Model Checking CTLK

Natasha Alechina Brian Logan

Utrecht University

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

Outline of this lecture

- interpreted systems and ISPL
- model checking CTLK
- more efficient model checking for CTLK

Interpreted systems and ISPL

From interpreted systems to ISPL

- how can we transform an infinite set of infinite runs into something we can model check?
- an interpreted system is essentially a finite state transition system, where the (global) states are tuples of local states of agents (and the environment)
- the state transition system is called the generator of the corresponding interpreted system
- the generator implicitly defines an infinite set of infinite paths

Generator of an interpreted system

- for CTLK, we can evaluate formulas directly in the state transition system (don't need runs, just paths from a state as before)
- indistinguishabilty relations between global states use local states:

$$\langle q_1, ..., q_k \rangle \sim_i \langle q'_1, ..., q'_k \rangle \text{ iff } q_i = q'_i$$

- i.e., two global states $\langle q_1, ..., q_k \rangle$ and $\langle q'_1, ..., q'_k \rangle$ are indistinguishable by agent i if i's local state is the same in both global states

Model checking CTLK is essentially the same as model checking CTL, but with more complex states

Model checking CTLK

Semantics of CTLK knowledge operators

- $\mathcal{M}, q \models E_A\varphi$ iff $\mathcal{M}, q' \models \varphi$ for every q' such that $q \sim_A^E q'$, where $\sim_A^E = \bigcup_{i \in A} \sim_i$

  this means: any step from q along any $\sim_i, i \in A$ ends in a φ-state

- $\mathcal{M}, q \models D_A\varphi$ iff $\mathcal{M}, q' \models \varphi$ for every q' such that $q \sim_A^D q'$, where $\sim_A^D = \bigcap_{i \in A} \sim_i$

  this means: every state accessible by all $\sim_i, i \in A$ satisfies φ

- $\mathcal{M}, q \models C_A\varphi$ iff $\mathcal{M}, q' \models \varphi$ for **every** q' such that $q \sim_A^C q'$, where $\sim_A^C$ is the transitive closure of $\sim_A^E$

  *i.e., $\sim_A^C$ contains **all edges** (q, q') where there exists a path along $\sim_A^E$ from q to q'*

  this means: every path from q along any relation $\sim_i, i \in A$ ends in a φ-state

Global model checking CTLK

- as in CTL global model checking, we work with sets of states
- for (group) knowledge modalities, we need two pre-images: $pre_\exists(rel, Q)$ and $pre_\forall(rel, Q)$:

$$pre_\exists(rel, Q) = \{q \mid \exists q'.q \; rel \; q' \;\&\; q' \in Q\}$$

$$pre_\forall(rel, Q) = \{q \mid \forall q'.q \; rel \; q' \Rightarrow q' \in Q\}$$

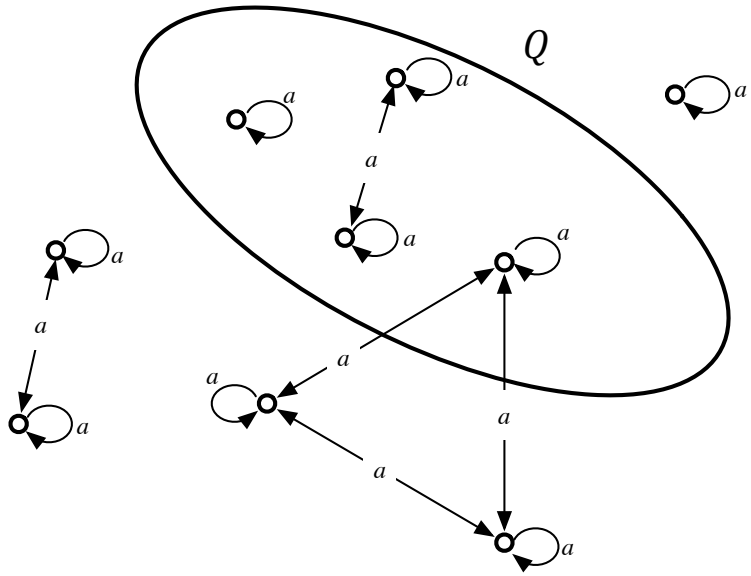
where $rel \in \{\sim_a, \sim_A^E, \sim_A^C, \sim_A^D\}$

Aside

- in the lecture on model checking CTL, $pre_{\exists}(Q)$ and $pre_{\forall}(Q)$ are equivalent to $pre_{\forall}(\rightarrow, Q)$ and $pre_{\exists}(\rightarrow, Q)$ respectively
- in the reader, $pre_{\forall}(\sim_a, Q)$ and $pre_{\exists}(\sim_a, Q)$ are denoted $pre_{\forall}(a, Q)$ and $pre_{\exists}(a, Q)$ respectively

Existential pre-image for $\sim_a$

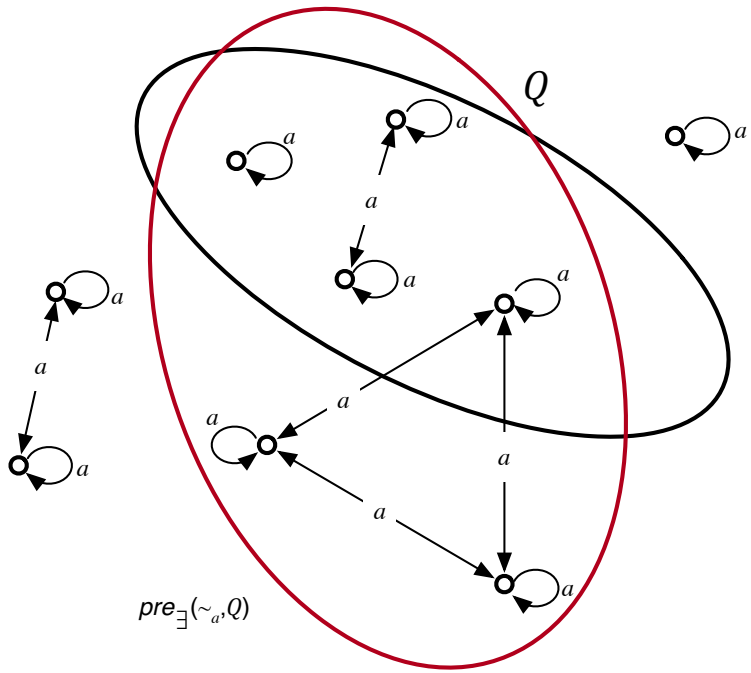
$pre_{\exists}(\sim_a, Q)$ is the set of all states q where there exists some state $q' \in Q$ such that $q \sim_a q'$



$$pre_{\exists}(\sim_a, Q) = \{q \mid \exists q'.q \sim_a q' \ \& \ q' \in Q\}$$

Existential pre-image for $\sim_a$

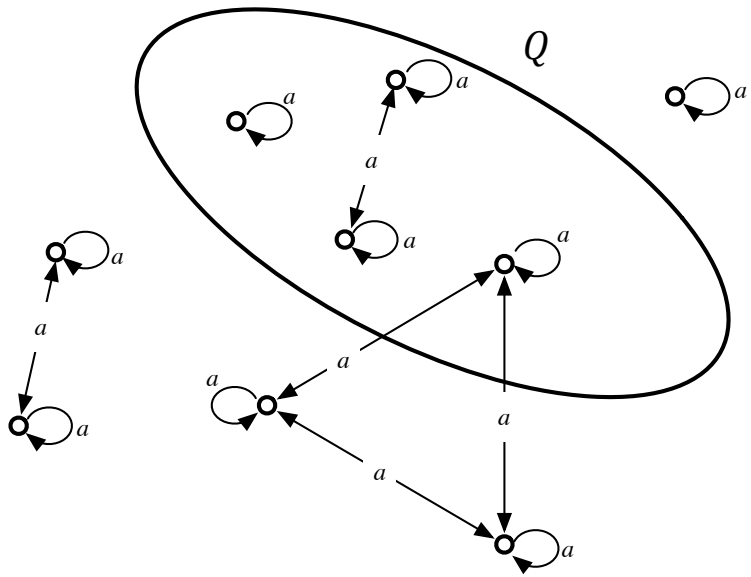
$pre_\exists(\sim_a, Q)$ is the set of all states q where there exists some state $q' \in Q$ such that $q \sim_a q'$



