

Algoritmos de *model checking*

A semântica dada para o CTL e o LTL, permite decidir se

$$\mathcal{M}, s_0 \models \phi$$

Mas é conveniente ter algoritmos de decisão.

Normalmente, dado \mathcal{M} e ϕ os algoritmos determinam *o conjunto de estados que satisfazem ϕ*

Depois basta ver se s_0 está nesse conjunto.

Model checking para o CTL – algoritmo de etiquetagem

Vamos apenas considerar um conjunto completo de conectivas, $\{\perp, \neg, \wedge, \text{AF}, \text{EU}, \text{EX}\}$.

[Clarke and Emerson 1981] [Turing award, 2007](#)

Dados: um modelo $\mathcal{M} = (S, \rightarrow, L)$ e uma fórmula CTL ϕ

Saída: conjunto de estados de \mathcal{M} que satisfazem ϕ

O algoritmo etiqueta cada estado de \mathcal{M} com as subfórmulas de ϕ que são satisfeitas nesse estado, começando das mais pequenas até ϕ .

O algoritmo procede por indução na estrutura de ϕ . Supondo que ψ é uma subfórmula de ϕ cujas subfórmulas imediatas já etiquetam todos os estados em que são satisfeitas, determina-se por análise de casos os estados que etiquetam ψ .

Model checking para o CTL – algoritmo de etiquetagem

Se ψ é

\perp Não etiqueta nenhum estado

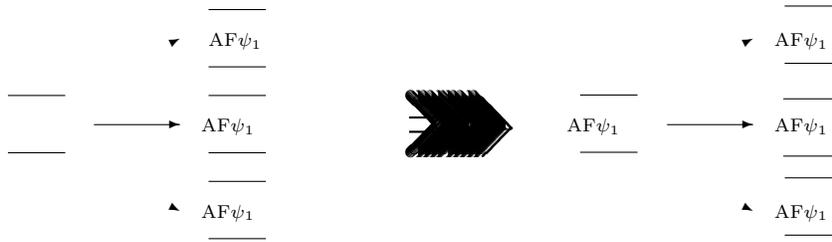
p etiquetar os estados s tal que $p \in L(s)$

$\psi_1 \wedge \psi_2$ etiquetar os estados s que estejam etiquetados com ψ_1 e ψ_2

$\neg\psi_1$ etiquetar os estados s que não estejam etiquetados com ψ_1

Model checking para o CTL – algoritmo de etiquetagem

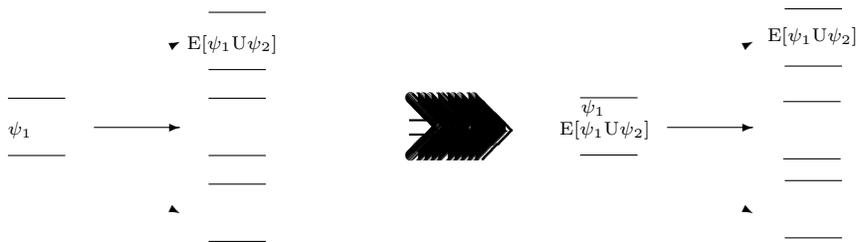
- AF ψ_1**
- Se existe s que está etiquetado com ψ_1 então etiquetá-lo com AF ψ_1 .
 - Repetir: Se todos os sucessores de um estado estiverem etiquetados com AF ψ_1 , etiquetar esse estado. Até não haver alteração.



$$AF\phi \equiv \phi \vee AXAF\phi$$

Model checking para o CTL – algoritmo de etiquetagem

- $E(\psi_1 U \psi_2)$
- Se existe s que está etiquetado com ψ_2 então etiquetá-lo com $E(\psi_1 U \psi_2)$.
 - Repetir: Etiquetar um estado com $E(\psi_1 U \psi_2)$ se está etiquetado com ψ_1 e se pelo menos um dos seus sucessores está etiquetado com $E(\psi_1 U \psi_2)$. Até não haver alteração.



$$E[\psi_1 U \psi_2] \equiv \psi_2 \vee (\psi_1 \wedge EXE[\psi_1 U \psi_2])$$

Model checking para o CTL – algoritmo de etiquetagem

$EX\psi_1$ etiquetar com $EX\psi_1$ um estado se pelo menos um dos seus sucessores estiver etiquetado com ψ_1 .

Model checking para o CTL – algoritmo de etiquetagem

Exercício 8.1. Considere o modelo $\mathcal{M} = (\{q_0, q_1, q_2, q_3\}, \{q_0 \rightarrow q_1, q_0 \rightarrow q_3, q_1 \rightarrow q_1, q_1 \rightarrow q_2, q_2 \rightarrow q_0, q_2 \rightarrow q_3, q_3 \rightarrow q_0\}, L(q_0) = \{p, q\}, L(q_1) = \{r\}, L(q_2) = \{p, t\}, L(q_3) = \{q, r\})$.

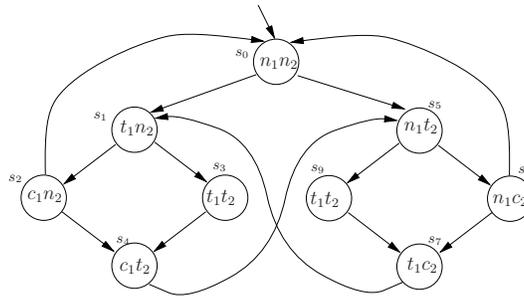
Determine os estados s tal que $\mathcal{M}, q_0 \models \phi$ para

a) $\phi = AFq$,

- b) $\phi = EXEXr$ e
 c) $\phi = AG(EF(p \vee r))$.
 \diamond

Model checking para o CTL – algoritmo de etiquetagem

Exemplo2: $\mathcal{M}_2, s_0 \models E(\neg c_2 U c_1)$?



Model checking para o CTL – algoritmo de etiquetagem

O algoritmo tem complexidade $O(f \cdot V \cdot (V + E))$, onde f é o número de conectivas da fórmula, V o número de estados e E ou número de transições.

Pode ser mais eficiente se se considerarem mais conectivas, p.e. EG.

- EG ψ**
- Etiquetar todos os estados com EG ψ .
 - Se um estado s não tiver a etiqueta ψ , apagar a etiqueta EG ψ .
 - Repetir: apagar a etiqueta EG ψ se nenhum dos sucessores do estado tiver a etiqueta EG ψ . Até não haver mudança.

Equivale ter $\neg AF\neg\psi\dots$

Uma variante mais eficiente

Se considerarmos EG em vez de AF no conjunto completo de conectivas podemos obter um algoritmo linear, $O(f \cdot (V + E))$.

Para EX e EU o algoritmo é o mesmo, mas considerando uma pesquisa em largura.

- EG ψ**
- Restringir o digrafo a estados que satisfazem ψ

- Determinar as componentes fortemente conexas (maximais) (SCC).
- Utilizar pesquisa em largura (para trás) para encontrar um estado que possa estar ligado a uma SCC.

