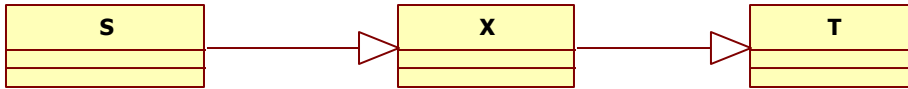


Analysis of Rules

Rule #1:

Input:



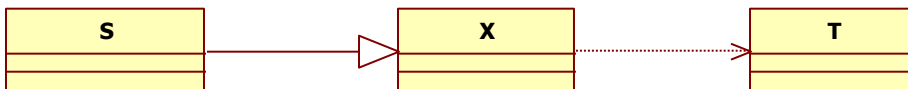
Proposed output by Egyed:



Decision:
Simply correct.

Rule #2:

Input:



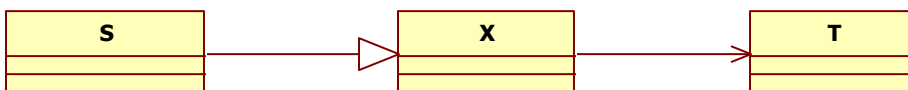
Proposed output by Egyed:



Decision:
This is correct, because the dependency between X and T where X is the client and T is supplier is also inherited by S.

Rule #3:

Input:



Output:

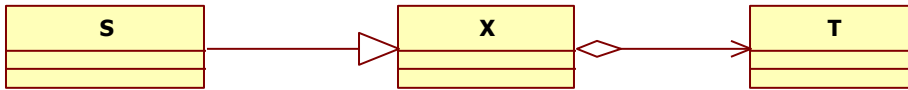


Decision:

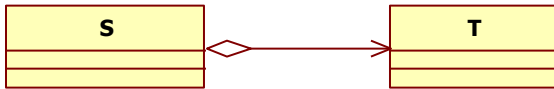
Again it is correct because of the same idea explained in Rule 2.

Rule #4:

Input:



Proposed output by Egyed:

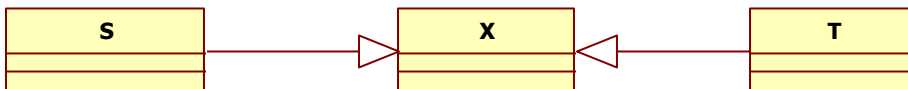


Decision:

It is correct, because the containment relationship between X and T is inherited by S.

Rule #5:

Input:



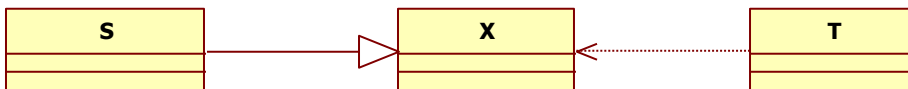
Proposed output by Egyed:
NO TRANSITIVE RELATIONSHIP

Decision:

Simply correct.

Rule #6:

Input:



Proposed output by Egyed:



Decision:

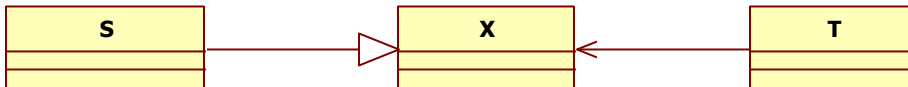
Like we discussed, the Egyed's output is not always true. If S modifies the definition in X, on which T is dependent, then we can use his reasoning.

Example: Let Polygon be an abstract class that contains a method named *calculatePerimeter*, which is left to its subclasses to implement. Let Triangle and Square two different classes that are subclasses of Polygon. Let Plotter be another class, which has a method that needs to use perimeters of different polygons by benefitting from polymorphism. In this case, Plotter depends on the definition of *calculatePerimeter* so that any change in the implementation of that function in either Triangle or Square might require a change in Plotter.

However, if S just adds a specialization, which is independent from the dependency between X and T, then we cannot claim that there also exists a dependency between S and T.

