An Application of the WDL Theory of System Composition to the Analysis of Cryptographic Protocols

Dr. Alfred P. Maneki
NSA

I ABSTRACT

Researchers at Lockheed Martin's Western Development Laboratories (WDL, formerly LORAL Aerospace Corp.) have defined a model for the composition of communications systems which we will refer to in this paper as the WDL model. In this model, system components are defined as sets of input-output ordered pairs of "information units". An input-output pair belongs to a component if the output is the result of a specific "execution" of the component using the given input. Thus, a system component is characterized by the set of all of its possible input-output pairs. Note that for a "deterministic" component, there is a unique output for each possible input, while for a "indeterministic" or "probabilistic" component, there may be multiple outputs for each input.

In the WDL model, we denote by IU the set of all information units. The elements of IU may be considered as primitives or undefined elements, and the purpose of the model is to define component properties and to prove theorems about these properties. Since a component is a subset of IU x IU, we will frequently refer to the elements of $\mathcal{P}$(IU) x $\mathcal{P}$(IU) where $\mathcal{P}$ denotes the power set of IU. Given WDL components D_1 and D_2 (i.e. D_1 and $D_2 \subseteq \mathcal{P}$(IU) x $\mathcal{P}$(IU)) we may "combine" D_1 and D_2 using the standard set theoretic operations (i.e., $D_1 \cup D_2$ (union), $D_1 \cap D_2$ (intersection), and $D_1 \setminus D_2$ (set difference)).

The WDL model provides a fourth and more interesting combining in which input-output pairs from D_1 and input-output pairs from D_2 which share identical output and input iu $\in$ IU, i.e. output of D_1 equals input of D_2, are "wired" together to yield a new component whose input-output pairs are sets of inputs of D_1 and outputs of D_2. This composition, denoted D_1 o D_2 and which will be given a formal definition in the body of this paper, has interesting properties. For example, although "o" satisfies the associative law, "o" does not distribute over unions and intersections, and "o" is obviously a non-commutative operation.

The immediate applications for the WDL model were to define "security" properties for network components and to determine conditions under which these properties are inherited in composite systems (systems defined by "o"). Research in this area has been limited but shows great promise of yielding significant results.

In the current research, we are attempting to apply the theory of WDL system composition to the analysis of cryptographic protocols. A cryptographic protocol is a scheme for defining the exchange of messages that communicating parties must undertake to establish a session cryptographic key. This key is based on secrets held by the communicating parties and possibly "nonces" or random values that are generated by the communicating parties. These protocols should also serve the additional function of "authenticating" the identities of the communicating parties,

and may involve the use of a trusted key distribution center. Examples of such protocols include the Diffie-Hellman, Needham-Schroder, SANDIA, and Selective Broadcast protocols.

Unlike the previous research efforts in protocol analysis, which include logic-based “back-tracking” search techniques (i.e. searching for paths leading to insecure states or vulnerabilities), our effort involves the representation of protocols in terms of equations in the algebra that is inherently defined in the WDL system composition theory. This algebra is defined by the relations that are satisfied by the WDL component operators $\cup$, $\cap$, $\backslash$, and “o”. We are concerned with the message modification attacks of a “possibly legitimate” intruder who wishes to interrupt, intercept, or insert messages between the communicating parties but are transmitted in encrypted form.

To simplify our model, suppose that communicating party A wishes to establish session key K with communicating party B for the purpose of transmitting the message M. At a certain level of abstraction (i.e. the level at which the protocol is transparent to A and B), A transmits M and B receives M. At this level one element of the WDL system component is represented by the input-output pair $\delta = (M,M)$. This δ represents a single “execution” of the protocol, where an execution is determined by the message M and the session key K. Also at this level the entire protocol is to be considered as the union of its executions.

Starting with $\delta = (M,M)$, and under the assumption that the protocol operates without interference by an intruder, we are able to derive a WDL equation of the form

$$[(M,M)] = D_1 \text{ o } D_2 \text{ o } D_3$$

where D_2 includes all of the input-output pairs that are defined by the message exchanges of the protocol, and D_1 and D_2 are components defined suitably. It turns out that we are able to define similar equations when the protocol is being exploited by an intruder.

