Accellera Technical Report 2004.01

Notes on the Semantics of Local Variables in Accellera SystemVerilog 3.1 Concurrent Assertions

John Havlicek [1], Kurt Shultz [1], Roy Armoni [2], Surrendra Dudani [3], and Eduard Cerny [3]

[1] Freescale Semiconductor, Inc.
[2] Intel Corp.
[3] Synopsys, Inc.

May 2004

STATEMENT OF USE
OF ACCELLERA STANDARDS

Accellera Standards documents are developed within Accellera and the Technical Committees of Accellera Organization, Inc. Accellera develops its standards through a consensus development process, approved by its members and board of directors, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of Accellera and serve without compensation. While Accellera administers the process and establishes rules to promote fairness in the consensus development process, Accellera does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an Accellera Standard is wholly voluntary. Accellera disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other Accellera Standard document.

Accellera does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or suitability for a specific purpose, or that the use of the material contained herein is free from patent infringement. Accellera Standards documents are supplied "**AS IS**."

The existence of an Accellera Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of an Accellera Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change due to developments in the state of the art and comments received from users of the standard. Every Accellera Standard is subjected to review periodically for revision and update. Users are cautioned to check to determine that they have the latest edition of any Accellera Standard.

In publishing and making this document available, Accellera is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is Accellera undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other Accellera Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of Accellera, Accellera will initiate action to prepare appropriate responses. Since Accellera Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, Accellera and the members of its Technical Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of Accellera Standards are welcome from any interested party, regardless of membership affiliation with Accellera. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Accellera Organization
1370 Trancas Street #163
Napa, CA 94558
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. Accellera shall not be responsible for identifying patents for which a license may be required by an Accellera standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Accellera is the sole entity that may authorize the use of Accellera-owned certification marks and/or trademarks to indicate compliance with the materials set forth herein.

Authorization to photocopy portions of any individual standard for internal or personal use must be granted by Accellera Organization, Inc., provided that permission is obtained from and any required fee is paid to Accellera. To arrange for authorization please contact Lynn Horobin, Accellera, 1370 Trancas Street #163, Napa, CA 94558, phone (707) 251-9977, e-mail lynn@accellera.org. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained from Accellera.

Abstract A discussion is given of the intuition behind the definitions of the semantics of local variables published in Annex G of the Accellera SystemVerilog 3.1 Language Reference Manual. A number of desirable technical consequences of the definitions are proved, including certain algebraic properties of sequence-building operators and all the claims made in Annex G. An alternative approach to the auxiliary functions *block* and *flow* is given, resulting in an equivalent definition of *flow* and a more intuitive relationship between the three auxiliary functions *sample*, *block*, and *flow*.

Acknowledgments: The local variables in SystemVerilog 3.1 Assertions are an extension and adaptation to the SystemVerilog context of the local variables from the CBV language, donated to Accellera by Motorola, Inc. CBV local variables were invented within Motorola Semiconductor Israel, Ltd. by a team led by Hillel Miller and including Efim Gukovsky, Nissan Levi, and Navit Fedida. Some of the work presented in these notes was done in the Semantics Subcommittee of the Accellera SystemVerilog 3.1 Assertions Committee in Spring, 2003. The members of the Semantics Subcommittee were Roy Armoni (Intel Corp.), Surrendra Dudani (Synopsys, Inc.), Cindy Eisner (IBM Haifa Research Lab), Faisal Haque (Cisco Systems, Inc.), John Havlicek (Motorola, Inc.), and Stephen Meier (Synopsys, Inc.).

Contents

1 Introduction

Accellera SystemVerilog 3.1 Assertions, abbreviated herein as *SVA 3.1*, constitute a major language feature of the Accellera SystemVerilog 3.1 standard. They are described in Section 17 and Annex G of [SV]. The assertions of SVA 3.1 are divided broadly into two categories: *immediate assertions* and *concurrent assertions*. The immediate assertions are analogous to assertions in programming languages like C. Because they are non-temporal and do not involve local variables, immediate assertions are not considered further in these notes. The concurrent assertions enable precise definition of a rich set of temporal properties for designs. The formal semantics of concurrent assertions is defined in Annex G of [SV].

The purpose of these notes is to record the intuition behind the formal definitions pertaining to local variables and to give evidence of the quality of (i.e., "sanity check") the definitions by proving various desirable technical consequences. Among these are results showing that certain algebraic properties of sequence-building operators are not disturbed by the addition of local variables. In particular, we show that "`intersect`", "`and`", and "`or`" are all associative and that "`intersect`" and "`and`" distribute over "`or`". All the claims made in Annex G about consequences of the definitions are also proved. The technical consequences are not mathematically deep. They are, for the most part, intuitive but not entirely obvious. As a result, these notes should be accessible to a reader who is familiar with elementary set theory and functions and who is comfortable reading Annex G of [SV].

The rest of the notes are organized as follows.

- Section 2 gives an intuitive discussion of the semantics of local variables. For simplicity, this discussion avoids derived operators and the technical details involved with the negation and abort operators. The focus is on sequences since the scoping of local variables in the presence of the branching sequence operators (e.g., "`intersect`", "`or`") is the least transparent part of the semantics.
- Section 3 covers notations, assumptions, and preliminary observations for the technical presentation of the consequences of the formal definitions.
- Section 4 proves some basic relationships between the auxiliary functions *sample*, *block*, and *flow*.
- Section 5 shows that the definition of tight satisfaction with local variables is consistent with the auxiliary functions and that the value of a local variable persists unless the variable is reassigned.
- Section 6 treats algebraic properties of sequence operators in the presence of local variables. The algebraic properties studied are commutativity, associativity, distributivity, existence of idempotent elements, and existence of identity elements.
- Section 7 presents an alternative approach to the auxiliary functions *block* and *flow*. The new definition of *flow* is equivalent to the one from Annex G, but *block* is changed in a way that yields a more intuitive relationship between *sample*, *block*, and *flow*.

For convenience, the definitions from Annex G of *sample*, *block*, and *flow* and of tight satisfaction with local variables appear in an Appendix.

2 Intuitive Discussion of the Semantics of Local Variables

The semantics of SVA 3.1 concurrent assertions is "linear" in the sense of Linear Temporal Logic (LTL). Annex G of [SV] defines a satisfaction relation of concurrent assertions by finite and infinite words over the extended alphabet $\Sigma = 2^{\mathbf{P}} \cup \{\top, \bot\}$, where $\mathbf{P}$ is the set of atomic propositions. For this intuitive discussion, we limit the alphabet to $2^{\mathbf{P}}$. The body of a concurrent assertion is a temporal *property*. A word satisfies the assertion iff one or more particular suffixes of the word satisfy the property. The particular suffixes depend on the form of the assertion ("`initial`" vs. "`always`"), the way it is clocked, and any enabling condition inferred from the context of the assertion.

Explicit or inferred clock events govern all parts of a property in a concurrent assertion of SVA 3.1. However, Annex G of [SV] defines the semantics of clocked properties indirectly in terms of a semantics for *unclocked* properties. Intuitively, time in an unclocked property proceeds at the granularity of the letters in the words for which the satisfaction relation is defined. Under this approach, clocked properties are transformed into unclocked properties via rewrite rules that represent the clocking information in the more basic terms of the finest granularity of time. The unclocked semantics is then applied to the transformed properties. In these notes, it is assumed that all properties have been transformed into unclocked properties in this way.

2.1 Semantics of sequences without local variables

Properties are built up from *sequences*.[1] In the absence of local variables, a sequence s can be thought of as a regular expression that denotes a set $\mathcal{L}(s)$ of finite words over $2^{\mathbf{P}}$. $\mathcal{L}(s)$ is the *language* of s over $2^{\mathbf{P}}$. The elements of $\mathcal{L}(s)$ are said to *tightly satisfy* s. The following partial grammar shows some of the basic sequence forms, exclusive of local variables:

$s ::= b$	[boolean expression]
$\mid$ `(`s`)`	[parenthesis]
$\mid$ `(`s `##1` s`)`	[concatenation]
$\mid$ `(`s `##0` s`)`	[fusion]
$\mid$ `(`s `or` s`)`	[union]
$\mid$ `(`s `intersect` s`)`	[intersection]
$\mid$ s`[*0]`	[empty]
$\mid$ s`[*1:$]`	[positive repetition]

Here b represents a boolean expression. $\mathcal{L}(b)$ is the set of words of length one whose single letter satisfies the boolean expression b. For the other forms, let xy denote the concatenation of words x and y, and let $|x|$ denote the length of word x. Then

- $\mathcal{L}(\texttt{(}s\texttt{)}) = \mathcal{L}(s)$.
- $\mathcal{L}(\texttt{(}s \texttt{ ##1 } t\texttt{)}) = \{xy \colon x \in \mathcal{L}(s) \text{ and } y \in \mathcal{L}(t)\}$.
- $\mathcal{L}(\texttt{(}s \texttt{ ##0 } t\texttt{)}) = \{xyz \colon |y| = 1 \text{ and } xy \in \mathcal{L}(s) \text{ and } yz \in \mathcal{L}(t)\}$.

---

[1] Sequences are also understood to be unclocked in these notes.

- $\mathcal{L}(\texttt{(}s\ \texttt{or}\ t\texttt{)}) = \mathcal{L}(s) \cup \mathcal{L}(t)$.
- $\mathcal{L}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) = \mathcal{L}(s) \cap \mathcal{L}(t)$.
- $\mathcal{L}(s\texttt{[*0]}) = \{x \colon |x| = 0\}$.
- $\mathcal{L}(s\texttt{[*1:\$]}) = \{x_1 \cdots x_j \colon j \geq 1 \text{ and } x_i \in \mathcal{L}(s) \text{ for all } i \text{ such that } 1 \leq i \leq j\}$.

2.2 Semantics of basic properties without local variables

The two fundamental property forms in SVA 3.1 are shown in the following partial grammar:

$$\begin{array}{rll} p ::= & s & \text{[sequence]} \\ \mid & \texttt{(}s\ \texttt{|->}\ s\texttt{)} & \text{[overlapping implication]} \end{array}$$

The satisfaction relation for properties is denoted "$\models$".

For non-empty words w over $2^{\mathbf{P}}$ and sequences s, t without local variable assignments, the satisfaction relation for the two fundamental property forms is defined by

- $w \models s$ iff there exist x, y such that $w = xy$ and $|x| > 0$ and $x \in \mathcal{L}(s)$.
- $w \models \texttt{(}s\ \texttt{|->}\ t\texttt{)}$ iff for all x, y, z such that $w = xyz$ and $|y| = 1$ and $xy \in \mathcal{L}(s)$, $yz \models t$.

SVA 3.1 has other property-building constructs (negation operator, abort operator), but they are not discussed in these notes.

2.3 Local variables

Local variables are a powerful feature of SVA 3.1 that allow the sampling and manipulation of data in a property without requiring the property writer to define auxiliary state machines (i.e., satellite logic). In SVA 3.1, the syntax

$$s ::= \texttt{(}b\texttt{,}\ v\ \texttt{=}\ e\texttt{)} \qquad \text{[boolean expression with local variable]}$$

attaches to boolean expression b the assignment of the value of expression e to local variable v. The boolean expression b is tested at a particular point in the evaluation of a property, and this test occurs in the same way regardless of the attached local variable assignment. If b holds (i.e., evaluates to true) when tested, then the attached assignment of the value of e to v occurs at that time. Otherwise, no assignment to v is made.

The value assigned to v on successful test of b can be accessed later in the property by referencing v. In this sense, v stores the value of e sampled at the test of b, and the scope of v "flows" forward through the property. For example, the property

$$\texttt{(((}b\texttt{,}\ v\ \texttt{=}\ e\texttt{) \#\#1 1) |->}\ f\ \texttt{==}\ v\ \texttt{)} \tag{1}$$

says intuitively, "if b holds in the first timestep and there is a second timestep, then the value of f in the second timestep must equal the value of e in the first timestep." A more interesting example is

$$\texttt{(((}b\texttt{,}\ v\ \texttt{=}\ e\texttt{) \#\#1 !}c\texttt{[*0:\$] \#\#1}\ c\texttt{) |->}\ f\ \texttt{==}\ v\ \texttt{)} \tag{2}$$

Here, "`[*0:$]`" is a derived sequence operator for zero or more repetitions, defined by

$$s\texttt{[*0:\$]} \equiv \texttt{(}s\texttt{[*0] or }s\texttt{[*1:\$])}$$

Therefore, the intuitive meaning of (2) is "if b holds in the first timestep and if there is a timestep after the first at which c holds, then the value of f in the earliest such timestep must equal the value of e in the first timestep."

Local variables can also be used to do arithmetic manipulations of data in a property. For example,

$$\texttt{(((}b\texttt{, }v\texttt{ = 0) \#\#1 (!}c\texttt{, }v\texttt{ = }v\texttt{ + 1)[*0:\$] \#\#1 }c\texttt{) |-> }f\texttt{ == }v\texttt{)} \tag{3}$$

has the intuitive meaning "if b holds in the first timestep and if there is a timestep after the first at which c holds, then the value of f in the earliest such timestep must equal the number of timesteps strictly between the earliest such timestep and the first timestep, counted modulo the overflow value of v."

It is easy to eliminate the local variable from (1) by adding an auxiliary state machine that delays the value of e by one timestep. Eliminating the local variable from (2) is not as simple. If property (2) is asserted "`always`", then every timestep at which b holds will trigger a check for f `==` v at the next future timestep at which c holds. If b and `!`c are not disjoint, then there could be multiple timesteps at which b holds before a timestep at which c holds. This kind of situation complicates the construction of the auxiliary state machines. Under a sequential protocol it may be reasonable to expect b not to hold a second time until after c holds. This expectation may not be reasonable under an interleaved protocol, for which a property like the following tagged variant of (2) may be appropriate:

```
(
   (
      (b, v = e, v' = intag)
      ##1
      !(c && (outtag == v'))[*0:$]
      ##1
      (c && (outtag == v'))
   )
   |->
   f == v
)
```

Here, "$\texttt{(}b\texttt{, }v\texttt{ = }e\texttt{, }v'\texttt{ = intag)}$" attaches both assignments, "$v\texttt{ = }e$" and "$v'\texttt{ = intag}$", to the boolean expression b. Creating an auxiliary state machine to take the place of the local variables in this example requires allocating and managing storage for as many tags as can be active simultaneously.

2.4 Semantics of sequences with local variables

In the presence of local variables, the semantics of a sequence is no longer adequately captured as a language of words as described in Subsection 2.1. This is because the values stored in the local variables can influence the expressions that determine whether a word tightly satisfies the sequence. For example, for tight satisfaction of the sequence

```
(
   (b, v = e, v' = intag)
   ##1
   !(c && (outtag == v'))[*0:$]
   ##1
   (c && (outtag == v'))
)
```

from above, it is essential that the value of `intag` sampled when b holds equal the value of `outtag` when c holds.

In order to capture the way local variables influence expression evaluation in tight satisfaction, Annex G of [SV] defines tight satisfaction of sequences with local variables as a four-way relation between finite words over Σ, pairs of *local variable contexts*, and sequences.[2] A local variable context is simply a function that has a set of local variable names as domain and that records which local variables have been assigned (i.e., can be referenced) and what sampled values they hold. The four-way relation is written

$$w, L_0, L_1 \mathrel{|\!\!\equiv} s$$

where w is a finite word over Σ, L_0 is the *incoming* local variable context, L_1 is the *outgoing* local variable context, and s is a sequence. Intuitively, $w, L_0, L_1 \mathrel{|\!\!\equiv} s$ means that if the local variable context L_0 is provided to start with, then the word w tightly satisfies s in a way that transforms L_0 into L_1.

As a simple example, let $w^0, w^1 \in 2^{\mathbf{P}}$ be letters such that b holds on w^0 and c holds on w^1, and let $w = w^0w^1$. If s is the sequence

```
( ((b, v = e) ##1 c) )
```

then

$$w, \{\}, \{(v, e[w^0])\} \mathrel{|\!\!\equiv} s$$

where $e[w^0]$ is a notation for the value obtained from expression e when sampled at w^0. In this example, the incoming local variable context is empty (i.e., no local variables have been assigned), and the outgoing local variable context is $L_1 = \{(v, e[w^0])\}$, using set-of-pairs notation for the function. In other words, L_1 has a single local variable, v, in its domain, and $L_1(v) = e[w^0]$.

More generally, for any local variable context L_0,

$$w, L_0, \{(v, e[L_0, w^0])\} \cup L_0|_{\mathrm{dom}(L_0)-\{v\}} \mathrel{|\!\!\equiv} s$$

Here, $\mathrm{dom}(L_0)$ denotes the domain of L_0, and $e[L_0, w^0]$ denotes the value obtained from e by first substituting local variable values from L_0 and then using w^0 to complete the evaluation on any remaining variables. This order of substitution ensures that local variable names in $\mathrm{dom}(L_0)$ take precedence over other variable names in case there is a collision in the evaluation of $e[L_0, w^0]$. The notation $L_0|_{\mathrm{dom}(L_0)-\{v\}}$ means the function obtained from L_0 by restricting its domain to $\mathrm{dom}(L_0) - \{v\}$. Therefore, $L_0|_{\mathrm{dom}(L_0)-\{v\}} = L_0 - \{(v, L_0(v))\}$ if $v \in \mathrm{dom}(L_0)$, and $L_0|_{\mathrm{dom}(L_0)-\{v\}} = L_0$ if $v \notin \mathrm{dom}(L_0)$. The effect is that if v already has a value in

---

[2] For reference, the definition of tight satisfaction with local variables appears in the Appendix to these notes.

L_0, then that value gets replaced by $e[L_0, w^0]$ in the outgoing context. Otherwise, the outgoing context is obtained from L_0 by adding v to the domain and giving it the value $e[L_0, w^0]$.

From this discussion, it should be intuitive that if s does not assign any local variables and $w, L_0, L_1 \models s$, then $L_1 = L_0$. This fact is not obvious from the definitions in Annex G of [SV] and is proved as Theorem 5.5 below. It should also be intuitive that if s does not involve local variables, then the relation $w \in \mathcal{L}(s)$ is equivalent to $w, \{\}, \{\} \models s$.

2.5 Subtleties of scoping of local variables

The examples shown so far have used only "linear" operators (concatenation, fusion, repetition, implication), and the forward flow of the local variables through the properties has been intuitively clear. Consider, however, the following examples:

$$\texttt{(((}b\texttt{, } v \texttt{ = } e\texttt{) or } c\texttt{) \#\#1 } f \texttt{ == } v \texttt{)} \tag{4}$$

$$\texttt{(((}b\texttt{, } v \texttt{ = } e\texttt{) intersect (}c\texttt{, } v \texttt{ = } e'\texttt{)) \#\#1 } f \texttt{ == } v \texttt{)} \tag{5}$$

Both of these examples have "unsafe" references to v in the expression "f `==` v". In (4) the reference is unsafe because the "`or`" can be matched without assignment to v if c holds at the first timestep. In (5) the reference is unsafe because the values assigned to v in the two branches of the "`intersect`" may be inconsistent. Thus, the branching and joining natures of "`or`" and "`intersect`" bring an additional complexity to the scoping of local variables that does not exist with the linear operators.

Annex G of [SV] gives the precise definitions for scoping of local variables. For sequences, this is done with three recursive functions, *sample*, *block*, and *flow*.[3] These definitions have several goals:

1. To disallow flow of a local variable out from an "`or`" if the local variable is not guaranteed to be assigned in both branches of the "`or`" (as in (4)) and the local variable was not assigned prior to the "`or`".

2. To disallow flow of a local variable out from an "`intersect`" if the local variable can be assigned in both branches of the "`intersect`" (as in (5)).

3. To require only simple syntactic analysis of the sequences.

4. To allow forward flow of a local variable unless in conflict with 1, 2, or 3.

5. Not to disturb familiar algebraic properties of the regular expression operators unless in conflict with 1, 2, 3, or 4.

The function *sample* takes a sequence as input and outputs the set of local variable names that are assigned in the sequence. The function *block* takes a sequence as input and outputs the set of local variable names whose scope must not be allowed to extend beyond the sequence in order to achieve 1, 2, and 3. The function *flow* takes a set X of local variable names and a sequence as inputs and outputs the set

---

[3] For reference, the definitions of *sample*, *block*, and *flow* appear in the Appendix to these notes.

of local variable names whose scopes extend (i.e., “flow”) out of the sequence given that the local variable names in X flow into the sequence.

Substituting the problematic “`intersect`” subsequence from (5) into the definitions from Annex G of [SV] yields

$$\begin{aligned} \mathit{sample}(\texttt{((}b\texttt{, } v \texttt{ = } e\texttt{) intersect (}c\texttt{, } v \texttt{ = } e'\texttt{)))} &= \{v\} \\ \mathit{block}(\texttt{((}b\texttt{, } v \texttt{ = } e\texttt{) intersect (}c\texttt{, } v \texttt{ = } e'\texttt{)))} &= \{v\} \\ \mathit{flow}(X, \texttt{((}b\texttt{, } v \texttt{ = } e\texttt{) intersect (}c\texttt{, } v \texttt{ = } e'\texttt{)))} &= X - \{v\} \end{aligned}$$

for any set X of local variable names. This illustrates how the definitions prevent the outward flow in the case that is forbidden by goal 2. Doing similarly with the “`or`” subsequence from (4) yields

$$\begin{aligned} \mathit{sample}(\texttt{((}b\texttt{, } v \texttt{ = } e\texttt{) or } c\texttt{))} &= \{v\} \\ \mathit{block}(\texttt{((}b\texttt{, } v \texttt{ = } e\texttt{) or } c\texttt{))} &= \{\} \\ \mathit{flow}(X, \texttt{((}b\texttt{, } v \texttt{ = } e\texttt{) or } c\texttt{))} &= X \end{aligned}$$

for any set X of local variable names. Note that *block* of this subsequence is empty, which might seem to fail to meet goal 1. However, according to the last of these equalities, if v flows into the sequence “$\texttt{((}b\texttt{, } v \texttt{ = } e\texttt{) or } c\texttt{)}$” (i.e., if $v \in X$), then v also flows out. The inward flow of v means that v was assigned a value prior to the “`or`”. The prior value of v is preserved in the branch that matches c, but v gets reassigned in the branch that matches b. The last equality also says that if $v \notin X$, then $v \notin \mathit{flow}(X, \texttt{((}b\texttt{, } v \texttt{ = } e\texttt{) or } c\texttt{))}$.

Of course, the scoping and flow of local variables is intimately connected with the domains of the incoming and outgoing local variable contexts in the tight satisfaction relation. This is no coincidence, as the definition of tight satisfaction in the cases of “`or`” and “`intersect`” relies on the scoping definitions. Intuitively, if

$$w, L_0, L_1 \models\!\equiv s$$

then it should be that

$$\mathrm{dom}(L_1) = \mathit{flow}(\mathrm{dom}(L_0), s)$$

This result is not obvious from the definitions in Annex G of [SV] and is proved as Theorem 5.4 below.

2.6 Semantics of basic properties with local variables

In the presence of local variables, the semantics of properties is defined in Annex G of [SV] as a three-way satisfaction relation among non-empty words, local variable contexts, and properties. The relation is written

$$w, L \models p$$

where w is a non-empty finite or infinite word over Σ, L is the *incoming* local variable context, and p is a property. Intuitively, $w, L \models p$ means that if the local variable context L is provided to start with, then the word w satisfies p.

For non-empty words w over $2^{\mathbf{P}}$ and sequences s, t, the satisfaction relation for the two fundamental property forms is defined by

- $w, L \models s$ iff there exist x, y, L' such that $w = xy$ and $|x| > 0$ and $x, L, L' \models\!\!\!\equiv s$.
- $w, L \models$ `(`s `|->` t`)` iff for all x, y, z, L' such that $w = xyz$ and $|y| = 1$ and $xy, L, L' \models\!\!\!\equiv s$, $yz, L' \models t$.

