Appendix B: Necessity of Basic Properties

B.1 Safety

Safety is violated if a component received unknown or unacceptable input events. The basic safety properties, if satisfied, ensure that a component will not receive unknown or unacceptable events.

B.1.1 Authenticity

Goal of Authenticity: No unknown event is produced. In this section we prove that authenticity, as defined, satisfies this goal.

Proof:

Assume that safety is violated by at least one unknown event, that is, there is at least one position in at least one of the output sequences where a component produces an event that is not a defined event. Prove that authenticity is not satisfied.

$$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \forall \alpha : \mathcal{F}ma\{\Phi(i(A)\} \cdot \sigma\{\mathbf{i}\}(out) \models_x \neg\alpha$$

$$= \exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \neg\exists \alpha : \mathcal{F}ma\{\Phi(i(A)\} \cdot \sigma\{\mathbf{i}\}(out) \models_x \alpha$$

$$= \exists x : \mathbb{N} \cdot \neg\forall i : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi(i(A)\} \cdot \sigma\{\mathbf{i}\}(out) \models_x \alpha$$

$$= \neg\forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi(i(A)\} \cdot \sigma\{\mathbf{i}\}(out) \models_x \alpha$$

$$= \neg \text{ Authenticity}$$ ❑

The proof of the violation of the environment assumption of authenticity follows. Assume that safety is violated by at least one unknown event, that is, there is at least one position in at least one of the sequences where the environment sends an event that is not a defined event. Prove that authenticity is not satisfied.

$$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \forall \alpha : \mathcal{F}ma\{\Phi(i(A)\} \cdot \sigma\{\mathbb{E}\}(out) \models_x \neg\alpha$$

$$= \exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \neg\exists \alpha : \mathcal{F}ma\{\Phi(i(A)\} \cdot \sigma(\mathbb{E};i)(out) \models_x \alpha$$

$$= \neg \text{ Authenticity}$$ ❑

B.1.2 Integrity

Goal of Integrity: No unacceptable modificaiton events are received at a component.

Proof:

Assume that safety is violated by at least one unacceptable modification event received from either a component or the environment, that is, there is at least one position in at least one of the input sequences containing an unacceptable modficiation event. Prove that integrity is not satisfied.

$$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \forall \alpha : \mathcal{F}ma\{\Phi(A,mod)\} \cdot \sigma(\mathbb{N}+\mathbb{E};i) \models_x \neg\alpha$$

$$\Rightarrow \neg\forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi(A,mod) \cup \Phi(obs)\} \cdot \sigma(\mathbb{N}+\mathbb{E};i) \models_x \alpha$$

$$= \neg \text{ Integrity}$$ ❑

B.1.3 Confidentiality

Goal of Confidentiality: No unacceptable observation events are received at a component.

Proof:

Assume that safety is violated by at least one unacceptable observation event received from either a component or the environment, that is, there is at least one position in at least one of the input sequences containing an unacceptable observation event. Prove that confidentiality is not satisfied.

$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \forall \alpha : \mathcal{F}ma\{\Phi(A,obs)\} \cdot \sigma(\mathbb{N}^{+}\mathbb{E};i) \models_x \neg\alpha$

$\Rightarrow \neg\forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi(A,obs) \cup \Phi(mod)\} \cdot \sigma(\mathbb{N}^{+}\mathbb{E};i) \models_x \alpha$

$= \neg$ Confidentiality ❑

B.1.4 Provenance

Provenance is formally defined as a progress property although it is necessary for the safety of a system. This is because safety considers the role of the event when it is received yet provenance includes a specification of the temporal behaviour that leads to an event being received.
Goal of Provenance: No events are received from an unknown source.
Proof:
Assume that safety is violated by at least one event received from an unknown source, that is, there is at least one position in at least one of the input sequences where the event is sent from an unknown source. Prove that provenance is not satisfied.

$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \forall j : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi\} \cdot \sigma\{\mathbf{i}\}(in) \models_x \Diamond\alpha \wedge \neg\, \sigma(j^{+}\mathbb{E})(out) \models_x \alpha$

$= \neg\forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \exists j : \mathbb{N} \cdot \forall \alpha : \mathcal{F}ma\{\Phi\} \cdot (\sigma\{\mathbf{i}\}(in) \models_x \Diamond\alpha \Rightarrow \sigma(j^{+}\mathbb{E})(out) \models_x \alpha)$

$= \neg$ Provenance ❑

B.1.5 Utility

Utility is necessary in that it is the contraposition of provenance. Viewing a system, by symmetry, we are able to postulate that if no event is received from an unknown source (provenance is satisfied) then so too no event should be sent to an unknown source.
Goal of Utility: No events are sent to an unknown source.
Proof:
Assume that safety is violated by at least one event sent to an unknown source, that is, there is at least one position in at least one of the input sequences where the event is sent sent unknown source. Prove that utility is not satisfied.

