

# CS 515: Specification and Verification with Stable Failures

Stefan D. Bruda

Winter 2019

## PROPERTY-ORIENTED SPECIFICATIONS



- The specification is a pair of predicates  $(S_T(tr), S_{SF}(tr, X))$ , one on traces and the other on stable failures

$$P \text{ sat } S(tr) = \forall tr \in \text{traces}(P) : S_T(tr) \wedge \forall (tr, X) \in \mathcal{SF}[P] : S_{SF}(tr, X)$$

- Safety specifications (“nothing bad will happen”) still captured by the predicate on traces  $S_T(tr)$
- Liveness specifications (“something good will happen”) can now be captured by the other predicate  $S_{SF}(tr, X)$ 
  - Liveness is expressed in terms of the ability of a process to engage in actions
  - That is, the process is guaranteed to offer (that is, not refuse after stabilizing) certain action at a certain point of the execution
  - The “certain point in the execution” is given by the trace, the “guarantee to offer certain action” by the refusal set
- We will not concern ourselves here with the trace properties  $S_T(tr)$  (we already saw those), but only with the stable failure properties  $S_{SF}(tr, X)$

## VERIFICATION RULES FOR BASIC PROCESSES



- STOP:**  $\frac{}{STOP \text{ sat } tr = \langle \rangle}$
- SKIP:**  $\frac{}{SKIP \text{ sat } tr = \langle \rangle \wedge \checkmark \in X \vee tr = \langle \checkmark \rangle}$
- DIV** satisfies everything, including *false!*  $\frac{}{DIV \text{ sat } S_{SF}(tr, X)}$
- CHAOS** only satisfies *true*:  $\frac{}{CHAOS \text{ sat } true(tr, X)}$
- RUN:**  $\frac{}{RUN \text{ sat } \checkmark \notin \sigma(tr) \implies X = \{\}}$

## PREFIX AND CHOICE



- Prefix:**  $\frac{P \text{ sat } S_{SF}(tr, X)}{a \rightarrow P \text{ sat } tr = \langle \rangle \wedge a \notin X \vee \text{head}(tr) = a \wedge S_{SF}(\text{tail}(tr), X)}$   
 $\frac{\forall a \in A : P(a) \text{ sat } S_a(tr, X)}{x : A \rightarrow P(x) \text{ sat } tr = \langle \rangle \wedge A \cap X = \{\}} \vee \frac{\exists a \in A : \text{head}(tr) = a \wedge S_a(\text{tail}(tr), X)}$
- Input and output:**  $\frac{P \text{ sat } S_{SF}(tr, X)}{c!v \rightarrow P \text{ sat } tr = \langle \rangle \wedge c.v \notin X \vee \text{head}(tr) = c.v \wedge S_{SF}(\text{tail}(tr), X)}$   
 $\frac{\forall v \in T : P(v) \text{ sat } S_v(tr, X)}{c?x : T \rightarrow P(x) \text{ sat } tr = \langle \rangle \wedge \text{in}.T \cap X = \{\}} \vee \frac{\exists v \in T : \text{head}(tr) = c.v \wedge S_v(\text{tail}(tr), X)}$
- External choice:**  $\frac{P_1 \text{ sat } S_1(tr, X) \quad P_2 \text{ sat } S_2(tr, X)}{P_1 \sqcap P_2 \text{ sat } (tr = \langle \rangle \implies S_1(tr, X) \wedge S_2(tr, X)) \wedge (tr \neq \langle \rangle \implies (S_1(tr, X) \vee S_2(tr, X)))}$
- Internal choice:**  $\frac{P_1 \text{ sat } S_1(tr, X) \quad P_2 \text{ sat } S_2(tr, X)}{P_1 \sqcap P_2 \text{ sat } S_1(tr, X) \vee S_2(tr, X)}$



