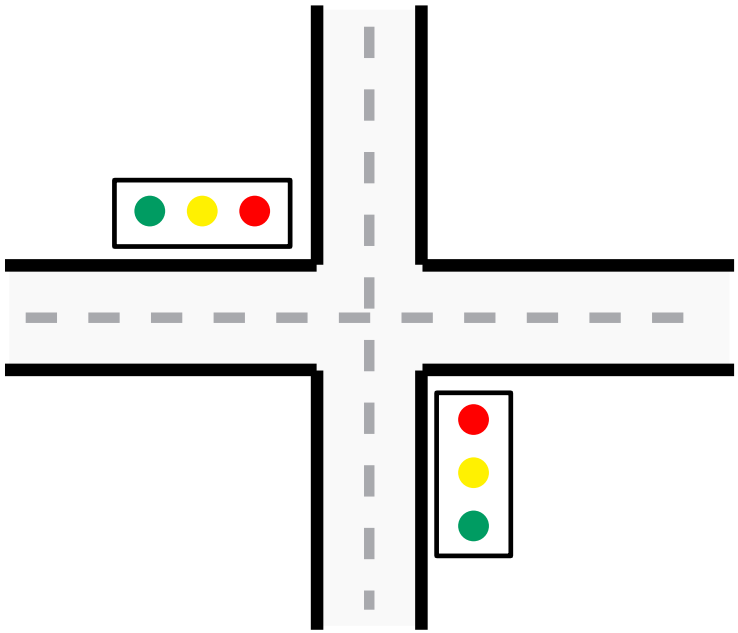
$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

↑

Modellierung
mit Annahmen zum
Echtzeitverhalten



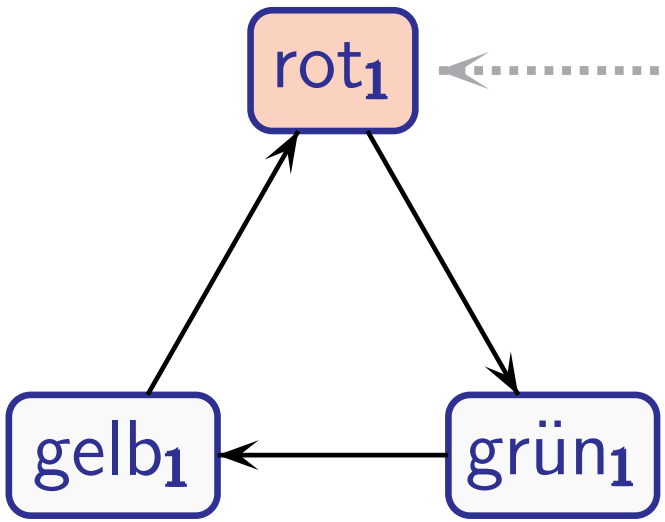
Echtzeitvariante der Ampel



paralleles System:

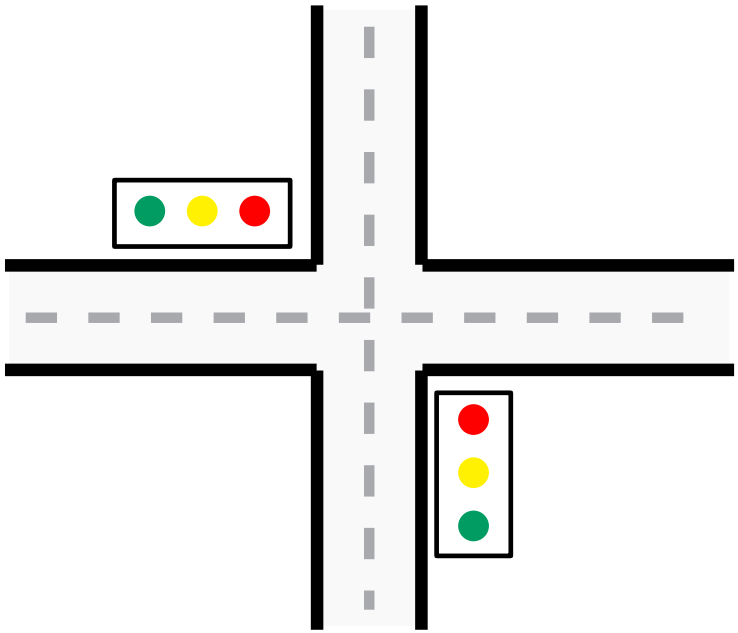
$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

Modellierung mit Annahmen zum Echtzeitverhalten



Verweildauer zwischen 2 und 3 min

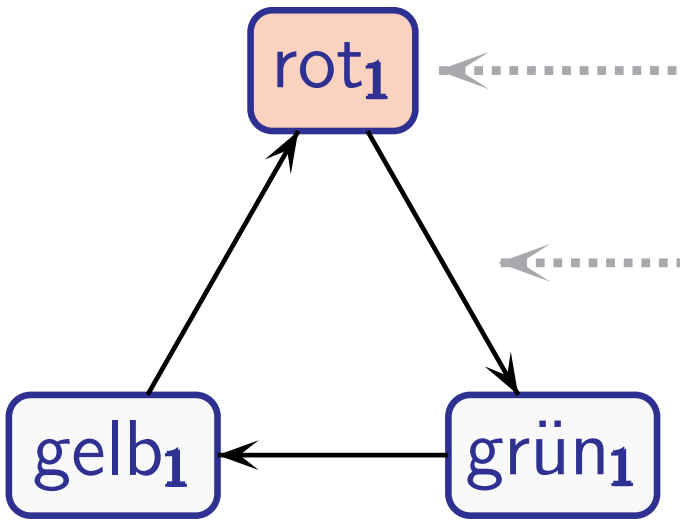
Echtzeitvariante der Ampel



paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

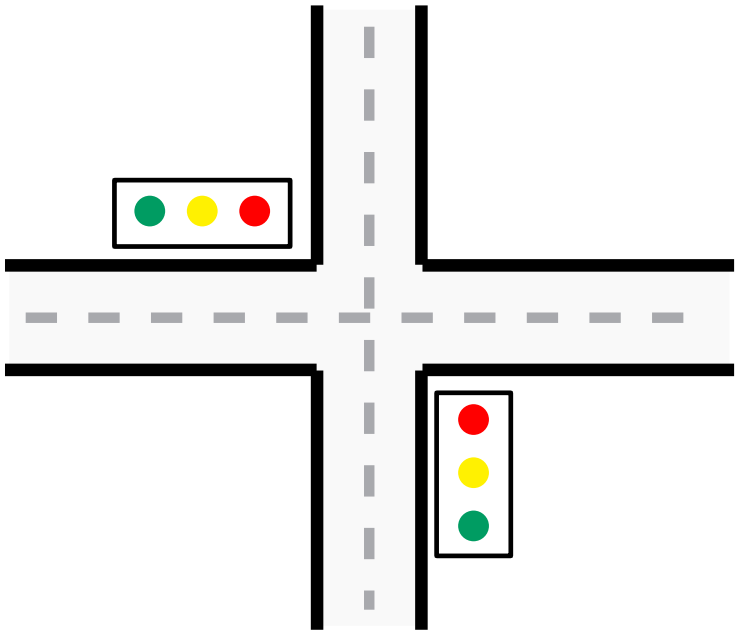
Modellierung
mit Annahmen zum
Echtzeitverhalten



Verweildauer zwischen 2 und 3 min

Verzögerung 1 sec

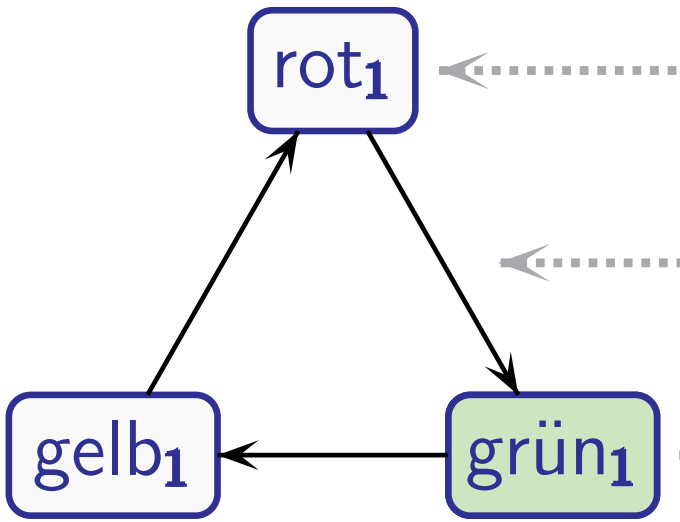
Echtzeitvariante der Ampel



paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

Modellierung
mit Annahmen zum
Echtzeitverhalten

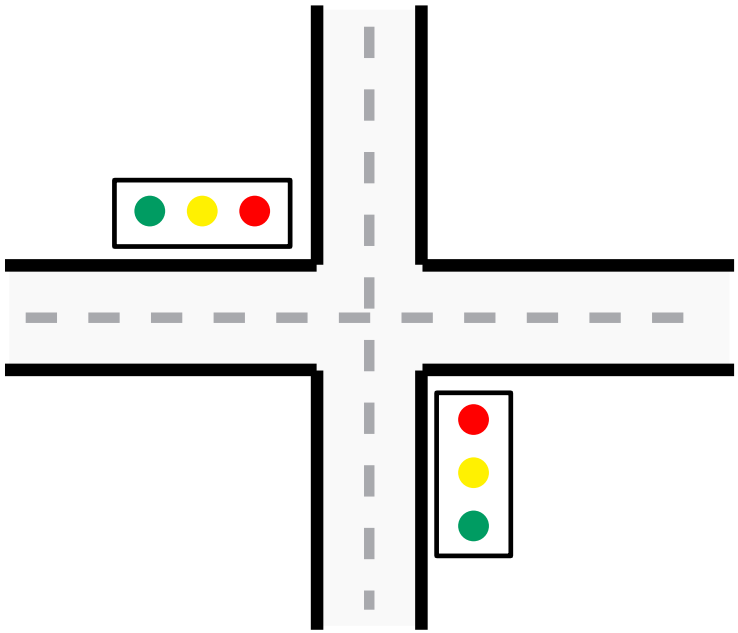


Verweildauer zwischen 2 und 3 min

Verzögerung 1 sec

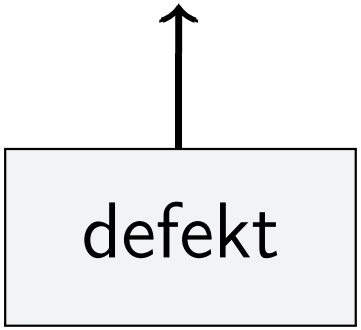
Verweildauer 3 min

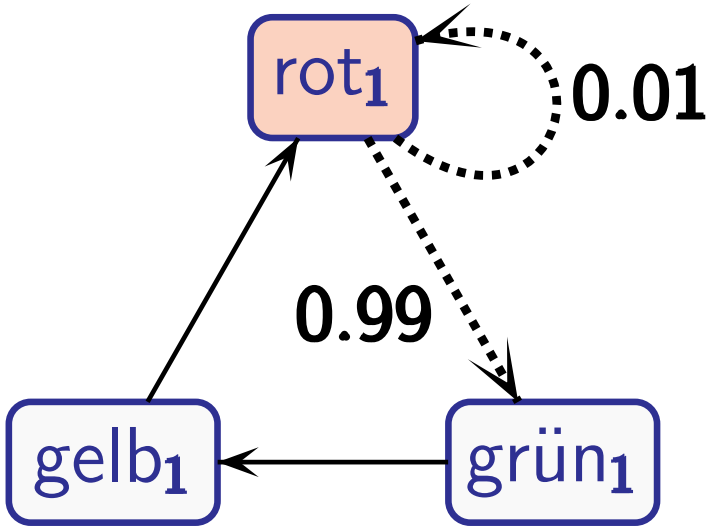
Defekte Ampel



paralleles System:

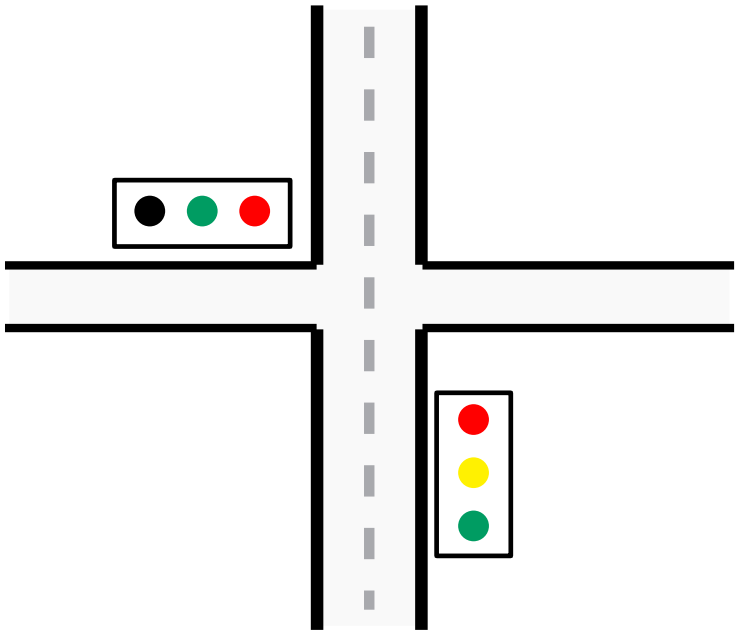
$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$





probabilistisches Modell für Fehlerwahrscheinlichkeit $\frac{1}{100}$

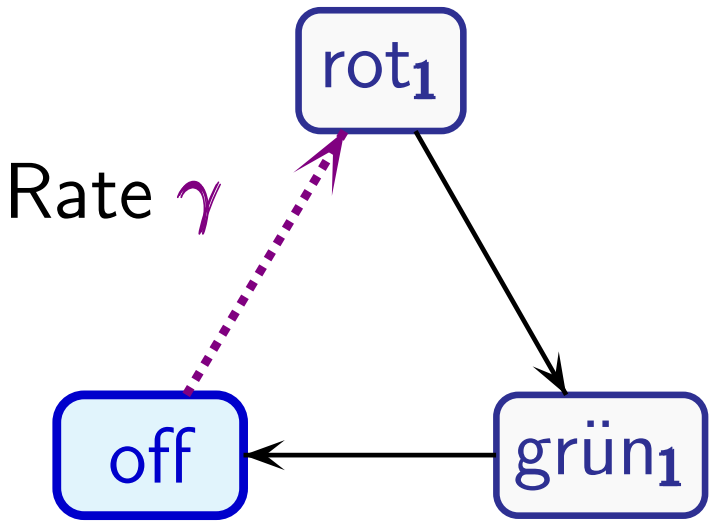
Fußgängerampel im Normalzustand “off”



paralleles System:

$$\mathit{Ampel}_1 \parallel \mathit{Controller} \parallel \mathit{Ampel}_2$$

Fußgängerampel
mit Bedarfsknopf



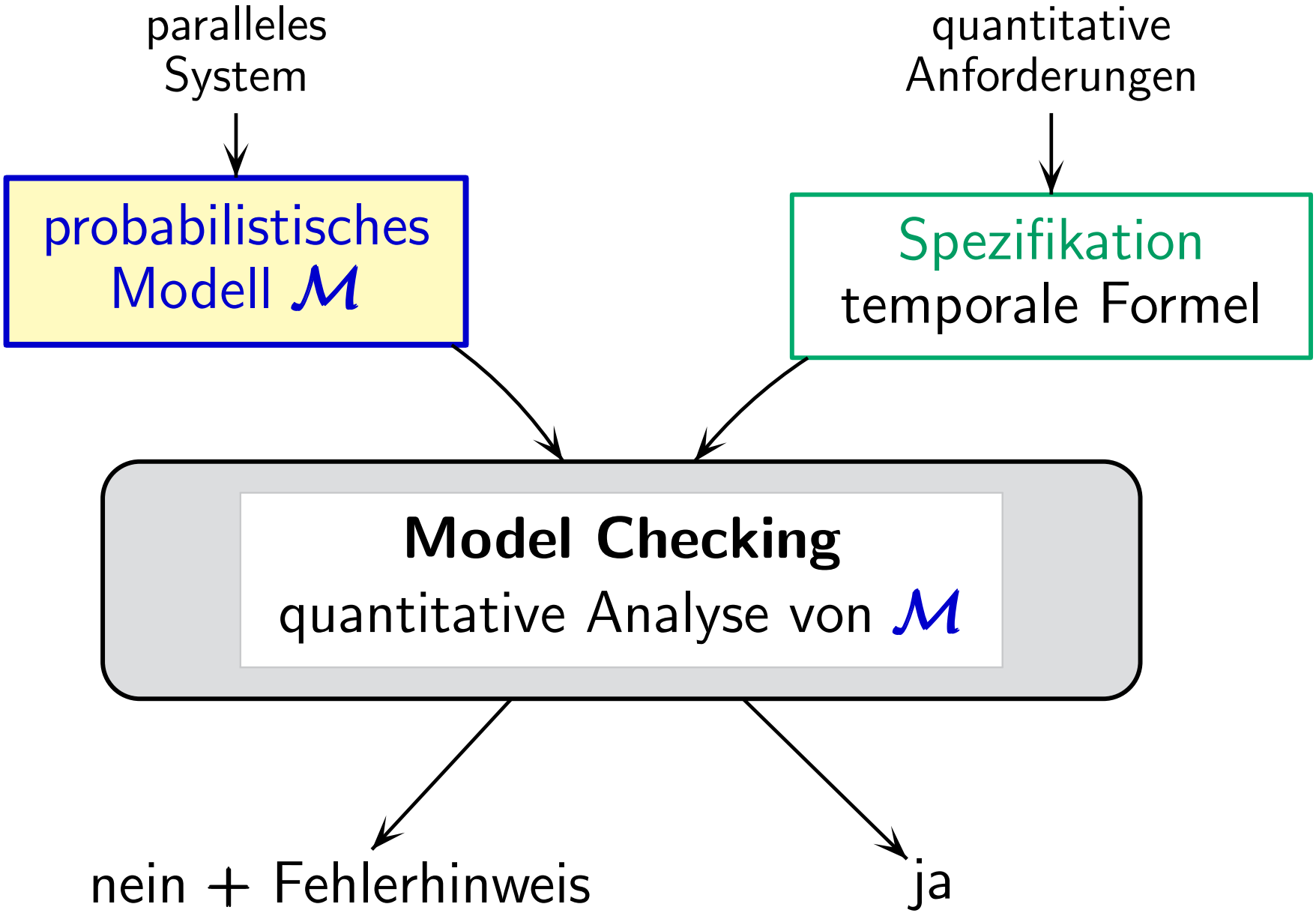
durchschnittlich:

