

Lecture 14: Nov 1
Beyond Safety Requirements

- Safety requirement is a requirement that can be violated by a finite trace
- Examples: mutual-exclusion, equal-opportunity
- Safety means “something bad never happens”
- Liveness: “something good eventually happens”
- Example: a requesting process eventually enters critical section
- STL formula

$$\forall \square (pc_1 = reqC \rightarrow \exists \diamond pc_1 = inC)$$

is weaker (guarantees only possible entry)

- We want entry on all paths (starvation freedom):

$$\forall \square (pc_1 = reqC \rightarrow \forall \diamond pc_1 = inC)$$

Update Commands in Pete

module P_1 **is**

update

$\parallel pc_1 = outcs$	$\rightarrow pc'_1 := reqcs; x'_1 := x_2$
$\parallel pc_1 = reqcs \wedge pc_2 = outcs$	$\rightarrow pc'_1 := incs$
$\parallel pc_1 = reqcs \wedge x_1 \neq x_2$	$\rightarrow pc'_1 := incs$
$\parallel pc_1 = incs$	$\rightarrow pc'_1 := outcs$
$\parallel true$	\rightarrow

module P_2 **is**

update

$\parallel pc_2 = outcs$	$\rightarrow pc'_2 := reqcs; x'_2 := \neg x_1$
$\parallel pc_2 = reqcs \wedge pc_1 = outcs$	$\rightarrow pc'_2 := incs$
$\parallel pc_2 = reqcs \wedge x_1 = x_2$	$\rightarrow pc'_2 := incs$
$\parallel pc_2 = incs$	$\rightarrow pc'_2 := outcs$
$\parallel true$	\rightarrow

Checking Liveness

- We will enrich specification languages so that liveness properties can be specified
- Obstacle: considering all infinite trajectories does not make sense
- *Pete* does not satisfy starvation freedom
 $(out, 0, out, 0)(req, 0, out, 0)(req, 0, out, 0)(req, 0, out, 0) \dots$
- Problem: modeling of asynchrony using nondeterminism (what if we choose to stutter at every step?)
- Not all infinite trajectories are realistic
- Plan: Change the model so that we can rule out certain infinite behaviors

Towards Fairness

- Nondeterministic update:

update x

$\parallel true \rightarrow x' := 0$

$\parallel true \rightarrow x' := 1$

- We want to say: both choices are treated fairly so that $0000\dots$ is not an acceptable behavior
- Fairness: specifies what should happen in the limit
- Fair modules = reactive modules + fairness on update choices
- Fair nondeterministic update

update x **weaklyfair** a, b

$\parallel true \xrightarrow{a} x' := 0$

$\parallel true \xrightarrow{b} x' := 1$

Motivation

- Reformulate model checking as well as refinement checking
 - Fairness in modules
 - Liveness in requirements
- First, let us study theory of ω -languages
- ω -word over A : infinite word over alphabet A
- ω -language over A : set of ω -words
- Questions:
 - What is the structure of ω -languages?
 - How do we define ω -languages from ordinary languages?

ω -words and ω -languages

- Notation:
 - word: \bar{a} , ω -word: \underline{a}
 - language: L , ω -language: \mathcal{L}
- Prefix, suffix, concatenation
- $\text{pref}(\underline{a})$: set of prefixes of \underline{a}
- Infinite repetition:
 - \bar{a}^ω : infinitely many copies of \bar{a}
 - L^ω is an ω -language
- Periodic words: \bar{a}^ω
- Eventually periodic words: $\bar{a} \cdot \bar{b}^\omega$
- Union, intersection, complementation of ω -languages
- Limit: An ω -word \underline{a} is called a limit of the language L if $\text{pref}(\underline{a}) \subseteq L$.

- An ω -word \underline{a} is a limit of the ω -language \mathcal{L} if it is a limit of $\text{pref}(\mathcal{L})$ (i.e. every finite prefix of \underline{a} can be extended to an ω -word in \mathcal{L}).
- The ω -language \mathcal{L} is limit-closed if it contains all its limits
- Examples:
 - $(b^*a)^\omega$: ω -words with infinitely many a
 - $(ba)^\omega$
 - $(a+b)^*b^\omega$: only finitely many a

Safety languages

- For a language L over A , $\mathit{safe}(L)$ is the ω -language

$$\{\underline{a} \mid \forall i \geq 0. \bar{a}_{0..i} \in L\}$$

(all prefixes are in L)

