Appendix D: Proofs of Progress Meta-Properties

In this appendix we include the details of the proofs of the meta-properties of the basic progress properties. In cases where the proof is straightforward or essentially the same as another proof, we omit the details of the proof, indicating only the major steps of the proof.

D.1 Accessibility

$AC \equiv \forall x : \mathbb{N} \cdot \forall i : \mathbf{N} \cdot \forall \alpha : \Phi \cdot \sigma\{\mathbf{i}\}(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha$

D.1.1 Reflexivity

The behaviour of the component without self-feedback satisfies accessibility.

$$\mathbf{sat}(C_i, AC)$$

$$= \forall x : \mathbb{N} \cdot \forall i : \mathbf{N} \cdot \forall \alpha : \Phi \cdot (\sigma\{\mathbf{i}\}(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha)$$

Assume that all inputs are received from the environment.

$$= \forall x : \mathbb{N} \cdot \forall i : \mathbf{N} \cdot \forall \alpha : \Phi \cdot (\sigma(\mathbb{E};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha)$$

Assume that component **i** has one input channel. Let the connection (environs $\rightarrow in^i$) be replaced by ($out^i \rightarrow in^i$).

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

This implies that the component will continue to satisfy accessbility if the inputs are received from the component via self-feedback. This is based on the assumption that any input received from the environment will be accepted, regardless of the state of the component. Therefore since the state of the component does not affect the accessibility of the component, it does not matter that possible environment events may become not possible on self-feedback.

If the accessibility of the component is dependent on the state of the component, and therefore also on particular inputs and sequences of inputs possible from the environment, we must ensure that these events remain possible when the component has self-feedback. If the connection of the component with self-feedback does not remove all possible environment input channels, then the environment will continue to be able to ensure accessibility at the component. If all environment connections are removed then we must ensure that the output of the component will be able to act in a manner similar to the environment to ensure accessibility.

$$\cong \mathbf{sat}(C_i^R, AC)$$

Assume that component **i** has more than one input channel. Let at least one connection (environs $\rightarrow in^i$) be replaced by ($out^i \rightarrow in^i$).

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

$$\cong \mathbf{sat}(C_i^R, AC)$$

Assume that inputs may be received from the environment or from interconnected components.

$$= \forall x : \mathbb{N} \cdot \forall i : \mathbf{N} \cdot \forall \alpha : \Phi \cdot (\sigma(\mathbf{N}+\mathbb{E};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha)$$

$$\cong \mathbf{sat}(C_i^R, AC)$$

D.1.2 Transitivity

Product Composition

$$\mathbf{sat}(\&^P(C_i,C_j), AC) \wedge \mathbf{sat}(\&^P(C_j,C_k), AC)$$
$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$
$$(\sigma(\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$
$$\wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(\text{Ɇ};k)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

If all behaviours are state-independent, then there are no inputs required to ensure accessibility by returning component to an accessible-state. Thus the transitivity of accessibility follows.

$$= \mathbf{sat}(\&^P(C_i,C_k), AC)$$

If there are state-dependent behaviours, then there are possible inputs for which the component cannot satisfy accessibility unless a related input is received to restore the state of the component to an accessible-state. Thus we must ensure that all such access-restoring inputs are still possible from the environment of the composition of **i** and **k**.

$$\cong \mathbf{sat}(\&^P(C_i,C_k), AC)$$

Cascade Composition

$$\mathbf{sat}(\&^C(C_i,C_j), AC) \wedge \mathbf{sat}(\&^C(C_j,C_k), AC)$$
$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$
$$(\sigma(\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^+\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$
$$\wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(j^+\text{Ɇ};k)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

Let all occurrences of $(out^i \rightarrow in^j)$ be replaced by $(out^i \rightarrow in^k)$, and $(out^j \rightarrow in^k)$ be replaced by $(out^j \rightarrow \text{environs})$.