Rule #7:

Input:



Proposed output by Egyed:

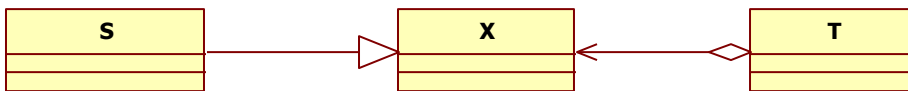


Decision:

It is correct as the association between X and T is inherited by S with the original association direction

Rule #8:

Input:



Proposed output by Egyed:

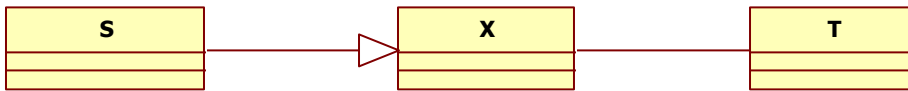


Decision:

It is correct as the containment relationship between X and T is inherited by S with the original association direction. In other words, if a set T contains another set X, then this implies that T also contains every subset of X where S is a subset of X.

Rule #9:

Input:



Proposed output by Egyed:

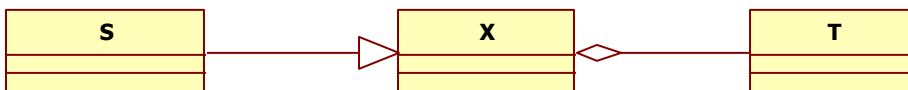


Decision:

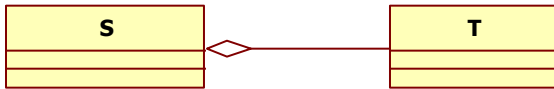
It is correct due to the similar reasons explained in Rule 1, and Rule 7.

Rule #10:

Input:



Proposed output by Egyed:

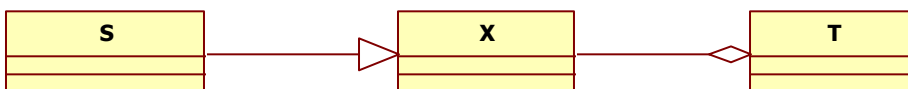


Decision:

It is correct due to the similar idea explained in Rule 4.

Rule #11:

Input:



Proposed output by Egyed:

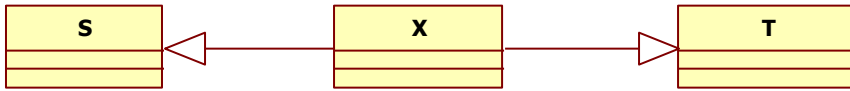


Decision:

It is correct due to the similar idea explained in Rule 8.

Rule #12:

Input:



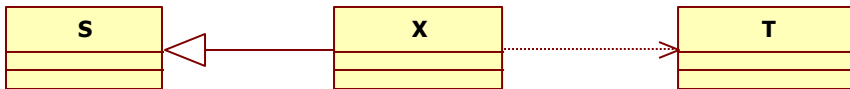
Proposed output by Egyed:
NO TRANSITIVE RULE

Decision:

Correct.

Rule #13:

Input:



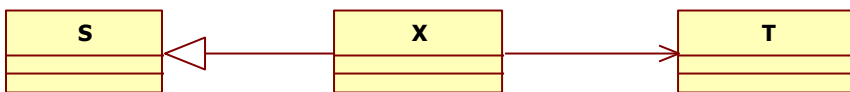
Proposed output by Egyed:
NO TRANSITIVE RULE

Decision:

Correct.

Rule #14:

Input:



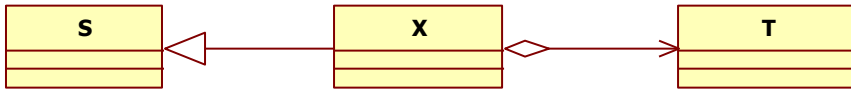
Proposed output by Egyed:
NO TRANSITIVE RULE

Decision:

Correct.

Rule #15:

Input:



Proposed output by Egyed:
NO TRANSITIVE RULE

Decision:
Correct.

Rule #16:

The following case is equivalent to Rule #1.