Using the WDL theory, we have converted the problem of searching for message modification attacks by an intruder to the equivalent problem of solving a class of equations in the WDL algebra. Unfortunately, at the time, very little is known about solving equations in the WDL algebra. However, we believe that the techniques we will derive in this paper merit further consideration, for there has to now been no attempt to consider protocol analysis from an algebraic point of view. We further believe that our techniques are quite general in that they can be used to represent all of the known intruder actions (i.e. passive attacks, active message switching attacks, and reachability analyses, etc.) Further research in this area should be encouraged.

II Notation and Terminology

Although the WDL composition theory can be applied to cryptographic protocols for more than two communicating parties, we will begin by considering protocols for two communication parties.

Let A and B denote the communicating parties, and suppose that A has a message M that it wishes to send to B in encrypted form. The protocol in question then is used to set up a two-way "session key" $K_{A,B}$ for this purpose. Starting with values (secret keys, nonces, etc) that are known uniquely to A and B, both parties must exchange a well-defined sequence of messages that would enable them to construct $K_{A,B}$. The final message that A sends to B is $E(K_{A,B}, M)$, the encryption of M with $K_{A,B}$. In the final step of the protocol B uses $K_{A,B}$ to recover the message M.

For purposes of authentication, the protocols we will consider require the use of public/private, encryption/decryption, pairs of keys denoted by (K,K^{-1}) in which K is privately held and K^{-1} is publicly known. Thus, we will assume that party A holds the secret key K_A, party B holds the secret key K_B and that the inverse keys K_A^{-1} and K_B^{-1} are publicly known.

Most protocols involve the generation of random numbers or "nonces." We will assume that the necessary secret keys and nonces are known to the respective communicating parties at the beginning of the protocol. Thus, we define

Known (A) = $\{K_A; r_{a,1}, r_{a,2}, \ldots, r_{a,n1}\}$
Known (B) = $\{K_B; r_{b,1}, r_{b,2}, \ldots, r_{b,n1}\}$

For representing cryptographic protocols in the WDL composition theory, it is useful to think of the protocol as a sequence (history) of messages that are exchanged between the communicating parties. At each "time" we must take into account what is known by a party, the message it has received, and the message it must transmit. Therefore we represent the protocol as a sequence of 2n ordered events as follows:

Time 1: A knows M and Known (A). A must generate message M_1, to be sent to B.
Time 2: B receives M_1; it knows Known (B) and will generate message M_2 and send to A.
Time 3: A knows M, Known (A), M_2 and will generate M_3 and send to B.
Time 4: B knows Known (B), M_1, M_2, M_3 and generates message M_4 and sends it to A

.

.

.

Time (2n-1) A knows M, Known(A), $M_1, M_2, \ldots M_{2n-2}$ and will generate $M_{2n-1} = E(K_{A,B},M)$, which will be sent to B.
Time 2n. B receives $E(K_{A,B},M)$ which it can decrypt to get M.

TABLE 1

III The WDL Model

The formal definitions and properties of the WDL composition theory are presented in the appendix. Readers who are unfamiliar with this theory should refer to the appendix or consult [1].

The task immediately confronting us is that of translating the 2n events in Table 1 into corresponding input/output WDL pairs. It is understood that messages and data items of the protocol are considered as the WDL "information units." We define (2n+1) input/output pairs $\delta_0, \delta_1, \ldots, \delta_{2n}$ in Table 2. The pairs $\delta_1, \ldots, \delta_{2n}$ correspond to the 2n events of Table 1. The utility of δ_0 will become evident shortly.

δ_0 ={M}; {Known (A), M_2, M_4; … M_{2n-2}}

δ_1 = {M, Known (A)} ; {M_1}

δ_2 = {Known (B), M_1} ; {M_2}

δ_3 = {M, Known (A), M_2} ; {M_3}

δ_4 = {Known (B), M_1, M_3} ; {M_4}

.

.

.