$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$
$$(\sigma(\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$
$$\wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^+\text{Ɇ};k)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

If all behaviours are state-independent, then there are no inputs required from component **j** to ensure accessibility of component **k**. Thus the transitivity of accessibility follows.

$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$
$$(\sigma(\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^+\text{Ɇ};k)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$
$$\Rightarrow \mathbf{sat}(\&^C(C_i,C_k), AC)$$

If there are state-dependent behaviours at component **k** and the required inputs are received from component **j**, then we must ensure that all such access-restoring inputs are still possible from the environment of the composition of **i** and **k** or as outputs from component **i**.

$$\cong \mathbf{sat}(\&^C(C_i,C_k), AC)$$

Feedback Composition

$$\mathbf{sat}(\&^F(C_i,C_j), AC) \wedge \mathbf{sat}(\&^F(C_j,C_k), AC)$$
$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$
$$(\sigma(j^+\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^+\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$
$$\wedge (\sigma(k^+\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha)$$
$$\wedge (\sigma(j^+\text{Ɇ};k)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

Let all occurrences of ($out^i \rightarrow in^j$) be replaced by ($out^i \rightarrow in^k$), ($out^j \rightarrow in^k$) be replaced by ($out^j \rightarrow$ environs), and ($out^k \rightarrow in^j$) be replaced by ($out^k \rightarrow in^i$), and ($out^j \rightarrow in^i$) be replaced by ($out^j \rightarrow$ environs).

$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$

$$(\sigma(k^{+}\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

$$\wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^{+}\text{Ɇ};k)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

If all behaviours are state-independent, then there are no inputs required from component **j** to ensure accessibility of component **i** or **k**. Thus the transitivity of accessibility follows.

$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$

$$(\sigma(k^{+}\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^{+}\text{Ɇ};k)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

$$= \mathbf{sat}(\mathbf{\&}^F(C_i,C_k), AC)$$

If there are state-dependent behaviours at component **i** or **k** and the required inputs are received from component **j**, then we must ensure that all such access-restoring inputs are still possible from the environment of the composition of **i** and **k** or as outputs from component **k** or **i**.

$$\cong \mathbf{sat}(\mathbf{\&}^F(C_i,C_k), AC)$$

D.1.3 Symmetry

Product Composition

$$\mathbf{sat}(\mathbf{\&}^P(C_i,C_j), AC)$$

$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$

$$(\sigma(\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

$$= \mathbf{sat}(\mathbf{\&}^P(C_j,C_i), AC)$$

Cascade Composition

$$\mathbf{sat}(\mathbf{\&}^C(C_i,C_j), AC)$$

$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$

$$(\sigma(\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^{+}\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

Let all occurrences of ($out^i \rightarrow in^j$) be replaced by ($out^i \rightarrow$ environs), and at least one occurrence of ($out^j \rightarrow$ environs) be replaced by ($out^j \rightarrow in^i$).

$$= \forall x : \mathbb{N} \cdot \forall\forall \alpha : \Phi \cdot$$

$$(\sigma(j^{+}\text{Ɇ};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(\text{Ɇ};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

If all behaviours are state-independent, then there are no inputs required from component **i** or the environment to ensure accessibility of component **j**, or from the environment to ensure the accessibility of component **i**. Thus the transitivity of accessibility follows.

$$\Rightarrow \mathbf{sat}(\mathbf{\&}^C(C_j,C_i), AC)$$

If there are state-dependent behaviours at component **j** and the required inputs are received from component **i**, then we must ensure that all such access-restoring inputs are still possible from the environment of the composition of **i** and **j**. We must also ensure that any access-restoring inputs required by component **i** still be possible inputs, either from the environment or from component **j**.

$$\cong \mathbf{sat}(\mathbf{\&}^C(C_j,C_i), AC)$$

D.1.4 Closure

$$\mathbf{sat}(C_i, AC) \wedge \mathbf{sat}(C_j, AC)$$
$$= \{\forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall \alpha : \Phi \cdot (\sigma\{\mathbf{i}\}(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha)\}$$
$$\wedge \{\forall x : \mathbb{N} \cdot \forall j : \{j\} \cdot \forall \alpha : \Phi \cdot (\sigma\{\mathbf{j}\}(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)\}$$
$$= \forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$
$$(\sigma\{\mathbf{i}\}(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma\{\mathbf{j}\}(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