γ Fußgänger-Requests
pro Zeiteinheit

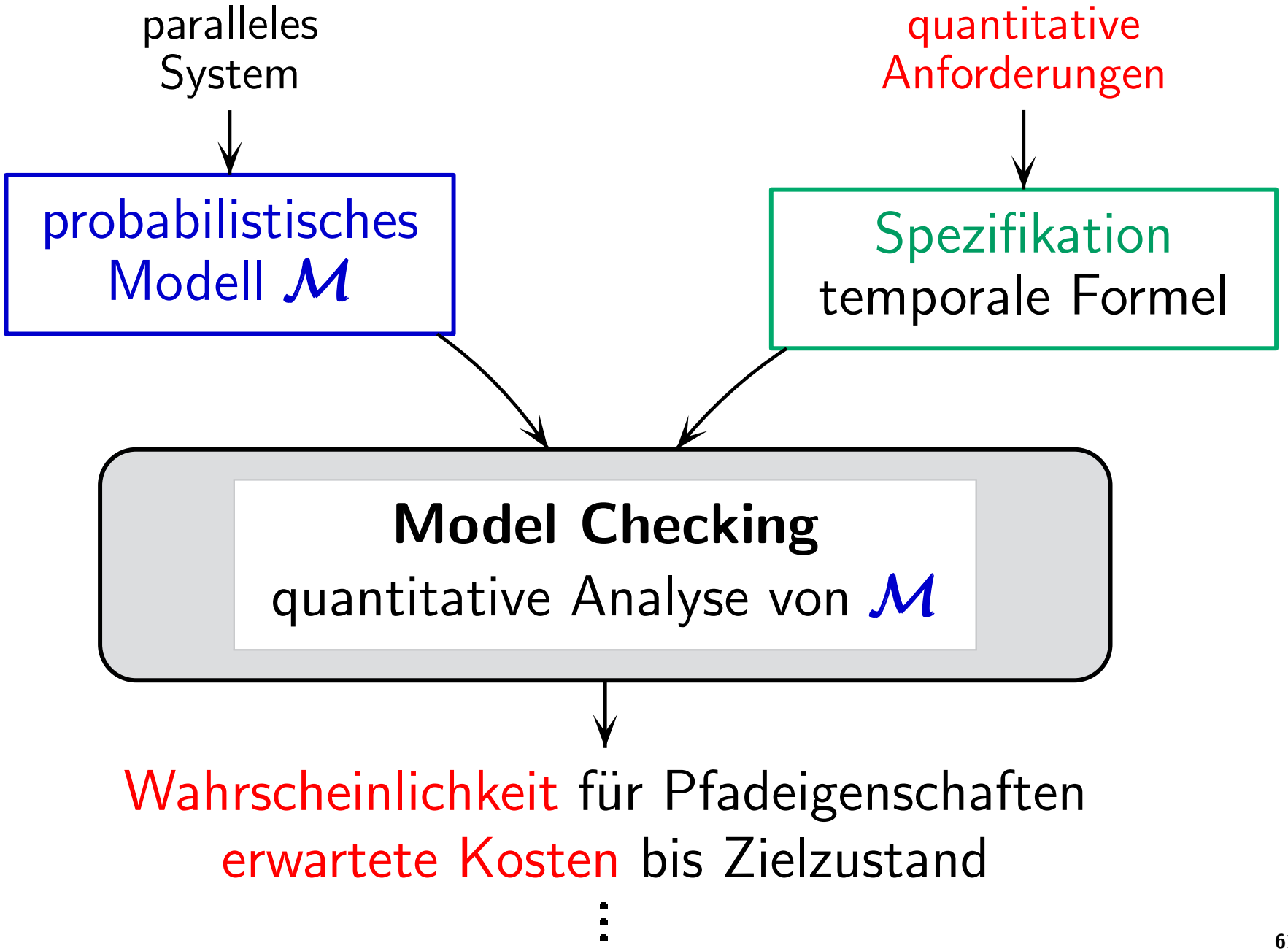
Exponentialverteilung

Probabilistisches Model Checking

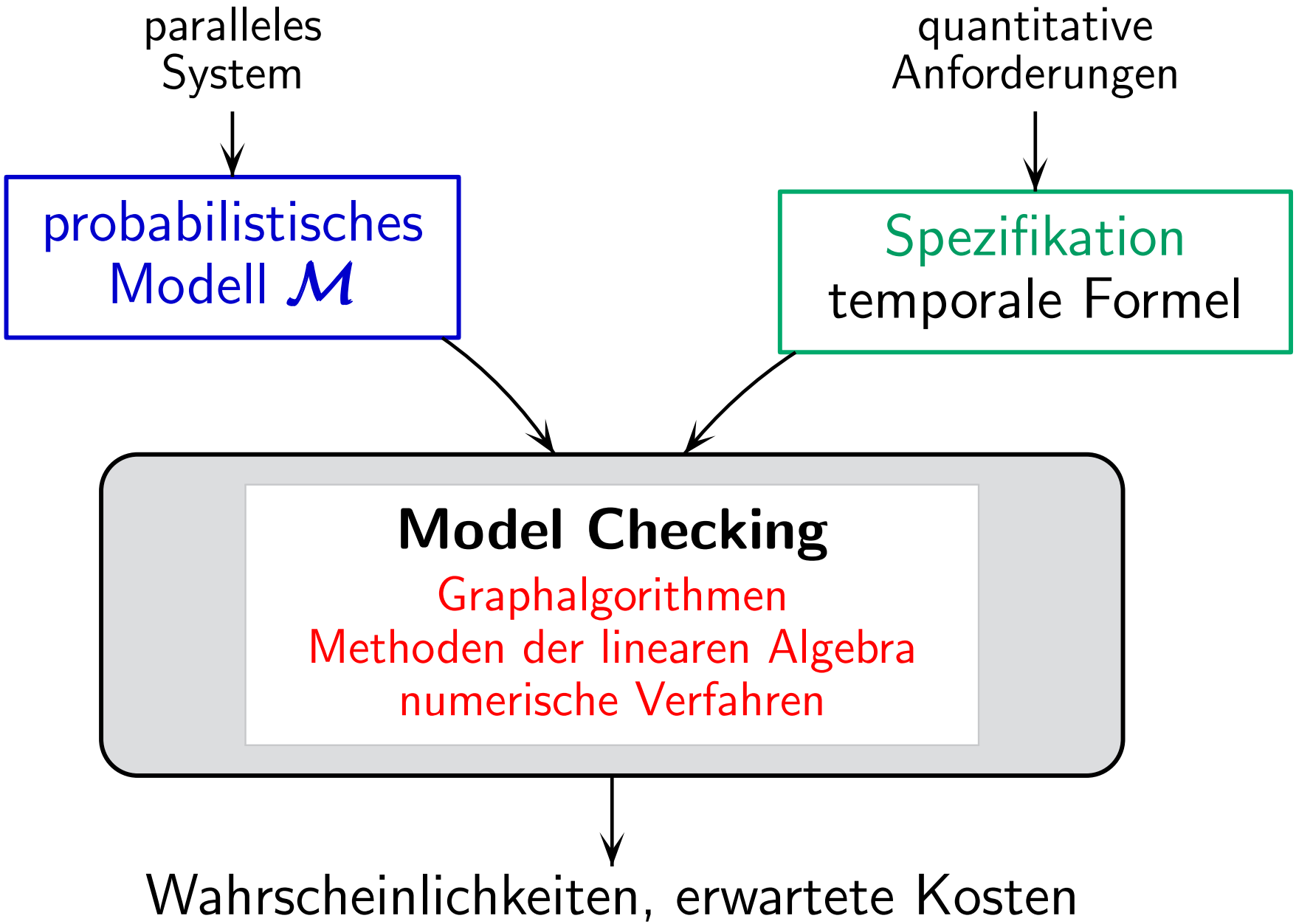
Probabilistisches Model Checking



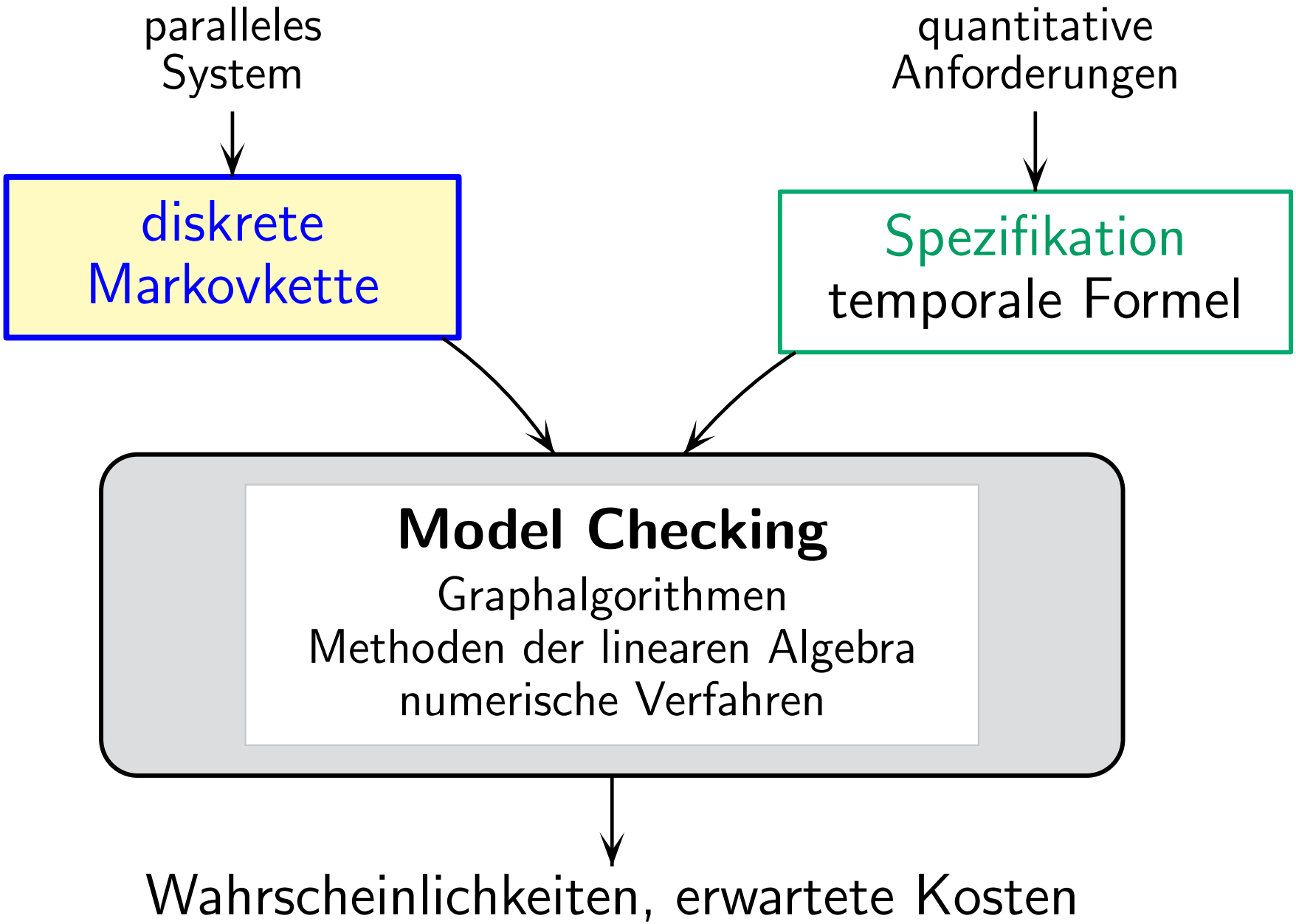
Probabilistisches Model Checking



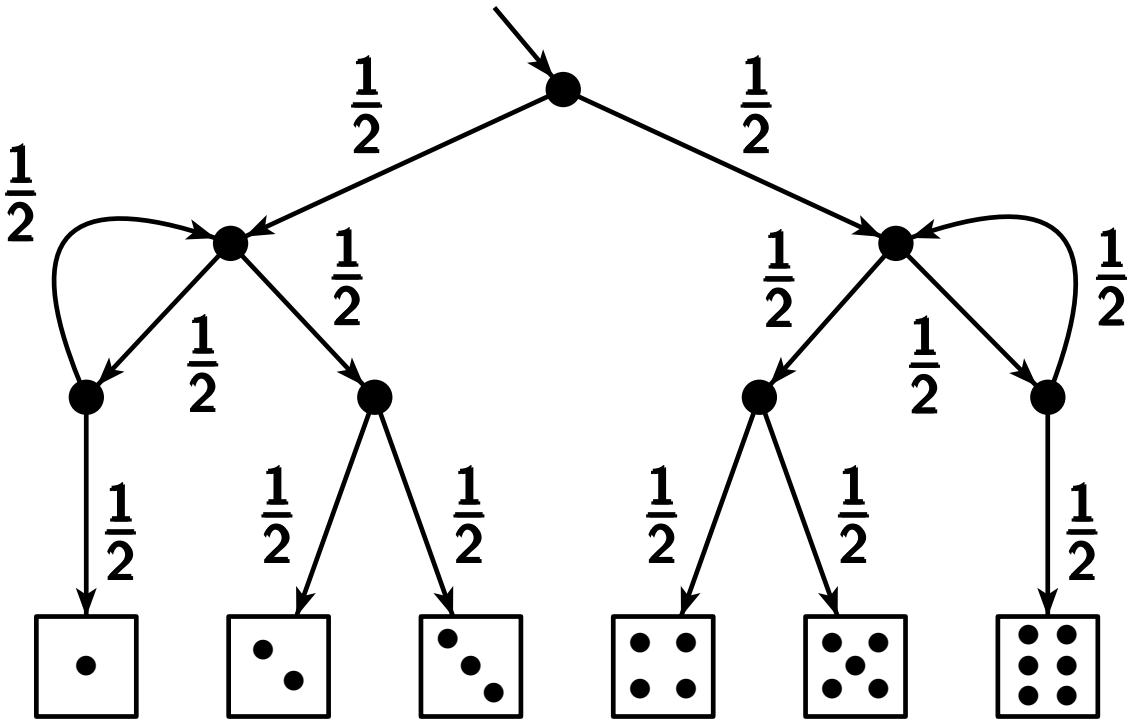
Probabilistisches Model Checking



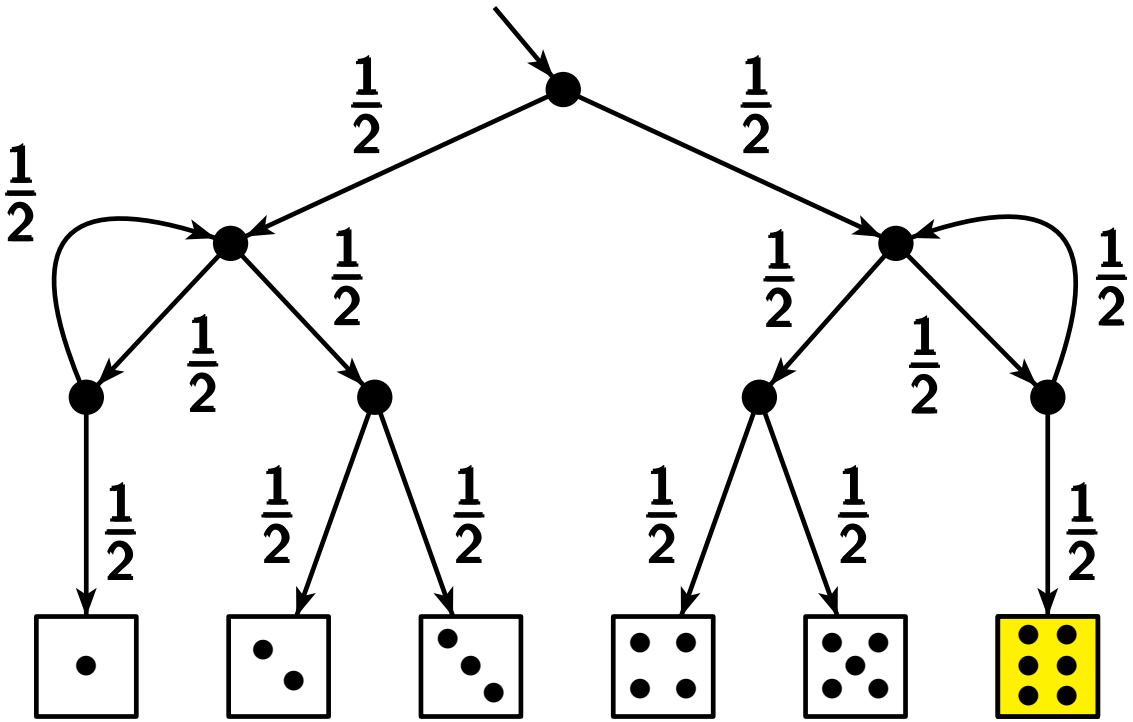
Probabilistisches Model Checking



Simulation eines Würfels durch Münzwurf [Knuth]

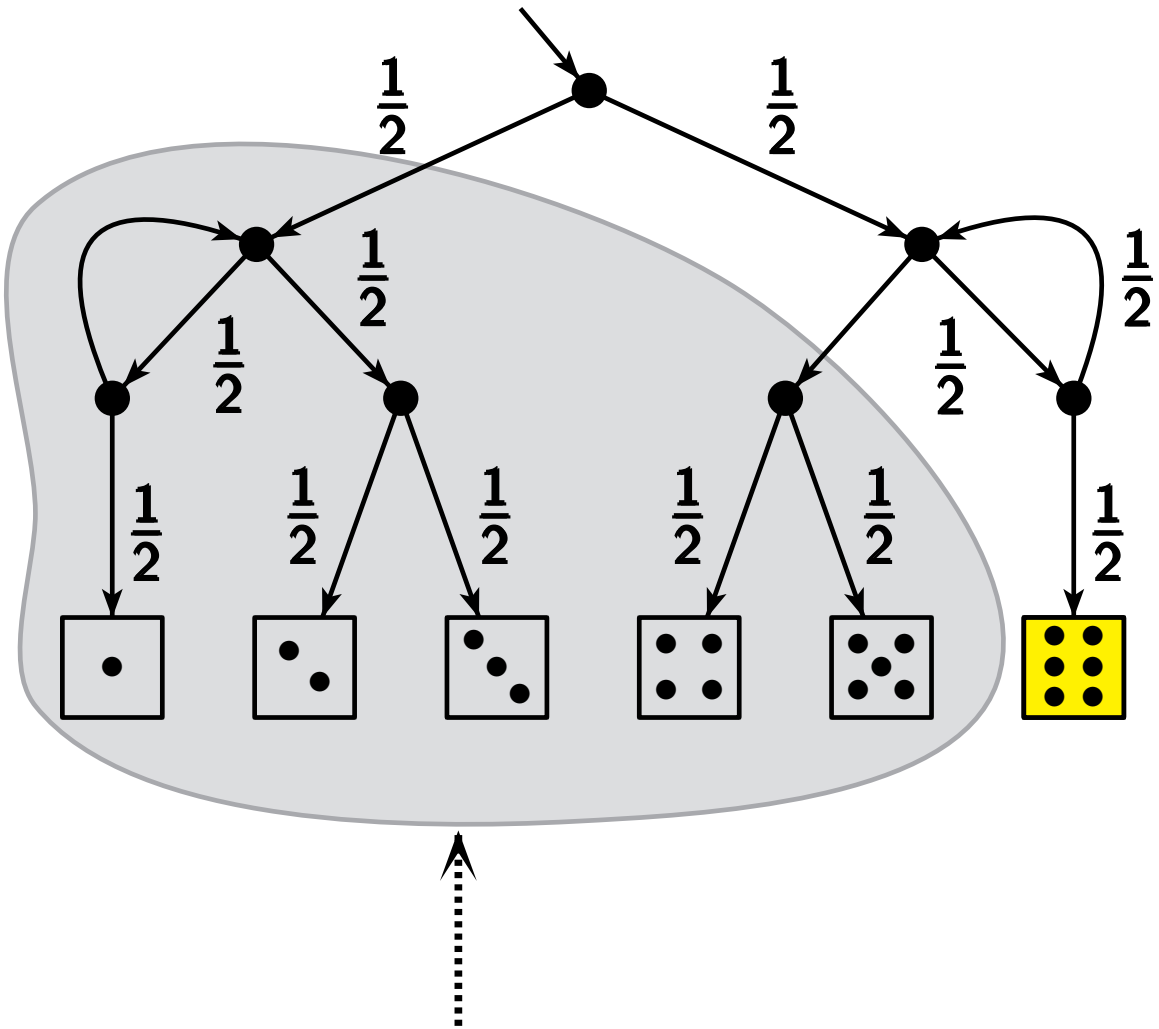


Simulation eines Würfels durch Münzwurf [Knuth]



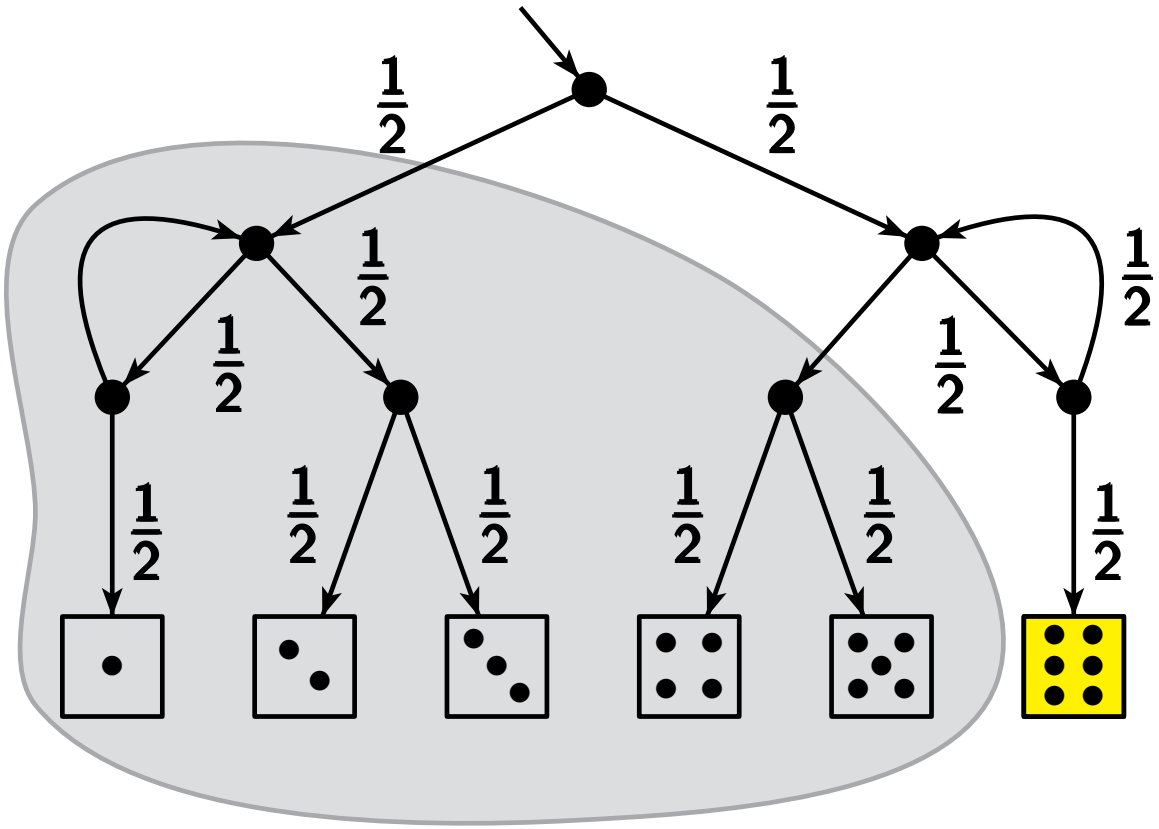
berechne die Wahrscheinlichkeit für
das Würfelergebnis “sechs”

Simulation eines Würfels durch Münzwurf [Knuth]



Würfelergebnis “sechs”
ist nicht erreichbar

Simulation eines Würfels durch Münzwurf [Knuth]

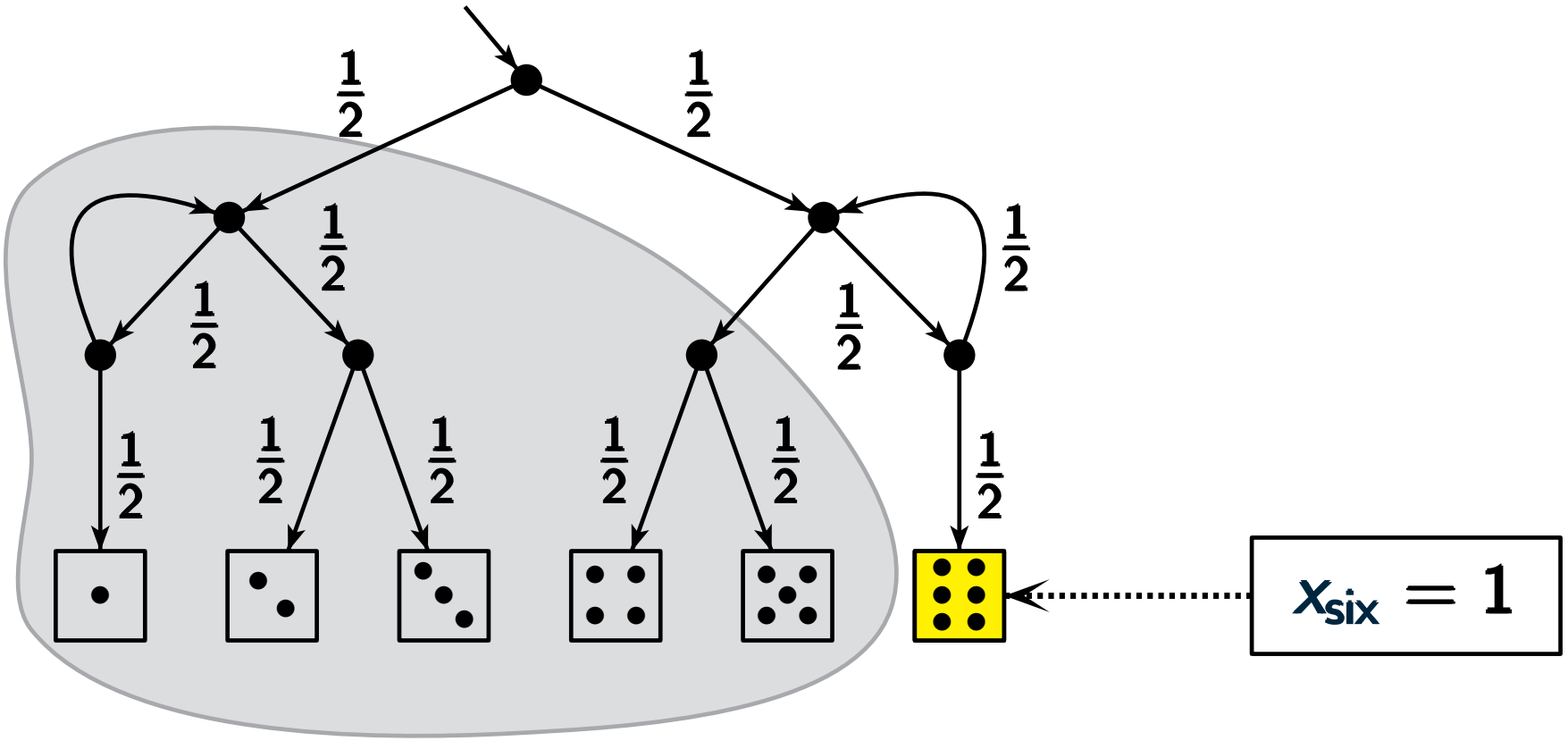


Würfelergebnis "sechs" ist nicht erreichbar

für alle anderen Zustände:

$$x_s = \begin{cases} \text{Wahrscheinlichkeit für} \\ \text{Würfelergebnis "sechs"} \\ \text{ab Zustand } s \end{cases}$$

Simulation eines Würfels durch Münzwurf [Knuth]

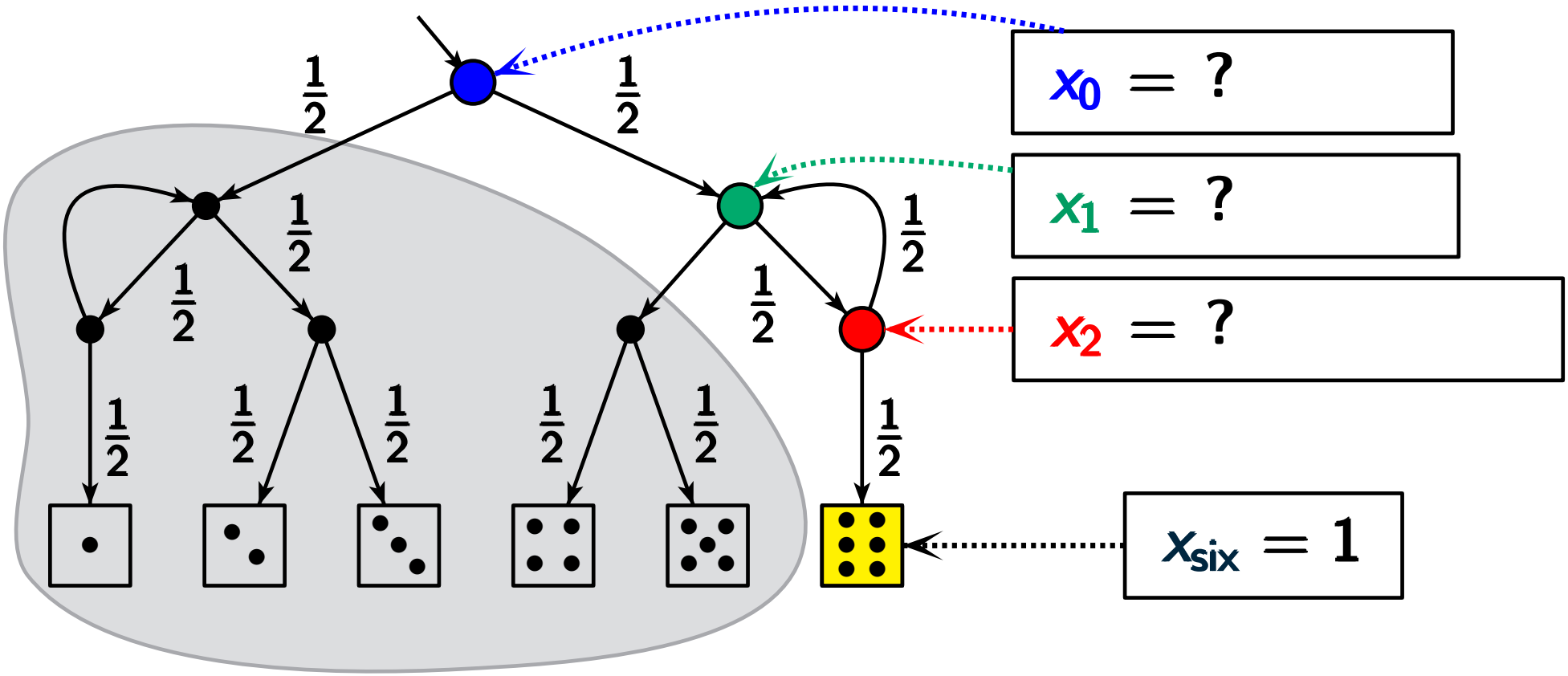


Würfelergebnis “sechs” ist nicht erreichbar

für alle anderen Zustände:

$$x_s = \begin{cases} \text{Wahrscheinlichkeit für} \\ \text{Würfelergebnis “sechs”} \\ \text{ab Zustand } s \end{cases}$$

Simulation eines Würfels durch Münzwurf [Knuth]

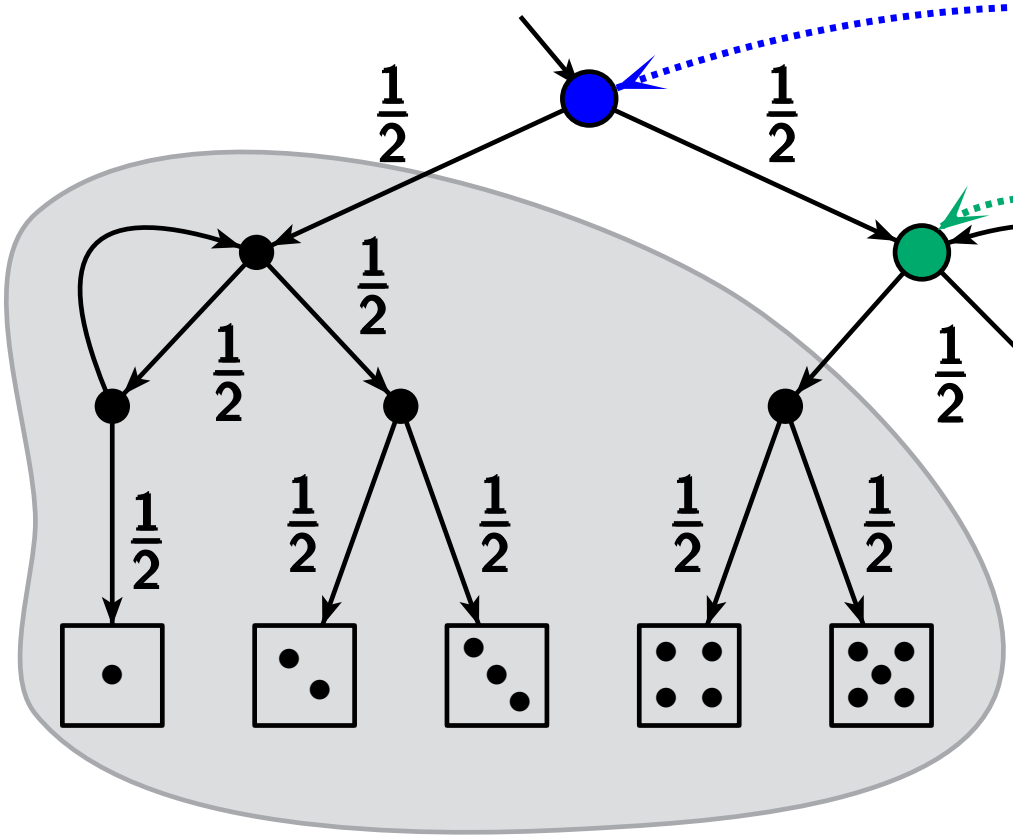


Würfelergebnis "sechs" ist nicht erreichbar

für alle anderen Zustände:

$$x_s = \begin{cases} \text{Wahrscheinlichkeit für} \\ \text{Würfelergebnis "sechs"} \\ \text{ab Zustand s} \end{cases}$$

Simulation eines Würfels durch Münzwurf [Knuth]



$x_0 = \frac{1}{2} x_1$

$x_1 = ?$

$x_2 = ?$

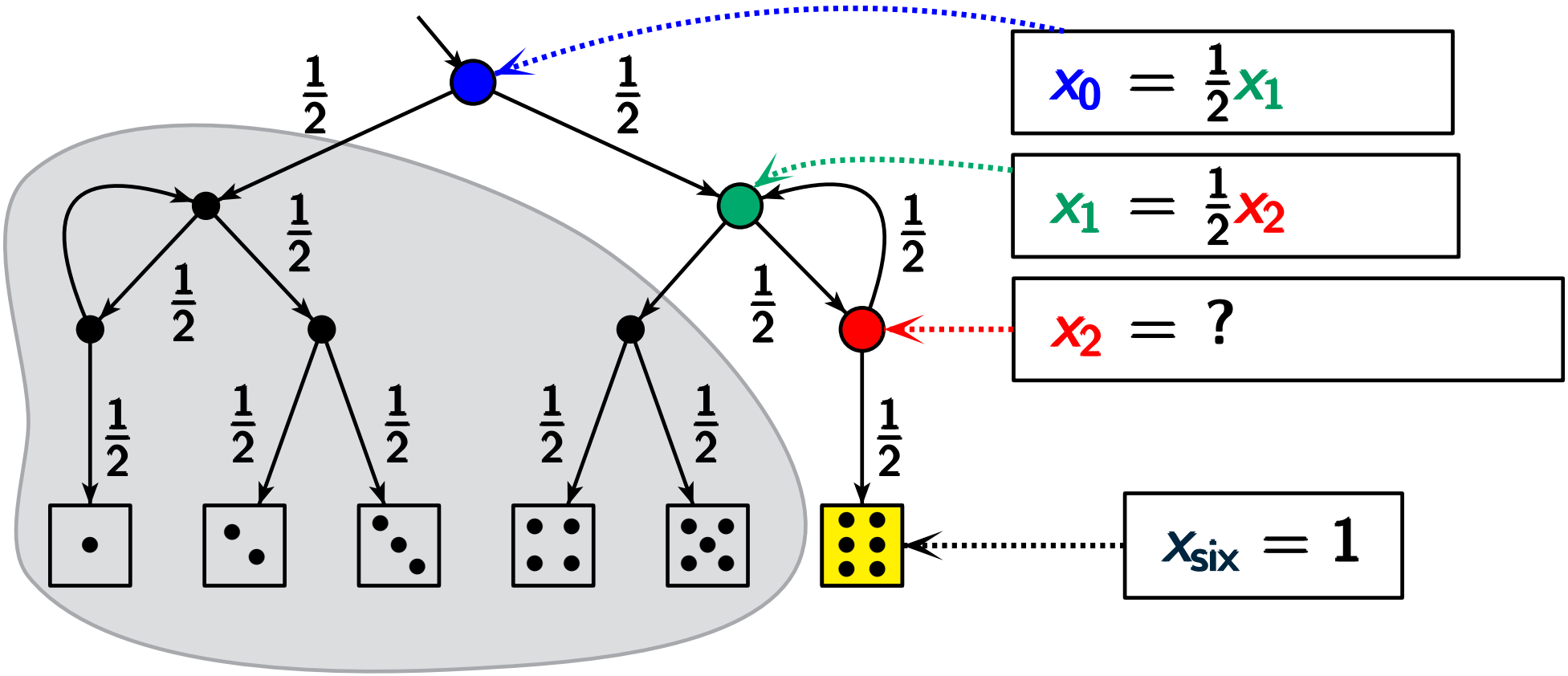
$x_{six} = 1$

Würfelergebnis "sechs" ist nicht erreichbar

für alle anderen Zustände:

$$x_s = \left\{ \begin{array}{l} \text{Wahrscheinlichkeit für} \\ \text{Würfelergebnis "sechs"} \\ \text{ab Zustand s} \end{array} \right.$$

Simulation eines Würfels durch Münzwurf [Knuth]

