System Verification with Model Checking

Περιεχόμενα

- Εισαγωγή – Πρόβλημα
- Μοντελοποίηση
- Χρονικές Λογικές
- Αλγόριθμος Model Checking

Εισαγωγή – Πρόβλημα

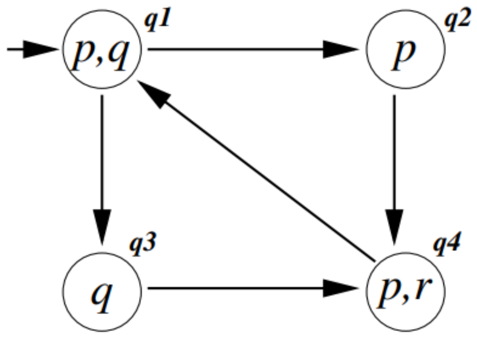
Formal methods

- **Abstract Interpretation:**
  Σε αυτή την μέθοδο, στόχος είναι ο υπολογισμός **invariants**, συνθηκών που θα ι σχύουν κάθε φορά που θα λειτουργεί το σύστημα, ανεξάρτητα της εισόδου του. Για παράδειγμα, θα μπορούσε σε ένα πρόγραμμα, η ανάλυση με αυτή την μέθοδ ο να καταλήγει στο συμπέρασμα ότι η τιμή μιας μεταβλητής είναι πάντα 5.
- **Model Checking:**Σε αυτή τη μέθοδο, ο χρήστης παρέχει ένα μοντέλο (ή ένα σύστημα) και τον προσδιορισμό λειτουργίας του, καθώς και τα δεδομένων εισόδου και η μ έθοδος αποφαίνεται αν μπορεί να υπάρξει κάποιο πιθανό λάθος ή γίνεται επιτυ χημένος έλεγχος λειτουργίας.
- **Equivalence checking:**
  Σε αυτή την μέθοδο, δύο μοντέλα συγκρίνονται μεταξύ τους για να βρεθεί πόσο όμοια συμπεριφέρονται κάτω από διάφορες συνθήκες.
- **Verification by Deduction:**
  Σε αυτή την μέθοδο, η ιδιότητα του συστήματος είτε αποδεικνύεται με κάποιας μορφής απόδειξη ή αποδεικνύεται ότι η ιδιότητα δεν ισχύει. Σε αυτή την μέθοδο ο χρήστης πρέπει να παρέχει **invariants** σε κάποια σημεία της λειτουργίας του συστήματος. Καθώς η απόδειξη μιας ιδιότητας μπορεί να πάρει πολύ καιρό, συν ήθως χρησιμοποιείται μόνο στις πιο **critical** ιδιότητες των συστημάτων και μπο ρεί να χρησιμοποιηθεί και για συστήματα που έχουν άπειρες καταστάσεις.

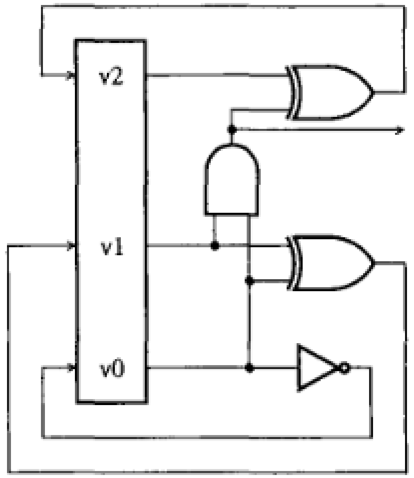
Kripke structure

A Kripke structure K is a quadruple $\mathrm{K} = (\mathrm{V}, \mathrm{E}, \mathrm{L}, \mathrm{I})$ with

- V a set of vertices (interpreted as system states),
- $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$ a set of edges (interpreted as possible transitions),
- $\mathrm{L} \in \mathrm{V} \to \mathcal{P}(\mathrm{AP})$ labels the vertices with atomic propositions that apply in the individual vertices,
- $\mathrm{I} \subseteq \mathrm{V}$ is a set of initial states.



Παράδειγμα



$v_0' = \neg v_0$

$v_1' = v_0 \oplus v_1$

$v_2' = (v_0 \wedge v_1) \oplus v_2$

$\mathcal{R}_0(V, V') \equiv (v_0' \Leftrightarrow \neg v_0)$

$\mathcal{R}_1(V, V') \equiv (v_1' \Leftrightarrow v_0 \oplus v_1)$

$\mathcal{R}_2(V, V') \equiv (v_2' \Leftrightarrow (v_0 \wedge v_1) \oplus v_2)$

$\mathcal{R}(V, V') \equiv \mathcal{R}_0(V, V') \wedge \mathcal{R}_1(V, V') \wedge \mathcal{R}_2(V, V')$

Μονοπάτια

A path π in a Kripke structure $K = (V, E, L, I)$ is an edge-consistent infinite sequence of vertices:

- $\pi \in V^\omega$,
- $(\pi_i, \pi_{i+1}) \in E$ for each $i \in \mathbb{N}$.