<u>Product Composition</u>

As long as any access-restoring events from the environment remain possible in the composite environment, then closure follows.

$$= \mathbf{sat}(\boldsymbol{\&}^P(C_i,C_j), AC)$$

<u>Cascade Composition</u>

Let at least one occurrence of ($out^i \rightarrow$ environs) be replaced by ($out^i \rightarrow in^j$).

$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$
$$(\sigma(\mathbb{E};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^+\mathbb{E};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

If all behaviours are state-independent, then there are no inputs required from the environment to ensure accessibility of component **i** or **j**. Thus the closure of accessibility follows.

$$= \mathbf{sat}(\boldsymbol{\&}^C(C_i,C_j), AC)$$

If there are state-dependent behaviours at component **j** and the required inputs are received from the environment, then we must ensure that all such access-restoring inputs are still possible from the environment of the composition of **i** and **j** or from component **i**. We must also ensure that any access-restoring inputs required by component **i** still be possible inputs, within the composite environment.

$$\cong \mathbf{sat}(\boldsymbol{\&}^C(C_j,C_i), AC)$$

<u>Feedback Composition</u>

Let at least one occurrence of ($out^i \rightarrow$ environs) be replaced by ($out^i \rightarrow in^j$) and ($out^j \rightarrow$ environs) be replaced by ($out^j \rightarrow in^i$).

$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$$
$$(\sigma(j^+\mathbb{E};i)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha) \wedge (\sigma(i^+\mathbb{E};j)(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\alpha)$$

If all behaviours are state-independent, then there are no inputs required from the environment to ensure accessibility of component **i** or **j**. Thus the closure of accessibility follows.

$$= \mathbf{sat}(\boldsymbol{\&}^F(C_i,C_j), AC)$$

If there are state-dependent behaviours at either component, and the required inputs are received from the environment, then we must ensure that all such access-restoring inputs are still possible from the environment of the composition of **i** and **j** or from the interconnected component.

$$\cong \mathbf{sat}(\boldsymbol{\&}^F(C_i,C_j), AC)$$

Those behaviours that do not rely on a certain ordering of input events to satisfy accessibility at a component exhibit reflexivity, transitivity, symmetry and closure and will continue to satisfy accessibility on composition with product, cascade and feedback structures.

Those behaviours that do rely on a given order of input events may or may not exhibit reflexivity, transitivity, symmetry and closure. This will depend on the composite structure and the specific details and dependencies of the component in question.

D.2 Termination

$TR \equiv \forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall \alpha : \Phi \cdot \sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond \neg \alpha$

D.2.1 Reflexivity

$\mathbf{sat}(C_i, AC)$

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

Let at least one connection (environs $\rightarrow in^i$) be replaced by ($out^i \rightarrow in^i$).

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

Every process that is "self-terminating," that is, does not require a certain input to be received in order to terminate, will continue to self-terminate under a self-feedback connection.

$= \mathbf{sat}(C_i^R, AC)$

If termination requires a terminating-input, then we cannot guarantee that such an input will be available under self-feedback: termination will depend on required input being sent by another component or environment and will depend on component's ability to receive event (accessibility).

$\cong \mathbf{sat}(C_i^R, AC)$

D.2.2 Transitivity

<u>Product Composition</u>

$\mathbf{sat}(\&^P(C_i,C_j), AC) \wedge \mathbf{sat}(\&^P(C_j,C_k), AC)$

$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot$

$(\sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond \neg \alpha) \wedge (\sigma\{\mathbf{j}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond \neg \alpha)$

$\wedge (\sigma\{\mathbf{j}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond \neg \alpha) \wedge (\sigma\{\mathbf{k}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{k}\} \models_x \Diamond \neg \alpha)$

$\Rightarrow \forall x : \mathbb{N} \cdot \forall i : \{i,k\} \cdot \forall \alpha : \Phi \cdot$

$(\sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond \neg \alpha) \wedge (\sigma\{\mathbf{k}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{k}\} \models_x \Diamond \neg \alpha)$

If every process is "self-terminating," then termination will exhibit transitivity. If there are non-self-terminating processes that rely on events that are possible from the composite environment of components **i** and **k**, then termination will exhibit transitivity.

