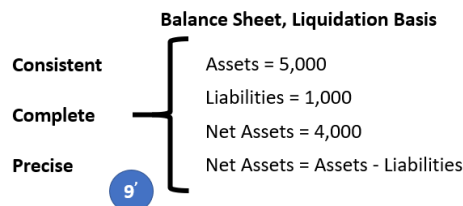
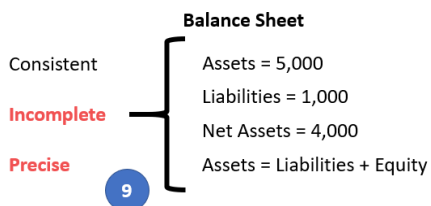
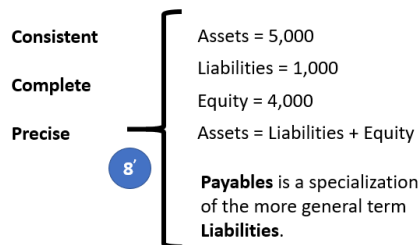
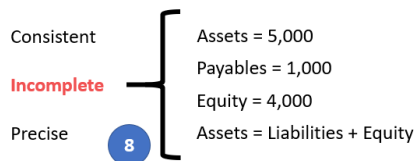
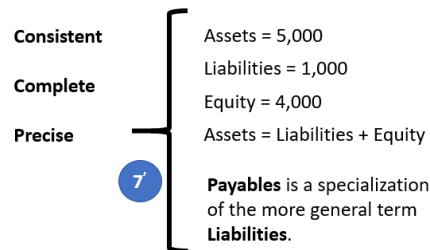
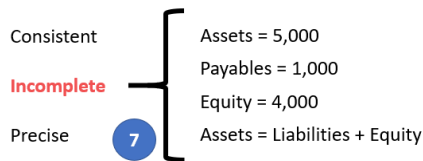
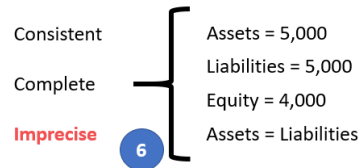
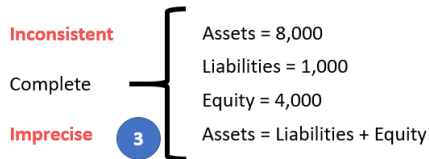
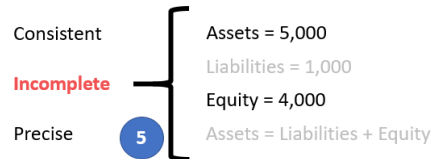
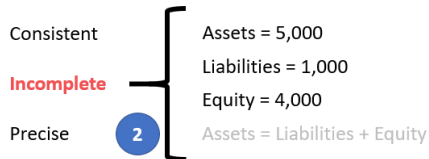
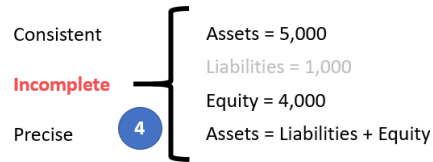
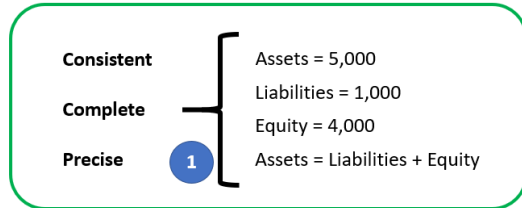


Comparison of States

The following is a comparison of 9 states of the same logical system¹:



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 System².

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 compare and contrast the properly functioning and improperly functioning logical systems.

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.

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

<p>Consistent</p> <p>Complete</p> <p>Precise</p>	<div style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; width: 20px; height: 20px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">1</div>	<p>Balance Sheet</p> <p>Assets = 5,000</p> <p>Liabilities = 1,000</p> <p>Equity = 4,000</p> <p>Assets = Liabilities + Equity</p>
---	---	---

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

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

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>

internal system understands more that the accounting equation system (i.e. you have LiabilitiesAndEquity). 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.

