CTL* model checking

CTL* model checking

given: finite TS $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$
CTL* formula Φ

question: does $\mathcal{T} \models \Phi$ hold **?**

CTL* model checking

given: finite TS $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$

CTL* formula Φ

question: does $\mathcal{T} \models \Phi$ hold ?

main procedure as for **CTL**:

```
FOR ALL subformulas Ψ of Φ DO
      compute Sat(Ψ) = {s ∈ S : s ⊨ Ψ}
OD
IF S0 ⊆ Sat(Φ)
      THEN   return "yes"
      ELSE   return "no"
FI
```

Recursive computation of satisfaction sets

Recursive computation of satisfaction sets

$$\left.\begin{array}{lcl} Sat(true) & = & S \\ Sat(a) & = & \{s \in S : a \in L(s)\} \\ Sat(\Phi_1 \wedge \Phi_2) & = & Sat(\Phi_1) \cap Sat(\Phi_2) \\ Sat(\neg\Phi) & = & S \setminus Sat(\Phi) \end{array}\right\} \text{ as for } \textbf{CTL}$$

Recursive computation of satisfaction sets

$$\left.\begin{array}{lcl} Sat(true) & = & S \\ Sat(a) & = & \{s \in S : a \in L(s)\} \\ Sat(\Phi_1 \wedge \Phi_2) & = & Sat(\Phi_1) \cap Sat(\Phi_2) \\ Sat(\neg\Phi) & = & S \setminus Sat(\Phi) \end{array}\right\} \text{ as for } \textbf{CTL}$$

$$\left.\begin{array}{lcl} Sat(\forall\varphi) & = & Sat_{LTL}(\varphi) \\ Sat(\exists\varphi) & = & S \setminus Sat_{LTL}(\neg\varphi) \end{array}\right\} \begin{array}{c}\text{using an } \textbf{LTL} \\ \text{model checker}\end{array}$$

Example: CTL* model checking

$$\Phi \; = \; \exists\Diamond\Box a \;\; \wedge \; \exists\Box\big(\bigcirc b \wedge \Diamond\, \neg\exists(a \,\mathsf{U}\, b)\,\big)$$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\Big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a \,\mathsf{U}\, b)}_{\Phi_2} \Big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\Big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a \, \mathsf{U} \, b)}_{\Phi_2}\Big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\Box\big(\bigcirc b \wedge \Diamond a_2\big)$$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a \,\mathsf{U}\, b)}_{\Phi_2}\big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\underbrace{\Box\big(\bigcirc b \wedge \Diamond a_2\big)}_{\text{LTL formula } \varphi}$$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\Big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a\,\mathsf{U}\,b)}_{\Phi_2}\Big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\underbrace{\Box\big(\bigcirc b \wedge \Diamond a_2\big)}_{\textbf{LTL}\text{ formula } \varphi} = a_1 \wedge \exists\varphi$$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a \,\mathsf{U}\, b)}_{\Phi_2}\big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\underbrace{\Box\big(\bigcirc b \wedge \Diamond a_2\big)}_{\textbf{LTL}\text{ formula } \varphi} = a_1 \wedge \exists\varphi$$

3. use an **LTL** model checker to compute $Sat(\exists\varphi)$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\Big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a \,\mathsf{U}\, b)}_{\Phi_2}\Big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\underbrace{\Box\big(\bigcirc b \wedge \Diamond a_2\big)}_{\textbf{LTL} \text{ formula } \varphi} = a_1 \wedge \exists\varphi$$

3. use an **LTL** model checker to compute $Sat(\exists\varphi)$

more precisely: existential **LTL** model checker

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a \,\mathsf{U}\, b)}_{\Phi_2}\big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\underbrace{\Box\big(\bigcirc b \wedge \Diamond a_2\big)}_{\textbf{LTL}\text{ formula } \varphi} = a_1 \wedge \exists\varphi$$

