Model Checking ATL

Natasha Alechina   Brian Logan

Utrecht University

n.a.alechina@uu.nl   b.s.logan@uu.nl

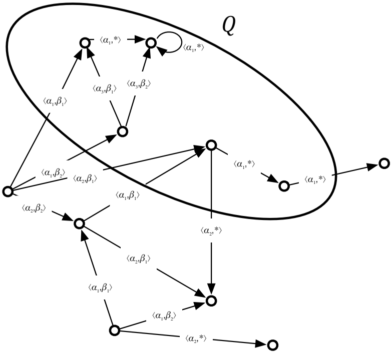
Global model checking ATL

- as in CTL global model checking, we work with sets of states
- for coalition modalities, we need a pre-image, $pre(A, Q)$, that returns the states q where the agents in A can cooperate to ensure that all outcomes q' of their joint action α_A are in Q:

$$pre(A, Q) = \{q \mid \exists \alpha_A \forall \alpha_{Agt \setminus A} o(q, \alpha_A, \alpha_{Agt \setminus A}) \in Q\}$$

Pre-image

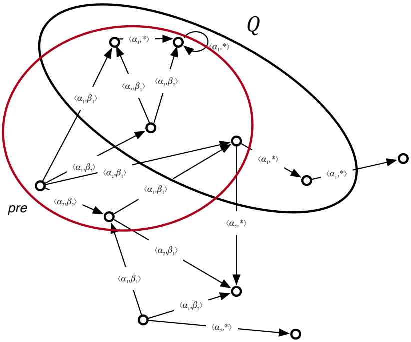
The pre-image of a set states Q for a coalition A, $pre(A, Q)$, is all states q where the agents in A can cooperate to ensure that all outcomes q' of α_A are in Q



$$pre(A, Q) = \{q \mid \exists \alpha_A \forall \alpha_{Agt \setminus A} o(q, \alpha_A, \alpha_{Agt \setminus A}) \in Q\}$$

Pre-image

The pre-image of a set states Q for a coalition A, $pre(A, Q)$, is all states q where the agents in A can cooperate to ensure that all outcomes q' of α_A are in Q



$$pre(A, Q) = \{q \mid \exists \alpha_A \forall \alpha_{Agt \setminus A} o(q, \alpha_A, \alpha_{Agt \setminus A}) \in Q\}$$

ATL model checking algorithm

function MCHECK(M, φ_0)