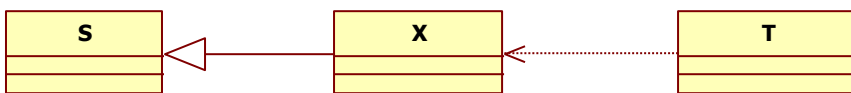
Rule #17:

The following case is equivalent to Rule #23.



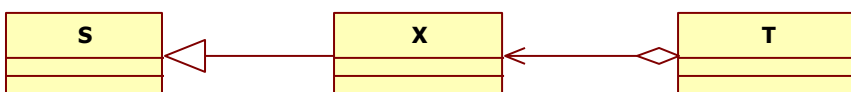
Rule #18

The following case is equivalent to Rule #45.



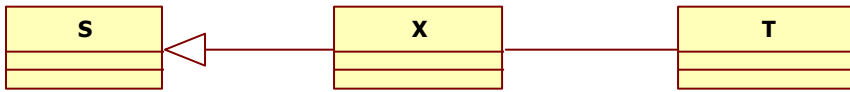
Rule #19:

The following case is equivalent to Rule #67.



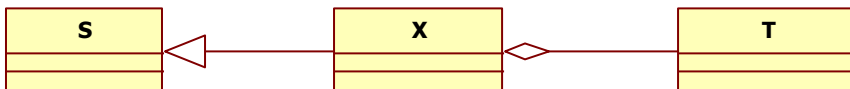
Rule #20:

The following case is equivalent to Rule #111.



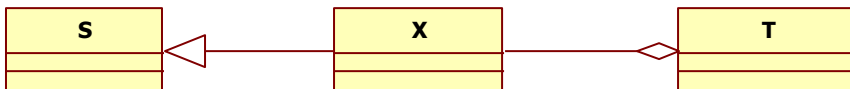
Rule #21:

The following case is equivalent to Rule #100.



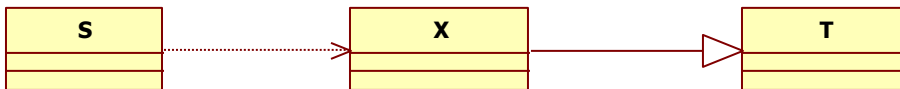
Rule #22:

The following case is equivalent to Rule #89.



Rule #23:

Input:



Proposed output by Egyed:

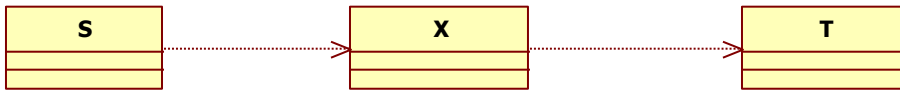


Decision:

To me, S should not dependent on T. The reason is that X is a specialization of T and S is only dependent on X in the original diagram. Therefore, we can infer that S is only dependent on some definition, which is not defined T, but defined in X. Therefore, we cannot propagate the dependency between S and X.

Rule #24:

Input:



Proposed output by Egyed:

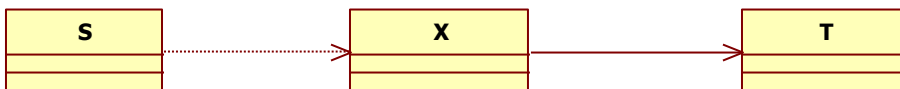


Decision:

I do not think that the output above is valid for every case. In other words, we cannot simply use the transitivity rule, because I can find a counterexample. For instance, if the dependency between S and X does not depend on the dependency between X and T, then we cannot conclude that there exists a dependency between S and T. In other words, we cannot apply the transitivity rule.

Rule #25:

Input:



Proposed output by Egyed:

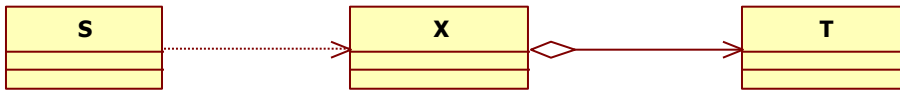


Decision:

Since we claim that every association is a dependency, then we can replace the association between X and T with a dependency. Then, we will obtain a configuration that is equivalent to the initial configuration in Rule #25. Thus, we transform Rule #25 to Rule#24. As a result, we can restate our decision in Rule #24 for Rule #25 as well.