3. use an **LTL** model checker to compute $Sat(\exists\varphi)$

more precisely: existential **LTL** model checker

1. construct an **NBA** for φ
2. check via nested DFS whether $\mathcal{T} \otimes \mathcal{A} \models \exists\Box\Diamond F$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\Big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a \,\mathsf{U}\, b)}_{\Phi_2}\Big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\underbrace{\Box\big(\bigcirc b \wedge \Diamond a_2\big)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists\varphi$$

3. compute $Sat(\exists\varphi)$ via NBA $\mathcal{A}$ for φ and nested DFS in $\mathcal{T} \otimes \mathcal{A}$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a\,\mathsf{U}\,b)}_{\Phi_2}\big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\underbrace{\Box(\bigcirc b \wedge \Diamond a_2)}_{\text{LTL formula } \varphi} = a_1 \wedge \exists\varphi$$

3. compute $Sat(\exists\varphi)$ via NBA $\mathcal{A}$ for φ and nested DFS in $\mathcal{T} \otimes \mathcal{A}$
4. return $Sat(\Phi) = Sat(a_1 \wedge \exists\varphi)$

Example: CTL* model checking

$$\Phi = \underbrace{\exists\Diamond\Box a}_{\Phi_1} \wedge \exists\Box\Big(\bigcirc b \wedge \Diamond \underbrace{\neg\exists(a \,\mathsf{U}\, b)}_{\Phi_2}\Big)$$

1. calculate recursively the satisfaction sets $Sat(\Phi_i)$
2. replace Φ_i with the atomic proposition a_i, $i = 1, 2$

$$\Phi \rightsquigarrow a_1 \wedge \exists\underbrace{\Box\Big(\bigcirc b \wedge \Diamond a_2\Big)}_{\textbf{LTL} \text{ formula } \varphi} = a_1 \wedge \exists\varphi$$

3. compute $Sat(\exists\varphi)$ via NBA $\mathcal{A}$ for φ and nested DFS in $\mathcal{T} \otimes \mathcal{A}$
4. return $Sat(\Phi) = Sat(a_1 \wedge \exists\varphi) = Sat(\Phi_1) \cap Sat(\exists\varphi)$

Fairness in CTL*

Correct or wrong?

Let $fair = \bigwedge_{1 \leq i \leq k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$s \models_{fair} \exists\Box a \quad \text{iff} \quad s \models \exists(fair \wedge \Box a)$$

Correct or wrong?

Let $fair = \bigwedge_{1 \leq i \leq k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$s \models_{fair} \exists\Box a \quad \text{iff} \quad s \models \exists(fair \wedge \Box a)$$

CTL with fairness

CTL* semantic

Correct or wrong?

Let $\mathit{fair} = \bigwedge_{1 \leq i \leq k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$s \models_{\mathit{fair}} \exists\Box a \quad \text{iff} \quad s \models \exists(\mathit{fair} \wedge \Box a)$$

CTL* path formula

Correct or wrong?

Let $fair = \bigwedge_{1 \le i \le k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$s \models_{fair} \exists\Box a \quad \text{iff} \quad s \models \exists(fair \wedge \Box a)$$

CTL* path formula

correct.

Correct or wrong?

Let $fair = \bigwedge_{1 \leq i \leq k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$s \models_{fair} \exists\Box a \quad \text{iff} \quad s \models \exists(fair \wedge \Box a)$$

correct.

$$s \models_{fair} \forall\Box a \quad \text{iff} \quad s \models \forall(fair \wedge \Box a)$$

Correct or wrong?

Let $fair = \bigwedge_{1 \leq i \leq k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$\boxed{s \models_{fair} \exists\Box a \quad \text{iff} \quad s \models \exists(fair \wedge \Box a)}$$

correct.

$$\boxed{s \models_{fair} \forall\Box a \quad \text{iff} \quad s \models \forall(fair \wedge \Box a)}$$

wrong.