for $\varphi' \in Sub(\varphi_0)$ **do**

case $\varphi' = p$

$[\varphi']_M \leftarrow \mathcal{V}(p)$

case $\varphi' = \neg\psi$

$[\varphi']_M \leftarrow St \setminus [\psi]_M$

case $\varphi' = \psi_1 \wedge \psi_2$

$[\varphi']_M \leftarrow [\psi_1]_M \cap [\psi_2]_M$

case $\varphi' = \psi_1 \vee \psi_2$

$[\varphi']_M \leftarrow [\psi_1]_M \cup [\psi_2]_M$

case $\varphi' = \langle\langle A \rangle\rangle X \psi$

$[\varphi']_M \leftarrow pre(A, [\psi]_M)$

case $\varphi' = \langle\langle A \rangle\rangle G\, \psi$

$Q_1 \leftarrow St; \quad Q_2 \leftarrow [\psi]_M$

while $Q_1 \not\subseteq Q_2$ **do**

$Q_1 \leftarrow Q_1 \cap Q_2; \quad Q_2 \leftarrow pre(A, Q_1) \cap [\psi]_M$

$[\varphi']_M \leftarrow Q_1$

case $\varphi' = \langle\langle A \rangle\rangle \psi_1\, U\, \psi_2$

$Q_1 \leftarrow \emptyset; \quad Q_2 \leftarrow [\psi_2]_M$

while $Q_2 \not\subseteq Q_1$ **do**

$Q_1 \leftarrow Q_1 \cup Q_2; \quad Q_2 \leftarrow pre(A, Q_1) \cap [\psi_1]_M$

$[\varphi']_M \leftarrow Q_1$

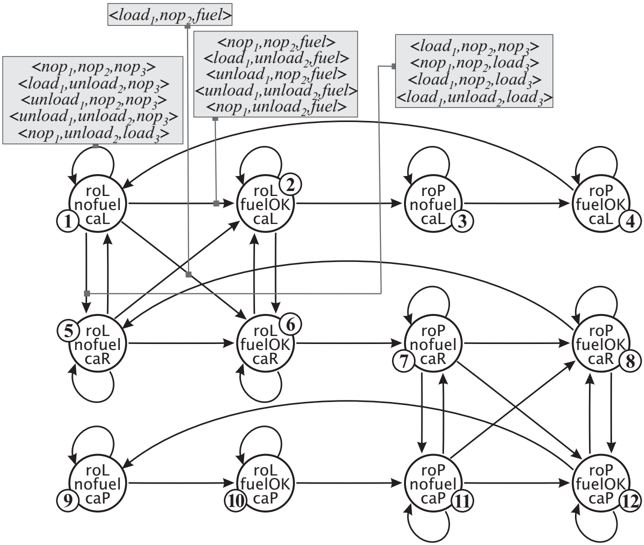
Example: Simple Rocket Domain

- assume that there are 3 workers in the rocket (agents 1, 2, and 3)
- each agent has different capabilities:
    - agent 1 can load the cargo, unload the cargo, initiate the flight, or do nothing (action nop)
    - agent 2 can do unload or nop
    - agent 3 can do load, refill the fuel tank (action fuel), or do nop

Example: Simple Rocket Domain

- if agent 1 initiates the flight, the actions of the other agents have no effect (but the rocket only flies if has fuel)
- if loading is attempted when there is no cargo, nothing happens
- same for unloading when the cargo is not in the rocket, and refilling a full tank
- if different agents try to load and unload at the same time then the majority prevails
- refilling fuel can be done in parallel with loading/unloading

Example: Simple Rocket Domain



Example: Simple Rocket Domain properties

- from which states do agents 1 and 3 have a strategy to move the cargo to London and Paris:

  $\langle\langle 1, 3 \rangle\rangle F \, \text{caP} \wedge \langle\langle 1, 3 \rangle\rangle F \, \text{caL}$

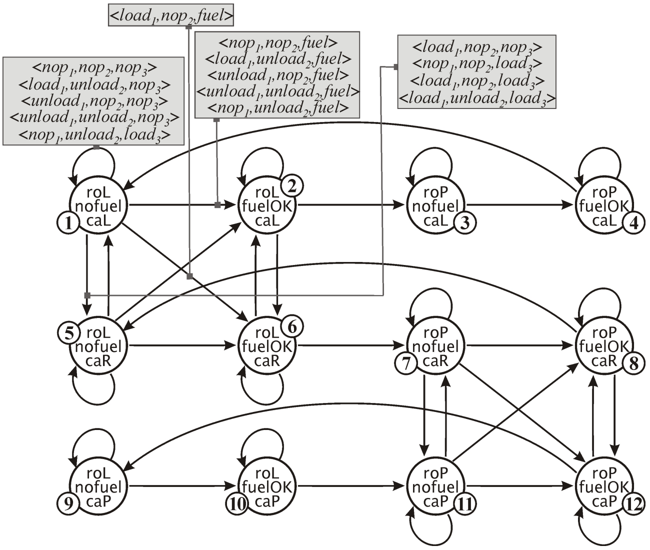
- from which states do agents 1 and 2 have a strategy to move the cargo to Paris:

  $\langle\langle 1, 2 \rangle\rangle F \, \text{caP}$

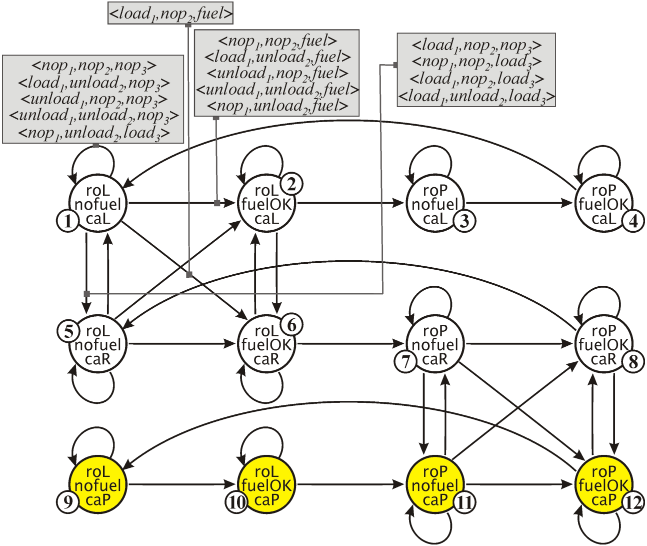
- from which states does agent 3 have a strategy to keep the cargo in Paris forever:

  $\langle\langle 3 \rangle\rangle G \, \text{caP}$

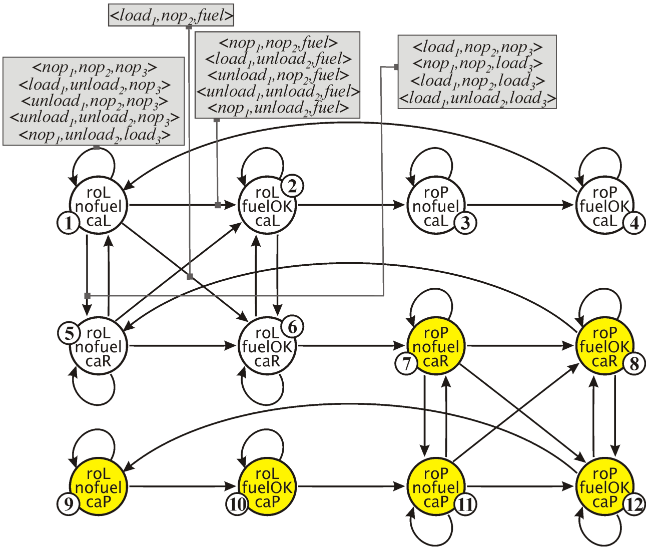
Simple Rocket Domain: verification of $\langle\langle 1,3 \rangle\rangle F \text{caP}$



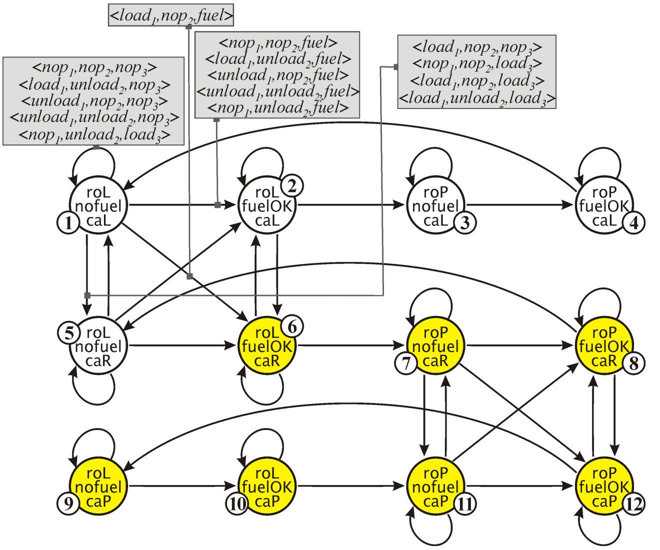
Simple Rocket Domain: verification of $\langle\langle 1,3 \rangle\rangle F\,\mathrm{caP}$



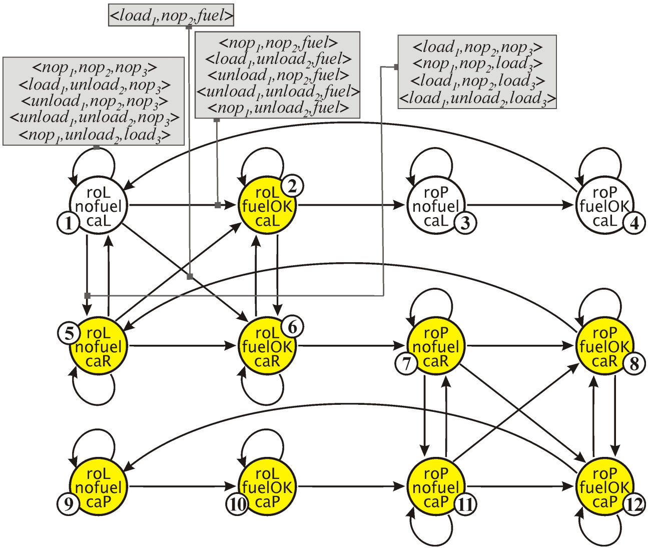
Simple Rocket Domain: verification of $\langle\langle 1,3\rangle\rangle F\,\mathrm{caP}$



