

Lecture 18: Nov 15
Temporal Liveness Requirements

- We studied STL, a temporal logic interpreted over states of observation structures
- Now we add
 - fairness constraints to the structure
 - additional connectives to STL
- Temporal logic based approach to verifying liveness requirements of fair modules

Fair Structures

- Fair Graph = Transition Graph + Fairness Assumption
- Fair Structure
 - Observation structure + Fairness
 - Fair Graph + Observations
- Components of fair structure \mathcal{K}
 - States Σ
 - Transitions \rightarrow
 - Initial region σ^I
 - Observations A
 - Labeling of states with observations $\langle\langle s \rangle\rangle$
 - Fairness assumption: set of action-pairs $\{(\alpha_1, \beta_1), \dots, (\alpha_k, \beta_k)\}$
- Fair traces: consider F -fair trajectories and map states to observations
- The fair language $\mathcal{L}_{\mathcal{K}}$ of \mathcal{K} the set of initialized fair traces of \mathcal{K} .

- Every fair module \mathcal{P} has an associated fair structure $\mathcal{K}_{\mathcal{P}}$
- When there are no fairness constraints then the ω -language \mathcal{L}_K is a safety language, and equals $safe(L_K)$
 - defined by finite traces
 - proof depends on the fact that finitely many initial states and finitely many successors per observation

Temporal Logic CTL

- Interpreted over states of fair structures
- Syntax extends that of STL with an additional connective $\exists\Box$ (possibly-always)

$$\phi ::= p \mid \phi \vee \phi \mid \neg\phi \mid \exists\bigcirc\phi \mid \exists\Box\phi \mid \phi\exists\mathcal{U}\phi$$

- Semantics of $\exists\Box$: State s of \mathcal{K} satisfies the formula $\exists\Box p$ if there exists a source- s fair trajectory \underline{s} of \mathcal{K} such that s_i satisfies p for all i
- Semantics of other operators as before
- Additional defined connectives:

Inevitably	$\forall\Diamond\phi$	for $\neg\exists\Box\neg\phi$;
Inevitably-until	$\psi\forall\mathcal{U}\phi$	for $\psi\forall\mathcal{W}\phi \wedge \forall\Diamond\phi$;
Possibly-waiting-for	$\psi\exists\mathcal{W}\phi$	for $\psi\exists\mathcal{U}\phi \vee \exists\Box\psi$.

CTL continued

- $\forall\Diamond p$ holds if every fair trajectory contains a p -state
 - p is inevitable
 - only way to avoid p is to violate the fairness assumption
- Concepts such as characteristic regions, model checking defined as before (eg. $\mathcal{K} \models \phi$ if every initial state of \mathcal{K} satisfies ϕ)
- Fair emptiness is a special case: fair language of \mathcal{K} is nonempty iff $\mathcal{K} \models \exists\Box true$
- Recurrence verification is a special case: p is a recurrent of a fair structure \mathcal{K} if $\mathcal{K} \models \forall\Box\forall\Diamond p$
- Response verification is a special case: q is a response to the request p in \mathcal{P} iff $\mathcal{P} \models \forall\Box(p \rightarrow \forall\Diamond q)$.

Mutual Exclusion: CTL specs

- Deadlock freedom:

$$\forall \square((pc_1 = req \vee pc_2 = req) \rightarrow \forall \diamond(pc_1 = in \vee pc_2 = in))$$

- Starvation freedom:

$$\forall \square((pc_1 = req \rightarrow \forall \diamond pc_1 = in) \wedge (pc_2 = req \rightarrow \forall \diamond pc_2 = in))$$

- Inevitably-until formula:

$$(pc_1 = req \rightarrow (pc_1 = req) \forall \mathcal{U}(pc_1 = in))$$

CTL Model Checking

- Input: fair structure \mathcal{K} and a CTL formula ϕ
- Output: characteristic region $\llbracket \phi \rrbracket$ (all states of \mathcal{K} that satisfy ϕ)
- Approach as in STL: compute characteristic regions of subformulas starting with the simplest ones
- Given ϕ , the function *OrderedSub* returns a list of subformulas of ϕ in nondecreasing order of size
- The algorithm computes, for each state s , the set $\lambda(s)$ of subformulas that are satisfied by the state s .
- The formulas in *OrderedSub*(ϕ) are processed one by one, and the sets $\lambda(s)$ are updated
- Cases corresponding to propositions, logical connectives, $\exists\bigcirc$, and $\exists\mathcal{U}$ as in STL
- New case: how to compute $\llbracket \exists\Box\psi \rrbracket$ from $\llbracket \psi \rrbracket$?