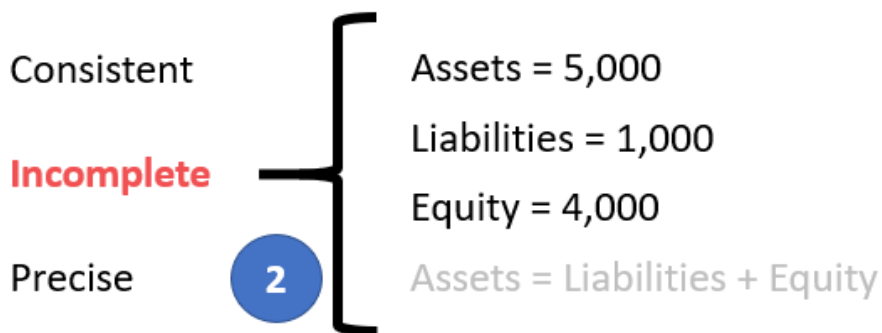
The point is that different economic entities have different models; but all models of a financial reporting scheme are reconcilable from/to one another in some manner⁷.

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 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

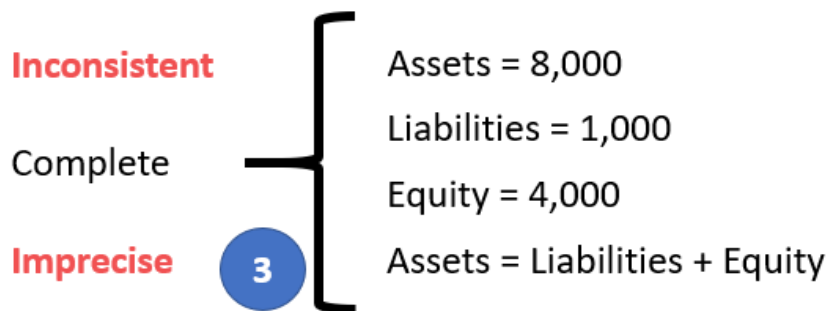
⁶ XBRL Formula to derive the value for LiabilitiesAndEquity, <http://xbrlsite.azurewebsites.net/2020/core/master-ae/fac-ImputeRule-LiabilitiesAndEquity-formula.xml>

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>

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.



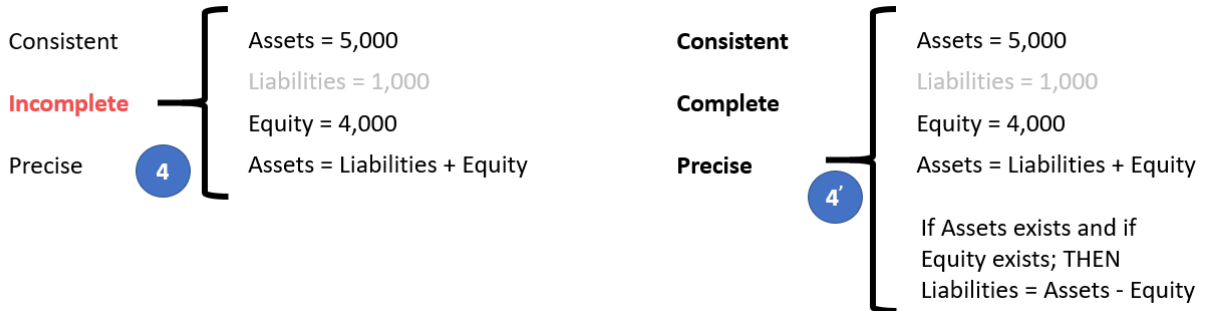
However, given that we know from state #1 that the value for Assets is 5,000 and not 8,000; the facts in this system is imprecise because the fact for Assets does not reflect reality.

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 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.

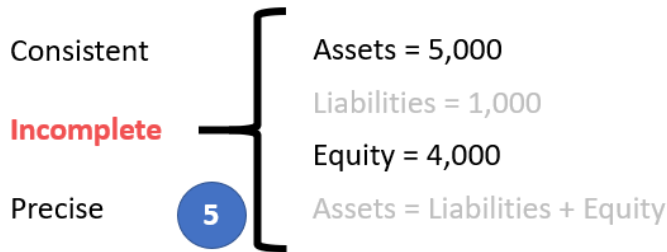
⁹ XBRL Formula rule for deriving Assets, <http://xbrl.azurewebsites.net/2020/core/master-ae/ImputeRule-Key-3-Code-BS-Impute-03-formula.xml>

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>

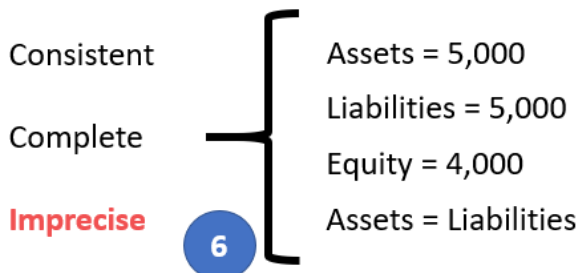


Again, consistent with state #4; Assets and Equity would require similar rules and there is no parity check of reported information.

