Reducing CTL-Live Model Checking to First-Order Logic Validity Checking

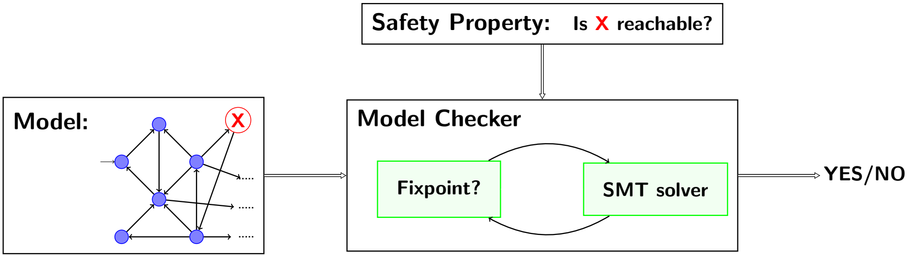
Amirhossein Vakili and Nancy A. Day

Cheriton School of Computer Science

24 October 2014

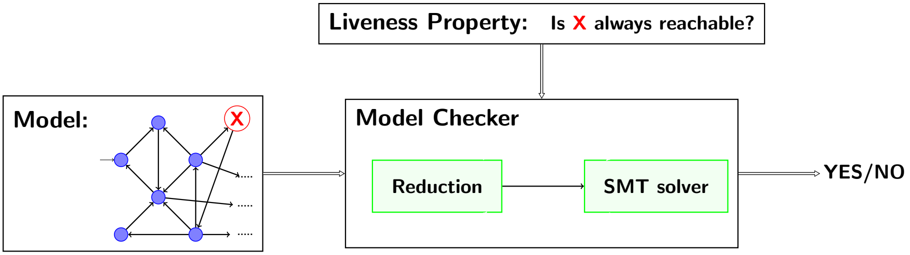
UNIVERSITY OF
Waterloo

Model Checking based on SAT/SMT Solving



- Focus on safety properties
- Iteratively calls the solver

Our Result: CTL-Live Model Checking as FOL Validity



- Focus on liveness properties
- Solved by first-order logic deduction techniques (e.g., SMT solvers)
- No need for abstraction or invariant generation

CTL-Live

CTL-Live includes CTL connectives that are defined using *the least fixpoint operator* of mu-calculus.

Temporal part		
φ	$::=$	$\pi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2$
	$::=$	$\mathbf{EX}\varphi \mid \mathbf{AX}\varphi \mid \mathbf{EF}\varphi \mid \mathbf{AF}\varphi$
	$::=$	$\varphi_1\mathbf{EU}\varphi_2 \mid \varphi_1\mathbf{AU}\varphi_2$
Propositional part		
π	$::=$	$P \mid \neg\pi \mid \pi_1 \vee \pi_2$
	where P is a labelling predicate.	

CTL-Live

CTL-Live includes CTL connectives that are defined using *the least fixpoint operator* of mu-calculus.

Temporal part		
φ	$::=$	$\pi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2$
	$::=$	$\mathbf{EX}\varphi \mid \mathbf{AX}\varphi \mid \mathbf{EF}\varphi \mid \mathbf{AF}\varphi$
	$::=$	$\varphi_1 \mathbf{EU} \varphi_2 \mid \varphi_1 \mathbf{AU} \varphi_2$
Propositional part		
π	$::=$	$P \mid \neg\pi \mid \pi_1 \vee \pi_2$
	where P is a labelling predicate.	

In CTL-Live

- $\mathbf{AF}\ P$
- $(\mathbf{EF}\neg P)\ \mathbf{AU}\ (\mathbf{AX}Q)$

CTL-Live

CTL-Live includes CTL connectives that are defined using *the least fixpoint operator* of mu-calculus.

Temporal part		
φ	$::=$	$\pi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2$
	$::=$	$\mathbf{EX}\varphi \mid \mathbf{AX}\varphi \mid \mathbf{EF}\varphi \mid \mathbf{AF}\varphi$
	$::=$	$\varphi_1\mathbf{EU}\varphi_2 \mid \varphi_1\mathbf{AU}\varphi_2$
Propositional part		
π	$::=$	$P \mid \neg\pi \mid \pi_1 \vee \pi_2$
	where P is a labelling predicate.	