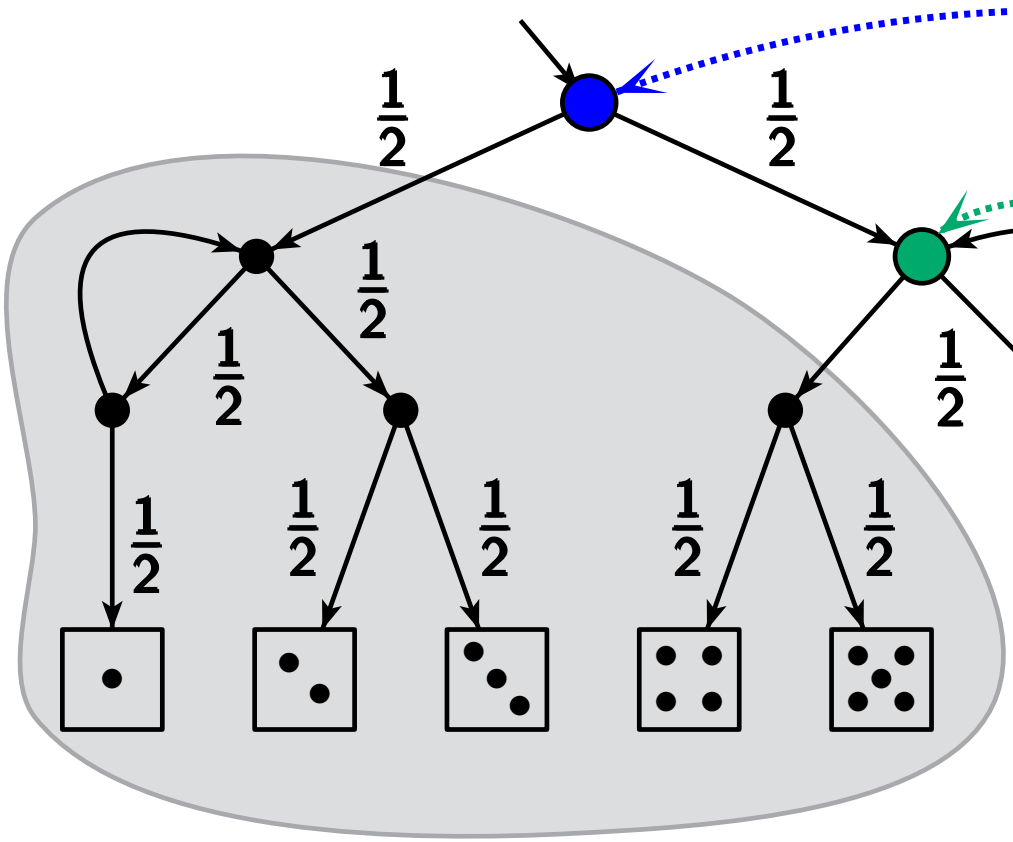


Würfelergebnis "sechs" ist nicht erreichbar

für alle anderen Zustände:

$$x_s = \begin{cases} \text{Wahrscheinlichkeit für} \\ \text{Würfelergebnis "sechs"} \\ \text{ab Zustand s} \end{cases}$$

Simulation eines Würfels durch Münzwurf [Knuth]



$x_0 = \frac{1}{2}x_1$

$x_1 = \frac{1}{2}x_2$

$x_2 = \frac{1}{2}x_1 + \frac{1}{2}$

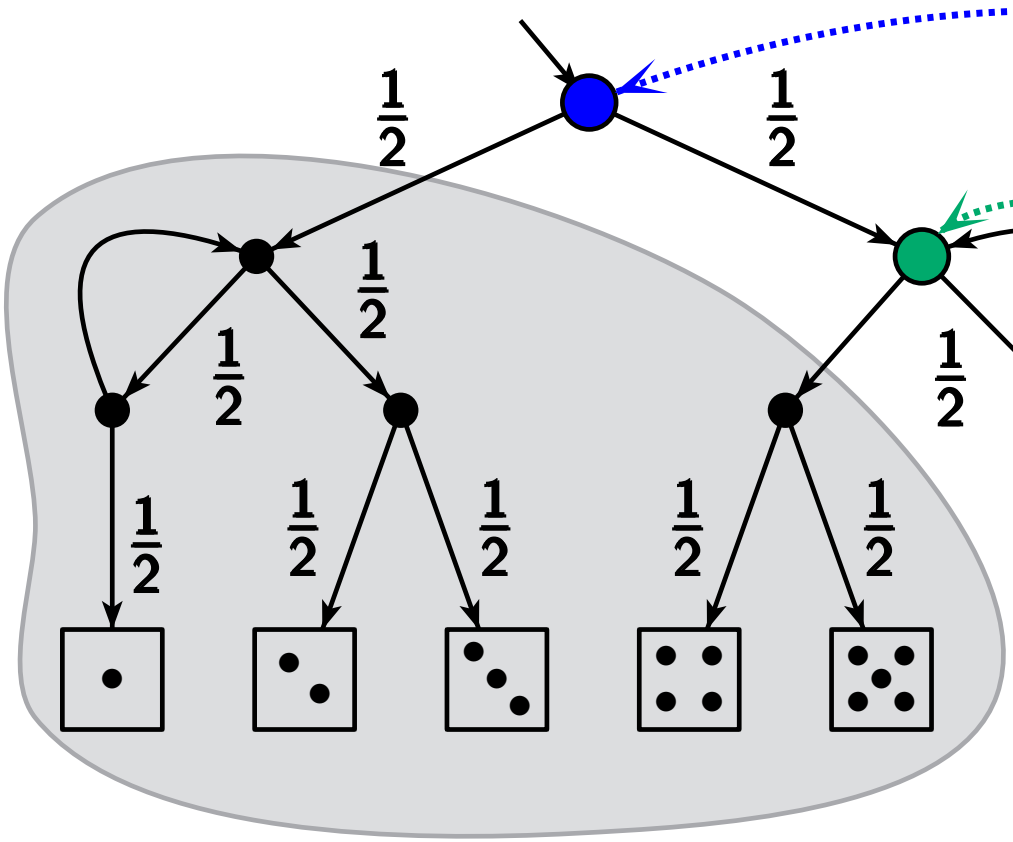
$x_{six} = 1$

Würfelergebnis "sechs" ist nicht erreichbar

für alle anderen Zustände:

$$x_s = \begin{cases} \text{Wahrscheinlichkeit für} \\ \text{Würfelergebnis "sechs"} \\ \text{ab Zustand s} \end{cases}$$

Simulation eines Würfels durch Münzwurf [Knuth]



$$x_0 = \frac{1}{2}x_1$$

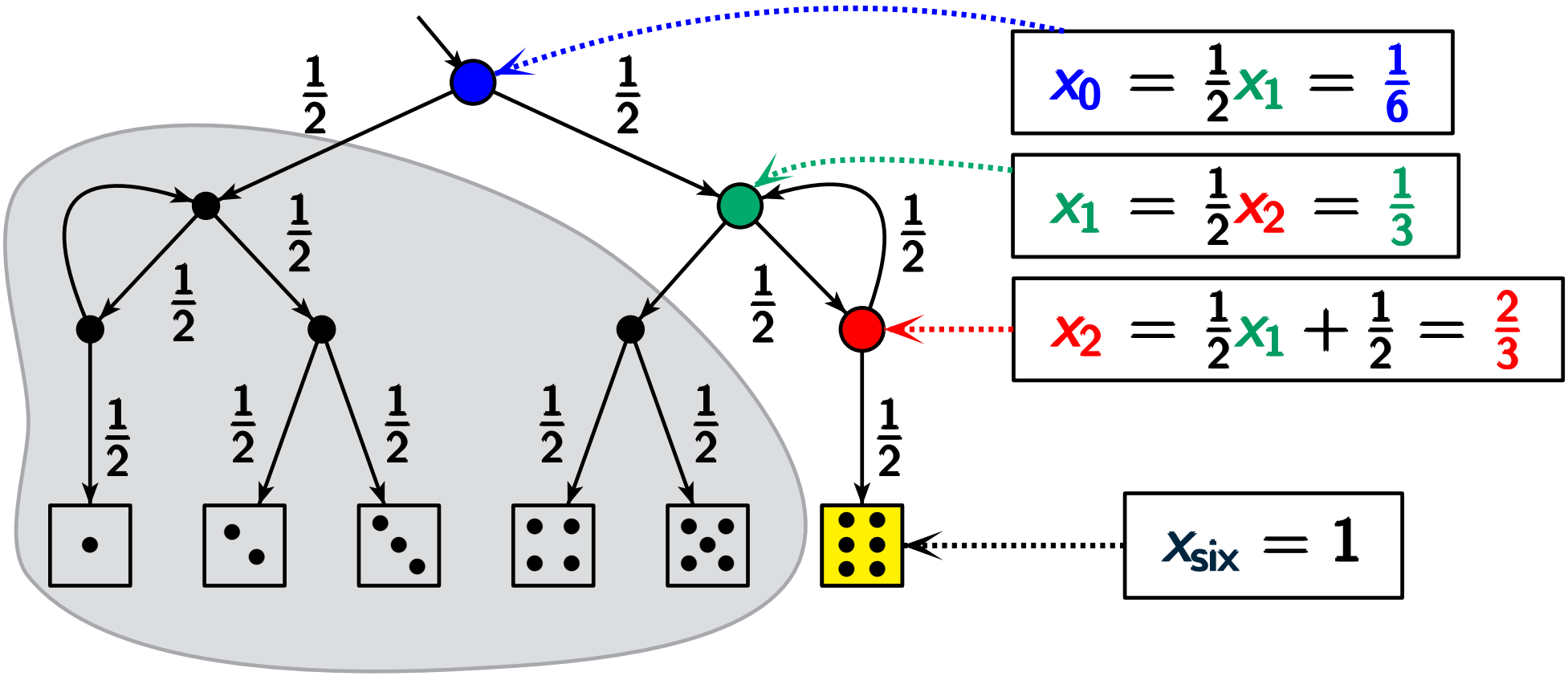
$$x_1 = \frac{1}{2}x_2$$

$$x_2 = \frac{1}{2}x_1 + \frac{1}{2}$$

$$x_{six} = 1$$

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

Simulation eines Würfels durch Münzwurf [Knuth]



Wahrscheinlichkeit für Würfelergebnis "sechs"

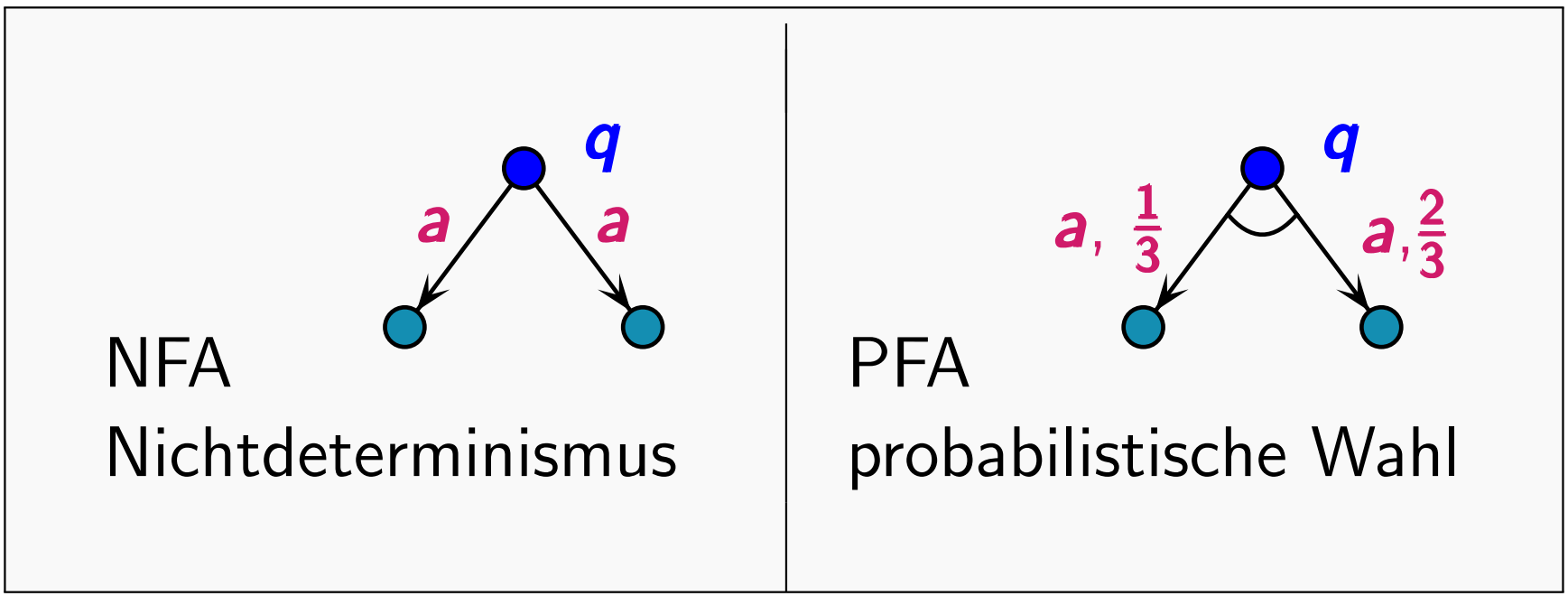
$$x_0 = \frac{1}{6}$$

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

Probabilistic finite automata (PFA)

Probabilistic finite automata (PFA)

PFA wie NFA, aber Auflösung aller nichtdeterministischen Entscheidungen durch Wahrscheinlichkeitsverteilungen



Probabilistic finite automata (PFA)

PFA $\mathcal{M} = (Q, \Sigma, \delta, \mu, F)$

- Q endlicher Zustandsraum
- Σ Alphabet
- Transitionsfunktion $\delta : Q \times \Sigma \times Q \to [0, 1]$, so dass für alle $q \in Q$ und $a \in \Sigma$:

$$\sum_{p \in Q} \delta(q, a, p) \in \{0, 1\}$$

- Anfangsverteilung $\mu : Q \to [0, 1]$
- Endzustandsmenge $F \subseteq Q$

Semantik von PFA

PFA $\mathcal{M} = (Q, \Sigma, \delta, \mu, F)$

Für jedes endliche Wort $x \in \Sigma^*$:

$$\begin{aligned} \Pr(x) &= \text{Akzeptanzwahrscheinlichkeit für } x \\ &= \text{W'keit der akzeptierenden Läufe für } x \end{aligned}$$

Semantik von PFA

PFA $\mathcal{M} = (Q, \Sigma, \delta, \mu, F)$

Für jedes endliche Wort $x \in \Sigma^*$:

$\Pr(x)$ = Akzeptanzwahrscheinlichkeit für x

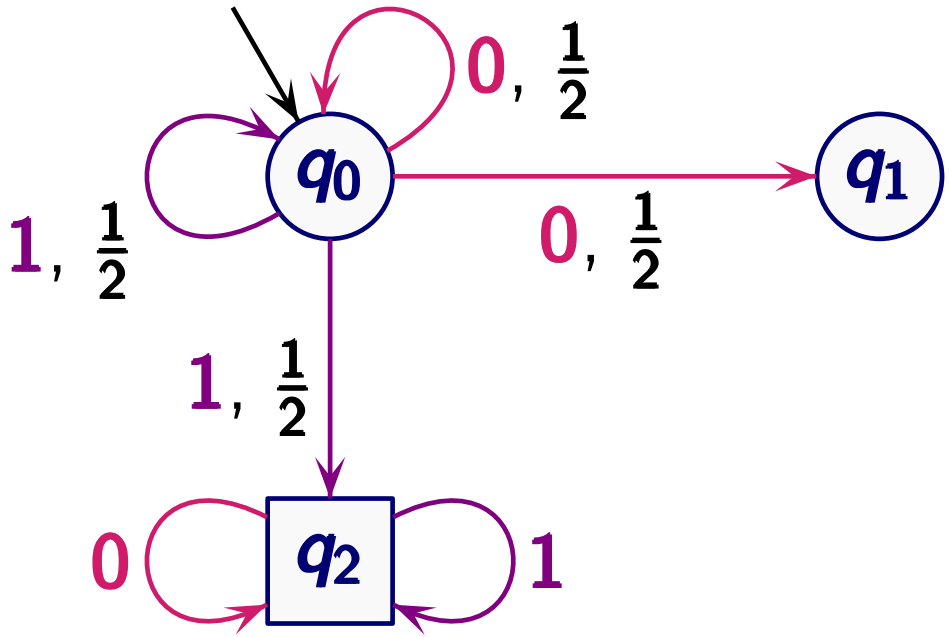
= W'keit der akzeptierenden Läufe für x

akzeptierte Sprache für Schwellwert $\lambda \in]0, 1[$

("threshold semantics")

$$\mathcal{L}_\lambda(\mathcal{M}) = \{ x \in \Sigma^* : \Pr(x) > \lambda \}$$

Beispiel: PFA



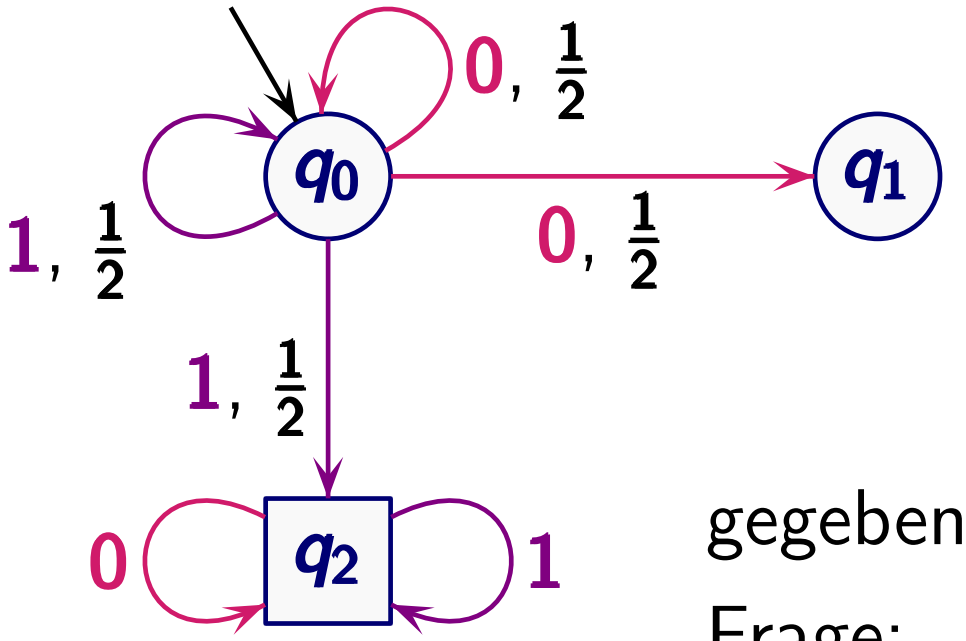
Alphabet $\Sigma = \{0, 1\}$

Anfangszustand q_0 (Anfangsverteilung $\mu(q_0) = 1$)

Endzustand q_2

Transitionen $q_2 \xrightarrow{0} q_2$ und $q_2 \xrightarrow{0} q_2$ mit W'keit 1

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

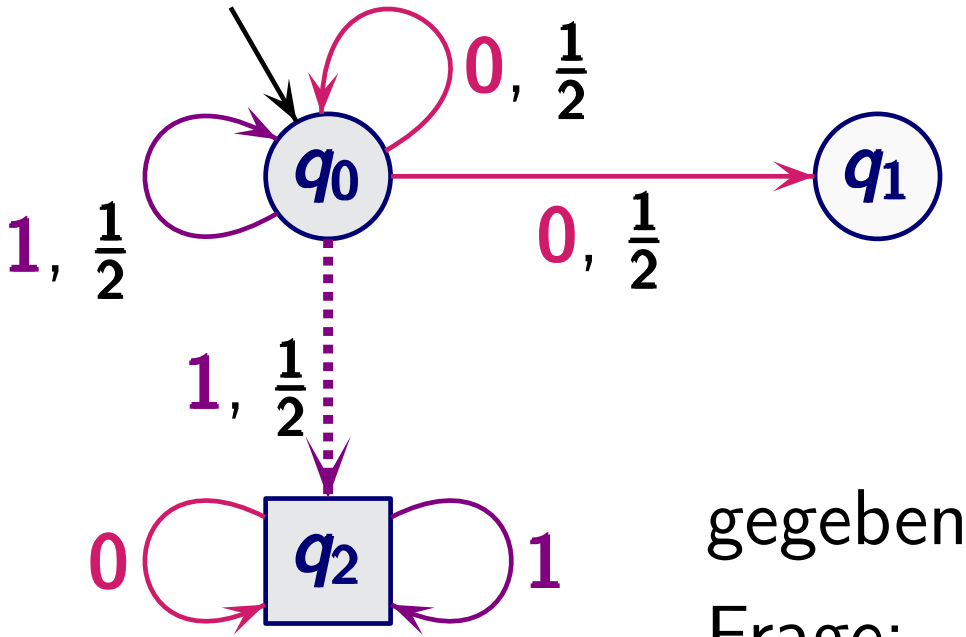
Frage: was ist $\Pr(x)$?

Anfangszustand q_0 (Anfangsverteilung $\mu(q_0) = 1$)

Endzustand q_2

Transitionen $q_2 \xrightarrow{0} q_2$ und $q_2 \xrightarrow{0} q_2$ mit W'keit 1

Beispiel: PFA



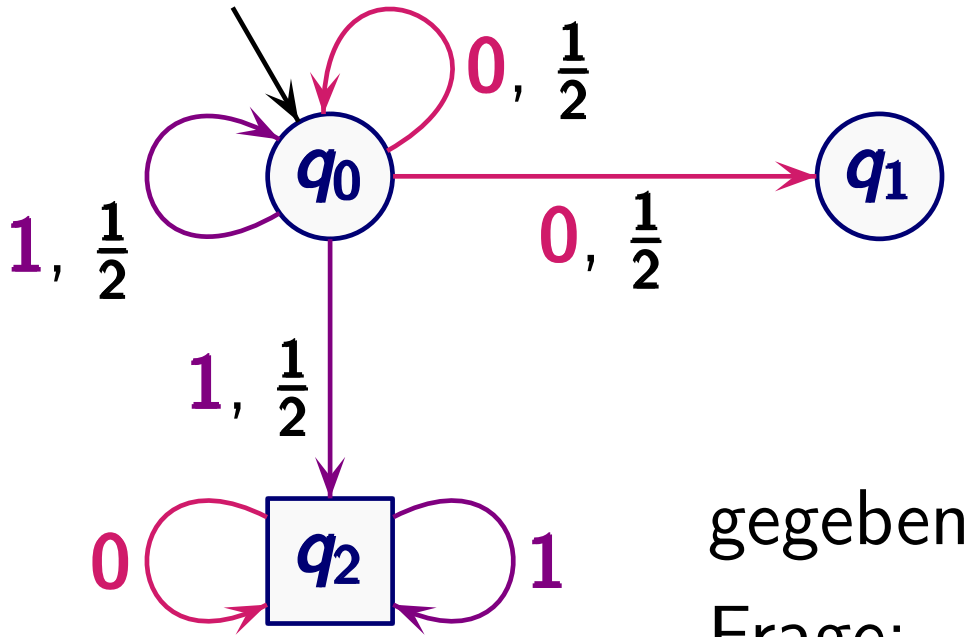
Alphabet $\Sigma = \{0, 1\}$

gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

Frage: was ist $\Pr(x)$?

$$\Pr(1) = \frac{1}{2}$$

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

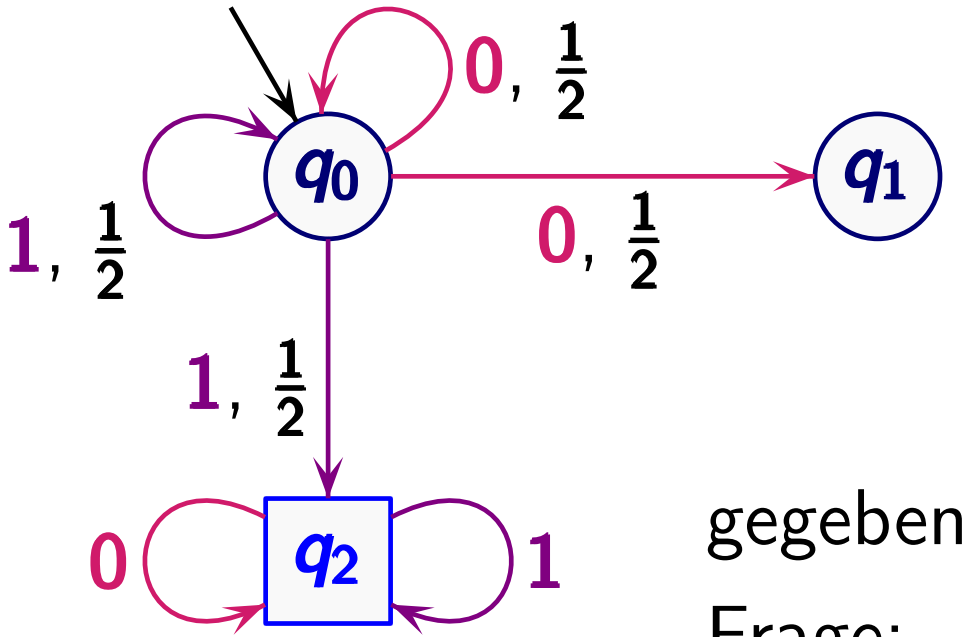
gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

Frage: was ist $\Pr(x)$?

$$\Pr(1) = \frac{1}{2}$$

$$\Pr(1\,0\,1) = ?$$

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

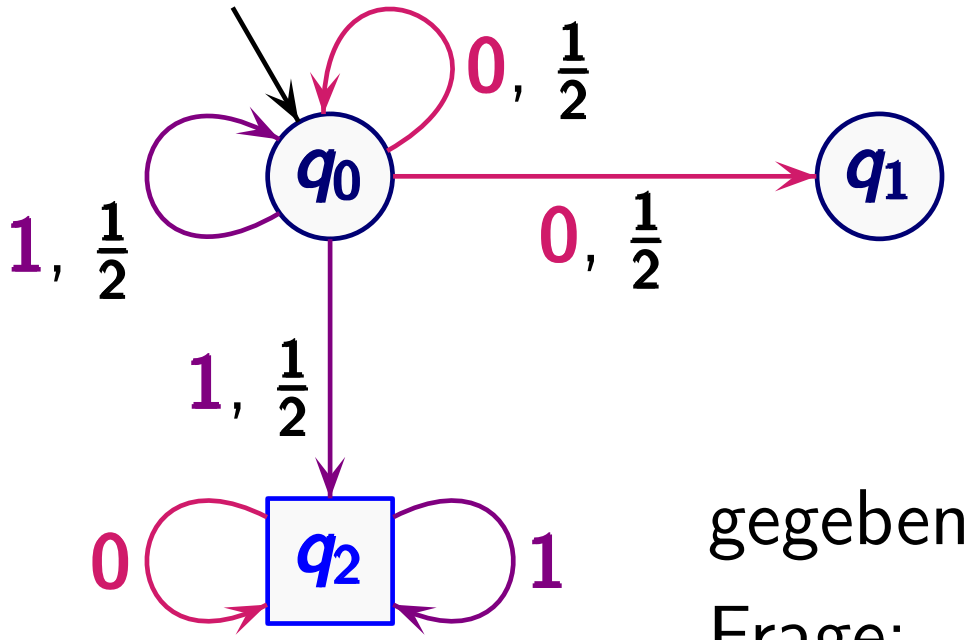
Frage: was ist $\Pr(x)$?

$$\Pr(1) = \frac{1}{2}$$

$$\Pr(1\,0\,1) = \frac{1}{2} + \ldots$$

Lauf $q_0 \xrightarrow{1} q_2 \xrightarrow{0} q_2 \xrightarrow{1} q_2$

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

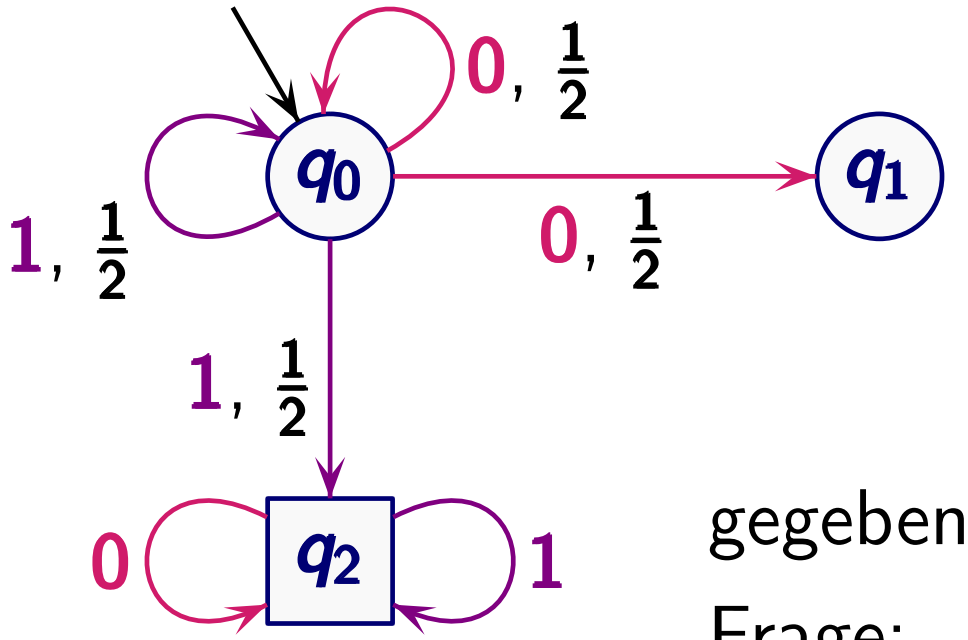
Frage: was ist $\Pr(x)$?

$$\Pr(1) = \frac{1}{2}$$

$$\Pr(101) = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2}$$

Lauf $q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_2$

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

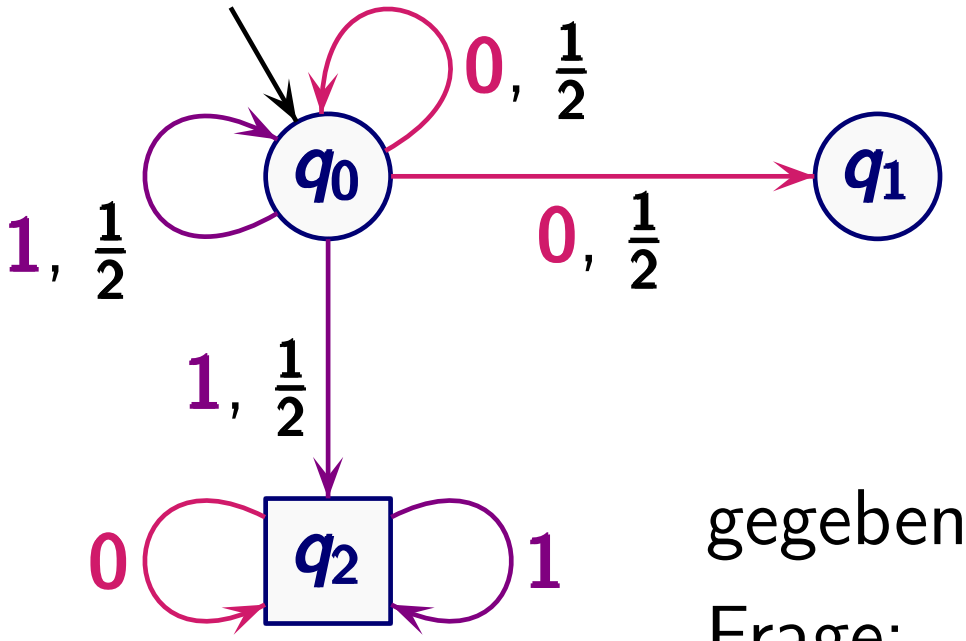
Frage: was ist $\Pr(x)$?

$$\Pr(1) = \frac{1}{2}$$

$$\Pr(101) = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} + \frac{1}{8}$$

$$\Pr(1101) = \frac{1}{2} + \frac{1}{2} \cdot \Pr(101) = \frac{1}{2} + \frac{1}{4} + \frac{1}{16}$$

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

gegeben: $x = a_1\, a_2 \ldots a_n \in \{0,1\}^+$

Frage: was ist $\Pr(x)$?