δ_{2n-2} = {Known (B), M_1, M_3 … M_{2n-3}} ; {M_{2n-2}}

δ_{2n-1} = {Known (A), M_2, M_4, … M_{2n-2}} ; {E(K_{AB},M)}

δ_{2n} = {E($K_{A,B}$,M)} ; {M}

TABLE 2

We will now arrange the (2n+1) pairs of Table 2 into three WDL components: First, P, and Last. For notational convenience let $\mu = (M_1, \ldots, M_{2n-2})$. We have:

First (A,M,Known (A),μ) = { δ_0 }

P(A,B,M,Known (A), Known (B), μ) = { $\delta_1, \delta_2, \ldots, \delta_{2n-1}$ }

Last (B, Known (B), μ) = { δ_{2n} }

The sequence of ordered pairs in Table 2 represents a single execution or instantiation of the protocol, i.e. a single "run" of the protocol with specified M, Known (A), and Known (B). We define an arbitrary instantiation of the protocol as the WDL component

PI (A,B,M, Known (A), Known (B), μ) =

First (A, M, Known (A), μ) $\bigcup P(A, B, M, Known(A), \mu)$

$\bigcup Last(B, Known(B), \mu)$ = { δ_0,, $\delta_1, \ldots, \delta_{2n}$ }

Finally, the totality of all instantiations of the protocol P is given by

$$\Pi(A,B) = \bigcup_{\substack{M,\, Known(A) \\ Known(B)}} PI(A,B,M, \text{Known}(A), \text{Known}(B), \mu)$$

The following two lemmas are easily proved from the definitions and notation we have just presented:

Lemma 1. For all messages M, for all Known (A), and all Known (B),

$\{(M, M)\}$ = First (A, M, Known (A), μ) o

P (A, B, M, Known (A), Known (B), μ) o

Last (B, Known (B), μ)

Lemma 2

$$\bigcup_{\substack{M,\, Known(A) \\ Known(B)}} First(A), M, Known(A), \mu)o$$

P(A, B, M, Known (A), Known (B), μ) o

Last (B, Known (B), μ)

is a subset of I = $\{(x,x) : x \varepsilon IU\}$

Remark 1. The significance of the ordered pair (M, M) in Lemma 1 is that whenever the protocol operates correctly, the message M which A wished to transmit to B in encrypted form, is properly received and decrypted by B.

Remark 2. If the protocol operates correctly and there is no penetration of the protocol by an intruder who is attempting to gain knowledge of M, then the set μ of messages that are transmitted between A and B is dependent only on Known (A) and Known (B). We include μ in the list of arguments for the functions First, P, and Last to give us the ability to represent penetrator actions in the protocol between A and B. In this case μ is a function of the actions of the intruder along with Known (A) and Known (B).

Lemma 3. For all $\delta_1, \delta_2, \ldots, \delta_{2n-1}$, in P (A, B, M, Known (A), Known (B), μ), we have Input $(\delta_1) \subseteq$ Input $(\delta_3) \subseteq \ldots \subseteq$ Input (δ_{2n-1}) and Input $(\delta_2) \subseteq$ Input $(\delta_4) \subseteq \ldots \subseteq$ Input (δ_{2n-2})

IV The Intruder

We will now attempt to use Lemma 1 to represent the protocol with penetrator actions. Let X denote a penetrator who wishes to recover knowledge of the message M, and Known (X) denote the information (message units) that X uses to modify the messages exchanged between A and B. If the protocol is defined in such a way that X can never recover the message M, then we are stating that the equation

(1) $\{(M, M)\}$ = First (A, M, Known (A), μ) o

$P(A, B, M, Known(A), Known(B), \mu) \circ Last(B, Known(B), \mu)$ has a unique solution in $Known(A), Known(B), \mu$.

More precisely, we must represent the protocol with penetrator actions as a pair of simultaneous equations. We distinguish the following two cases:

1. Suppose that X is a passive intruder, i.e. the protocol is designed in such a way that X is able to recover M simply by observing the messages exchanged between A and B. In this case the existence of the the passive attack is represented by a solution of the the pair of simultaneous equations.