Note that in the second of these rules, the outgoing local variable context L' from the tight satisfaction $xy, L, L' \models\!\!\!\equiv s$ serves as the incoming local variable context for the property satisfaction $yz, L' \models t$. As a result, local variables intuitively flow across the implication operator "`|->`".

3 Technical Preliminaries and Notation

This section sets notations and makes assumptions and preliminary observations to be used in the remainder of these notes.

The following set notations are used.

- "$\cup$" denotes set union.
- "$\cap$" denotes set intersection.
- "$-$" denotes set difference.
- "$\{\}$" denotes the empty set.
- "$\in$" denotes the set membership relation.

Elementary set algebra with set differences is used in many of the proofs in these notes. Let A, B, M, N be sets. The following elementary set theory facts are assumed to be familiar.

- $(A \cup B) - M = ((A - M) \cup B) - M = (A - M) \cup (B - M)$.
- $(A \cap B) - M = (A - M) \cap B = (A - M) \cap (B - M)$.
- $A - (M \cup N) = (A - M) \cap (A - N)$.
- $A - (M \cap N) = (A - M) \cup (A - N)$.
- $(A - M) - N = A - (M \cup N)$.
- $(A - M) \cup B = (A \cup B) - (M - B)$.
- $(A - M) \cap (B - N) = (A \cap B) - (M \cup N)$.
- $A - (N - B) = (A \cap B) \cup (A - N)$.
- $(A - M) - (N - B) = ((A \cap B) - M) \cup (A - (M \cup N))$.
- $(A - M) - (A - N) = (A - M) \cap N = (A \cap N) - M$.

The term "sequence" means an unclocked sequence as defined in the abstract syntax of Annex G, Subsection G.2.1, p. 344 of [SV]. The following sequence notations are used.

- e denotes an expression.
- v, v' denote local variable names when appearing in a local variable assignment sequence, such as "`(1,` v `=` e`)`".
- r, s, t denote sequences.
- b denotes a boolean expression.

The ordinary alphabet for the semantics of SystemVerilog 3.1 Assertions is denoted $2^{\mathbf{P}}$. Let $\mathcal{B}$ denote the set of boolean expressions. There is understood to be a relation of boolean satisfaction $\Vdash \subseteq 2^{\mathbf{P}} \times \mathcal{B}$. The notation "$\ell \Vdash b$" (respectively, "$\ell \nVdash b$") means that $(\ell, b) \in \Vdash$ (respectively, $(\ell, b) \notin \Vdash$). The "true" (respectively, "false") element of $\mathcal{B}$ is denoted "`1`" (respectively, "`0`"), and it is understood that

$$\ell \Vdash \texttt{1} \ \text{ and } \ \ell \nVdash \texttt{0}$$

for all $\ell \in 2^{\mathbf{P}}$. The extended alphabet for the semantics of SystemVerilog 3.1 Assertions is

$$\Sigma = 2^{\mathbf{P}} \cup \{\top, \bot\}$$

The relation $\Vdash$ is extended to letters in Σ by defining

$$\top \Vdash b \ \text{ and } \ \bot \nVdash b$$

for all $b \in \mathcal{B}$. Unlike letters of the ordinary alphabet, $\top \Vdash \texttt{0}$ and $\bot \nVdash \texttt{1}$.

A *word* over Σ is a sequence of letters from Σ. The number of letters in word w is called the *length* of w and is denoted $|w|$. The letters of a word are assumed to be indexed consecutively beginning at zero. If $|w| = 0$, then w has no letters and is said to be *empty*. If $|w| > 0$, then the first letter of w is denoted w^0; if $|w| > 1$, then the second letter of w is denoted w^1; and so forth.

A *local variable context* is a function that assigns values to local variable names. If L is a local variable context, then $\mathrm{dom}(L)$ denotes the set of local variable names that are in the domain of L. If D is a set of local variable names, then $L|_D$ denotes the local variable context that is obtained from L by restricting its domain to be $\mathrm{dom}(L) \cap D$. If $D \subseteq \mathrm{dom}(L)$, then $\mathrm{dom}(L|_D) = D$.

The semantics of sequences is defined via a relation of tight satisfaction by finite (possibly empty) words. Tight satisfaction is a four-way relation defining when a finite word w together with an input local variable context L_0 satisfies a sequence r and yields an output local variable context L_1. This relation is denoted

$$w, L_0, L_1 \models\!\equiv r$$

The definition of tight satisfaction is given in Annex G, Subsection G.3.4, p. 350 of [SV]. This definition depends on three auxiliary functions, *sample*, *block*, and *flow*, which are defined in Annex G, Subsection G.3.3.3, pp. 348-349 of [SV]. For reference, the definitions of tight satsifaction and of the auxiliary functions appear in an appendix to these notes with notation adapted to the conventions presented here.

The following notations are used throughout these notes.

- u, v, w, x, y, z and primed and/or subscripted versions of these denote finite (possibly empty) words over Σ. [Exception: v, v' are also used to denote local variable names when appearing in a local variable assignment sequence as described previously.]

- X, Y denote sets of local variable names.

- L and primed and/or subscripted versions of L denote local variable contexts.

- S_r is shorthand for $\mathit{sample}(r)$.
- B_r is shorthand for $\mathit{block}(r)$.
- F_r is shorthand for $\mathit{flow}(\{\}, r)$.
- X/r is shorthand for $\mathit{flow}(X, r)$.

These notes rely on the following correction of an erratum to the Accellera SystemVerilog 3.1 Assertions definitions that pertain to well-definedness of the formal semantics of assertions over the extended alphabet $\Sigma = 2^{\mathbf{P}} \cup \{\top, \bot\}$: it is understood that for any expression e and any local variable context L, $e[L, \top]$ and $e[L, \bot]$ are arbitrary constants of the type of e.

4 Basic Properties of *sample*, *block*, and *flow*

The following theorem is claimed in Annex G, Subsection G.3.3.3, p. 349 of [SV]. It provides a fundamental decomposition of $\mathit{flow}(X, r)$ in terms of sets that depend on the individual arguments X and r.

Theorem 4.1. $\mathit{flow}(X, r) = (X \cup \mathit{flow}(\{\}, r)) - \mathit{block}(r)$.

Proof: By induction over the structure of r.

- $r = \texttt{(1, }v\texttt{ = }e\texttt{)}$.

$$
\begin{aligned}
&\mathit{flow}(X, \texttt{(1, }v\texttt{ = }e\texttt{)}) \\
={}& X \cup \{v\} \\
={}& (X \cup (\{\} \cup \{v\})) - \{\} \\
={}& (X \cup \mathit{flow}(\{\}, \texttt{(1, }v\texttt{ = }e\texttt{)})) - \mathit{block}(\texttt{(1, }v\texttt{ = }e\texttt{)})
\end{aligned}
$$

- $r = b$.

$$
\begin{aligned}
&\mathit{flow}(X, b) \\
={}& X \\
={}& (X \cup \{\}) - \{\} \\
={}& (X \cup \mathit{flow}(\{\}, b)) - \mathit{block}(b)
\end{aligned}
$$

- $r = \texttt{(}s\texttt{)}$.

$$
\begin{aligned}
&\mathit{flow}(X, \texttt{(}s\texttt{)}) \\
={}& \mathit{flow}(X, s) \\
={}& [\text{induction}] \\
&(X \cup F_s) - B_s \\
={}& (X \cup \mathit{flow}(\{\}, \texttt{(}s\texttt{)})) - \mathit{block}(\texttt{(}s\texttt{)})
\end{aligned}
$$

- $r = \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}$. Note that

$$
\begin{aligned}
&\mathit{flow}(\{\}, \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}) \\
={}& \mathit{flow}(F_s, t) \\
={}& [\text{induction}] \\
&(F_s \cup F_t) - B_t
\end{aligned}
$$

  Then

$$
\begin{aligned}
&\mathit{flow}(X, \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}) \\
={}& \mathit{flow}(\mathit{flow}(X, s), t) \\
={}& [\text{induction}] \\
&\mathit{flow}((X \cup F_s) - B_s, t) \\
={}& [\text{induction}] \\
&(((X \cup F_s) - B_s) \cup F_t) - B_t \\
={}& [(A - M) \cup B = (A \cup B) - (M - B)] \\
&(((X \cup F_s) \cup F_t) - (B_s - F_t)) - B_t
\end{aligned}
$$

$$\begin{aligned}
&= (X \cup (F_s \cup F_t)) - ((B_s - F_t) \cup B_t) \\
&= [(A \cup B) - (M \cup N) = (A \cup (B - N)) - (M \cup N)] \\
& (X \cup ((F_s \cup F_t) - B_t)) - ((B_s - F_t) \cup B_t) \\
&= (X \cup \mathit{flow}(\{\}, \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)})) - \mathit{block}(\texttt{(}s\ \texttt{\#\#1}\ t\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\ \texttt{\#\#0}\ t\texttt{)}$. Analogous to the argument for $r = \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}$.
- $r = \texttt{(}s\ \texttt{or}\ t\texttt{)}$.

$$\begin{aligned}
&\mathit{flow}(X, \texttt{(}s\ \texttt{or}\ t\texttt{)}) \\
&= \mathit{flow}(X, s) \cap \mathit{flow}(X, t) \\
&= [\text{induction}] \\
& ((X \cup F_s) - B_s) \cap ((X \cup F_t) - B_t) \\
&= [(A - M) \cap (B - N) = (A \cap B) - (M \cup N)] \\
& ((X \cup F_s) \cap (X \cup F_t)) - (B_s \cup B_t) \\
&= (X \cup (F_s \cap F_t)) - (B_s \cup B_t) \\
&= (X \cup \mathit{flow}(\{\}, \texttt{(}s\ \texttt{or}\ t\texttt{)})) - \mathit{block}(\texttt{(}s\ \texttt{or}\ t\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\ \texttt{intersect}\ t\texttt{)}$.

$$\begin{aligned}
&\mathit{flow}(X, \texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \\
&= (\mathit{flow}(X, s) \cup \mathit{flow}(X, t)) - \mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \\
&= [\text{induction}] \\
& (((X \cup F_s) - B_s) \cup ((X \cup F_t) - B_t)) - \mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \\
&= [B_s \subseteq \mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}), \text{ similarly for } B_t] \\
& (X \cup (F_s \cup F_t)) - \mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \\
&= [(A \cup B) - M = (A \cup (B - M)) - M] \\
& (X \cup ((F_s \cup F_t) - \mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}))) - \mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \\
&= (X \cup \mathit{flow}(\{\}, \texttt{(}s\ \texttt{intersect}\ t\texttt{)})) - \mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)})
\end{aligned}$$

- $r = \texttt{first\_match(}s\texttt{)}$. Analogous to the argument for $r = \texttt{(}s\texttt{)}$.
- $r = s\texttt{[*0]}$. Analogous to the argument for $r = b$.
- $r = s\texttt{[*1:\$]}$. Analogous to the argument for $r = \texttt{(}s\texttt{)}$.

□

Corollary 4.2.

1. *flow*$(X, r) \cap$ *block*$(r) = \{\}$. *In particular,* *flow*$(\{\}, r) \cap$ *block*$(r) = \{\}$.
2. *flow*$(X, r) = (X -$ *block*$(r)) \cup$ *flow*$(\{\}, r)$.

□

The first part of Corollary 4.2 shows that *block* and *flow* are disjoint when applied to the same sequence. This justifies the intuitive notion that local variables that flow from a sequence must not have been blocked, and local variables that are blocked by a sequence do not flow from it. The specialization of Corollary 4.2(1) to $X = \{\}$ is noted in Annex G, Subsection G.3.3.3, p. 349 of [SV].

The next proposition shows that, in its first argument, *flow* respects set union and is monotonic.

Proposition 4.3.

1. *$\mathit{flow}(X \cup Y, r) = \mathit{flow}(X, r) \cup \mathit{flow}(Y, r)$.*

2. *If $X \subseteq Y$, then $\mathit{flow}(X, r) \subseteq \mathit{flow}(Y, r)$.*

Proof:

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{flow}(X \cup Y, r)} \\
= & \text{[Theorem 4.1]} \\
 & (X \cup Y \cup F_r) - B_r \\
= & ((X \cup F_r) \cup (Y \cup F_r)) - B_r \\
= & ((X \cup F_r) - B_r) \cup ((Y \cup F_r) - B_r) \\
= & \text{[Theorem 4.1]} \\
 & \mathit{flow}(X, r) \cup \mathit{flow}(Y, r)
\end{array}$$

This proves part 1. If $X \subseteq Y$, then $X \cup Y = Y$, so part 2 follows from part 1. $\square$

The next proposition shows that fixing the second argument of *flow* yields an idempotent operator on sets of local variables.

Proposition 4.4. *$\mathit{flow}(\mathit{flow}(X, r), r) = \mathit{flow}(X, r)$.*

Proof:

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{flow}(\mathit{flow}(X, r), r)} \\
= & \text{[Theorem 4.1]} \\
 & \mathit{flow}((X \cup F_r) - B_r, r) \\
= & \text{[Theorem 4.1]} \\
 & (((X \cup F_r) - B_r) \cup F_r) - B_r \\
= & [F_r \cap B_r = \{\}] \\
 & (((X \cup F_r) \cup F_r) - B_r) - B_r \\
= & (X \cup F_r) - B_r \\
= & \text{[Theorem 4.1]} \\
 & \mathit{flow}(X, r)
\end{array}$$

$\square$

Remark: Notice that Proposition 4.4 implies that for any $m > 0$,

$$\mathit{flow}(X, r\texttt{[*}m\texttt{]}) = \mathit{flow}(X, r)$$

This shows that the set of local variables that flows through one or more concatenations or repetitions of a sequence does not depend on the number of repetitions. This is an important consistency condition, and it is a justification of the definition

$$\mathit{flow}(X, r\texttt{[*1:\$]}) = \mathit{flow}(X, r)$$

$\square$

Part 1 of the following theorem corresponds to the intuitive notion that if there are no incoming local variables, then any local variable that flows from a sequence must have been assigned in the sequence. Part 1 is claimed in Annex G, Subsection G.3.3.3, p. 349 of [SV]. Part 2 of the theorem corresponds to the intuitive notion that only local variables assigned in a sequence can be blocked by the sequence. This relationship is expected because the blocking rules are designed to eliminate inconsistently assigned local variables.

Theorem 4.5.

1. $\mathit{flow}(\{\}, r) \subseteq \mathit{sample}(r)$.

2. $\mathit{block}(r) \subseteq \mathit{sample}(r)$.

Proof: By induction over the structure of r.

- $r = \texttt{(1, }v\texttt{ = }e\texttt{)}$.

$$\begin{aligned}
&\mathit{flow}(\{\}, \texttt{(1, }v\texttt{ = }e\texttt{)})\\
=\ &\{v\}\\
=\ &\mathit{sample}(\texttt{(1, }v\texttt{ = }e\texttt{)})
\end{aligned}$$

$$\begin{aligned}
&\mathit{block}(\texttt{(1, }v\texttt{ = }e\texttt{)})\\
=\ &\{\}\\
\subseteq\ &\mathit{sample}(\texttt{(1, }v\texttt{ = }e\texttt{)})
\end{aligned}$$

- $r = b$.

$$\begin{aligned}
&\mathit{flow}(\{\}, b)\\
=\ &\{\}\\
=\ &\mathit{sample}(b)
\end{aligned}$$

$$\begin{aligned}
&\mathit{block}(b)\\
=\ &\{\}\\
=\ &\mathit{sample}(b)
\end{aligned}$$

- $r = \texttt{(}s\texttt{)}$.

$$\begin{aligned}
&\mathit{flow}(\{\}, \texttt{(}s\texttt{)})\\
=\ &\mathit{flow}(\{\}, s)\\
\subseteq\ &[\text{induction}]\\
&\mathit{sample}(s)\\
=\ &\mathit{sample}(\texttt{(}s\texttt{)})
\end{aligned}$$

$$\begin{aligned}
&\mathit{block}(\texttt{(}s\texttt{)})\\
=\ &\mathit{block}(s)\\
\subseteq\ &[\text{induction}]\\
&\mathit{sample}(s)\\
=\ &\mathit{sample}(\texttt{(}s\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\texttt{ \#\#1 }t\texttt{)}$.

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{flow}(\{\}, \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)})} \\
= & \mathit{flow}(F_s, t) \\
= & [\text{Theorem 4.1}] \\
 & (F_s \cup F_t) - B_t \\
\subseteq & [\text{induction}] \\
 & (S_s \cup S_t) - B_t \\
\subseteq & (S_s \cup S_t) \\
= & \mathit{sample}(\texttt{(}s\ \texttt{\#\#1}\ t\texttt{)})
\end{array}$$

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{block}(\texttt{(}s\ \texttt{\#\#1}\ t\texttt{)})} \\
= & (B_s - F_t) \cup B_t \\
\subseteq & B_s \cup B_t \\
\subseteq & [\text{induction}] \\
 & S_s \cup S_t \\
= & \mathit{sample}(\texttt{(}s\ \texttt{\#\#1}\ t\texttt{)})
\end{array}$$

- $r = \texttt{(}s\ \texttt{\#\#0}\ t\texttt{)}$. Analogous to the argument for $r = \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}$.
- $r = \texttt{(}s\ \texttt{or}\ t\texttt{)}$.

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{flow}(\{\}, \texttt{(}s\ \texttt{or}\ t\texttt{)})} \\
= & F_s \cap F_t \\
\subseteq & F_s \cup F_t \\
\subseteq & [\text{induction}] \\
 & S_s \cup S_t \\
= & \mathit{sample}(\texttt{(}s\ \texttt{or}\ t\texttt{)})
\end{array}$$

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{block}(\texttt{(}s\ \texttt{or}\ t\texttt{)})} \\
= & B_s \cup B_t \\
\subseteq & [\text{induction}] \\
 & S_s \cup S_t \\
= & \mathit{sample}(\texttt{(}s\ \texttt{or}\ t\texttt{)})
\end{array}$$

- $r = \texttt{(}s\ \texttt{intersect}\ t\texttt{)}$.

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{flow}(\{\}, \texttt{(}s\ \texttt{intersect}\ t\texttt{)})} \\
= & (F_s \cup F_t) - \mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \\
\subseteq & F_s \cup F_t \\
\subseteq & [\text{induction}] \\
 & S_s \cup S_t \\
= & \mathit{sample}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)})
\end{array}$$

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)})} \\
= & B_s \cup B_t \cup (S_s \cap S_t) \\
\subseteq & [\text{induction}] \\
 & S_s \cup S_t \cup (S_s \cap S_t) \\
= & S_s \cup S_t \\
= & \mathit{sample}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)})
\end{array}$$

- $r =$ `first_match(`s`)`. Analogous to the argument for $r =$ `(`s`)`.
- $r = s$`[*0]`. Analogous to the argument for $r = b$.
- $r = s$`[*1:$]`. Analogous to the argument for $r =$ `(`s`)`.

$\square$

5 Basic Properties of Tight Satisfaction

The first theorem in this section shows consistency between the treatment of local variable flow in the definitions of *flow* and of tight satisfaction. The second theorem shows a conservation result that in the absence of assignment, local variables persist with values unchanged. Both are important for showing that the definitions achieve the intended intuitive behavior of local variables.

Lemma 5.1. *If $w, L_0, L_1 \models\!\equiv r$, then no letter of w is $\bot$.*

Proof: By induction over the structure of r. Write "$\mathit{good}(w)$" to mean that no letter of w is $\bot$.

- $r = \texttt{(1, } v \texttt{ = } e\texttt{)}$.

  $w, L_0, L_1 \models\!\equiv \texttt{(1, } v \texttt{ = } e\texttt{)}$
  $\Rightarrow$ $|w| = 1$ and $w^0 \Vdash \texttt{1}$
  $\Rightarrow$ $|w| = 1$ and $w^0 \neq \bot$
  $\Rightarrow$ $\mathit{good}(w)$

- $r = b$.

  $w, L_0, L_1 \models\!\equiv b$
  iff $|w| = 1$ and $w^0 \Vdash b[L_0]$ and $L_1 = L_0$
  $\Rightarrow$ $|w| = 1$ and $w^0 \neq \bot$
  $\Rightarrow$ $\mathit{good}(w)$

- $r = \texttt{(}r_1\texttt{)}$.

  $w, L_0, L_1 \models\!\equiv \texttt{(}r_1\texttt{)}$
  iff $w, L_0, L_1 \models\!\equiv r_1$
  $\Rightarrow$ [induction]
  $\mathit{good}(w)$

- $r = \texttt{(}r_1 \texttt{ ##1 } r_2\texttt{)}$.

  $w, L_0, L_1 \models\!\equiv \texttt{(}r_1 \texttt{ ##1 } r_2\texttt{)}$
  iff there exist u, v, L such that $w = uv$ and $u, L_0, L \models\!\equiv r_1$ and $v, L, L_1 \models\!\equiv r_2$
  $\Rightarrow$ [induction]
  there exist u, v such that $w = uv$ and $\mathit{good}(u)$ and $\mathit{good}(v)$
  $\Rightarrow$ $\mathit{good}(w)$

- $r = \texttt{(}r_1 \texttt{ ##0 } r_2\texttt{)}$.

  $w, L_0, L_1 \models\!\equiv \texttt{(}r_1 \texttt{ ##0 } r_2\texttt{)}$
  iff there exist x, y, z, L such that $w = xyz$ and $|y| = 1$ and $xy, L_0, L \models\!\equiv r_1$ and $yz, L, L_1 \models\!\equiv r_2$
  $\Rightarrow$ [induction]
  there exist x, y, z such that $w = xyz$ and $\mathit{good}(xy)$ and $\mathit{good}(yz)$
  $\Rightarrow$ $\mathit{good}(w)$

- $r = $ `(`r_1 `or` r_2`)`.

  $w, L_0, L_1 \models\!\equiv$ `(`r_1 `or` r_2`)`
  $\Rightarrow$ there exists L such that $w, L_0, L \models\!\equiv r_1$ or $w, L_0, L \models\!\equiv r_2$
  $\Rightarrow$ [induction]
  good(w) or *good*(w)
  iff *good*(w)

- $r = $ `(`r_1 `intersect` r_2`)`.

  $w, L_0, L_1 \models\!\equiv$ `(`r_1 `intersect` r_2`)`
  $\Rightarrow$ there exists L, L' such that $w, L_0, L \models\!\equiv r_1$ and $w, L_0, L' \models\!\equiv r_2$
  $\Rightarrow$ [induction]
  good(w) and *good*(w)
  iff *good*(w)

- $r = $ `first_match(`r_1`)`.

  $w, L_0, L_1 \models\!\equiv$ `first_match(`r_1`)`
  $\Rightarrow$ $w, L_0, L_1 \models\!\equiv r_1$
  $\Rightarrow$ [induction]
  good(w)

- $r = r_1$`[*0]`.

  $w, L_0, L_1 \models\!\equiv r_1$`[*0]`
  $\Rightarrow$ $|w| = 0$
  $\Rightarrow$ *good*(w)

- $r = r_1$`[*1:$]`.

  $w, L_0, L_1 \models\!\equiv r_1$`[*1:$]`
  iff there exist $k > 0$ and $L_0 = L'_0, w_1, L'_1, \ldots, L'_{k-1}, w_k, L'_k = L_1$ such that $w = w_1 \cdots w_k$ and $w_j, L'_{j-1}, L'_j \models\!\equiv r_1$ for all $1 \leq j \leq k$
  $\Rightarrow$ [induction]
  there exist $k > 0$ and $w_1, \ldots, w_k$ such that $w = w_1 \cdots w_k$ and *good*(w_j) for all $1 \leq j \leq k$
  $\Rightarrow$ *good*(w)

$\square$

Lemma 5.2.