$$pre_\exists(\sim_a, Q) = \{q \mid \exists q'.q \sim_a q' \ \& \ q' \in Q\}$$

Universal pre-image for $\sim_a$

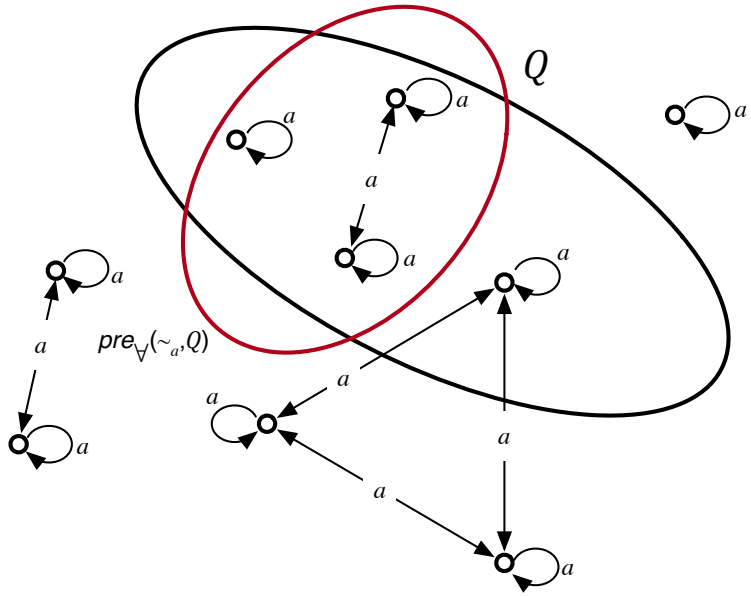
$pre_\forall(\sim_a, Q)$ is the set of all states q where for all states $q' \sim_a q$, $q' \in Q$



$$pre_\forall(\sim_a, Q) = \{q \mid \forall q'.q \sim_a q' \Rightarrow q' \in Q\}$$

Universal pre-image for $\sim_a$

$pre_\forall(\sim_a, Q)$ is the set of all states q where for all states $q' \sim_a q$, $q' \in Q$



$$pre_\forall(\sim_a, Q) = \{q \mid \forall q'.q \sim_a q' \Rightarrow q' \in Q\}$$

Pre-image for $\sim_A^E$ and $\sim_A^D$

- $pre_\forall(\sim_A^E, Q)$ is the set of all states q where for all agents $a \in A$ and all states $q' \sim_a q$, $q' \in Q$

$$pre_\forall(\sim_A^E, Q) = \{q \mid \forall q'.q \sim_A^E q' \Rightarrow q' \in Q\}$$

where $\sim_A^E = \bigcup_{i \in A} \sim_i$

- $pre_\forall(\sim_A^D, Q)$ is the set of all states q where for all states q' such that $q' \sim_a q$ for all $a \in A$, $q' \in Q$

$$pre_\forall(\sim_A^D, Q) = \{q \mid \forall q'.q \sim_A^D q' \Rightarrow q' \in Q\}$$

where $\sim_A^D = \bigcap_{i \in A} \sim_i$

Pre-image for $\sim_A^C$

- is more complicated, as we need to compute the transitive closure of *rel*

function $compute_{\sim_A^C}(M, A)$

$R_1 \leftarrow \emptyset$; $\quad R_2 \leftarrow \bigcup_{i \in A} \sim_i$ $\quad \triangleright$ i.e., R_2 is initialised to $\sim_A^E$

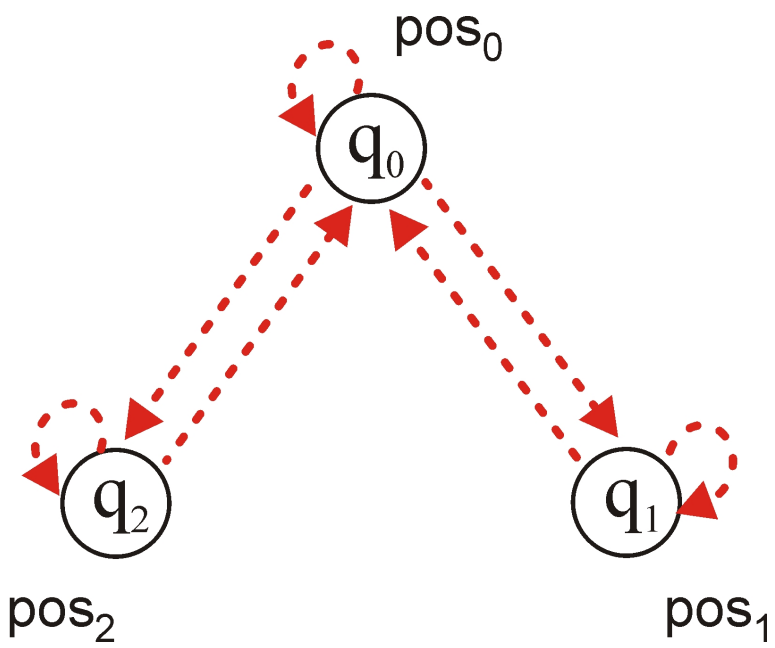
while $R_2 \not\subseteq R_1$ **do**

$R_1 \leftarrow R_1 \cup R_2$; $\quad R_2 \leftarrow \{(q_1, q_2) \mid \exists q_3.(q_1, q_3), (q_3, q_2) \in R_1\}$