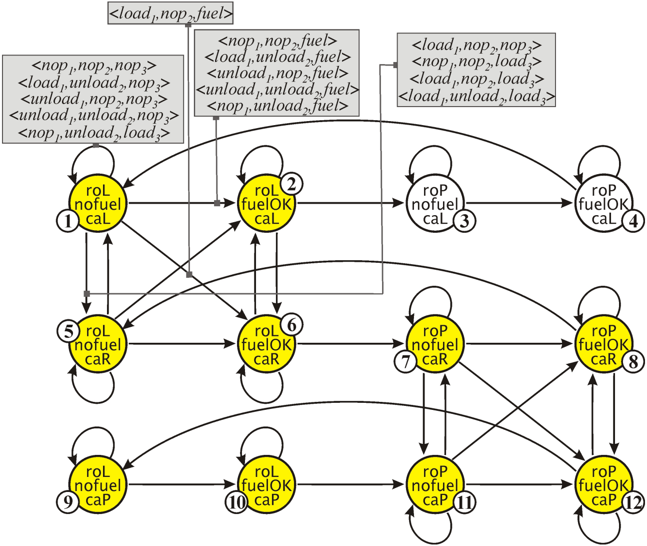
Simple Rocket Domain: verification of $\langle\langle 1, 3\rangle\rangle F\,\mathrm{caP}$



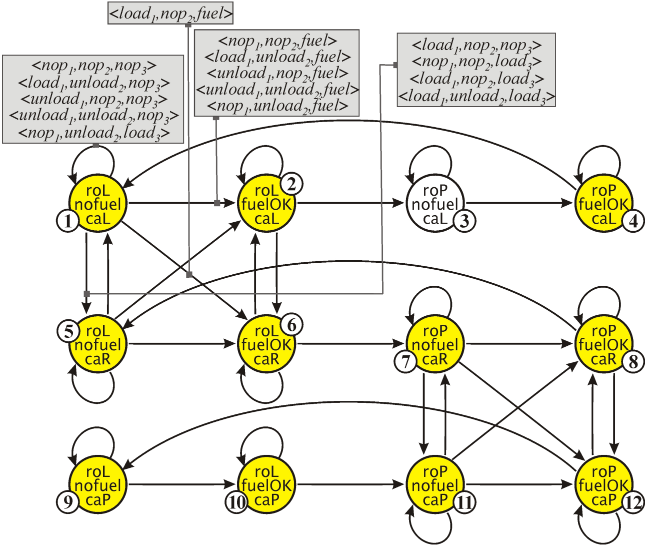
Simple Rocket Domain: verification of $\langle\langle 1, 3\rangle\rangle F\,\text{caP}$



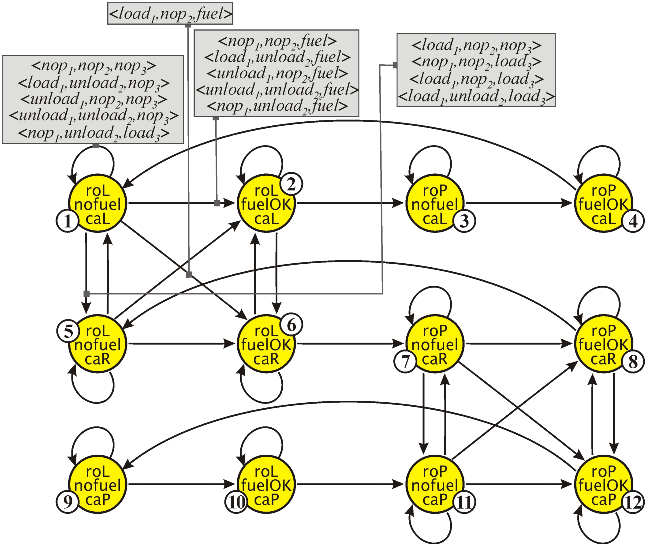
Simple Rocket Domain: verification of $\langle\langle 1, 3\rangle\rangle F\,\mathrm{caP}$



