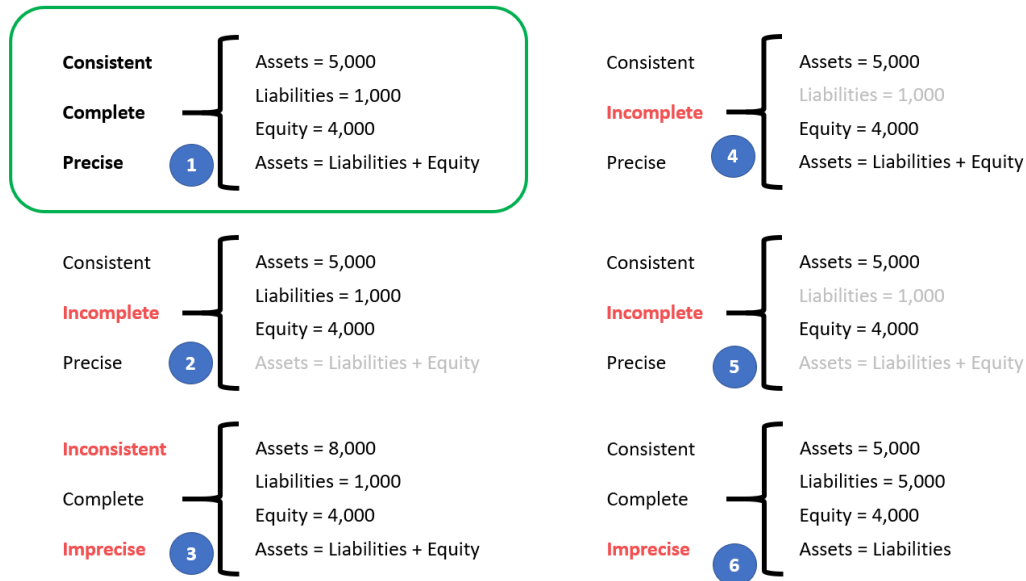


Understanding Errors that Can Occur which Method Detects and Prevents (Comparison of States)

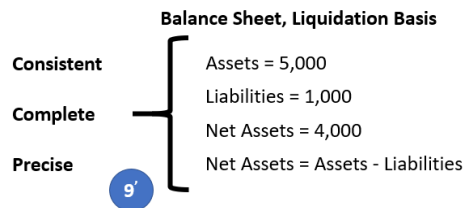
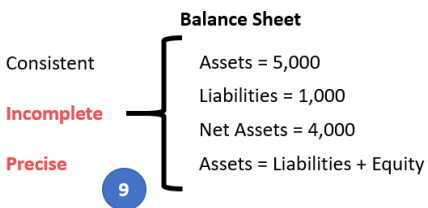
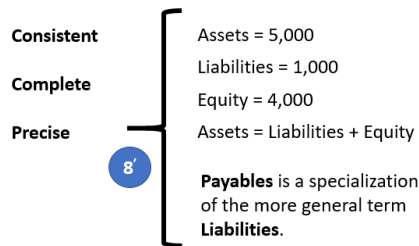
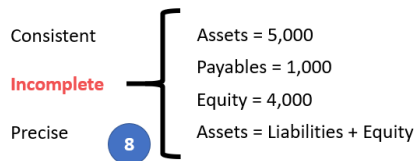
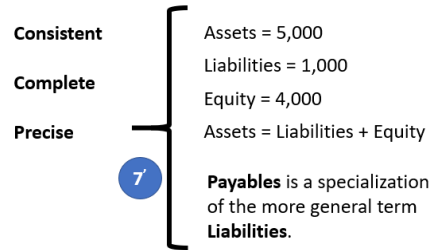
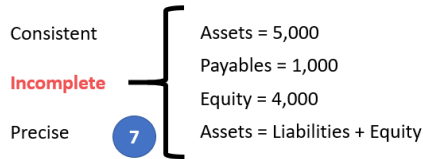
You can understand the types of errors that can occur in the Microsoft 10-K or any US GAAP or IFRS based XBRL-based financial report by looking at the types of errors that can exist in the accounting equation representation. This section explains those error types.

The following is a comparison of 9 states of the same simple financial report logical system, the accounting equation¹. The point of using such a simple financial report logical system is to explain specific things that can go wrong so that a reader can understand why each of the categories of rules are necessary. These 9 states can occur in any financial report with one fragment, two fragments, or 194 fragments like the Microsoft 10 K.

There are many ways to get a report wrong. Here is a summary of all nine states with the first state outlined in green being the only properly functioning logical system proven to be complete, consistent, and precise:



¹ Accounting equation, <http://xbrlsite.azurewebsites.net/2020/master/ae/index.html>



In the following sections I want to make some adjustments to the logical system which make the logical system either inconsistent, incomplete, or imprecise and explain why the system is then not a properly functioning logical system. I made videos that explain each of these impediments to a properly functioning logical system which you can see in this video playlist, *Understanding the Financial Report Logical System*².

Before we get to the improperly functioning logical systems, let's take one final look at the properly functioning logical system so that you can use that as a baseline for comparing and contrasting the properly functioning and improperly functioning logical systems so that you can understand they sorts of errors that could occur in any XBRL-based financial report.

State 1: Properly Functioning Logical System

For completeness, I want to start by mentioning again our properly functioning logical system which is consistent, complete, and precise. It can be helpful to contrast other states to this state to understand the difference between properly functioning logical systems and improperly functioning systems.

² Understanding the Financial Report Logical System, https://www.youtube.com/playlist?list=PLqMZRUzQ64B7EWamzDP-WaYbS_WORL9nt

Balance Sheet [Abstract]	Period [Axis]
	2020-12-31
Balance Sheet [Abstract]	
Assets	5,000
Liabilities	1,000
Equity	4,000

Result	Rule
Pass	\$Assets = \$Liabilities + \$Equity

Balance Sheet

Consistent

Complete

Precise

1

Assets = 5,000
Liabilities = 1,000
Equity = 4,000
Assets = Liabilities + Equity

Again, this is considered a properly functioning logical system because (a) all the statements within the system are **consistent**; (b) the set of statements that describe the system is **complete**; and (c) the information conveyed by the system is **precise** in its representation of reality. Further, we are formally declaring this “reality”³ to be our base understanding.

Also, we need to be explicit. We defined three terms “Assets”, “Liabilities”, and “Equity”.