(2) $\{(M, M)\} = First(A, M, Known(A), \mu) \circ$

$P(A, B, M, Known(A), Known(B), \mu) \circ$

$Last(B, Known(B), \mu)$ and

(3) $\{(M, M)\} = First(A, M, Known(A), \mu) \circ$

$P(A, X, M, Known(A), Known(X), \mu) \circ$

$Last(X, Known(X), \mu)$

having variables M, Known (A), Known (B), and Known (X). We include M as a variable to allow for the fact that the passive attack may exist for a certain class of messages.

Suppose X is an <u>active</u> intruder, i.e. the protocol is designed in such a way that X is able to recover M by modifying the messages that are transmitted between A and B. The existence of an active attack is represented by solutions to two sumultaneous equations, the first representing the successful operation of the protocol between A and B with the intruder's modified messages and the second representing exploctation of the protocol by the intruder. We denote by μ the modified messages that are transmitted between A and B. Thus, an active attack exists if the equations

(4) $\{(M, M)\} = First(A, M, Known(A), \mu) \circ$

$P(A, B, M, Known(A), Known(B), \mu) \circ$

$Last(B, Known(B), \mu)$ and

(5) $\{(M, M)\} = First(A, M, Known(A), \mu) \circ$

$P(A, X, M, Known(A), Known(X), \mu) \circ$

$Last(X, Known(X) \mu)$

have a solution in the variables M, Known (A), Known (B), Known (X), and μ. We again include M as a variable to represent the fact that the active attack may exist for a certain class of messages.

Remark 3. In some active attacks party B is cut out of the protocol, either because B's secret data have been compromised or because party A has decided at some point in the protocol that party X is party B. In this case the system of equations (4) and (5) reduces to the simple equation (5).

Remark 4. Although we have derived equations for two communicating parties, our techniques may be generalized to include other parties such as a Trusted Key Distribution Center. In the case of the Needham-Schroeder protocol, which involves two communicating parties and the Key Distribution Center, we still have a system of two simultaneous equations to represent the exploitation of the protocol by the intruder.

Remark 5. We want to emphasize the point that a solution μ for equations (2) and (3) and for a pair of solutions (μ , μ') for equations (4) and (5) must consist of sequences of messages which can be derived from Known (A), Known (B), Known (X), and previous messages in the sequence. It is not enough for these message sequences merely to be solutions of the corresponding WDL equations. Indeed, all message sequences of the form $\mu = (Y_1, Y_2, \ldots\ Y_{2n-2})$ will generate a sequence $\delta_0, \delta_1, \ldots, \delta_{2n}$ which will satisfy all the equations 1-5. Stating this matter in another way, equations 1-5 are identities.

Remark 6. Short of a breakthrough in this area of research, we must still rely on automated tools such as the Interrogator for assistance with protocol analysis. It is interesting to note, however, that in some sense, the Interrogator attempts to solve the equations we have defined in this section. Using an exhaustive backwards search technique, the Interrogator starts by "guessing" the final message in μ and deducing what possible previous messages in μ could have produced that final message.

V CONCLUSION

In this paper, we have developed a representation for cryptographic protocols in the WDL theory of system composition. However, as we have seen in Remark 5, the search for protocol vulnerabilities cannot be found merely by solving the equations which represent the actions of the legitimate users and the intruder. It now appears that a totally algebraic approach to protocol analysis would require either an extension of the WDL theory or an application of an entirely unrelated algebraic structure.

We intend to continue these investigations in the analysis of cryptographic protocols.

VI REFERENCES

1. G. W. Dinolt, L. A. Benzinger, M. G. Yatabe. Combining Components and Policies. In Proceddings of teh Computer Scurity Foundations Workshop VII, pages 22-23, Franconia, NH, June, 1994. IEEE Computer Security Society.

2. Jonathan K. Millen. The Interrogator User's guide for Sun OS/X11. The MITRE Corporation, September, 1995.

3. Catherine Meadows. Thr NRL Protocol Analyzer: An Overview. In Proceedings of the Second International Conference on the Practical Applications of Prolog, 1994.

4. G. B. Purdy, G, J. Simmons, and J. A. Studier, "A Software protection scheme," in Proceedings 1982 Symposium on Security and Privacy, IEEE Computer Society, Catlog No. 85CH2150-1.