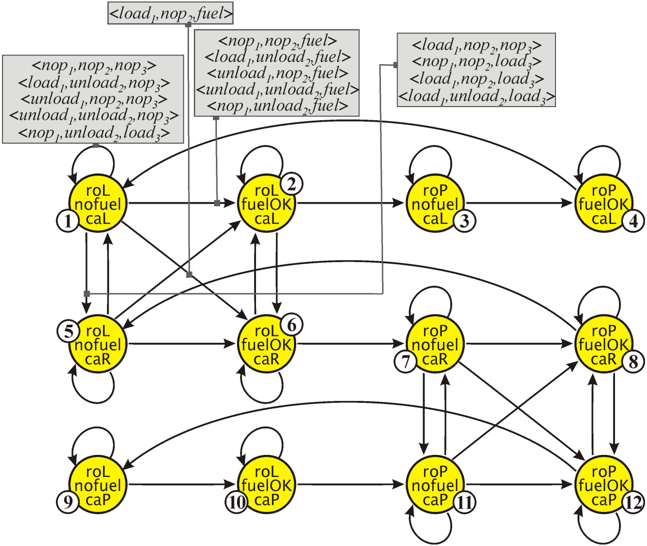
Simple Rocket Domain: verification of $\langle\langle 1,3\rangle\rangle F\,\mathrm{caP}$



Simple Rocket Domain: verification of $\langle\langle 1, 3\rangle\rangle F\,\mathrm{caP}$

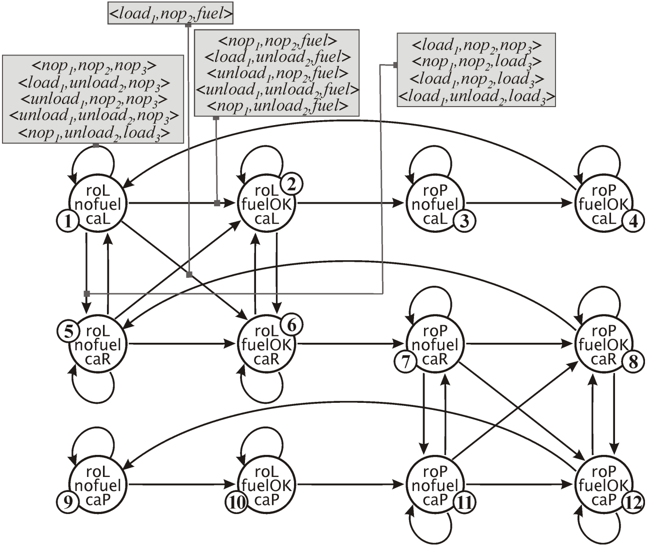


Simple Rocket Domain: verification of $\langle\langle 1,3\rangle\rangle F\,\mathrm{caP}$

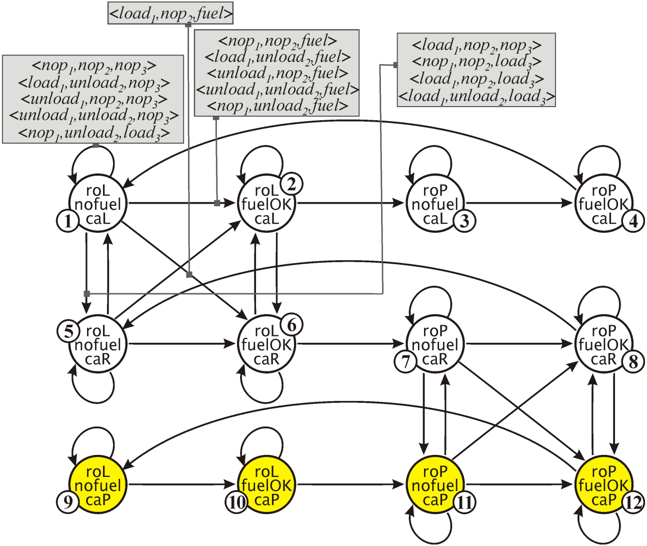


Done!