Now, you may know what those three terms are; but a computer does not. You have to define what you work with relative to something that you know. Imagine our system defines four terms, “fac:Assets”, “fac:Liabilities”, “fac:Equity”, and “fac:LiabilitiesAndEquity”⁴. You understand your system but you have to map every external system into your system⁵. Your internal system understands more than the accounting equation system (i.e. you have LiabilitiesAndEquity and the rule “LiabilitiesAndEquity = Liabilities + Equity”). You have to be able to compute that value based on some other system’s information⁶. It is perfectly reasonable for our system to create a concept LiabilitiesAndEquity and compute that value even though the accounting equation logical system does not have that explicit value.

The point is that different economic entities have different models; but all models of a financial reporting scheme are reconcilable from/to one another in some manner⁷.

State 2: Incomplete Coverage by Rules

The logical system #2 below is intended to show exactly the same information as our #1 properly functioning logical system, except that #2 leaves out the rule “Assets = Liabilities and

³ YouTube, *Reality*, <https://youtu.be/eq2Jw6waaCl>

⁴ Fundamental accounting concepts, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac.xsd>

⁵ Mapping from accounting equation to fundamental accounting concepts in our system, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml>

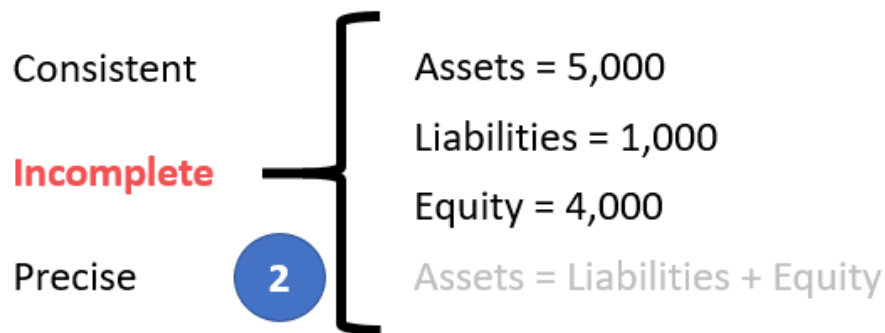
⁶ XBRL Formula to derive the value for LiabilitiesAndEquity, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-ImputeRule-LiabilitiesAndEquity-formula.xml>

⁷ Charles Hoffman, CPA, *Special Theory of Machine-based Automated Communication of Semantic Information of Financial Statements*, <http://xbrl.squarespace.com/journal/2019/12/30/special-theory-of-machine-based-automated-communication-of-s.html>

Equity” which is showed as grayed out (i.e. because it is assumed to be missing from the logical system.

Coverage is a measure of how well you **do** or **can** represent a domain of information within a logical system. “Do” is about using the tools you have correctly and effectively. “Can” is about the capabilities of the tools you are using to represent the rule.

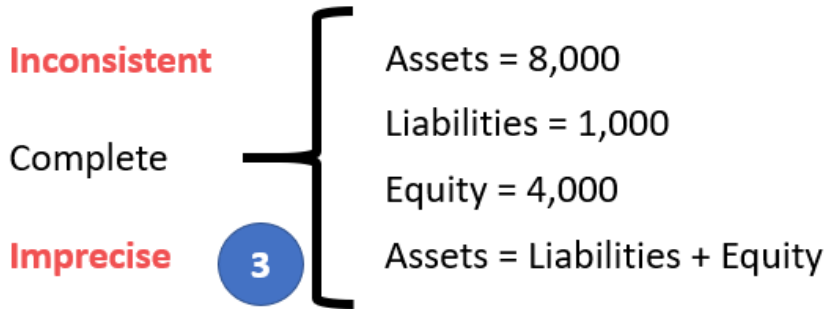
For example, if your logical system neglects to include the rule “Assets = Liabilities + Equity” or if your tools don’t provide the capabilities to allow you to represent that rule; then there is the possibility that the facts being represented to be represented incorrectly and the system will not detect the inconsistency. As such, that logical system has **incomplete coverage**.



While this specific state #2 does have the Assets, Liabilities, and Equity facts consistent with the absent rule; the system is still incomplete because the coverage can be improved by adding the missing rule. If that missing rule is added, then the logical system can be considered complete again.

State 3: Inconsistent and Imprecise

All the statements in the system must be consistent for the logical system to be considered properly functioning. If statements are inconsistent, the logical system is not properly functioning. In this system #3, the values for Assets, Liabilities, and Equity are inconsistent with the rule “Assets = Liabilities + Equity”. From looking at the information provided, it is impossible to know exactly which of the three facts are incorrect; it is only possible to understand that the statements made within the logical system is inconsistent. It could be the case that the rule is incorrect.



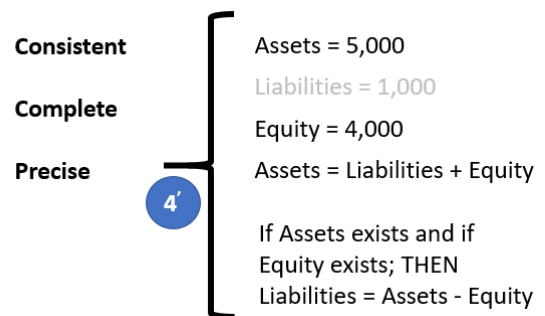
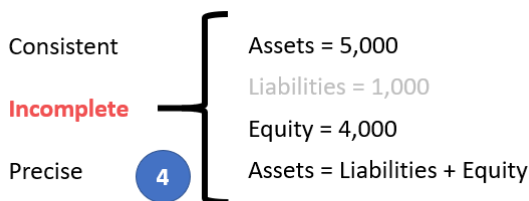
However, given that we know from state #1 that the value for Assets is 5,000 and not 8,000; the facts in this system is imprecise because the fact for Assets does not reflect reality. (We have, for the purposes of explaining these examples, defined reality and in that reality Assets = 5,000.)

State 4: Unreported Facts

In state #4, the situation is that the economic entity representing information in their report neglected to include the fact for Liabilities. Whether it is the case that a fact can, or cannot, be left unreported is a decision that can be made by the stakeholders of the system.

If it is the case that it is decided that the fact “Liabilities” can be omitted if both Assets and Equity are reported; then you must provide a rule to derive the value of Liabilities when that fact is not reported.

Below you see that the system has been adjusted in state #4’ to add the rule “IF Assets exists and if Equity exists; THEN Liabilities = Assets - Equity”⁸. (NOTE that this rule should actually be “IF Assets exists and if Equity exists and if not(exists) Liabilities; THEN Liabilities = Assets - Equity”)



⁸ Here is the impute or derivation rule that would be added to the accounting equation logical system for this situation, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-1-Code-BS-Impute-01-formula.xml>

If it were likewise true that either Assets⁹ or Equity¹⁰ could also be left unreported, similarly derivation rules could be created for each of those facts. Note that XBRL Formula chaining¹¹ can be used to physically derive unreported facts if any one of these three facts remain unreported. Note that it is impossible to derive missing information if any two of the facts remain unreported. Adding the derivation rule makes the system complete.