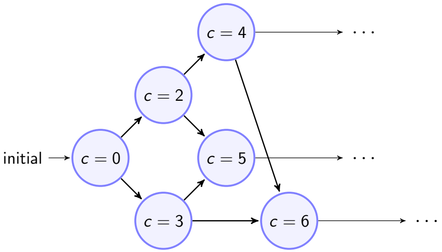
In CTL-Live

- $\mathbf{AF}\ P$
- $(\mathbf{EF}\neg P)\ \mathbf{AU}\ (\mathbf{AX}Q)$

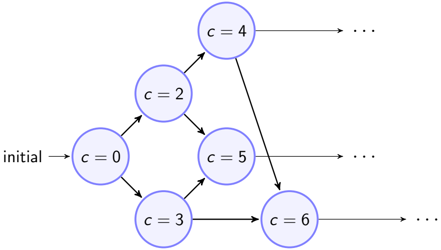
Not In CTL-Live

- $\neg(\mathbf{AF}\ P)$
- $\mathbf{AG}\ P$

Symbolic Kripke Structures in FOL

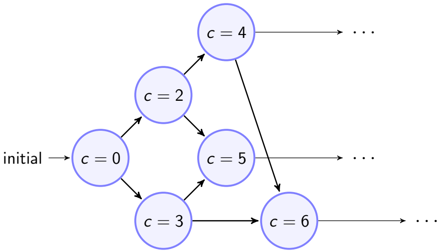


Symbolic Kripke Structures in FOL



- $S = \{0,\ 1,\ 2,\ 3, ..\}$ `state space`
- $S_0(c) \quad \Leftrightarrow \quad c = 0$ `initial states`
- $N(c, c') \quad \Leftrightarrow \quad c' = c + 2 \vee c' = c + 3$ `next-state relation`

Symbolic Kripke Structures in FOL



- $S = \{0,\ 1,\ 2,\ 3, ..\}$ `state space`
- $S_0(c) \quad \Leftrightarrow \quad c = 0$ `initial states`
- $N(c, c') \quad \Leftrightarrow \quad c' = c + 2 \vee c' = c + 3$ `next-state relation`

Notation

- $symbolic(K) \models_c \mathbf{AF}\ c > 3$
- $[\mathbf{AF}\ c > 3] = \{0, 1, 2,\ \ldots\}$

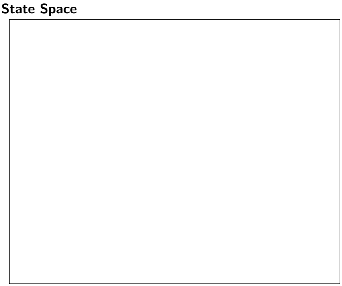
Intuition: States Satisfying **AF** P

According to encoding of **AF** in mu-calculus, [**AF** P] is the **smallest** set Y that satisfies:

$$\textbf{(1)} \forall s \bullet \; P(s) \;\Rightarrow\; Y(s)$$
$$\textbf{(2)} \forall s \bullet \; \left(\forall s' \bullet \; N(s,s') \Rightarrow \; Y(s')\right) \Rightarrow Y(s)$$

According to encoding of **AF** in mu-calculus, [**AF** P] is the **smallest** set Y that satisfies:

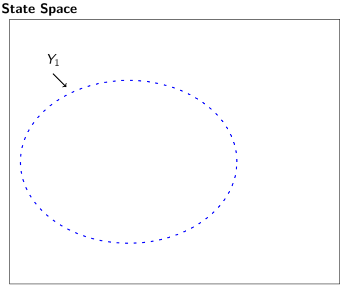
$$\textbf{(1)} \forall s \bullet \; P(s) \; \Rightarrow \; Y(s)$$
$$\textbf{(2)} \forall s \bullet \; (\forall s' \bullet \; N(s, s') \Rightarrow \; Y(s')) \Rightarrow Y(s)$$



Intuition: States Satisfying **AF** P

According to encoding of **AF** in mu-calculus, [**AF** P] is the **smallest** set Y that satisfies:

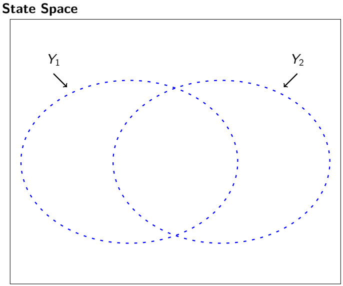
$$(1)\forall s \bullet \; P(s) \;\Rightarrow\; Y(s)$$
$$(2)\forall s \bullet \; \left(\forall s' \bullet \; N(s,s') \Rightarrow \; Y(s')\right) \Rightarrow Y(s)$$



Intuition: States Satisfying **AF** P

According to encoding of **AF** in mu-calculus, [**AF** P] is the **smallest** set Y that satisfies:

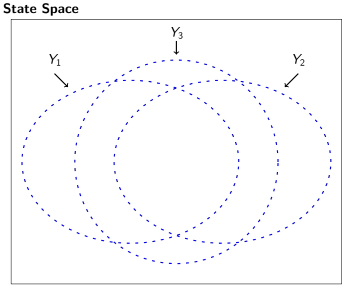
$$\textbf{(1)} \forall s \bullet \ P(s) \ \Rightarrow \ Y(s)$$
$$\textbf{(2)} \forall s \bullet \ \left(\forall s' \bullet \ N(s, s') \Rightarrow \ Y(s')\right) \Rightarrow Y(s)$$



Intuition: States Satisfying **AF** P

According to encoding of **AF** in mu-calculus, [**AF** P] is the **smallest** set Y that satisfies:

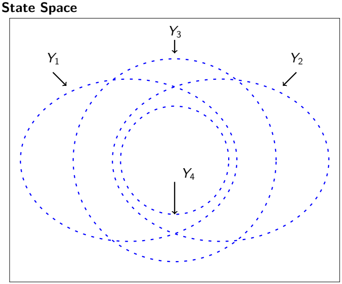
$$\textbf{(1)} \forall s \bullet \; P(s) \;\Rightarrow\; Y(s)$$
$$\textbf{(2)} \forall s \bullet \; \left(\forall s' \bullet \; N(s, s') \Rightarrow \; Y(s')\right) \Rightarrow Y(s)$$



Intuition: States Satisfying **AF** P

According to encoding of **AF** in mu-calculus, [**AF** P] is the **smallest** set Y that satisfies:

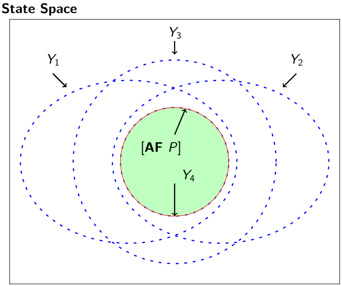
$$(1) \forall s \bullet \; P(s) \;\Rightarrow\; Y(s)$$
$$(2) \forall s \bullet \; \left(\forall s' \bullet \; N(s,s') \Rightarrow \; Y(s')\right) \Rightarrow Y(s)$$



Intuition: States Satisfying **AF** P

According to encoding of **AF** in mu-calculus, [**AF** P] is the **smallest** set Y that satisfies:

$$\textbf{(1)} \forall s \bullet \ P(s) \ \Rightarrow \ Y(s)$$
$$\textbf{(2)} \forall s \bullet \ \big(\forall s' \bullet \ N(s, s') \Rightarrow \ Y(s')\big) \Rightarrow Y(s)$$



$$[\textbf{AF}\ P] = \bigcap_{Y \,\in\, \Theta} Y \qquad \text{where} \ \ \Theta = \{Y\text{s satisfying } \textbf{(1)}, \textbf{(2)}\}$$

Intuition: Model Checking **AF** P

Model checking is about a subset relation, $S_0 \subseteq [\mathbf{AF}\ P]$:

$$S_0 \subseteq \bigcap_{Y \in \Theta} Y$$

Intuition: Model Checking **AF** P

Model checking is about a subset relation, $S_0 \subseteq [\mathbf{AF}\ P]$:

$$S_0 \subseteq \bigcap_{Y \in \Theta} Y \quad \text{iff} \quad \forall Y \in \Theta \bullet S_0 \subseteq Y$$

Intuition: Model Checking **AF** P

Model checking is about a subset relation, $S_0 \subseteq [\mathbf{AF}\ P]$:

$$S_0 \subseteq \bigcap_{Y \in \Theta} Y \quad \text{iff} \quad \forall Y \in \Theta \bullet S_0 \subseteq Y$$

- Higher-order universal quantifier

Intuition: Model Checking **AF** P

Model checking is about a subset relation, $S_0 \subseteq [\mathbf{AF}\ P]$:

$$S_0 \subseteq \bigcap_{Y \in \Theta} Y \quad \text{iff} \quad \forall Y \in \Theta \bullet S_0 \subseteq Y$$

- Higher-order universal quantifier
- First-order logic formula

Intuition: Model Checking **AF** P

Model checking is about a subset relation, $S_0 \subseteq [\mathbf{AF}\ P]$:

$$S_0 \subseteq \bigcap_{Y \in \Theta} Y \quad \text{iff} \quad \forall Y \in \Theta \bullet S_0 \subseteq Y$$

- Higher-order universal quantifier
- First-order logic formula

Definition (FOL Validity)

$\Gamma \models \Phi$ iff every interpretation that satisfies Γ also satisfies Φ.