return R_1

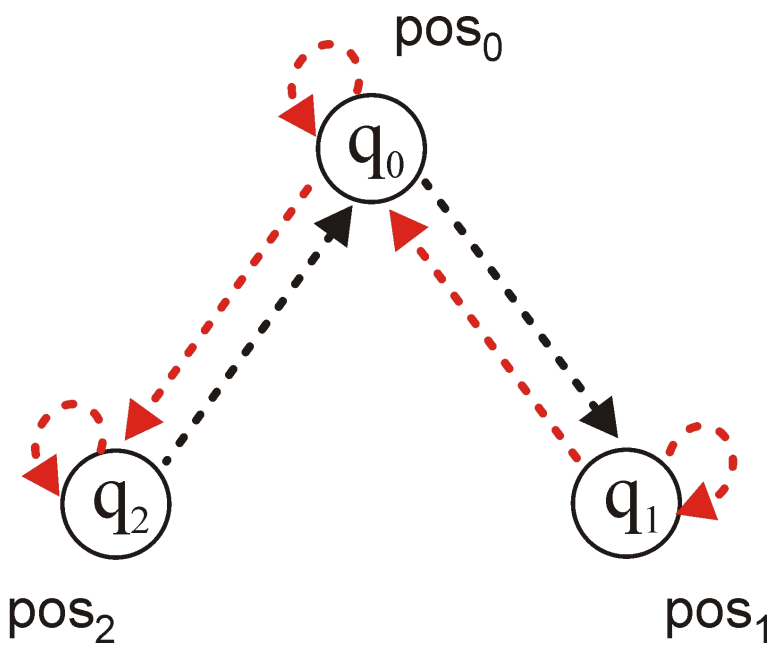
Example: computing $\sim^{C}_{\{1,2\}}$

start with $\sim^{E}_{\{1,2\}}$



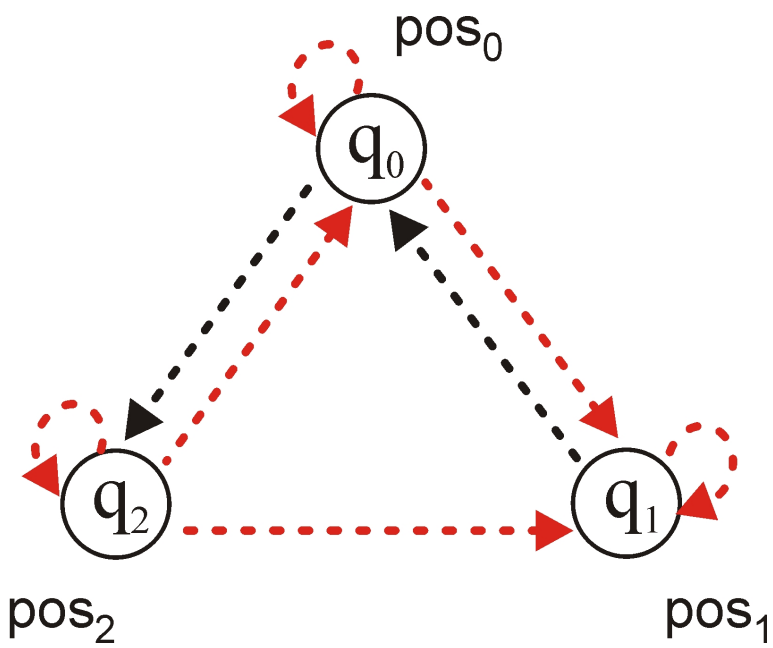
Example: computing $\sim^C_{\{1,2\}}$

transitive closure: $(q_2, q_0) \in \sim^C_{\{1,2\}}, (q_0, q_1) \in \sim^C_{\{1,2\}}$



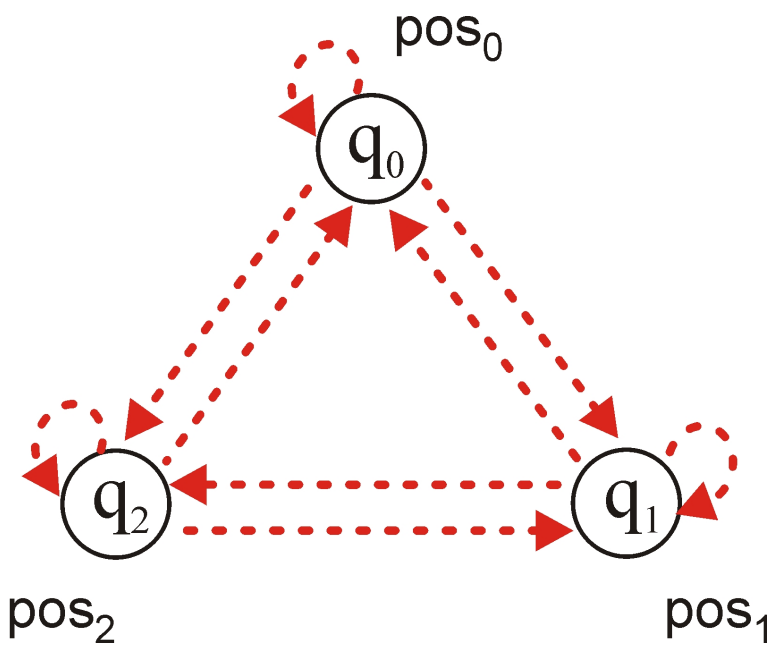
Example: computing $\sim^{C}_{\{1,2\}}$

transitive closure $(q_2, q_0) \in \sim^{C}_{\{1,2\}}, (q_0, q_1) \in \sim^{C}_{\{1,2\}} \Rightarrow (q_2, q_1) \in \sim^{C}_{\{1,2\}}$



Example: computing $\sim^{C}_{\{1,2\}}$

transitive closure: $(q_1, q_0) \in \sim^{C}_{\{1,2\}}, (q_0, q_2) \in \sim^{C}_{\{1,2\}} \Rightarrow (q_1, q_2) \in \sim^{C}_{\{1,2\}}$



function $mcheck_{CTLK}(M, \varphi_0)$

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 $\varphi' = K_a\psi$

$[\varphi']_M \leftarrow pre_\forall(\sim_a, [\psi]_M)$

case $\varphi' = E_A\psi$

$\sim_A^E \leftarrow \bigcup_{i\in A} \sim_i$

$[\varphi']_M \leftarrow pre_\forall(\sim_A^E, [\psi]_M)$

case $\varphi' = D_A\psi$

$\sim_A^D \leftarrow \bigcap_{i\in A} \sim_i$

$[\varphi']_M \leftarrow pre_\forall(\sim_A^D, [\psi]_M)$

case $\varphi' = C_A\psi$

$\sim_A^C \leftarrow compute_{\sim_A^C}(M, A)$

$[\varphi']_M \leftarrow pre_\forall(\sim_A^C, [\psi]_M)$

case $\varphi' = EX\psi$

$[\varphi']_M \leftarrow pre_\exists([\psi]_M)$

case $\varphi' = EG\,\psi$

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

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

$Q_1 \leftarrow Q_2; \quad Q_2 \leftarrow pre_\exists(Q_1) \cap Q_1$

$[\varphi']_M \leftarrow Q_1$

case $\varphi' = E\psi_1\, U\, \psi_2$

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

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

$Q_1 \leftarrow Q_1 \cup Q_2; \quad Q_2 \leftarrow pre_\exists(Q_1) \cap [\psi_1]_M$

$[\varphi']_M \leftarrow Q_1$

case $\varphi' = AX\psi$

$[\varphi']_M \leftarrow pre_\forall([\psi]_M)$

case $\varphi' = AG\,\psi$

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

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

$Q_1 \leftarrow Q_2; \quad Q_2 \leftarrow pre_\forall(Q_1) \cap Q_1$

$[\varphi']_M \leftarrow Q_1$

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

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

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

$Q_1 \leftarrow Q_1 \cup Q_2; \quad Q_2 \leftarrow pre_\forall(Q_1) \cap [\psi_1]_M$

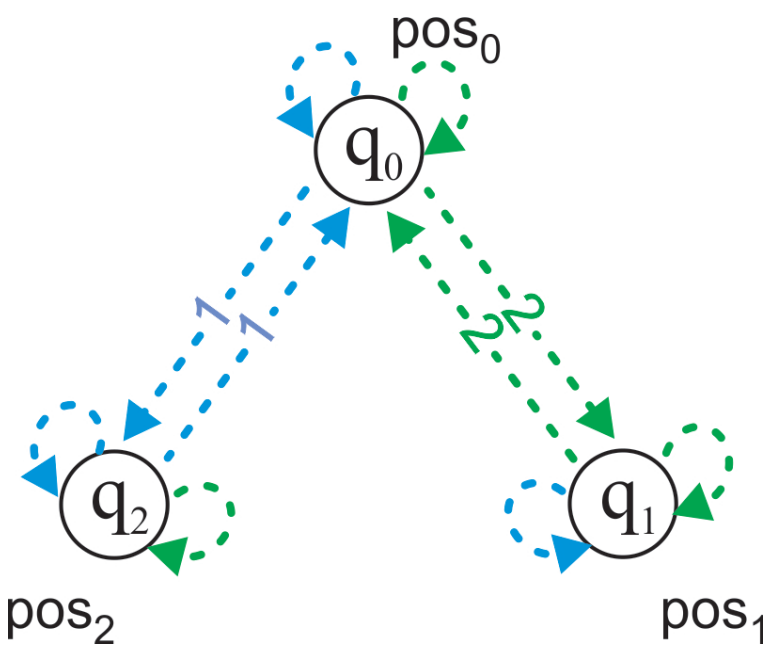
$[\varphi']_M \leftarrow Q_1$

Where did the runs go?