Computing $\llbracket \exists \square \psi \rrbracket$

- Given (G, F) , consider the fair subgraph (H, F) obtained by deleting states that do not satisfy ψ
- Compute fair components of (H, F)
- A state s satisfies $\exists \square \psi$ iff from s some fair component is reachable
- Compute fair components and apply pre^* (DFS along reversed edges)
- Complexity: complexity of computing fair components

Enumerative Complexity

- Number of subformulas: $|\phi|$
- Handling each subformula, except $\exists\Box$ -formulas:
linear in K
- Handling $\exists\Box$ -formula: $O((m+n)\ell^2)$ if there are ℓ constraints
- Overall complexity: $O((n+m) \cdot \ell^2 \cdot |\phi|)$
- When F has only weak-fairness constraints,
computing fair components is easier:
 $O((n+m) \cdot \ell \cdot |\phi|)$
- If no fairness constraints: $O((n+m) \cdot |\phi|)$
- Added complexity of CTL over STL is due to
fairness constraints, and not due to $\exists\Box$

Symbolic Model Checking

- Goal: develop a symbolic algorithm for CTL model checking
- Problematic case: computing $\llbracket \exists \square p \rrbracket$ (set of states s such that there is a source- s fair trajectory containing only p -states)
- How to handle fairness constraints symbolically?
- Tool: μ -calculus

μ -calculus: introduction

- Notation to formulate symbolic fixpoint computation algorithms
- Much more expressive than CTL
- Interpreting μ -calculus formulas is difficult
 - Not really meant as a specification language for the user to write requirements, but as an intermediate language for the tool
- Interpreted over observation structures rather than fair structures
 - Fairness is specifiable within the calculus

$\exists\Diamond$ as a fixpoint

- The characteristic region $\llbracket\exists\Diamond p\rrbracket_K$ consists of all states from which a p -state is reachable.
- It is the smallest region σ that contains $\llbracket p \rrbracket$ as well as $pre(\sigma)$.
- Consider the the function \mathcal{F} that maps regions of K to regions of K :

$$\mathcal{F}(\sigma) = \llbracket p \rrbracket \cup pre(\sigma).$$

- $\llbracket\exists\Diamond p\rrbracket$ is the least fixpoint of \mathcal{F} .
- The corresponding μ -calculus formula is

$$\mu X. (p \vee \exists\bigcirc X)$$

- The variable X ranges over regions, it is bound by the least-fixpoint quantifier μX

Syntax of μ -calculus

- Dual of μ is denoted ν : greatest fixpoint operator
- $\llbracket \forall \square p \rrbracket$ is the maximal region σ whose states satisfy p and have all successors within σ :

$$\nu X. (p \wedge \forall \bigcirc X)$$

- Syntax: negation applied only to atomic formulas for monotonicity
- For atomic formula p and region variable X :

$$p \mid \neg p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \exists \bigcirc \phi \mid \forall \bigcirc \phi \mid \mu X. \phi \mid \nu X. \phi \mid X$$

- Each region variable can be free or bound
- Closed formula: contains only bound variables

Semantics

- A region environment \mathcal{E} assigns to each region variable $X \in \mathbf{Var}$ a region $\sigma \subseteq \Sigma$.
- The meaning $\llbracket \phi \rrbracket$ of a formula is defined wrt to an observation structure K and a region environment \mathcal{E}
- Region environment is used to evaluate the free variables
- $s \models_{K, \mathcal{E}} p$ iff $\langle\langle s \rangle\rangle \models p$
- $s \models_{K, \mathcal{E}} \neg p$ iff $\langle\langle s \rangle\rangle \models \neg p$
- $s \models_{K, \mathcal{E}} \phi_1 \wedge \phi_2$ iff $s \models_{K, \mathcal{E}} \phi_1$ and $s \models_{K, \mathcal{E}} \phi_2$
- $s \models_{K, \mathcal{E}} \phi_1 \vee \phi_2$ iff $s \models_{K, \mathcal{E}} \phi_1$ or $s \models_{K, \mathcal{E}} \phi_2$
- $s \models_{K, \mathcal{E}} \exists \bigcirc \phi$ iff for some state $t \in \text{post}_K(s)$, $t \models_{K, \mathcal{E}} \phi$

- $s \models_{K,\mathcal{E}} \forall \bigcirc \phi$ iff for all states $t \in \text{post}_K(s)$,
 $t \models_{K,\mathcal{E}} \phi$
- $\llbracket X \rrbracket_{K,\mathcal{E}} = \mathcal{E}(X)$
- To define the meaning of $\llbracket \mu X. \phi \rrbracket_{K,\mathcal{E}}$, define the function \mathcal{F} from regions to regions:
 - $\mathcal{F}(\sigma) = \llbracket \phi \rrbracket_{K,\mathcal{E}[X:=\sigma]}$
 - \mathcal{F} evaluates ϕ after setting X to σ
- $\llbracket \mu X. \phi \rrbracket_{K,\mathcal{E}}$ is the least fixpoint of \mathcal{F}
- $\llbracket \nu X. \phi \rrbracket_{K,\mathcal{E}}$ is the greatest fixpoint of \mathcal{F}
- Lemma: \mathcal{F} is monotonic functions on regions
- By Knaster-Tarski fixpoint theorem, the least and greatest fixpoints are defined.
- Least fixpoint of \mathcal{F} is the intersection of all its fixpoints
- On finitely branching structures, pre is continuous, and hence, so is \mathcal{F}