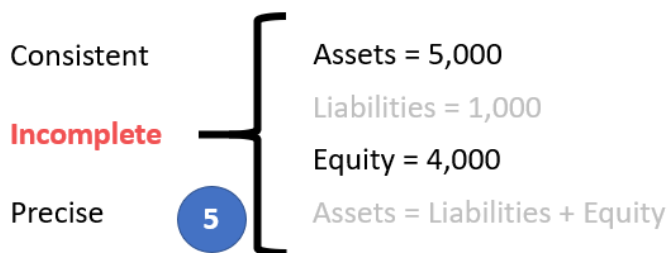
Allowing certain line items of a report to go unreported specifies the need to create rules to derive missing information. Or saying this another way, omitting the possibility of unreported facts negates the need for creating derivation rules.

A second downside of allowing unreported facts is that you lose the parity check or cross check if facts can go unreported. Said another way, it would be considered good or best practice to not leave important high-level financial report line items to go unreported.

State 5: Incomplete

Similar to state #4, in state #5 the logical system is incomplete because both (a) the fact Liabilities is unreported and also (b) the consistency rule “Assets = Liabilities + Equity” is missing from the logical system. Because both a fact and the rule are missing from the logical system, it would be impossible to deduce the value of Liabilities in this case. There is not enough information in the logical system to allow Liabilities to be derived.

At a minimum, a consistency crosscheck rule¹² plus the derivation rule to impute Liabilities¹³ would be necessary.



⁹ XBRL Formula rule for deriving Assets, <http://xbrl.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-3-Code-BS-Impute-03-formula.xml>

¹⁰ XBRL Formula rule for deriving Equity, <http://xbrl.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-2-Code-BS-Impute-02-formula.xml>

¹¹ Deriving Facts Using XBRL Formula Chaining (Example), <http://xbrl.squarespace.com/journal/2019/4/24/deriving-information-using-xbrl-formula-chaining-example.html>

¹² XBRL Formula consistency crosscheck rule Assets = Liabilities + Equity, <http://xbrl.azurewebsites.net/2020/core/master-ae/Consistency-5-Code-BS01-formula.xml>

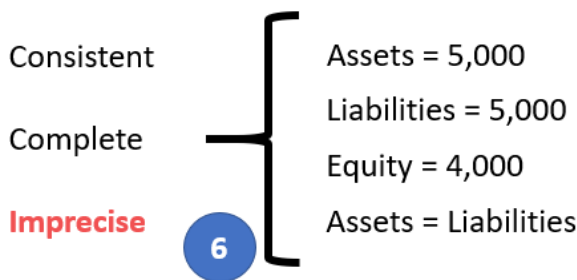
¹³ XBRL Formula derivation rule to impute Liabilities, <http://xbrl.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-1-Code-BS-Impute-01-formula.xml>

Again, consistent with state #4; Assets and Equity would require similar rules and there is no parity check of reported information.

State 6: Imprecise

A logical system is a true and fair representation of some agreed upon realism within some area of knowledge. **Precision** is a measure of how precisely you do or can represent the information of a domain within a logical theory. The reality that we formalized in state #1 indicates that “Assets = Liabilities + Equity”. Yet, in the state #6 example, the rule “Assets = Liabilities” was provided. Further, the values of Assets and Liabilities are, in fact, consistent with the rule that has been provided.

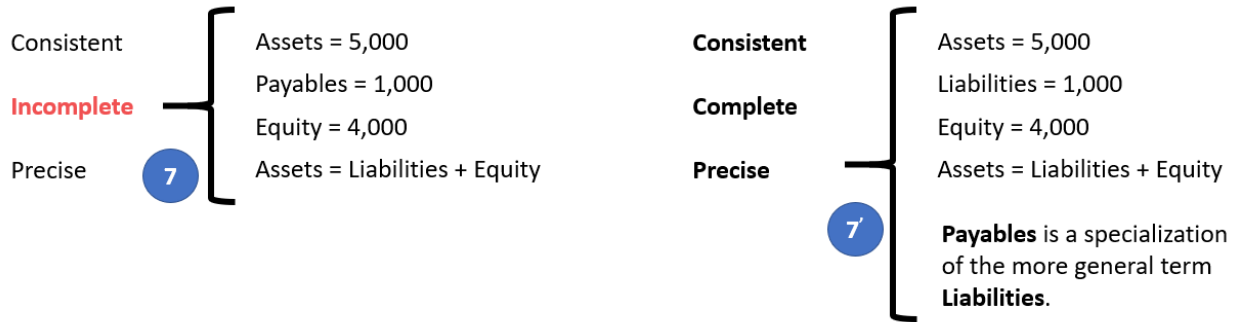
Remember that in state #1 we formalized our truth to be that “Assets = Liabilities + Equity”. As such, this logical system can be described as being imprecise. To make this logical system precise, all that needs to be done is to fix the rule “Assets = Liabilities” and make that rule consistent with our reality which states that “Assets = Liabilities + Equity”.



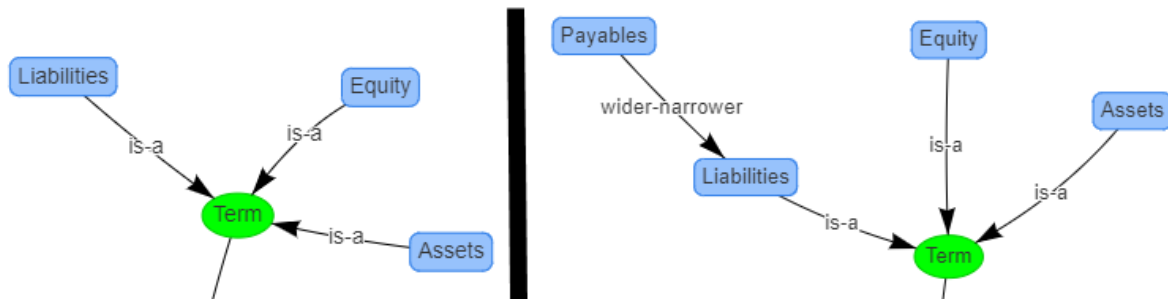
State 7: Extension Concept

In state #7 on the left, what we are trying to convey is that the economic entity reported the fact for Liabilities using the extension concept “Payables” that it had created. If a fact is represented using an extension concept created by a reporting entity; then a “general-special” or “wider-narrower” or “class-equivalentClass” association must be created to indicate to software applications of the relationship so that information can be used correctly. State #7’ on the right, the rule “Payables is a specialization of the more general term Liabilities” has been added to the logical system which allows the system to operate effectively¹⁴.