- The ω -language \mathcal{L} is a safety language if there is a language L such that $\mathcal{L} = \mathit{safe}(L)$.
- Theorem: \mathcal{L} is a safety language iff it is limit-closed
- A safety language \mathcal{L} is completely specified by $\mathit{pref}(\mathcal{L})$
- Examples
 - $(Aa)^\omega$: every alternate symbol is a
 - a^ω : only a symbols
 - A^*bA^ω : at least one b
 - $(b^*a)^\omega$: infinitely many a symbols
 - ω -words \underline{a} such that for all $i \geq 0$, if i is a prime number, then $a_i = a$

Closure properties of safety

- Intersection of two safety languages is safe

$$\mathit{safe}(L_1) \cap \mathit{safe}(L_2) = \mathit{safe}(L_1 \cap L_2)$$

- safe does not distribute over union:

$$\mathit{safe}(L_1) \cup \mathit{safe}(L_2) \neq \mathit{safe}(L_1 \cup L_2)$$

- Union of two safety languages is a safety language:

$$\mathit{safe}(L_1) \cup \mathit{safe}(L_2) = \mathit{safe}(\mathit{pref}(L_1) \cup \mathit{pref}(L_2))$$

- Safety languages are not closed under complementation: a^ω is safe, but not A^*bA^ω
- Complementing a safety language gives a guarantee language

Guarantee Languages

- For a language L over A , $guar(L)$ is the ω -language

$$\{\underline{a} \mid \exists i \geq 0. \bar{a}_{0..i} \in L\}$$

(some prefix is in L)

- The ω -language \mathcal{L} is a guarantee language if there is a language L such that $\mathcal{L} = guar(L)$.
- Classical guarantee requirement: termination
- Thm. The ω -language \mathcal{L} is a guarantee language iff there is a language L such that $\mathcal{L} = L \cdot A^\omega$
- Examples:
 - A^*bA^ω
 - ω -words that contain at least 10 occurrences of a
 - $(b^*a)^\omega$: infinitely many a symbols

Closure properties of guarantee

- Thm: The ω -language \mathcal{L} is a safety language iff the complementary language $A^\omega \setminus \mathcal{L}$ is a guarantee language.

$$- \mathcal{L} = \text{safe}(L) \text{ iff } A^\omega \setminus \mathcal{L} = \text{guar}(A^+ \setminus L)$$

- By duality, guarantee properties are closed under union and inetersection
- guar dsitributes over union:

$$\text{guar}(L_1) \cup \text{guar}(L_2) = \text{guar}(L_1 \cup L_2)$$

- guar does not dsitribute over intersection, but still closed under intersection

$$\text{guar}(L_1) \cap \text{guar}(L_2) = \text{guar}((L_1 \cdot A^*) \cap (L_2 \cdot A^*))$$

Obligation properties

- Obligation languages are obtained by finite boolean combinations of safety or guarantee languages
- Least class of ω -languages that
 - contains all safety languages
 - closed under all boolean operations
- The ω -language a^*ba^ω consisting of ω -words with precisely one b symbol, is an obligation language:
 $safe(a^*ba^* + a^*) \cap guar(a^*b)$.

Response languages

- For a language L over A , $recur(L)$ is the ω -language

$$\{\underline{a} \mid \forall j \geq 0. \exists i \geq j. \bar{a}_{0..i} \in L\}$$

(infinitely many prefixes in L)

- The ω -language \mathcal{L} is a response language if there is a language L such that $\mathcal{L} = recur(L)$.
- Typical example: every request is followed by a response (starvation freedom)
- Infinitely many b symbols:

$$(a^*b)^\omega = recur((a^*b)^*)$$

- All safety, guarantee, and obligation languages are response languages

Closure properties of response

- Response languages are closed under union:

$$\text{recur}(L_1) \cup \text{recur}(L_2) = \text{recur}(L_1 \cup L_2)$$

- Closure under intersection not immediately obvious:

$$\text{recur}(L_1) \cap \text{recur}(L_2) \neq \text{recur}(L_1 \cap L_2)$$

- $\text{recur}(L_1) \cap \text{recur}(L_2)$ equals $\text{recur}(L_{12})$, where $\bar{a}_{0\dots m}$ is in L_{12} if

- it is in L_2 and

- it has a prefix $\bar{a}_{0\dots k}$ in L_1 with $\bar{a}_{0\dots j} \notin L_2$ for $k < j < m$.