APPENDIX A - Examples

In this appendix we represent three protocols (Needham-Schroeder, TMN, and Sandia) and known attacks against them in the equations of the WDL composition theory. Descriptions of these protocols have been taken from reference [1].

1. Needham-Schroeder Protocol

Communicating parties A and B use a trusted key distribution center (KDC). In this protocol, A and B need to develop $K_{A,B}$ so that A can send $E(K_{A,B}, M)$ to B which B decrypts using two way authentication.

Step 1. A sends to KDC: (A,B)

Step 2: KDC sends to A: $E(K_A,(B,K_{A,B},E(K_B,(A,K_{A,B}))))$

Step 3: A sends to B: $E(K_B,(A,K_{A,B}))$

Step 4; B send to A: $E(K_{A,B},I_B)$

Step 5: A sends to B: $E(K_{A,B},(I_B\text{-}1,I_B))$

Step 6: B sends to A: $E(K_{A,B},I_A\text{-}1)$

Step 7: A sends to B: $E(K_{A,B},M)$

Step 8: B decrypts $E(K_{A,B},M)$ to receive M

In WDL notation:

Known(KDC) = { K_A, K_B, $K_{A,B}$ }
Known(A) = { I_a, I_A, K_A, M}
Known(B) = { I_B, K_B}

δ_1 = { M, Known(A) } ; { M_1 = { A, B} }
δ_2 = { Known(KDC), M_1 } ; { $M_2 = E(K_A,(B,K_{A,B},E(K_B,(A,K_{A,B}))))$ }
δ_3 = { M, Known(A), M_2 } ; { $M_3 = E(K_B,(A,K_{A,B}))$ }
δ_4 = { Known(B), M_3 } ; { $M_4 = E(K_{A,B}, I_B)$ }
δ_5 = { M, Known(A), M_2 , M_4 } ; { $M_5 = E(K_{A,B}, (I_B\text{-}1,I_A))$ }
δ_6 = { M, Known(B), M_3 , M_5 } ; { $M_6 = E(K_{A,B}, I_A\text{-}1)$ }
δ_7 = { M, Known(A), M_2 , M_4 , M_6} ; { $M_7 = E(K_{A,B}, M)$ }
δ_8 = { M } ; { M, Known(A), M_2 , M_4 , M_6 }

$\delta_9 = \{ M \} ; \{ M_1 , M_2 , M_3 , M_4 , M_5 , M_6 \}$
$\mu = \{ M_1 , M_2 , M_3 , M_4 , M_5 , M_6 \}$

For the attack, assume that the intruder X knows a previously used key $K'_{A,B}$ and $E(K_A,(B,K'_{A,B},E(K_B,(A,K'_{A,B}))))$. Then for the attack, we have

Step 1. $A \rightarrow KDC$: { A, B }

Step 2: $KDC \rightarrow A$: $E(K_A,(B,K_{A,B},E(K_B,(A,K_{A,B}))))$

X intercepts step 2 and send to A

Step 2': $X \rightarrow A : E(K_A,(A,B,K'_{A,B},E(K_B,(A,K'_{A,B}))))$

(Note: A believes that X's message (step 2') came from KDC.)

Step 3' $A \rightarrow B : E(K_B,(A,K'_{A,B}))$

Step 4' $B \rightarrow A : E(K'_{A,B},I_B)$

Step 5' $A \rightarrow B : E(K'_{A,B},(I_B\text{-}1,I_A))$

Step 6' $B \rightarrow A : E(K'_{A,B},I_A\text{-}1)$

Step 7' $A \rightarrow B : E(K'_{A,B},M)$

Step 8: B decrypts $E(K'_{A,B},M)$ to recover M.

Step 8': X intercepts $E(K'_{A,B},M)$ to recover M.