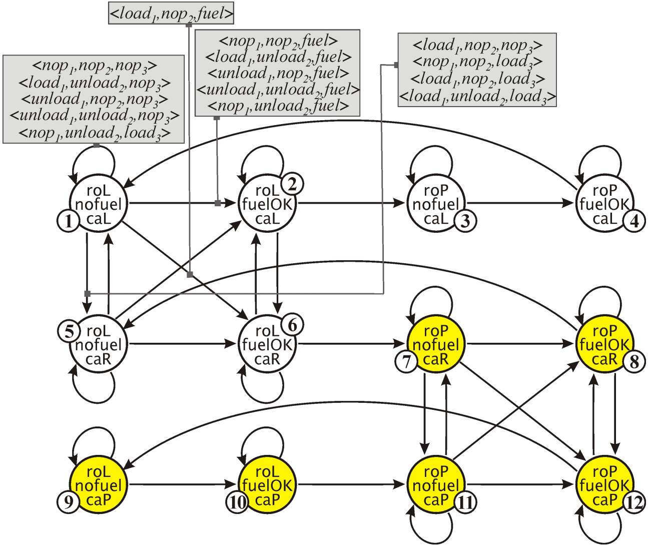
Simple Rocket Domain: verification of $\langle\langle 1, 2\rangle\rangle F\, \mathrm{caP}$



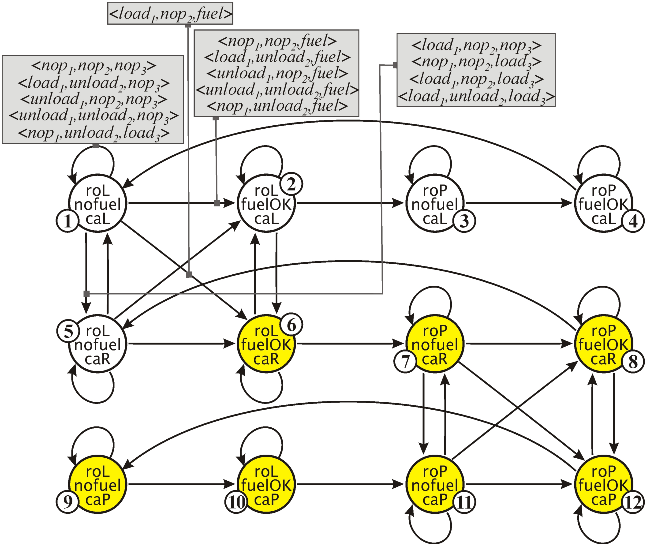
Simple Rocket Domain: verification of $\langle\langle 1, 2\rangle\rangle F\,\mathrm{caP}$



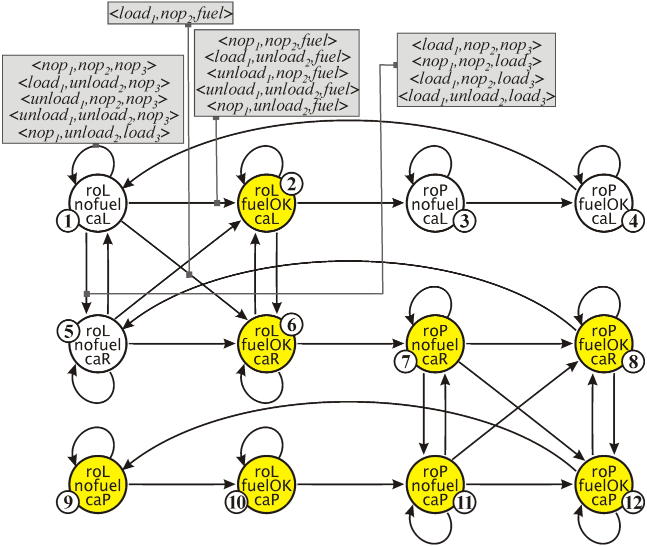
Simple Rocket Domain: verification of $\langle\langle 1, 2\rangle\rangle F\,\mathrm{caP}$



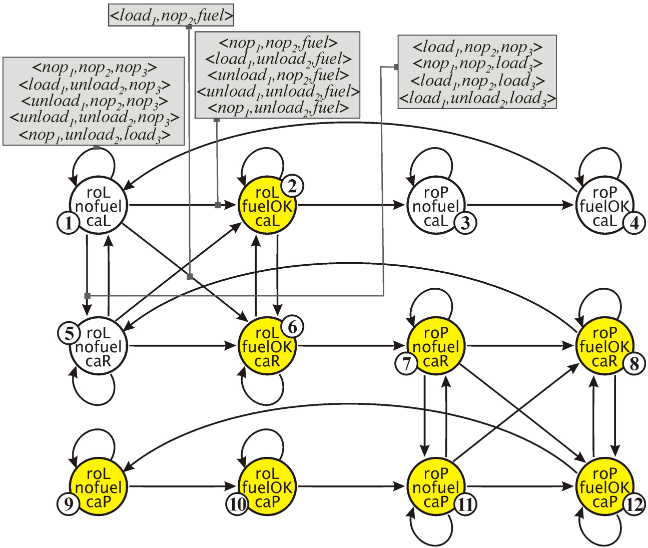
Simple Rocket Domain: verification of $\langle\langle 1, 2\rangle\rangle F\, \mathrm{caP}$