Rule #26:

Input:



Proposed output by Egyed:

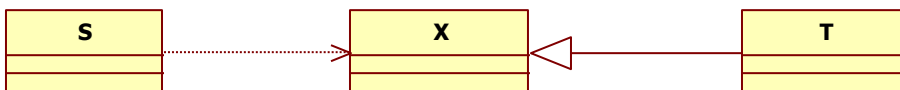


Decision:

Again, we cannot generalize, because we do not know whether the dependency between S and X is related to the aggregation between X and T.

Rule #27:

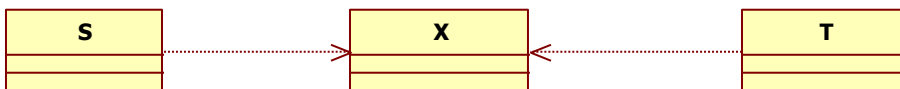
Input:



It is equivalent to Rule #6.

Rule #28:

Input:



Proposed output by Egyed:

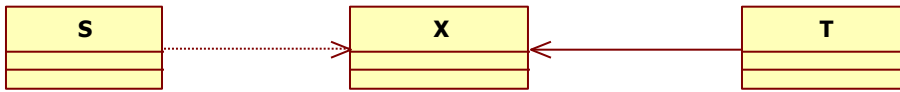
NO TRANSITIVE RELATIONSHIP

Decision:

It is simply true.

Rule #29:

Input:



Proposed output by Egyed:

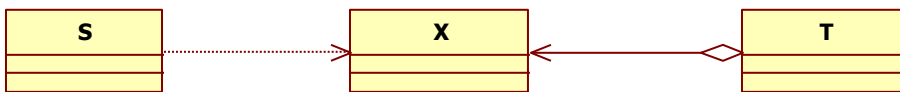
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule # 30:

Input:



Proposed output by Egyed:

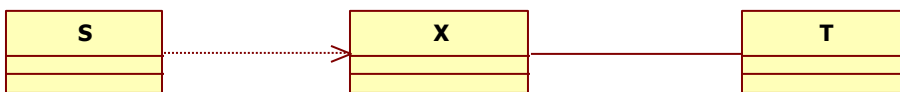
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #31:

Input:



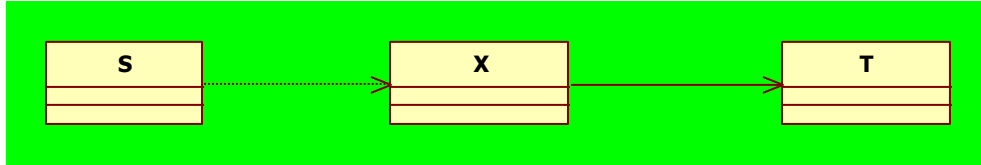
Proposed output by Egyed:



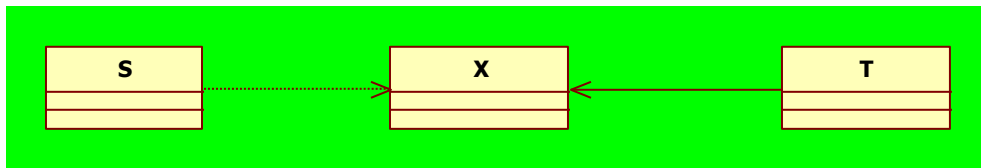
Decision:

The association between X and T is a binary association, which means that we can navigate either from X to T or from T to X. Therefore, we can claim that the original input is the union of the following two different combinations (C1 and C2):

C1:



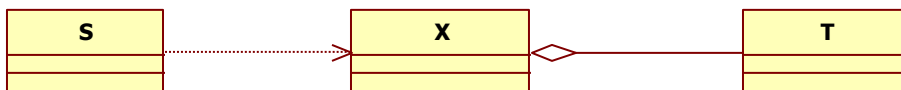
C2:



In Rule #29, we claim that the resulting relationship must be empty. Thus, C2 concludes NO TRANSITIVE RELATIONSHIP. On the other hand, in Rule #25, we claim that the transitivity exists if the dependency between S and X depends on the association between X and T. Thus, we cannot create an output for C2, that is valid for every case. As a result, we can reclaim our statement in Rule #24 for Rule #31 as well.