1. *block*`((`s `intersect` t`))` $\cup$ *sample*(t) = *block*$(s) \cup$ *sample*(t).

2. *block*`((`s `intersect` t`))` $\cup$ *sample*(s) = *block*$(t) \cup$ *sample*(s).

3. $($*block*$(s) \cup$ *sample*$(t)) \cap ($*block*$(t) \cup$ *sample*$(s))$ = *block*`((`s `intersect` t`))`.

Proof:

$$\begin{aligned}
&\mathit{block}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \cup \mathit{sample}(t)\\
=\ & B_s \cup B_t \cup (S_s \cap S_t) \cup S_t\\
=\ & B_s \cup B_t \cup S_t\\
=\ & [\text{Theorem 4.5}]\\
& B_s \cup S_t
\end{aligned}$$

This proves part 1. Part 2 follows by symmetry. For part 3,

$$\begin{aligned}
&(B_s \cup S_t) \cap (B_t \cup S_s)\\
=\ & [\text{parts 1 and 2}]\\
& (B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup S_t) \cap (B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup S_s)\\
=\ & B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup (S_s \cap S_t)\\
=\ & B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}
\end{aligned}$$

□

The following lemma gives a representation of the *flow* through "`intersect`" as a symmetric union. It shows the way in which *flow* through "`intersect`" is analogous to symmetric difference.

Lemma 5.3.

1. $X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)} = (X/s - S_t) \cup (X/t - S_s)$.

2. *The sets D', D'' in the definition of tight satisfaction of intersection are*

$$\begin{aligned}
D' &= \mathit{flow}(\mathrm{dom}(L_0), s) - \mathit{sample}(t)\\
D'' &= \mathit{flow}(\mathrm{dom}(L_0), t) - \mathit{sample}(s)
\end{aligned}$$

Proof:

$$\begin{aligned}
&X/s - S_t\\
=\ & [\text{Theorem 4.1}]\\
& (X \cup F_s) - (B_s \cup S_t)\\
=\ & (X - (B_s \cup S_t)) \cup (F_s - (B_s \cup S_t))
\end{aligned}$$

Note that

$$\begin{aligned}
&F_s - (B_s \cup S_t)\\
=\ & [\text{Lemma 5.2}]\\
& F_s - (B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup S_t)\\
=\ & [\text{by Theorem 4.5(1), } F_s \subseteq S_s]\\
& F_s - (B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup (S_s \cap S_t))\\
=\ & [S_s \cap S_t \subseteq B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}]\\
& F_s - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}
\end{aligned}$$

Similarly

$$X/t - S_s = (X - (B_t \cup S_s)) \cup (F_t - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}})$$

Therefore

$$
\begin{array}{ll}
\multicolumn{2}{l}{(X/s - S_t) \cup (X/t - S_s)} \\
= & (X - (B_s \cup S_t)) \ \cup \ (X - (B_t \cup S_s)) \\
 & \cup \\
 & (F_s - B_{\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}}) \ \cup \ (F_t - B_{\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}}) \\
= & (X - ((B_s \cup S_t) \cap (B_t \cup S_s))) \\
 & \cup \\
 & ((F_s \cup F_t) - B_{\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}}) \\
= & [\text{Lemma 5.2}] \\
 & (X - B_{\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}}) \\
 & \cup \\
 & ((F_s \cup F_t) - B_{\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}}) \\
= & (X - B_{\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}}) \ \cup \ F_{\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}} \\
= & [\text{Corollary 4.2(2)}] \\
 & X/\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}
\end{array}
$$

This proves part 1. Let $X = \mathrm{dom}(L_0)$.

$$
\begin{array}{ll}
\multicolumn{2}{l}{D'} \\
= & X/s - (B_{\mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)}} \cup S_t) \\
= & [\text{Lemma 5.2}] \\
 & X/s - (B_s \cup S_t) \\
= & [\text{by Corollary 4.2(1), } X/s \cap B_s = \{\}] \\
 & X/s - S_t
\end{array}
$$

Similarly,

$$D'' = X/t - S_s$$

This proves part 2. □

Remark: Lemma 5.3 shows that

$$(\mathit{flow}(X, s) - \mathit{sample}(t)) \ \cup \ (\mathit{flow}(X, t) - \mathit{sample}(s))$$

could be taken as the definition of

$$\mathit{flow}(X, \mathtt{(}s\ \mathtt{intersect}\ t\mathtt{)})$$

eliminating the dependency of the definition of *flow* on the auxiliary function *block*. This topic is discussed further in Section 7. □

The following theorem shows consistency in the treatment of local variables in the definition of *flow* and in the definition of tight satisfaction. It shows that the domains of the incoming and outgoing local variable contexts in the tight satisfaction relation are related by *flow*. This theorem is claimed in Annex G, Subsection G.3.4, p. 350 of [SV].

Theorem 5.4. *If* $w, L_0, L_1 \mathrel{|\!\!\equiv} r$, *then* $\mathrm{dom}(L_1) = \mathit{flow}(\mathrm{dom}(L_0), r)$.

Proof: By induction over the structure of r. Assume $w, L_0, L_1 \mathrel{|\!\!\equiv} r$.

- $r = \mathtt{(1,}\ v\ \mathtt{=}\ e\mathtt{)}$.

$$\begin{aligned}
&\mathrm{dom}(L_1)\\
=\;\;&\mathrm{dom}(\{(v, e[L_0, w^0])\} \cup L_0|_{\mathrm{dom}(L_0)-\{v\}})\\
=\;\;&\mathrm{dom}(L_0) \cup \{v\}\\
=\;\;&\mathit{flow}(\mathrm{dom}(L_0), \texttt{(1, }v\texttt{ = }e\texttt{)})
\end{aligned}$$

- $r = b$.

$$\begin{aligned}
&\mathrm{dom}(L_1)\\
=\;\;&\mathrm{dom}(L_0)\\
=\;\;&\mathit{flow}(\mathrm{dom}(L_0), b)
\end{aligned}$$

- $r = \texttt{(}s\texttt{)}$. Then $w, L_0, L_1 \mathrel{|}\!\equiv s$, so

$$\begin{aligned}
&\mathrm{dom}(L_1)\\
=\;\;&[\text{induction}]\\
&\mathit{flow}(\mathrm{dom}(L_0), s)\\
=\;\;&\mathit{flow}(\mathrm{dom}(L_0), \texttt{(}s\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\texttt{ ##1 }t\texttt{)}$. Then there exist x, y, L' such that $x, L_0, L' \mathrel{|}\!\equiv s$ and $y, L', L_1 \mathrel{|}\!\equiv t$. Then

$$\begin{aligned}
&\mathrm{dom}(L_1)\\
=\;\;&[\text{induction}]\\
&\mathit{flow}(\mathrm{dom}(L'), t)\\
=\;\;&[\text{induction}]\\
&\mathit{flow}(\mathit{flow}(\mathrm{dom}(L_0), s), t)\\
=\;\;&\mathit{flow}(\mathrm{dom}(L_0), \texttt{(}s\texttt{ ##1 }t\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\texttt{ ##0 }t\texttt{)}$. Then there exist x, y, z, L' such that $w = xyz$ and $|y| = 1$ and $xy, L_0, L' \mathrel{|}\!\equiv s$ and $yz, L', L_1 \mathrel{|}\!\equiv t$. Then

$$\begin{aligned}
&\mathrm{dom}(L_1)\\
=\;\;&[\text{induction}]\\
&\mathit{flow}(\mathrm{dom}(L'), t)\\
=\;\;&[\text{induction}]\\
&\mathit{flow}(\mathit{flow}(\mathrm{dom}(L_0), s), t)\\
=\;\;&\mathit{flow}(\mathrm{dom}(L_0), \texttt{(}s\texttt{ ##0 }t\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\texttt{ or }t\texttt{)}$. Then there exists L' such that both of the following hold:

  - either $w, L_0, L' \mathrel{|}\!\equiv s$ or $w, L_0, L' \mathrel{|}\!\equiv t$, and
  - $L_1 = L'|_D$, where $D = \mathit{flow}(\mathrm{dom}(L_0), \texttt{(}s\texttt{ or }t\texttt{)})$.

  Suppose $w, L_0, L' \mathrel{|}\!\equiv s$. By induction, $\mathrm{dom}(L') = \mathit{flow}(\mathrm{dom}(L_0), s)$. Since

$$D = \mathit{flow}(\mathrm{dom}(L_0), s) \cap \mathit{flow}(\mathrm{dom}(L_0), t) \subseteq \mathrm{dom}(L')$$

  we get $\mathrm{dom}(L_1) = D$. Similarly if $w, L_0, L' \mathrel{|}\!\equiv t$.

- $r = (s\ \texttt{intersect}\ t)$. By Lemma 5.3, there exist L', L'' such that $w, L_0, L' \models\!\!\equiv s$ and $w, L_0, L'' \models\!\!\equiv t$ and $L_1 = L'|_{D'} \cup L''|_{D''}$, where

  $$\begin{aligned} D' &= \mathit{flow}(\mathrm{dom}(L_0), s) - \mathit{sample}(t) \\ D'' &= \mathit{flow}(\mathrm{dom}(L_0), t) - \mathit{sample}(s) \end{aligned}$$

  Lemma 5.3 also implies that

  $$D' \cup D'' = \mathit{flow}(\mathrm{dom}(L_0), (s\ \texttt{intersect}\ t))$$

  By induction, $\mathrm{dom}(L') = \mathit{flow}(\mathrm{dom}(L_0), s)$ and $\mathrm{dom}(L'') = \mathit{flow}(\mathrm{dom}(L_0), t)$, so $D' \subseteq \mathrm{dom}(L')$ and $D'' \subseteq \mathrm{dom}(L'')$. Therefore,

  $$\mathrm{dom}(L_1) = D' \cup D''$$

- $r = \texttt{first\_match(}s\texttt{)}$. Then $w, L_0, L_1 \models\!\!\equiv s$. By induction,

  $$\mathrm{dom}(L_1) = \mathit{flow}(\mathrm{dom}(L_0), s) = \mathit{flow}(\mathrm{dom}(L_0), \texttt{first\_match(}s\texttt{)})$$

- $r = s\texttt{[*0]}$. Then $L_1 = L_0$, so

  $$\mathrm{dom}(L_1) = \mathrm{dom}(L_0) = \mathit{flow}(\mathrm{dom}(L_0), s\texttt{[*0]})$$

- $r = s\texttt{[*1:\$]}$. Then there exist $L_{(0)} = L_0, w_1, L_{(1)}, w_2, L_{(2)}, \ldots, w_j, L_{(j)} = L_1$ $(j \geq 1)$ such that $w = w_1 \cdots w_j$ and for every i such that $1 \leq i \leq j$, $w_i, L_{(i-1)}, L_{(i)} \models\!\!\equiv s$. By induction, for every i such that $1 \leq i \leq j$,

  $$\mathrm{dom}(L_{(i)}) = \mathit{flow}(\mathrm{dom}(L_{(i-1)}), s)$$

  By Proposition 4.4, $\mathit{flow}(\mathit{flow}(X, s), s) = \mathit{flow}(X, s)$. Therefore, for every i such that $1 \leq i \leq j$,

  $$\mathit{flow}(\mathrm{dom}(L_{(i)}), s) = \mathit{flow}(\mathrm{dom}(L_{(i-1)}), s)$$

  and so

  $$\mathrm{dom}(L_1) = \mathit{flow}(\mathrm{dom}(L_0), s) = \mathit{flow}(\mathrm{dom}(L_0), s\texttt{[*1:\$]})$$

□

The following theorem shows a conservation result that in the absence of assignment, local variables persist with values unchanged.

Theorem 5.5. *If $w, L_0, L_1 \models\!\!\equiv r$ and $v \in \mathrm{dom}(L_0) - \mathit{sample}(r)$, then $v \in \mathrm{dom}(L_1)$ and $L_1(v) = L_0(v)$.*

Proof: By induction over the structure of r. Note that Theorems 4.1, 4.5, and 5.4 give

$$\begin{aligned} \mathrm{dom}(L_1) &= \mathit{flow}(\mathrm{dom}(L_0), r) \\ &= (\mathrm{dom}(L_0) \cup \mathit{flow}(\{\}, r)) - \mathit{block}(r) \\ &\supseteq \mathrm{dom}(L_0) - \mathit{sample}(r) \end{aligned}$$

Therefore, $v \in \mathrm{dom}(L_1)$.

- $r =$ `(1,` v' `=` e`)`. Then

  $$L_1 = \{(v', e[L_0, w^0])\} \cup L_0|_{\mathrm{dom}(L_0)-\{v'\}}$$

  Since *sample*(`(1,` v' `=` e`)`) $= \{v'\}$, $v \neq v'$, and thus $L_1(v) = L_0(v)$.

- $r = b$. Then $L_1 = L_0$.

- $r =$ `(`s`)`. Then $w, L_0, L_1 \models\equiv s$. Since *sample*$(r) =$ *sample*(s), $v \in \mathrm{dom}(L_0) -$ *sample*(s), so induction gives $L_1(v) = L_0(v)$.

- $r =$ `(`s `##1` t`)`. Then there exist x, y, L' such that $x, L_0, L' \models\equiv s$ and $y, L', L_1 \models\equiv t$. Note that *sample*$(r) =$ *sample*$(s) \cup$ *sample*(t). Therefore, $v \in \mathrm{dom}(L_0) -$ *sample*(s), so by induction $v \in \mathrm{dom}(L')$ and $L'(v) = L_0(v)$. Therefore $v \in \mathrm{dom}(L') -$ *sample*(t), so by induction $L_1(v) = L'(v)$, hence $L_1(v) = L_0(v)$.

- $r =$ `(`s `##0` t`)`. Similar to the argument for $r =$ `(`s `##1` t`)`.

- $r =$ `(`s `or` t`)`. Then there exists L' such that both of the following hold:

  – either $w, L_0, L' \models\equiv s$ or $w, L_0, L' \models\equiv t$, and

  – $L_1 = L'|_D$, where $D =$ *flow*$(\mathrm{dom}(L_0),$ `(`s `or` t`)`$)$.

  Suppose $w, L_0, L' \models\equiv s$. Since *sample*$(r) =$ *sample*$(s) \cup$ *sample*(t), it follows that $v \in \mathrm{dom}(L_0) -$ *sample*(s), and so by induction, $v \in \mathrm{dom}(L')$ and $L'(v) = L_0(v)$. Since $L_1 = L'|_D$ and $v \in \mathrm{dom}(L_1)$, it follows that $L_1(v) = L'(v)$. Similarly if $w, L_0, L' \models\equiv t$.

- $r =$ `(`s `intersect` t`)`. Then there exist L', L'', D', D'' such that $w, L_0, L' \models\equiv s$ and $w, L_0, L'' \models\equiv t$ and $L_1 = L'|_{D'} \cup L''|_{D''}$. Note that *sample*$(r) =$ *sample*$(s) \cup$ *sample*(t). Therefore, $v \in \mathrm{dom}(L_0) -$ *sample*(s), so by induction $v \in \mathrm{dom}(L')$ and $L'(v) = L_0(v)$. Similarly, $v \in \mathrm{dom}(L'')$ and $L''(v) = L_0(v)$. Since $v \in L_1$, either $v \in \mathrm{dom}(L'|_{D'})$ or $v \in \mathrm{dom}(L''|_{D''})$, and in either case it follows that $L_1(v) = L_0(v)$.

- $r =$ `first_match(`s`)`. Then $w, L_0, L_1 \models\equiv s$. Since *sample*$(r) =$ *sample*(s), $v \in \mathrm{dom}(L_0) -$ *sample*(s). By induction, $L_1(v) = L_0(v)$.

- $r = s$`[*0]`. Then $L_1 = L_0$.

- $r = s$`[*1:$]`. Then there exist $L_{(0)} = L_0, w_1, L_{(1)}, w_2, L_{(2)}, \ldots, w_j, L_{(j)} = L_1$ $(j \geq 1)$ such that $w = w_1 \cdots w_j$ and for every i such that $1 \leq i \leq j$, $w_i, L_{(i-1)}, L_{(i)} \models\equiv s$. Note that *sample*$(r) =$ *sample*(s). Therefore, by induction, for every i such that $1 \leq i \leq j$,

  $$v \in \mathrm{dom}(L_{(i)}) \ \text{ and } \ L_{(i)}(v) = L_{(i-1)}(v)$$

  It follows that $L_1(v) = L_0(v)$.

□

A local variable context is, by definition, a function. In the definition of tight satisfaction for "`intersect`", the outgoing local variable context L_1 is required to be equal to a union of functions, "$L'|_{D'} \cup L''|_{D''}$". The following proposition shows that this union is guaranteed to be a function, so there is no extra condition imposed by well-definedness of L_1. This proposition is claimed in Annex G, Subsection G.3.4, p. 350 of [SV].

Proposition 5.6. *In the definition of tight satisfaction for intersection, $L'|_{D'} \cup L''|_{D''}$ is a function.*

Proof: Each of L', L'' is a function. Let $X = \mathrm{dom}(L_0)$. By Lemma 5.3,

$$\begin{aligned} D' &= X/s - S_t \\ D'' &= X/t - S_s \end{aligned}$$

By Theorem 5.4, $\mathrm{dom}(L') = X/s \supseteq D'$ and $\mathrm{dom}(L'') = X/t \supseteq D''$. It suffices to show that if $v \in D' \cap D''$, then $L'(v) = L''(v)$. Note that

$$\begin{aligned} & D' \cap D'' \\ = \;& (X/s \cap X/t) - (S_s \cup S_t) \\ = \;& [\text{Theorem 4.1}] \\ & (((X \cup F_s) - B_s) \cap ((X \cup F_t) - B_t)) - (S_s \cup S_t) \\ = \;& [\text{By Theorem 4.5, } F_s \cup B_s \cup F_t \cup B_t \subseteq S_s \cup S_t] \\ & X - (S_s \cup S_t) \end{aligned}$$

Let $v \in D' \cap D''$. Then $v \in X - S_s$. Since $w, L_0, L' \models\!\equiv s$, Theorem 5.5 gives $L'(v) = L_0(v)$. Similarly, $L''(v) = L_0(v)$. □

6 Algebraic Properties

The algebraic properties of the various sequence-building operators can be studied at several levels. At the level of the sequences themselves, there is not much interesting to say since syntactically distinct sequences are distinct. The study of algebraic properties is more fruitful on equivalence classes of sequences relative to various equivalence relations. The equivalence relations of immediate interest are those derived from the definitions of tight satisfaction and of the auxiliary functions, *sample*, *block*, and *flow*.

Definition 6.0.

- s and t are *sample equivalent* iff $\mathit{sample}(s) = \mathit{sample}(t)$.
- s and t are *block equivalent* iff $\mathit{block}(s) = \mathit{block}(t)$.
- s and t are *flow equivalent* iff for all X, $\mathit{flow}(X, s) = \mathit{flow}(X, t)$.
- s and t are *semantically equivalent* iff for all w, L_0, L_1, $w, L_0, L_1 \models\!\!\equiv s$ iff $w, L_0, L_1 \models\!\!\equiv t$.

□

Thus, to say that "`or`" is commutative at the level of *sample* means that

$$\mathit{sample}(\texttt{(}s \texttt{ or } t\texttt{)}) = \mathit{sample}(\texttt{(}t \texttt{ or } s\texttt{)})$$

for all sequences s, t. On the other hand, to say that "`or`" is commutative at the semantic level means that for all s, t, w, L_0, L_1,

$$w, L_0, L_1 \models\!\!\equiv \texttt{(}s \texttt{ or } t\texttt{)} \ \text{ iff } \ w, L_0, L_1 \models\!\!\equiv \texttt{(}t \texttt{ or } s\texttt{)}$$

For users of SystemVerilog 3.1 Assertions, algebraic properties at the semantic level are the most important. Therefore, this section is concerned mainly with the semantic level. The semantic level is also the most difficult, since it involves not only which local variables flow, but also what values they have in the local variable contexts. Results about the other levels of equivalence are, not surprisingly, established along the way.

6.1 Commutativity

If "$\odot$" is a binary operator on a set A, then "$\odot$" is said to be *commutative* if $a \odot b = b \odot a$ for all $a, b \in A$.

The symmetry of the definitions for "`or`", "`intersect`", and "`and`" leads easily to the following propositions, which are stated without proof. They show that "`or`", "`intersect`", and "`and`" are all commutative at the levels of *sample*, *block*, *flow*, and at the semantic level.

Proposition 6.1.1 (commutativity of "`or`").

1. $S_{\texttt{(}s \texttt{ or } t\texttt{)}} = S_{\texttt{(}t \texttt{ or } s\texttt{)}}$.

2. $B_{(s\ \texttt{or}\ t)} = B_{(t\ \texttt{or}\ s)}$.

3. $X/(s\ \texttt{or}\ t) = X/(t\ \texttt{or}\ s)$.

4. $w, L_0, L_1 \models\!\equiv (s\ \texttt{or}\ t)$ *iff* $w, L_0, L_1 \models\!\equiv (t\ \texttt{or}\ s)$.

□

Proposition 6.1.2 (commutativity of "`intersect`").

1. $S_{(s\ \texttt{intersect}\ t)} = S_{(t\ \texttt{intersect}\ s)}$.

2. $B_{(s\ \texttt{intersect}\ t)} = B_{(t\ \texttt{intersect}\ s)}$.

3. $X/(s\ \texttt{intersect}\ t) = X/(t\ \texttt{intersect}\ s)$.

4. $w, L_0, L_1 \models\!\equiv (s\ \texttt{intersect}\ t)$ *iff* $w, L_0, L_1 \models\!\equiv (t\ \texttt{intersect}\ s)$.

□

Proposition 6.1.3 (commutativity of "`and`").

1. $S_{(s\ \texttt{and}\ t)} = S_{(t\ \texttt{and}\ s)}$.

2. $B_{(s\ \texttt{and}\ t)} = B_{(t\ \texttt{and}\ s)}$.

3. $X/(s\ \texttt{and}\ t) = X/(t\ \texttt{and}\ s)$.

4. $w, L_0, L_1 \models\!\equiv (s\ \texttt{and}\ t)$ *iff* $w, L_0, L_1 \models\!\equiv (t\ \texttt{and}\ s)$.

□

6.2 Associativity

If "$\odot$" is a binary operator on a set A, then "$\odot$" is said to be *associative* if $a \odot (b \odot c) = (a \odot b) \odot c$ for all $a, b, c \in A$.

Associativity of "`intersect`"

The following proposition shows that "`intersect`" is associative at the level of *flow*.