Simple Rocket Domain: verification of $\langle\langle 1, 2 \rangle\rangle F\, \mathrm{caP}$

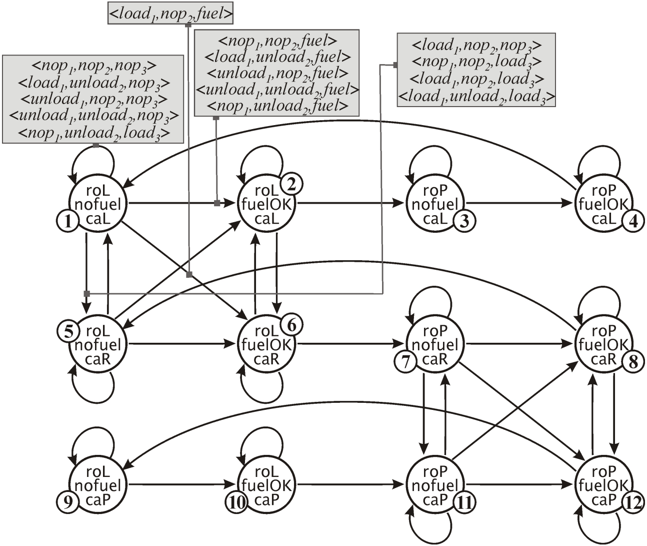


Simple Rocket Domain: verification of $\langle\langle 1, 2\rangle\rangle F\,\mathrm{caP}$

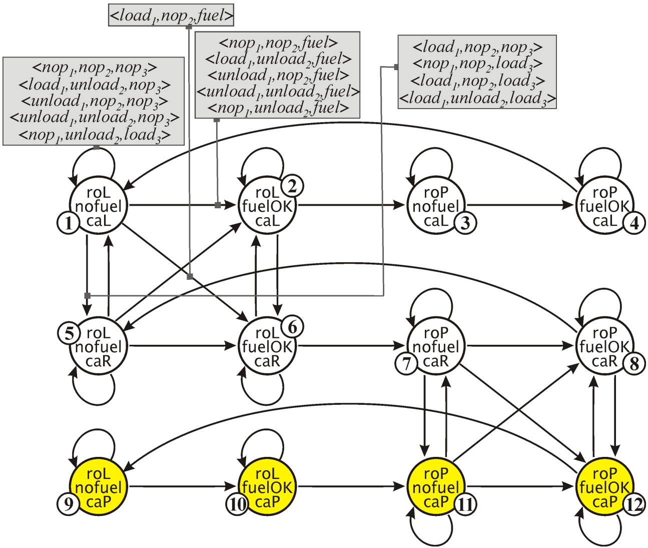


Done!

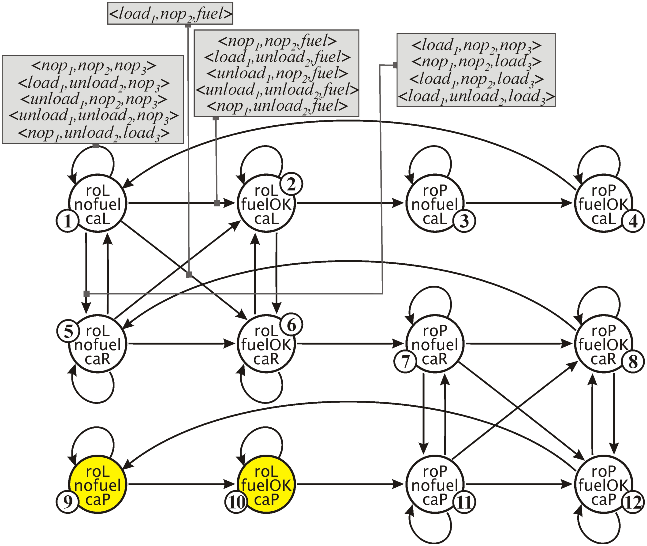
Simple Rocket Domain: verification of $\langle\langle 3 \rangle\rangle G\,$caP



