$M \models \varphi$? (CTL)

AG(raro)
$A \neg (x \, b \rightarrow EG ($

3.2 Verifica formale di proprietà mediante model checking

Angelo Gargantini

Testing e Verifica del Software AA 2526

March 3, 2026

Outline

Outline

- model checking algorithm for CTL logics

TESTO DI RIFERIMENTO: M.R.A. Ruth, M.D. Ryan Logic in Computer Science Modelling and Reasoning about systems - Capitolo 3 - allegato a questi appunti

Model checking algorithm

- We want to solve this problem $M, s \overset{???}{\models} \phi$
- There are everal approaches to formal verification
  - (automated) theorem prover
  - **model checking** (several types) ... Spin, PRISM, JavaPathFinder, ...

Model checking algorithm - CTL

- We want to solve this problem $M, s \overset{???}{\models} \phi$
  - model checking
  - by a labelling algorithm:
    - INPUT: a model M = (S, $\rightarrow$, L) and a CTL formula φ .
    - OUTPUT: the set of states of M which satisfy φ.
  1. First, change φ in terms of the connectives AF, EU, EX, $\wedge$, $\neg$ and $\bot$ using the equivalences given earlier.
  2. Next, label the states of M in which φ holds

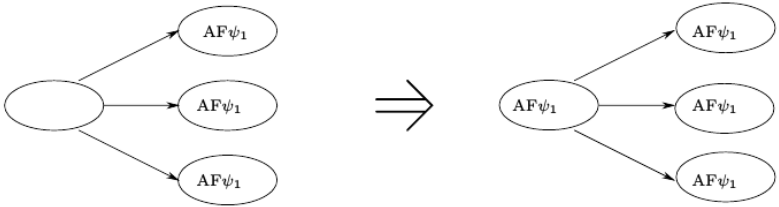
Labeling algorithm

Case analysis over ψ. If ψ is

- $\bot$: then no states are labelled with $\bot$
- p: then label every s such that p $\in L(s)$
- $\psi 1 \wedge \psi 2$:
  - do labelling with $\psi 1$ and with $\psi 2$
  - label s with $\psi 1 \wedge \psi 2$ if s is already labelled both with $\psi 1$ and with $\psi 2$
- $\neg\psi$:
  - do labelling with ψ
  - label s with $\neg\psi 1$ if s is not labelled with ψ.

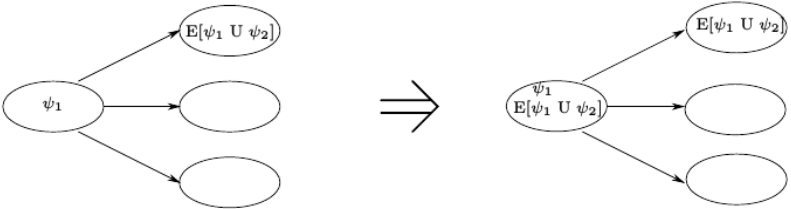
Labeling algorithm - AF

- AF ψ :
  - do labeling with ψ
    - If any state s is labelled with ψ , label it with AF ψ.
    - Repeat: label any state with AF ψ if all successor states are labelled with AF ψ , until there is no change. See picture



Labeling algorithm - EU

- E[ψ1 U ψ2]
  - do labeling for ψ1 and ψ2
    - If any state s is labelled with ψ2, label it with E[ψ1 U ψ2]
    - Repeat: label any state with E[ψ1 U ψ2] if it is labelled with ψ1 and at least one of its successors is labelled with E[ψ1 U ψ2], until there is no change.



Labeling algorithm - EX

- EX ψ :
  - do labeling for ψ
    - label any state with EXψ if one of its successors is labelled with ψ

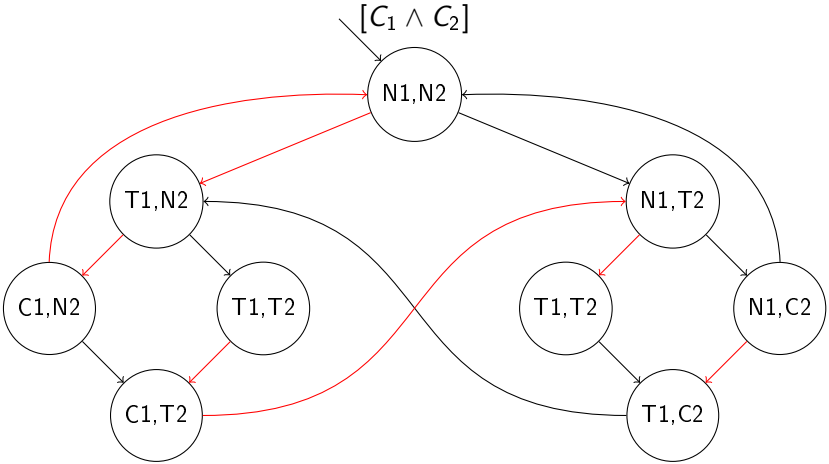
Complessità

- The complexity of this algorithm is $O(f \cdot V \cdot (V + E))$, where f is the number of connectives in the formula, V is the number of states and E is the number of transitions; the algorithm is linear in the size of the formula and quadratic in the size of the model.
  - Some improvements
    - Handling EG directly
  - LTL is treated differently (skip)

Altri casi

Come faccio in tutti gli altri casi????? TODO

Esempio



State Explosion problem

The 'state explosion' problem Although the labelling algorithm (with the clever way of handling EG) is linear in the size of the model, unfortunately the size of the model is itself more often than not exponential in the number of variables and the number of components of the system which execute in parallel. This means that, for example, adding a boolean variable to your program will double the complexity of verifying a property of it. The tendency of state spaces to become very large is known as the state explosion problem. A lot of research has gone into finding ways of overcoming it, including the use of:

- efficient data structure BDDs
  - ....