- Alphabetized parallel:

$$\frac{P_1 \text{ sat } S_1(tr, X) \quad P_2 \text{ sat } S_2(tr, X)}{P_1 \parallel_{A_1, A_2} P_2 \text{ sat } \exists X_1, X_2 : S_1(tr \upharpoonright A_1^\vee, X_1) \wedge S_2(tr \upharpoonright A_2^\vee, X_2) \wedge \sigma(tr) \subseteq (A_1 \cup A_2)^\vee \wedge X \cap (A_1 \cup A_2)^\vee = (X \cap A_1^\vee) \cup (X \cap A_2^\vee)}$$

- Interleaving:

$$\frac{P_1 \text{ sat } S_1(tr, X) \quad P_2 \text{ sat } S_2(tr, X)}{P_1 \parallel\parallel P_2 \text{ sat } \exists tr_1, tr_2, X_1, X_2 : S_1(tr_1, X_1) \wedge S_2(tr_2, X_2) \wedge tr \text{ interleaves } tr_1, tr_2 \wedge X_1 \cup X_2 = X \wedge X_1 \setminus \{\checkmark\} = X_2 \setminus \{\checkmark\}}$$

- Hiding:

$$\frac{P \text{ sat } S(tr, X)}{P \setminus A \text{ sat } \exists tr_1 : S(tr_1, X \cup A) \wedge tr = tr_1 \setminus A}$$

$$\frac{P \text{ sat } S(tr, X)}{P \setminus A \text{ sat } S(tr, X)} \quad [\forall tr, X : S(tr, X) \text{ iff } S(tr \setminus A, X \setminus A)]$$



- Renaming:

$$\frac{P \text{ sat } S(tr, X)}{\exists tr_1 : S(tr_1, f^{-1}(X)) \wedge f(tr_1) = tr}$$

$$\frac{P \text{ sat } S(tr, X)}{f^{-1}(P) \text{ sat } S(f(tr), f(X))}$$

- Sequential composition:

$$\frac{P_1 \text{ sat } (S_f(tr), S_1(tr, X)) \quad P_2 \text{ sat } S_2(tr, X)}{P_1; P_2 \text{ sat } \checkmark \notin \sigma(tr) \wedge S_1(tr, X \cup \{\checkmark\}) \vee \exists tr_1, tr_2 : tr = tr_1 \hat{\ } tr_2 \wedge S_T(tr_1 \hat{\ } \langle \checkmark \rangle) \wedge S_2(tr_2, X)}$$

- Interrupt:

$$\frac{P_1 \text{ sat } (S_f(tr), S_1(tr, X)) \quad P_2 \text{ sat } S_2(tr, X)}{P_1 \triangle P_2 \text{ sat } S_1(tr, X) \wedge (S_2(\langle \rangle, X) \vee \checkmark \in \sigma(tr)) \vee \exists tr_1, tr_2 : tr = tr_1 \hat{\ } tr_2 \wedge \checkmark \notin \sigma(tr_1) \wedge S_T(tr_1) \wedge S_2(tr_2, X)}$$



$$\frac{\forall Y : Y \text{ sat } S_{SF}(tr, X) \implies F(Y) \text{ sat } S_{SF}(tr, X)}{N \text{ sat } S_{SF}(tr, X)} \quad [N = F(N)]$$

- No basis needed, because all specifications are satisfied by DIV



- Refinement relation based on set inclusion:  $(T_1, SF_1) \sqsubseteq_{SF} (T_2, SF_2)$  iff  $T_2 \subseteq T_1 \wedge SF_2 \subseteq SF_1 \implies$  obvious definition for  $P_1 \sqsubseteq_{SF} P_2$ 
  - The stable failures and traces of  $P_2$  do not introduce any new behaviour to  $P_1$
  - Alternative definition:  $P_1 \sqsubseteq_{SF} P_2$  iff  $P_1 = P_1 \sqcap P_2$
- Same as trace refinement: An implementation  $I$  meets a specification  $S$  ( $S \sqsubseteq_{SF} I$ ) whenever all the possible behaviours of  $I$  are allowed by  $S$ 
  - The difference being that now behaviours are defined by stable failures
- Stable failure refinement also captured by **failure trace testing**
  - Tests as before, plus a special action  $\theta$  (deadlock) with the lowest priority:

$$\frac{P \xrightarrow{\tau} P'}{P \parallel T \xrightarrow{\tau} P' \parallel T} \quad \frac{T \xrightarrow{\tau} t'}{P \parallel T \xrightarrow{\tau} P' \parallel T} \quad \frac{P \xrightarrow{a} P' \quad T \xrightarrow{a} T'}{P \parallel T \xrightarrow{a} P' \parallel T'} \quad a \in \Sigma$$

$$\overline{P \parallel \text{SUCCESS} \xrightarrow{\omega} \text{STOP}}$$

$$\frac{T \xrightarrow{\theta} T'}{P \parallel T \xrightarrow{\theta} P \parallel T'} \quad \neg(\exists x \in \Sigma \cup \{\tau, \omega\} : P \parallel T \xrightarrow{x})$$



- Let  $\text{init}(P) = \{a \in \Sigma : P \xrightarrow{a}\}$
- Let  $P \xrightarrow{\varepsilon} P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} P_n$
- Then  $f = A_0 a_1 A_1 a_2 A_2 \dots a_n A_n$ ,  $n \geq 0$ ,  $a_i \in \Sigma^*$ ,  $A_i \subseteq \Sigma$  is a **failure trace** of  $P$  if:
  - If  $\neg(p_i \xrightarrow{\tau})$ , then  $A_i \subseteq (A \setminus \text{init}(p_i))$ , and
  - If  $p_i \xrightarrow{\tau}$  then  $A_i = \{\}$
- We obtain a failure trace of  $P$  by taking a trace of  $P$  and inserting refusal sets after stable states

## Lemma

*Failure traces and stable failures describe the same behaviour*

- Every stable failure is a failure trace
- Every failure trace  $f = A_0 a_1 A_1 a_2 A_2 \dots a_n A_n$  is equivalent with the set of stable failures  $\{(\langle \rangle, A_0)\} \cup \{(\langle a_1, \dots, a_i \rangle, A_i) : i \geq 0\}$

# FAILURE TRACE TESTING VS STABLE FAILURES



## Lemma

*For any failure trace test  $T$  there exists a set  $T_s \subseteq ST$  such that  $P \text{ may } T$  if and only if  $\exists T' \in T_s : P \text{ may } T'$*

- $T_s = \{\text{seqt}(f) : f \text{ a failure trace of } T\}$

## Theorem

$P \sqsubseteq_{SF} Q$  iff  $P \text{ may } T \implies Q \text{ may } T$  for all failure trace tests  $T$

- We go from stable failures to failure trace tests (and the other way around) via failure traces and sequential tests



- The set of sequential failure trace tests  $ST$ :
  - $SUCCESS \in ST$
  - If  $T \in ST$  then  $a \rightarrow T \in ST$  for any  $a \in \Sigma$
  - If  $t \in ST$  then  $(\Box_{a \in A'} a \rightarrow STOP) \Box \theta \rightarrow T \in ST$  for any  $A' \subseteq A$

## Lemma

*There exists a bijection between failure traces and sequential tests*

- For a sequential test  $t$  the failure trace  $\text{ftr}(t)$  is:
  - $\text{ftr}(SUCCESS) = \{\}$
  - $\text{ftr}(a \rightarrow T') = a \text{ ftr}(T')$
  - $\text{ftr}((\Box_{a \in A'} a \rightarrow STOP) \Box \theta \rightarrow T') = A' \text{ ftr}(T')$
- Conversely, for a failure trace  $f$  the sequential test  $\text{seqt}(f)$  is:
  - $\text{seqt}(\{\}) = SUCCESS$
  - $\text{seqt}(af) = a \rightarrow \text{seqt}(f)$
  - $\text{seqt}(Af) = (\Box_{a \in A} a \rightarrow STOP) \Box \theta \rightarrow \text{seqt}(f)$
- Note that  $\text{ftr}(\text{seqt}(f)) = f$  and  $\text{seqt}(\text{ftr}(t)) = t$