$= \mathbf{sat}(\&^P(C_i,C_k), AC)$

If termination requires a terminating-input that may not be possible from the composite environment, then we cannot guarantee that such an input will be available under self-feedback.

$\cong \mathbf{sat}(\&^P(C_i,C_k), AC)$

<u>Cascade Composition</u>

$\mathbf{sat}(\&^C(C_i,C_j), AC) \wedge \mathbf{sat}(\&^C(C_j,C_k), AC)$

As with product composition, if processes continue to terminate,

$= \mathbf{sat}(\&^C(C_i,C_k), AC)$

If termination requires a terminating-input, then we cannot guarantee that such an input will be available under the cascade composition of components **i** and **k**.

$\cong \mathbf{sat}(\&^C(C_i,C_k), AC)$

Feedback Composition

$\mathbf{sat}(\mathbf{\&}^F(C_i,C_j), AC) \wedge \mathbf{sat}(\mathbf{\&}^F(C_j,C_k), AC)$

As with product composition, if processes continue to terminate,

$= \mathbf{sat}(\mathbf{\&}^F(C_i,C_k), AC)$

If termination requires a terminating-input, then we cannot guarantee that such an input will be available under the cascade composition of components **i** and **k**.

$\cong \mathbf{sat}(\mathbf{\&}^F(C_i,C_k), AC)$

D.2.3 Symmetry

Product Composition

$\mathbf{sat}(\mathbf{\&}^P(C_i,C_j), AC)$

As with accessibility

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

$= \mathbf{sat}(\mathbf{\&}^P(C_j,C_i), AC)$

Cascade Composition

$\mathbf{sat}(\mathbf{\&}^C(C_i,C_j), AC)$

If every process is "self-terminating," then termination will exhibit transitivity. If there are non-self-terminating processes that rely on events that are possible from the composite environment of components **i** and **j**, then termination will exhibit transitivity.

$= \mathbf{sat}(\mathbf{\&}^C(C_i,C_j), AC)$

If termination requires a terminating-input that may not be possible from the component's meta-environment, then we cannot guarantee that such an input will be available under the symmetric cascade composition.

$\cong \mathbf{sat}(\mathbf{\&}^C(C_j,C_i), AC)$

D.2.4 Closure

$\mathbf{sat}(C_i, AC) \wedge \mathbf{sat}(C_j, AC)$

$= \forall x : \mathbb{N} \cdot \forall i : \mathbf{N} \cdot \forall \alpha : \Phi \cdot$

$(\sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\neg\alpha) \wedge (\sigma\{\mathbf{j}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\neg\alpha)$

Product Composition

If all processes are self-terminating, or all terminating-inputs are possible from composite environment, then product composition is closed.

$= \mathbf{sat}(\mathbf{\&}^P(C_i,C_j), AC)$

Otherwise,

$\cong \mathbf{sat}(\mathbf{\&}^P(C_i,C_j), AC)$

Cascade Composition

If all processes are self-terminating, or all terminating-inputs remain possible events in component'e meta-environment, the cascade composition is closed.

$= \mathbf{sat}(\mathbf{\&}^C(C_i,C_j), AC)$

Otherwise,

$\cong \mathbf{sat}(\mathbf{\&}^C(C_i,C_j), AC)$

Feedback Composition

If all processes are self-terminating, or all terminating-inputs remain posisble events in component's meta-environment, then feedback composition is closed.