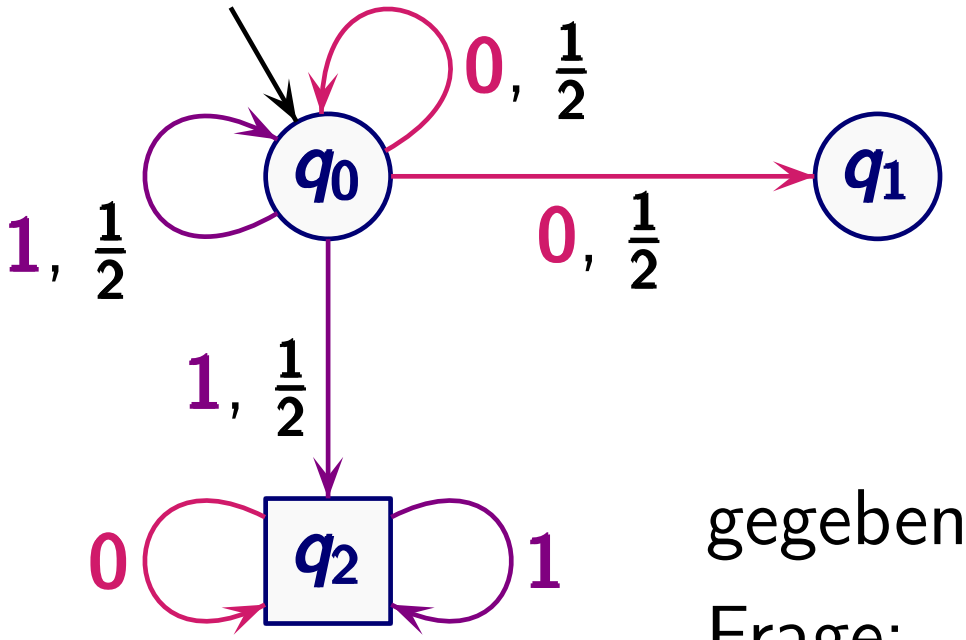
Antwort: $\Pr(x) = bin(x) = \sum_{i=1}^{n} a_i\, 2^{-i}$

$$\Pr(1) = \tfrac{1}{2}$$

$$\Pr(101) = \tfrac{1}{2} + \tfrac{1}{2}\cdot\tfrac{1}{2}\cdot\tfrac{1}{2} = \tfrac{1}{2} + \tfrac{1}{8}$$

$$\Pr(1101) = \tfrac{1}{2} + \tfrac{1}{2}\cdot\Pr(101) = \tfrac{1}{2} + \tfrac{1}{4} + \tfrac{1}{16}$$

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

Schwellwert $\lambda \in]0, 1[$

gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

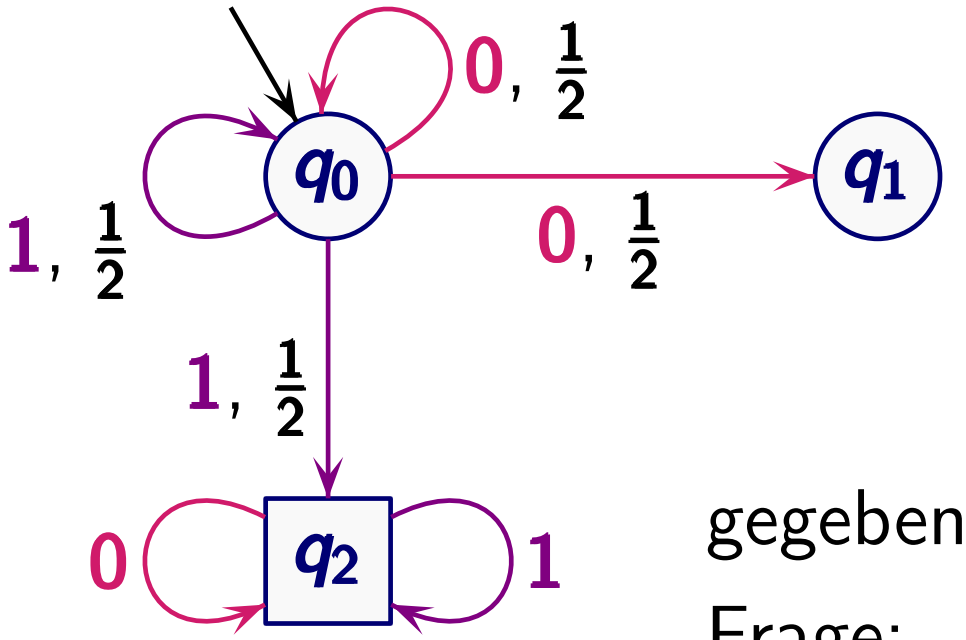
Frage: was ist $\Pr(x)$?

Antwort: $\Pr(x) = bin(x) = \sum_{i=1}^{n} a_i\, 2^{-i}$

akzeptierte Sprache:

$$\mathcal{L}_\lambda(\mathcal{M}) = \{\, x \in \{0, 1\}^+ : bin(x) > \lambda \,\}$$

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

Schwellwert $\lambda \in]0, 1[$

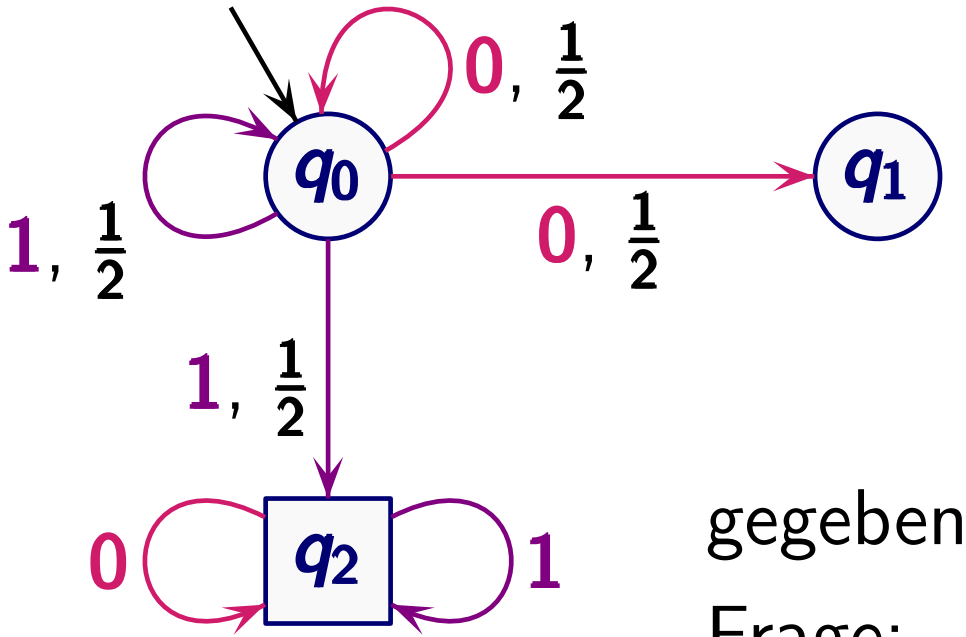
gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

Frage: was ist $\Pr(x)$?

Antwort: $\Pr(x) = bin(x) = \sum_{i=1}^{n} a_i\, 2^{-i}$

Fazit: Die Sprachen $\mathcal{L}_\lambda(\mathcal{M})$ sind paarweise verschieden

Beispiel: PFA



Alphabet $\Sigma = \{0, 1\}$

Schwellwert $\lambda \in]0, 1[$

gegeben: $x = a_1\, a_2 \ldots a_n \in \{0, 1\}^+$

Frage: was ist $\Pr(x)$?

Antwort: $\Pr(x) = bin(x) = \sum_{i=1}^{n} a_i\, 2^{-i}$

Fazit: Die Sprachen $\mathcal{L}_\lambda(\mathcal{M})$ sind paarweise verschieden und manche Sprachen $\mathcal{L}_\lambda(\mathcal{M})$ sind nicht regulär.

Probabilistisches Model Checking (PMC)

- aktuelles Forschungsgebiet seit ca. 20 Jahren
- Tools: PRISM (Oxford/Birm.), Storm (Aachen), ...

Probabilistisches Model Checking (PMC)

- aktuelles Forschungsgebiet seit ca. 20 Jahren
- Tools: PRISM (Oxford/Birm.), Storm (Aachen), ...
- zahlreiche Anwendungsgebiete
  - Koordinationsalgorithmen für verteilte Systeme
  - Kommunikations-, Multimediaprotokolle
  - biologische Systeme, fehlertolerante Systeme, u.v.m.

Probabilistisches Model Checking (PMC)

- aktuelles Forschungsgebiet seit ca. 20 Jahren
- Tools: PRISM (Oxford/Birm.), Storm (Aachen), ...
- zahlreiche Anwendungsgebiete
  - Koordinationsalgorithmen für verteilte Systeme
  - Kommunikations-, Multimediaprotokolle
  - biologische Systeme, fehlertolerante Systeme, u.v.m.

zentrales Forschungsthema unserer Gruppe:

multi-kriterielle Systemanalyse mittels PMC

- Modellierungskonzepte, Logiken, Analysealgorithmen, ...
- inspiriert aus Kooperationen in Forschungsprojekten

Probabilistisches Model Checking (PMC)

- aktuelles Forschungsgebiet seit ca. 20 Jahren
- Tools: PRISM (Oxford/Birm.), Storm (Aachen), ...
- zahlreiche Anwendungsgebiete
  - Koordinationsalgorithmen für verteilte Systeme
  - Kommunikations-, Multimediaprotokolle
  - biologische Systeme, fehlertolerante Systeme, u.v.m.

Zweiter Teil: PMC-basierte Analyse von Demand-Response Protokollen in modernen Stromnetzen

Model Checking

Technische Universität Dresden

Institut für Theoretische Informatik

Algebraische und Logische Grundlagen der Informatik

Christel Baier Philipp Chrszon Clemens Dubslaff
Florian Funke Simon Jantsch Sascha Klüppelholz
David Müller Jakob Piribauer Sascha Wunderlich

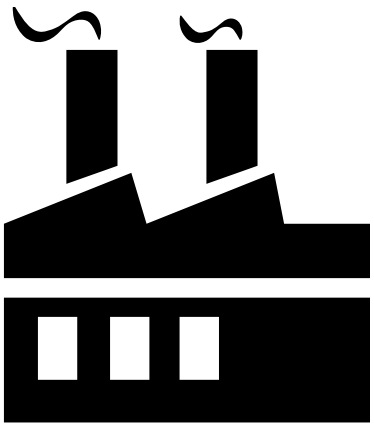
Formal Analysis of Demand Response

What is Demand Response and why is it useful?

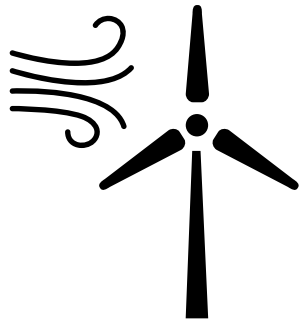
Why and how do we formally analyse it?

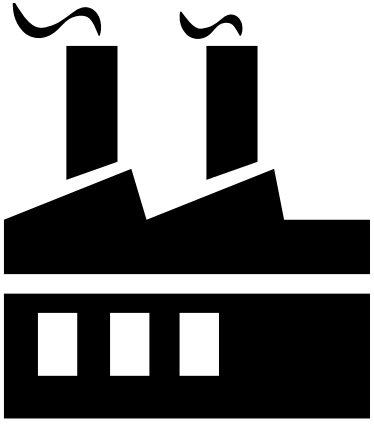
What can we find out? **Demo**

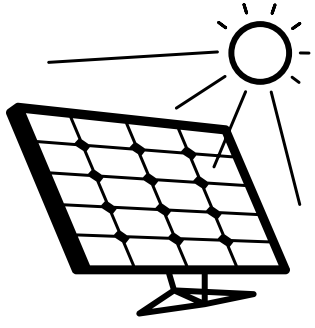
Trends in Energy Supply



Trends in Energy Supply

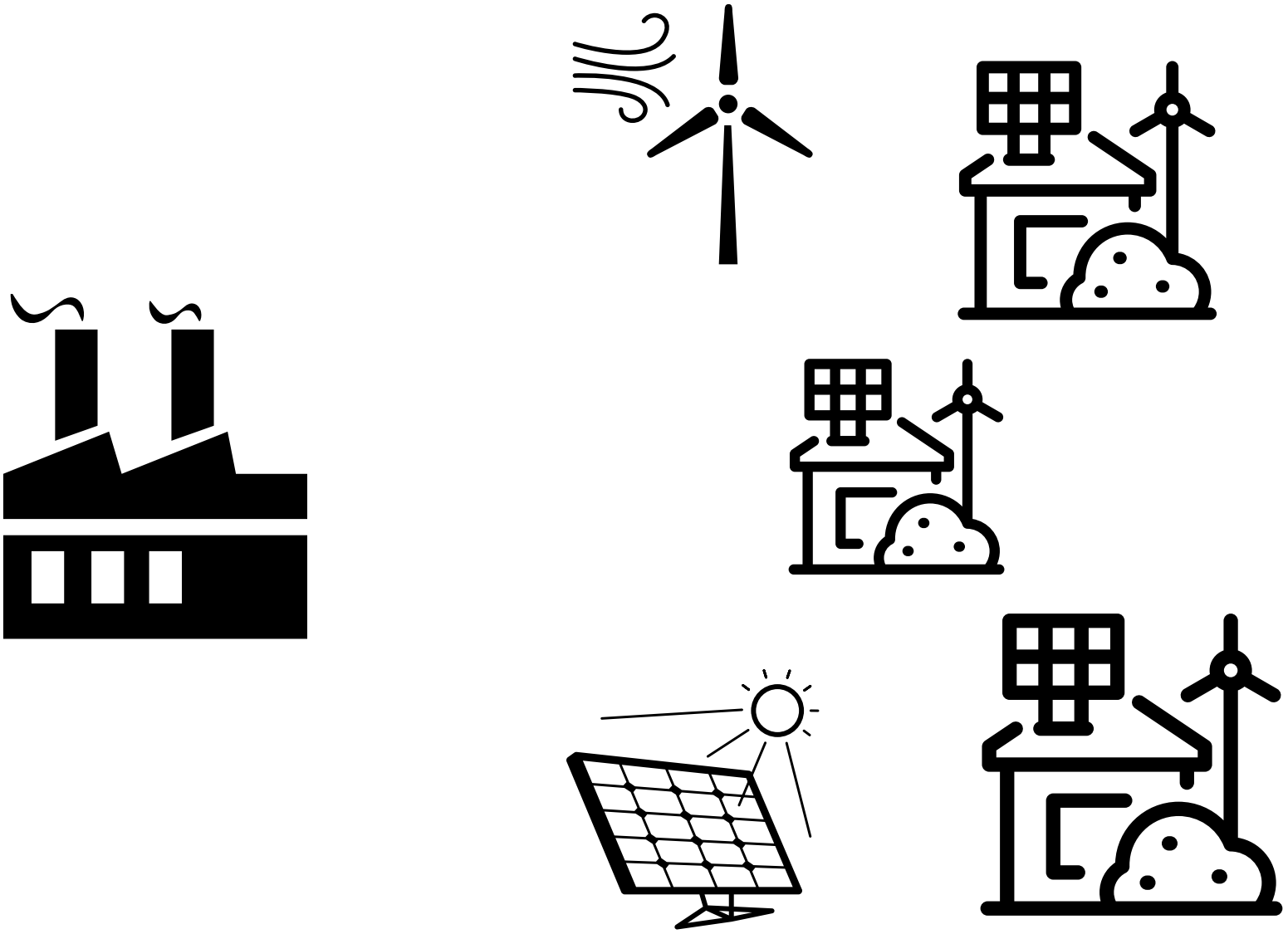






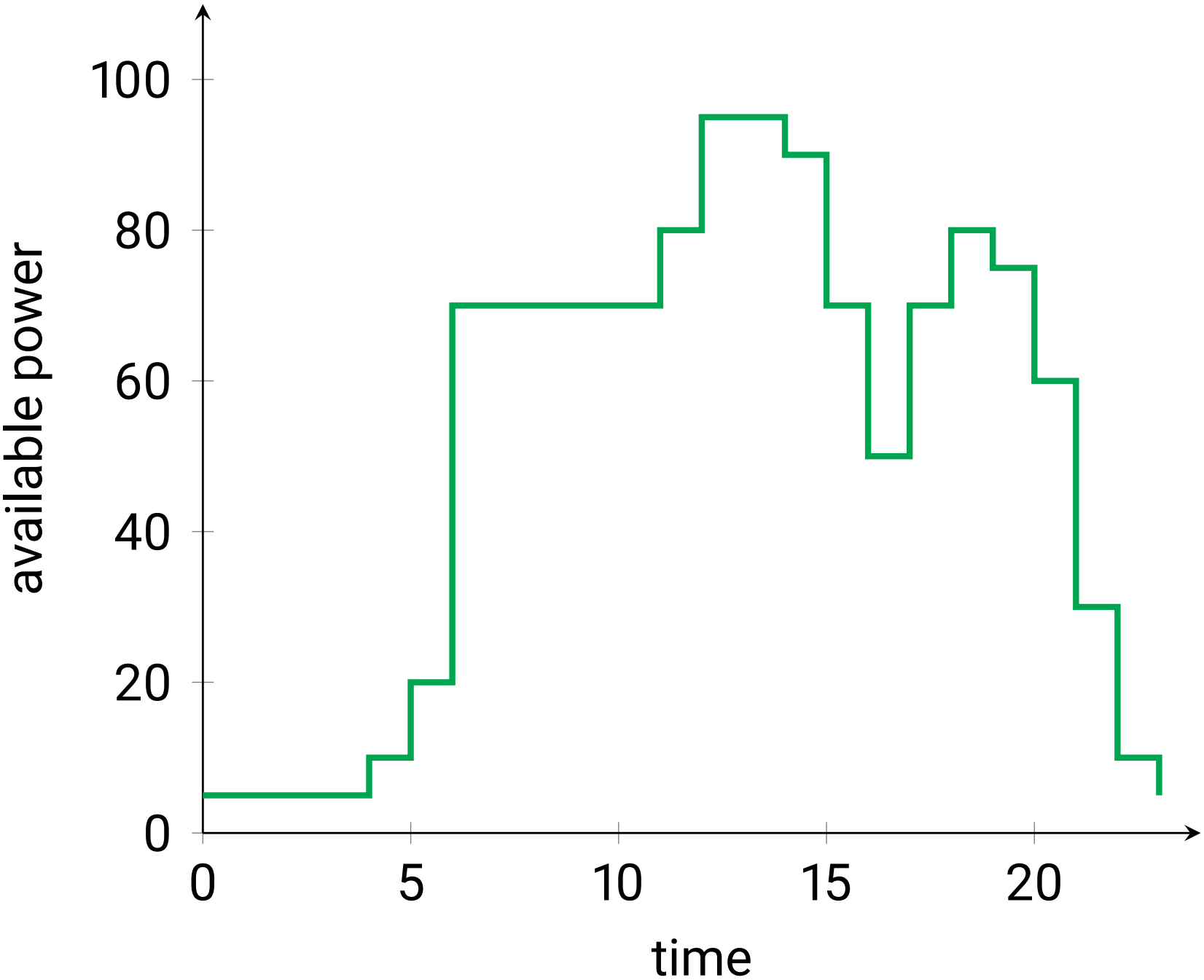
decarbonization

Trends in Energy Supply

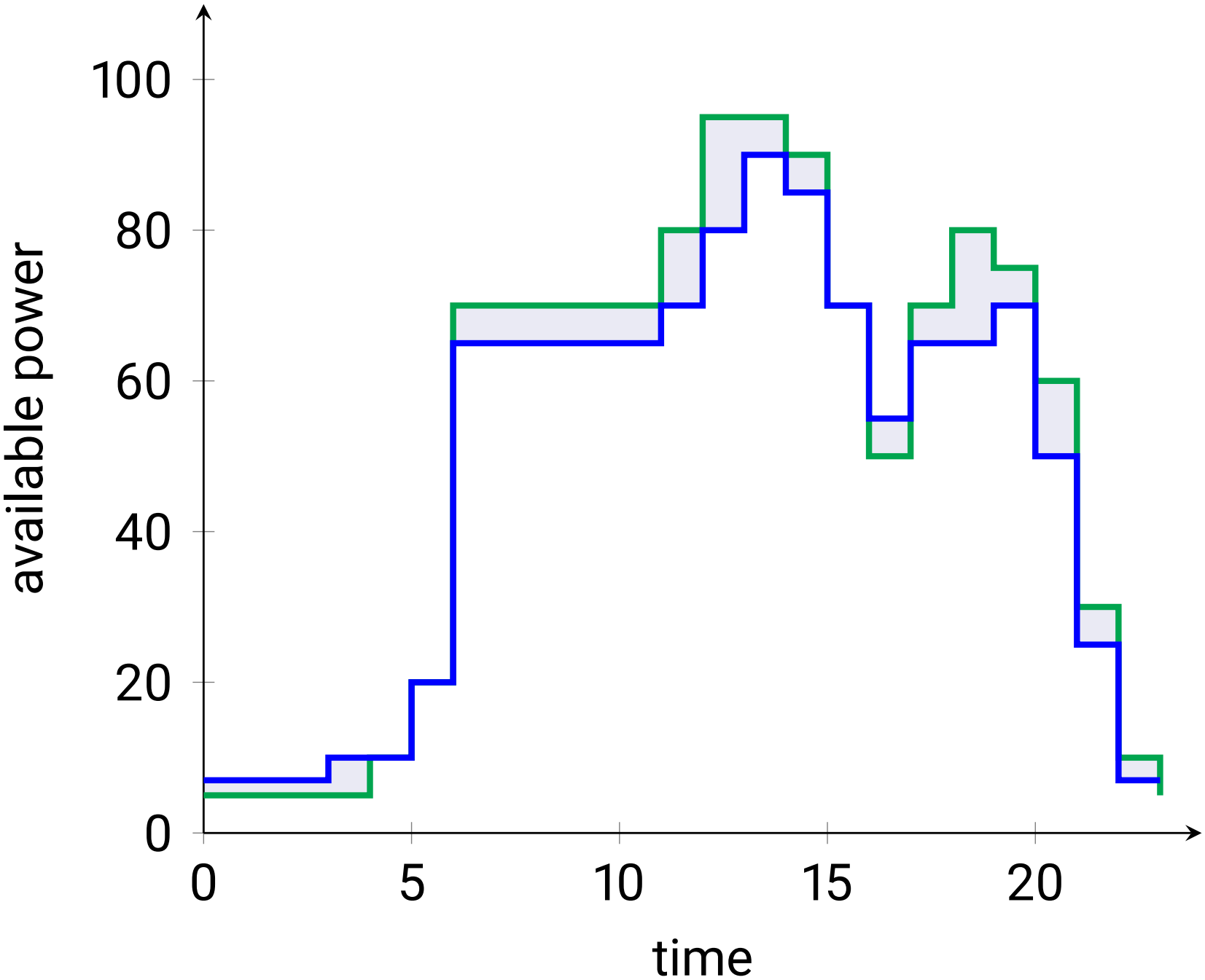


decarbonization and decentralization

Fluctuations in the Grid



Fluctuations in the Grid

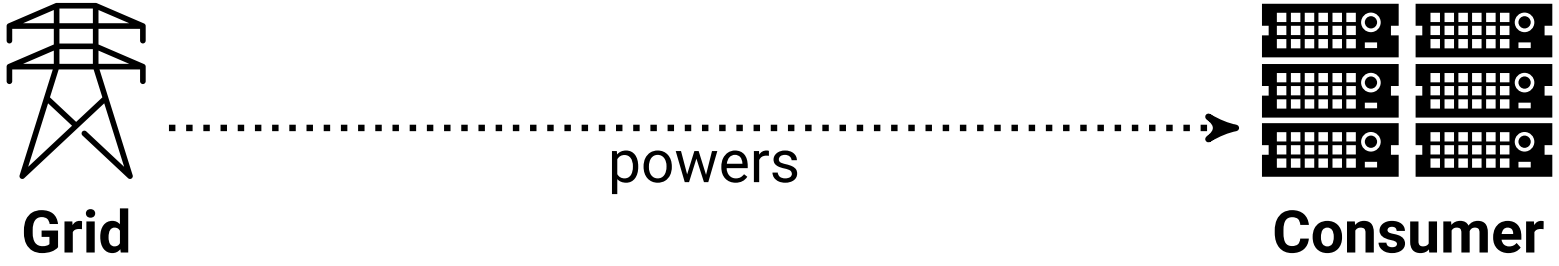


Demand Response

... a class of protocols to mitigate **fluctuations** in the power grid.

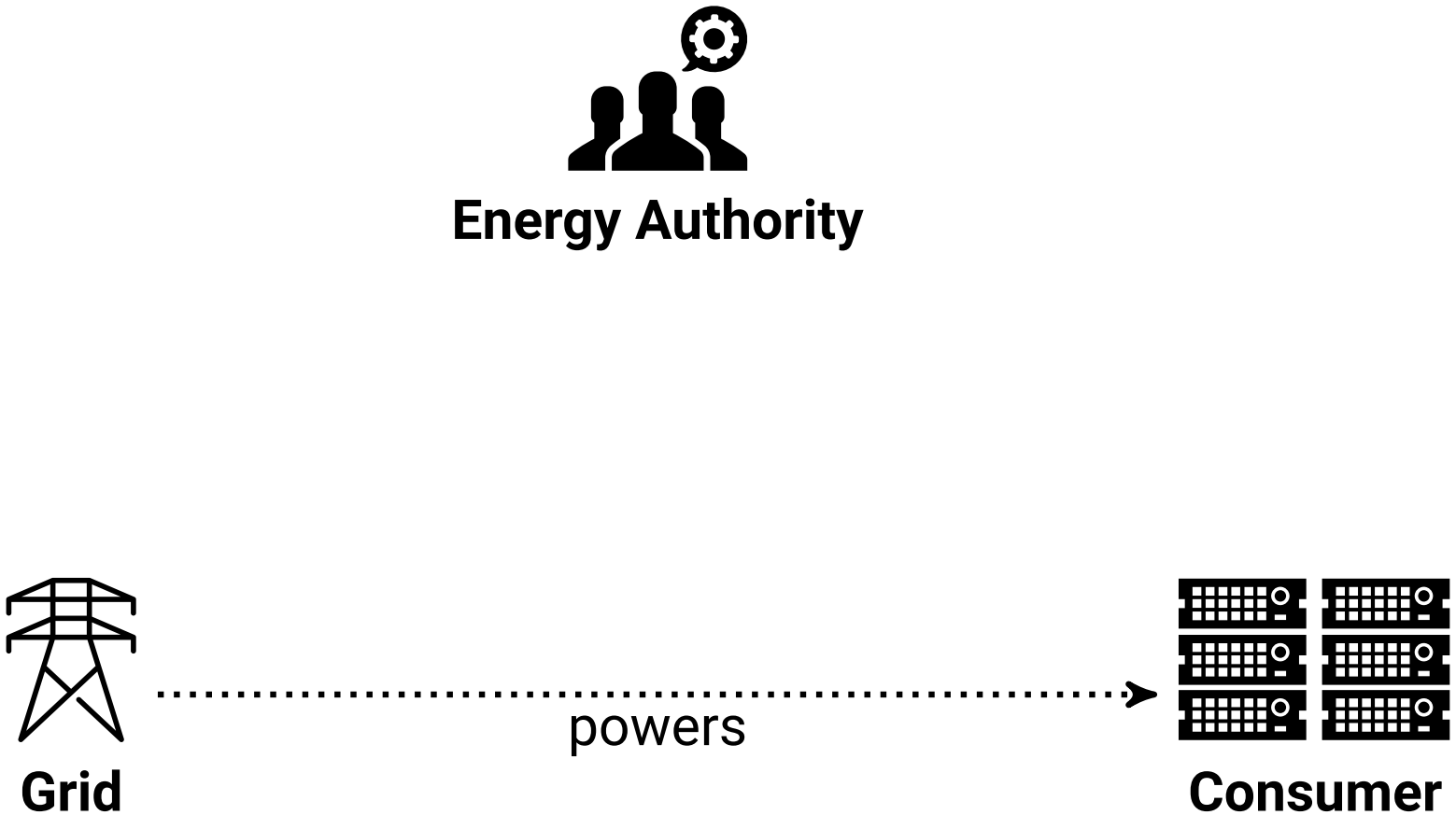
Demand Response

... a class of protocols to mitigate **fluctuations** in the power grid.