Proposition 6.2.1.

$$\begin{aligned}
& X/(r\ \texttt{intersect}\ (s\ \texttt{intersect}\ t)) \\
= \;& X/((r\ \texttt{intersect}\ s)\ \texttt{intersect}\ t) \\
= \;& (X/r - (S_s \cup S_t)) \cup (X/s - (S_r \cup S_t)) \cup (X/t - (S_r \cup S_s)) \\
= \;& (X/r \cup X/s \cup X/t) - (B_r \cup B_s \cup B_t \cup (S_s \cap S_t) \cup (S_r \cap S_s) \cup (S_r \cap S_t))
\end{aligned}$$

Proof:

$$\begin{aligned}
& X/\texttt{(}r\ \texttt{intersect}\ \texttt{(}s\ \texttt{intersect}\ t\texttt{))}\\
= {} & \text{[Lemma 5.3]}\\
& (X/r - S_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}) \cup (X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)} - S_r)\\
= {} & \text{[Lemma 5.3]}\\
& (X/r-(S_s\cup S_t)) \cup (((X/s - S_t)\cup(X/t - S_s)) - S_r)\\
= {} & (X/r-(S_s\cup S_t))\cup(X/s-(S_r\cup S_t))\cup(X/t-(S_r\cup S_s))\\
= {} & (((X/r-S_s)\cup(X/s-S_r))-S_t)\cup(X/t-(S_r\cup S_s))\\
= {} & \text{[Lemma 5.3]}\\
& (X/\texttt{(}r\ \texttt{intersect}\ s\texttt{)} - S_t)\cup(X/t - S_{\texttt{(}r\ \texttt{intersect}\ s\texttt{)}})\\
= {} & \text{[Lemma 5.3]}\\
& X/\texttt{((}r\ \texttt{intersect}\ s\texttt{)}\ \texttt{intersect}\ t\texttt{)}
\end{aligned}$$

This proves the first two equalities.

$$\begin{aligned}
& X/\texttt{(}r\ \texttt{intersect}\ \texttt{(}s\ \texttt{intersect}\ t\texttt{))}\\
= {} & (X/r \cup X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)})\\
& - [B_r \cup B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup (S_r \cap S_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}})]\\
= {} & (X/r \cup ((X/s \cup X/t) - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}))\\
& - (B_r \cup B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup (S_r \cap S_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}))\\
= {} & (X/r \cup X/s \cup X/t) - (B_r \cup B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup (S_r \cap S_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}))\\
= {} & (X/r \cup X/s \cup X/t) - (B_r \cup B_s \cup B_t \cup (S_s \cap S_t) \cup (S_r \cap (S_s \cup S_t)))\\
= {} & (X/r \cup X/s \cup X/t) - (B_r \cup B_s \cup B_t \cup (S_s \cap S_t) \cup (S_r \cap S_s) \cup (S_r \cap S_t))
\end{aligned}$$

This proves the third equality. □

Theorem 6.2.2. *$w, L_0, L_1 \models \texttt{(}r\ \texttt{intersect}\ \texttt{(}s\ \texttt{intersect}\ t\texttt{))}$ iff there exist L_r, L_s, L_t such that*

$$w, L_0, L_r \models r, \quad w, L_0, L_s \models s, \quad w, L_0, L_t \models t$$

and

$$L_1 = L_r|_{D_r} \cup L_s|_{D_s} \cup L_t|_{D_t}$$

where

$$\begin{aligned}
D_r &= \mathrm{dom}(L_0)/r - (S_s \cup S_t)\\
D_s &= \mathrm{dom}(L_0)/s - (S_r \cup S_t)\\
D_t &= \mathrm{dom}(L_0)/t - (S_r \cup S_s)
\end{aligned}$$

Proof: Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \models \texttt{(}r\ \texttt{intersect}\ \texttt{(}s\ \texttt{intersect}\ t\texttt{))}$
iff [Lemma 5.3]
there exist L', L'' such that $w, L_0, L' \models r$ and $w, L_0, L'' \models \texttt{(}s\ \texttt{intersect}\ t\texttt{)}$,
and $L_1 = L'|_{D'} \cup L''|_{D''}$, where
$D' = X/r - S_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}$
$D'' = X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)} - S_r$
iff [definition of $w, L_0, L'' \models \texttt{(}s\ \texttt{intersect}\ t\texttt{)}$, Lemma 5.3]
there exist L', L'', L'_*, L''_* such that $w, L_0, L' \models r$ and $w, L_0, L'_* \models s$ and
$w, L_0, L''_* \models t$, and such that $L'' = L'_*|_{D'_*} \cup L''_*|_{D''_*}$, where
$D'_* = X/s - S_t$

$D''_* = X/t - S_s$,

and such that $L_1 = L'|_{D'} \cup L''|_{D''}$, where

$D' = X/r - S_{(s\ \texttt{intersect}\ t)}$

$D'' = X/(s\ \texttt{intersect}\ t) - S_r$

iff [L'' is determined by $L'' = L'_*|_{D'_*} \cup L''_*|_{D''_*}$]

there exist L', L'_*, L''_* such that $w, L_0, L' \models r$ and $w, L_0, L'_* \models s$ and $w, L_0, L''_* \models t$, and such that $L_1 = L'|_{D'} \cup (L'_*|_{D'_*} \cup L''_*|_{D''_*})|_{D''}$, where

$D'_* = X/s - S_t$

$D''_* = X/t - S_s$

$D' = X/r - S_{(s\ \texttt{intersect}\ t)}$

$D'' = X/(s\ \texttt{intersect}\ t) - S_r$

iff [let $L_r = L'$, $L_s = L'_*$, $L_t = L''_*$]

there exist L_r, L_s, L_t such that $w, L_0, L_r \models r$ and $w, L_0, L_s \models s$ and $w, L_0, L_t \models t$, and such that $L_1 = L_r|_{D'} \cup (L_s|_{D'_*} \cup L_t|_{D''_*})|_{D''}$, where

$D'_* = X/s - S_t$

$D''_* = X/t - S_s$

$D' = X/r - S_{(s\ \texttt{intersect}\ t)}$

$D'' = X/(s\ \texttt{intersect}\ t) - S_r$

iff [$(L_s|_{D'_*} \cup L_t|_{D''_*})|_{D''} = L_s|_{D'_* \cap D''} \cup L_t|_{D''_* \cap D''}$]

there exist L_r, L_s, L_t such that $w, L_0, L_r \models r$ and $w, L_0, L_s \models s$ and $w, L_0, L_t \models t$, and such that $L_1 = L_r|_{D'} \cup L_s|_{D'_* \cap D''} \cup L_t|_{D''_* \cap D''}$, where

$D'_* = X/s - S_t$

$D''_* = X/t - S_s$

$D' = X/r - S_{(s\ \texttt{intersect}\ t)}$

$D'' = X/(s\ \texttt{intersect}\ t) - S_r$

iff [let $D_r = D'$, $D_s = D'_* \cap D''$, $D_t = D''_* \cap D''$]

(A):

there exist L_r, L_s, L_t such that $w, L_0, L_r \models r$ and $w, L_0, L_s \models s$ and $w, L_0, L_t \models t$, and such that $L_1 = L_r|_{D_r} \cup L_s|_{D_s} \cup L_t|_{D_t}$, where

$$\begin{array}{rl} D_r = & X/r - S_{(s\ \texttt{intersect}\ t)} \\ D_s = & (X/s - S_t) \\ & \cap \\ & (X/(s\ \texttt{intersect}\ t) - S_r) \\ D_t = & (X/t - S_s) \\ & \cap \\ & (X/(s\ \texttt{intersect}\ t) - S_r) \end{array}$$

Note that

$$D_r = X/r - (S_s \cup S_t)$$

Also,

$$\begin{array}{rl} & X/(s\ \texttt{intersect}\ t) - S_r \\ = & ((X/s \cup X/t) - (B_s \cup B_t \cup (S_s \cap S_t))) - S_r \\ = & (X/s \cup X/t) - (B_s \cup B_t \cup S_r \cup (S_s \cap S_t)) \end{array}$$

Therefore,

$$\begin{array}{rl} & D_s \\ = & (X/s - S_t) \\ & \cap \end{array}$$

$$\begin{array}{rl}
 & ((X/s \cup X/t) - (B_s \cup B_t \cup S_r \cup (S_s \cap S_t))) \\
= & X/s - (B_s \cup B_t \cup S_r \cup S_t) \\
= & \text{[Theorem 4.5(2)]} \\
 & X/s - (B_s \cup S_r \cup S_t) \\
= & \text{[by Corollary 4.2(1), } X/s \cap B_s = \{\}\text{]} \\
 & X/s - (S_r \cup S_t)
\end{array}$$

Similarly,

$$D_t = X/t - (S_r \cup S_s)$$

Therefore, (A) holds iff there exist L_r, L_s, L_t such that $w, L_0, L_r \models\equiv r$, $w, L_0, L_s \models\equiv s$, $w, L_0, L_t \models\equiv t$ and $L_1 = L_r|_{D_r} \cup L_s|_{D_s} \cup L_t|_{D_t}$, where

$$D_r = X/r - (S_s \cup S_t)$$

$$D_s = X/s - (S_r \cup S_t)$$

$$D_t = X/t - (S_r \cup S_s)$$

□

The next corollary follows from the commutativity of "`intersect`" and the symmetry of the semantic characterization in Theorem 6.2.2. It shows that "`intersect`" is associative at the semantic level.

Corollary 6.2.3.

$$w, L_0, L_1 \models\equiv \texttt{(}r \texttt{ intersect (}s\texttt{ intersect }t\texttt{))}$$

iff

$$w, L_0, L_1 \models\equiv \texttt{((}r \texttt{ intersect }s\texttt{) intersect }t\texttt{)}$$

□

Associativity of "`or`"

The following proposition shows that "`or`" is associative at the level of *flow*.

Proposition 6.2.4.

$$X/\texttt{(}r\texttt{ or (}s\texttt{ or }t\texttt{))} = X/\texttt{((}r\texttt{ or }s\texttt{) or }t\texttt{)} = X/r \cap X/s \cap X/t$$

Proof:

$$\begin{array}{rl}
 & X/\texttt{(}r\texttt{ or (}s\texttt{ or }t\texttt{))} \\
= & X/r \cap X/\texttt{(}s\texttt{ or }t\texttt{)} \\
= & X/r \cap (X/s \cap X/t) \\
= & (X/r \cap X/s) \cap X/t \\
= & X/\texttt{(}r\texttt{ or }s\texttt{)} \cap X/t \\
= & X/\texttt{((}r\texttt{ or }s\texttt{) or }t\texttt{)}
\end{array}$$

□

Theorem 6.2.5. $w, L_0, L_1 \mathrel{|\!\!\equiv} (r\ \texttt{or}\ (s\ \texttt{or}\ t))$ *iff there exists* L *such that*

$$w, L_0, L \mathrel{|\!\!\equiv} r \quad \textit{or} \quad w, L_0, L \mathrel{|\!\!\equiv} s \quad \textit{or} \quad w, L_0, L \mathrel{|\!\!\equiv} t$$

and such that $L_1 = L|_D$*, where* $D = \mathrm{dom}(L_0)/r \cap \mathrm{dom}(L_0)/s \cap \mathrm{dom}(L_0)/t$*.*

Proof: Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \mathrel{|\!\!\equiv} (r\ \texttt{or}\ (s\ \texttt{or}\ t))$
iff there exists L such that either $w, L_0, L \mathrel{|\!\!\equiv} r$ or $w, L_0, L \mathrel{|\!\!\equiv} (s\ \texttt{or}\ t)$ and such that $L_1 = L|_D$, where $D = X/(r\ \texttt{or}\ (s\ \texttt{or}\ t))$
iff [Proposition 6.2.4]
there exists L such that either $w, L_0, L \mathrel{|\!\!\equiv} r$ or $w, L_0, L \mathrel{|\!\!\equiv} (s\ \texttt{or}\ t)$ and such that $L_1 = L|_D$, where $D = X/r \cap X/s \cap X/t$
iff [definition of $w, L_0, L \mathrel{|\!\!\equiv} (s\ \texttt{or}\ t)$]
there exists L such that
either
$w, L_0, L \mathrel{|\!\!\equiv} r$ and $L_1 = L|_D$, where $D = X/r \cap X/s \cap X/t$
or
there exists L' such that either $w, L_0, L' \mathrel{|\!\!\equiv} s$ or $w, L_0, L' \mathrel{|\!\!\equiv} t$
and such that $L_1 = L|_D$ and $L = L'|_{D'}$, where $D = X/r \cap X/s \cap X/t$ and $D' = X/(s\ \texttt{or}\ t) = X/s \cap X/t$
iff [in the second disjunct, L is determined and $L_1 = L'|_{D \cap D'} = L'|_D$]
either
there exists L such that $w, L_0, L \mathrel{|\!\!\equiv} r$ and $L_1 = L|_D$, where $D = X/r \cap X/s \cap X/t$
or
there exists L' such that either $w, L_0, L' \mathrel{|\!\!\equiv} s$ or $w, L_0, L' \mathrel{|\!\!\equiv} t$
and such that $L_1 = L'|_D$, where $D = X/r \cap X/s \cap X/t$
iff there exists L such that $L_1 = L|_D$, where $D = X/r \cap X/s \cap X/t$, and such that either $w, L_0, L \mathrel{|\!\!\equiv} r$ or $w, L_0, L \mathrel{|\!\!\equiv} s$ or $w, L_0, L \mathrel{|\!\!\equiv} t$

□

The next corollary follows from the commutativity of "`or`" and the symmetry of the semantic characterization in Theorem 6.2.5. It shows that "`or`" is associative at the semantic level.

Corollary 6.2.6.

$$w, L_0, L_1 \mathrel{|\!\!\equiv} (r\ \texttt{or}\ (s\ \texttt{or}\ t)) \quad \textit{iff} \quad w, L_0, L_1 \mathrel{|\!\!\equiv} ((r\ \texttt{or}\ s)\ \texttt{or}\ t)$$

□

Associativity of "`and`"

The "`and`" operator is derived according to

$(s\ \texttt{and}\ t)$
$\equiv (((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)\ \texttt{or}\ (s\ \texttt{intersect}\ (t\ \texttt{\#\#1 1[*0:\$]})))$

The following lemmas derive the local variable flow and semantics of tight satisfaction for "`and`".

Lemma 6.2.7.

1. $S_{\texttt{1[*0:\$]}} = \{\}$.

2. $S_{(s\ \texttt{\#\#1 1[*0:\$]})} = S_s$.

3. $S_{((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)} = S_s \cup S_t$.

4. $S_{(s\ \texttt{and}\ t)} = S_s \cup S_t$.

Proof:

$$\begin{array}{ll} \multicolumn{2}{l}{S_{\texttt{1[*0:\$]}}} \\ = & S_{(\texttt{1[*0] or 1[*1:\$]})} \\ = & S_{\texttt{1[*0]}} \cup S_{\texttt{1[*1:\$]}} \\ = & \{\} \cup S_{\texttt{1}} \\ = & \{\} \cup \{\} \\ = & \{\} \end{array}$$

This proves part 1.

$$\begin{array}{ll} \multicolumn{2}{l}{S_{(s\ \texttt{\#\#1 1[*0:\$]})}} \\ = & S_s \cup S_{\texttt{1[*0:\$]}} \\ = & [\text{part 1}] \\ & S_s \cup \{\} \\ = & S_s \end{array}$$

This proves part 2.

$$\begin{array}{ll} \multicolumn{2}{l}{S_{((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)}} \\ = & S_{(s\ \texttt{\#\#1 1[*0:\$]})} \cup S_t \\ = & [\text{part 2}] \\ & S_s \cup S_t \end{array}$$

This proves part 3.

$$\begin{array}{ll} \multicolumn{2}{l}{S_{(s\ \texttt{and}\ t)}} \\ = & S_{(((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)\ \texttt{or}\ (s\ \texttt{intersect}\ (t\ \texttt{\#\#1 1[*0:\$]})))} \\ = & S_{((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)} \cup S_{(s\ \texttt{intersect}\ (t\ \texttt{\#\#1 1[*0:\$]}))} \\ = & [\text{part 3, commutativity of ``}\texttt{intersect}\text{''}] \\ & (S_s \cup S_t) \cup (S_s \cup S_t) \\ = & S_s \cup S_t \end{array}$$

This proves part 4. □

Lemma 6.2.8.

1. $B_{\texttt{1[*0:\$]}} = \{\}$.

2. $B_{(s\ \texttt{\#\#1 1[*0:\$]})} = B_s$.

3. $B_{((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)} = B_{(s\ \texttt{intersect}\ t)}$.

4. $B_{(s\ \texttt{and}\ t)} = B_{(s\ \texttt{intersect}\ t)}$.

Proof:

$$\begin{aligned}
&B_{\texttt{1[*0:\$]}}\\
=\ & B_{(\texttt{1[*0] or 1[*1:\$]})}\\
=\ & B_{\texttt{1[*0]}} \cap B_{\texttt{1[*1:\$]}}\\
=\ & \{\} \cap B_{\texttt{1}}\\
=\ & \{\}
\end{aligned}$$

This proves part 1.

$$\begin{aligned}
&B_{(s\ \texttt{\#\#1 1[*0:\$]})}\\
=\ & (B_s - F_{\texttt{1[*0:\$]}}) \cup B_{\texttt{1[*0:\$]}}\\
=\ & \text{[part 1]}\\
& B_s - F_{\texttt{1[*0:\$]}}\\
=\ & B_s - F_{(\texttt{1[*0] or 1[*1:\$]})}\\
=\ & B_s - (F_{\texttt{1[*0]}} \cap F_{\texttt{1[*1:\$]}})\\
=\ & B_s - (\{\} \cap F_{\texttt{1}})\\
=\ & B_s - \{\}\\
=\ & B_s
\end{aligned}$$

This proves part 2.

$$\begin{aligned}
&B_{((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)}\\
=\ & B_{(s\ \texttt{\#\#1 1[*0:\$]})} \cup B_t \cup (S_{(s\ \texttt{\#\#1 1[*0:\$]})} \cap S_t)\\
=\ & \text{[part 2, Lemma 6.2.8(2)]}\\
& B_s \cup B_t \cup (S_s \cap S_t)\\
=\ & B_{(s\ \texttt{intersect}\ t)}
\end{aligned}$$

This proves part 3.

$$\begin{aligned}
&B_{(s\ \texttt{and}\ t)}\\
=\ & B_{(((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)\ \texttt{or}\ (s\ \texttt{intersect}\ (t\ \texttt{\#\#1 1[*0:\$]})))}\\
=\ & B_{((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t)} \cup B_{(s\ \texttt{intersect}\ (t\ \texttt{\#\#1 1[*0:\$]}))}\\
=\ & \text{[part 3, commutativity of ``}\texttt{intersect}\text{'']}\\
& B_{(s\ \texttt{intersect}\ t)} \cup B_{(s\ \texttt{intersect}\ t)}\\
=\ & B_{(s\ \texttt{intersect}\ t)}
\end{aligned}$$

This proves part 4. □

Lemma 6.2.9.

1. $X/\texttt{1[*0:\$]} = X$.

2. $X/(s\ \texttt{\#\#1 1[*0:\$]}) = X/s$.

3. $X/((s\ \texttt{\#\#1 1[*0:\$]})\ \texttt{intersect}\ t) = X/(s\ \texttt{intersect}\ t)$.

4. $X/(s\ \texttt{and}\ t) = X/(s\ \texttt{intersect}\ t)$.

Proof:

$$\begin{aligned}
&X/\texttt{1[*0:\$]}\\
=\ & X/(\texttt{1[*0] or 1[*1:\$]})\\
=\ & (X/\texttt{1[*0]}) \cap (X/\texttt{1[*1:\$]})\\
=\ & X \cap (X/\texttt{1})\\
=\ & X \cap X\\
=\ & X
\end{aligned}$$

This proves part 1.

$$\begin{aligned}
&X/\texttt{(}s\ \texttt{\#\#1 1[*0:\$])}\\
=\ &(X/s)/\texttt{1[*0:\$]}\\
=\ &[\text{part 1}]\\
&X/s
\end{aligned}$$

This proves part 2.

$$\begin{aligned}
&X/\texttt{((}s\ \texttt{\#\#1 1[*0:\$]) intersect}\ t\texttt{)}\\
=\ &(X/\texttt{(}s\ \texttt{\#\#1 1[*0:\$])} \cup X/t) - (B_{\texttt{(}s\ \texttt{\#\#1 1[*0:\$])}} \cup B_t \cup (S_{\texttt{(}s\ \texttt{\#\#1 1[*0:\$])}} \cap S_t))\\
=\ &[\text{Lemma 6.2.7(2), Lemma 6.2.8(2)}]\\
&(X/\texttt{(}s\ \texttt{\#\#1 1[*0:\$])} \cup X/t) - (B_s \cup B_t \cup (S_s \cap S_t))\\
=\ &[\text{part 2}]\\
&(X/s \cup X/t) - (B_s \cup B_t \cup (S_s \cap S_t))\\
=\ &X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)}
\end{aligned}$$

This proves part 3.

$$\begin{aligned}
&X/\texttt{(}s\ \texttt{and}\ t\texttt{)}\\
=\ &X/\texttt{((}s\ \texttt{intersect (}t\ \texttt{\#\#1 1[*0:\$])) or ((}s\ \texttt{\#\#1 1[*0:\$]) intersect}\ t\texttt{))}\\
=\ &X/\texttt{(}s\ \texttt{intersect (}t\ \texttt{\#\#1 1[*0:\$]))} \cap X/\texttt{((}s\ \texttt{\#\#1 1[*0:\$]) intersect}\ t\texttt{)}\\
=\ &[\text{part 3, commutativity of ``}\texttt{intersect}\text{''}]\\
&X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)} \cap X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)}\\
=\ &X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)}
\end{aligned}$$

This proves part 4. □

Lemma 6.2.9(4) and Proposition 6.2.1 yield the following corollary, which shows that "`and`" is associative at the level of *flow*.

Corollary 6.2.10.

$$\begin{aligned}
&X/\texttt{(}r\ \texttt{and (}s\ \texttt{and}\ t\texttt{))}\\
=\ &X/\texttt{((}r\ \texttt{and}\ s\texttt{) and}\ t\texttt{)}\\
=\ &(X/r - (S_s \cup S_t)) \cup (X/s - (S_r \cup S_t)) \cup (X/t - (S_r \cup S_s))\\
=\ &(X/r \cup X/s \cup X/t) - (B_r \cup B_s \cup B_t \cup (S_s \cap S_t) \cup (S_r \cap S_s) \cup (S_r \cap S_t))
\end{aligned}$$

□

Lemma 6.2.11.

0. $w, L_0, L_1 \models$ `1[*1:$]` *iff* $L_1 = L_0$, $|w| > 0$, *and no letter of* w *is* $\bot$.

1. $w, L_0, L_1 \models$ `1[*0:$]` *iff* $L_1 = L_0$ *and no letter of* w *is* $\bot$.

2. $w, L_0, L_1 \models \texttt{(}s\ \texttt{\#\#1 1[*0:\$])}$ *iff there exist* u, v *such that* $w = uv$ *and* $u, L_0, L_1 \models s$ *and no letter of* v *is* $\bot$.

3. $w, L_0, L_1 \models \texttt{((}s\ \texttt{\#\#1 1[*0:\$]) intersect}\ t\texttt{)}$ *iff there exist* u, v, L_s, L_t *such that* $w = uv$ *and* $u, L_0, L_s \models s$ *and* $w, L_0, L_t \models t$ *and* $L_1 = L_s|_{D_s} \cup L_t|_{D_t}$, *where*

$$\begin{aligned}
D_s &= \mathrm{dom}(L_0)/s - S_t\\
D_t &= \mathrm{dom}(L_0)/t - S_s
\end{aligned}$$

4. *$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{(}s\ \texttt{and}\ t\texttt{)}$ iff there exist $u_s, v_s, u_t, v_t, L_s, L_t$ such that $w = u_s v_s = u_t v_t$ and at least one of v_s, v_t is empty and $u_s, L_0, L_s \mathrel{|\!\!\equiv} s$ and $u_t, L_0, L_t \mathrel{|\!\!\equiv} t$ and $L_1 = L_s|_{D_s} \cup L_t|_{D_t}$, where*

$$D_s = \mathrm{dom}(L_0)/s - S_t$$
$$D_t = \mathrm{dom}(L_0)/t - S_s$$

Proof: Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{1[*1:\$]}$
iff there exist $L_{(0)} = L_0, w_1, L_{(1)}, w_2, L_{(2)}, \ldots, w_j, L_{(j)} = L_1$ $(j \geq 1)$ such that $w = w_1 \cdots w_j$ and for every i such that $1 \leq i \leq j$, $w_i, L_{(i-1)}, L_{(i)} \mathrel{|\!\!\equiv} \texttt{1}$
iff there exist $L_{(0)} = L_0, w_1, L_{(1)}, w_2, L_{(2)}, \ldots, w_j, L_{(j)} = L_1$ $(j \geq 1)$ such that $w = w_1 \cdots w_j$ and for every i such that $1 \leq i \leq j$, $|w_i| = 1$ and $w_i \neq \bot$ and $L_{(i-1)} = L_{(i)}$
iff $|w| > 0$ and no letter of w is $\bot$ and $L_0 = L_1$

This proves part 0.