Note that a path need not start in an initial state!

The labelling L assigns to each path π a propositional trace

$$\mathrm{tr}_\pi = L(\pi) \stackrel{\text{def}}{=} \langle L(\pi_0), L(\pi_1), L(\pi_2), \ldots \rangle$$

that *path formulae* $(\mathrm{X}\phi, \mathrm{F}\phi, \mathrm{G}\phi, \phi\mathrm{U}\psi)$ can be interpreted on.

CTL

- We start from a countable set AP of atomic propositions. The CTL formulae are then defined inductively:
- Any proposition $p \in AP$ is a CTL formula.
- The symbols $\perp$ and $\top$ are CTL formulae.
- If φ and ψ are CTL formulae, so are $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ **EX** φ, **AX** φ **EF** φ, **AF** φ **EG** φ, **AG** φ φ **EU** ψ, φ **AU** ψ

Σημασιολογία

E and A are path quantifiers:

A: for all paths in the computation tree . . .

E: for some path in the computation tree . . .

X, F, G und U are temporal operators which refer to the path under investigation, as known from LTL:

X φ (Next): evaluate φ in the next state on the path

F φ (Finally): φ holds for some state on the path

G φ (Globally): φ holds for all states on the path

φ U ψ (Until): φ holds on the path at least until ψ holds

Σημασιολογία

Let $K = (V, E, L, I)$ be a Kripke structure and $v \in V$ a vertex of K.

- $v, K \models \top$
- $v, K \not\models \bot$
- $v, K \models p$ for $p \in AP$ iff $p \in L(v)$
- $v, K \models \neg\phi$ iff $v, K \not\models \phi$,
- $v, K \models \phi \land \psi$ iff $v, K \models \phi$ and $v, K \models \psi$,
- $v, K \models \phi \lor \psi$ iff $v, K \models \phi$ or $v, K \models \psi$,
- $v, K \models \phi \Rightarrow \psi$ iff $v, K \not\models \phi$ or $v, K \models \psi$.

Σημασιολογία

- $v, K \models \mathrm{EX}\,\phi$ iff there is *a path* π in K s.t. $v = \pi_1$ and $\pi_2, K \models \phi$,
- $v, K \models \mathrm{AX}\,\phi$ iff *all paths* π in K with $v = \pi_1$ satisfy $\pi_2, K \models \phi$,
- $v, K \models \mathrm{EF}\,\phi$ iff there is *a path* π in K s.t. $v = \pi_1$ and $\pi_i, K \models \phi$ for some i,
- $v, K \models \mathrm{AF}\,\phi$ iff *all paths* π in K with $v = \pi_1$ satisfy $\pi_i, K \models \phi$ for some i (that may depend on the path),
- $v, K \models \mathrm{EG}\,\phi$ iff there is *a path* π in K s.t. $v = \pi_1$ and $\pi_i, K \models \phi$ for all i,
- $v, K \models \mathrm{AG}\,\phi$ iff *all paths* π in K with $v = \pi_1$ satisfy $\pi_i, K \models \phi$ for all i,
- $v, K \models \phi\,\mathrm{EU}\,\psi$, iff there is *a path* π in K s.t. $v = \pi_1$ and some $k \in \mathbb{N}$ s.t. $\pi_i, K \models \phi$ for *each* $i < k$ and $\pi_k, K \models \psi$,
- $v, K \models \phi\,\mathrm{AU}\,\psi$, iff *all paths* π in K with $v = \pi_1$ have some $k \in \mathbb{N}$ s.t. $\pi_i, K \models \phi$ for *each* $i < k$ and $\pi_k, K \models \psi$.

A Kripke structure $K = (V, E, L, I)$ satisfies ϕ iff all its initial states satisfy ϕ,

i.e. iff $is, K \models \phi$ for all $is \in I$.