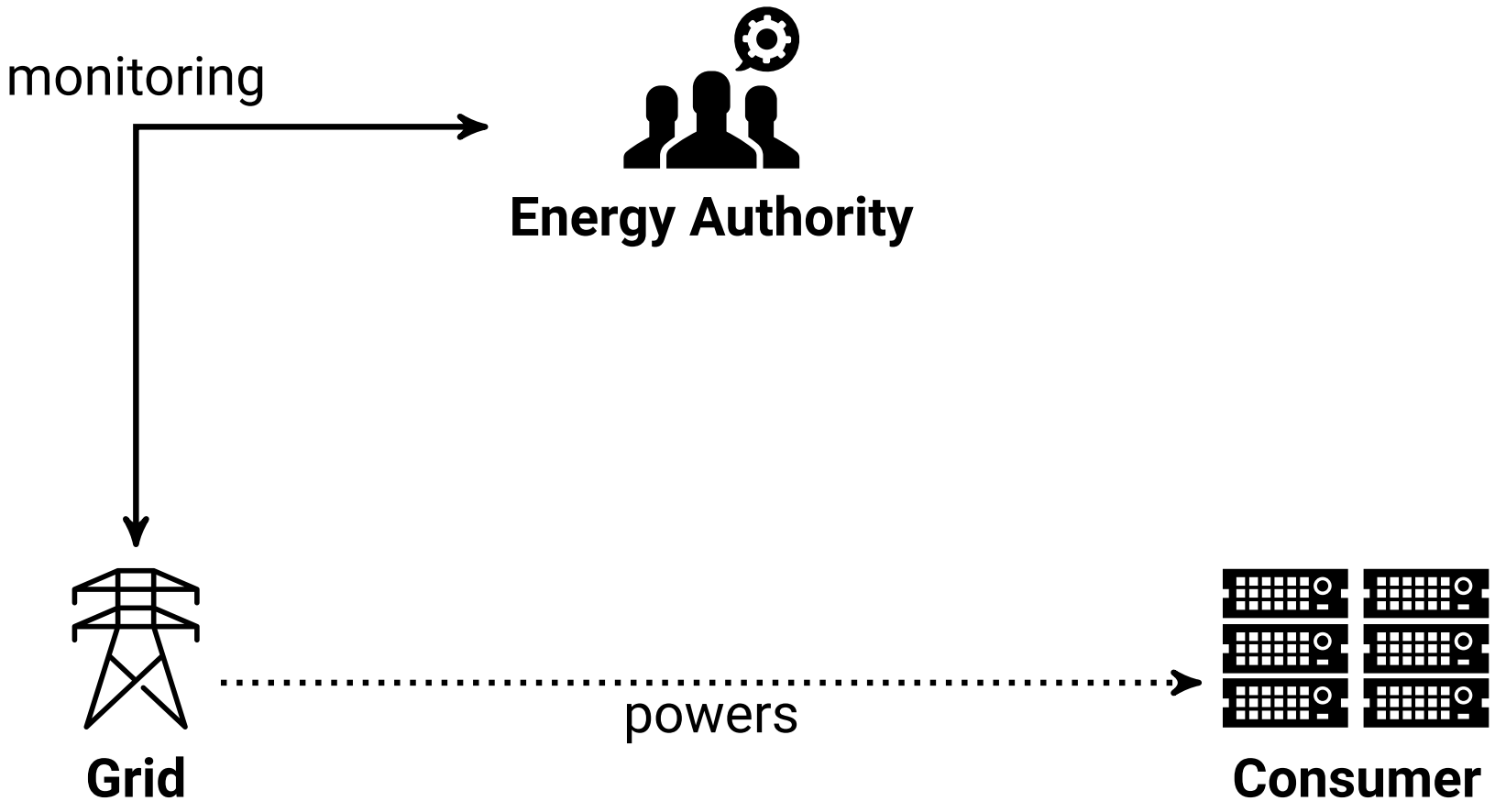
Demand Response

... a class of protocols to mitigate **fluctuations** in the power grid.



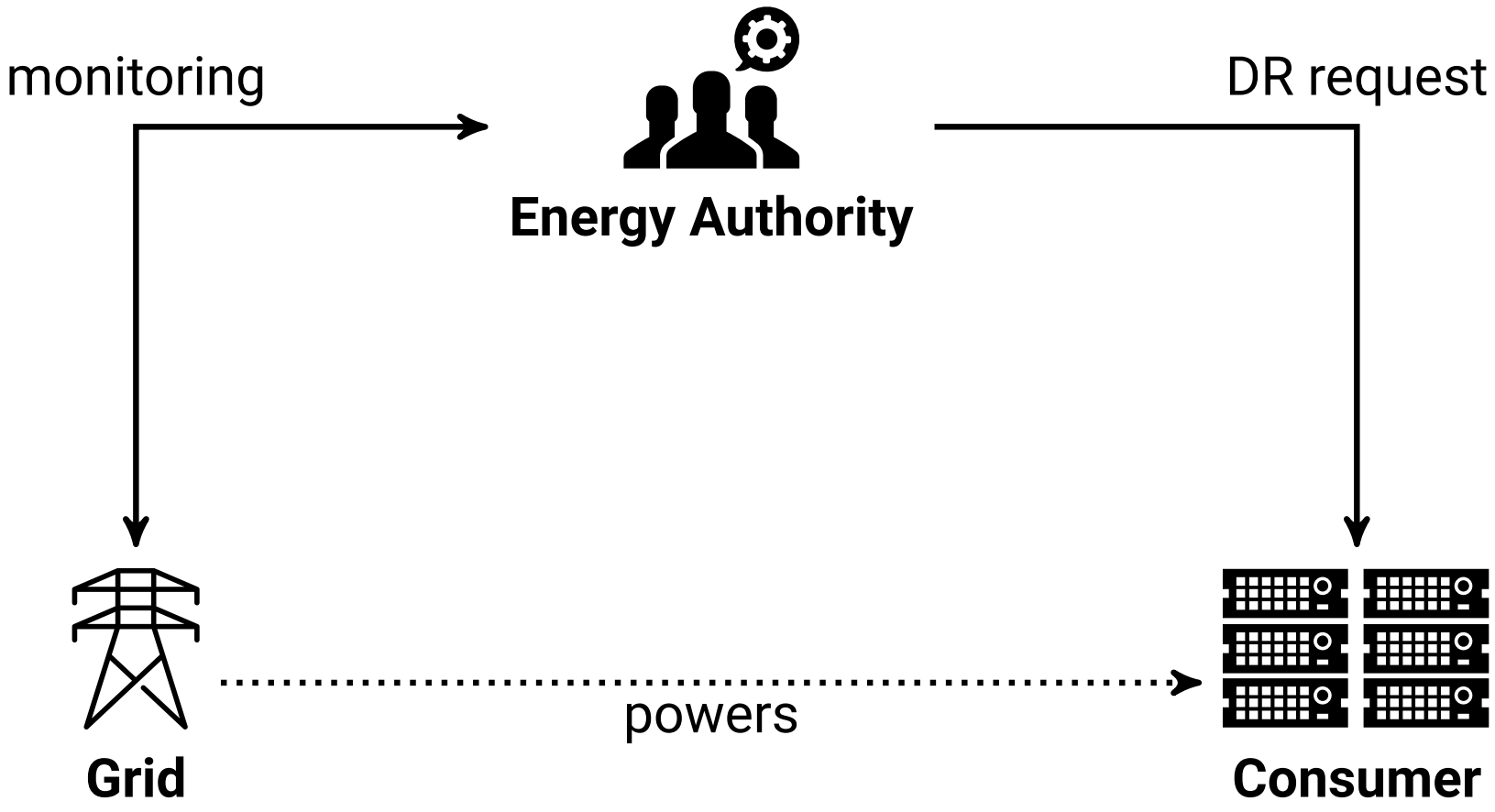
Demand Response

... a class of protocols to mitigate **fluctuations** in the power grid.



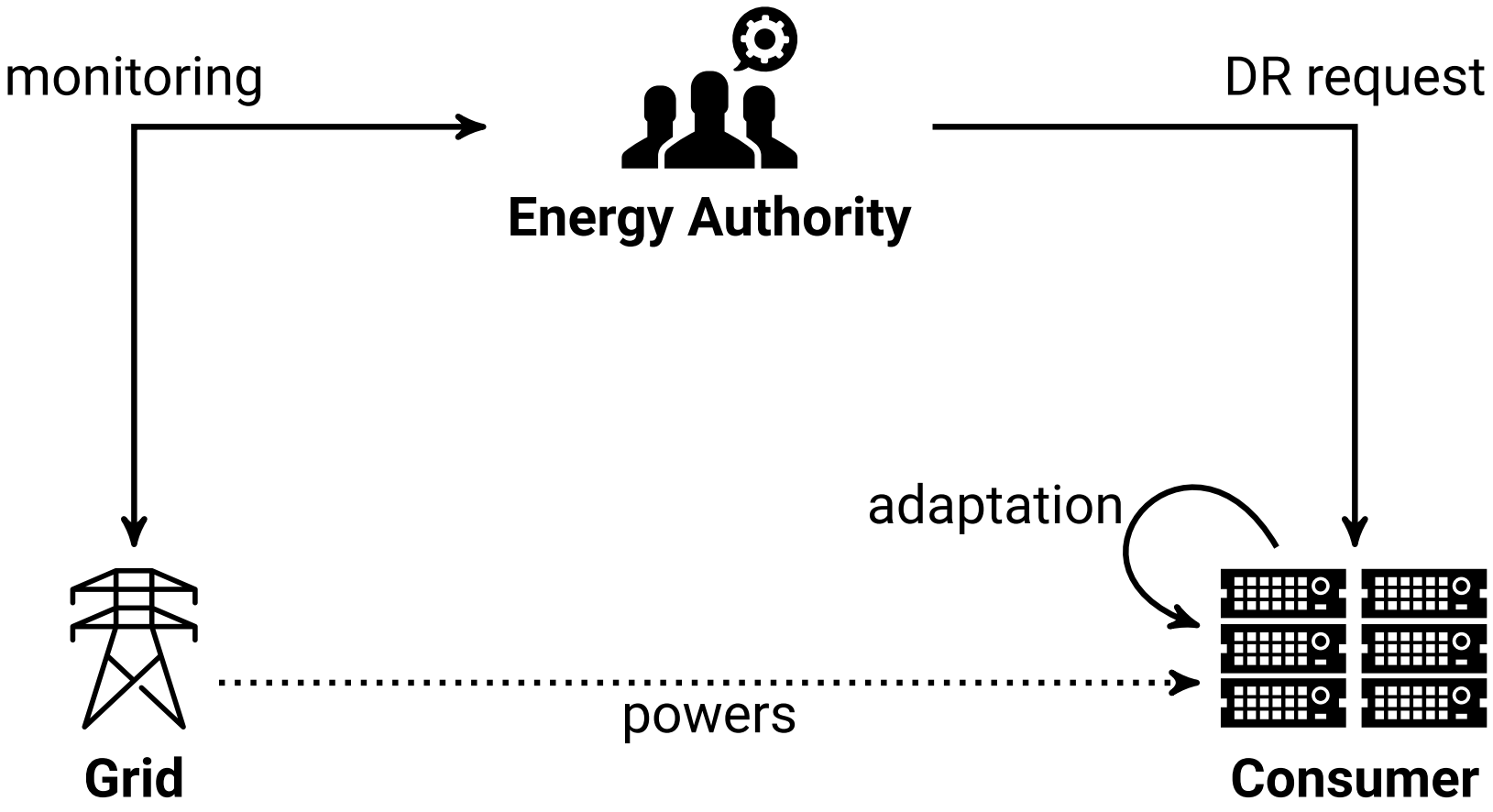
Demand Response

... a class of protocols to mitigate **fluctuations** in the power grid.



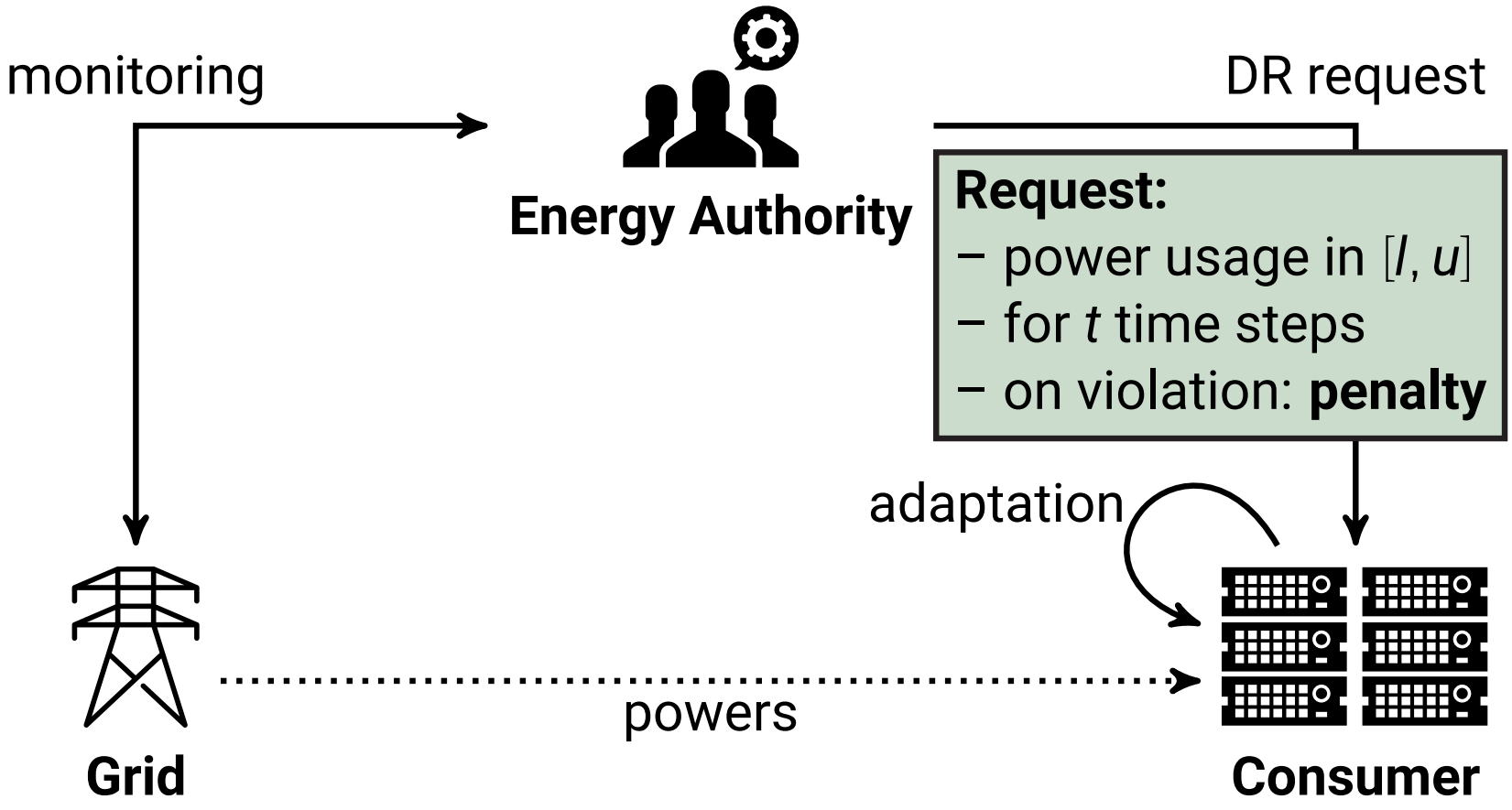
Demand Response

... a class of protocols to mitigate **fluctuations** in the power grid.



Demand Response

... a class of protocols to mitigate **fluctuations** in the power grid.



Formal Analysis of Demand Response

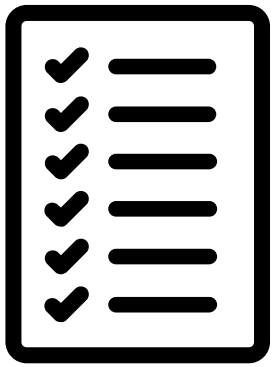
What is Demand Response and why is it useful?

Why and how do we formally analyse it?

What can we find out? **Demo**

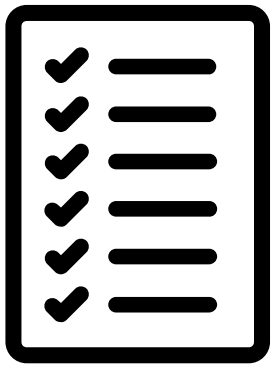
Goals of Formal Analysis

Goals of Formal Analysis



reliability

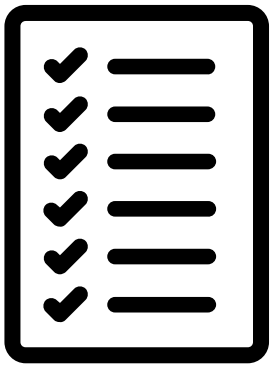
Goals of Formal Analysis



reliability

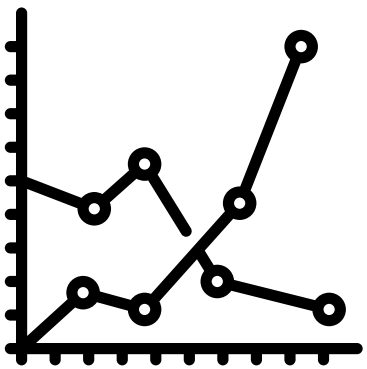
- core functionality
- contracts

Goals of Formal Analysis



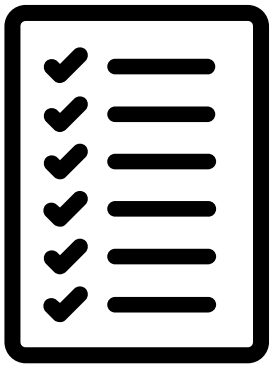
reliability

- core functionality
- contracts



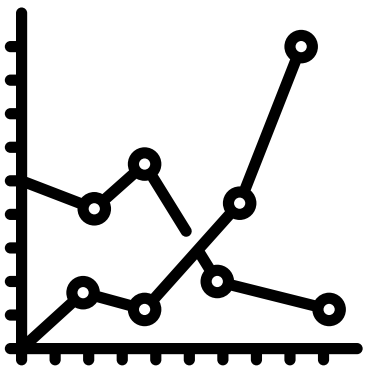
optimization

Goals of Formal Analysis



reliability

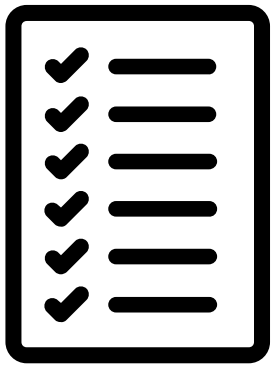
- core functionality
- contracts



optimization

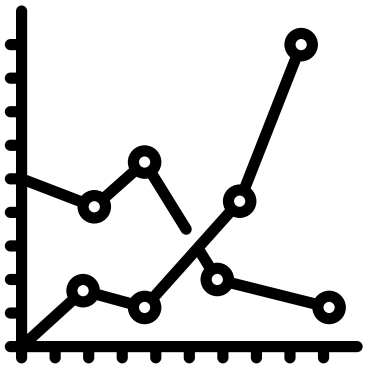
- minimum penalty
- minimum energy consumption
- maximum utility