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{1[*0:\$]}$
iff $w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{(1[*0] or 1[*1:\$])}$
iff there exists L such that both of the following hold:
– either $w, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*0]}$ or $w, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*1:\$]}$, and
– $L_1 = L|_D$, where $D = X/\texttt{(1[*0] or 1[*1:\$])}$
iff [proof of Lemma 6.2.9(1)]
there exists L such that both of the following hold:
– either $w, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*0]}$ or $w, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*1:\$]}$, and
– $L_1 = L|_X$
iff [part 0]
there exists L such that $L_1 = L|_X$ and either
– $|w| = 0$ and $L_0 = L$, or
– $|w| > 0$ and $L_0 = L$ and no letter of w is $\bot$
iff there exists L such that $L_1 = L|_X$ and $L = L_0$ and no letter of w is $\bot$
iff $L_1 = L_0$ and no letter of w is $\bot$

This proves part 1.

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{(}s\ \texttt{\#\#1 1[*0:\$])}$
iff there exist u, v, L such that $w = uv$ and $u, L_0, L \mathrel{|\!\!\equiv} s$ and $v, L, L_1 \mathrel{|\!\!\equiv} \texttt{1[*0:\$]}$
iff [part 1]
there exist u, v, L such that $w = uv$ and $u, L_0, L \mathrel{|\!\!\equiv} s$ and $L_1 = L$ and no letter of v is $\bot$
iff there exist u, v such that $w = uv$ and $u, L_0, L_1 \mathrel{|\!\!\equiv} s$ and no letter of v is $\bot$

This proves part 2.

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{((}s\ \texttt{\#\#1 1[*0:\$]) intersect}\ t\texttt{)}$
iff [Lemma 5.3]
there exist L_s, L_t such that $w, L_0, L_s \mathrel{|\!\!\equiv} \texttt{(}s\ \texttt{\#\#1 1[*0:\$])}$ and $w, L_0, L_t \mathrel{|\!\!\equiv} t$ and $L_1 = L_s|_{D_s} \cup L_t|_{D_t}$, where
$D_s = X/\texttt{(}s\ \texttt{\#\#1 1[*0:\$])} - S_t$

$D_t = X/t - S_{(s\ \texttt{\#\#1 1[*0:\$]})}$

iff [Lemma 6.2.7(2), Lemma 6.2.9(2)]

there exist L_s, L_t such that $w, L_0, L_s \mathrel{|}\!\!\equiv \texttt{(}s\ \texttt{\#\#1 1[*0:\$])}$ and $w, L_0, L_t \mathrel{|}\!\!\equiv t$ and $L_1 = L_s|_{D_s} \cup L_t|_{D_t}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

iff [part 2]

there exist u, v, L_s, L_t such that $w = uv$ and $u, L_0, L_s \mathrel{|}\!\!\equiv s$ and no letter of v is $\bot$ and $w, L_0, L_t \mathrel{|}\!\!\equiv t$ and $L_1 = L_s|_{D_s} \cup L_t|_{D_t}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

iff [by Lemma 5.1, if $w, L_0, L_t \mathrel{|}\!\!\equiv t$ then no letter of w is $\bot$]

there exist u, v, L_s, L_t such that $w = uv$ and $u, L_0, L_s \mathrel{|}\!\!\equiv s$ and $w, L_0, L_t \mathrel{|}\!\!\equiv t$ and $L_1 = L_s|_{D_s} \cup L_t|_{D_t}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

This proves part 3.

$w, L_0, L_1 \mathrel{|}\!\!\equiv \texttt{(}s\ \texttt{and}\ t\texttt{)}$

iff there exists L such that $L_1 = L|_D$, where $D = X/\texttt{(}s\ \texttt{and}\ t\texttt{)}$, and such that either

– $w, L_0, L \mathrel{|}\!\!\equiv \texttt{(}s\ \texttt{intersect (}t\ \texttt{\#\#1 1[*0:\$]))}$, or

– $w, L_0, L \mathrel{|}\!\!\equiv \texttt{((}s\ \texttt{\#\#1 1[*0:\$]) intersect}\ t\texttt{)}$

iff [part 3, commutativity of "`intersect`"]

there exists L such that $L_1 = L|_D$, where $D = X/\texttt{(}s\ \texttt{and}\ t\texttt{)}$, and such that either

– there exist u_t, v_t, L_s, L_t such that $w = u_t v_t$ and $w, L_0, L_s \mathrel{|}\!\!\equiv s$ and $u_t, L_0, L_t \mathrel{|}\!\!\equiv t$ and $L = L_s|_{D_s} \cup L_t|_{D_t}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

or

– there exist u_s, v_s, L_s, L_t such that $w = u_s v_s$ and $u_s, L_0, L_s \mathrel{|}\!\!\equiv s$ and $w, L_0, L_t \mathrel{|}\!\!\equiv t$ and $L = L_s|_{D_s} \cup L_t|_{D_t}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

iff there exists L such that $L_1 = L|_D$, where $D = X/\texttt{(}s\ \texttt{and}\ t\texttt{)}$, and such that there exist $u_s, v_s, u_t, v_t, L_s, L_t$ such that $w = u_s v_s = u_t v_t$ and at least one of v_s, v_t is empty and $u_s, L_0, L_s \mathrel{|}\!\!\equiv s$ and $u_t, L_0, L_t \mathrel{|}\!\!\equiv t$ and $L = L_s|_{D_s} \cup L_t|_{D_t}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

iff [L is determined by $L = L_s|_{D_s} \cup L_t|_{D_t}$]

(A):

there exist $u_s, v_s, u_t, v_t, L_s, L_t$ such that $w = u_s v_s = u_t v_t$ and at least one of v_s, v_t is empty and $u_s, L_0, L_s \mathrel{|}\!\!\equiv s$ and $u_t, L_0, L_t \mathrel{|}\!\!\equiv t$ and such that $L_1 = L_s|_{D_s \cap D} \cup L_t|_{D_t \cap D}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

$D = X/\texttt{(}s\ \texttt{and}\ t\texttt{)}$

By Lemma 6.2.9(4) and Lemma 5.3,

$$X/(s\ \texttt{and}\ t) = X/(s\ \texttt{intersect}\ t) = D_s \cup D_t$$

Therefore $D_s \cap D = D_s$ and $D_t \cap D = D_t$, and (A) holds iff there exist $u_s, v_s, u_t, v_t, L_s, L_t$ such that $w = u_s v_s = u_t v_t$ and at least one of v_s, v_t is empty and $u_s, L_0, L_s \models s$ and $u_t, L_0, L_t \models t$ and $L_1 = L_s|_{D_s} \cup L_t|_{D_t}$, where

$$\begin{aligned} D_s &= X/s - S_t \\ D_t &= X/t - S_s \end{aligned}$$

This proves part 4. □

Theorem 6.2.12. $w, L_0, L_1 \models (r\ \texttt{and}\ (s\ \texttt{and}\ t))$ *iff there exist* $u_r, v_r, u_s, v_s, u_t, v_t$ *and there exist* L_r, L_s, L_t *such that* $w = u_r v_r = u_s v_s = u_t v_t$ *and at least one of* v_r, v_s, v_t *is empty and*

$$u_r, L_0, L_r \models r, \quad u_s, L_0, L_s \models s, \quad u_t, L_0, L_t \models t$$

and

$$L_1 = L_r|_{D_r} \cup L_s|_{D_s} \cup L_t|_{D_t}$$

where

$$\begin{aligned} D_r &= \mathrm{dom}(L_0)/r - (S_s \cup S_t) \\ D_s &= \mathrm{dom}(L_0)/s - (S_r \cup S_t) \\ D_t &= \mathrm{dom}(L_0)/t - (S_r \cup S_s) \end{aligned}$$

Proof: Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \models (r\ \texttt{and}\ (s\ \texttt{and}\ t))$
iff [Lemma 6.2.11(4)]
there exist u_r, v_r, u, v, L_r, L such that $w = u_r v_r = uv$ and at least one of v_r, v is empty and $u_r, L_0, L_r \models r$ and $u, L_0, L \models (s\ \texttt{and}\ t)$,
and $L_1 = L_r|_{D_r} \cup L|_D$, where
$D_r = X/r - S_{(s\ \texttt{and}\ t)}$
$D = X/(s\ \texttt{and}\ t) - S_r$
iff [Lemma 6.2.11(4), applied to $u, L_0, L \models (s\ \texttt{and}\ t)$]
there exist u_r, v_r, u, v, L_r, L such that $w = u_r v_r = uv$ and at least one of v_r, v is empty and $u_r, L_0, L_r \models r$ and there exist $u_s, v'_s, u_t, v'_t, L_s, L_t$ such that $u = u_s v'_s = u_t v'_t$ and at least one of v'_s, v'_t is empty and $u_s, L_0, L_s \models s$ and $u_t, L_0, L_t \models t$ and $L = L_s|_{D'_s} \cup L_t|_{D'_t}$, where
$D'_s = X/s - S_t$
$D'_t = X/t - S_s$
and $L_1 = L_r|_{D_r} \cup L|_D$, where
$D_r = X/r - S_{(s\ \texttt{and}\ t)}$
$D = X/(s\ \texttt{and}\ t) - S_r$
iff [$u = u_s v'_s = u_t v'_t$ determines u; $L = L_s|_{D'_s} \cup L_t|_{D'_t}$ determines L]

there exist $u_r, v_r, u_s, v'_s, u_t, v'_t, v, L_r, L_s, L_t$ such that $w = u_r v_r = u_s v'_s v = u_t v'_t v$ and at least one of v_r, v is empty and at least one of v'_s, v'_t is empty and $u_r, L_0, L_r \models r$ and $u_s, L_0, L_s \models s$ and $u_t, L_0, L_t \models t$ and $L_1 = L_r|_{D_r} \cup L_s|_{D'_s \cap D} \cup L_t|_{D'_t \cap D}$, where

$D_r = X/r - S_{(s \texttt{ and } t)}$
$D'_s = X/s - S_t$
$D'_t = X/t - S_s$
$D = X/(s \texttt{ and } t) - S_r$

iff [let $v_s = v'_s v$, $v_t = v'_t v$; at least one of v_r, v_s, v_t is empty iff both at least one of v_r, v is empty and at least one of v'_s, v'_t is empty]
(A):
there exist $u_r, v_r, u_s, v_s, u_t, v_t, L_r, L_s, L_t$ such that $w = u_r v_r = u_s v_s = u_t v_t$ and at least one of v_r, v_s, v_t is empty and $u_r, L_0, L_r \models r$ and $u_s, L_0, L_s \models s$ and $u_t, L_0, L_t \models t$ and $L_1 = L_r|_{D_r} \cup L_s|_{D'_s \cap D} \cup L_t|_{D'_t \cap D}$, where

$D_r = X/r - S_{(s \texttt{ and } t)}$
$D'_s = X/s - S_t$
$D'_t = X/t - S_s$
$D = X/(s \texttt{ and } t) - S_r$

By Lemma 6.2.7(4),

$$D_r = X/r - (S_s \cup S_t)$$

Also,

$$\begin{aligned} & D \\ = \;& X/(s \texttt{ and } t) - S_r \\ = \;& [\text{Lemma 6.2.9(4)}] \\ & X/(s \texttt{ intersect } t) - S_r \\ = \;& (X/s \cup X/t) - (B_s \cup B_t \cup S_r \cup (S_s \cap S_t)) \end{aligned}$$

Therefore,

$$\begin{aligned} & D'_s \cap D \\ = \;& (X/s - S_t) \\ & \cap \\ & ((X/s \cup X/t) - (B_s \cup B_t \cup S_r \cup (S_s \cap S_t))) \\ = \;& [\text{proof of Theorem 6.2.2}] \\ & X/s - (S_r \cup S_t) \end{aligned}$$

Similarly,

$$D'_t \cap D = X/t - (S_r \cup S_s)$$

Therefore, letting $D_s = D'_s \cap D$ and $D_t = D'_t \cap D$, (A) holds iff there exist $u_r, v_r, u_s, v_s, u_t, v_t$ and there exist L_r, L_s, L_t such that $w = u_r v_r = u_s v_s = u_t v_t$ and at least one of v_r, v_s, v_t is empty and

$$u_r, L_0, L_r \models r, \quad u_s, L_0, L_s \models s, \quad u_t, L_0, L_t \models t$$

and

$$L_1 = L_r|_{D_r} \cup L_s|_{D_s} \cup L_t|_{D_t}$$

where

$$D_r = X/r - (S_s \cup S_t)$$
$$D_s = X/s - (S_r \cup S_t)$$
$$D_t = X/t - (S_r \cup S_s)$$

□

The next corollary follows from the commutativity of "`and`" and the symmetry of the semantic characterization in Theorem 6.2.12. It shows that "`and`" is associative at the semantic level.

Corollary 6.2.13.

$$w, L_0, L_1 \models\!\equiv \texttt{(}r\ \texttt{and (}s\ \texttt{and}\ t\texttt{))} \quad \textit{iff} \quad w, L_0, L_1 \models\!\equiv \texttt{((}r\ \texttt{and}\ s\texttt{) and}\ t\texttt{)}$$

□

Associativity of "`##1`", "`##0`"

The following proposition shows that each of "`##1`" and "`##0`" is associative at the level of *flow*.

Proposition 6.2.14.

1. $X/\texttt{(}r\ \texttt{\#\#1 (}s\ \texttt{\#\#1}\ t\texttt{))} = X/\texttt{((}r\ \texttt{\#\#1}\ s\texttt{) \#\#1}\ t\texttt{)} = ((X/r)/s)/t$.

2. $X/\texttt{(}r\ \texttt{\#\#0 (}s\ \texttt{\#\#0}\ t\texttt{))} = X/\texttt{((}r\ \texttt{\#\#0}\ s\texttt{) \#\#0}\ t\texttt{)} = ((X/r)/s)/t$.

Proof:

$$\begin{array}{ll} & X/\texttt{(}r\ \texttt{\#\#1 (}s\ \texttt{\#\#1}\ t\texttt{))} \\ = & (X/r)/\texttt{(}s\ \texttt{\#\#1}\ t\texttt{)} \\ = & ((X/r)/s)/t \\ = & (X/\texttt{(}r\ \texttt{\#\#1}\ s\texttt{)})/t \\ = & X/\texttt{((}r\ \texttt{\#\#1}\ s\texttt{) \#\#1}\ t\texttt{)} \end{array}$$

The argument for `##0` runs similarly. □

The following proposition shows that each of "`##1`" and "`##0`" is associative at the semantic level.

Proposition 6.2.15.

1. $w, L_0, L_1 \models\!\equiv \texttt{(}r\ \texttt{\#\#1 (}s\ \texttt{\#\#1}\ t\texttt{))}$ *iff* $w, L_0, L_1 \models\!\equiv \texttt{((}r\ \texttt{\#\#1}\ s\texttt{) \#\#1}\ t\texttt{)}$.

2. $w, L_0, L_1 \models\!\equiv \texttt{(}r\ \texttt{\#\#0 (}s\ \texttt{\#\#0}\ t\texttt{))}$ *iff* $w, L_0, L_1 \models\!\equiv \texttt{((}r\ \texttt{\#\#0}\ s\texttt{) \#\#0}\ t\texttt{)}$.

Proof:

$w, L_0, L_1 \models$ `(`r `##1 (`s `##1` t`))`
iff there exist x, u, L_r such that $w = xu$ and $x, L_0, L_r \models r$ and
$u, L_r, L_1 \models$ `(`s `##1` t`)`
iff there exist x, u, L_r such that $w = xu$ and $x, L_0, L_r \models r$ and there exist
y, z, L_s such that $u = yz$ and $y, L_r, L_s \models s$ and $z, L_s, L_1 \models t$
iff [u is determined by $u = yz$]
there exist x, y, z, L_r, L_s such that $w = xyz$ and $x, L_0, L_r \models r$ and
$y, L_r, L_s \models s$ and $z, L_s, L_1 \models t$
iff [let $v = xy$]
there exist v, z, L_s such that $w = vz$ and $v, L_0, L_s \models$ `(`r `##1` s`)` and
$z, L_s, L_1 \models t$
iff $w, L_0, L_1 \models$ `((`r `##1` s`) ##1` t`)`

The argument for `##0` runs similarly. □

Remark: In general, "`##1`" and "`##0`" do not associate with one another. For example,

$w \models$ `((1 ##1 1[*0]) ##0 1)`

iff $|w| = 1$ and $w^0 \neq \bot$. However, there does not exist w such that

$w \models$ `(1 ##1 (1[*0] ##0 1))`

□

6.3 Idempotent elements

If "$\odot$" is a binary operator on a set A, then an element $a \in A$ is said to be *idempotent* with respect to "$\odot$" if $a \odot a = a$.

The following proposition shows that every sequence is idempotent with respect to "`or`" at the level of *flow* and at the semantic level.

Proposition 6.3.1.

1. $X/$`(`r `or` r`)` $= X/r$.

2. $w, L_0, L_1 \models$ `(`r `or` r`)` *iff* $w, L_0, L_1 \models r$.

Proof:

$$X/(r \texttt{ or } r) = X/r \cap X/r = X/r$$

This proves part 1. Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \models$ `(`r `or` r`)`
iff there exists L such that $L_1 = L|_D$, where $D = X/$`(`r `or` r`)`, and such that
$w, L_0, L \models r$
iff [part 1]
there exists L such that $L_1 = L|_{X/r}$ and such that $w, L_0, L \models r$
iff [by Theorem 5.4, if $w, L_0, L \models r$, then $\mathrm{dom}(L) = X/r$]
$w, L_0, L_1 \models r$

This proves part 2. $\square$

One should not expect a sequence with local variable assignments to be idempotent with respect to "`intersect`" or "`and`" at the level of *flow* or at the semantic level. Local variables that are assigned in both operands of "`intersect`" are not allowed to flow out from the intersection. Indeed,

$$\mathit{sample}(r) \subseteq \mathit{block}(\texttt{(}r \texttt{ intersect } r\texttt{)})$$

The definitions of *block* and *flow* for "`intersect`" are designed to eliminate inconsistently assigned values coming from the two operands in a way that is simple, that suits generic operands, and that does not require detailed analysis of the operands. From this point of view, placing the same sequence into each operand is not generic, and accommodating the coincidence of the operands of "`intersect`" or "`and`" is not something the definitions are designed to do.

The following proposition shows that if r has no local variable assignment, then it is idempotent with respect to "`intersect`" and "`and`" at the level of *flow* and at the semantic level.

Proposition 6.3.2. *Let* $S_r = \{\}$.

1. $X/\texttt{(}r \texttt{ intersect } r\texttt{)} = X/\texttt{(}r \texttt{ and } r\texttt{)} = X/r$.

2. $w, L_0, L_1 \models\!\equiv \texttt{(}r \texttt{ intersect } r\texttt{)}$ *iff* $w, L_0, L_1 \models\!\equiv r$.

3. $w, L_0, L_1 \models\!\equiv \texttt{(}r \texttt{ and } r\texttt{)}$ *iff* $w, L_0, L_1 \models\!\equiv r$.

Proof: By Lemma 6.2.9(4), $X/\texttt{(}r \texttt{ and } r\texttt{)} = X/\texttt{(}r \texttt{ intersect } r\texttt{)}$.

$X/\texttt{(}r \texttt{ intersect } r\texttt{)}$
$=$ [Lemma 5.3]
$(X/r - S_r) \cup (X/r - S_r)$
$=$ $X/r - S_r$
$=$ $[S_r = \{\}]$
X/r

This proves part 1. Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \models\!\equiv \texttt{(}r \texttt{ intersect } r\texttt{)}$
iff [Lemma 5.3]
there exist L', L'' such that $w, L_0, L' \models\!\equiv r$ and $w, L_0, L'' \models\!\equiv r$ and
$L_1 = L'|_{D'} \cup L''|_{D''}$, where
$D' = X/r - S_r$
$D'' = X/r - S_r$
iff $[S_r = \{\}]$
there exist L', L'' such that $w, L_0, L' \models\!\equiv r$ and $w, L_0, L'' \models\!\equiv r$ and
$L_1 = L'|_{X/r} \cup L''|_{X/r}$
iff [by Theorem 5.4, if $w, L_0, L' \models\!\equiv r$, then $\mathrm{dom}(L') = X/r$; similarly with L'']
$w, L_0, L_1 \models\!\equiv r$

This proves part 2.

$w, L_0, L_1 \mathrel{|}\!\equiv \texttt{(}r\ \texttt{and}\ r\texttt{)}$

iff [Lemma 6.2.11(4)]

there exist u, v, u', v', L, L' such that $w = uv = u'v'$ and at least one of v, v' is empty and $u, L_0, L' \mathrel{|}\!\equiv r$ and $u', L_0, L' \mathrel{|}\!\equiv r$ and $L_1 = L|_D \cup L'|_{D'}$, where

$D = X/r - S_r$

$D' = X/r - S_r$

iff $[S_r = \{\}]$

there exist u, v, u', v', L, L' such that $w = uv = u'v'$ and at least one of v, v' is empty and $u, L_0, L' \mathrel{|}\!\equiv r$ and $u', L_0, L' \mathrel{|}\!\equiv r$ and $L_1 = L|_{X/r} \cup L'|_{X/r}$

iff [by Theorem 5.4, if $w, L_0, L \mathrel{|}\!\equiv r$, then $\mathrm{dom}(L) = X/r$; similarly with L']

there exist u, v, u', v' such that $w = uv = u'v'$ and at least one of v, v' is empty and $u, L_0, L_1 \mathrel{|}\!\equiv r$ and $u', L_0, L_1 \mathrel{|}\!\equiv r$

iff [if v is empty, then $w = u$; similarly with v']

$w, L_0, L_1 \mathrel{|}\!\equiv r$

This proves part 3. □

6.4 Distributivity

If "$\odot$" and "$\oplus$" are commutative binary operators on a set A, then "$\odot$" is said to *distribute over* "$\oplus$" if $a \odot (b \oplus c) = (a \odot b) \oplus (a \odot c)$ for all $a, b, c \in A$.

The following proposition shows that "`intersect`" distributes over "`or`" at the level of *flow*.

Proposition 6.4.1.

$$\begin{aligned}
&(X/r - (S_s \cup S_t)) \cup ((X/s \cap X/t) - S_r)\\
&= X/\texttt{(}r\ \texttt{intersect (}s\ \texttt{or}\ t\texttt{))}\\
&= X/\texttt{((}r\ \texttt{intersect}\ s\texttt{) or (}r\ \texttt{intersect}\ t\texttt{))}\\
&= ((X/r - S_s) \cup (X/s - S_r)) \cap ((X/r - S_t) \cup (X/t - S_r))
\end{aligned}$$

Proof: The first and last equalities follow by using Lemma 5.3 and the definitions.

$$\begin{aligned}
&X/\texttt{(}r\ \texttt{intersect (}s\ \texttt{or}\ t\texttt{))}\\
=\ &(X/r \cup X/\texttt{(}s\ \texttt{or}\ t\texttt{)}) - (B_r \cup B_{\texttt{(}s\ \texttt{or}\ t\texttt{)}} \cup (S_r \cap S_{\texttt{(}s\ \texttt{or}\ t\texttt{)}}))\\
=\ &(X/r \cup (X/s \cap X/t)) - (B_r \cup B_s \cup B_t \cup (S_r \cap (S_s \cup S_t)))\\
=\ &((X/r \cup X/s) \cap (X/r \cup X/t)) - (B_r \cup B_s \cup B_t \cup (S_r \cap S_s) \cup (S_r \cap S_t))\\
=\ &((X/r \cup X/s) - (B_r \cup B_s \cup (S_r \cap S_s))\\
&\cap ((X/r \cup X/t) - (B_r \cup B_t \cup (S_r \cap S_t))\\
=\ &X/\texttt{(}r\ \texttt{intersect}\ s\texttt{)} \cap X/\texttt{(}r\ \texttt{intersect}\ s\texttt{)}\\
=\ &X/\texttt{((}r\ \texttt{intersect}\ s\texttt{) or (}r\ \texttt{intersect}\ t\texttt{))}
\end{aligned}$$

This proves the middle equality. □

Using Lemma 6.2.9(4), Proposition 6.4.1 gives the following corollary, which shows that "`and`" distributes over "`or`" at the level of *flow*.