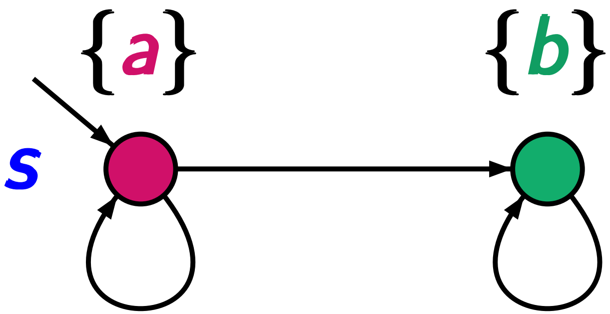
Let $\mathit{fair} = \bigwedge_{1 \le i \le k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$s \models_{\mathit{fair}} \exists\Box a \quad \text{iff} \quad s \models \exists(\mathit{fair} \wedge \Box a)$$

correct.

$$s \models_{\mathit{fair}} \forall\Box a \quad \text{iff} \quad s \models \forall(\mathit{fair} \wedge \Box a)$$

wrong.



$\mathit{fair} = \Box\Diamond\neg b$

Correct or wrong?

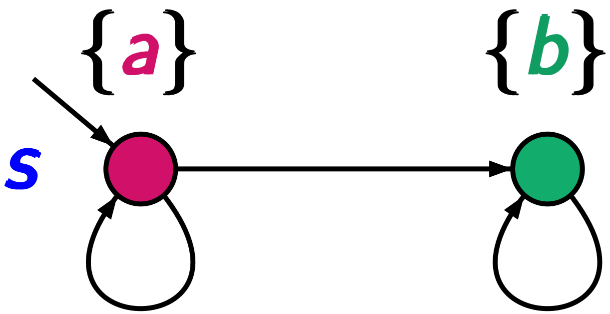
Let $fair = \bigwedge_{1 \leq i \leq k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$s \models_{fair} \exists\Box a \quad \text{iff} \quad s \models \exists(fair \wedge \Box a)$$

correct.

$$s \models_{fair} \forall\Box a \quad \text{iff} \quad s \models \forall(fair \wedge \Box a)$$

wrong.



$fair = \Box\Diamond\neg b$

$s \models_{fair} \forall\Box a$

Correct or wrong?

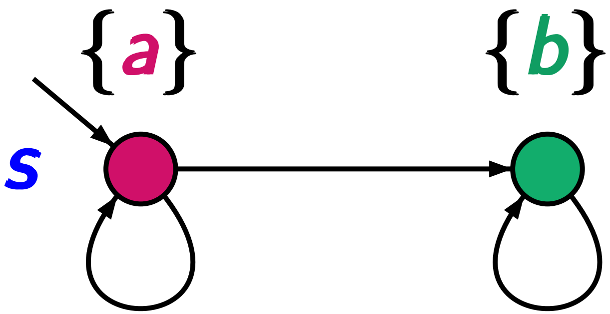
Let $fair = \bigwedge_{1 \leq i \leq k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$s \models_{fair} \exists\Box a \quad \text{iff} \quad s \models \exists(fair \wedge \Box a)$$

correct.

$$s \models_{fair} \forall\Box a \quad \text{iff} \quad s \models \forall(fair \wedge \Box a)$$

wrong.



$fair = \Box\Diamond\neg b$

$s \models_{fair} \forall\Box a$

$s \not\models \forall(fair \wedge \Box a)$

Correct or wrong?

Let $fair = \bigwedge_{1 \leq i \leq k} \Box\Diamond c_i$ be an unconditional **LTL** fairness assumption

$$\boxed{\quad s \models_{fair} \exists\Box a \quad \text{iff} \quad s \models \exists(fair \wedge \Box a) \quad}$$

correct.

$$\boxed{\quad s \models_{fair} \forall\Box a \quad \text{iff} \quad s \models \forall(fair \wedge \Box a) \quad}$$

wrong. But we have:

$$s \models_{fair} \forall\Box a \quad \text{iff} \quad s \models \forall(fair \rightarrow \Box a)$$