From this attach, the solution to equation (5) can be constructed from

$\mu' = \{ M'_1 , M'_2 , \ldots , M'_7 \}$

where

$M'_1 = \{ A, B \}$
$M'_2 = E(K_A,(A,B,K'_{A,B},E(K_B,(A,K'_{A,B}))))$
$M'_3 = E(K_B,(A,K'_{A,B}))$
$M'_4 = E(K'_{A,B},I_B)$
$M'_5 = E(K'_{A,B}(I_B\text{-}1,I_A))$
$M'_6 = E(K'_{A,B},I_A\text{-}1)$
$M'_7 = E(K'_{A,B},M)$

2. TMN Protocol

This protocol is a key distribution scheme by which a pair of ground stations in a mobile communications system obtains a common session key from a trusted server. Public key encryption is used to talk to the server, which uses an exclusive-or mode of encryption to reply. The session key encrypts data in a conventional single key modes. The basic problem with the protocol is that a third ground station that knows the public key of the server can pose as different station. [Interrogator Users Guide, p. 54].

The message exchange for this protocol is as follows: (A, B are mobile ground stations, S = trusted server, K_P is S's public encryption key which only S can decrypt,R_1 , R_2 are random numbers.)

Step 1: $A \rightarrow S$: $[B, E(K_P,R_1)]$
Step 2: $S \rightarrow B$: $[A]$
Step 3: $B \rightarrow S$: $[E(K_P ,R_2)]$
Step 4: $S \rightarrow A$: $[R_1 \oplus R_2]$
Step 5: $A \rightarrow B$: $[E(R_2,M)]$
Step 6: B decrypts $E(R_2,M)$ with R_2 to recover M

For the attack, replace the message $[E(K_P ,R_2)]$ in Step 3 with the message $[E(K_P,R_3)]$. Then the message exchanged for the attack is:

Step 1: $A \rightarrow S$: $[B, E(K_P ,R_1)]$
Step 2: $S \rightarrow B$: $[A]$
Step 3': $B \rightarrow S$: $[E(K_P ,R_3)]$
Step 4': $S \rightarrow A$: $[R_1 \oplus R_3]$
Step 5': $A \rightarrow X$: $[E(R_3,M)]$
Step 6': X decrypts $E(R_3,M)$ to recover M.

To construct equations (4) and (5),
let $\mu = \{ [B, E(K_P ,R_1)],[A],[E(K_P ,R_2)],[R_1 \oplus R_2],[E(R_2,M)]\}$
and $\mu' = \{ [B, E(K_P ,R_1)],[A],[E(K_P ,R_3)],[R_1 \oplus R_3],[E(R_3,M)]\}$

3. SANDIA Protocol

This is a software distribution protocol [2, p 57]. It enables a vendor to broadcast a software product in encrypted form, yet select which customers can decrypt it. Each customer is given a special tamperproof “Secure Crypto Module” (SCM) concerning a different “unique” key. A customer entitled to decrypt the software is given another key that combines the customer’s unique with the key used to encrypt the software. The customers and vendor all have the same kind of SCM, which has three functions: key generation encryption and decryption. The normal message exchange for this protocol is as follows:

m is the common key for all SCM’s
qc is the customer c’s unique key
qv is the vendor v’s unique key
sw is the software
j is a key used to encrypt the software

Step 1: $V \rightarrow C : E(qv \oplus D(m,j),sw)$
Step 2: $C \rightarrow V : E(m,qc)$
Step 3: $V \rightarrow C : E(m,qc \oplus qv \oplus D(m,j))$
Step 4: $C \rightarrow SCM_C : [E(m,qc \oplus qv \oplus D(m,j)),E(qv \oplus D(m,j),sw)]$
Step 4: $SCM_C \rightarrow C : (sw)$

An attack against this protocol is due in part of the fact that the customers and vendors all have the same kind of SCM. A customer X who had not received a decryption key could use the generation function of his own SCM with a key sent by the sender to another customer C, and produce his own key, i.e., X can pass the messages obtained in steps 2 and 3 to its own SCM to recover sw.

In the WDL representation, we would introduce two additional messages exchanges between steps 3 and 4 representing $SCM_X \rightarrow X$. Also, in the WDL model we would solve both equations (4,5) because the communication between C and V still goes through; they are not aware of what X is doing. Details of the construction of these equations is left as an exercise.

