Minimal Safety Criterion Generator G_{safe}— Formal Specification

---

1. State Space

Define the system state space as a finite set:

$$X := \{0,1,2,3,4,5,6,7,8,9\}. \qquad (1.1)$$

---

2. Criterion Language

2.1 Atomic Criteria

For any $i \in X$, define the atomic criterion function:

$$a_i(x) := \begin{cases} 1, & x = i, \\ 0, & x \neq i. \end{cases} \qquad (2.1)$$

2.2 Criterion Language $\mathcal{L}$

The criterion language is defined as the closure of atomic criteria under Boolean operations:

$$\mathcal{L} := \mathrm{Bool}(\{a_i \mid i \in X\}). \qquad (2.2)$$

The allowed Boolean operators are $\neg$, $\wedge$, and $\vee$.

---

3. Criterion Set and Encoding

3.1 Criterion Set

Let the current criterion set adopted by the system be a finite set:

$$P := \{p_1, p_2, \dots, p_m\}, p_i \in \mathcal{L}. \qquad (3.1)$$

3.2 Criterion Encoding Mapping

Define the encoding vector of a state under the criterion set P:

$$\phi_P(x) := (p_1(x), p_2(x), \dots, p_m(x)) \in \{0,1\}^m. \qquad (3.2)$$

3.3 Criterion-Induced Equivalence Relation

Define the equivalence relation induced by the criterion set:

$$x \sim_P y \iff \phi_P(x) = \phi_P(y). \qquad (3.3)$$

This induces a partition of the state space:

$$X/\sim_P. \qquad (3.4)$$

---

4. Candidate Criterion Complexity

4.1 Syntactic Complexity

Define the syntactic complexity of a criterion as the number of nodes in its syntax tree:

$$\mathcal{C}(p) := \#(\text{syntax nodes of } p). \qquad (4.1)$$

4.2 Complexity Upper Bound

Given a global complexity upper bound $b \in \mathbb{N}$, the allowed candidate criterion set is:

$$\mathcal{L}_{\le b} := \{p \in \mathcal{L} \mid \mathcal{C}(p) \le b\}. \qquad (4.2)$$

---

5. Responsibility-Domain Constraints

5.1 Responsibility Mapping

Define a mapping from states to responsibility domains:

$$\rho: X \to \mathcal{R}, \qquad (5.1)$$

where $\mathcal{R}$is a finite set of responsibility domains.

5.2 Atomic Dependency Set

Define the atomic dependency set of a criterion:

$$\mathrm{Dep}(p) := \{i \in X \mid a_i \text{ appears in the syntax tree of } p\}. \qquad (5.2)$$

5.3 Responsibility Set of a Criterion

Define the responsibility set of a criterion:

$$\mathrm{Resp}(p) := \{\rho(i) \mid i \in \mathrm{Dep}(p)\}. \qquad (5.3)$$

5.4 Responsibility-Domain Upper Bound

Given a responsibility-domain bound $r \in \mathbb{N}$, require:

$$|\mathrm{Resp}(p)| \leq r. \qquad (5.4)$$

---

6. Non-Expansion (Refinement) Constraints

6.1 Binary Split Within an Equivalence Class

For any equivalence class $c \in X/{\sim_P}$, define:

$$c_0 := \{x \in c \mid p(x) = 0\}, c_1 := \{x \in c \mid p(x) = 1\}. \qquad (6.1)$$

Singleton equivalence classes are treated as maximally unstable splits.

6.2 Intra-Class Bias Measure

Define the bias of a criterion on an equivalence class:

$$\beta(c,p) := \frac{\big|\, |c_1| - |c_0| \,\big|}{|c|}. \qquad (6.2)$$

6.3 Global Maximum Bias

Define the global maximum bias of a criterion:

$$B(P,p) := \max_{c \in X/\sim_P} \beta(c,p). \qquad (6.3)$$

6.4 Bias Upper Bound

Given a bias threshold $\varepsilon \in [0,1]$, require:

$$B(P,p) \leq \varepsilon. \qquad (6.4)$$

---

7. Rollback Constraints

7.1 Criterion Increment Principle

The evolution of the criterion set is only allowed via single-element increments:

$$P' := P \cup \{p\}. \qquad (7.1)$$

7.2 Rollback Operator

Define the rollback operator:

$$R(P',p) := P' \setminus \{p\}. \qquad (7.2)$$

7.3 Rollback Consistency

Require:

$$R(P \cup \{p\}, p) = P. \qquad (7.3)$$

---

8. Criterion Acceptance Condition

Define the criterion acceptance predicate:

$$\mathrm{Accept}(P,p) := \mathbf{1}[p \in \mathcal{L}_{\leq b} \ \wedge \ |\,\mathrm{Resp}(p)\,| \leq r \ \wedge \ B(P,p) \leq \varepsilon]. \qquad (8.1)$$

9. Criterion Generation Operator

9.1 Definition

Define the minimal safety criterion generation operator:

$$G_{\text{safe}}(P) := \begin{cases} P \cup \{p^{\backslash *}\}, & p^{\backslash *} = \arg \min_{p \in \mathcal{L}_{\le b},\ \text{Accept}(P,p)=1} J(P,p), \\ P, & \text{if no acceptable } p \text{ exists.} \end{cases} \tag{9.1}$$

9.2 Scoring Function

Define the scoring function as a lexicographic objective:

$$J(P,p) := (\mathcal{C}(p),\ |\text{Resp}(p)|,\ B(P,p)). \tag{9.2}$$

10. Termination and Decidability

Since X, $\mathcal{L}_{\le b}$, and $\mathcal{R}$are all finite sets,
all predicates, operators, and optimisation procedures defined above are decidable
and guaranteed to terminate.