Goals of Formal Analysis



reliability

- core functionality
- contracts

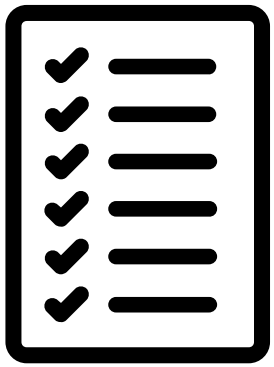


optimization

- minimum penalty
- minimum energy consumption
- maximum utility

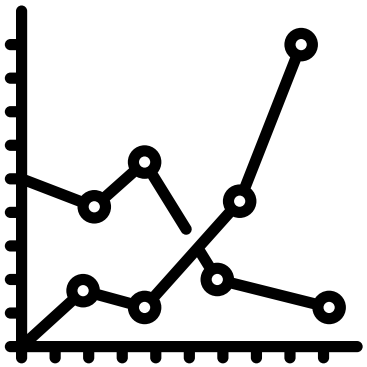
trade-offs

Goals of Formal Analysis



reliability

- core functionality
- contracts



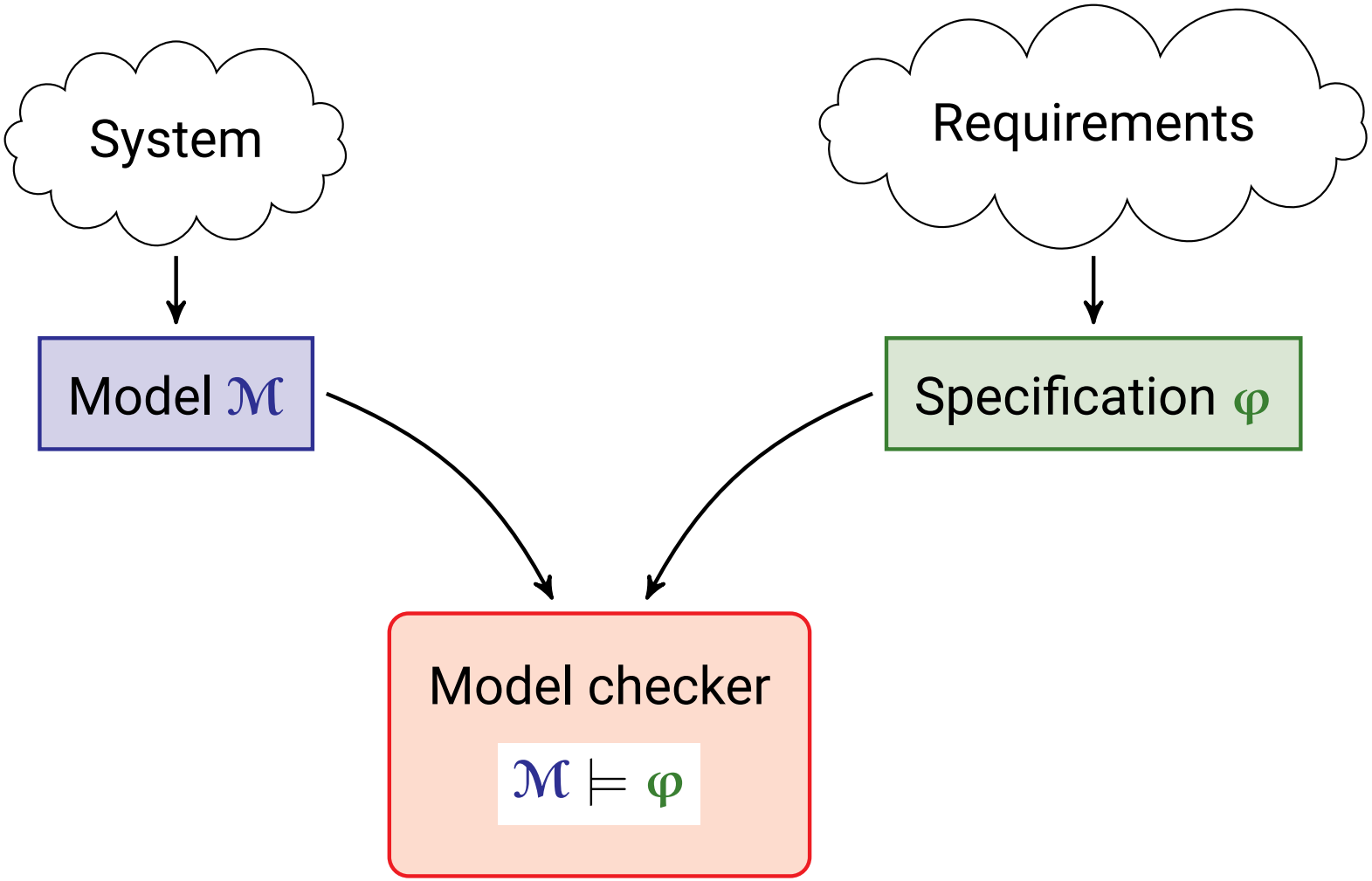
optimization

- minimum penalty
- minimum energy consumption
- maximum utility

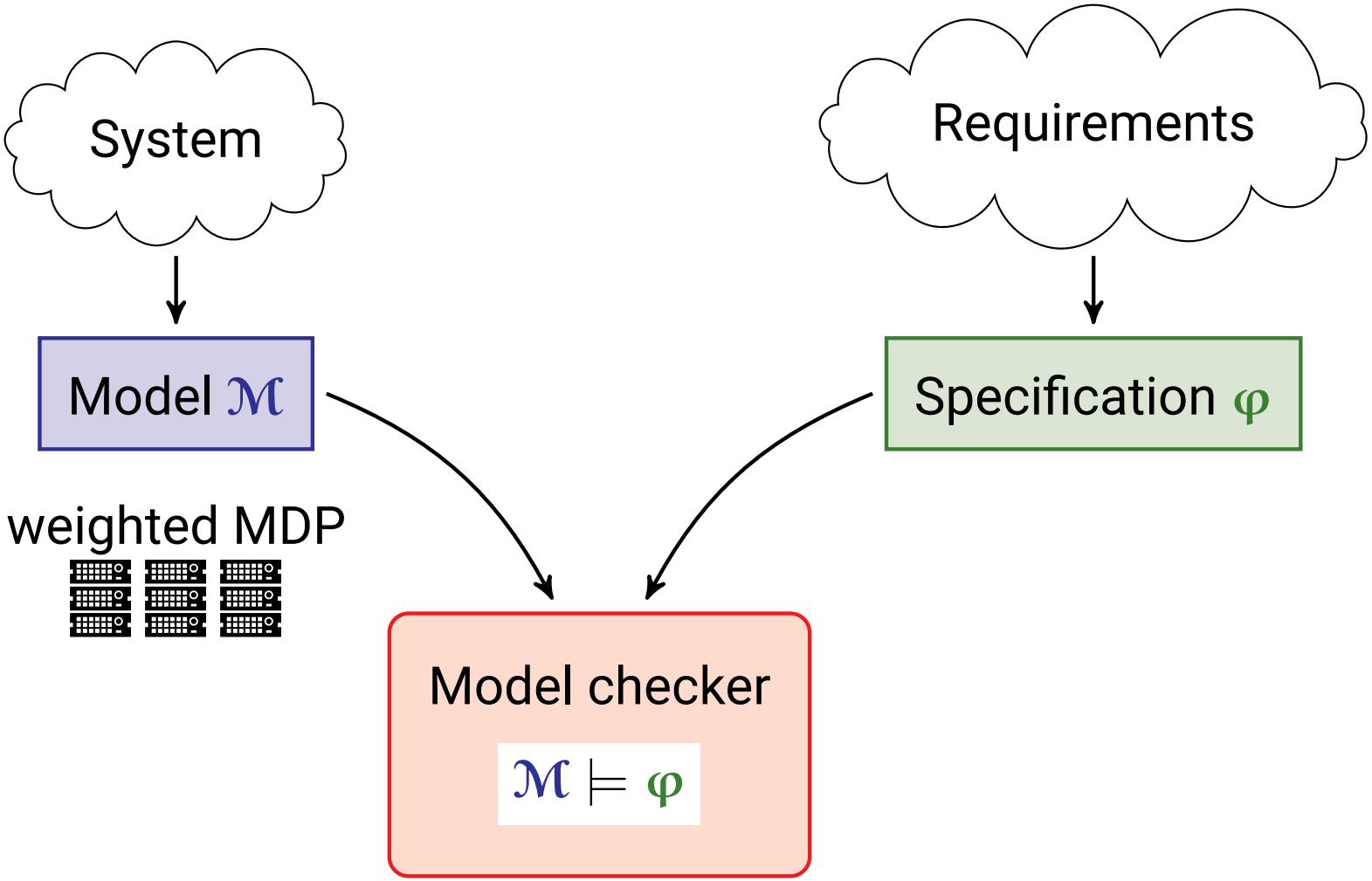
trade-offs

- penalty per utility
- energy per utility

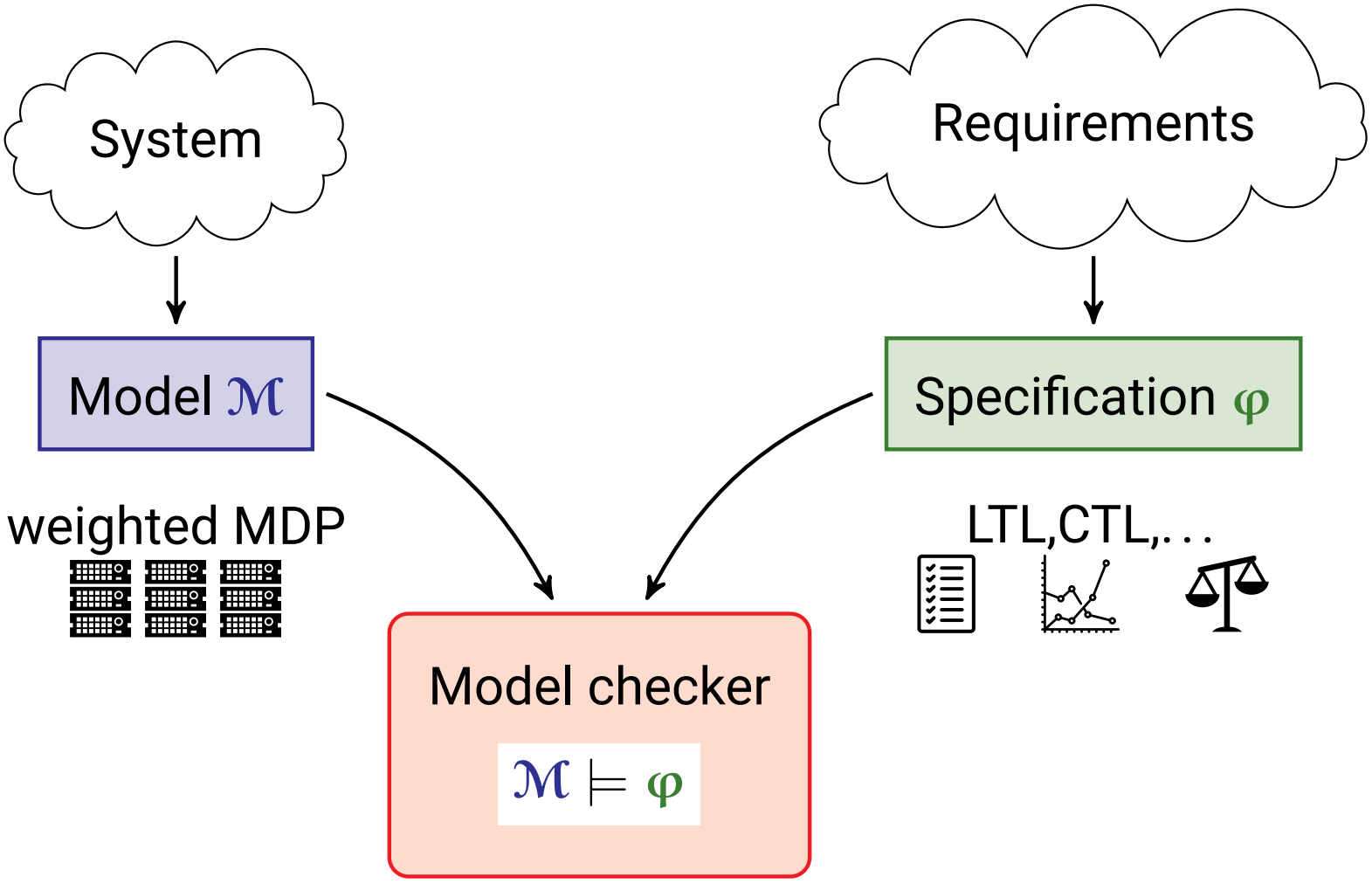
Model checking



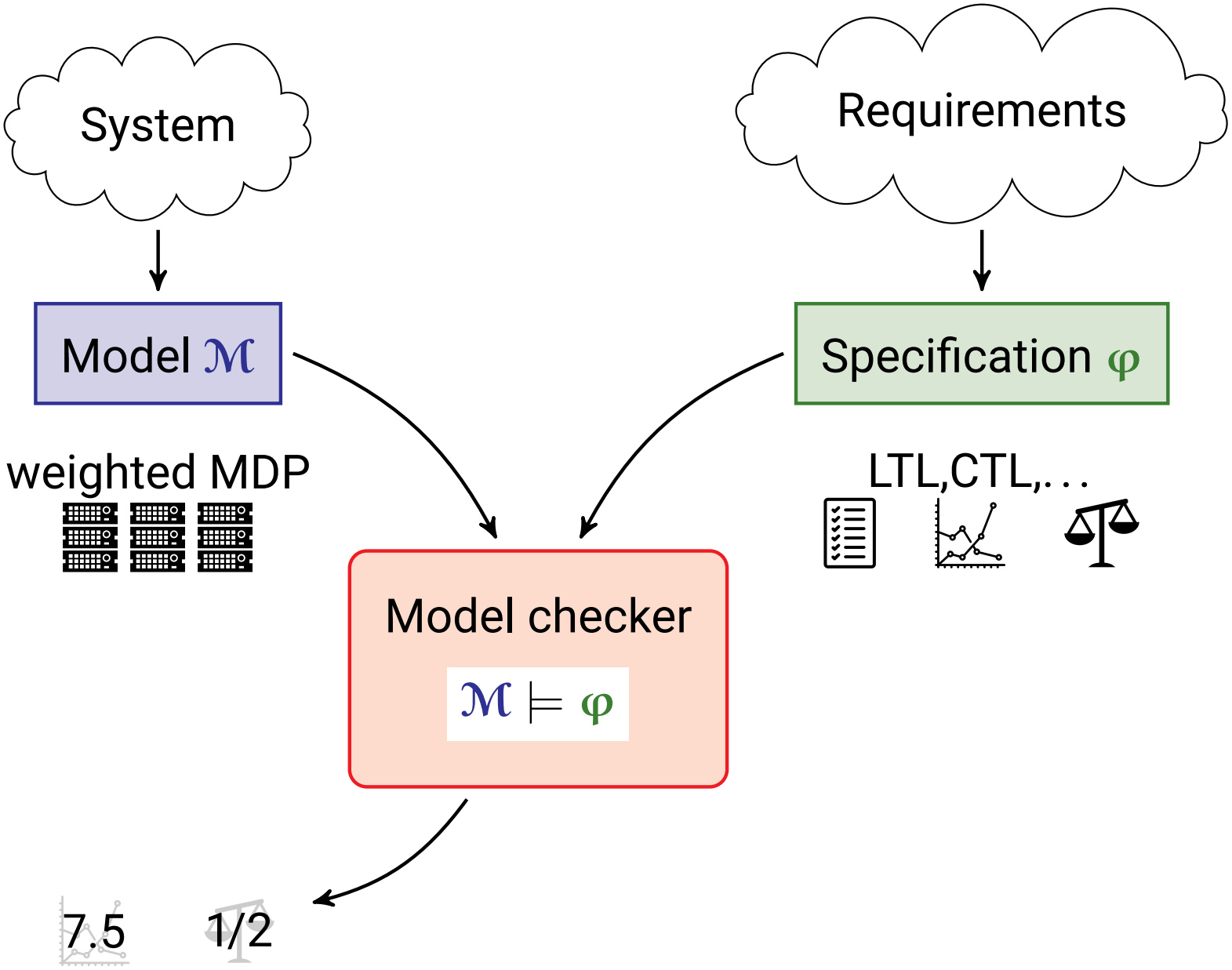
Model checking



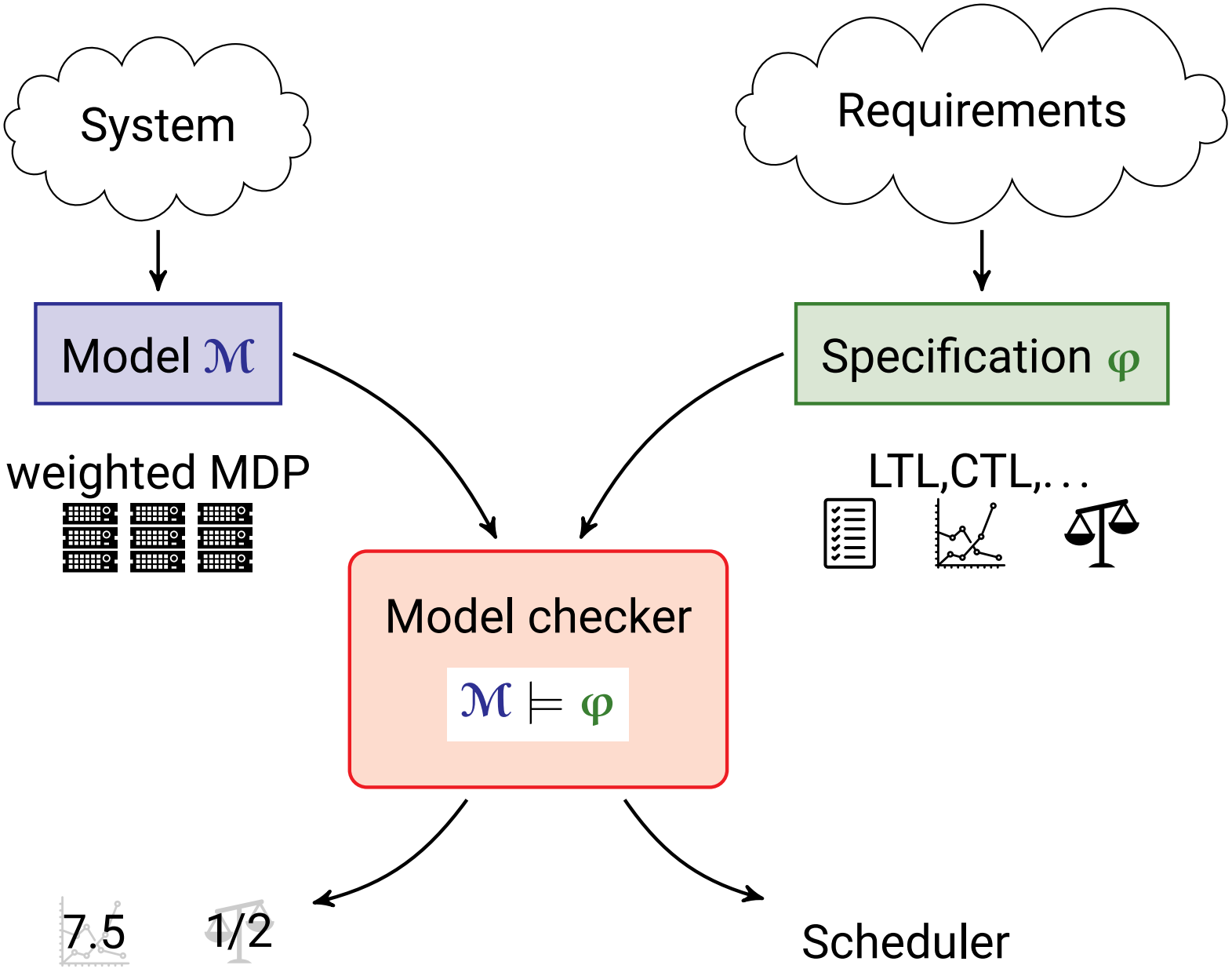
Model checking



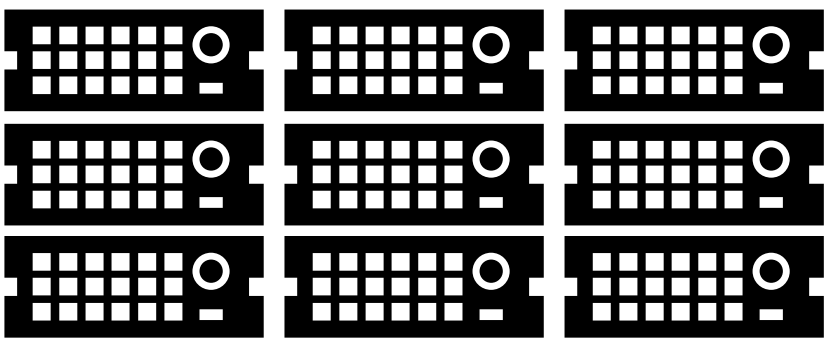
Model checking



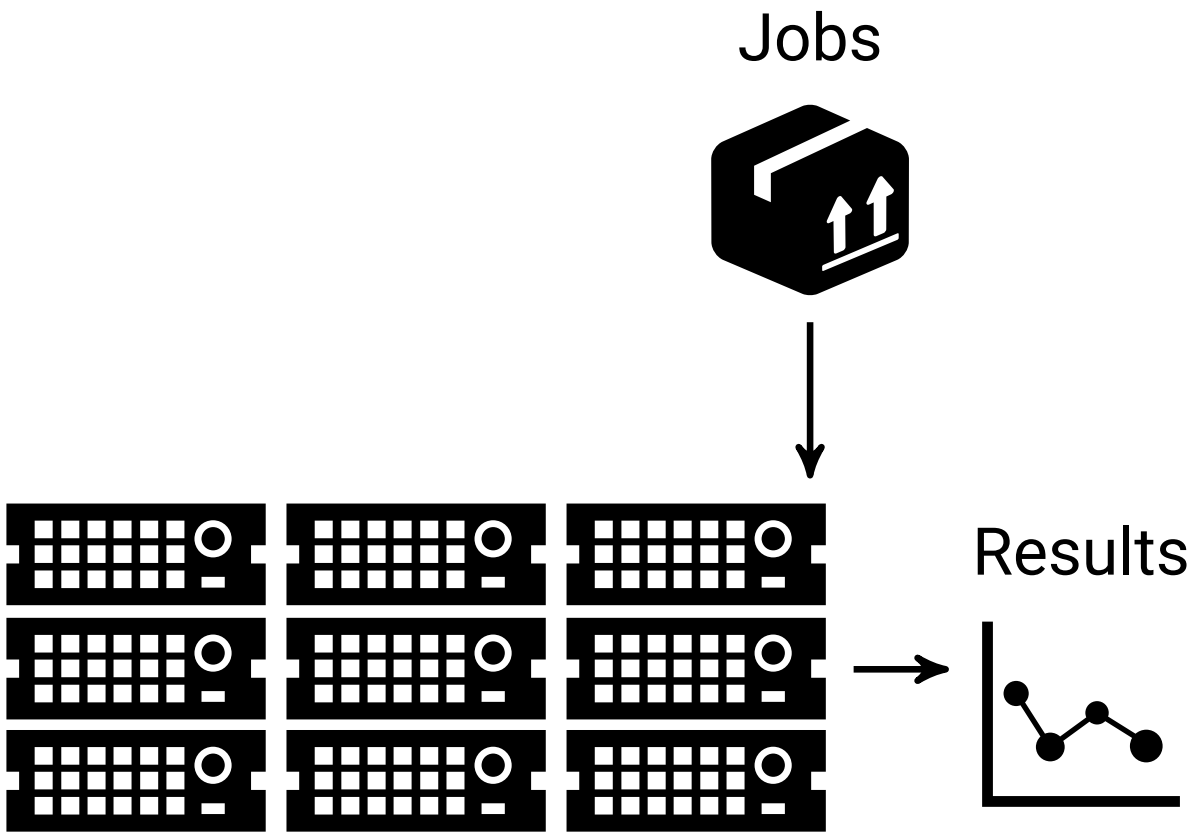
Model checking



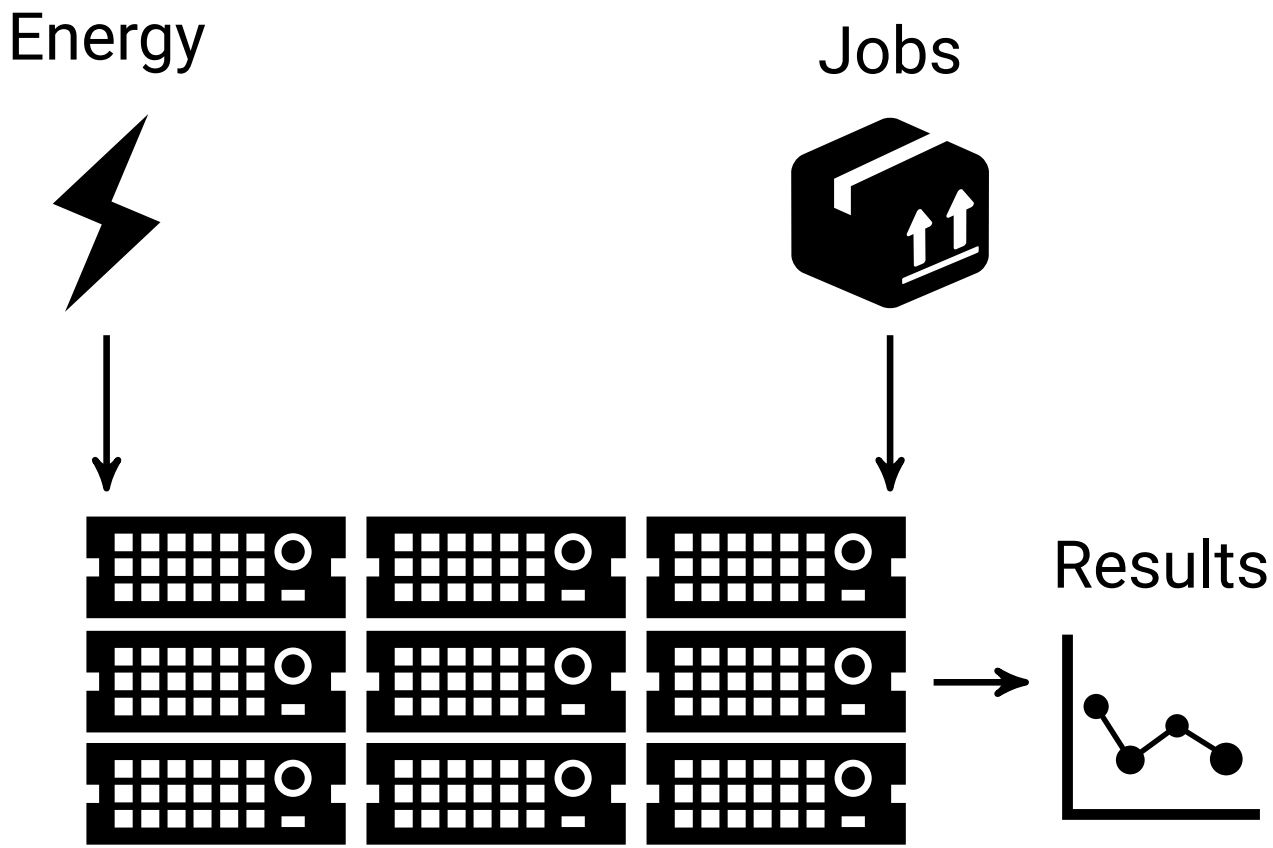
Data Centers as DR Consumers



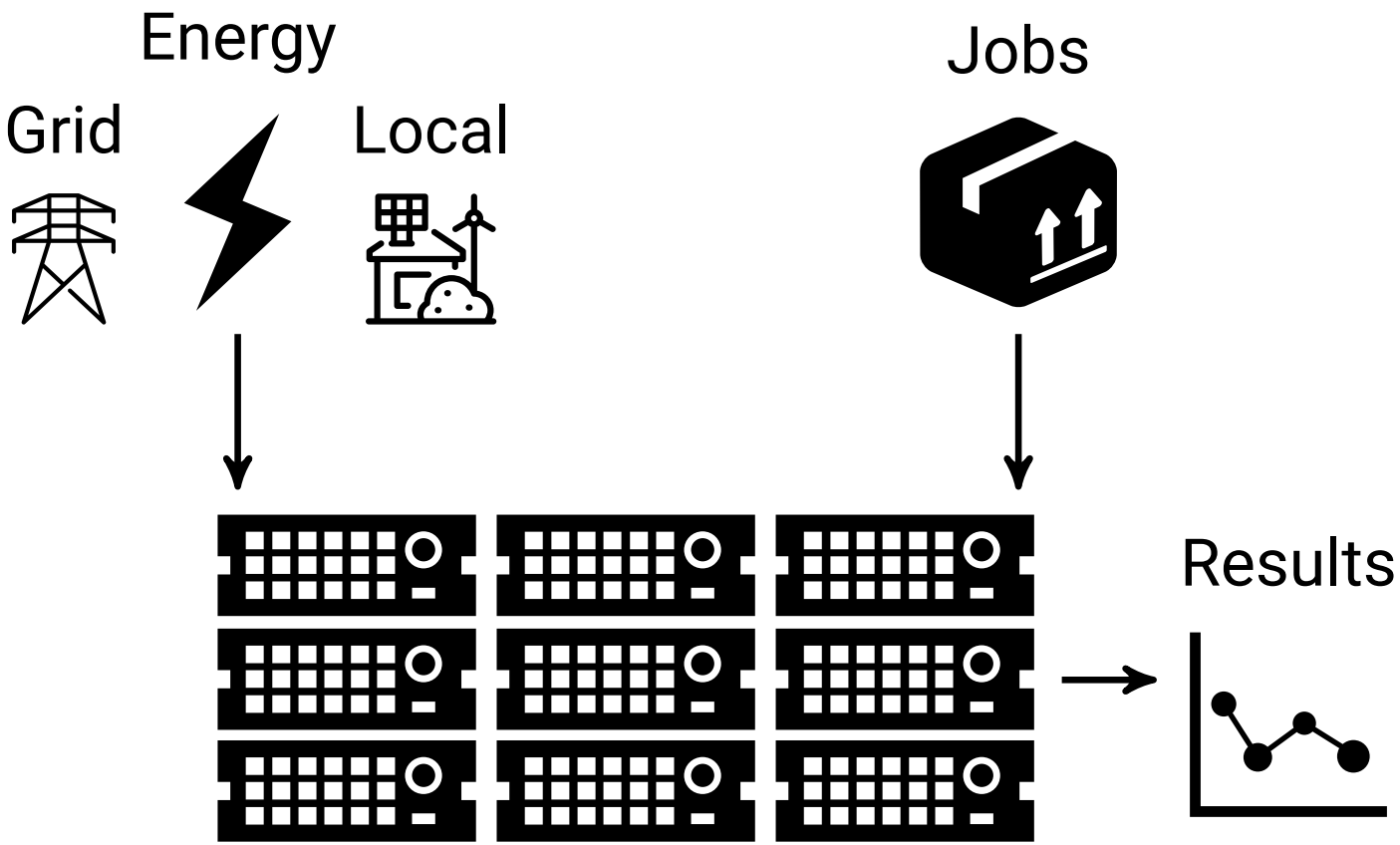
Data Centers as DR Consumers



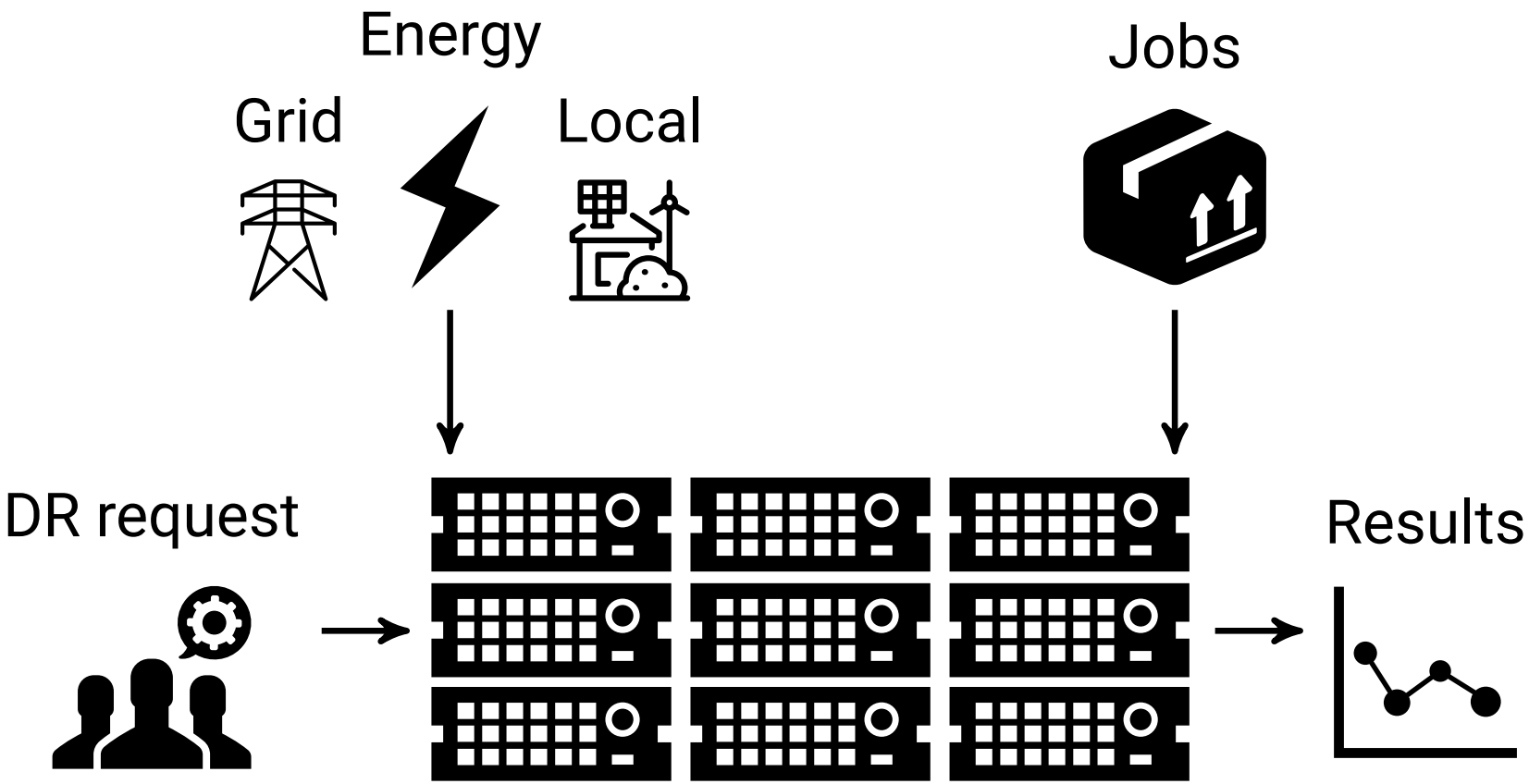
Data Centers as DR Consumers