APPENDIX B – A Survey of the WDL Model

1. Introduction

We list here the definitions and algebraic properties of the WDL model for information flows. Readers interested in a thorough treatment of this subject should consult [].

2. Undefined Terms

Denote by IU a set of abstract entities called information units. One might think of an information unit, denoted by iu, as the basic abstraction of what is manipulated in our system. In addition to information units, we introduce two more undefined terms that are a part of a system security model: data and security level.

DATA: This is the set of all possible data

SL: This is the set of all security labels

The set SL of security labels with the partial order relation, $\leq$, form a lattice. The set SL contains a least element denoted by syslo.

3. Definitions

We let $\mathbf{D}$ = P(IU) x P(IU) where P(IU) is the power set of IU. If $\delta \in \mathbf{D}$ we write δ = (A, B) where A, B $\subseteq$ IU. We designate A the input set and B the output set. A "system" D is a subset of $\mathbf{D}$, D $\subseteq \mathbf{D}$ = P(IU) x P(IU).

4. Functions

Define Inputs_1: $\mathbf{D} \rightarrow$ P(IU) and Outputs_1: $\mathbf{D} \rightarrow$ P(IU). For $\delta \in \mathbf{D}$, Inputs_1 and Outputs_1 are defined by $\delta = (\text{Inputs}_1(\delta), \text{Outputs}_1(\delta))$.

Let $\mathbf{X} \subseteq \mathbf{D}$. Define

$$\text{Inputs}(\mathbf{X}) = \bigcup_{x \in \mathbf{X}} \text{Inputs}_1(x) \text{ and}$$

$$\text{Outputs}(\mathbf{X}) = \bigcup_{x \in \mathbf{X}} \text{Outputs}_1(x).$$

If $\mathbf{X} = \{\delta\}$, we abuse the notation slightly and write Inputs(δ) for Inputs($\{\delta\}$) and Outputs(δ) for Outputs($\{\delta\}$).

Let $data_1$: IU $\rightarrow$ DATA and data: P(IU) $\rightarrow$ DATA, then for A $\subseteq$ IU,

$$data(A) = \bigcup_{iu\, \varepsilon\, A} data_1(iu)$$

Let sl_1: IU -> SL, then for A $\subseteq$ IU, let

$$sl(A) = \begin{cases} \underset{iu\, \varepsilon\, A}{\text{lub}}\ sl_1(iu) \\ \text{syslo if } A = \{\} \end{cases}$$

5. Component Combination

If $D_1, D_2 \subseteq \mathbf{D}$ then $D_1 \cup D_2$; $D_1 \cap D_2$; and $D_1 - D_2$ are respectively, the set theoretic union, intersection, and difference of the components of D_1 and D_2.

For a more complex form of combinations, we would like something for components analogous to the composition of functions. In our method of component combinations there are three parts:

1. the inputs to the composition of $D_1 \bullet D_2$ are related to the inputs of D_1

2. the outputs of the composition of $D_1 \bullet D_2$ are related to the outputs of D_2

3. the outputs of D_1 and the inputs to D_2 are related.

Let $D_1, D_2 \subseteq \mathbf{D}$. We define D_1 o D_2 = { $\delta \mid \exists\ X_\delta \subseteq D_1, Y_\delta \subseteq D_2$ where Inputs(δ) = Inputs(X_δ), Outputs(δ) = Outputs(Y_δ) and Outputs(X_δ) = Inputs(Y_δ)}

6. Composition Properties

Theorem 1: If $D_1, D_2, D_3 \subseteq \mathbf{D}$, then

(1) $(D_1 \bullet D_2) \bullet D_3 = D_1 \bullet (D_2 \bullet D_3)$

(2) $D_1 \subseteq D_2$ implies $D_1 \bullet D_3 \subseteq D_2 \bullet D_3$

(3) $D_1 \subseteq D_2$ implies $D_3 \bullet D_1 \subseteq D_3 \bullet D_2$

(4) $D_1 \bullet (D_2 \cap D_3) \subseteq (D_1 \bullet D_2) \cap (D_1 \bullet D_3)$

(5) $(D_1 \cap D_2) \bullet D_3 \subseteq (D_1 \bullet D_3) \cap (D_2 \bullet D_3)$