- By Kleene fixpoint theorem, $\mu X. \phi$ can be evaluated by approximations, starting with false:

$$\mu \mathcal{F} = \cup_i \mathcal{F}^i(\emptyset)$$

- Example: evaluating $\llbracket \mu X. p \vee \exists \circ X \rrbracket$:

$$\llbracket false \rrbracket \cup \llbracket p \rrbracket \cup \llbracket \exists \circ p \rrbracket \cup \llbracket \exists \circ \exists \circ p \rrbracket \cup \dots$$

- Similarly, greatest fixpoint $\nu X. \mathcal{F} = \cap_i \mathcal{F}^i(true)$

- Example: evaluating $\llbracket \mu X. p \vee \exists \circ X \rrbracket$

$$\llbracket false \rrbracket \cup \llbracket p \rrbracket \cup \llbracket \exists \circ p \rrbracket \cup \llbracket \exists \circ \exists \circ p \rrbracket \cup \dots$$

Alternation Depth

- Number of switches from μ to ν
- Better indicator of complexity
- Examples

$$\mu X. p \vee \exists \bigcirc X$$

$$\mu X. ((\nu Y. p \wedge \forall \bigcirc Y) \vee \exists \bigcirc X)$$

$$\nu X. (p \wedge \exists \bigcirc \nu Y. (q \wedge \forall \bigcirc Y \vee \exists \bigcirc X))$$

$$\nu X. \mu Y. ((p \wedge X) \vee \exists \bigcirc Y)$$

- Alternation-free mu-calculus: alternation depth is 1

From CTL to μ -calculus

- Every CTL operator can be expressed in alternation-free μ -calculus

- $\exists\Diamond p$ equals $\mu X. (p \vee \exists\bigcirc X)$

- Fixpoint characterization of until: Observe

$$(\phi \exists\mathcal{U} \psi) \leftrightarrow \psi \vee (\phi \wedge (\phi \exists\mathcal{U} \psi)).$$

- $\phi \exists\mathcal{U} \psi$ equals $\mu X. (\psi \vee (\phi \wedge \exists\bigcirc X))$

- Characterizing $\exists\Box$ (over observation structure i.e. no fairness): $\exists\Box p$ equals $\nu X. (p \wedge \exists\bigcirc X)$

- Theorem: Every CTL formula ϕ is equivalent to an alternation-free μ -calculus formula of length $O(|\phi|)$.

- We can freely use CTL connectives in μ -calculus

Alternation-free μ -calculus versus CTL

- CTL cannot count: recall even property: p is true in every even state
- μ -calculus formula for even: $\nu X. (p \wedge \forall \bigcirc \forall \bigcirc X)$
- μ -calculus can express alternating reachability (winning in AND-OR graphs)

$$\mu X. (q \vee (p \wedge \exists \bigcirc X) \vee (\neg p \wedge \forall \bigcirc X))$$

- CTL cannot express alternating reachability
- Alternation-free μ -calculus is more expressive than CTL
- Distinguishing powers of μ -calculus and CTL are same: bisimilarity (bisimilar states satisfy the same set of μ -calculus formulas)

Specifying Büchi constraint

- Büchi fairness: a specified region repeats infinitely often
- Define the repetition operator $\Box\Diamond$: $s \models \exists\Box\Diamond p$ iff there is a source- s p -fair ω -trajectory
- To express $\exists\Box\Diamond$ in μ -calculus, we need nested fixpoints
- $\llbracket \exists\Box\Diamond p \rrbracket$ is the maximal region σ such that from every state in σ , some state in $\sigma \cap \llbracket p \rrbracket$ is reachable in one or more steps.
- This translates to $\nu X. \exists\Diamond^+(X \wedge p)$, that is,

$$\nu X. \mu Y. \exists\bigcirc ((X \wedge p) \vee Y)$$

- Computation has two loops. outer loop computes

$$\llbracket true \rrbracket \cap \llbracket \exists\Diamond^+ p \rrbracket \cap \llbracket \exists\Diamond^+(p \wedge \exists\Diamond^+ p) \rrbracket \cap \dots$$