$= \mathbf{sat}(\&^F(C_i,C_j), AC)$

Otherwise,

$\cong \mathbf{sat}(\&^F(C_i,C_j), AC)$

D.3 Eventual Progress

$$EP \equiv \forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall \alpha : \Phi \cdot \exists \chi : \Phi \cdot \sigma\{\mathbf{i}\} \models_x \Diamond\alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\chi$$

D.3.1 Reflexivity

$\mathbf{sat}(C_i, AC)$

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

Let at least one occurrence of (environs $\rightarrow in^i$) be replaced by ($out^i \rightarrow in^i$).

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

As with termination, the reflexivity of eventual progress depends on the possible inputs that remain possible or are rendered impossible by the reflexive connection. If an input is possible, then eventual progress will be satisfied as long as all possible inputs remain possible. This requirement ensures that all processes will terminate, allowing eventual progress. If the reflexive composition is such that no inputs to the component are possible, then it is academic whether eventual progress is satisfied as it has no opportunity to be satisfied.

$= \mathbf{sat}(C_i^R, AC)$

D.3.2 Transitivity

Transitivity of eventual progress for product, cascade, and feedback composite system follows using the same arguments as with termination.

Product Composition

$\mathbf{sat}(\&^P(C_i,C_j), AC) \wedge \mathbf{sat}(\&^P(C_j,C_k), AC)$

$\cong \mathbf{sat}(\&^P(C_i,C_k), AC)$

Cascade Composition

$\mathbf{sat}(\&^C(C_i,C_j), AC) \wedge \mathbf{sat}(\&^C(C_j,C_k), AC)$

$\cong \mathbf{sat}(\&^C(C_i,C_k), AC)$

Feedback Composition

$\mathbf{sat}(\&^F(C_i,C_j), AC) \wedge \mathbf{sat}(\&^F(C_j,C_k), AC)$

$\cong \mathbf{sat}(\&^F(C_i,C_k), AC)$

D.3.3 Symmetry

Symmetry of product, cascade, and feedback compositions follows as with termination.

Product Composition

$\mathbf{sat}(\&^P(C_i,C_j), AC)$

$= \mathbf{sat}(\&^P(C_j,C_i), AC)$

Cascade Composition

$\textbf{sat}(\&^C(C_i,C_j), AC)$

$\cong \textbf{sat}(\&^C(C_j,C_i), AC)$

D.3.4 Closure

$\textbf{sat}(C_i, AC) \wedge \textbf{sat}(C_j, AC)$

$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot \exists \chi : \Phi \cdot$

$(\sigma\{\mathbf{i}\} \models_x \Diamond\alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\chi) \wedge (\sigma\{\mathbf{j}\} \models_x \Diamond\alpha \Rightarrow \sigma\{\mathbf{j}\} \models_x \Diamond\chi)$

Closure of eventual progress follows as with termination.

Product Composition

$\cong \textbf{sat}(\&^P(C_i,C_j), AC)$

Cascade Composition

$\cong \textbf{sat}(\&^C(C_i,C_j), AC)$

Feedback Composition

$\cong \textbf{sat}(\&^F(C_i,C_j), AC)$

D.4 Eventual Output

$EO \equiv \forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall \alpha : \Phi \cdot \exists \beta : \Phi \cdot \sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\}(out) \models_x \Diamond\beta$

D.4.1 Reflexivity

$\textbf{sat}(C_i, AC)$

$= \forall x : \mathbb{N} \cdot \forall i : \{i,j\} \cdot \forall \alpha : \Phi \cdot \exists \beta : \Phi \cdot \sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\}(out) \models_x \Diamond\beta$

Reflexivity of eventual output follows as with termination and eventual progress.