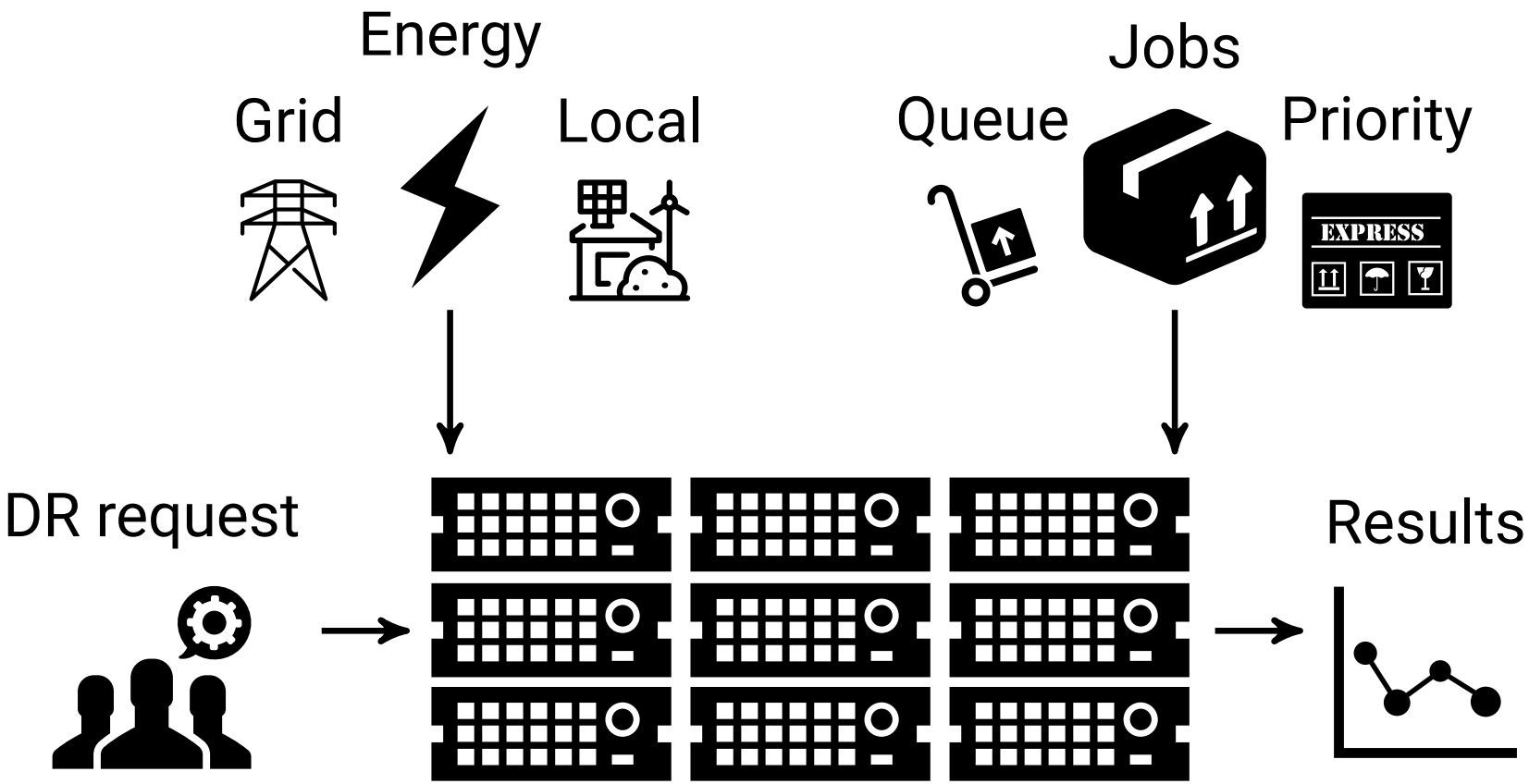
Data Centers as DR Consumers



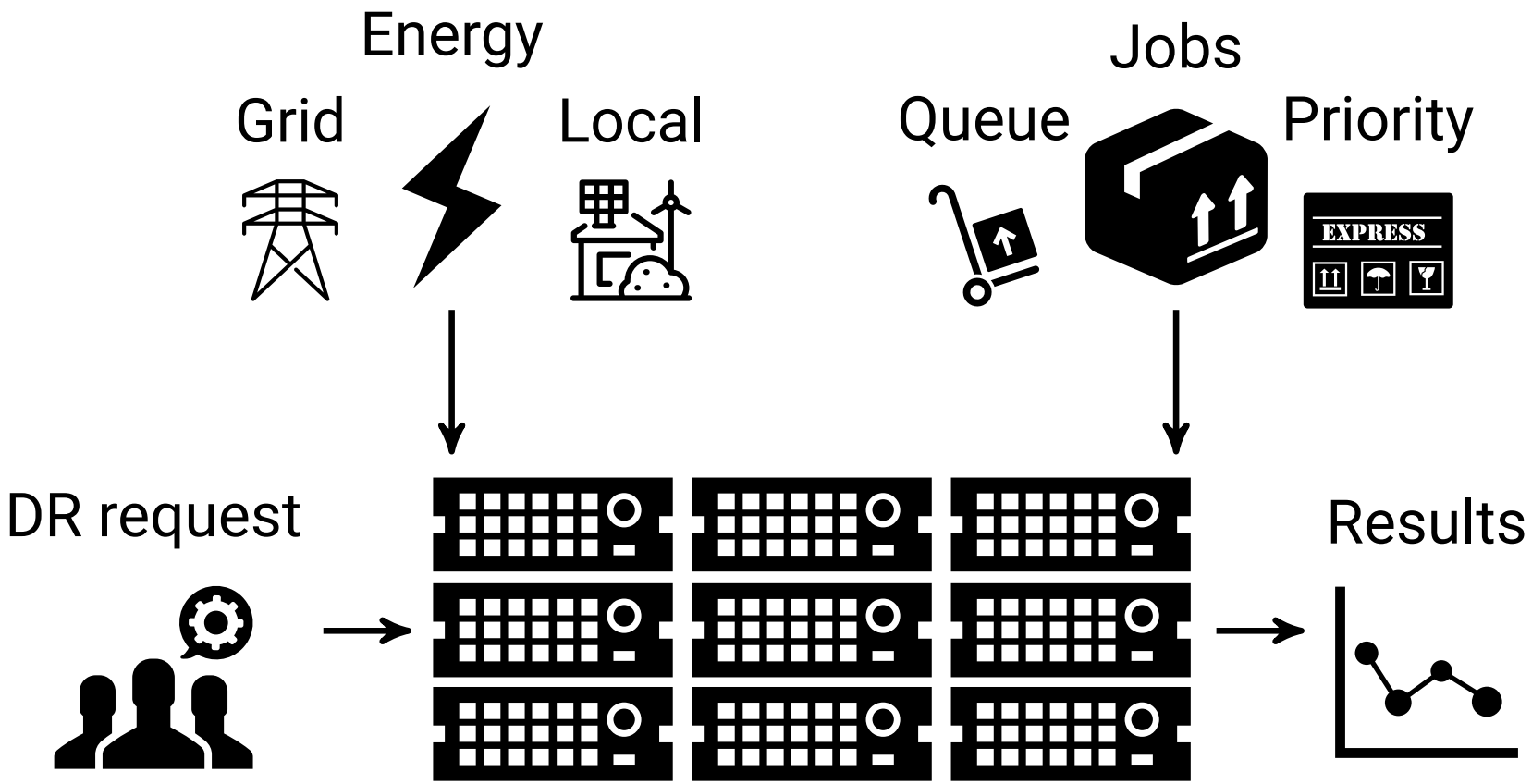
Data Centers as DR Consumers



Data Centers as DR Consumers

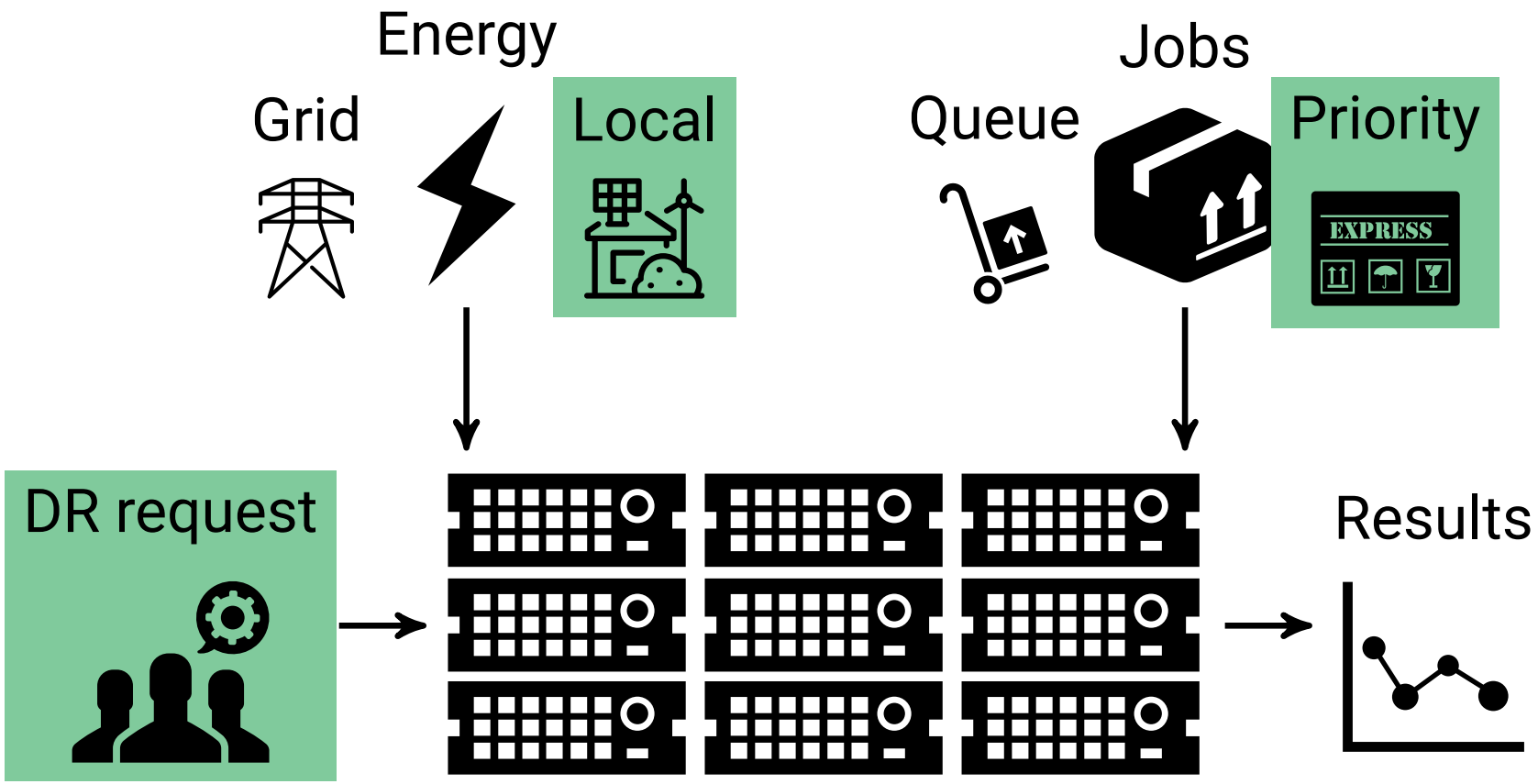


Data Centers as DR Consumers



time resolution, capacity

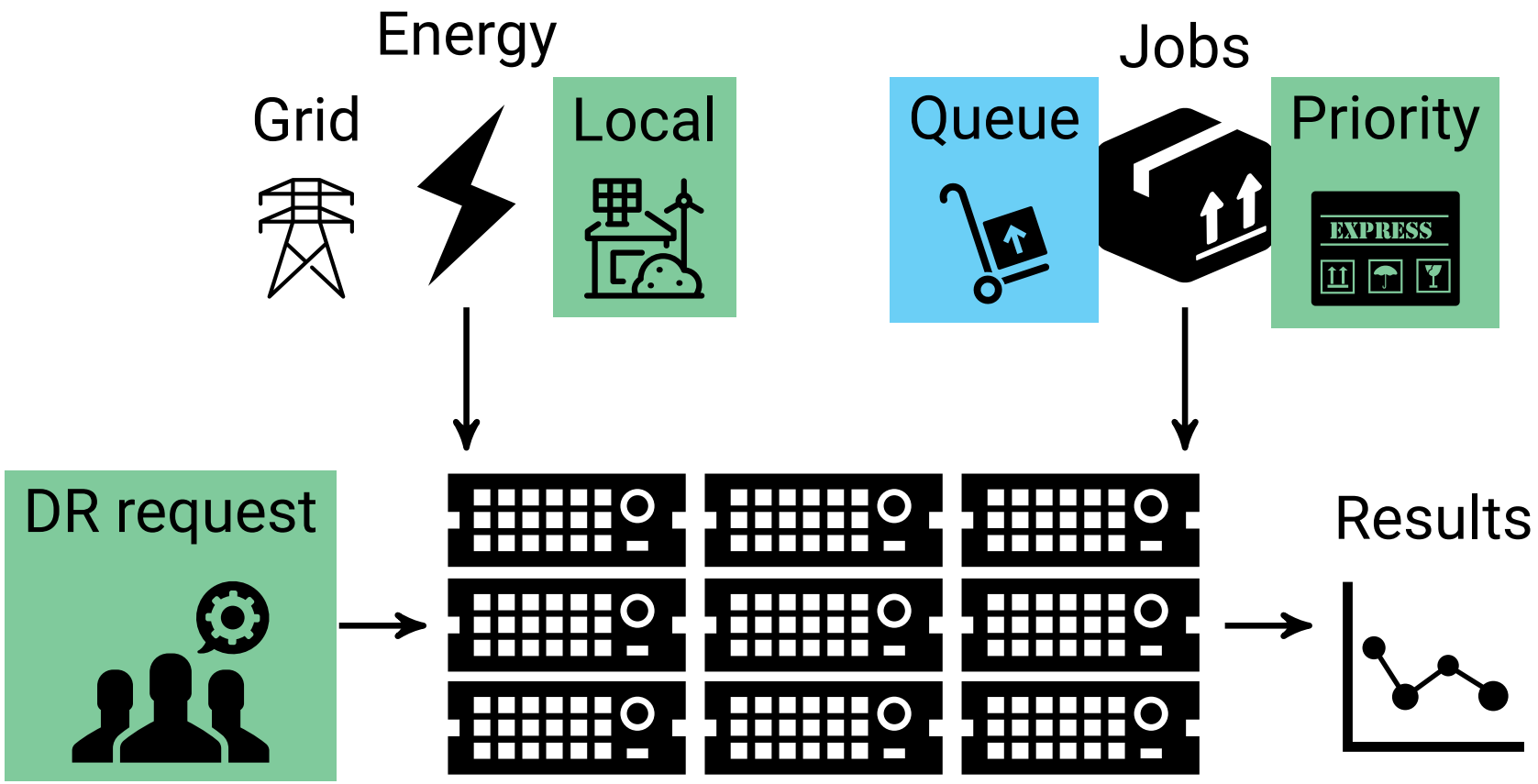
Data Centers as DR Consumers



time resolution, capacity

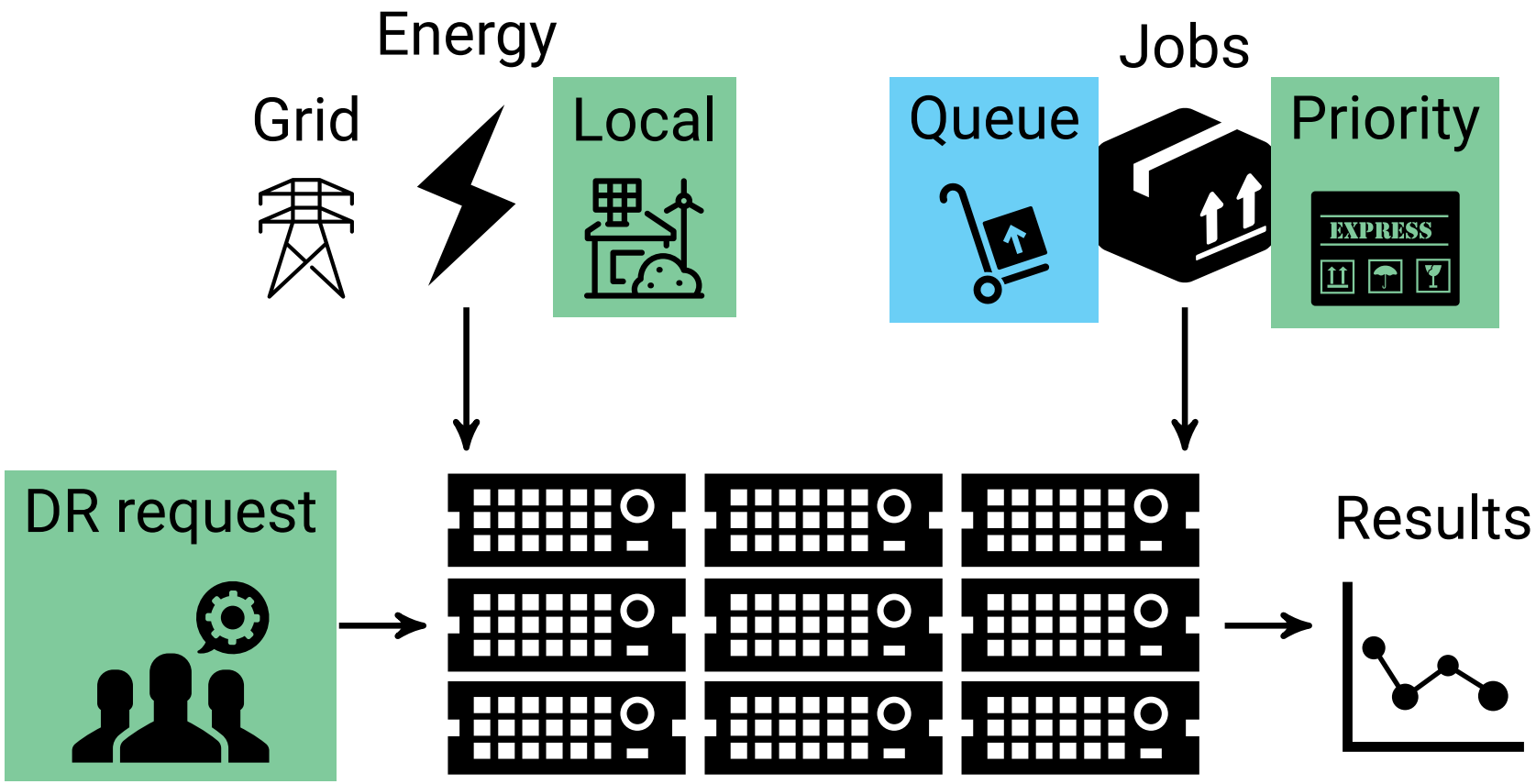
environmental

Data Centers as DR Consumers



time resolution, capacity
environmental
(non-det.) choice

Data Centers as DR Consumers



time resolution, capacity
environmental
(non-det.) choice

many more modules

hard limits, heterogeneous jobs, dependencies, multiple centers, . . .

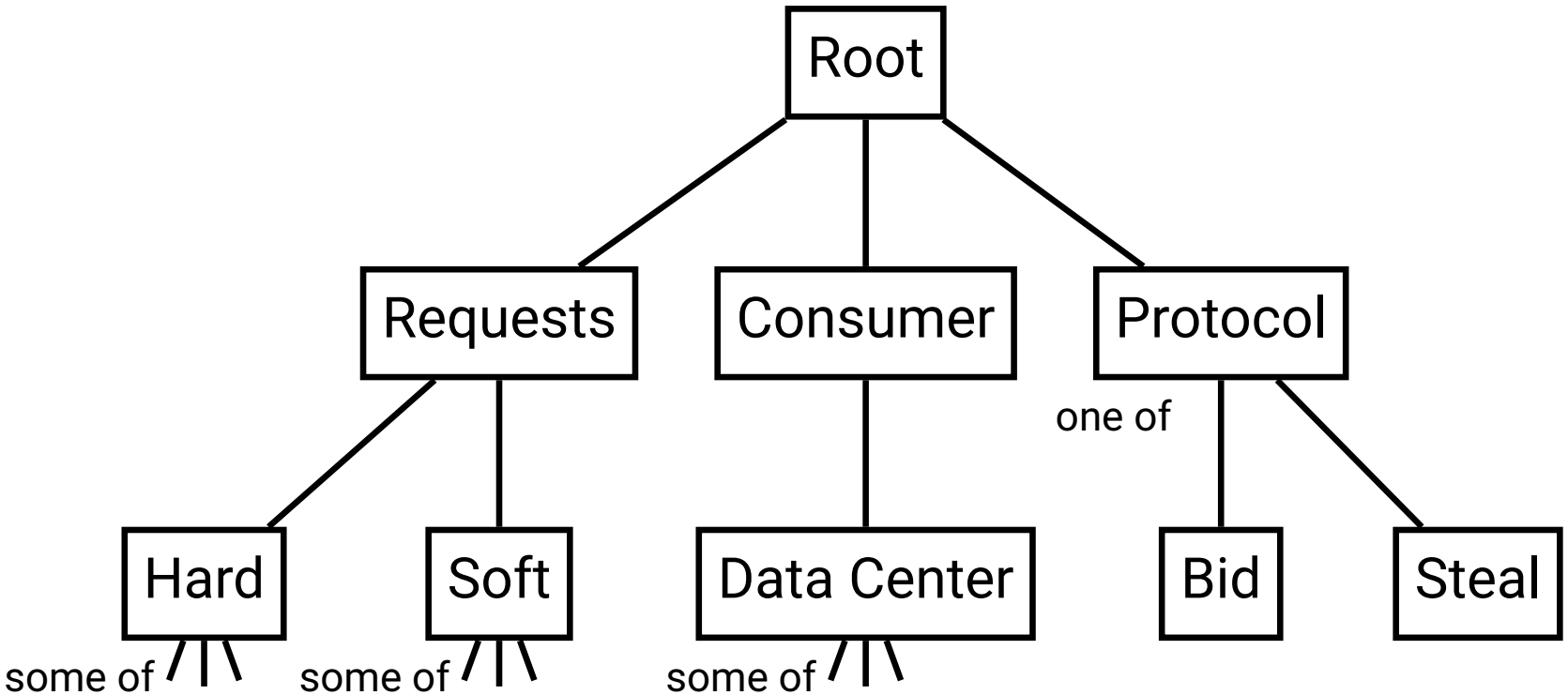
Feature Model

Feature Model

...a hierarchical model description with replaceable parts.

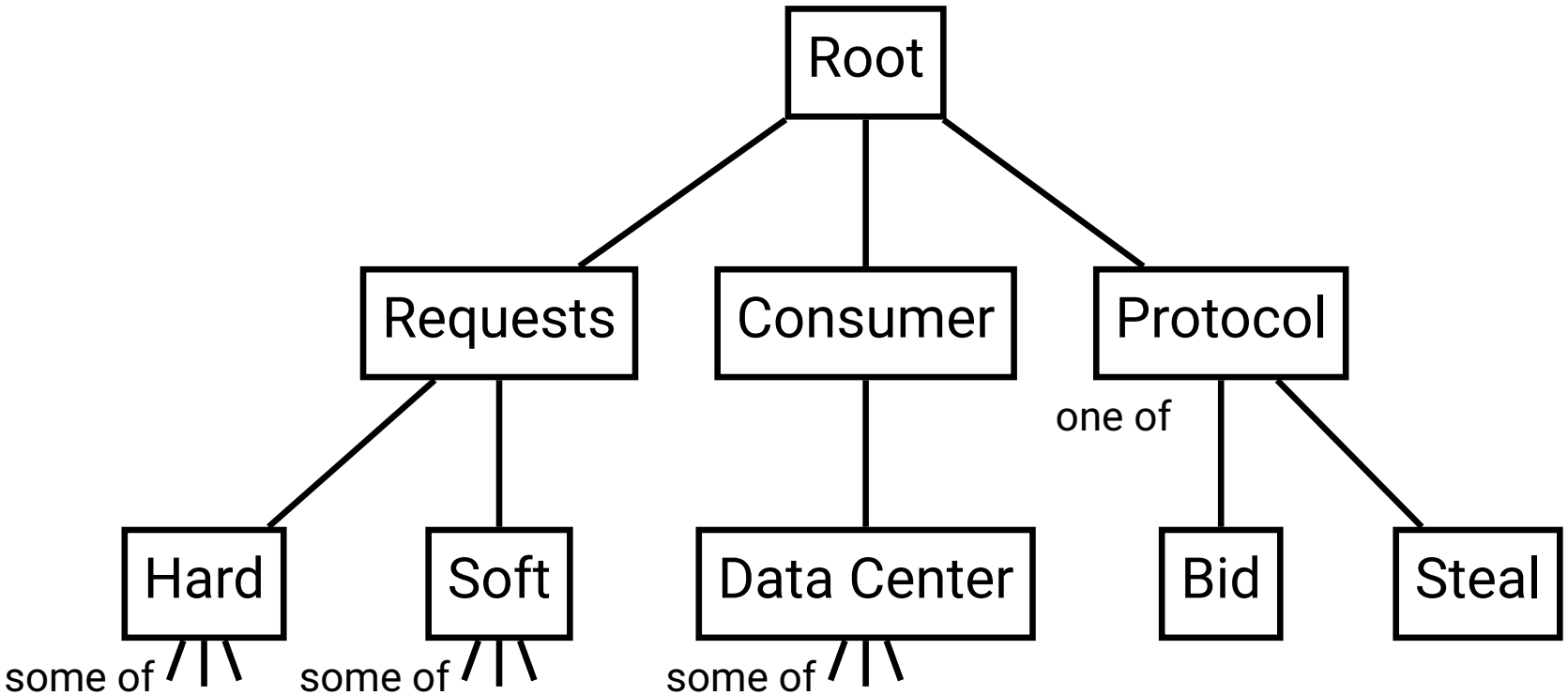
Feature Model

...a hierarchical model description with replaceable parts.



Feature Model

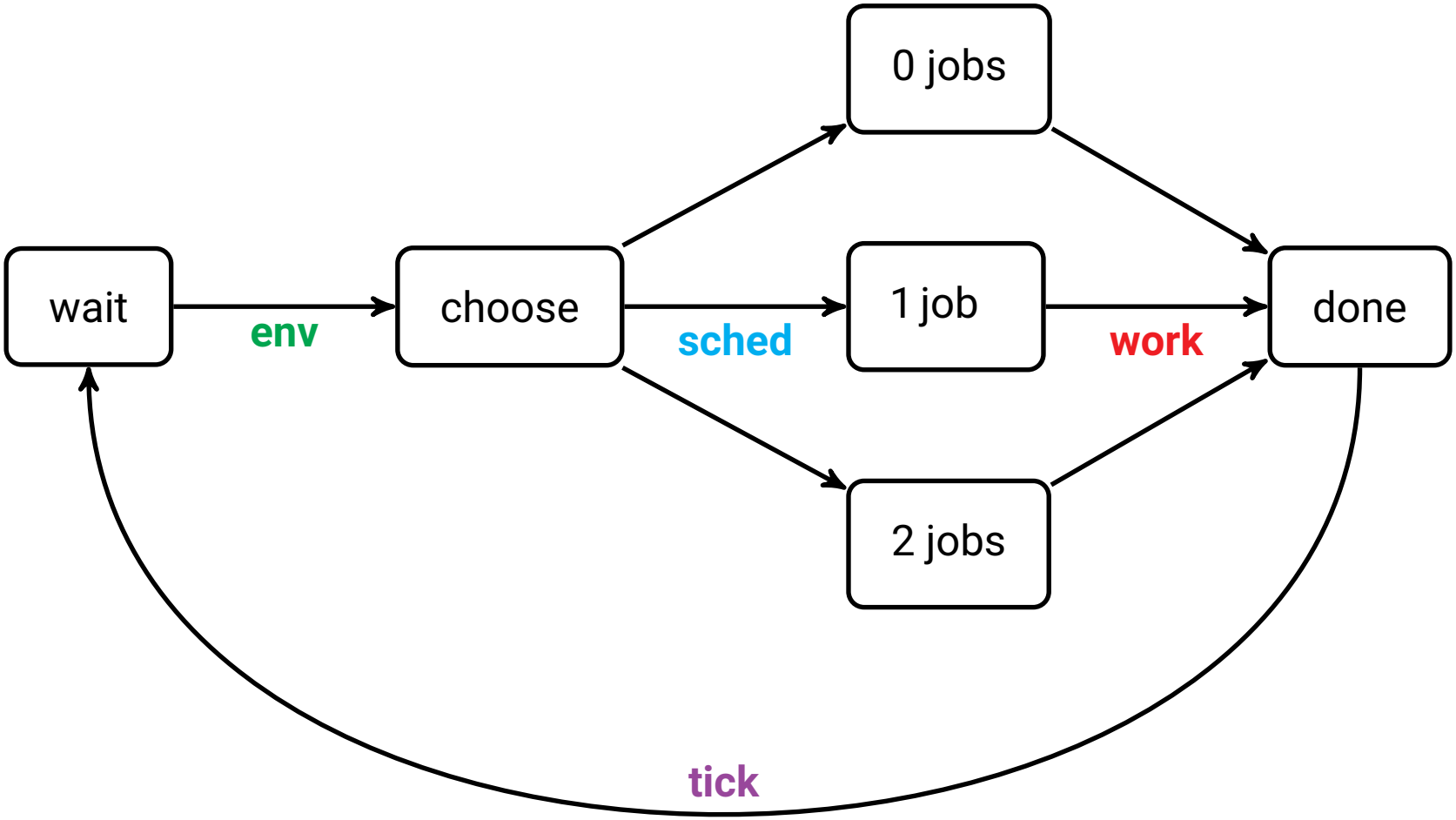
...a hierarchical model description with replaceable parts.



