

Lecture 19: Nov 20  
Specifying Büchi constraint

- Büchi fairness: a specified region repeats infinitely often
- Define the repetition operator  $\Box\Diamond$ :  $s \models \Box\Diamond p$  iff there is a source- $s$   $p$ -fair  $\omega$ -trajectory
- To express  $\Box\Diamond$  in  $\mu$ -calculus, we need nested fixpoints
- $\llbracket \Box\Diamond p \rrbracket$  is the maximal region  $\sigma$  such that from every state in  $\sigma$ , 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

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

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

- Source region of  $\alpha$ -fair  $\omega$ -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  $\mu$ -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  $\mu$ -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  $\mu$ -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  $\phi$  and a fair structure  $\mathcal{K} = (K, F)$ , there exists a formula  $\psi$  of  $\mu$ -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  $\phi$  of  $\mu$ -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  $\exists \bigcirc \chi$ :  $\sigma := Pre(Eval(\chi), K)$ 
  case  $\forall \bigcirc \chi$ :  $\sigma := \Sigma \setminus Pre(\Sigma \setminus Eval(\chi), K)$ 
  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

## Automata-theoretic liveness verification

- $\omega$ -automata: finite observation structure + accepting conditions
- Theory of  $\omega$ -regular languages
- Useful for specifying liveness requirements of fair modules

## Types of $\omega$ -automata

- Büchi: accepting condition specified by a repeating region  $\sigma^A$ . Only  $\sigma^A$ -fair trajectories are accepted
- Multi-Büchi: set  $F$  of repeating regions
- CoBüchi: accepting condition specified by a stable region  $\sigma^A$ . Accepting trajectories should eventually stay within  $\sigma^A$
- Streett: accepting condition is a set  $F$  of region-pairs:

$$\bigwedge (\sigma, \tau) \in F. [\neg(\sigma\text{-fair}) \vee \tau\text{-fair}],$$

- Rabin: syntactically like Streett, but semantically its dual:

$$\bigvee (\sigma, \tau) \in F. [\sigma\text{-fair} \wedge \neg(\tau\text{-fair})].$$

- Muller: set  $F$  of regions.  $\underline{s}$  is accepted if the set  $\{s \in \Sigma \mid s_i = s \text{ for infinitely many } i \geq 0\}$  is in  $F$

## Specifications using $\omega$ -automata

- Automaton logic for liveness specification
- Formulas are  $\omega$ -automata whose observations are boolean expressions over module variables
- Model checking problem reduces to language-containment for  $\omega$ -automata

## Product of $\omega$ -automata

- All classes of  $\omega$ -automata are closed under product (intersection)
- Let  $\mathcal{M}_1 = (K_1, F_1)$  and  $\mathcal{M}_2 = (K_2, F_2)$  be two Streett automata. The product  $\mathcal{M}_1 \times \mathcal{M}_2$  is  $(K_1 \times K_2, \{(\sigma \uparrow, \tau \uparrow) \mid (\sigma, \tau) \in F_1 \cup F_2\})$ .
- Product of Büchi automata  $\mathcal{M}_1$  and  $\mathcal{M}_2$ : introduce a counter that makes sure that both accepting regions are visited repeatedly

## Complementation

- How do we complement a Büchi automaton?
- Can we determinize?
- Subset construction does not work. What if a subset has repeating and nonrepeating states?
- Theorem: For a Büchi automaton  $\mathcal{M}$ , there exists a finite congruence  $\sim$  over  $A^*$  such that both the  $\omega$ -language of  $\mathcal{M}$ , and its complement, are unions of  $L_1 \cdot L_2^\omega$  where  $L_1$  and  $L_2$  are equivalence classes of  $\sim$
- It follows that Büchi automata are closed under complement
- This approach constructs an automaton for the complement with  $2^{3n^3+4n^2+1}$  states
- Best known (and optimal) complementation construction:  $2^{O(n \log n)}$

## $\omega$ -regular languages

- The  $\omega$ -language  $\mathcal{L}$  is a regular-safety language if there is a regular language  $L$  such that  $\mathcal{L} = \text{safe}(L)$ .
- Regular-guarantee, regular-response, and regular-persistence languages defined similarly
- The  $\omega$ -language  $\mathcal{L}$  is  $\omega$ -regular if it is a boolean combination of regular-response and regular-persistence languages.
- All the regular subclasses have similar properties as the original classes

## Expressiveness of $\omega$ -automata

- Theorem: An  $\omega$ -language  $\mathcal{L}$  is  $\omega$ -regular iff it is accepted by a Büchi automaton
- Other types of automata such as Streett, Rabin, Muller have the same expressive power
- The same class can be defined using  $\omega$ -regular expressions

$$\varphi := a \mid \varphi \cdot \varphi \mid \varphi + \varphi \mid \varphi^* \mid \varphi^\omega$$

- Deterministic Streett also accept all  $\omega$ -regular languages, but deterministic Büchi are less expressive (cannot define  $A^*a^\omega$ )
- Theorem: An  $\omega$ -language  $\mathcal{L}$  is regular-response language iff it is accepted by a deterministic Büchi automaton