$\cong \textbf{sat}(C_i^R, AC)$

D.4.2 Transitivity

Product Composition

$\textbf{sat}(\&^P(C_i,C_j), AC) \wedge \textbf{sat}(\&^P(C_j,C_k), AC)$

$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot \exists \beta : \Phi \cdot$

$(\sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma(i;\mathcal{E})(out) \models_x \Diamond\beta) \wedge (\sigma\{\mathbf{j}\} \models_x \alpha \Rightarrow \sigma(j;\mathcal{E})(out) \models_x \Diamond\beta)$

$\wedge (\sigma\{\mathbf{j}\} \models_x \alpha \Rightarrow \sigma(j;\mathcal{E})(out) \models_x \Diamond\beta) \wedge (\sigma\{\mathbf{k}\} \models_x \alpha \Rightarrow \sigma(k;\mathcal{E})(out) \models_x \Diamond\beta)$

As with termination and eventual progress, if all possible inputs remain possible in composite environment, then transitivity of eventual output follows.

$\Rightarrow \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot \exists \beta : \Phi \cdot$

$(\sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma(i;\mathcal{E})(out) \models_x \Diamond\beta) \wedge (\sigma\{\mathbf{k}\} \models_x \alpha \Rightarrow \sigma(k;\mathcal{E})(out) \models_x \Diamond\beta)$

$\Rightarrow \textbf{sat}(\&^P(C_i,C_k), AC)$

Otherwise,

$\cong \textbf{sat}(\&^P(C_i,C_k), AC)$

Cascade Composition

$$\mathbf{sat}(\&^{C}(C_i,C_j),\ AC) \wedge \mathbf{sat}(\&^{C}(C_j,C_k),\ AC)$$
$$= \forall x : \mathbb{N} \cdot \forall \alpha : \Phi \cdot \exists \beta : \Phi \cdot$$
$$(\sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma(i;j^{+}\text{Ɇ})(out) \models_x \Diamond\beta) \wedge (\sigma\{\mathbf{j}\} \models_x \alpha \Rightarrow \sigma(j;\text{Ɇ})(out) \models_x \Diamond\beta)$$
$$\wedge (\sigma\{\mathbf{j}\} \models_x \alpha \Rightarrow \sigma(j;k^{+}\text{Ɇ})(out) \models_x \Diamond\beta) \wedge (\sigma\{\mathbf{k}\} \models_x \alpha \Rightarrow \sigma(k;\text{Ɇ})(out) \models_x \Diamond\beta)$$

As with termination and eventual progress,

$$\cong \mathbf{sat}(\&^{C}(C_i,C_k),\ AC)$$

Feedback Composition

$$\mathbf{sat}(\&^{F}(C_i,C_j),\ AC) \wedge \mathbf{sat}(\&^{F}(C_j,C_k),\ AC)$$

As with termination and eventual progress,

$$\cong \mathbf{sat}(\&^{F}(C_i,C_k),\ AC)$$

D.4.3 Symmetry

Product Composition

$$\mathbf{sat}(\&^{P}(C_i,C_j),\ AC)$$

As with termination and eventual progress,

$$= \mathbf{sat}(\&^{P}(C_j,C_i),\ AC)$$

Cascade Composition

$$\mathbf{sat}(\&^{C}(C_i,C_j),\ AC)$$

As with termination and eventual progress,

$$\cong \mathbf{sat}(\&^{C}(C_j,C_i),\ AC)$$

D.4.4 Closure

$$\mathbf{sat}(C_i,\ AC) \wedge \mathbf{sat}(C_j,\ AC)$$
$$= \forall x : \mathbb{N} \cdot \forall i : \{i\} \cdot \forall j : \{j\} \cdot \forall \alpha : \Phi \cdot \exists \beta : \Phi \cdot$$
$$(\sigma\{\mathbf{i}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\}(out) \models_x \Diamond\beta) \wedge (\sigma\{\mathbf{j}\} \models_x \alpha \Rightarrow \sigma\{\mathbf{j}\}(out) \models_x \Diamond\beta)$$