¹⁴ XBRL Definition relations showing example of a mapping rule, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml>



And so, the graphic below shows a fragment of the knowledge graph on page 9 above before and after the information that “Payables is a specialization of the more general term Liabilities,” was added. On the left you see State 7, the taxonomy before the information was added and on the right you see “Payable” being added as an extension concept indicating that there is a “wider-narrower” relationship between Payables and Liabilities. Therefore, a machine based process can utilize the information per State 7’ because the process understands Liabilities in the taxonomy, understands the “wider-narrower” relationship therefore knowing that “Payables” is a type of Liability.

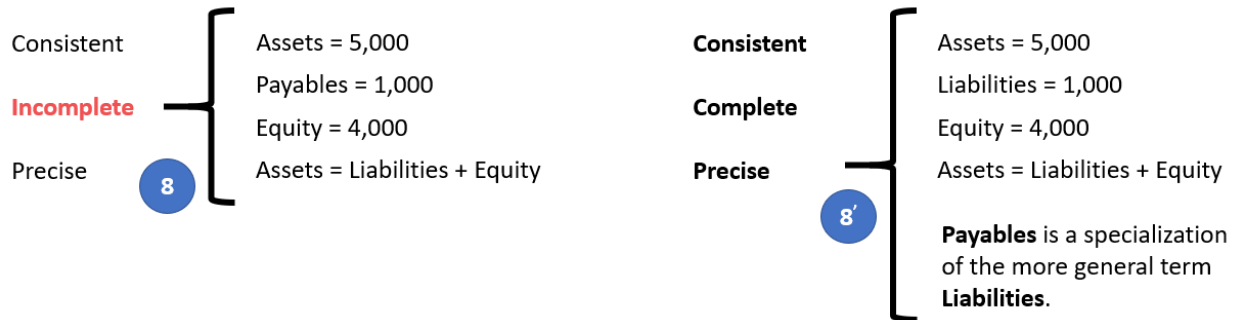


State 8: Base Taxonomy Wider/Narrower Concept Use

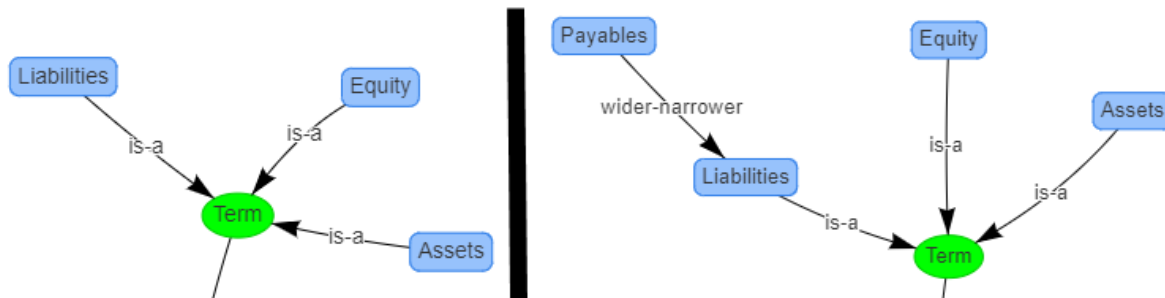
State #8 on the left below is similar to state #7 in that a different concept is used to report a fact; but while state #7 focuses on using an extension concept; state #8 points out that using a wider or narrower base taxonomy concept gives exactly the same result.

Now, our base state #1 does not have the concept “Payables”; but let’s assume for a moment that it does have the concept “Payables”. Also suppose that there was no information in the base logical system indicating the relationship between “Payables” and any other concept. If a fact is represented using a BASE TAXONOMY CONCEPT by a reporting entity; then a “general-special” or “wider-narrower” or “class-equivalentClass” association must exist in that base taxonomy to indicate that some concept is a permissible alternative for some other concept.

State #8' on the right adds the rule "Payables is a specialization of the more general term Liabilities"¹⁵.



And so, the graphic below shows a fragment of the knowledge graph on page 9 above before and after the information that "Payables is a specialization of the more general term Liabilities," was added. On the left you see State 8, the base taxonomy before the information was added and on the right you see "Payable" being added as an extension concept indicating that there is a "wider-narrower" relationship between Payables and Liabilities. Therefore, a machine based process can utilize the information per State 8' because the process understands Liabilities in the base taxonomy, understands the "wider-narrower" relationship therefore knowing that "Payables" is a type of Liability, and therefore can understand what you are conveying.



State 9: Defining a Completely New Structure

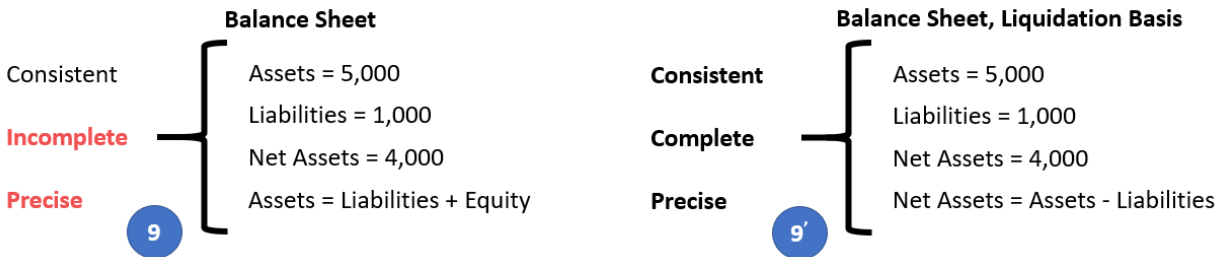
State #9 below on the left focuses on the structure as contrast all the prior examples which focused on the terms and rules. If a new structure is created, the new structure must be referenced to the base taxonomy and the new structure needs to be explained using machine-readable rules¹⁶. Even base taxonomy structures need to be defined in order to be referred

¹⁵ XBRL Definition relations showing example of a mapping rule, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-mapping-definition.xml>

¹⁶ XBRL Definition relations used to represent structure rules, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/dm-1355-rules-def.xml>

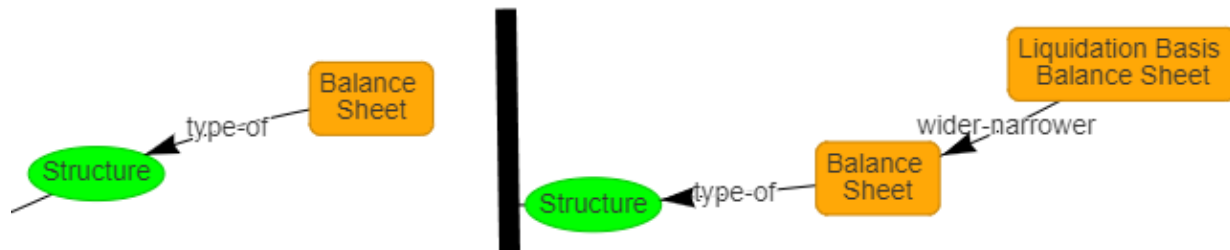
to¹⁷. When you say “Balance Sheet” you know what that means. But a machine does not know.