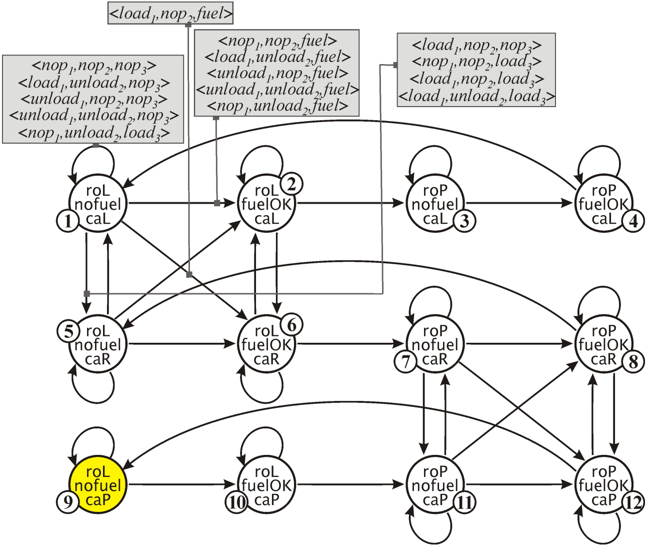
Simple Rocket Domain: verification of $\langle\langle 3 \rangle\rangle G\,\text{caP}$



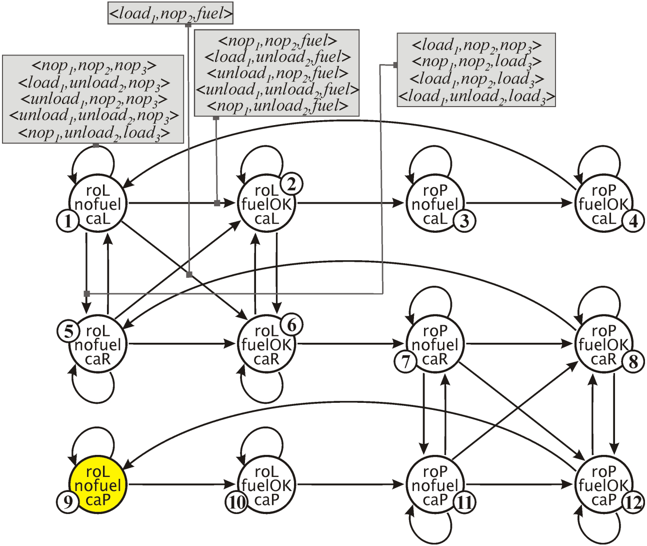
Simple Rocket Domain: verification of $\langle\langle 3\rangle\rangle G\,\mathrm{caP}$



Simple Rocket Domain: verification of $\langle\langle 3 \rangle\rangle G$ caP



Simple Rocket Domain: verification of $\langle\langle 3 \rangle\rangle G\,\mathrm{caP}$



Done!

Model checking ATL: complexity

Theorem (Alur, Kupferman & Henzinger 1998/2002)

Model checking ATL is P-complete, and can be done in time $O(ml)$ where $m =$ #transitions in the model and $l =$ #symbols in the formula.

Model checking temporal and strategic logics: summary

	m, l
CTL	P-complete
LTL	PSPACE-complete
CTL*	PSPACE-complete
ATL	P-complete
ATL*	PSPACE-complete

For strategies with perfect recall:

	m, l
ATL	P-complete
ATL*	2EXPTIME-complete

Synthesis of strategies

- a witness for an ATL formula of the form $\langle\langle A \rangle\rangle \Phi$ (where Φ starts with any of X, G, U) in a model M and state q, is a strategy that makes $\langle\langle A \rangle\rangle \Phi$ true in M, q
- a model checking algorithm can be easily modified to produce a witness for a true $\langle\langle A \rangle\rangle \Phi$ formula
- for example, MCMAS returns a memoryless strategy as a witness
- model checking can be used for the automated synthesis of strategies