

Chapter 5

CTL*

Course “Specification and Verification of Parallel Systems”

Tobias John

Autumn 2023

Plan for today

- repeat CTL*
- idea of CTL* model checking
- complexity of CTL* model checking





Section

Syntax

Autumn 2023

Remember: Syntax of CTL*

Definition (CTL* formulas)

Let AP be a set of atomic propositions and $p \in \mathrm{AP}$. The set of CTL* formulas is defined as the set of all *state formulas* Φ with

$$\begin{array}{lcll} \Phi & ::= & \top \mid \bot \mid p \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \exists\varphi \mid \forall\varphi & \text{state formulas} \\ \varphi & ::= & \Phi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi\ U\ \varphi & \text{path formulas} \end{array}$$



CTL* vs. CTL vs. LTL

- remember: CTL and LTL are both contained in CTL*
  - CTL formula Φ:



CTL* vs. CTL vs. LTL

- remember: CTL and LTL are both contained in CTL*
  - CTL formula Φ: already in CTL*
  - LTL formula φ:



CTL* vs. CTL vs. LTL

- remember: CTL and LTL are both contained in CTL*
  - CTL formula Φ: already in CTL*
  - LTL formula φ: $\forall\varphi$ is equivalent in CTL*
- CTL* contains formulas that are neither in LTL nor CTL

$$\Phi = (\forall\Diamond\Box p) \vee (\forall\Box\exists\Diamond b)$$



CTL* vs. CTL vs. LTL

- remember: CTL and LTL are both contained in CTL*
  - CTL formula Φ: already in CTL*
  - LTL formula φ: $\forall\varphi$ is equivalent in CTL*
- CTL* contains formulas that are neither in LTL nor CTL

$$\Phi = (\forall\Diamond\Box p) \vee (\forall\Box\exists\Diamond b)$$

  (first part not in CTL, second part not in LTL)
- thus: CTL* is proper superset





Section

CTL* model checking

Autumn 2023

Closer look at CTL* formulas

- consider the following CTL* formula:

$$\Phi = (\exists \Box \bigcirc p) \vee \neg(\forall \Diamond \Box r)$$

- observation: state formula that contains path formulas



Closer look at CTL* formulas

- consider the following CTL* formula:

$$\Phi = (\exists \Box \bigcirc p) \vee \neg(\forall \Diamond \Box r)$$

- observation: state formula that contains path formulas
- MC idea: do LTL MC for sub-formulas first, then do CTL MC:
  1. Calculate Sat set for $\exists \Box \bigcirc p$ and $\forall \Diamond \Box r$ using the following equivalences for LTL formulas φ:

$$\mathrm{Sat}_{\mathsf{CTL}^*}(\forall \varphi) = \{q \mid q \models_{\mathsf{LTL}} \varphi\} \quad \text{and} \quad \mathrm{Sat}_{\mathsf{CTL}^*}(\exists \varphi) = \{q \mid q \not\models_{\mathsf{LTL}} \neg\varphi\}$$

  2. do normal CTL MC for Φ using the results from 1.



Closer look at CTL* formulas

- consider the following more nested CTL* formula:

$$\Phi = \neg\forall\bigcirc((\forall\Diamond\Box(p \wedge \forall\bigcirc p)) \vee q)$$

- multiple switches between LTL and CTL MC required
  1. LTL MC for



Closer look at CTL* formulas

- consider the following more nested CTL* formula:

$$\Phi = \neg\forall\bigcirc((\forall\Diamond\Box(p \wedge \forall\bigcirc p)) \vee q)$$

- multiple switches between LTL and CTL MC required
  1. LTL MC for $\varphi_1 = \bigcirc p$
  2. CTL MC for



Closer look at CTL* formulas

- consider the following more nested CTL* formula:

$$\Phi = \neg\forall\bigcirc((\forall\Diamond\Box(p \wedge \forall\bigcirc p)) \vee q)$$

- multiple switches between LTL and CTL MC required
  1. LTL MC for $\varphi_1 = \bigcirc p$
  2. CTL MC for $\Phi_1 = p \wedge \forall\varphi_1$
  3. LTL MC for



Closer look at CTL* formulas

- consider the following more nested CTL* formula:

$$\Phi = \neg\forall\bigcirc((\forall\Diamond\Box(p \wedge \forall\bigcirc p)) \vee q)$$

- multiple switches between LTL and CTL MC required
  1. LTL MC for $\varphi_1 = \bigcirc p$
  2. CTL MC for $\Phi_1 = p \wedge \forall\varphi_1$
  3. LTL MC for $\varphi_2 = \Diamond\Box\Phi_1$
  4. CTL MC for



Closer look at CTL* formulas

- consider the following more nested CTL* formula:

$$\Phi = \neg\forall\bigcirc((\forall\Diamond\Box(p \wedge \forall\bigcirc p)) \vee q)$$

- multiple switches between LTL and CTL MC required
  1. LTL MC for $\varphi_1 = \bigcirc p$
  2. CTL MC for $\Phi_1 = p \wedge \forall\varphi_1$
  3. LTL MC for $\varphi_2 = \Diamond\Box\Phi_1$
  4. CTL MC for $\Phi_2 = (\forall\varphi_2) \vee q$
  5. LTL MC for



Closer look at CTL* formulas

- consider the following more nested CTL* formula:

$$\Phi = \neg\forall\bigcirc((\forall\Diamond\Box(p \wedge \forall\bigcirc p)) \vee q)$$

- multiple switches between LTL and CTL MC required
  1. LTL MC for $\varphi_1 = \bigcirc p$
  2. CTL MC for $\Phi_1 = p \wedge \forall\varphi_1$
  3. LTL MC for $\varphi_2 = \Diamond\Box\Phi_1$
  4. CTL MC for $\Phi_2 = (\forall\varphi_2) \vee q$
  5. LTL MC for $\varphi_3 = \bigcirc\Phi_2$
  6. CTL MC for



Closer look at CTL* formulas

- consider the following more nested CTL* formula:

$$\Phi = \neg\forall\bigcirc((\forall\Diamond\Box(p \wedge \forall\bigcirc p)) \vee q)$$

- multiple switches between LTL and CTL MC required
  1. LTL MC for $\varphi_1 = \bigcirc p$
  2. CTL MC for $\Phi_1 = p \wedge \forall\varphi_1$
  3. LTL MC for $\varphi_2 = \Diamond\Box\Phi_1$
  4. CTL MC for $\Phi_2 = (\forall\varphi_2) \vee q$
  5. LTL MC for $\varphi_3 = \bigcirc\Phi_2$
  6. CTL MC for $\Phi = \neg\forall\varphi_3$
- observation: use results for sub-formulas for larger formulas (analog to CTL MC)
- for CTL* formulas with $\exists$-quantifier, some negated sub-formulas must be considered (see previous frame)



CTL* MC algorithm (idea)

- similar to CTL: build Sat sets for formulas of increasing size
- use LTL MC when encountering a quantified path formula
  - depending on quantifier, treatment might vary slightly
- constant switching between LTL and CTL model checking





Section

Complexity analysis

Autumn 2023

Complexity of CTL* MC

- dominant parts of time for MC algorithm: LTL MC
  (rest is polynomial in size of transition system and formula)
- hence: exponential in length of formula and polynomial in size of transition system
- further analysis: PSPACE-complete (as LTL MC)



Summary

- CTL* entails CTL and LTL
- CTL* MC is decidable and PSPACE-complete
- MC algorithm: similar to CTL MC
- complexity: same as for LTL



References I

Bibliography