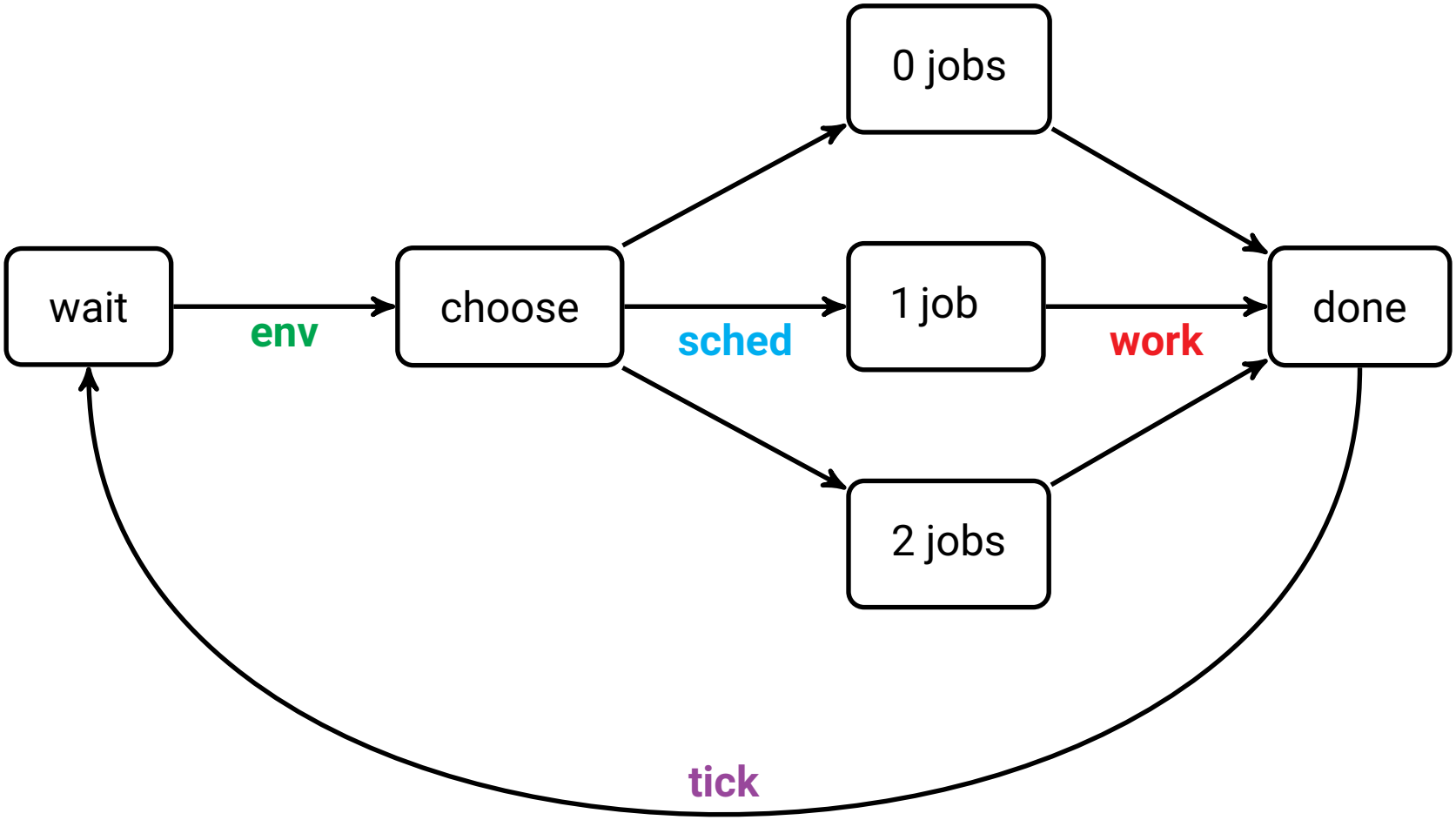
Allows for **model families** with variant members.

Data Center - Single Step

Data Center - Single Step



Data Center - Single Step



Possible weights:

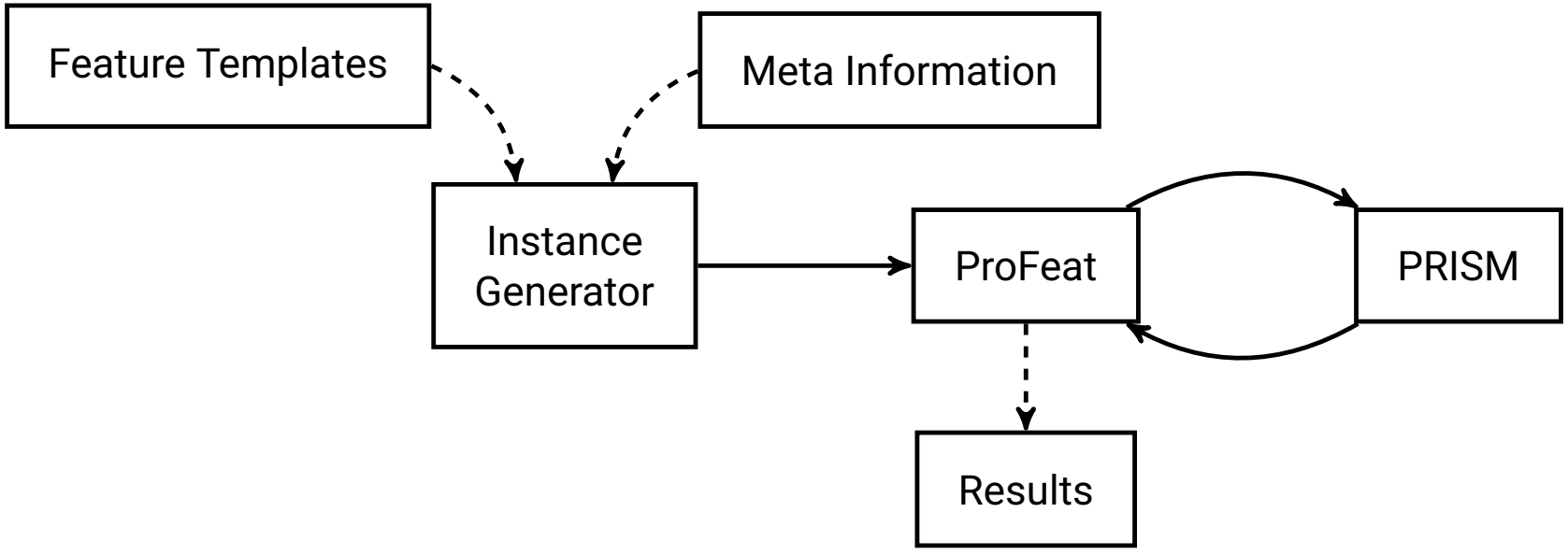
work **jobs**=2, **energy**=4, **penalty**=1

tick **steps**=1

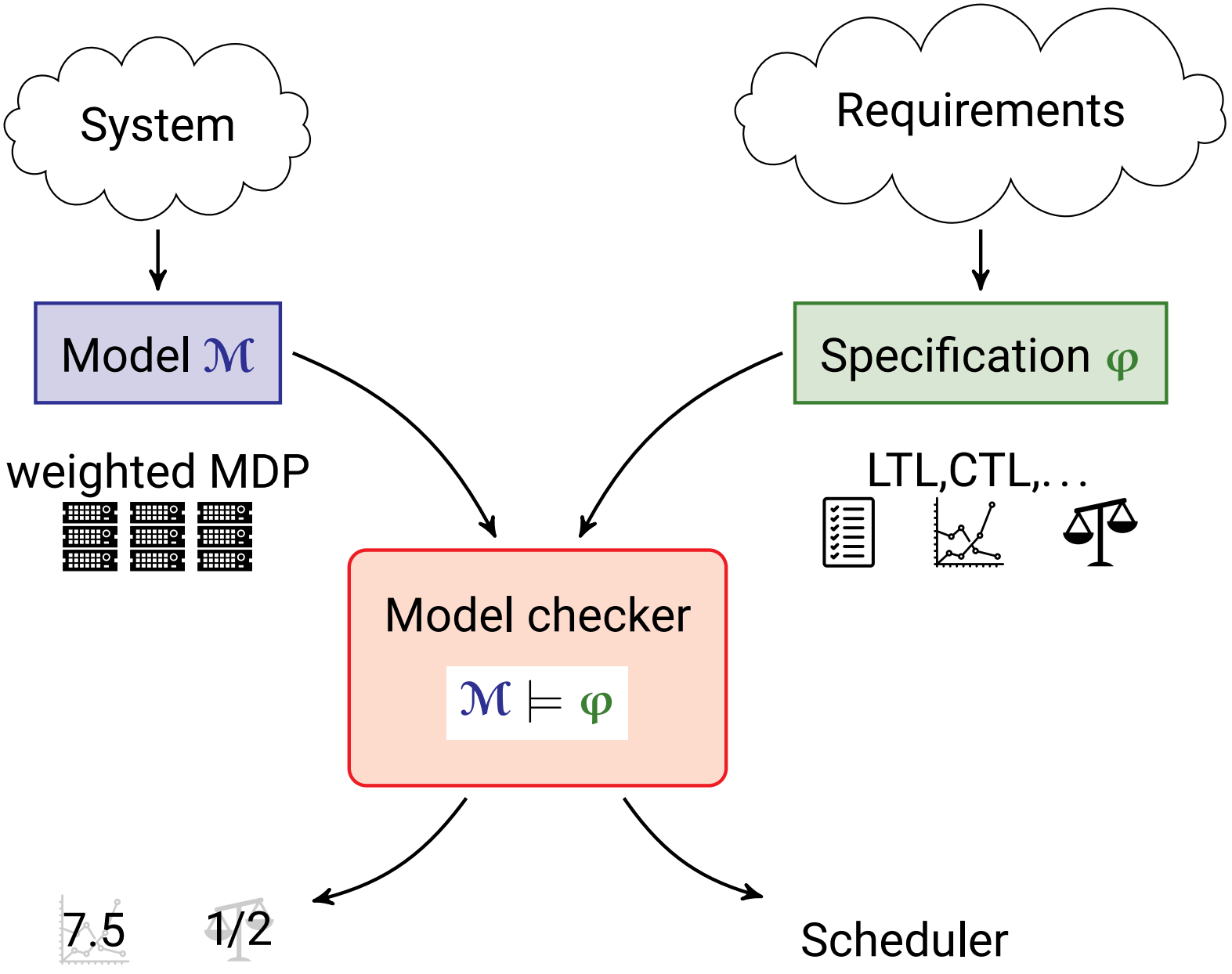
Family-based Analysis

Family-based Analysis

Toolchain:



Model checking



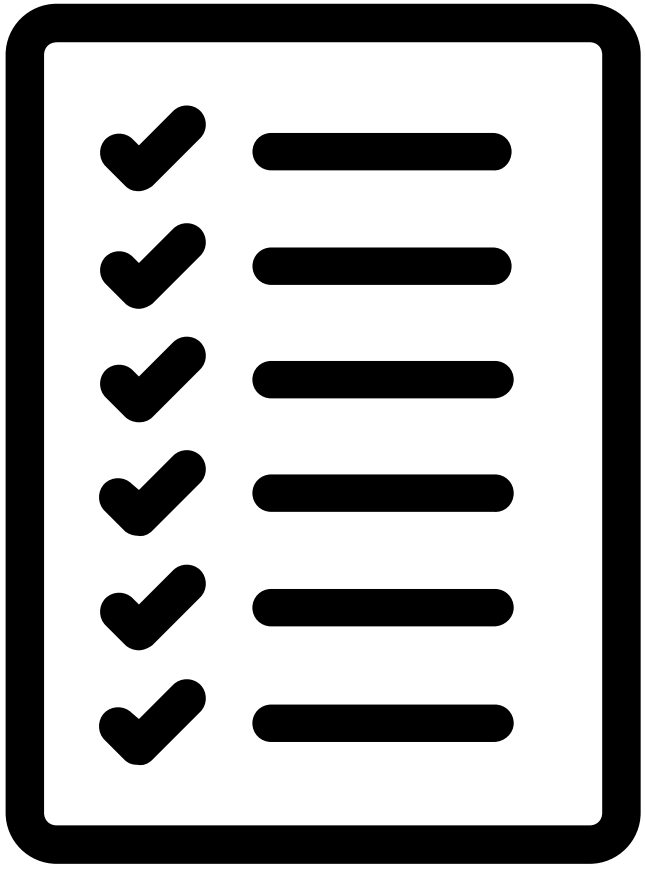
Formal Analysis of Demand Response

What is Demand Response and why is it useful?

Why and how do we formally analyse it?

What can we find out? **Demo**

Some Results



Reliability

Some Results

The end of the day will be reached (when time’s up).

Some Results

The end of the day will be reached (when time's up).

$$\Pr^{\min}(\Diamond \text{eod}) \qquad \Pr^{\min}(\Diamond^{\#\text{steps}=4*24} \text{eod})$$

$\Diamond$ . . . finally

eod . . . end of day

Some Results

The end of the day will be reached (when time's up).

$$\Pr^{\min}(\Diamond \text{eod}) = 1 \qquad \Pr^{\min}(\Diamond^{\#\text{steps}=4*24} \text{eod}) = 1$$

$\Diamond$ . . . finally

eod . . . end of day

Some Results

The end of the day will be reached (when time's up).

$$\Pr^{\min}(\Diamond \text{eod}) = 1 \qquad \Pr^{\min}(\Diamond^{\#\text{steps}=4*24} \text{eod}) = 1$$

It is possible to complete all jobs.

$\Diamond$ . . . finally

eod . . . end of day

Some Results

The end of the day will be reached (when time's up).

$$\Pr^{\min}(\Diamond \text{eod}) = 1 \qquad \Pr^{\min}(\Diamond^{\#\text{steps}=4*24} \text{eod}) = 1$$

It is possible to complete all jobs.

$$\Pr^{\max}(\Diamond(J = 0))$$

$\Diamond$ . . . finally

eod . . . end of day

Some Results

The end of the day will be reached (when time's up).

$$\Pr^{\min}(\Diamond \text{eod}) = 1 \qquad \Pr^{\min}(\Diamond^{\#\text{steps}=4*24} \text{eod}) = 1$$

It is possible to complete all jobs.

$$\Pr^{\max}(\Diamond(J = 0)) \approx 0.25$$

$\Diamond$ . . . finally

eod . . . end of day

Some Results

The end of the day will be reached (when time's up).

$$\Pr^{\min}(\Diamond \text{eod}) = 1 \qquad \Pr^{\min}(\Diamond^{\#\text{steps}=4*24} \text{eod}) = 1$$

It is possible to complete all jobs.

$$\Pr^{\max}(\Diamond(J = 0)) \approx 0.25$$

Don't use grid energy and finish all jobs.

$\Diamond$ . . . finally eod . . . end of day

Some Results

The end of the day will be reached (when time's up).

$$\Pr^{\min}(\Diamond \text{eod}) = 1 \qquad \Pr^{\min}(\Diamond^{\#\text{steps}=4*24} \text{eod}) = 1$$

It is possible to complete all jobs.

$$\Pr^{\max}(\Diamond(J = 0)) \approx 0.25$$

Don't use grid energy and finish all jobs.

$$\Pr^{\max}(\Box(\text{no\_grid}) \wedge \Diamond(J = 0))$$

$\Diamond$ . . . finally $\Box$ . . . globally eod . . . end of day

Some Results

The end of the day will be reached (when time's up).

$$\Pr^{\min}(\Diamond \text{eod}) = 1 \qquad \Pr^{\min}(\Diamond^{\#\text{steps}=4*24} \text{eod}) = 1$$

It is possible to complete all jobs.

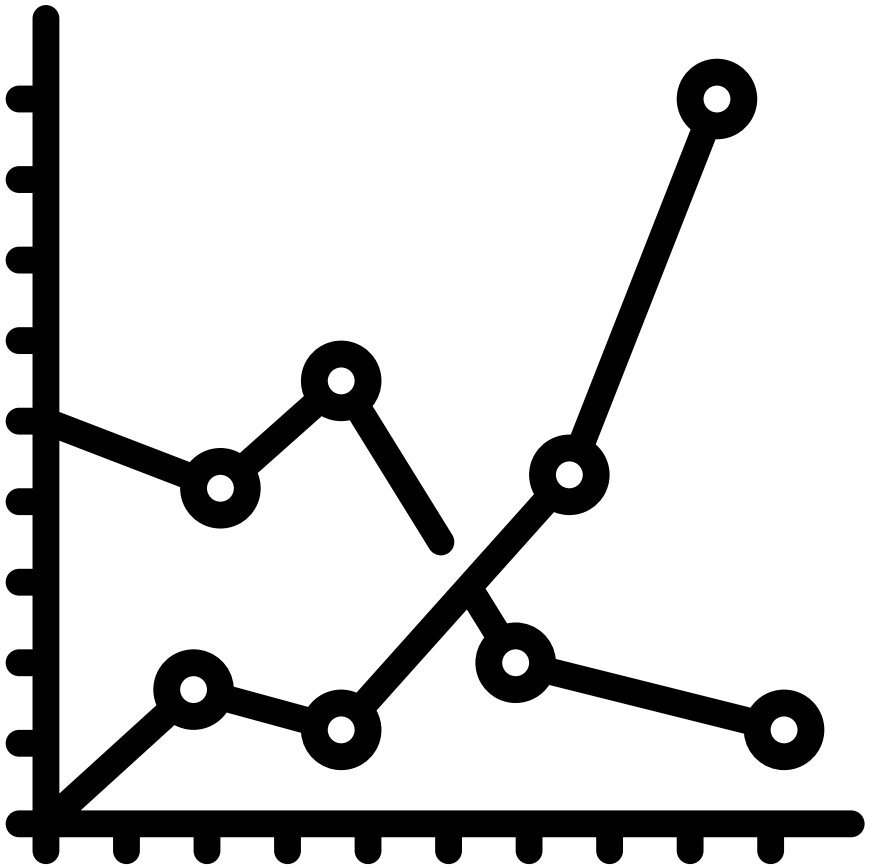
$$\Pr^{\max}(\Diamond(J = 0)) \approx 0.25$$

Don't use grid energy and finish all jobs.

$$\Pr^{\max}(\Box(\text{no\_grid}) \wedge \Diamond(J = 0)) = 0$$

$\Diamond$ . . . finally $\Box$ . . . globally eod . . . end of day

Some Results



Optimization

Some Results

Expected number of jobs until end of day.

Some Results

Expected number of jobs until end of day.

$$\mathbb{E}^{\max}_{\#\text{jobs}}(\Diamond \text{eod}) \approx 230$$

Some Results

Expected number of jobs until end of day.

$$\mathbb{E}^{\max}_{\#\text{jobs}}(\Diamond \text{eod}) \approx 230$$

Expected penalty until end of day.

Some Results

Expected number of jobs until end of day.

$$\mathbb{E}^{\max}_{\#\text{jobs}}(\Diamond \text{eod}) \approx 230$$

Expected penalty until end of day.

$$\mathbb{E}^{\max}_{\text{penalty}}(\Diamond \text{eod}) \qquad \mathbb{E}^{\min}_{\text{penalty}}(\Diamond \text{eod})$$

Some Results

Expected number of jobs until end of day.

$$\mathbb{E}^{\max}_{\#\text{jobs}}(\Diamond \text{eod}) \approx 230$$

Expected penalty until end of day.

$$\mathbb{E}^{\max}_{\text{penalty}}(\Diamond \text{eod}) \approx 51 \qquad \mathbb{E}^{\min}_{\text{penalty}}(\Diamond \text{eod}) \approx 4$$

Some Results

Expected number of jobs until end of day.

$$\mathbb{E}^{\max}_{\#\text{jobs}}(\Diamond \text{eod}) \approx 230$$

Expected penalty until end of day.

$$\mathbb{E}^{\max}_{\text{penalty}}(\Diamond \text{eod}) \approx 51 \qquad \mathbb{E}^{\min}_{\text{penalty}}(\Diamond \text{eod}) \approx 4$$

. . . and the corresponding schedulers.

Some Results

Tradeoff

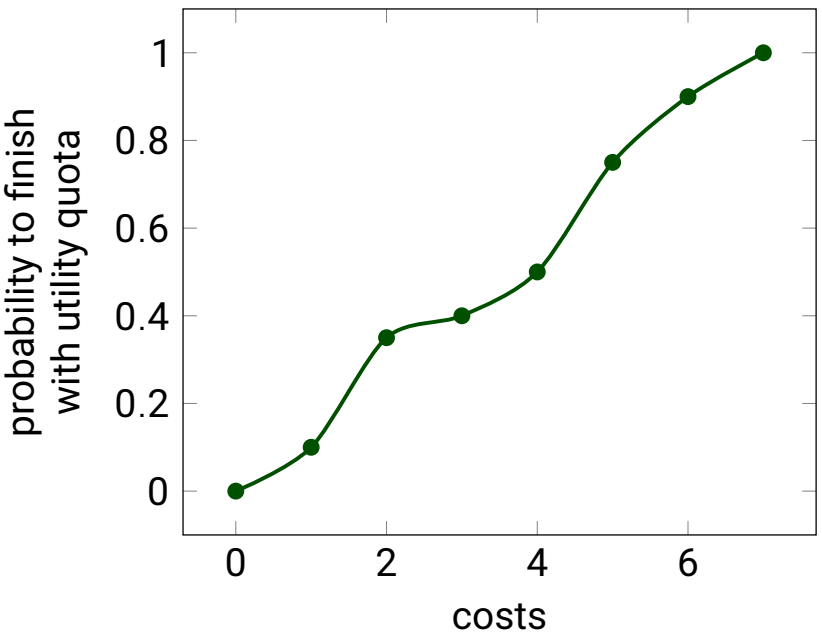
Some Results

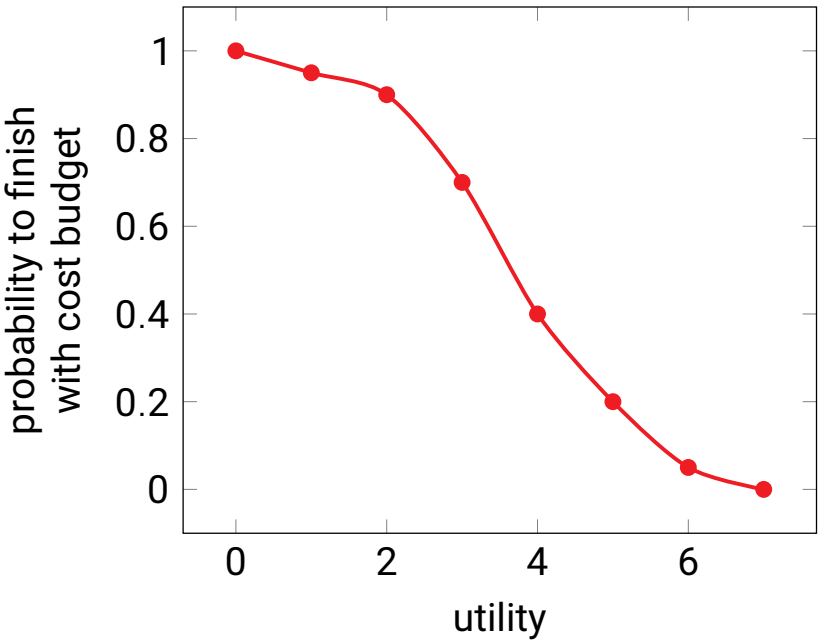
We want **high** utility and **low** costs at the same time.

Some Results

We want **high** utility and **low** costs at the same time.

This is usually **impossible**:

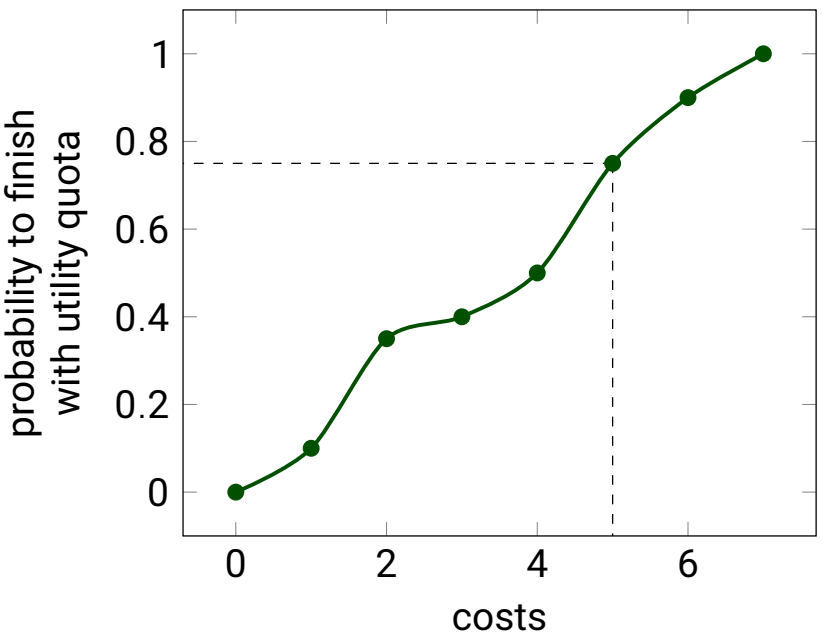


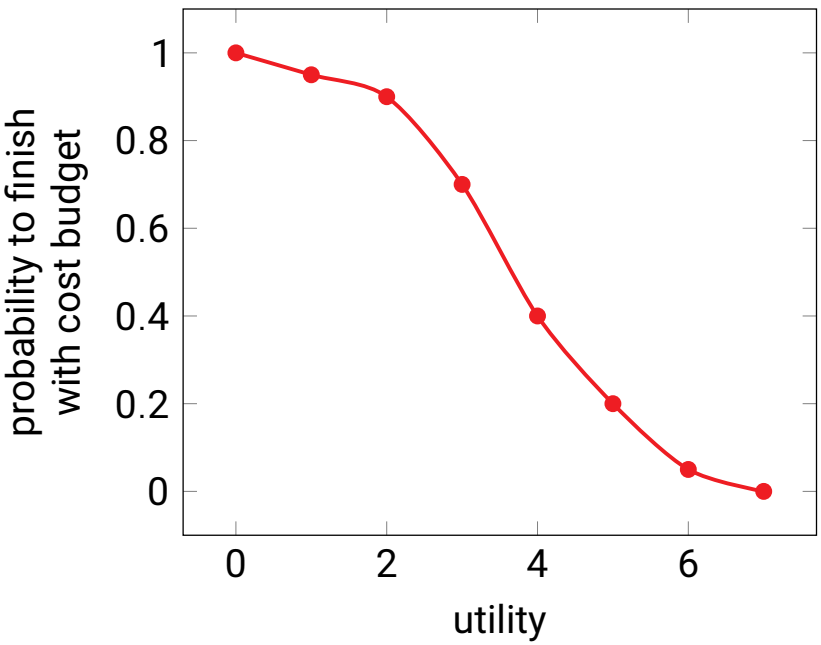


Some Results

We want **high** utility and **low** costs at the same time.

This is usually **impossible**:

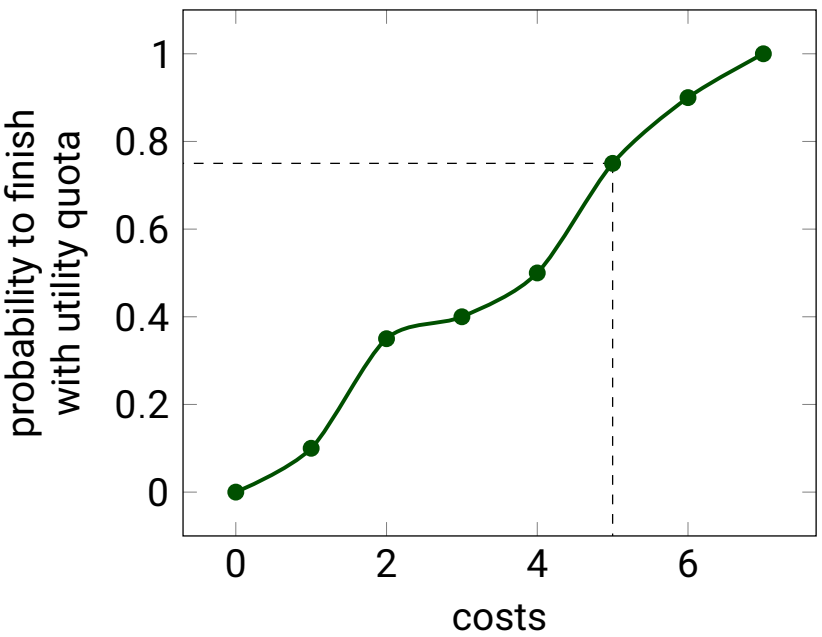


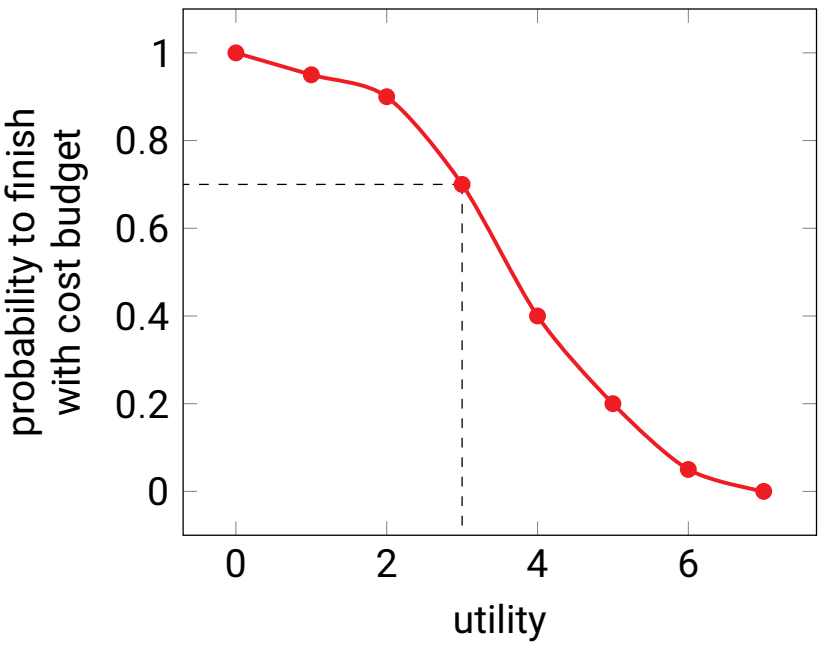


Some Results

We want **high** utility and **low** costs at the same time.

This is usually **impossible**:





Time for a...

DEMO

Conclusions

Probabilistic Model Checking

- fully automated quantitative analysis
- many areas of application
- efficient tools

Our Group

- theoretical questions
- modeling and analysis
- algorithms and implementation