Ταυτολογίες

The tautologies

$$\begin{aligned}
\phi \lor \psi &\Leftrightarrow \neg(\neg\phi \land \neg\psi) \\
\mathrm{AX}\,\phi &\Leftrightarrow \neg\mathrm{EX}\,\neg\phi \\
\mathrm{AG}\,\phi &\Leftrightarrow \neg\mathrm{EF}\,\neg\phi \\
\mathrm{EF}\,\phi &\Leftrightarrow \top\,\mathrm{EU}\,\phi \\
\mathrm{EG}\,\phi &\Leftrightarrow \neg\mathrm{AF}\,\neg\phi \\
\phi\,\mathrm{AU}\,\psi &\Leftrightarrow \neg((\neg\psi)\,\mathrm{EU}\,\neg(\phi \lor \psi)) \land \mathrm{AF}\,\psi
\end{aligned}$$

indicate that we can rewrite each formula to one only containing atomic propositions, $\neg$, $\land$, EX, EU, AF.

Αλγόριθμος

We will extend the idea of verification/falsification by exhaustive state-space exploration to full CTL.

- Main technique will again be breadth-first search, i.e. graph coloring.
- Need to extend this to other modalities than `AG` ..
- Need to deal with nested modalities.

Γενική ιδέα

Given: a Kripke structure $K = (V, E, L, I)$ and a CTL formula ϕ

Core algorithm: find the set $V_\phi \subseteq V$ of vertices in K satisfying ϕ by

1. for each atomic subformula p of ϕ, mark the set $V_p \subseteq V$ of vertices in K satisfying ϕ
2. for increasingly larger subformulae ψ of ϕ, synthesize the marking $V_\psi \subseteq V$ of nodes satisfying ψ from the markings for ψ's immediate subformulae

until all subformulae of ϕ have been processed (including ϕ itself)

Result: report "$K \models \phi$" iff $V_\phi \supseteq I$

Ατομική πρόταση

Given: A finite Kripke structure with vertices V and edges E and a labelling function L assigning atomic propositions to vertices.

Furthermore an atomic proposition p to be checked.

Algorithm: Mark all vertices that have p as a label.

Complexity: $O(|V|)$

Άρνηση

Given: A set V_ϕ of vertices satisfying formula ϕ.

Algorithm: Mark all vertices not belonging to V_ϕ.

Complexity: $O(|V|)$

Τομή
$\phi \wedge \psi$

Given: Sets V_ϕ and V_ψ of vertices satisfying formulae ϕ or ψ, resp.

Algorithm: Mark all vertices belonging to $V_\phi \cap V_\psi$.

Complexity: $O(|V|)$

EX ϕ

Given: Set V_ϕ of vertices satisfying formulae ϕ.

Algorithm: Mark all vertices that have a successor state in V_ϕ.

Complexity: $O(|V| + |E|)$

ϕEU ψ

Given: Sets V_ϕ and V_ψ of vertices satisfying formulae ϕ or ψ, resp.

Algorithm: Incremental marking by

1. Mark all vertices belonging to V_ψ.
2. Repeat
   if there is a state in V_ϕ that has some successor state marked then mark it also
   until no new state is found.

Termination: Guaranteed due to finiteness of $V_\phi \subset V$.

Complexity: $O(|V| + |E|)$ if breadth-first search is used.

AF ϕ

Given: Set V_ϕ of vertices satisfying formula ϕ.

Algorithm: Incremental marking by

1. Mark all vertices belonging to V_ϕ.
2. Repeat

   if there is a state in V that has *all* successor states marked
   then mark it also

   until no new state is found.

Termination: Guaranteed due to finiteness of V.

Complexity: $O(|V| \cdot (|V| + |E|))$.

EG ϕ

Given: Set V_ϕ of vertices satisfying formula ϕ.

Algorithm: Incremental marking by

1. Strip Kripke structure to V_ϕ-states:
   $(V, E) \rightsquigarrow (V_\phi, E \cap (V_\phi \times V_\phi))$.

$\rightsquigarrow$ Complexity: $O(|V| + |E|)$

2. Mark all states belonging to loops in the reduced graph.

$\rightsquigarrow$ Complexity: $O(|V_\phi| + |E_\phi|)$ by identifying *strongly connected components*.

3. Repeat
   if there is a state in V_ϕ that has *some* successor states marked then mark it also
   until no new state is found.

$\rightsquigarrow$ Complexity: $O(|V_\phi| + |E_\phi|)$

Complexity: $O(|V| + |E|)$.

Τελικό αποτέλεσμα

Theorem: It is decidable whether a finite Kripke structure (V, E, L, I) satisfies a CTL formula ϕ.

The complexity of the decision procedure is $O(|\phi| \cdot (|V| + |E|))$, i.e.

- linear in the size of the formula, given a fixed Kripke structure,
- linear in the size of the Kripke structure, given a fixed formula.