A base taxonomy should (a) provide all necessary structures separately, not intermingle different models in the same set of associations and (b) define what each structure must look like. Remember, computers are like babies and need to be led by the hand in order to understand the details you need them to understand.



Finally, in our case we have only one disclosure, the Balance Sheet. In our case, the Balance Sheet is always required to be reported per this logical system. As such, that rule is stated in a machine-readable reporting checklist¹⁸. Other logical systems with more disclosures will have more rules relating to when a disclosure is required to be provided in a report.

Similar to how “Payables” was added as an extension of the terms in the logical system; we can extend the structures to include a “Liquidation Basis Balance Sheet” structure which is a specialization of a Balance Sheet:



And such, an automated process will be able to understand the new structure because it is related to an existing structure. Other structures could be added in this same manner and only identified as a type of structure. But if you want to understand what that structure is, you need to associate any newly defined structure relative to some existing structure. Humans will only understand the difference by reading the documented associated with the new structure.

¹⁷ XBRL taxonomy schema used to define “Balance Sheet”, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/disclosures.xsd>

¹⁸ XBRL Definition relations used to represent a reporting checklist or disclosure rules, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/reporting-checklist-rules-def.xml>

Test: State 1 (Properly functioning logical system)

% Accounting Equation: State 1 %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state1/instance.xml",
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-
def.xml", "http://www.xbrlsite.com/2021/testing/ae/disclosure-
mechanics/dm.xsd", "http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-
def.xml", "http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],
[valueAssertionsCanDerive, cacheValidity(0)],Result).
```

Properly functioning logical system:

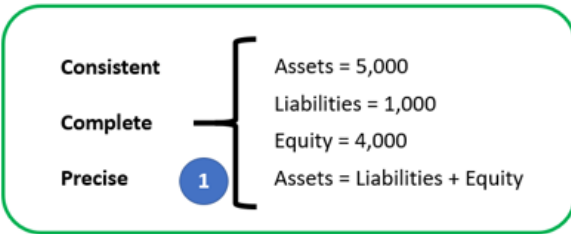


Table	
Balance Sheet [Hypercube]	Period
Concept	2020-12-31
Balance Sheet [Arithmetic]	
Assets	5,000
Liabilities	1,000
Equity	4,000

#	Type	Name	Rule Expression
1	valueAssertion	REPORT_Arithmetic_BS01 <ul style="list-style-type: none"> derived:0 ok:1 failed:0 	Assets=Liabilities+Equity 1 instance: ae:Assets[5000] = ae:Liabilities[1000] + ae:Equity[4000]

Test: State 2 (Incomplete coverage by rules)

% Accounting Equation: State 2 %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state2/instance.xml",  
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-  
def.xml","http://www.xbrlsite.com/2021/testing/ae/disclosure-  
mechanics/dm.xsd","http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-  
def.xml","http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],  
[valueAssertionsCanDerive, cacheValidity(0)],Result).
```

Incomplete (missing rule “Assets = Liabilities + Equity”)

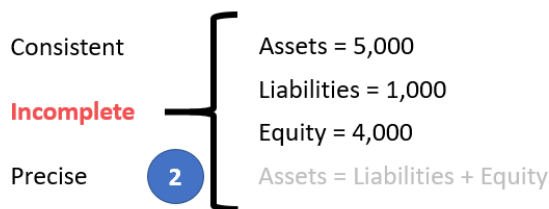


Table	
Balance Sheet [Hypercube]	Period
Concept	2020-12-31
Balance Sheet [Arithmetic]	
Assets	5,000
Liabilities	1,000
Equity	4,000

The XBRL formula for “Assets = Liabilities + Equity” does not exist so the BalanceSheet is not discovered. Also, note that the underlines shown in State1 are not shown in State2 which indicates that the XBRL formula is missing.

Test: State 3 (Inconsistent and Imprecise)

% Accounting Equation: State 3 %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state3/instance.xml",
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-
def.xml","http://www.xbrlsite.com/2021/testing/ae/disclosure-
mechanics/dm.xsd","http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-
def.xml","http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],
[valueAssertionsCanDerive, cacheValidity(0)],Result).
```

Note that the value of Assets was erroneously entered as “8,000” (should be 5,000). This error was discovered because the rule “Assets = Liabilities + Equity” is provided. Note that if the XBRL Formula was not provided (i.e. like in State2) this inconsistency would not have been discovered by the software application.

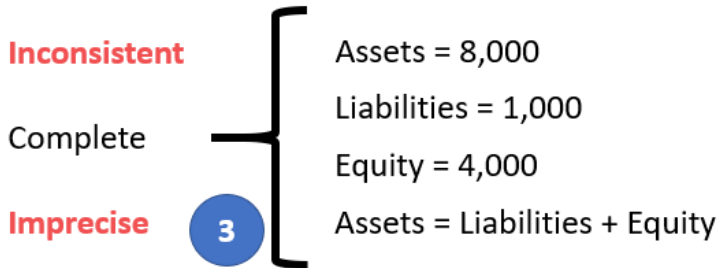


Table	
Balance Sheet [Hypercube]	Period
Concept	2020-12-31
Balance Sheet [Arithmetic]	
Assets	8,000
Liabilities	1,000
Equity	4,000

#	Type	Name	Rule Expression
1	valueAssertion	REPORT_Arithmetic_BS01 <ul style="list-style-type: none"> derived:0 ok:0 failed:1 	Assets=Liabilities+Equity 1 instance: <i>ae:Assets[8000] = ae:Liabilities[1000] + ae:Equity[4000]</i>

Test: State 4 (Unreported fact)

% Accounting Equation: State 4 %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state4/instance.xml",
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-
def.xml","http://www.xbrlsite.com/2021/testing/ae/disclosure-
mechanics/dm.xsd","http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-
def.xml","http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],
[valueAssertionsCanDerive, cacheValidity(0)],Result).
```

Note that the fact value for Liabilities was not reported. Further, it could not be derived because there was no derivation rule provided.