(6) $(D_1 \bullet D_2) \cup (D_1 \bullet D_3) \subseteq D_1 \bullet (D_2 \cup D_3)$

(7) $(D_1 \bullet D_3) \cup (D_2 \bullet D_3) \subseteq (D_1 \cup D_2) \bullet D_3$

7. Closure and Identity

If $D \subseteq \mathbf{D}$, then D is **closed** if for any $X \subseteq D$, we have (Inputs(X), Outputs(X)) ε D. If $D \subseteq \mathbf{D}$, then $\overline{D} = \{ \delta \, \varepsilon \, \mathbf{D} \mid \exists\, X \subseteq D$ for which Inputs(δ) = Inputs(X) and Outputs(δ) = Outputs(X)$\}$. $\overline{D}$ is called the **closure** of D.

The set $I = \{ \delta \, \varepsilon \, \mathbf{D}\mathbf{:} \mid$ Outputs(δ) = Inputs(δ)$\}$ is termed the identity relation of $\mathbf{D}$.

Theorem 2: For all $D \subseteq \mathbf{D}$

(1) $D \subseteq I \bullet D = D \bullet I = \overline{D}$

(2) $I = I \bullet I = \overline{I}$

Theorem 3: For all $D, D_1, D_2 \subseteq \mathbf{D}$, we have

(1) $\overline{\overline{D}} = \overline{D}$

(2) $D_1 \bullet D_2 = \overline{D_1} \bullet D_2$

(3) $D_1 \bullet D_2 = D_1 \bullet \overline{D_2}$

(4) $D_1 \bullet D_2 = \overline{D_1 \bullet D_2}$

(5) $D_1 \bullet D_2 = \overline{D_1} \bullet \overline{D_2}$

(6)D is closed $\Leftrightarrow D = \overline{D}$

(7) D is closed $\Leftrightarrow I \bullet D = D \bullet I = D$

(8) $\overline{D_1} \cap \overline{D_2} = \overline{D_1 \cap D_2}$

(9) $\overline{D_1} \cup \overline{D_2} = \overline{D_1 \cup D_2}$

8. Transitivity

Let $S \subseteq \mathbf{D}$. If A, B ε P(IU), then we write $A \sim_S B$ to mean (A,B) ε S. Let $S \subseteq \mathbf{D}$. If $A \sim_S B$ and $B \sim_S C$ implies $A \sim_S C$ then we say that S is transitive.

Theorem 4: Let $\overline{D_1}$, $\overline{D_2}$ $\subseteq P \subseteq \mathbf{D}$. Let P be transitive. Then

$$D_1 \bullet D_2 \subseteq \overline{D_1} \bullet \overline{D_2} = \overline{D_1 \bullet D_2} \subseteq P.$$

Corollary 1: If $P \subseteq \mathbf{D}$ is transitive and closed, then $P \bullet P \subseteq P$.

Corollary 2: If $P \subseteq \mathbf{D}$ is transitive and closed, and $D_1, D_2 \subseteq P$ then $D_1 \bullet D_2 \subseteq P$.

Lemma 1: Let $P_1, P_2 \subseteq \mathbf{D}$ be closed and transitive. The $P_1 \cap P_2$ is closed and transitive.

Theorem 5: Let $P_1, P_2 \subseteq \mathbf{D}$. If P_1 and P_2 are closed and transitive, then

$$(P_1 \cap P_2) \bullet (P_1 \cap P_2) \subseteq (P_1 \cap P_2)$$

Corollary 3: Let $P_1, P_2 \subseteq \mathbf{D}$ and suppose that $D_1, D_2 \subseteq (P_1 \cap P_2)$. If P_1 and P_2 are closed and transitive, then

$$D_1 \bullet D_2 \subseteq (P_1 \cap P_2)$$

Theorem 6: Let $P \subseteq \mathbf{D}$. If for all $D_1, D_2 \subseteq P$, $D_1 \bullet D_2 \subseteq P$, then P is transitive.

Corollary 4: Let $P \subseteq \mathbf{D}$. If $P \bullet P \subseteq P$, then P is transitive.