- the semantics of $K_i\varphi$ (and the other epistemic operators) is defined in terms of runs
- the model checker implementation doesn't refer to runs or time points — what's going on?
- the model checker works in terms of the set of states in the model
- these are all the states that could appear at any moment on any run
- the *pre* functions effectively consider all the time points that could occur!

Example: Robots and Carriage

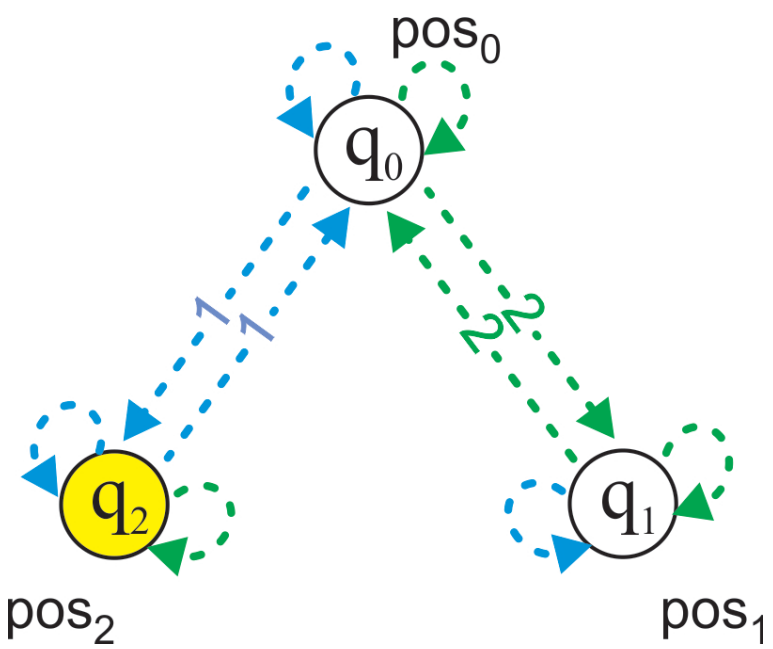
Task: compute $mcheck_{CTLK}(M, K_1 \neg \text{pos}_2)$



$K_1 \neg \text{pos}_2$

Example: Robots and Carriage

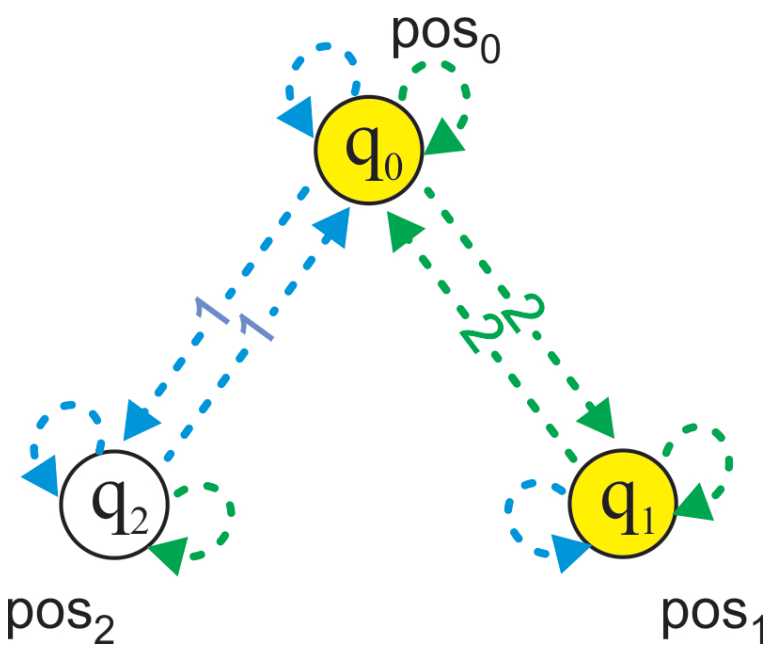
Task: compute $mcheck_{CTLK}(M, K_1 \neg \text{pos}_2)$



$K_1 \neg \text{pos}_2$

Example: Robots and Carriage

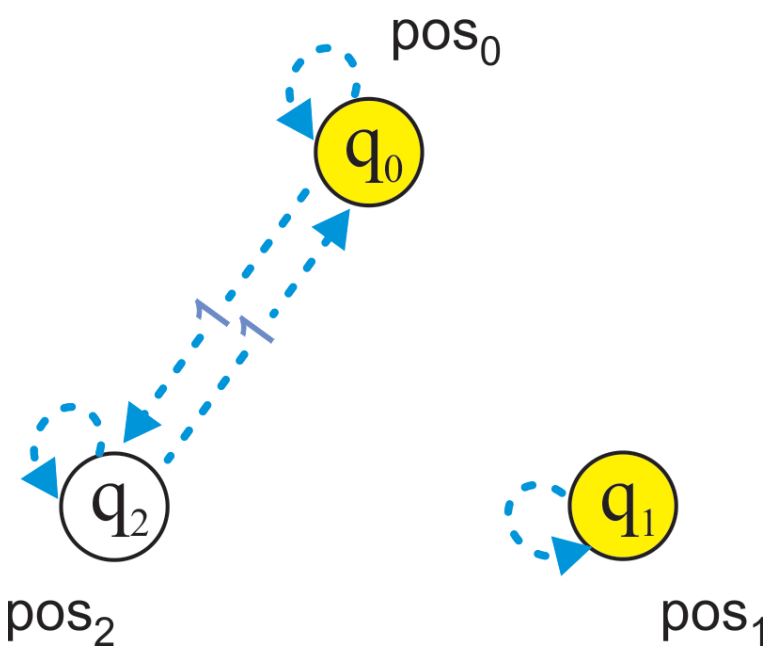
Task: compute $mcheck_{CTLK}(M, K_1 \neg \text{pos}_2)$



$K_1 \neg \text{pos}_2$

Example: Robots and Carriage

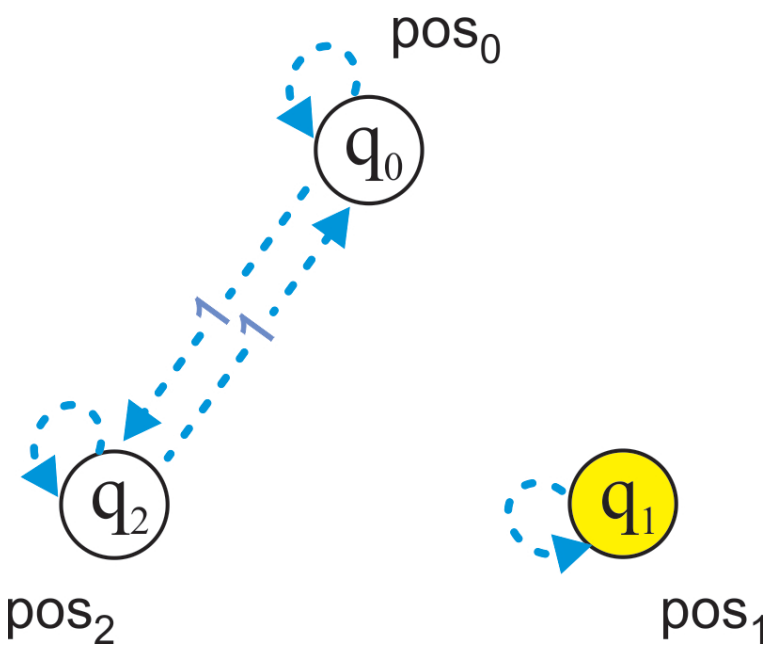
Task: compute $mcheck_{CTLK}(M, K_1 \neg \text{pos}_2)$