State 6: Imprecise

A logical system is a true and fair representation of some agreed upon realism. **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.

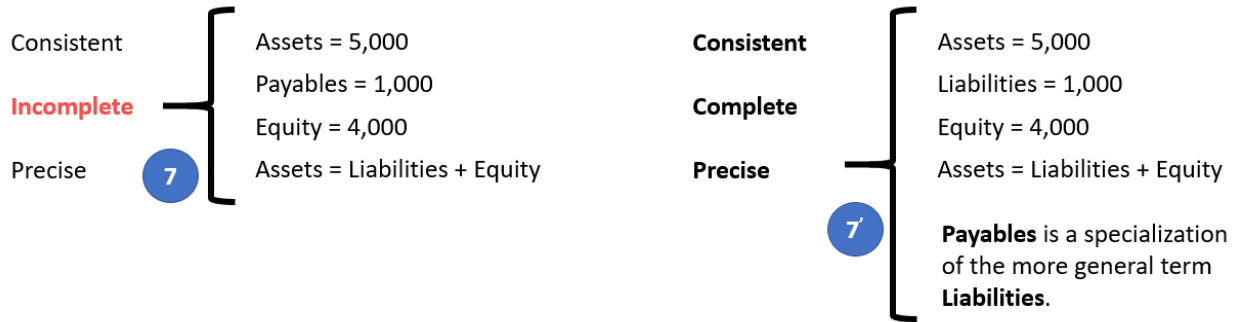
Remember that in state #1 we formalized our truth to be that “Assets = Liabilities + Equity”. As such, this logical system can be described as being imprecise. To make this logical system precise, all that needs to be done is to fix the rule.



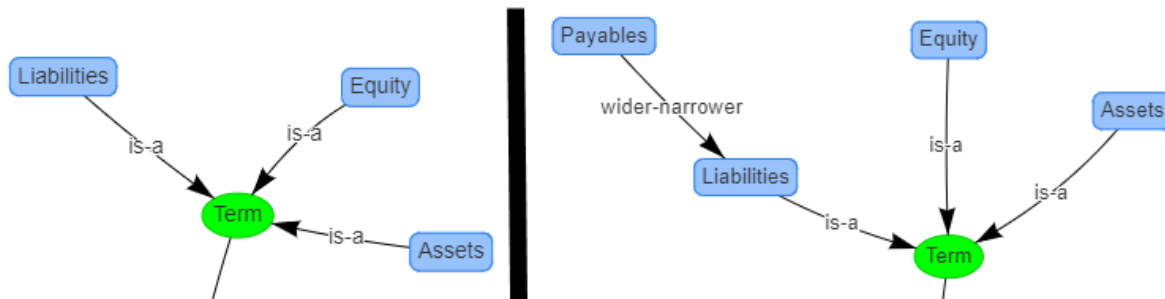
State 7: Extension Concept

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.