- Not closed under complement: complement of $(a^*b)^\omega$ is not a response language
- Alternative definition of response languages: \mathcal{L} is response if it is intersection of countably many guarantee languages

Persistence languages

- For a language L over A , $\mathit{persist}(L)$ is the ω -language

$$\{\underline{a} \mid \exists j \geq 0. \forall i \geq j. \bar{a}_{0..i} \in L\}$$

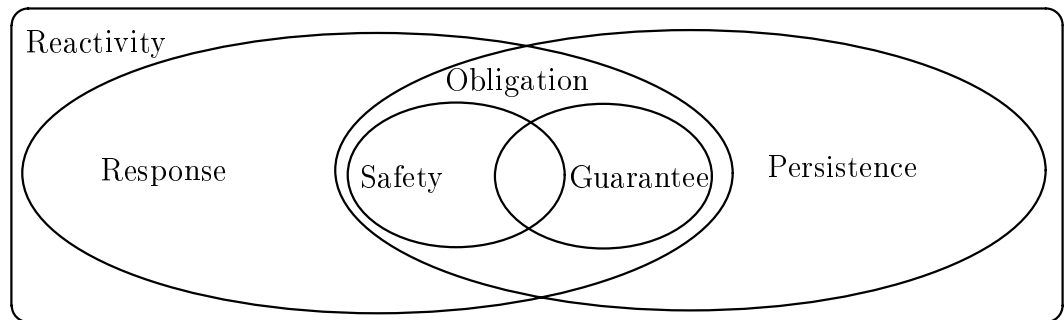
(eventually all prefixes in \mathcal{L})

- The ω -language \mathcal{L} is a persistence language if there is a language L such that $\mathcal{L} = \mathit{persist}(L)$.
- Typical persistence requirement: a protocol eventually stabilizes
- Only finitely many b symbols: $A^*a^\omega = \mathit{persist}(A^*a^*)$
- \mathcal{L} is a response language iff the complementary language $A^\omega \setminus \mathcal{L}$ is a persistence language.
- Persistence languages are closed under union, intersection, but not under complementation

Reactivity languages

- The ω -language \mathcal{L} is a 1-reactivity language if there exists a persistence language \mathcal{L}_1 and a response language \mathcal{L}_2 such that $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.
- The ω -language \mathcal{L} is a k -reactivity language, for a natural number k , if there exist k 1-reactivity languages $\mathcal{L}_1, \dots, \mathcal{L}_k$ such that $\mathcal{L} = \mathcal{L}_1 \cap \dots \cap \mathcal{L}_k$
- Class of reactivity languages is closed under all boolean operations
- Example: For $A = \{a, b, c\}$,
 - the ω -language consisting of ω -words with infinitely many b symbols or only finitely many a symbols is a 1-reactivity language
 - ω -language consisting of ω -words with infinitely many b symbols and only finitely many a symbols is a 2-reactivity language

Classes of ω -languages



Safety-Liveness Classification

- Safe languages: limit-closed, defined by finite prefixes
- Live languages: finite prefixes give no information
- The ω -language \mathcal{L} is live if $\text{pref}(\mathcal{L}) = A^*$ (every finite word can be extended to \mathcal{L}).
- The ω -language A^ω is the only ω -language over the alphabet A that is both safe and live
- Machine-closure: Given a safe ω -language \mathcal{L}_S and a live ω -language \mathcal{L}_L , the pair $(\mathcal{L}_S, \mathcal{L}_L)$ is machine-closed if $\text{pref}(\mathcal{L}_S \cap \mathcal{L}_L) = \text{pref}(\mathcal{L}_S)$.
- Intuition: after a finite prefix, if safety is not violated then there is a possibility to produce an extension in $\mathcal{L}_S \cap \mathcal{L}_L$
- Examples:
 - $\mathcal{L}_1 = (b^*a)^\omega$ is live

- $\mathcal{L}_2 = ((a + b)a)^\omega$ is safe
 - $(\mathcal{L}_2, \mathcal{L}_1)$ is machine-closed
 - $\mathcal{L}_4 = a^\omega + b^\omega$ is safe
 - $(\mathcal{L}_4, \mathcal{L}_1)$ is not machine-closed
- Set of infinite executions will be specified by a machine-closed pair
 - Safety component specified by transition relation
 - Liveness component specified by fairness constraints
 - Machine closure implies that fairness can be ignored while verifying safety properties
 - Theorem: Let $(\mathcal{L}_S, \mathcal{L}_L)$ be a machine-closed specification of the ω -language \mathcal{L} , and let \mathcal{L}' be a safe language. Then, $\mathcal{L} \subseteq \mathcal{L}'$ iff $\mathcal{L}_S \subseteq \mathcal{L}'$

Topological Characterization

- All the classes can be defined using classical topological concepts
- Consider metric d over the set of all ω -words
 - $d(\underline{a}, \underline{a})$ equals 0
 - $d(\underline{a}, \underline{b})$ equals $1/2^i$ if i is the maximum integer j such that $\bar{a}_{0\dots j} = \bar{b}_{0\dots j}$
- Safe languages: closed sets
- Live languages: dense sets
- Guarantee languages: open sets
- Response languages: countable intersections of open sets G_δ
- Persistence languages: countable unions of closed sets F_σ