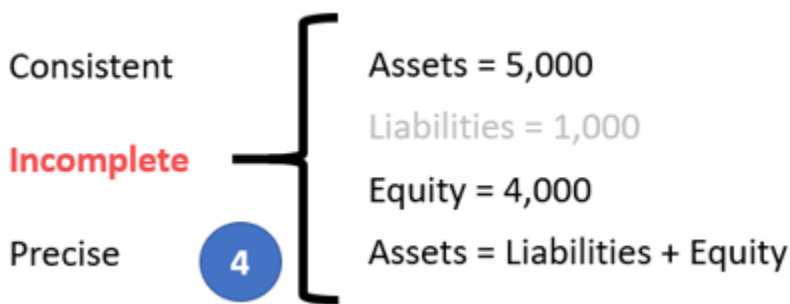


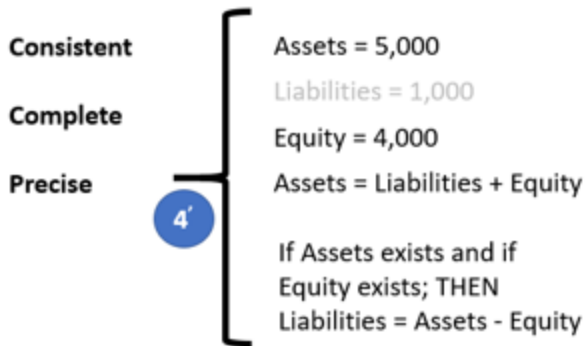
Table		Period		Unit	
Balance Sheet [Hypercube]		2020-12-31		iso4217:USD	
Concept		Unit		null	
Balance Sheet [Arithmetic]					
Assets			5,000		
Liabilities				0	
Equity			4,000		

Test: State 4a (Unreported facts, but fact is derived)

% Accounting Equation: State 4a %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state4a/instance.xml",
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-
def.xml","http://www.xbrlsite.com/2021/testing/ae/disclosure-
mechanics/dm.xsd","http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-
def.xml","http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],
[newRulesFormat, removePrecondFallbacks, removeValueAssertionFallbacks,
cacheValidity(0)],Result).
```

This report omits the fact “Liabilities”, however a derivation rule is used to impute the value of that missing fact. This is the report:



Report:

Table		Balance Sheet [Hypercube]		Period	Unit
Concept		Period	2020-12-31		
		Unit	iso4217:USD		
Balance Sheet [Arithmetic]					
	Assets		5,000		
	Equity		4,000		

This is the FAC fact values:

Fundamental accounting concept relations: (unreported fact is derived using provided rule)

Table ▾																			
Balance Sheet [Hypercube] ▾ ↕ ↔	Period ▾																		
Concept ▾	<table border="1"> <thead> <tr> <th></th> <th>Period</th> <th></th> </tr> </thead> <tbody> <tr> <td>Concept</td> <td></td> <td>2020-12-31</td> </tr> <tr> <td>Balance Sheet [Arithmetic]</td> <td></td> <td></td> </tr> <tr> <td>Assets</td> <td></td> <td>5,000</td> </tr> <tr> <td>Liabilities</td> <td></td> <td>1,000</td> </tr> <tr> <td>Equity</td> <td></td> <td>4,000</td> </tr> </tbody> </table>		Period		Concept		2020-12-31	Balance Sheet [Arithmetic]			Assets		5,000	Liabilities		1,000	Equity		4,000
	Period																		
Concept		2020-12-31																	
Balance Sheet [Arithmetic]																			
Assets		5,000																	
Liabilities		1,000																	
Equity		4,000																	

This is the derivation rule used to derive the unreported line item Liabilities so that the fact value could be used in the FAC analysis:

#	Type	Name	Rule Expression
1	formula	IMPUTE_BS_Impute_01 <ul style="list-style-type: none"> • derived:1 • ok:0 • failed:0 	if exists(Assets)and exists(Equity)and Liabilities eq null then Assets-Equity else null derives fac:Liabilities <i>1 instance:</i> <i>if exists fac:Assets[5000] and exists fac:Equity[4000] and fac:Liabilities[0] eq null then fac:Assets[5000] - fac:Equity[4000] else null</i> derived fac:Liabilities[1000]

The unreported fact is effectively derived using the provided derivation rule.

Test: State 5 (Incomplete, missing fact and derivation rule)

% Accounting Equation: State 5 %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state5/instance.xml",  
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-  
def.xml", "http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],  
[newRulesFormat, removePrecondFallbacks, removeValueAssertionFallbacks,  
cacheValidity(0)], Result).
```

This report is incomplete because the fact value for the line item “Liabilities” is not provided, the rule “Assets = Liabilities + Equity” is not provided, there is no derivation rule to compute the value for the unreported fact Liabilities.

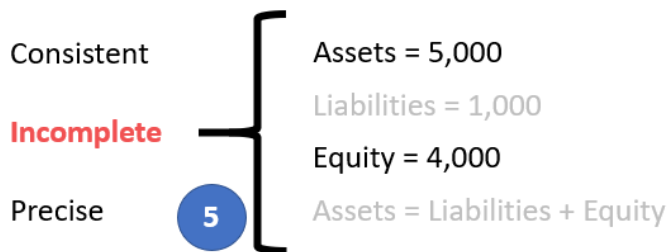


Table	
Balance Sheet [Hypercube]	Period
Concept	2020-12-31
Balance Sheet [Arithmetic]	
Assets	5,000
Equity	4,000

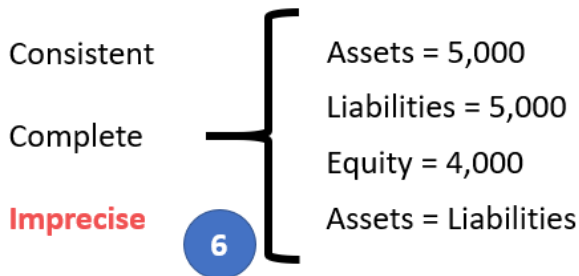
	TERMS				
	Mappings				
	Type-subtype graph				
	Type-subtype table				
	All FACTS (technical listing)				
1	01-Balance Sheet	Structures	Facts	Pivots	
	Blocks				
	Blocks Graph				
	Messages				

As there are missing fact values, rules, and derivation rules; this report can be considered incomplete.