$K_1 \neg \text{pos}_2$

Example: Robots and Carriage

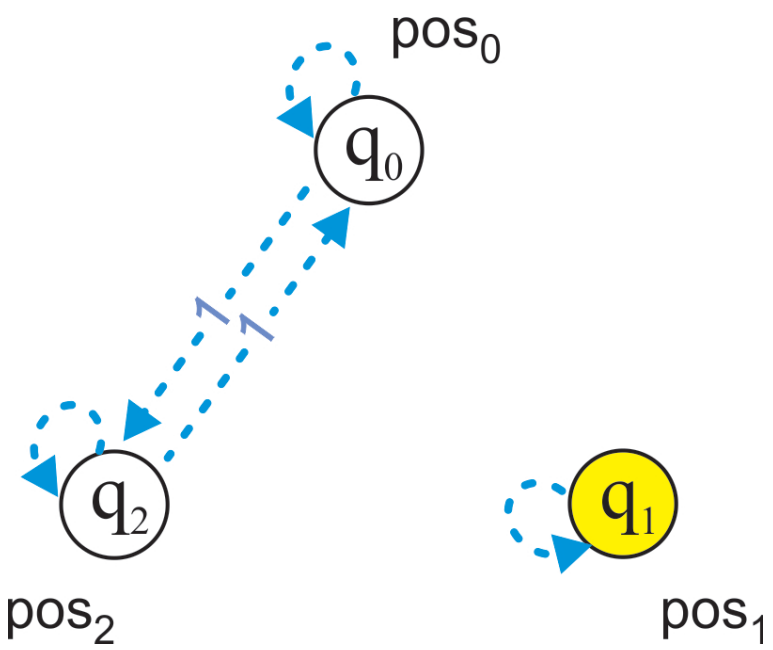
Task: compute $mcheck_{CTLK}(M, K_1 \neg \text{pos}_2)$



$K_1 \neg \text{pos}_2$

Example: Robots and Carriage

Task: compute $mcheck_{CTLK}(M, K_1 \neg \text{pos}_2)$

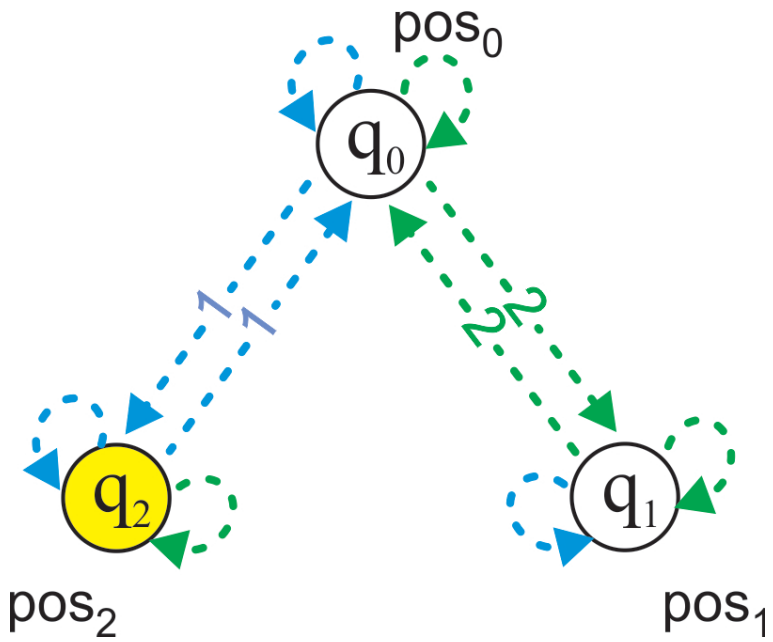


$K_1 \neg \text{pos}_2$

Done!