Corollary 6.4.2.

$$\begin{aligned}
&(X/r - (S_s \cup S_t)) \cup ((X/s \cap X/t) - S_r) \\
&= X/(r \texttt{ and } (s \texttt{ or } t)) \\
&= X/((r \texttt{ and } s) \texttt{ or } (r \texttt{ and } t)) \\
&= ((X/r - S_s) \cup (X/s - S_r)) \cap ((X/r - S_t) \cup (X/t - S_r))
\end{aligned}$$

$\square$

Lemma 6.4.3. *$X/r - S_r = X - S_r$.*

Proof:

$$\begin{aligned}
& X/r - S_r \\
= \;& \text{[Theorem 4.1]} \\
& (X \cup F_r) - (B_r \cup S_r) \\
= \;& \text{[by Theorem 4.5, } F_r \cup B_r \subseteq S_r\text{]} \\
& X - S_r
\end{aligned}$$

$\square$

Lemma 6.4.4. *Let f, g be functions, and let A, B, A', B' be sets such that $A \cup A' \subseteq \operatorname{dom}(f)$, $B \cup B' \subseteq \operatorname{dom}(g)$, and $A \cup B = A' \cup B'$. Then the following conditions are equivalent:*

1. *$f|_A \cup g|_B$ and $f|_{A'} \cup g|_{B'}$ are both functions and are equal to one another.*
2. *For all $x \in (A \cup A') \cap (B \cup B')$, $f(x) = g(x)$.*

Proof: Note that since $A \subseteq \operatorname{dom}(f)$, $\operatorname{dom}(f|_A) = A$. Similarly, $\operatorname{dom}(f|_{A'}) = A'$, $\operatorname{dom}(g|_B) = B$, and $\operatorname{dom}(g|_{B'}) = B'$. Note also that

$$(A \cup A') \cap (B \cup B') = (A \cap B) \cup (A' \cap B') \cup (A \cap B') \cup (A' \cap B)$$

Assume 1. Since $f|_A \cup g|_B$ is a function, $f(x) = g(x)$ for all $x \in A \cap B$. Since $f|_{A'} \cup g|_{B'}$ is a function, $f(x) = g(x)$ for all $x \in A' \cap B'$. If $x \in A \cap B'$, then

$$f(x) = (f|_A \cup g|_B)(x) = (f|_{A'} \cup g|_{B'})(x) = g(x)$$

Similarly, if $x \in B \cap A'$, then

$$g(x) = (f|_A \cup g|_B)(x) = (f|_{A'} \cup g|_{B'})(x) = f(x)$$

Assume 2. Since $f(x) = g(x)$ for all $x \in A \cap B$, $f|_A \cup g|_B$ is a function. Since $f(x) = g(x)$ for all $x \in A' \cap B'$, $f|_{A'} \cup g|_{B'}$ is a function. If $x \in A \cap A'$, then

$$(f|_A \cup g|_B)(x) = f(x) = (f|_{A'} \cup g|_{B'})(x)$$

If $x \in B \cap B'$, then

$$(f|_A \cup g|_B)(x) = g(x) = (f|_{A'} \cup g|_{B'})(x)$$

If $x \in A \cap B'$, then

$$(f|_A \cup g|_B)(x) = f(x) = g(x) = (f|_{A'} \cup g|_{B'})(x)$$

If $x \in B \cap A'$, then

$$(f|_A \cup g|_B)(x) = g(x) = f(x) = (f|_{A'} \cup g|_{B'})(x)$$

$\square$

The following theorem shows that "`intersect`" distributes over "`or`" at the semantic level.

Theorem 6.4.5.

$$w, L_0, L_1 \models \texttt{(}r \texttt{ intersect (}s \texttt{ or } t\texttt{))}$$

iff

$$w, L_0, L_1 \models \texttt{((}r \texttt{ intersect } s\texttt{) or (}r \texttt{ intersect } t\texttt{))}$$

Proof: Note that

$w, L_0, L_1 \models \texttt{(}r \texttt{ intersect (}s \texttt{ or } t\texttt{))}$
iff [Lemma 5.3]
there exist L_r, L such that $w, L_0, L_r \models r$ and $w, L_0, L \models \texttt{(}s \texttt{ or } t\texttt{)}$ and $L_1 = L_r|_{D_{r,st}} \cup L|_{D_{st,r}}$, where
$D_{r,st} = X/r - S_{\texttt{(}s \texttt{ or } t\texttt{)}} = X/r - (S_s \cup S_r)$
$D_{st,r} = X/\texttt{(}s \texttt{ or } t\texttt{)} - S_r = (X/s \cap X/t) - S_r$
iff there exist L_r, L, L' such that $w, L_0, L_r \models r$ and such that either $w, L_0, L' \models s$ or $w, L_0, L' \models t$ and such that $L_1 = L_r|_{D_{r,st}} \cup L|_{D_{st,r}}$ and $L = L'|_{D'}$, where
$D_{r,st} = X/r - (S_s \cup S_t)$
$D_{st,r} = (X/s \cap X/t) - S_r$
$D' = X/\texttt{(}s \texttt{ or } t\texttt{)} = X/s \cap X/t$
iff [L is determined by $L = L'|_{D'}$]
there exist L_r, L' such that $w, L_0, L_r \models r$ and such that either $w, L_0, L' \models s$ or $w, L_0, L' \models t$ and such that $L_1 = L_r|_{D_{r,st}} \cup L'|_{D_{st,r} \cap D'}$, where
$D_{r,st} = X/r - (S_s \cup S_t)$
$D_{st,r} = (X/s \cap X/t) - S_r$
$D' = X/s \cap X/t$
iff [$D_{st,r} \subseteq D'$]
(A):
there exist L_r, L' such that $w, L_0, L_r \models r$ and such that either $w, L_0, L' \models s$ or $w, L_0, L' \models t$ and such that $L_1 = L_r|_{D_{r,st}} \cup L'|_{D_{st,r}}$, where
$D_{r,st} = X/r - (S_s \cup S_t)$
$D_{st,r} = (X/s \cap X/t) - S_r$

Note also that

$w, L_0, L_1 \models \texttt{((}r \texttt{ intersect } s\texttt{) or (}r \texttt{ intersect } t\texttt{))}$
iff there exists L'' such that $L_1 = L''|_{D''}$, where
$D'' = X/\texttt{((}r \texttt{ intersect } s\texttt{) or (}r \texttt{ intersect } t\texttt{))}$

and such that either
$w, L_0, L'' \mathrel{|}\!\!\equiv (r\ \texttt{intersect}\ s)$
or
$w, L_0, L'' \mathrel{|}\!\!\equiv (r\ \texttt{intersect}\ t)$
iff [Lemma 5.3]
there exists L'' such that $L_1 = L''|_{D''}$, where
$D'' = X/((r\ \texttt{intersect}\ s)\ \texttt{or}\ (r\ \texttt{intersect}\ t))$
and such that either
there exist L_r, L_s such that $w, L_0, L_r \mathrel{|}\!\!\equiv r$ and $w, L_0, L_s \mathrel{|}\!\!\equiv s$ and
$L'' = L_r|_{D_{r,s}} \cup L_s|_{D_{s,r}}$, where
$D_{r,s} = X/r - S_s$
$D_{s,r} = X/s - S_r$
or
there exist L_r, L_t such that $w, L_0, L_r \mathrel{|}\!\!\equiv r$ and $w, L_0, L_t \mathrel{|}\!\!\equiv t$ and
$L'' = L_r|_{D_{r,t}} \cup L_t|_{D_{t,r}}$, where
$D_{r,t} = X/r - S_t$
$D_{t,r} = X/t - S_r$
iff [L'' is determined by $L'' = L_r|_{D_{r,s}} \cup L_s|_{D_{s,r}}$ or by $L'' = L_r|_{D_{r,t}} \cup L_t|_{D_{t,r}}$]
(B):
letting
$D'' = X/((r\ \texttt{intersect}\ s)\ \texttt{or}\ (r\ \texttt{intersect}\ t))$,
there exists L_r such that $w, L_0, L_r \mathrel{|}\!\!\equiv r$ and such that either
there exists L_s such that $w, L_0, L_s \mathrel{|}\!\!\equiv s$ and $L_1 = L_r|_{D_{r,s} \cap D''} \cup L_s|_{D_{s,r} \cap D''}$,
where
$D_{r,s} = X/r - S_s$
$D_{s,r} = X/s - S_r$
or
there exists L_t such that $w, L_0, L_t \mathrel{|}\!\!\equiv t$ and $L_1 = L_r|_{D_{r,t} \cap D''} \cup L_t|_{D_{t,r} \cap D''}$,
where
$D_{r,t} = X/r - S_t$
$D_{t,r} = X/t - S_r$

Note that by Proposition 6.4.1,

$$D_{r,st} \cup D_{st,r} = D'' = (D_{r,s} \cup D_{s,r}) \cap (D_{r,t} \cup D_{t,r}) \tag{$\dagger$}$$

Note also that

$$D_{r,st} = D_{r,s} \cap D_{r,t} \quad \text{and} \quad D_{st,r} = D_{s,r} \cap D_{t,r} \tag{$*$}$$

Assume (A). Assume that $w, L_0, L' \mathrel{|}\!\!\equiv s$. This assumption is without loss of generality since a symmetric argument can be given with t replacing s. Let $L_s = L'$. To show (B), it suffices to show that

$$L_r|_{D_{r,st}} \cup L'|_{D_{st,r}} = L_r|_{D_{r,s} \cap D''} \cup L'|_{D_{s,r} \cap D''}$$

Theorem 5.4 implies that $D_{r,st} \subseteq \mathrm{dom}(L_r)$, $D_{r,s} \subseteq \mathrm{dom}(L_r)$, $D_{st,r} \subseteq \mathrm{dom}(L')$, and $D_{s,r} \subseteq \mathrm{dom}(L')$. Also

$(D_{r,s} \cap D'') \cup (D_{s,r} \cap D'')$
$= \ (D_{r,s} \cup D_{s,r}) \cap D''$

$$\begin{aligned}
&= [\text{by } (\dagger)] \\
&\quad D'' \\
&= [\text{by } (\dagger)] \\
&\quad D_{r,st} \cup D_{st,r}
\end{aligned}$$

Following the notation of Lemma 6.4.4, let $A = D_{r,st}$, $B = D_{st,r}$, $A' = D_{r,s} \cap D''$, and $B' = D_{s,r} \cap D''$. Notice that

$$\begin{aligned}
&A \cup A' \\
&= D_{r,st} \cup (D_{r,s} \cap D'') \\
&= (D_{r,st} \cup D_{r,s}) \cap (D_{r,st} \cup D'') \\
&= [\text{by } (*),\ (\dagger)] \\
&\quad D_{r,s} \cap D'' \\
&= A'
\end{aligned}$$

Similarly,

$$B \cup B' = D_{s,r} \cap D'' = B'$$

Therefore, by Lemma 6.4.4 it suffices to prove that L_r and L' agree on the set

$$\begin{aligned}
&A' \cap B' \\
&= (D_{r,s} \cap D'') \cap (D_{s,r} \cap D'') \\
&= [\text{by } (\dagger)] \\
&\quad D_{r,s} \cap D_{s,r} \cap (D_{r,st} \cup D_{st,r}) \\
&= (D_{r,s} \cap D_{s,r} \cap D_{r,st}) \cup (D_{r,s} \cap D_{s,r} \cap D_{st,r}) \\
&= [\text{by } (*)] \\
&\quad (D_{s,r} \cap D_{r,st}) \cup (D_{r,s} \cap D_{st,r}) \\
&= ((X/s - S_r) \cap (X/r - (S_s \cup S_t))) \cup ((X/r - S_s) \cap ((X/s \cap X/t) - S_r)) \\
&= ((X/r \cap X/s) - (S_r \cup S_s \cup S_t)) \cup ((X/r \cap X/s \cap X/t) - (S_r \cup S_s)) \\
&\subseteq (X/r \cap X/s) - (S_r \cup S_s) \\
&= (X/r - S_r) \cap (X/s - S_s) \\
&= [\text{Lemma } 6.4.3] \\
&\quad (X - S_r) \cap (X - S_s) \\
&= X - (S_r \cup S_s)
\end{aligned}$$

By Theorem 5.5, if $v \in X - (S_r \cup S_s)$, then $L_r(v) = L_0(v) = L'(v)$, as desired.

Now assume (B). Assume that there exists L_s such that $w, L_0, L_s \mathrel{|}\!\!\equiv s$ and $L_1 = L_r|_{D_{r,s} \cap D''} \cup L_s|_{D_{s,r} \cap D''}$, where

$$\begin{aligned}
D_{r,s} &= X/r - S_s \\
D_{s,r} &= X/s - S_r
\end{aligned}$$

This assumption is without loss of generality since a symmetric argument can be given with t replacing s. Let $L' = L_s$. To show (A), it suffices to show that

$$L_r|_{D_{r,st}} \cup L_s|_{D_{st,r}} = L_r|_{D_{r,s} \cap D''} \cup L_s|_{D_{s,r} \cap D''}$$

Theorem 5.4 implies that $D_{r,st} \subseteq \mathrm{dom}(L_r)$, $D_{r,s} \subseteq \mathrm{dom}(L_r)$, $D_{st,r} \subseteq \mathrm{dom}(L_s)$, and $D_{s,r} \subseteq \mathrm{dom}(L_s)$. As above,

$$(D_{r,s} \cap D'') \cup (D_{s,r} \cap D'') = D_{r,st} \cup D_{st,r}$$

Defining A, B, A', B' as above, it follows from Lemma 6.4.4 that it is enough to show that L_r and L_s agree on the set $(A \cup A') \cap (B \cup B')$. From above, $A \cup A' = A'$, $B \cup B' = B'$, and

$$A' \cap B' \subseteq X - (S_r \cup S_s)$$

By Theorem 5.5, if $v \in X - (S_r \cup S_s)$, then $L_r(v) = L_0(v) = L_s(v)$, as desired. $\square$

The following theorem shows that "`and`" distributes over "`or`" at the semantic level. The proof is very similar to that of Theorem 6.4.5. There is a bit more bookkeeping due to the use of Lemma 6.2.11(4) instead of Lemma 5.3.

Theorem 6.4.6.

$$w, L_0, L_1 \models \texttt{(}r \texttt{ and (}s \texttt{ or } t\texttt{))}$$

iff

$$w, L_0, L_1 \models \texttt{((}r \texttt{ and } s\texttt{) or (}r \texttt{ and } t\texttt{))}$$

Proof: Note that

$w, L_0, L_1 \models \texttt{(}r \texttt{ and (}s \texttt{ or } t\texttt{))}$
iff [Lemma 6.2.11(4)]
there exist u_r, v_r, u, v, L_r, L such that $w = u_r v_r = uv$, where at least one of v_r, v is empty, and such that $u_r, L_0, L_r \models r$ and $u, L_0, L \models \texttt{(}s \texttt{ or } t\texttt{)}$ and $L_1 = L_r|_{D_{r,st}} \cup L|_{D_{st,r}}$, where
$D_{r,st} = X/r - S_{\texttt{(}s \texttt{ or } t\texttt{)}} = X/r - (S_s \cup S_r)$
$D_{st,r} = X/\texttt{(}s \texttt{ or } t\texttt{)} - S_r = (X/s \cap X/t) - S_r$
iff there exist $u_r, v_r, u, v, L_r, L, L'$ such that $w = u_r v_r = uv$, where at least one of v_r, v is empty, and such that $u_r, L_0, L_r \models r$ and such that either $u, L_0, L' \models s$ or $u, L_0, L' \models t$ and such that $L_1 = L_r|_{D_{r,st}} \cup L|_{D_{st,r}}$ and $L = L'|_{D'}$, where
$D_{r,st} = X/r - (S_s \cup S_t)$
$D_{st,r} = (X/s \cap X/t) - S_r$
$D' = X/\texttt{(}s \texttt{ or } t\texttt{)} = X/s \cap X/t$
iff [L is determined by $L = L'|_{D'}$]
there exist u_r, v_r, u, v, L_r, L' such that $w = u_r v_r = uv$, where at least one of v_r, v is empty, and such that $u_r, L_0, L_r \models r$ and such that either $u, L_0, L' \models s$ or $u, L_0, L' \models t$ and such that $L_1 = L_r|_{D_{r,st}} \cup L'|_{D_{st,r} \cap D'}$, where
$D_{r,st} = X/r - (S_s \cup S_t)$
$D_{st,r} = (X/s \cap X/t) - S_r$
$D' = X/s \cap X/t$
iff [$D_{st,r} \subseteq D'$]
(A):
there exist u_r, v_r, u, v, L_r, L' such that $w = u_r v_r = uv$, where at least one of v_r, v is empty, and such that $u_r, L_0, L_r \models r$ and such that either $u, L_0, L' \models s$ or $u, L_0, L' \models t$ and such that $L_1 = L_r|_{D_{r,st}} \cup L'|_{D_{st,r}}$, where
$D_{r,st} = X/r - (S_s \cup S_t)$
$D_{st,r} = (X/s \cap X/t) - S_r$

Note also that

$w, L_0, L_1 \mathrel{|}\!\equiv \texttt{((}r\ \texttt{and}\ s\texttt{) or (}r\ \texttt{and}\ t\texttt{))}$
iff there exists L'' such that $L_1 = L''|_{D''}$, where
$D'' = X/\texttt{((}r\ \texttt{and}\ s\texttt{) or (}r\ \texttt{and}\ t\texttt{))}$
and such that either
$w, L_0, L'' \mathrel{|}\!\equiv \texttt{(}r\ \texttt{and}\ s\texttt{)}$
or
$w, L_0, L'' \mathrel{|}\!\equiv \texttt{(}r\ \texttt{and}\ t\texttt{)}$
iff [Lemma 6.2.11(4)]
there exists L'' such that $L_1 = L''|_{D''}$, where
$D'' = X/\texttt{((}r\ \texttt{and}\ s\texttt{) or (}r\ \texttt{and}\ t\texttt{))}$
and such that either
there exist $u_r, v_r, u_s, v_s, L_r, L_s$ such that $w = u_r v_r = u_s v_s$, where at least one of v_r, v_s is empty, and such that $u_r, L_0, L_r \mathrel{|}\!\equiv r$ and $u_s, L_0, L_s \mathrel{|}\!\equiv s$ and $L'' = L_r|_{D_{r,s}} \cup L_s|_{D_{s,r}}$, where
$D_{r,s} = X/r - S_s$
$D_{s,r} = X/s - S_r$
or
there exist $u_r, v_r, u_t, v_t, L_r, L_t$ such that $w = u_r v_r = u_t v_t$, where at least one of v_r, v_t is empty, and such that $u_r, L_0, L_r \mathrel{|}\!\equiv r$ and $u_t, L_0, L_t \mathrel{|}\!\equiv t$ and $L'' = L_r|_{D_{r,t}} \cup L_t|_{D_{t,r}}$, where
$D_{r,t} = X/r - S_t$
$D_{t,r} = X/t - S_r$
iff [L'' is determined by $L'' = L_r|_{D_{r,s}} \cup L_s|_{D_{s,r}}$ or by $L'' = L_r|_{D_{r,t}} \cup L_t|_{D_{t,r}}$]
(B):
letting
$D'' = X/\texttt{((}r\ \texttt{and}\ s\texttt{) or (}r\ \texttt{and}\ t\texttt{))}$,
there exist u_r, v_r, u, v, L_r such that $w = u_r v_r = uv$, where at least one of v_r, v is empty, and such that $u, L_0, L_r \mathrel{|}\!\equiv r$ and such that either
there exists L_s such that $u, L_0, L_s \mathrel{|}\!\equiv s$ and $L_1 = L_r|_{D_{r,s} \cap D''} \cup L_s|_{D_{s,r} \cap D''}$, where
$D_{r,s} = X/r - S_s$
$D_{s,r} = X/s - S_r$
or
there exists L_t such that $u, L_0, L_t \mathrel{|}\!\equiv t$ and $L_1 = L_r|_{D_{r,t} \cap D''} \cup L_t|_{D_{t,r} \cap D''}$, where
$D_{r,t} = X/r - S_t$
$D_{t,r} = X/t - S_r$

Note that by Corollary 6.4.2, (†) and (∗) from the proof of Theorem 6.4.5 hold.

Assume (A). Assume that $u, L_0, L' \mathrel{|}\!\equiv s$. This assumption is without loss of generality since a symmetric argument can be given with t replacing s. Let $L_s = L'$. To show (B), it suffices to show that

$$L_r|_{D_{r,st}} \cup L'|_{D_{st,r}} = L_r|_{D_{r,s} \cap D''} \cup L'|_{D_{s,r} \cap D''}$$

This equality is established by the same argument as in the proof of Theorem 6.4.5.

Now assume (B). Assume that there exists L_s such that $u, L_0, L_s \mathrel{|}\!\equiv s$ and $L_1 =$

$L_r|_{D_{r,s}\cap D''} \cup L_s|_{D_{s,r}\cap D''}$, where

$$\begin{aligned} D_{r,s} &= X/r - S_s \\ D_{s,r} &= X/s - S_r \end{aligned}$$

This assumption is without loss of generality since a symmetric argument can be given with t replacing s. Let $L' = L_s$. To show (A), it suffices to show that

$$L_r|_{D_{r,st}} \cup L_s|_{D_{st,r}} = L_r|_{D_{r,s}\cap D''} \cup L_s|_{D_{s,r}\cap D''}$$

This equality is established by the same argument as in the proof of Theorem 6.4.5.