Test: State 6 (Imprecise rule)

% Accounting Equation: State 6 %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state6/instance.xml",  
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-  
def.xml", "http://www.xbrlsite.com/2021/testing/ae/disclosure-  
mechanics/dm.xsd", "http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-  
def.xml", "http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],  
[newRulesFormat, removePrecondFallbacks, removeValueAssertionFallbacks,  
cacheValidity(0)],Result).
```



Note that the report shows no errors, but the information in the report is imprecise because the rule “Assets = Liabilities” is not consistent with reality. (i.e. the rule should be Assets = Liabilities + Equity as we have stated).

Table	
Balance Sheet [Hypercube]	Period
Concept	2020-12-31
Balance Sheet [Arithmetic]	
Assets	5,000
Liabilities	5,000
Equity	4,000

Note that the verification information gives no indication that there is an error in the report, all indicators show GREEN which means everything checks out. However, reading the rules used by the report clearly show the mistake.

	TERMS			
	Mappings			
	Type-subtype graph			
	Type-subtype table			
	All FACTS (technical listing)			
1	01-Balance Sheet	Structures	Facts	Pivots
	Blocks			
	Blocks Graph			
	Value Assertions			
All Rules	Disclosure Mechanics rules			
	Report Checklist Rules			
	Messages			

#	Type	Name	Rule Expression
1	valueAssertion	REPORT_Arithmetic_BS01 <ul style="list-style-type: none"> derived:0 ok:1 failed:0 	Assets=Liabilities <i>1 instance:</i> $ae:Assets[5000] = ae:Liabilities[5000]$

To correct this report, the rule “Assets = Liabilities” needs to be corrected to be consistent with the area of knowledge which says that “Assets = Liabilities + Equity”.

Test: State 7 (Extension concept, missing metadata)

% Accounting Equation: State 7 %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state7/instance.xml",
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-
def.xml","http://www.xbrlsite.com/2021/testing/ae/disclosure-
mechanics/dm.xsd","http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-
def.xml","http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],
[newRulesFormat, removePrecondFallbacks, removeValueAssertionFallbacks,
cacheValidity(0)],Result).
```

In this report the **extension concept** “Payables” was created for the report and the base taxonomy concept “Liabilities” was not used by the report. The rule “Assets = Liabilities + Equity” was NOT CHANGED. The rendering of the facts looks as follows:

Table		Period	
Balance Sheet [Hypercube]		2020-12-31	
Concept			
Balance Sheet [Arithmetic]			
Assets		5,000	
Payables		1,000	
Equity		4,000	

The rule that tests the value does not fire because the rule has no knowledge that “Payables” is a type of liabilities, the rule needs the term Liabilities which is not available within the report:

#	Type	Name	Rule Expression
1	valueAssertion	REPORT_Arithmetic_BS01 <ul style="list-style-type: none"> derived:0 ok:0 failed:0 	Assets=Liabilities+Equity

There is no information in the report or report model for software to use to determine that “Payables” is a type of Liabilities.

The reporting checklist did not locate the BalanceSheet disclosure

#	Type	Name	Rule Expression
1	disclosureCheck	require [disclosures:BalanceSheet] <ul style="list-style-type: none"> ok:0 failed:1 	Require disclosure: <ul style="list-style-type: none"> disclosures:BalanceSheet 1 instance: <i>Did not detect [disclosures:BalanceSheet]</i>

This is because the BalanceSheet disclosure is defined to include the concept Liabilities:

#	Type	Name	Rule Expression
1	disclosure	disclosures:BalanceSheet <ul style="list-style-type: none"> detections:0 	<i>Balance Sheet</i> <ul style="list-style-type: none"> disclosures:BalanceSheet requires: <ul style="list-style-type: none"> Hypercube ae:BalanceSheetHypercube Concept Arrangement Pattern sbrm:ArithmeticExpression <ul style="list-style-type: none"> with ae:Assets Concept ae:Liabilities Concept ae:Equity

As such, the disclosure mechanics rules do not detect the BalanceSheet disclosure and so the reporting checklist could not find the BalanceSheet disclosure therefore the reporting checklist rules do not detect the required disclosure BalanceSheet.

Basically, there is not enough information provided in the report or the report model to help software understand that Payables is a narrower concept of the Liabilities concept and is acceptable to report that fact.

Test: State 7a (Extension concept, no missing metadata)

% Accounting Equation: State 7a %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state7a/instance.xml",
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-
def.xml","dm-BalanceSheet-rules-
def.xml","http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-
def.xml","http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],
[newRulesFormat, removePrecondFallbacks, removeValueAssertionFallbacks,
cacheValidity(0)],Result).
```

In this report adjustments are made for an **extension concept** “Payables” in order to make certain that the report is properly functioning and that investors and analysts attempting to extract information from the report using machine-based processes can understand extracted information effectively.

		TERMS		
		Mappings		
		Type-subtype graph		
		Type-subtype table		
		All FACTS (technical listing)		
1	01-Balance Sheet	Structures	Facts	Pivots
2	FAC-1-Balance Sheet	Structures	Facts	Pivots
		Graph of reasoning		
		Blocks		
		Blocks Graph		
		Derivation Rules		
		Value Assertions		
	All Rules	Disclosure Mechanics rules		
		Report Checklist Rules		
		Messages		

Messages

NONE.

Everything is brought back into equilibrium.

**CC0 1.0 Universal (CC0 1.0)
Public Domain Dedication**

CC0 1.0 Universal (CC0 1.0) Public Domain Dedication <https://creativecommons.org/publicdomain/zero/1.0/>

Rule:

2	valueAssertion	<p>REPORT_Arithmetic_BS01</p> <ul style="list-style-type: none"> derived:0 ok:1 failed:0 	<p>Assets=Liabilities+Equity</p> <p><i>1 instance:</i></p> <p><i>ae:Assets[5000] = report:Payables[1000] + ae:Equity[4000]</i></p>
---	----------------	--	--

Reporting checklist:

#	Type	Name	Rule Expression
1	disclosureCheck	<p>require [disclosures:BalanceSheet]</p> <ul style="list-style-type: none"> ok:1 failed:0 	<p>Require disclosure:</p> <ul style="list-style-type: none"> disclosures:BalanceSheet <p><i>1 instance:</i></p> <p><i>See link above</i></p>

Disclosure mechanics:

#	Type	Name	Rule Expression
1	disclosure	<p>disclosures:BalanceSheet</p> <ul style="list-style-type: none"> detections:1 	<p><i>Balance Sheet</i></p> <ul style="list-style-type: none"> disclosures:BalanceSheet requires: <ul style="list-style-type: none"> Hypercube ae:BalanceSheetHypercube Concept Arrangement Pattern <i>sbrm:ArithmeticExpression</i> <ul style="list-style-type: none"> with ae:Assets Concept ae:Liabilities OR: <ul style="list-style-type: none"> report:Payables Concept ae:Equity <p><i>1 instance:</i></p> <p><i>ae:BalanceSheetHypercube is presented</i></p> <p><i>report:Payables is presented</i></p> <p><i>ae:Equity is presented</i></p> <p><i>Detected block 1.1.1.1.arithmetic with ae:Assets</i></p>

Fundamental accounting concept relations:

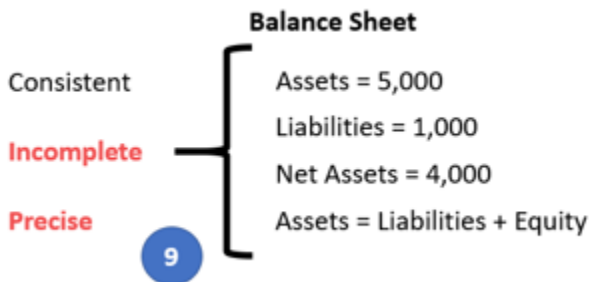
Table ▾																			
Balance Sheet [Hypercube] ▾ ↕ ↔	Period ▾																		
Concept ▾	<table border="1"> <thead> <tr> <th></th> <th>Period</th> <th>2020-12-31</th> </tr> </thead> <tbody> <tr> <td>Concept</td> <td></td> <td></td> </tr> <tr> <td>Balance Sheet [Arithmetic]</td> <td></td> <td></td> </tr> <tr> <td>Assets</td> <td></td> <td><u>5,000</u></td> </tr> <tr> <td>Liabilities</td> <td></td> <td>1,000</td> </tr> <tr> <td>Equity</td> <td></td> <td>4,000</td> </tr> </tbody> </table>		Period	2020-12-31	Concept			Balance Sheet [Arithmetic]			Assets		<u>5,000</u>	Liabilities		1,000	Equity		4,000
	Period	2020-12-31																	
Concept																			
Balance Sheet [Arithmetic]																			
Assets		<u>5,000</u>																	
Liabilities		1,000																	
Equity		4,000																	

Test: State 9 (Defining completely new structure, missing information)

% Accounting Equation: State 9 %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state9/instance.xml",  
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-  
def.xml", "http://www.xbrlsite.com/2021/testing/ae/disclosure-  
mechanics/dm.xsd", "http://www.xbrlsite.com/2021/testing/ae/reporting-checklist/dr-rules-  
def.xml", "http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-subtype-def.xml"],  
[newRulesFormat, removePrecondFallbacks, removeValueAssertionFallbacks,  
cacheValidity(0)], Result).
```