$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \forall j : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi\} \cdot \sigma(j^{+}\mathbb{E})(out) \models_x \alpha \wedge \neg\sigma\{\mathbf{i}\}(in) \models_x \Diamond\alpha$

$= \neg\forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \exists j : \mathbb{N} \cdot \forall \alpha : \mathcal{F}ma\{\Phi\} \cdot (\sigma(j^{+}\mathbb{E})(out) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\}(in) \models_x \Diamond\alpha)$

$= \neg$ Utility ❑

B.2 Progress

Progress is violated if a component is unable to produce an output in response to a given input.

B.2.1 Accessibility

Goal of Accessibility: That every input received at a component will initiate a process at the component.
Proof:
Assume that an input is received and processing is not initiated. Prove that accessibility is not satisfied.

$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi\} \cdot \sigma\{\mathbf{i}\}(in) \models_x \alpha \wedge \sigma\{\mathbf{i}\} \models_x \neg(\Diamond\alpha)$

$= \exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi\} \cdot \sigma\{\mathbf{i}\}(in) \models_x \alpha \wedge \neg(\sigma\{\mathbf{i}\} \models_x \Diamond\alpha)$

$= \exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathcal{F}ma\{\Phi\} \cdot \neg\,(\neg(\sigma\{\mathbf{i}\}(in) \models_x \alpha) \vee \sigma\{\mathbf{i}\} \models_x \Diamond\alpha)$

$= \neg$ Accessibility ❑

B.2.2 Eventual Output

Goal of Eventual Output: That every process executing at a component will eventually produce an output.

Proof:
Assume that there is a process that will never produce an output. Prove that eventual output is not satisfied.

$$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathit{Fma}\{\Phi\} \cdot \forall \beta : \mathit{Fma}\{\Phi\} \cdot \ \sigma\{\mathbf{i}\} \models_x \alpha \wedge \sigma\{\mathbf{i}\}(\text{out}) \models_x \circ \neg\beta$$

$$= \exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathit{Fma}\{\Phi\} \cdot \forall \beta : \mathit{Fma}\{\Phi\} \cdot \ \sigma\{\mathbf{i}\} \models_x \alpha \wedge \sigma\{\mathbf{i}\}(\text{out}) \models_x \neg \Diamond\beta$$

$$= \exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathit{Fma}\{\Phi\} \cdot \forall \beta : \mathit{Fma}\{\Phi\} \cdot \ \neg (\neg\sigma\{\mathbf{i}\} \models_x \alpha \vee \sigma\{\mathbf{i}\} \models_x \Diamond\beta)$$

$$= \neg \text{ Eventual Output}$$ ❑

B.2.3 Termination

Goal of Termination: That every process executing at a component will terminate.
Proof:
Assume that there is a process that is initiated that does not terminate. Prove that termination is not satisfied.

$$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathit{Fma}\{\Phi\} \cdot \sigma\{\mathbf{i}\} \models_x \alpha \wedge \sigma\{\mathbf{i}\} \models_x \circ \alpha$$

$$= \exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathit{Fma}\{\Phi\} \cdot \neg (\neg\sigma\{\mathbf{i}\} \models_x \alpha \vee \sigma\{\mathbf{i}\} \models_x (\Diamond\neg\alpha))$$

$$= \neg \text{ Termination}$$ ❑

B.2.4 Eventual Progress

Goal of Eventual Progress: That every process executing at a component will eventually lead to another process executing.
Proof:
Assume that there is a process that will never lead to another process. Prove that eventual progress is not satisfied.

$$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \mathit{Fma}\{\Phi\} \cdot \forall \chi : \mathit{Fma}\{\Phi\} \cdot \ \sigma\{\mathbf{i}\} \models_x \Diamond\alpha \wedge \sigma\{\mathbf{i}\} \models_x \circ \neg\chi$$

$$= \neg (\forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall \alpha : \mathit{Fma}\{\Phi\} \cdot \exists \chi : \mathit{Fma}\{\Phi\} \cdot \ (\sigma\{\mathbf{i}\} \models_x \Diamond\alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\chi))$$

$$= \neg \text{ Eventual Progress}$$ ❑

B.2.5 Faithfulness

Goal of Faithfulness: That the component will eventually produce a known, predictable output for every given input.
Proof:
Assume that there is an input that will not cause an known, predicted output. Prove that faithfulness is not satisfied.

$$\exists x : \mathbb{N} \cdot \exists i : \mathbb{N} \cdot \exists \alpha : \Phi \cdot \forall \beta : \Phi \cdot (\alpha;\beta) \wedge \sigma\{\mathbf{i}\}(\text{in}) \models_x \alpha \wedge \sigma\{\mathbf{i}\}(\text{out}) \models_x \circ \neg\beta)$$

$$= \neg \forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall \alpha : \Phi \cdot \exists \beta : \Phi \cdot ((\alpha;\beta) \wedge \sigma\{\mathbf{i}\} \models_x \alpha) \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\beta)$$

$$= \neg \text{ Faithfulness}$$ ❑