More on Büchi constraints

- $\exists\Box\Diamond$ is not expressible in CTL
- $\exists\Box\Diamond$ is not expressible in alternation-free μ -calculus (thus, nesting is essential)
- The strategy generalizes to
 - Multiple Büchi constraints
 - Weak-fairness constraints that require repetition of actions

- Action α specified as $(p_0 \wedge q'_0) \vee \dots \vee (p_k \wedge q'_k)$

- Source region of α -fair ω -trajectories is

$$\nu X. \exists\Diamond \vee 0 \leq i \leq k. (p_i \wedge \exists\bigcirc (q_i \wedge X))$$

- Example: Fairness wrt the choice

$$pc_1 = in \rightarrow pc'_1 := out$$

is expressed by the formula

$$\nu X. \exists\Diamond [(pc_1 = in \wedge \exists\bigcirc (pc_1 = out \wedge X)) \vee (pc_1 \neq in \wedge \exists\bigcirc X)]$$

Single Streett constraints

- Single Streett constraint: if p -fair then q -fair
- Equivalent to

$$\exists \diamond (\exists \square \neg p \vee \exists \square \diamond q)$$

- Single Streett can be specified in μ -calculus using alternation depth 2

Multiple Streett constraints

- For $0 \leq i \leq k$, if p_i -fair then q_i -fair
- Equivalent to: there exists a subset $I \subseteq \{0 \dots k\}$ such that eventually
 - all states satisfy $\neg p_i$ for $i \notin I$
 - infinitely many states satisfy q_i for $i \in I$
- This leads to an exponential-size formula in μ -calculus
- Improved formulation (Emerson-Lei Theorem)

$$\exists \diamond \nu X. \wedge (p, q) \in F. [\exists \bigcirc (X \exists \mathcal{U} (q \wedge X)) \vee (\neg p \wedge \exists \bigcirc X)].$$

From CTL over fair structures to μ -calculus

- Suppose \mathcal{K} is a fair structure with Streett assumption F
- The characteristic region $\llbracket \exists \square p \rrbracket$ equals

$$p \exists \mathcal{U} \nu X. p \wedge \wedge (q, r) \in F. [\exists \bigcirc (X \exists \mathcal{U} (r \wedge X)) \vee (\neg q \wedge \exists \bigcirc X)].$$
- Generalizes when constraints are pairs of actions
- Theorem: For every CTL formula ϕ and a fair structure $\mathcal{K} = (K, F)$, there exists a formula ψ of μ -calculus formula of alternation depth 2 such that $\llbracket \phi \rrbracket_{\mathcal{K}} = \llbracket \psi \rrbracket_K$ and $|\psi| = O(|\phi| \cdot |F|)$.

Model checking

- Input: a closed formula ϕ of μ -calculus and an observation structure K
- Output: the characteristic region $\llbracket \phi \rrbracket$
- Symbolic computation using recursive function $Eval$

Recursive evaluation

```
function Eval
input  $\psi$ : form
  case  $p$ :  $\sigma := AtomEval(p, K)$ 
  case  $\neg p$ :  $\sigma := \Sigma \setminus AtomEval(p, K)$ 
  case  $\chi_1 \vee \chi_2$ :  $\sigma := Eval(\chi_1) \cup Eval(\chi_2)$ 
  case  $\chi_1 \wedge \chi_2$ :  $\sigma := Eval(\chi_1) \cap Eval(\chi_2)$ 
  case  $\exists \bigcirc \chi$ :  $\sigma := PreQueue(Eval(\chi), K)$ 
  case  $\forall \bigcirc \chi$ :  $\sigma := \Sigma \setminus PreQueue(\Sigma \setminus$ 
     $Eval(\chi), K)$ 
  case  $\mu X. \chi$ :
     $\mathcal{E}(X) := \emptyset$ ;
    repeat
       $\sigma := \mathcal{E}(X)$ ;  $\mathcal{E}(X) := Eval(\chi)$ 
    until  $\sigma = \mathcal{E}(X)$ ;
  case  $\nu X. \chi$ :
     $\mathcal{E}(X) := \Sigma$ ;
    repeat
       $\sigma := \mathcal{E}(X)$ ;  $\mathcal{E}(X) := Eval(\chi)$ 
    until  $\sigma = \mathcal{E}(X)$ ;
  case  $X$ :  $\sigma := \mathcal{E}(X)$ ;
end case
return  $\sigma$ 
```

Analysis

- Can be implemented using BDDs
- To check $K \models \phi$ check $\sigma^I \subseteq \llbracket \phi \rrbracket$
- Complexity: $O(n^k)$ where k is the size of the formula (nesting depth of the expression)
- Improvements to obtain $O(n^d)$, where d is the nesting depth
 - Do not reevaluate closed formulas
 - Exploit monotonicity