Note that effectively a new disclosure was created. The balance sheet was changed to a liquidation basis balance sheet. Further, the rule “Assets = Liabilities + Equity” was not adjusted to reflect this new structure.



The report appears to look OK, but there are a number of issues with the report that is shown in the verification of the report.

Table		Period	
Balance Sheet [Hypercube]		2020-12-31	
Concept	Period	2020-12-31	
Balance Sheet [Arithmetic]			
Assets		5,000	
Liabilities		1,000	
Net Assets		4,000	

	TERMS			
	Mappings			
	Type-subtype graph			
	Type-subtype table			
	All FACTS (technical listing)			
1	01-Balance Sheet	Structures	Facts	Pivots
2	FAC-1-Balance Sheet	Structures	Facts	Pivots
	Blocks			
	Blocks Graph			
	Derivation Rules			
	Calculations			
All Rules	Value Assertions			
	Disclosure Mechanics rules			
	Report Checklist Rules			
	Messages			

Messages

#	Type	Stage	Message
1	error	Type-subtype violation	item is not a subtype of the type of its sum in rollUp in network(http://www.xbrl.org/2003/role/link): sum report:NetAssets , parcel ae:Assets in calculation
2	error	Type-subtype violation	item is not a subtype of the type of its sum in rollUp in network(http://www.xbrl.org/2003/role/link): sum report:NetAssets , parcel ae:Liabilities in calculation
3	inconsistency	require [disclosures:BalanceSheet]	Require disclosure: <ul style="list-style-type: none"> disclosures:BalanceSheet

In addition the fundamental accounting concept relations verification is inconsistent from expectation because this disclosure is not included in the FAC verification rules.

However, State 9a corrects each of these issues and again brings the report, report model, and the rules into equilibrium.

Test: State 9a (Defining completely new structure properly)

% Accounting Equation: State 9a %

```
checkReport3("http://www.xbrlsite.com/2021/testing/ae/state9a/instance.xml",  
["http://www.xbrlsite.com/2021/testing/ae/model-structure/model-structure-strict-  
def.xml", "dm-NetAssets-rules-def.xml", "http://www.xbrlsite.com/2021/testing/ae/reporting-  
checklist/dr-rules-def.xml", "http://www.xbrlsite.com/2021/testing/ae/type-subtype/type-  
subtype-def.xml"], [newRulesFormat, removePrecondFallbacks,  
removeValueAssertionFallbacks, cacheValidity(0)], Result).
```

Note that an **entirely new structure was created**, new disclosure mechanics rules for the disclosure represented by that structure, new mathematical relations, and the entire report still checks out 100%:

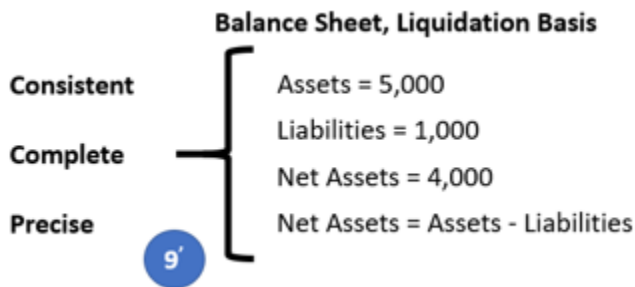


Table	
Net Assets [Hypercube]	Period
Concept	2020-12-31
Net Assets [Roll Up]	
Assets	5,000
Liabilities	1,000
Net Assets	<u>4,000</u>

		TERMS		
		Mappings		
		Type-subtype graph		
		Type-subtype table		
		All FACTS (technical listing)		
1	02-Balance Sheet, Liquidation Basis	Structures	Facts	Pivots
2	FAC-1-Balance Sheet, Liquidation basis	Structures	Facts	Pivots
		Blocks		
		Blocks Graph		
	All Rules	Calculations		
		Value Assertions		
		Disclosure Mechanics rules		
		Report Checklist Rules		
		Messages		

Messages

NONE.

In addition to the newly reported information which is not include within the base model, rules that indicate what is permissible are also created and used to verify that this entirely new section is consistent with what would be expected.

Everything is brought back into equilibrium between the report, the report model, and the rules that determine what is permissible and used to control the process. This yields verifiably correct reports and reported information.