Example: $mcheck(M, C_{1,2}\neg \text{pos2})$

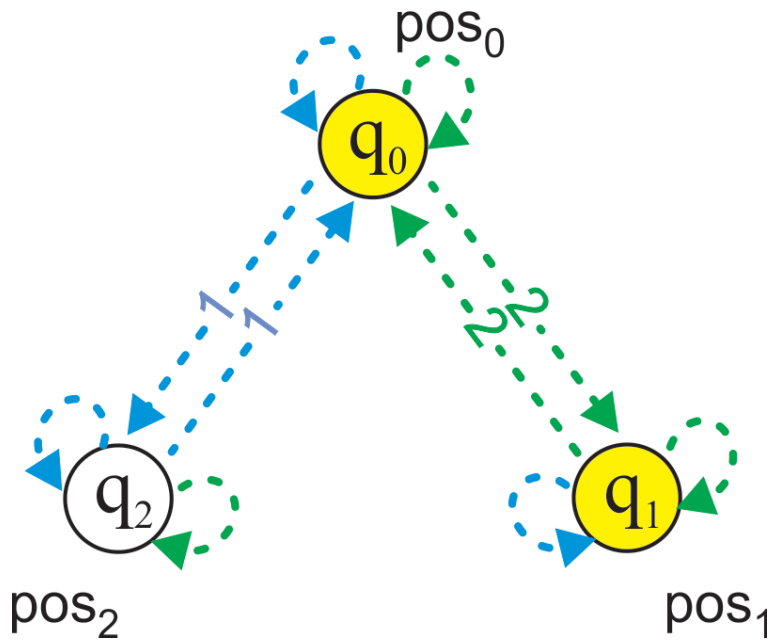
start with pos_2



$C_{\{1,2\}}\neg \text{pos}_2$

Example: $mcheck(M, C_{1,2}\neg \text{pos2})$

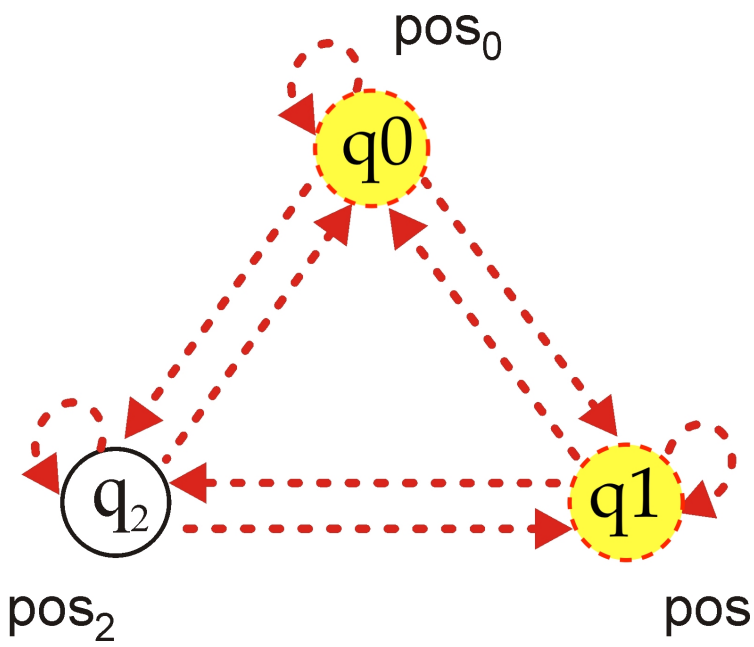
labelling $\neg \text{pos}_2$



$C_{\{1,2\}}\neg \text{pos}_2$

Example: $mcheck(M, C_{1,2}\neg \text{pos2})$

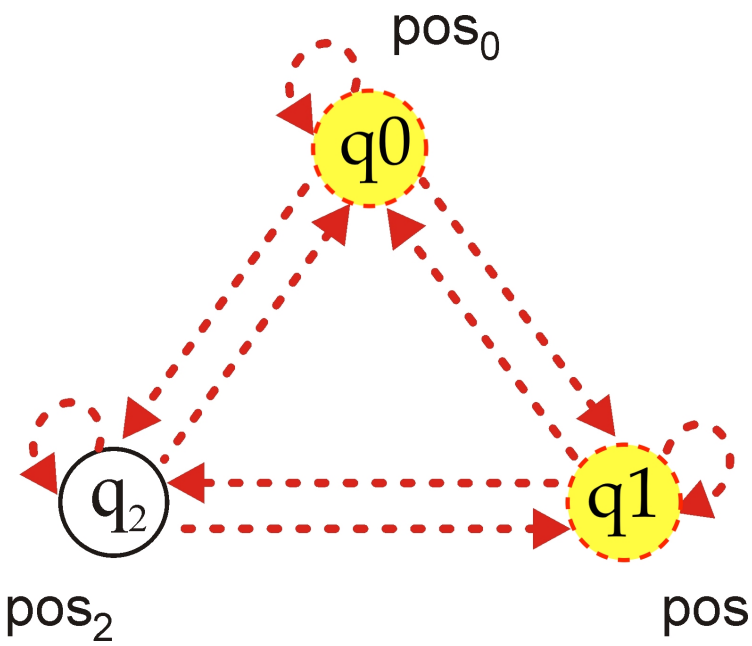
compute $\sim^{C}_{\{1,2\}}$



$C_{\{1,2\}}\neg \text{pos}_2$

Example: $mcheck(M, C_{1,2}\neg \text{pos2})$

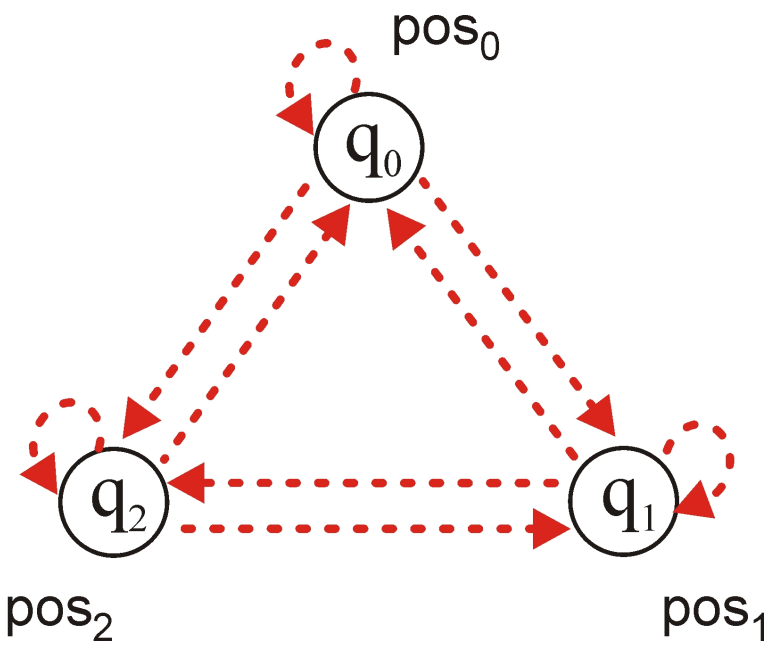
take $pre_{\forall}(\sim^{C}_{\{1,2\}}, \{q_0, q_1\})$



$C_{\{1,2\}}\neg \text{pos}_2$

Example: $mcheck(M, C_{1,2}\neg \text{pos2})$

$$pre_{\forall}(\sim^{C}_{\{1,2\}}, \{q_0, q_1\}) = \emptyset$$



$$[C_{\{1,2\}}\neg \text{pos}_2]_M = \emptyset$$

More efficient model checking for CTLK

More efficient model checking with dual operators

- this gives a polynomial time (quadratic) algorithm for model checking CTLK — however we can do better ...
- it is more efficient to model-check dual versions of (group) knowledge modalities using only the existential pre-image
- $\overline{K}_i\varphi := \neg K_i \neg \varphi$ translates as 'agent i considers φ possible'
- $\mathcal{M}, q \models \overline{K}_i\varphi$ iff exists q' such that $q \sim_i q'$ and $\mathcal{M}, q' \models \varphi$
- i.e., there is some step from q along $\sim_i$ that ends in a φ-state
- the model-checking algorithm only needs to consider $\overline{K}_i\varphi$, because $K_i\varphi$ is definable as $\neg\overline{K}_i\neg\varphi$

Semantics of $\overline{E}_A$

- $\overline{E}_A\varphi := \neg E_A \neg\varphi$ translates as 'at least one of agents in A considers φ possible'
- $\mathcal{M}, q \models \overline{E}_A\varphi$ iff exists q' such that $q \sim_A^E q'$ and $\mathcal{M}, q' \models \varphi$ where $\sim_A^E = \bigcup_{i \in A} \sim_i$
- i.e., there is some step from q along any of $\sim_i$ for $i \in A$ that ends in a φ-state
- the model-checking algorithm only needs to consider $\overline{E}_A\varphi$, because $E_A\varphi$ is definable as $\neg\overline{E}_A\neg\varphi$

Semantics of $\overline{D}_A$

- $\overline{D}_A\varphi := \neg D_A \neg\varphi$ translates as 'φ is a possibility given distributed knowledge in A' (none of the agents in A rules φ out)
- $\mathcal{M}, q \models \overline{D}_A\varphi$ iff exists q' with $q \sim_A^D q'$ such that $\mathcal{M}, q' \models \varphi$, where $\sim_A^D = \bigcap_{i \in A} \sim_i$
- i.e., some state accessible by all $\sim_i, i \in A$ satisfies φ
- $D_A\varphi$ can be replaced by $\neg\overline{D}_A\neg\varphi$

Semantics of $\overline{C}_A$

- $\overline{C}_A\varphi := \neg C_A \neg\varphi$ translates approximately as 'someone in A considers possible that someone else in A considers possible etc. that φ
- $\mathcal{M}, q \models \overline{C}_A\varphi$ iff for **some** q' with $q \sim_A^C q'$, $\mathcal{M}, q' \models \varphi$, where $\sim_A^C$ is the transitive closure of $\sim_A^E$
- i.e., some path from q composed of relations $\sim_i$ for $i \in A$ ends in a φ-state
- again we can replace all occurrences of $C_A\varphi$ with $\neg\overline{C}_A\neg\varphi$ and just give a model checking case for $\overline{C}_A\varphi$

function $\overline{mcheck}_{CTLK}(M, \varphi_0)$

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 $\varphi' = \overline{K}_a\psi$