$\square$

One should not expect "`or`" to distribute over "`intersect`" (respectively, "`and`") at the level of *flow* or at the semantic level. This is because such distribution produces non-generic operands of the "`intersect`" (respectively, "`and`"), and local variables assigned in both operands will not flow out. At the level of *flow*,

$$X/\texttt{(}r\ \texttt{or}\ \texttt{(}s\ \texttt{intersect}\ t\texttt{))} = X/r \cap X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)}$$

while

$$\begin{aligned}
& X/\texttt{((}r\ \texttt{or}\ s\texttt{)}\ \texttt{intersect}\ \texttt{(}r\ \texttt{or}\ t\texttt{))} \\
=\ & (X/\texttt{(}r\ \texttt{or}\ s\texttt{)} \cup X/\texttt{(}r\ \texttt{or}\ t\texttt{)}) - (B_{\texttt{(}r\ \texttt{or}\ s\texttt{)}} \cup B_{\texttt{(}r\ \texttt{or}\ t\texttt{)}} \cup (S_{\texttt{(}r\ \texttt{or}\ s\texttt{)}} \cap S_{\texttt{(}r\ \texttt{or}\ t\texttt{)}})) \\
=\ & ((X/r \cap X/s) \cup (X/r \cap X/t)) - (B_r \cup B_s \cup B_t \cup ((S_r \cup S_s) \cap (S_r \cup S_t))) \\
=\ & (X/r \cap (X/s \cup X/t)) - (B_r \cup B_s \cup B_t \cup (S_r \cup (S_s \cap S_t))) \\
=\ & [\text{Theorem 4.5}] \\
& (X/r \cap (X/s \cup X/t)) - (S_r \cup B_s \cup B_t \cup (S_s \cap S_t)) \\
=\ & (X/r - S_r) \cap ((X/s \cup X/t) - (B_s \cup B_t \cup (S_s \cap S_t))) \\
=\ & (X/r - S_r) \cap X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)}
\end{aligned}$$

If the sequence r has no local variable assignment (i.e., $S_r = \{\}$), then

$$X/\texttt{(}r\ \texttt{or}\ \texttt{(}s\ \texttt{intersect}\ t\texttt{))} = X/\texttt{((}r\ \texttt{or}\ s\texttt{)}\ \texttt{intersect}\ \texttt{(}r\ \texttt{or}\ t\texttt{))}$$

Thus, a restricted form of distributivity of "`or`" over "`intersect`" holds at the level of *flow*. At the semantic level, there are issues of consistency of the values of local variables that prevent distributivity, even if the sequence r has no local variable assignment. For example, for $\ell \in 2^{\mathbf{P}}$,

$$\ell, \{(v,\texttt{0})\}, \{(v,\texttt{1})\} \models\!\equiv \texttt{((1 or 0) intersect (1 or (1, } v \texttt{ = 1)))}$$

but

$$\ell, \{(v,\texttt{0})\}, \{(v,\texttt{1})\} \not\models\!\equiv \texttt{(1 or (0 intersect (1, } v \texttt{ = 1)))}$$

The following implication does hold.

Theorem 6.4.7. *Let $S_r = \{\}$. Then*

$$w, L_0, L_1 \models\!\equiv \texttt{(}r\ \texttt{or}\ \texttt{(}s\ \texttt{intersect}\ t\texttt{))}$$

implies

$$w, L_0, L_1 \models\!\equiv \texttt{((}r\ \texttt{or}\ s\texttt{)}\ \texttt{intersect}\ \texttt{(}r\ \texttt{or}\ t\texttt{))}$$

Proof: Let $X = \mathrm{dom}(L_0)$. Note that since $S_r = \{\}$, Theorem 4.5 implies that $F_r = B_r = \{\}$, and so by Theorem 4.1,

$$X/r = (X \cup F_r) - B_r = X$$

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{(}r\ \texttt{or (}s\ \texttt{intersect}\ t\texttt{))}$

iff there exists L such that $L_1 = L|_D$, where

$$\begin{aligned} D &= X/\texttt{(}r\ \texttt{or (}s\ \texttt{intersect}\ t\texttt{))} \\ &= X/r \cap X/\texttt{(}s\ \texttt{intersect}\ t\texttt{)} \\ &= X \cap ((X/s \cup X/t) - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}) \\ &= ((X \cap X/s) \cup (X \cap X/t)) - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \\ &= ((X - B_s) \cup (X - B_t)) - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \\ &= X - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \end{aligned}$$

and such that either $w, L_0, L \mathrel{|\!\!\equiv} r$ or $w, L_0, L \mathrel{|\!\!\equiv} \texttt{(}s\ \texttt{intersect}\ t\texttt{)}$

iff [Lemma 5.3]

there exists L such that $L_1 = L|_D$, where $D = X - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}$, and such that either

$w, L_0, L \mathrel{|\!\!\equiv} r$

or

there exist L_s, L_t such that $w, L_0, L_s \mathrel{|\!\!\equiv} s$ and $w, L_0, L_t \mathrel{|\!\!\equiv} t$ and $L = L_s|_{D_s} \cup L_t|_{D_t}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

iff [if $w, L_0, L \mathrel{|\!\!\equiv} r$, then $L = L_0$]

letting $D = X - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}$, either

$w, L_0, L_0 \mathrel{|\!\!\equiv} r$ and $L_1 = L_0|_D$

or

there exist L, L_s, L_t such that $w, L_0, L_s \mathrel{|\!\!\equiv} s$ and $w, L_0, L_t \mathrel{|\!\!\equiv} t$ and $L = L_s|_{D_s} \cup L_t|_{D_t}$ and $L_1 = L|_D$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

iff [L is determined in the second disjunct]

letting $D = X - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}$, either

$w, L_0, L_0 \mathrel{|\!\!\equiv} r$ and $L_1 = L_0|_D$

or

there exist L_s, L_t such that $w, L_0, L_s \mathrel{|\!\!\equiv} s$ and $w, L_0, L_t \mathrel{|\!\!\equiv} t$ and $L_1 = L_s|_{D_s \cap D} \cup L_t|_{D_t \cap D}$, where

$D_s = X/s - S_t$

$D_t = X/t - S_s$

Note that

$$\begin{aligned} D_s \cap D &= (X \cap X/s) - (B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}} \cup S_t) \\ &= (X - B_s) - (B_s \cup B_t \cup (S_s \cap S_t) \cup S_t) \\ &= X - (B_s \cup S_t) \end{aligned}$$

and similarly

$$D_t \cap D = X - (B_t \cup S_s)$$

It follows that

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{(}r\ \texttt{or}\ \texttt{(}s\ \texttt{intersect}\ t\texttt{))}$
iff
(A):
either
$w, L_0, L_0 \mathrel{|\!\!\equiv} r$ and $L_1 = L_0|_D$, where $D = X - B_{\texttt{(}s\ \texttt{intersect}\ t\texttt{)}}$,
or
there exist L_s, L_t such that $w, L_0, L_s \mathrel{|\!\!\equiv} s$ and $w, L_0, L_t \mathrel{|\!\!\equiv} t$ and
$L_1 = L_s|_{X-(B_s \cup S_t)} \cup L_t|_{X-(B_t \cup S_s)}$

Since $S_r = \{\}$ and $X/r = X$, it follows that

$$\begin{aligned} S_{\texttt{(}r\ \texttt{or}\ s\texttt{)}} &= S_s \\ S_{\texttt{(}r\ \texttt{or}\ t\texttt{)}} &= S_t \\ X/\texttt{(}r\ \texttt{or}\ s\texttt{)} &= X - B_s \\ X/\texttt{(}r\ \texttt{or}\ t\texttt{)} &= X - B_t \end{aligned}$$

Then

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{((}r\ \texttt{or}\ s\texttt{)}\ \texttt{intersect}\ \texttt{(}r\ \texttt{or}\ t\texttt{))}$
iff [Lemma 5.3]
there exist L'_*, L''_* such that $w, L_0, L'_* \mathrel{|\!\!\equiv} \texttt{(}r\ \texttt{or}\ s\texttt{)}$ and $w, L_0, L''_* \mathrel{|\!\!\equiv} \texttt{(}r\ \texttt{or}\ t\texttt{)}$
and $L_1 = L'_*|_{D'} \cup L''_*|_{D''}$, where
$D' = \ X/\texttt{(}r\ \texttt{or}\ s\texttt{)} - S_{\texttt{(}r\ \texttt{or}\ t\texttt{)}} = (X - B_s) - S_t = X - (B_s \cup S_t)$
and
$D'' = \ X/\texttt{(}r\ \texttt{or}\ t\texttt{)} - S_{\texttt{(}r\ \texttt{or}\ s\texttt{)}} = (X - B_t) - S_s = X - (B_t \cup S_s)$
iff there exist L'_*, L''_* such that
there exists L' such that $L'_* = L'|_{X-B_s}$ and either
$w, L_0, L' \mathrel{|\!\!\equiv} r$ or $w, L_0, L' \mathrel{|\!\!\equiv} s$
and
there exists L'' such that $L''_* = L''|_{X-B_t}$ and either
$w, L_0, L'' \mathrel{|\!\!\equiv} r$ or $w, L_0, L'' \mathrel{|\!\!\equiv} t$
and $L_1 = L'_*|_{D'} \cup L''_*|_{D''}$, where
$D' = \ X - (B_s \cup S_t)$
$D'' = \ X - (B_t \cup S_s)$
iff [L'_*, L''_* are determined]
there exist L', L'' such that
either $w, L_0, L' \mathrel{|\!\!\equiv} r$ or $w, L_0, L' \mathrel{|\!\!\equiv} s$
and
either $w, L_0, L'' \mathrel{|\!\!\equiv} r$ or $w, L_0, L'' \mathrel{|\!\!\equiv} t$
and $L_1 = L'|_{(X-B_s)\cap D'} \cup L''|_{(X-B_t)\cap D''}$, where
$D' = \ X - (B_s \cup S_t)$
$D'' = \ X - (B_t \cup S_s)$
iff [$(X - B_s) \cap D' = D'$, $(X - B_t) \cap D'' = D''$]

(B):
there exist L', L'' such that
either $w, L_0, L' \mathrel{|\!\!\equiv} r$ or $w, L_0, L' \mathrel{|\!\!\equiv} s$
and

either $w, L_0, L'' \models r$ or $w, L_0, L'' \models t$
and $L_1 = L'|_{X-(B_s \cup S_t)} \cup L''|_{X-(B_t \cup S_s)}$

Assume (A). Suppose $w, L_0, L_0 \models r$ and $L_1 = L_0|_D$. Let $L' = L'' = L_0$. Then

$$\begin{aligned}
& L'|_{X-(B_s \cup S_t)} \cup L''|_{X-(B_t \cup S_s)} \\
= \;& L_0|_{X-(B_s \cup S_t)} \cup L_0|_{X-(B_t \cup S_s)} \\
= \;& L_0|_{X-((B_s \cup S_t) \cap (B_t \cup S_s))} \\
= \;& [\text{Lemma 5.2}] \\
& L_0|_{X-B_{(s \texttt{ intersect } t)}} \\
= \;& L_0|_D \\
= \;& L_1
\end{aligned}$$

This proves (B) in this case. Suppose now that there exist L_s, L_t such that $w, L_0, L_s \models s$ and $w, L_0, L_t \models t$ and

$$L_1 = L_s|_{X-(B_s \cup S_t)} \cup L_t|_{X-(B_t \cup S_s)}$$

Let $L' = L_s$, and let $L'' = L_t$. Then

$$L'|_{X-(B_s \cup S_t)} \cup L''|_{X-(B_t \cup S_s)} = L_s|_{X-(B_s \cup S_t)} \cup L_t|_{X-(B_t \cup S_s)} = L_1$$

This proves (B) in this case. □

6.5 Identity elements

An *identity element* for a binary operator "$\odot$" on a set A is an element $i \in A$ such that $i \odot a = a \odot i = a$ for all $a \in A$. If "$\odot$" is a commutative operator, then i is an identity element iff $i \odot a = a$ for all $a \in A$.

The following proposition shows that "`1[*0:$]`" is an identity element for "`intersect`" at the level of *flow* and at the semantic level.

Proposition 6.5.1.

1. $X/(\texttt{1[*0:\$] intersect } r) = X/r$.

2. $w, L_0, L_1 \models (\texttt{1[*0:\$] intersect } r)$ *iff* $w, L_0, L_1 \models r$.

Proof:

$$\begin{aligned}
& X/(\texttt{1[*0:\$] intersect } r) \\
= \;& [\text{Lemma 5.3}] \\
& (X/\texttt{1[*0:\$]} - S_r) \cup (X/r - S_{\texttt{1[*0:\$]}}) \\
= \;& [\text{Lemma 6.2.7(1), Lemma 6.2.9(1)}] \\
& (X - S_r) \cup (X/r - \{\}) \\
= \;& [\text{Lemma 6.4.3}] \\
& (X/r - S_r) \cup X/r \\
= \;& X/r
\end{aligned}$$

This proves part 1. Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{(1[*0:\$] intersect } r\texttt{)}$

iff [Lemma 5.3]

there exist L, L_r such that $w, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*0:\$]}$ and $w, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L|_D \cup L_r|_{D_r}$,

where

$D = X/\texttt{1[*0:\$]} - S_r = X - S_r$

$D_r = X/r - S_{\texttt{1[*0:\$]}} = X/r$

iff [by Theorem 5.4, if $w, L_0, L_r \mathrel{|\!\!\equiv} r$, then $\mathrm{dom}(L_r) = X/r$]

there exist L, L_r such that $w, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*0:\$]}$ and $w, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L|_{X-S_r} \cup L_r$

iff [Lemma 6.2.11(1)]

there exist L, L_r such that $L = L_0$ and no letter of w is $\perp$ and $w, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L|_{X-S_r} \cup L_r$

iff [by Lemma 5.1, if $w, L_0, L_r \mathrel{|\!\!\equiv} r$, then no letter of w is $\perp$]

there exist L, L_r such that $L = L_0$ and $w, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L|_{X-S_r} \cup L_r$

iff [$L = L_0$ determines L]

there exists L_r such that $w, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L_0|_{X-S_r} \cup L_r$

iff [by Theorem 5.5, if $w, L_0, L_r \mathrel{|\!\!\equiv} r$, then $L_0|_{X-S_r} \subseteq L_r$]

there exists L_r such that $w, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L_r$

iff [L_r is determined by $L_1 = L_r$]

$w, L_0, L_1 \mathrel{|\!\!\equiv} r$

This proves part 2. □

The following proposition shows that "`1[*0]`" is an identity element for "`and`" at the level of *flow* and at the semantic level.

Proposition 6.5.2.

1. $X/\texttt{(1[*0] and } r\texttt{)} = X/r$.

2. $w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{(1[*0] and } r\texttt{)}$ *iff* $w, L_0, L_1 \mathrel{|\!\!\equiv} r$.

Proof:

$X/\texttt{(1[*0] and } r\texttt{)}$

= [Lemma 6.2.9(4), Lemma 5.3]

$(X/\texttt{1[*0]} - S_r) \cup (X/r - S_{\texttt{1[*0]}})$

= $(X - S_r) \cup (X/r - \{\})$

= [Lemma 6.4.3]

$(X/r - S_r) \cup X/r$

= X/r

This proves part 1. Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \mathrel{|\!\!\equiv} \texttt{(1[*0] and } r\texttt{)}$

iff [Lemma 6.2.11(4)]

there exist u, v, u_r, v_r, L, L_r such that $w = uv = u_r v_r$ and at least one of v, v_r is empty and $u, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*0]}$ and $u_r, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L|_D \cup L_r|_{D_r}$,

where
$D = X/\texttt{1[*0]} - S_r = X - S_r$
$D_r = X/r - S_{\texttt{1[*0]}} = X/r$
iff [by Theorem 5.4, if $u_r, L_0, L_r \mathrel{|\!\!\equiv} r$, then $\mathrm{dom}(L_r) = X/r$]
there exist u, v, u_r, v_r, L, L_r such that $w = uv = u_r v_r$ and at least one of v, v_r is empty and $u, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*0]}$ and $u_r, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L|_{X-S_r} \cup L_r$
iff there exist u, v, u_r, v_r, L, L_r such that $w = uv = u_r v_r$ and at least one of v, v_r is empty and $|u| = 0$ and $L = L_0$ and $u_r, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L|_{X-S_r} \cup L_r$
iff [$L = L_0$ determines L; $|u| = 0$ determines u, hence $w = v$ determines v]
there exist u_r, v_r, L_r such that $w = u_r v_r$ and at least one of w, v_r is empty and $u_r, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L_0|_{X-S_r} \cup L_r$
iff [if $w = u_r v_r$ and at least one of w, v_r is empty, then $w = u_r$]
there exists L_r such that $w, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L_0|_{X-S_r} \cup L_r$
iff [by Theorem 5.5, if $w, L_0, L_r \mathrel{|\!\!\equiv} r$, then $L_0|_{X-S_r} \subseteq L_r$]
there exists L_r such that $w, L_0, L_r \mathrel{|\!\!\equiv} r$ and $L_1 = L_r$
iff [L_r is determined by $L_1 = L_r$]
$w, L_0, L_1 \mathrel{|\!\!\equiv} r$

This proves part 2. □

Remark: Note that "`1`" is not an identity element for "`and`" at the semantic level. This is because a word that tightly satisfies "`(1 and` r`)`" must be non-empty. □

Since $X/(s \texttt{ or } t) = X/s \cap X/t$, "`or`" cannot have an identity element at the level of *flow* or at the semantic level. If the sequences are restricted to those without local variable assignments and the words are restricted to the alphabet $2^{\mathbf{P}}$, then "`0`" is an identity element for "`or`" at the level of *flow* and at the semantic level, as shown in the following proposition.

Proposition 6.5.3. *Let* $S_r = \{\}$.

1. $X/(\texttt{0 or } r) = X = X/r$.

2. *If* w *is a word over* $2^{\mathbf{P}}$*, then* $w, L_0, L_1 \mathrel{|\!\!\equiv} (\texttt{0 or } r)$ *iff* $w, L_0, L_1 \mathrel{|\!\!\equiv} r$.

Proof: Since $S_r = \{\}$, Theorem 4.5 gives $F_r = B_r = \{\}$, and $X = X/r$ follows from Theorem 4.1. Then

$X/(\texttt{0 or } r)$
$= X/\texttt{0} \cap X/r$
$= X \cap X$
$= X$

This proves part 1. Let $X = \mathrm{dom}(L_0)$.

$w, L_0, L_1 \mathrel{|\!\!\equiv} (\texttt{0 or } r)$
iff there exists L such that $L_1 = L|_D$, where $D = X/(\texttt{0 or } r)$, and such that either $w, L_0, L \mathrel{|\!\!\equiv} \texttt{0}$ or $w, L_0, L \mathrel{|\!\!\equiv} r$
iff [part 1]

there exists L such that $L_1 = L|_X$ and such that either $w, L_0, L \mathrel{|\!\!\equiv} r$ or
$w, L_0, L \mathrel{|\!\!\equiv} \texttt{0}$
iff there exists L such that $L_1 = L|_X$ and such that either $w, L_0, L \mathrel{|\!\!\equiv} r$ or
$|w| = 1$ and $w^0 \Vdash \texttt{0}$ and $L = L_0$
iff $[w^0 \in 2^{\mathbf{P}}$ implies $w^0 \nVdash \texttt{0}]$
there exists L such that $L_1 = L|_X$ and $w, L_0, L \mathrel{|\!\!\equiv} r$
iff [by Theorem 5.4, if $w, L_0, L \mathrel{|\!\!\equiv} r$, then $\mathrm{dom}(L) = X/r = X$]
there exists L such that $L_1 = L$ and $w, L_0, L \mathrel{|\!\!\equiv} r$
iff [L is determined by $L_1 = L$]
$w, L_0, L_1 \mathrel{|\!\!\equiv} r$

$\square$

The following proposition shows that "`1[*0]`" is an identity element for "`##1`" at the level of *flow* and at the semantic level.

Proposition 6.5.4.

1. $X/(\texttt{1[*0] ##1 } r) = X/(r \texttt{ ##1 1[*0]}) = X/r$.

2. $w, L_0, L_1 \mathrel{|\!\!\equiv} (\texttt{1[*0] ##1 } r)$ *iff* $w, L_0, L_1 \mathrel{|\!\!\equiv} (r \texttt{ ##1 1[*0]})$ *iff* $w, L_0, L_1 \mathrel{|\!\!\equiv} r$.

Proof:

$X/(\texttt{1[*0] ##1 } r)$
$= (X/\texttt{1[*0]})/r$
$= X/r$
$= (X/r)/\texttt{1[*0]}$
$= X/(r \texttt{ ##1 1[*0]})$

This proves part 1.

$w, L_0, L_1 \mathrel{|\!\!\equiv} (\texttt{1[*0] ##1 } r)$
iff there exist x, y, L such that $w = xy$ and $x, L_0, L \mathrel{|\!\!\equiv} \texttt{1[*0]}$ and $y, L, L_1 \mathrel{|\!\!\equiv} r$
iff there exist x, y, L such that $w = xy$ and $|x| = 0$ and $L = L_0$ and
$y, L, L_1 \mathrel{|\!\!\equiv} r$
iff [$L = L_0$ determines L; $w = xy$ and $|x| = 0$ determine x and y]
$w, L_0, L_1 \mathrel{|\!\!\equiv} r$

The argument that $w, L_0, L_1 \mathrel{|\!\!\equiv} (r \texttt{ ##1 1[*0]})$ iff $w, L_0, L_1 \mathrel{|\!\!\equiv} r$ is similar. This proves part 2. $\square$

Remark: There is no identity element for "`##0`" at the semantic level because, for example, "`1[*0]`" cannot be semantically equivalent to "$(s \texttt{ ##0 1[*0]})$" for any s.

$\square$

7 An Alternative Approach to *block* and *flow*

The definitions of *block* and *flow* are unsatisfying for two reasons. The first is simply that the definitions are unnecessarily interdependent. Independent definitions allow for selective implementation of the functions, possibly for performance gains. The second reason is the fact that the intuitive notion, "that which is sampled and does not flow is blocked", does not hold. That is, $\mathit{sample}(s) - \mathit{flow}(X, s)$ does not, in general, equal $\mathit{block}(s)$. Consider the case $X = \{\}$, $s =$ `((1,` v `=` e`) or` b`)`. Then $\mathit{sample}(s) = \{v\}$ and $\mathit{flow}(X, s) = \{\}$, but $\mathit{block}(s) = \{\} \neq \{v\} - \{\}$.

This section introduces new functions block' and flow' that are independent and for which the intuitive notion, "that which is sampled and does not flow is blocked", does hold. It then shows that the definition of flow' is equivalent to that of *flow* and that block' behaves as expected.

7.1 Additional facts and notation used within this section

The following notation and shorthand is used freely within this section:

- "$\ominus$" denotes set symmetric difference.
- F'_r is shorthand for $\mathit{flow}'(\{\}, r)$.
- $F'(X, r)$ is shorthand for $\mathit{flow}'(X, r)$.
- B'_r is shorthand for $\mathit{block}'(\{\}, r)$.
- $B'(X, r)$ is shorthand for $\mathit{block}'(X, r)$.

The following facts for arbitrary sets A, B, M, and N are assumed to be well-known:

- $(A - M) \cup B = (A \cup B) - (M - B)$.
- $A \cap B = (A \cup B) - (A \ominus B)$.
- $(A \cup M) \ominus (A \cup N) = (M \ominus N) - A$.

7.2 Definition of block' and flow'

The function block' takes a set of local variable names and a sequence as input and returns a set of local variable names as output. This function defines the set of local variable names that are blocked by the sequence given the set of local variable names that flow into the sequence.

Definition 7.2.1 (block')**.**

- $\mathit{block}'(X,$ `(1,` v `=` e`)`$) = \{\}$
- $\mathit{block}'(X, b) = \{\}$
- $\mathit{block}'(X,$ `(`s`)`$) = \mathit{block}'(X, s)$

- $\mathit{block}'(X, \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}) = (\mathit{block}'(X,s) - \mathit{sample}(t))$
  $\cup\ \mathit{block}'((X \cup \mathit{sample}(s)) - \mathit{block}'(X,s), t)$
- $\mathit{block}'(X, \texttt{(}s\ \texttt{\#\#0}\ t\texttt{)}) = (\mathit{block}'(X,s) - \mathit{sample}(t))$
  $\cup\ \mathit{block}'((X \cup \mathit{sample}(s)) - \mathit{block}'(X,s), t)$
- $\mathit{block}'(X, \texttt{(}s\ \texttt{or}\ t\texttt{)}) = \mathit{block}'(X,s) \cup \mathit{block}'(X,t) \cup ((\mathit{sample}(s) \ominus \mathit{sample}(t)) - X)$
- $\mathit{block}'(X, \texttt{(}s\ \texttt{intersect}\ t\texttt{)}) = \mathit{block}'(X,s) \cup \mathit{block}'(X,t) \cup (\mathit{sample}(s) \cap \mathit{sample}(t))$
- $\mathit{block}'(X, \texttt{first\_match(}s\texttt{)}) = \mathit{block}'(X,s)$
- $\mathit{block}'(X, s\texttt{[*0]}) = \{\}$
- $\mathit{block}'(X, s\texttt{[*1:\$]}) = \mathit{block}'(X,s)$

□

The function *flow′* takes a set of local variable names and a sequence as input and returns a set of local variable names as output. This function defines the set of local variable names that flow out of the sequence given the set of local variable names that flow into the sequence.