Product Composition

As with termination and eventual progress,

$$\cong \mathbf{sat}(\&^{P}(C_i,C_j),\ AC)$$

Cascade Composition

As with termination and eventual progress,

$$\cong \mathbf{sat}(\&^{C}(C_i,C_j),\ AC)$$

Feedback Composition

As with termination and eventual progress,

$$\cong \mathbf{sat}(\&^{F}(C_i,C_j),\ AC)$$

D.5 Faithfulness

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

Given that $i(\alpha;\beta)$ and $\sigma\{\mathbf{i}\}(in) \models_x \alpha$

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

By Accessibility

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

By Eventual Output

$= \forall x{:}\mathbb{N} \cdot \forall i{:}\mathbf{N} \cdot \forall \alpha{:}\Phi(i(in)) \cdot \exists \beta{:}\Phi(i(out)) \cdot$

$i(\alpha;\beta) \wedge \sigma\{\mathbf{i}\}(in) \models_x \alpha \Rightarrow \sigma\{\mathbf{i}\} \models_x \Diamond\alpha \Rightarrow \sigma\{\mathbf{i}\}(out) \models_x \Diamond\Diamond\beta$

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

= Faithfulness

By definition, faithfulness will also imply eventual progress and termination for those behaviours that are input-state independent. If these properties are not satisfied by all possible behaviours, we cannot guarantee faithfulness because we cannot guarantee that every input α to a component **i** such that $i(\alpha;\beta)$ will eventually produce an output β. Thus if accessibility, eventual output, eventual progress, and termination exhibit a given meta-property, then so will faithfulness.

D.5.1 Reflexivity

$\mathbf{sat}(C_i, FT)$

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

If AC, EO, EP and TR exhibit reflexivity

$\Rightarrow \mathbf{sat}(C_i^R, FT)$

D.5.2 Transitivity

If AC, EO, EP and TR exhibit transitivity for a given composite structure then so will faithfulness.

D.5.3 Symmetry

If AC, EO, EP and TR exhibit symmetry for a given composite structure then so will faithfulness.

D.5.4 Closure

If AC, EO, EP and TR exhibit closure for a given composite structure then so will faithfulness.

D.6 Provenance

$$PV \equiv \forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall A : \mathcal{F}ma\{\Phi\} \cdot \sigma\{\mathbf{i}\}(in) \models_x \Diamond A \Rightarrow \sigma(\mathbb{N}^+\mathbb{E};i)(out) \models_x A$$

D.6.1 Reflexivity

$\mathbf{sat}(C_i, PV)$

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

If connections are correct, then reflexivity of provenance follows,

$= \mathbf{sat}(C_i^R, PV)$

D.6.2 Transitivity

If connections are correct, then transivity of provenance for product, cascade and feedback composition follows because the replacement of one known source by another known source does not violate the provenance of the events in the system.

D.6.3 Symmetry

If connections are correct, then symmetry of provenance for product, cascade and feedback composition follows because the replacement of one known source by another known source does not violate the provenance of the events in the system.

D.6.4 Closure

If connections are correct, then closure of provenance for product, cascade and feedback composition follows because the replacement of one known source (the environment) by another known source (a known component) does not violate the provenance of the events in the system.

D.7 Utility

$$UT \equiv \forall x : \mathbb{N} \cdot \forall i : \mathbb{N} \cdot \forall A : \mathcal{F}ma\{\Phi\} \cdot \sigma(\mathbb{N}^+\mathbb{E};i)(out) \models_x A \Rightarrow \sigma\{\mathbf{i}\}(in) \models_x \Diamond A$$

D.7.1 Reflexivity

$\mathbf{sat}(C_i, UT)$

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

If connections are correct, then reflexivity of utility follows,

$= \mathbf{sat}(C_i^R, UT)$

D.7.2 Transitivity

If connections are correct, then transitivity of utility for product, cascade and feedback composition follows because the replacement of one known source by another known source does not violate the utility the system.

D.7.3 Symmetry

If connections are correct, then symmetry of utility for product, cascade and feedback composition follows because the replacement of one known source by another known source does not violate the utility of the system.

D.7.4 Closure

If connections are correct, then closure of utility for product, cascade and feedback composition follows because the replacement of one known source by another known source does not violate the utility of the system.