Intuition: Model Checking **AF** P

Model checking is about a subset relation, $S_0 \subseteq [\mathbf{AF}\ P]$:

$$S_0 \subseteq \bigcap_{Y \in \Theta} Y \quad \text{iff} \quad \forall Y \in \Theta \bullet S_0 \subseteq Y$$

- Higher-order universal quantifier → $\forall Y \in \Theta$
- First-order logic formula → $S_0 \subseteq Y$

Definition (FOL Validity)

$\Gamma \models \Phi$ iff every interpretation that satisfies Γ also satisfies Φ.

$$\begin{array}{c}\text{Description of model}\\ \mathit{symbolic}(K)\end{array} + \begin{array}{l}\forall s \bullet P(s) \Rightarrow Y(s)\\ \forall s \bullet (\forall s' \bullet N(s,s') \Rightarrow Y(s')) \Rightarrow Y(s)\end{array} \models S_0 \subseteq Y$$

Our Result

Reduction Procedure:

INPUT:

symbolic(K) : *symbolic representation of a Kripke structure.*

φ : *a CTL-Live formula.*

OUTPUT:

$symbolic(K) \bigcup \texttt{CTLL2FOL}(\varphi) \models S_0 \subseteq \lceil\varphi\rceil$

Theorem (Reduction of CTL-Live Model Checking to FOL Validity)

$$symbolic(K) \models_c \varphi$$

iff

$$symbolic(K) \bigcup \texttt{CTLL2FOL}(\varphi) \models S_0 \subseteq \lceil\varphi\rceil$$

Our Result

Reduction Procedure:

`INPUT:`

symbolic(K) : *symbolic representation of a Kripke structure.*

φ : *a CTL-Live formula.*

`OUTPUT:`

$symbolic(K) \bigcup \texttt{CTLL2FOL}(\varphi) \models S_0 \subseteq \lceil \varphi \rceil$

Example:

$$
\begin{aligned}
\forall c \bullet\ S_0(c) \Leftrightarrow c = 0 \\
\forall c, c' \bullet\ N(c, c') \Leftrightarrow c' = c + 2 \vee c' = c + 3 \\
\forall c \bullet\ c > 3 \Rightarrow Y(c) \\
\forall c \bullet\ (\forall c' \bullet\ N(c, c') \Rightarrow Y(c')) \Rightarrow Y(c) \quad \models \quad S_0 \subseteq Y
\end{aligned}
$$

Our Result

Reduction Procedure:

INPUT:

$symbolic(K)$: *symbolic representation of a Kripke structure.*

φ : *a CTL-Live formula.*

OUTPUT:

$symbolic(K) \bigcup \mathtt{CTLL2FOL}(\varphi) \models S_0 \subseteq \lceil \varphi \rceil$

Example:

$$
\begin{aligned}
\forall c \bullet \; S_0(c) \Leftrightarrow c = 0 \\
\forall c, c' \bullet \; N(c, c') \Leftrightarrow c' = c + 2 \vee c' = c + 3 \\
\forall c \bullet \; c > 3 \; \Rightarrow \; Y(c) \\
\forall c \bullet \; (\forall c' \bullet \; N(c, c') \Rightarrow \; Y(c')) \Rightarrow Y(c) \quad \models \quad S_0 \subseteq Y
\end{aligned}
$$

Our Result

Reduction Procedure:

`INPUT:`

symbolic(K) : *symbolic representation of a Kripke structure.*

φ : *a CTL-Live formula.*

`OUTPUT:`

symbolic$(K) \bigcup \texttt{CTLL2FOL}(\varphi) \models S_0 \subseteq [\varphi]$

Example:

$$\begin{aligned} \forall c \bullet \; S_0(c) \Leftrightarrow c = 0 \\ \forall c, c' \bullet \; N(c, c') \Leftrightarrow c' = c + 2 \vee c' = c + 3 \\ \forall c \bullet \; c > 3 \; \Rightarrow \; Y(c) \\ \forall c \bullet \; (\forall c' \bullet \; N(c, c') \Rightarrow \; Y(c')) \Rightarrow Y(c) \end{aligned} \quad \models \quad S_0 \subseteq Y$$

Current Progress: Infinite State Model Checking

- Based on this result, we used Z3 and CVC4 to model check CTL-Live properties of 4 infinite systems.
- Case studies were from different domains.
- SMT solvers are efficient in model checking CTL-Live properties.

Vakili and Day, "Verifying CTL-live Properties of Infinite State Models using SMT Solvers," To appear in the proceedings of FSE'14.

Conclusion

- Presented CTL-Live, a fragment of CTL such that its model checking is reducible to FOL validity.
  - No need for abstraction or invariant generation
  - Use state-of-the-art FOL reasoners for model checking
  - Only FOL reasoning is required for verification