$[\varphi']_M \leftarrow pre_\exists(\sim_a, [\psi]_M)$

case $\varphi' = \overline{E}_A\psi$

$[\varphi']_M \leftarrow \bigcup_{i\in A} pre_\exists(\sim_i, [\psi]_M)$

case $\varphi' = \overline{D}_A\psi$

$[\varphi']_M \leftarrow \bigcap_{i\in A} pre_\exists(\sim_i, [\psi]_M)$

case $\varphi' = \overline{C}_A\psi$

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

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

$Q_1 \leftarrow Q_1 \cup Q_2; \quad Q_2 \leftarrow \bigcup_{i\in A} pre_\exists(\sim_i, Q_2)$

$[\varphi']_M \leftarrow Q_1$

case $\varphi' = EX\psi$

$[\varphi']_M \leftarrow pre_\exists([\psi]_M)$

case $\varphi' = EG\,\psi$

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

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

$Q_1 \leftarrow Q_2; \quad Q_2 \leftarrow pre_\exists(Q_1) \cap Q_1$

$[\varphi']_M \leftarrow Q_1$

case $\varphi' = E\psi_1\, U\, \psi_2$

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

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

$Q_1 \leftarrow Q_1 \cup Q_2; \quad Q_2 \leftarrow pre_\exists(Q_1) \cap [\psi_1]_M$

$[\varphi']_M \leftarrow Q_1$

case $\varphi' = AX\psi$

$[\varphi']_M \leftarrow pre_\forall([\psi]_M)$

case $\varphi' = AG\,\psi$

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

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

$Q_1 \leftarrow Q_2; \quad Q_2 \leftarrow pre_\forall(Q_1) \cap Q_1$

$[\varphi']_M \leftarrow Q_1$

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

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

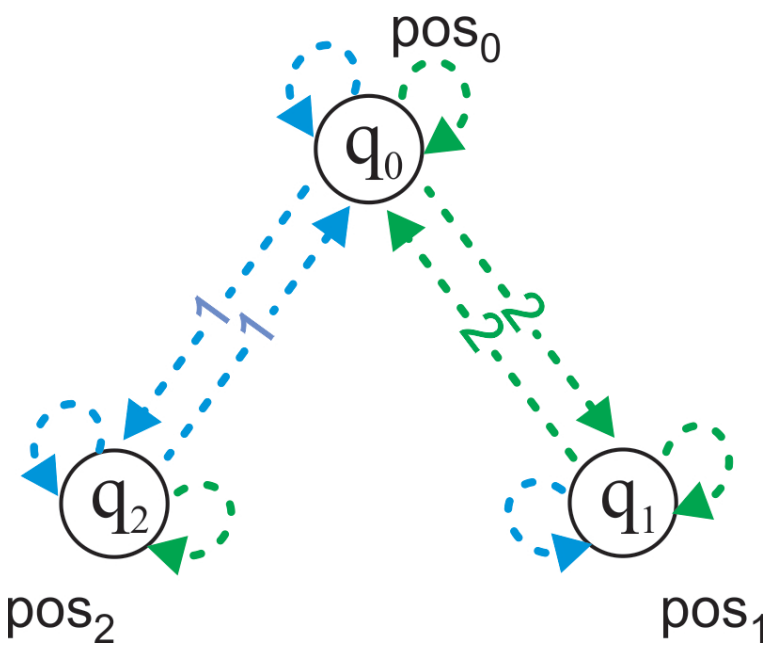
while $Q_2 \not\subseteq Q_1$ **do**

$Q_1 \leftarrow Q_1 \cup Q_2; \quad Q_2 \leftarrow pre_\forall(Q_1) \cap [\psi_1]_M$

$[\varphi']_M \leftarrow Q_1$

Example: Robots and Carriage

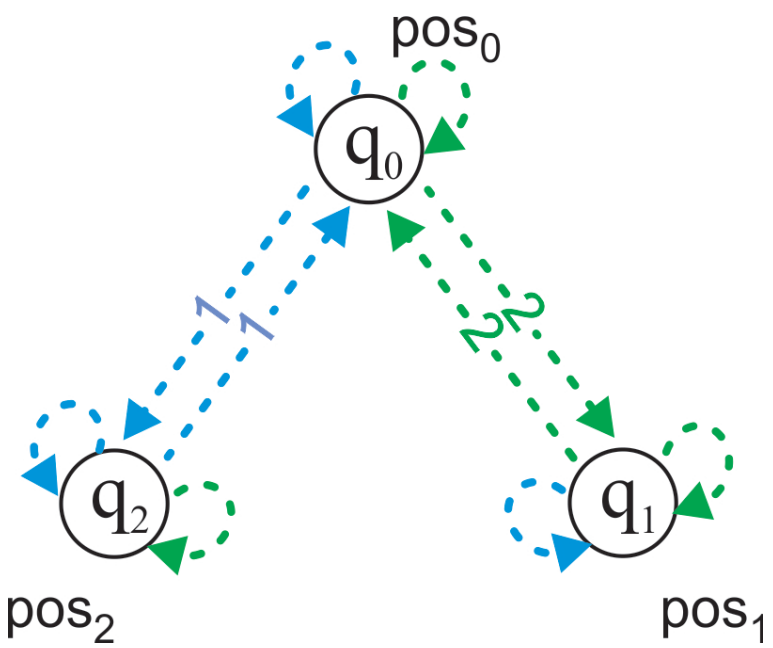
Compute $mcheck_{CTLK}(M, \neg K_1 \text{pos}_2)$ using dual $\overline{K}_1 \neg \text{pos}_2$



$\neg K_1 \text{pos}_2$

Example: Robots and Carriage

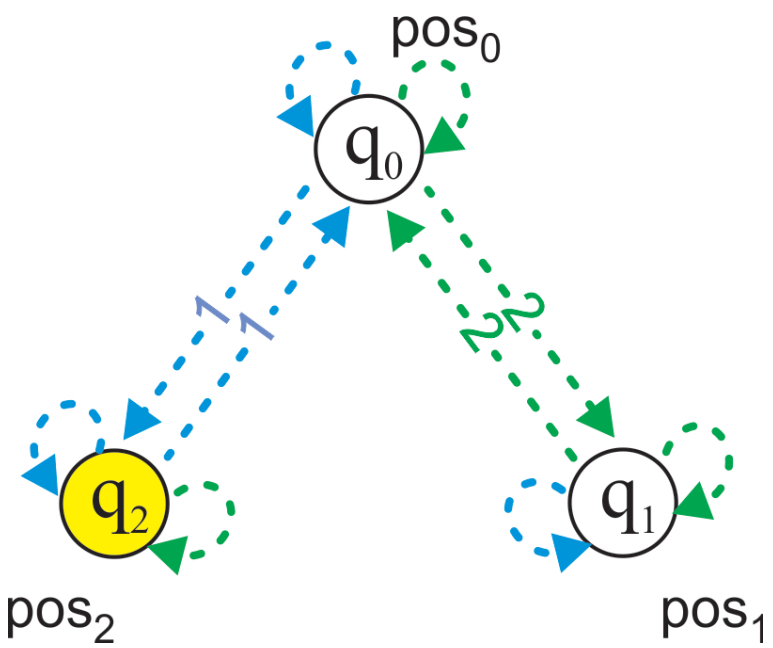
Compute $\overline{mcheck}_{CTLK}(M, \overline{K}_1 \neg \text{pos}_2)$