Definition 7.2.2 (*flow′*).

- $\mathit{flow}'(X, \texttt{(1,}\ v\ \texttt{=}\ e\texttt{)}) = X \cup \{v\}$
- $\mathit{flow}'(X, b) = X$
- $\mathit{flow}'(X, \texttt{(}s\texttt{)}) = \mathit{flow}'(X,s)$
- $\mathit{flow}'(X, \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}) = \mathit{flow}'(\mathit{flow}'(X,s), t)$
- $\mathit{flow}'(X, \texttt{(}s\ \texttt{\#\#0}\ t\texttt{)}) = \mathit{flow}'(\mathit{flow}'(X,s), t)$
- $\mathit{flow}'(X, \texttt{(}s\ \texttt{or}\ t\texttt{)}) = \mathit{flow}'(X,s) \cap \mathit{flow}'(X,t)$
- $\mathit{flow}'(X, \texttt{(}s\ \texttt{intersect}\ t\texttt{)}) = (\mathit{flow}'(X,s) - \mathit{sample}(t)) \cup (\mathit{flow}'(X,t) - \mathit{sample}(s))$
- $\mathit{flow}'(X, \texttt{first\_match(}s\texttt{)}) = \mathit{flow}'(X,s)$
- $\mathit{flow}'(X, s\texttt{[*0]}) = X$
- $\mathit{flow}'(X, s\texttt{[*1:\$]}) = \mathit{flow}'(X,s)$

□

7.3 Equivalence of *flow* and *flow′*

Proposition 7.3.1. *$\mathit{flow}'(X,r) = \mathit{flow}(X,r)$.*

Proof: Equivalence is established by induction over the structure of r. For all cases but $r = \texttt{(}s\ \texttt{intersect}\ t\texttt{)}$, the proof follows directly from the definitions. For the remaining case, the proof follows easily from Lemma 5.3. □

7.4 Containment of *block′* in *sample*

Proposition 7.4.1. $\mathit{block}'(X, r) \subseteq \mathit{sample}(r)$.

Proof: By induction over the structure of r.

- $r = \texttt{(1, }v\texttt{ = }e\texttt{)}$.

$$\begin{aligned}
&\mathit{block}'(X, \texttt{(1, }v\texttt{ = }e\texttt{)})\\
=\ &\{\}\\
\subseteq\ &\{v\}\\
=\ &\mathit{sample}(\texttt{(1, }v\texttt{ = }e\texttt{)})
\end{aligned}$$

- $r = b$.

$$\begin{aligned}
&\mathit{block}'(X, b)\\
=\ &\{\}\\
=\ &\mathit{sample}(b)
\end{aligned}$$

- $r = \texttt{(}s\texttt{)}$.

$$\begin{aligned}
&\mathit{block}'(X, \texttt{(}s\texttt{)})\\
=\ &B'(X, s)\\
\subseteq\ &[\text{induction}]\\
&S_s\\
=\ &\mathit{sample}(\texttt{(}s\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\texttt{ ##1 }t\texttt{)}$.

$$\begin{aligned}
&\mathit{block}'(X, \texttt{(}s\texttt{ ##1 }t\texttt{)})\\
=\ &(B'(X, s) - S_t) \cup B'((X \cup S_s) - B'(X, s), t)\\
\subseteq\ &[\text{induction}]\\
&(S_s - S_t) \cup S_t\\
=\ &S_s \cup S_t\\
=\ &\mathit{sample}\texttt{(}s\texttt{ ##1 }t\texttt{)}
\end{aligned}$$

- $r = \texttt{(}s\texttt{ ##0 }t\texttt{)}$. Analogous to the argument for $r = \texttt{(}s\texttt{ ##1 }t\texttt{)}$.

- $r = \texttt{(}s\texttt{ or }t\texttt{)}$.

$$\begin{aligned}
&\mathit{block}'(X, \texttt{(}s\texttt{ or }t\texttt{)})\\
=\ &B'(X, s) \cup B'(X, t) \cup ((S_s \ominus S_t) - X)\\
\subseteq\ &[\text{induction}]\\
&S_s \cup S_t\\
=\ &\mathit{sample}(\texttt{(}s\texttt{ or }t\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\texttt{ intersect }t\texttt{)}$.

$$\begin{array}{ll} & \mathit{block}'(X, \texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \\ = & B'(X,s) \cup B'(X,t) \cup (S_s \cap S_t) \\ \subseteq & \text{[induction]} \\ & S_s \cup S_t \\ = & \mathit{sample}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \end{array}$$

- $r = \texttt{first\_match(}s\texttt{)}$. Analogous to the argument for $r = \texttt{(}s\texttt{)}$.
- $r = s\texttt{[*0]}$. Analogous to the argument for $r = b$.
- $r = s\texttt{[*1:\$]}$. Analogous to the argument for $r = \texttt{(}s\texttt{)}$.

$\square$

7.5 Relationship of *flow′* and *block′*

Lemma 7.5.1. $B \cup D \cup ((A - B) \ominus (C - D)) = B \cup D \cup (A \ominus C)$.

Proof:

$$\begin{array}{ll} & B \cup D \cup ((A - B) \ominus (C - D)) \\ = & B \cup D \cup ((A - B) - (C - D)) \cup ((C - D) - (A - B)) \\ = & B \cup D \cup ((A \cap D) - B) \cup (A - (B \cup C)) \cup ((C \cap B) - D) \cup (C - (D \cup A)) \\ = & B \cup D \cup (A \cap D) \cup (A - (B \cup C)) \cup (C \cap B) \cup (C - (D \cup A)) \\ = & B \cup D \cup (A - (B \cup C)) \cup (C - (D \cup A)) \\ = & B \cup D \cup (A - C) \cup (C - A) \\ = & B \cup D \cup (A \ominus C) \end{array}$$

$\square$

Proposition 7.5.2. $\mathit{flow}'(X, r) = (X \cup \mathit{sample}(r)) - \mathit{block}'(X, r)$.

Proof: By induction over the structure of r.

- $r = \texttt{(1,}\ v\ \texttt{=}\ e\texttt{)}$.

$$\begin{array}{ll} & \mathit{flow}'(X, \texttt{(1,}\ v\ \texttt{=}\ e\texttt{)}) \\ = & X \cup \{v\} \\ = & (X \cup \{v\}) - \{\} \\ = & (X \cup \mathit{sample}(\texttt{(1,}\ v\ \texttt{=}\ e\texttt{)})) - \mathit{block}'(X, \texttt{(1,}\ v\ \texttt{=}\ e\texttt{)}) \end{array}$$

- $r = b$.

$$\begin{array}{ll} & \mathit{flow}'(X, b) \\ = & X \\ = & (X \cup \{\}) - \{\} \\ = & (X \cup \mathit{sample}(b)) - \mathit{block}'(X, b) \end{array}$$

- $r = \texttt{(}s\texttt{)}$.

$$\begin{aligned}
&\mathit{flow}'(X, \texttt{(}s\texttt{)}) \\
={}& \mathit{flow}'(X, s) \\
={}& [\text{induction}] \\
& (X \cup \mathit{sample}(s)) - \mathit{block}'(X, s) \\
={}& (X \cup \mathit{sample}(\texttt{(}s\texttt{)})) - \mathit{block}'(X, \texttt{(}s\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}$.

$$\begin{aligned}
&\mathit{flow}'(X, \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}) \\
={}& F'(F'(X, s), t) \\
={}& [\text{induction}] \\
& F'((X \cup S_s) - B'(X, s), t) \\
={}& [\text{induction}] \\
& (((X \cup S_s) - B'(X, s)) \cup S_t) - B'((X \cup S_s) - B'(X, s), t) \\
={}& ((X \cup S_s \cup S_t) - (B'(X, s) - S_t)) - B'((X \cup S_s) - B'(X, s), t) \\
={}& (X \cup S_s \cup S_t) - ((B'(X, s) - S_t) \cup B'((X \cup S_s) - B'(X, s), t)) \\
={}& (X \cup \mathit{sample}(\texttt{(}s\ \texttt{\#\#1}\ t\texttt{)})) - \mathit{block}'(X, \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\ \texttt{\#\#0}\ t\texttt{)}$. Analogous to the argument for $r = \texttt{(}s\ \texttt{\#\#1}\ t\texttt{)}$.
- $r = \texttt{(}s\ \texttt{or}\ t\texttt{)}$.

$$\begin{aligned}
&\mathit{flow}'(X, \texttt{(}s\ \texttt{or}\ t\texttt{)}) \\
={}& F'(X, s) \cap F'(X, t) \\
={}& [\text{induction}] \\
& ((X \cup S_s) - B'(X, s)) \cap ((X \cup S_t) - B'(X, t)) \\
={}& ((X \cup S_s) \cap (X \cup S_t)) - (B'(X, s) \cup B'(X, t)) \\
={}& (((X \cup S_s) \cup (X \cup S_t)) - ((X \cup S_s) \ominus (X \cup S_t))) - (B'(X, s) \cup B'(X, t)) \\
={}& ((X \cup S_s) \cup (X \cup S_t)) - (B'(X, s) \cup B'(X, t) \cup ((X \cup S_s) \ominus (X \cup S_t))) \\
={}& ((X \cup S_s) \cup (X \cup S_t)) - (B'(X, s) \cup B'(X, t) \cup ((S_s \ominus S_t) - X)) \\
={}& (X \cup \mathit{sample}(\texttt{(}s\ \texttt{or}\ t\texttt{)})) - \mathit{block}'(X, \texttt{(}s\ \texttt{or}\ t\texttt{)})
\end{aligned}$$

- $r = \texttt{(}s\ \texttt{intersect}\ t\texttt{)}$.

$$\begin{aligned}
&\mathit{flow}'(\texttt{(}s\ \texttt{intersect}\ t\texttt{)}) \\
={}& (F'(X, s) - S_t) \cup (F'(X, t) - S_s) \\
={}& [\text{induction}] \\
& (((X \cup S_s) - B'(X, s)) - S_t) \cup (((X \cup S_t) - B'(X, t)) - S_s) \\
={}& ((X \cup S_s) - (B'(X, s) \cup S_t)) \cup ((X \cup S_t) - (B'(X, t) \cup S_s)) \\
={}& ((X \cup S_s \cup S_t) - (B'(X, s) \cup S_t)) \cup ((X \cup S_s \cup S_t) - (B'(X, t) \cup S_s)) \\
={}& (X \cup S_s \cup S_t) - ((B'(X, s) \cup S_t) \cap (B'(X, t) \cup S_s)) \\
={}& (X \cup S_s \cup S_t) - ((B'(X, s) \cap (B'(X, t) \cup S_s)) \cup (S_t \cap (B'(X, t) \cup S_s))) \\
={}& (X \cup S_s \cup S_t) \\
& \quad - (((B'(X, s) \cap B'(X, t)) \cup (B'(X, s) \cap S_s)) \\
& \qquad\qquad \cup ((S_t \cap B'(X, t)) \cup (S_t \cap S_s))) \\
={}& [\text{Proposition 7.4.1}] \\
& (X \cup S_s \cup S_t) - (B'(X, s) \cup B'(X, t) \cup (S_t \cap S_s)) \\
={}& (X \cup \mathit{sample}(\texttt{(}s\ \texttt{intersect}\ t\texttt{)})) - \mathit{block}'(X, \texttt{(}s\ \texttt{intersect}\ t\texttt{)})
\end{aligned}$$

- $r = \texttt{first\_match}(s)$. Analogous to the argument for $r = \texttt{(}s\texttt{)}$.
- $r = s\texttt{[*0]}$. Analogous to the argument for $r = b$.
- $r = s\texttt{[*1:\$]}$. Analogous to the argument for $r = \texttt{(}s\texttt{)}$.

□

Proposition 7.4.1, together with Proposition 7.5.2, gives

Corollary 7.5.3.

1. $\mathit{flow}'(X, r) \cap \mathit{block}'(X, r) = \{\}$.
2. $\mathit{block}'(X, r) = (X \cup \mathit{sample}(r)) - \mathit{flow}'(X, r)$.

□

Proposition 7.5.2, together with Lemma 7.5.1, gives

Corollary 7.5.4.

$$\mathit{block}'(X, \texttt{(}s \texttt{ or } t\texttt{)}) = \mathit{block}'(X, s) \cup \mathit{block}'(X, t) \cup (\mathit{flow}'(X, s) \ominus \mathit{flow}'(X, t))$$

Proof:

$$\begin{aligned}
& \mathit{block}'(X, \texttt{(}s \texttt{ or } t\texttt{)}) \\
= \;& B'(X, s) \cup B'(X, t) \cup ((S_s \ominus S_t) - X) \\
= \;& B'(X, s) \cup B'(X, t) \cup ((X \cup S_s) \ominus (X \cup S_t)) \\
= \;& [\text{Lemma 7.5.1}] \\
& B'(X, s) \cup B'(X, t) \cup (((X \cup S_s) - B'(X, s)) \ominus ((X \cup S_t) - B'(X, t))) \\
= \;& [\text{Proposition 7.5.2}] \\
& B'(X, s) \cup B'(X, t) \cup (F'(X, s) \ominus F'(X, t))
\end{aligned}$$

□

Proposition 7.5.5. $\mathit{flow}'(X, r) = (X \cup \mathit{flow}'(\{\}, r)) - \mathit{block}'(X, r)$.

Proof:

$$\begin{aligned}
& F'(X, r) \\
= \;& [\text{Corollary 7.5.3}] \\
& F'(X, r) - B'(X, r) \\
= \;& [\text{Proposition 7.3.1}] \\
& X/r - B'(X, r) \\
= \;& [\text{Theorem 4.1}] \\
& ((X \cup F_r) - B_r) - B'(X, r) \\
= \;& (X \cup F_r) - (B_r \cup B'(X, r)) \\
= \;& ((X \cup F_r) - B_r) \cap ((X \cup F_r) - B'(X, r)) \\
= \;& [\text{Theorem 4.1}] \\
& X/r \cap ((X \cup F_r) - B'(X, r)) \\
= \;& F'(X, r) \cap ((X \cup F_r) - B'(X, r))
\end{aligned}$$

But then $F'(X, r) \subseteq (X \cup F_r) - B'(X, r) \subseteq (X \cup S_r) - B'(X, r) = F'(X, r)$, whence $F'(X, r) = (X \cup F_r) - B'(X, r) = (X \cup F'_r) - B'(X, r)$. □

7.6 Relationship of *block* to *block′*

Proposition 7.6.1. $\mathit{block}(r) \subseteq \mathit{block}'(X, r)$.

Proof: Proposition 7.3.1 states $F'(X, r) = X/r$. Since $X/r \cap B_r = \{\}$ it follows that $F'(X, r) = F'(X, r) - B_r$, whence, by Proposition 7.5.2,

$$F'(X, r) = ((X \cup S_r) - B'(X, r)) - B_r = (X \cup S_r) - (B_r \cup B'(X, r)).$$

Therefore

$$(X \cup S_r) - B'(X, r) = F'(X, r) = (X \cup S_r) - (B_r \cup B'(X, r)).$$

It follows that $B'(X, r) \cap (X \cup S_r) = (B_r \cup B'(X, r)) \cap (X \cup S_r)$. But $B'(X, r) \subseteq S_r$ and $B_r \subseteq S_r$, so $B'(X, r) = B_r \cup B'(X, r)$, whence $B_r \subseteq B'(X, r)$. □

Proposition 7.6.2. $(\mathit{block}'(X, r) - \mathit{block}(r)) \cap X = \{\}$.

Proof:

$$\begin{aligned}
&(\mathit{block}'(X, r) - \mathit{block}(r)) \cap X \\
= \;\; &(((X \cup S_r) - F'(X, r)) - B_r) \cap X \\
= \;\; &(((X \cup S_r) - X/r) - B_r) \cap X \\
= \;\; &(((X \cup S_r) - (X \cup F_r)) - B_r) \cap X \\
= \;\; &((X \cup S_r) - (X \cup F_r \cup B_r)) \cap X \\
= \;\; &((X \cup S_r) \cap X) - (X \cup F_r \cup B_r) \\
= \;\; &\{\}
\end{aligned}$$

□

References

[SV] *Accellera SystemVerilog 3.1 Language Reference Manual*. Accellera Organization, Inc., Napa, California, 2003.

A Appendix

A.1 Flow definitions

The definitions below are from Annex G, Subsection G.3.3.3, pp. 348-349 of [SV]. The notation has been changed to conform to the conventions of these notes.

The function *sample* takes a sequence as input and returns a set of local variable names as output.

Definition A.1.1 (*sample*)**.**

- $\mathit{sample}(\texttt{(1, }v\texttt{ = }e\texttt{)}) = \{v\}$
- $\mathit{sample}(b) = \{\}$
- $\mathit{sample}(\texttt{(}s\texttt{)}) = \mathit{sample}(s)$
- $\mathit{sample}(\texttt{(}s\texttt{ ##1 }t\texttt{)}) = \mathit{sample}(s) \cup \mathit{sample}(t)$
- $\mathit{sample}(\texttt{(}s\texttt{ ##0 }t\texttt{)}) = \mathit{sample}(s) \cup \mathit{sample}(t)$
- $\mathit{sample}(\texttt{(}s\texttt{ or }t\texttt{)}) = \mathit{sample}(s) \cup \mathit{sample}(t)$
- $\mathit{sample}(\texttt{(}s\texttt{ intersect }t\texttt{)}) = \mathit{sample}(s) \cup \mathit{sample}(t)$
- $\mathit{sample}(\texttt{first\_match(}s\texttt{)}) = \mathit{sample}(s)$
- $\mathit{sample}(s\texttt{[*0]}) = \{\}$
- $\mathit{sample}(s\texttt{[*1:\$]}) = \mathit{sample}(s)$

□

The function *block* takes a sequence as input and returns a set of local variable names as output.

Definition A.1.2 (*block*)**.**

- $\mathit{block}(\texttt{(1, }v\texttt{ = }e\texttt{)}) = \{\}$
- $\mathit{block}(b) = \{\}$
- $\mathit{block}(\texttt{(}s\texttt{)}) = \mathit{block}(s)$
- $\mathit{block}(\texttt{(}s\texttt{ ##1 }t\texttt{)}) = (\mathit{block}(s) - \mathit{flow}(\{\}, t)) \cup \mathit{block}(t)$
- $\mathit{block}(\texttt{(}s\texttt{ ##0 }t\texttt{)}) = (\mathit{block}(s) - \mathit{flow}(\{\}, t)) \cup \mathit{block}(t)$
- $\mathit{block}(\texttt{(}s\texttt{ or }t\texttt{)}) = \mathit{block}(s) \cup \mathit{block}(t)$
- $\mathit{block}(\texttt{(}s\texttt{ intersect }t\texttt{)}) = \mathit{block}(s) \cup \mathit{block}(t) \cup (\mathit{sample}(s) \cap \mathit{sample}(t))$
- $\mathit{block}(\texttt{first\_match(}s\texttt{)}) = \mathit{block}(s)$

- $block(s\texttt{[*0]}) = \{\}$
- $block(s\texttt{[*1:\$]}) = block(s)$

$\square$

The function *flow* takes a set of local variable names and a sequence as input and returns a set of local variable names as output.

Definition A.1.3 (*flow*).

- $flow(X, \texttt{(1, } v \texttt{ = } e\texttt{)}) = X \cup \{v\}$
- $flow(X, b) = X$
- $flow(X, \texttt{(}s\texttt{)}) = flow(X, s)$
- $flow(X, \texttt{(}s \texttt{ \#\#1 } t\texttt{)}) = flow(flow(X, s), t)$
- $flow(X, \texttt{(}s \texttt{ \#\#0 } t\texttt{)}) = flow(flow(X, s), t)$
- $flow(X, \texttt{(}s \texttt{ or } t\texttt{)}) = flow(X, s) \cap flow(X, t)$
- $flow(X, \texttt{(}s \texttt{ intersect } t\texttt{)})$
  $= (flow(X, s) \cup flow(X, t)) - block(\texttt{(}s \texttt{ intersect } t\texttt{)})$
  $= (flow(X, s) \cup flow(X, t)) - (block(s) \cup block(t) \cup (sample(s) \cap sample(t)))$
- $flow(X, \texttt{first\_match(}s\texttt{)}) = flow(X, s)$
- $flow(X, s\texttt{[*0]}) = X$
- $flow(X, s\texttt{[*1:\$]}) = flow(X, s)$

$\square$

A.2 Tight satisfaction definition

The definition below is from Annex G, Subsection G.3.4, p. 350 of [SV]. The notation has been changed to conform to the conventions of these notes.

Definition A.2.1 (tight satisfaction).

- $w, L_0, L_1 \models\!\equiv \texttt{(1, } v \texttt{ = } e\texttt{)}$ iff $|w| = 1$ and $w^0 \Vdash \texttt{1}$ and

  $$L_1 = \{(v, e[L_0, w^0])\} \cup L_0|_{\mathrm{dom}(L_0)-\{v\}}$$

  where $e[L_0, w^0]$ denotes the value obtained from e by evaluating first according to L_0 and second according to w^0.
- $w, L_0, L_1 \models\!\equiv b$ iff $|w| = 1$ and $w^0 \Vdash b[L_0]$ and $L_1 = L_0$, where $b[L_0]$ denotes the expression obtained from b by substituting values from L_0.
- $w, L_0, L_1 \models\!\equiv \texttt{(}s\texttt{)}$ iff $w, L_0, L_1 \models\!\equiv s$.

- $w, L_0, L_1 \models$ `(`s `##1` t`)` iff there exist x, y, L' such that $w = xy$ and $x, L_0, L' \models s$ and $y, L', L_1 \models t$.

- $w, L_0, L_1 \models$ `(`s `##0` t`)` iff there exist x, y, z, L' such that $w = xyz$ and $|y| = 1$ and $xy, L_0, L' \models s$ and $yz, L', L_1 \models t$.

- $w, L_0, L_1 \models$ `(`s `or` t`)` iff there exists L' such that both of the following hold:
  - either $w, L_0, L' \models s$ or $w, L_0, L' \models t$, and
  - $L_1 = L'|_D$, where $D = \mathit{flow}(\mathrm{dom}(L_0), (s \texttt{ or } t))$.

- $w, L_0, L_1 \models$ `(`s `intersect` t`)` iff there exist L', L'' such that $w, L_0, L' \models s$ and $w, L_0, L'' \models t$ and $L_1 = L'|_{D'} \cup L''|_{D''}$, where

$$D' = \mathit{flow}(\mathrm{dom}(L_0), s) - (\mathit{block}((s \texttt{ intersect } t)) \cup \mathit{sample}(t))$$
$$D'' = \mathit{flow}(\mathrm{dom}(L_0), t) - (\mathit{block}((s \texttt{ intersect } t)) \cup \mathit{sample}(s))$$

- $w, L_0, L_1 \models$ `first_match(`s`)` iff both
  - $w, L_0, L_1 \models s$ and
  - if there exist x, y, L' such that $w = xy$ and $x, L_0, L' \models s$, then y is empty.

- $w, L_0, L_1 \models s$`[*0]` iff $|w| = 0$ and $L_1 = L_0$.

- $w, L_0, L_1 \models s$`[*1:$]` iff there exist $L_{(0)} = L_0, w_1, L_{(1)}, w_2, L_{(2)}, \ldots, w_j, L_{(j)} = L_1$ $(j \geq 1)$ such that $w = w_1 \cdots w_j$ and for every i such that $1 \leq i \leq j$, $w_i, L_{(i-1)}, L_{(i)} \models s$.

□