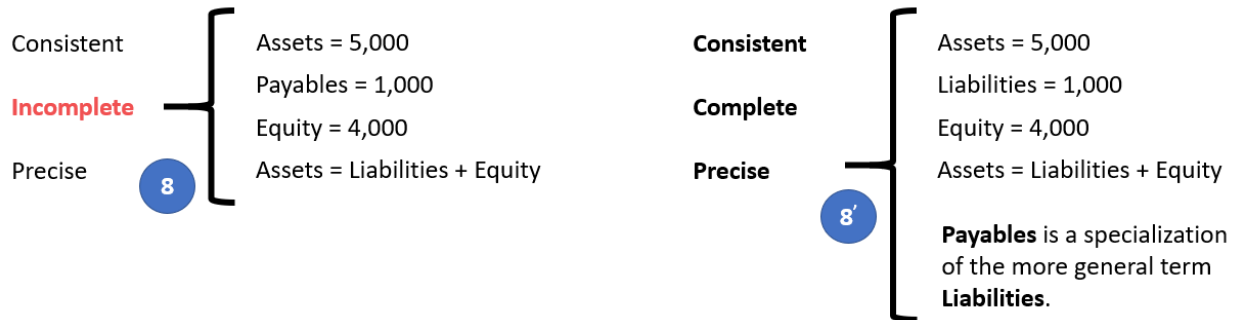


State 8: Base Taxonomy Wider/Narrower Concept Use

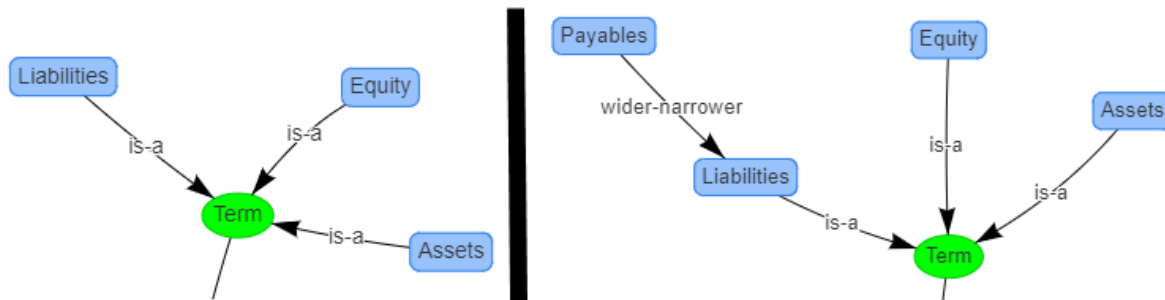
State #8 on the left below is similar to state #7 in that a different concept is used to report a fact; but while state #7 focuses on using an extension concept; state #8 points out that using a wider or narrower base taxonomy concept gives exactly the same result.

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.



State 9: Defining a Completely New Structure

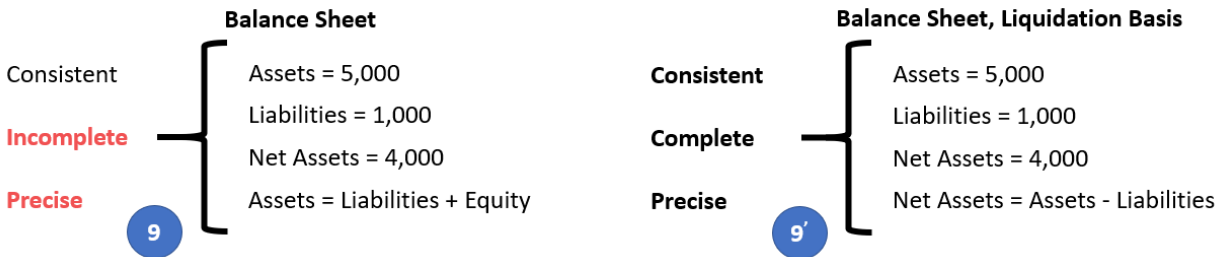
State #9 below on the left focuses on the structure as contrast all the prior examples which focused on the terms and rules. If a new structure is created, the new structure must be referenced to the base taxonomy and the new structure needs to be explained using machine-readable rules¹⁶. Even base taxonomy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

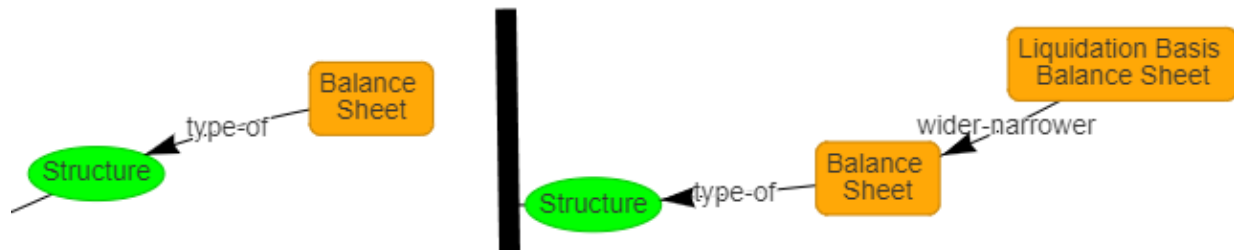
to¹⁷. When you say “Balance Sheet” you know what that means. But a machine does not know.

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 and only identified as a type of 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>