$\overline{K}_1 \neg \text{pos}_2$

Example: Robots and Carriage

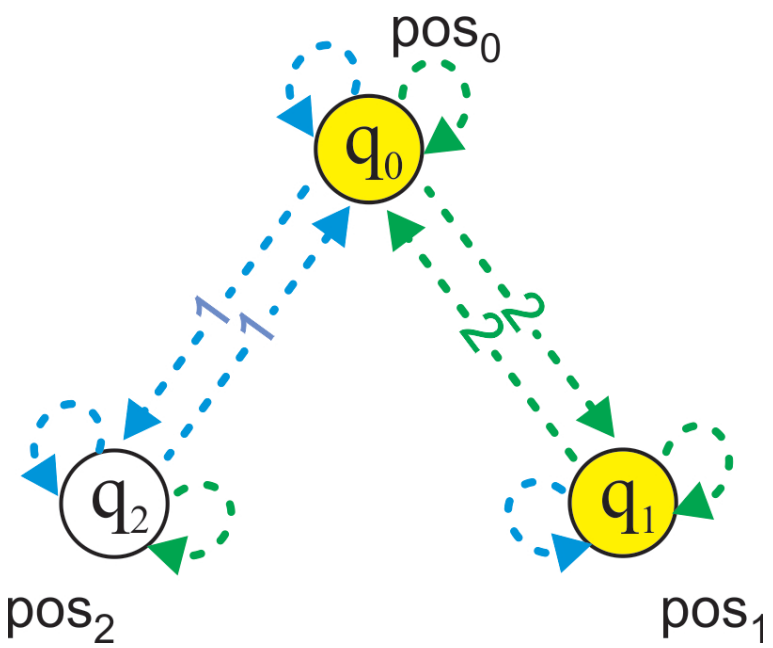
Compute $\overline{mcheck}_{CTLK}(M, \overline{K}_1 \neg \text{pos}_2)$



$\overline{K}_1 \neg \text{pos}_2$

Example: Robots and Carriage

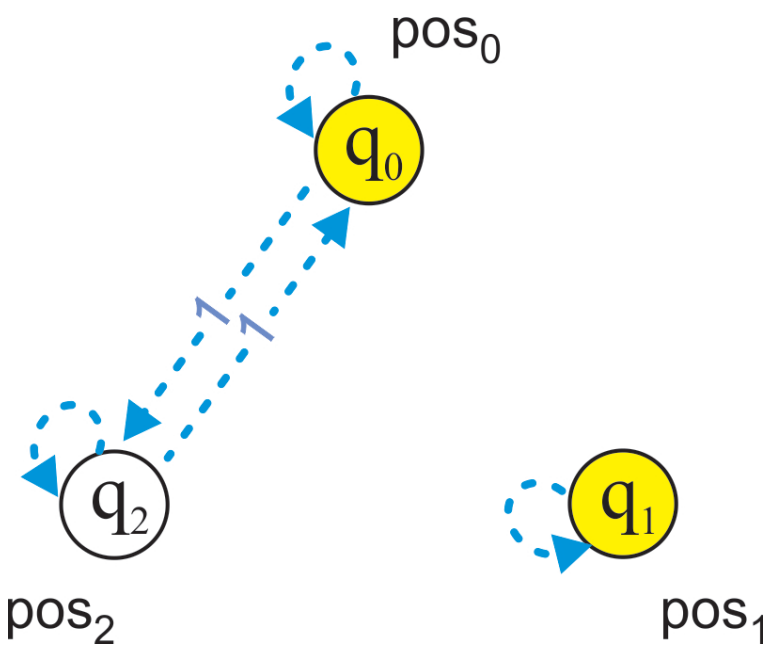
Compute $\overline{mcheck}_{CTLK}(M, \overline{K}_1 \neg \text{pos}_2)$



$\overline{K}_1 \neg \text{pos}_2$

Example: Robots and Carriage

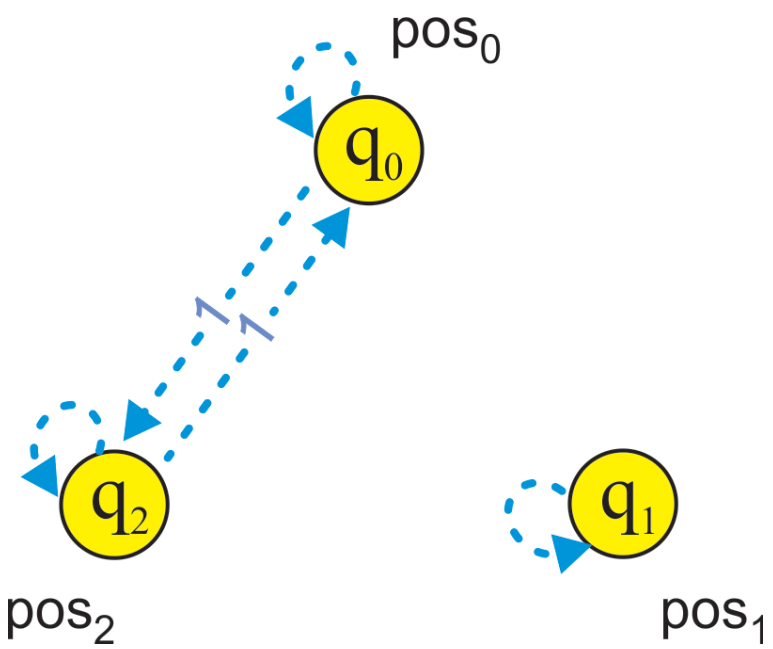
Compute $\overline{mcheck}_{CTLK}(M, \overline{K}_1 \neg \text{pos}_2)$



$\overline{K}_1 \neg \text{pos}_2$

Example: Robots and Carriage

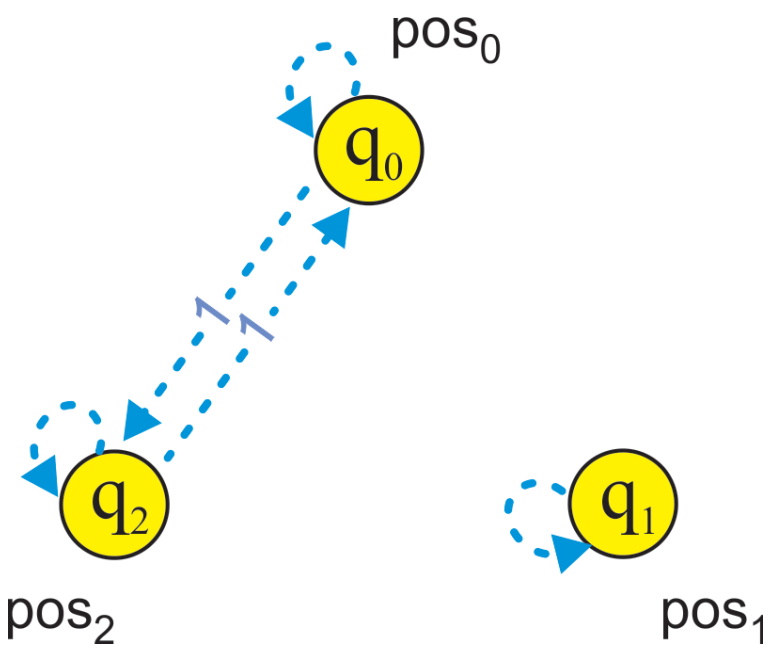
Compute $\overline{mcheck}_{CTLK}(M, \overline{K}_1 \neg \text{pos}_2)$



$\overline{K}_1 \neg \text{pos}_2$

Example: Robots and Carriage

Compute $\overline{mcheck}_{CTLK}(M, \overline{K}_1 \neg \text{pos}_2)$



$\overline{K}_1 \neg \text{pos}_2$

Done!

Complexity of model checking for CTLK

Theorem

Model checking of CTLK can be done in linear time with respect to the size of the Kripke model and the length of the formula and is P-complete.