Rule #32:

Input:



Proposed output by Egyed:

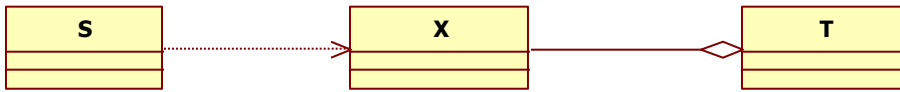


Decision:

Again, we cannot generalize as we do not know whether the dependency between S and X is related to the aggregation between X and T.

Rule #33:

Input:



Proposed output by Egyed:

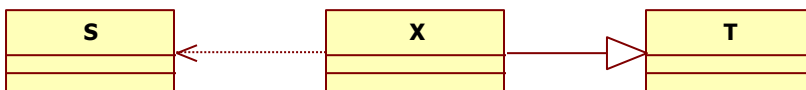


Decision:

Again, we cannot generalize as we do not know whether the dependency between S and X is related to the aggregation between X and T.

Rule #34:

Input:



Proposed output by Egyed:

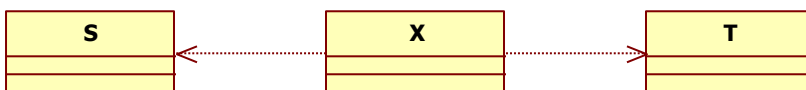
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #35:

Input:



Proposed output by Egyed:

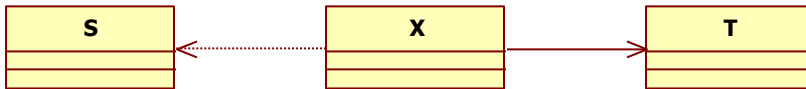
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #36:

Input:



Proposed output by Egyed:

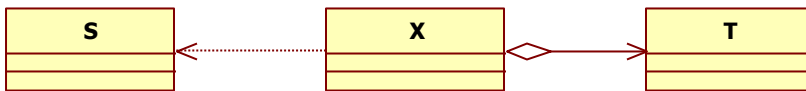
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #37:

Input:



Proposed output by Egyed:

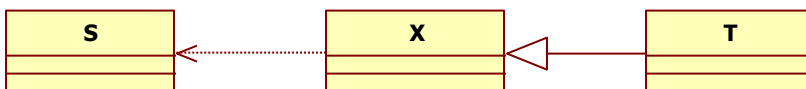
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

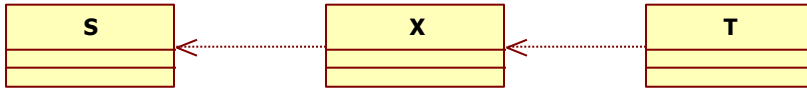
Rule #38:

The following case is equivalent to Rule #2.



Rule #39:

The following case is equivalent to Rule #24.



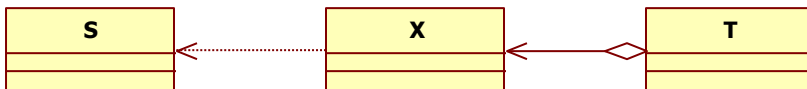
Rule #40:

The following case is equivalent to Rule #46.



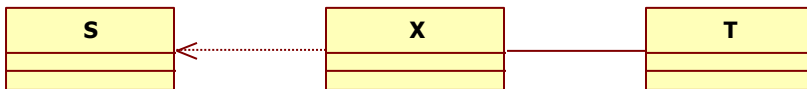
Rule #41:

The following case is equivalent to Rule #68.



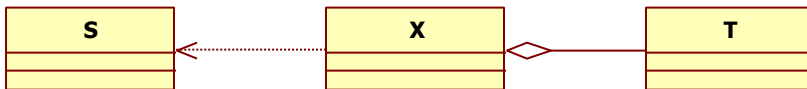
Rule #42:

The following case is equivalent to Rule #112.



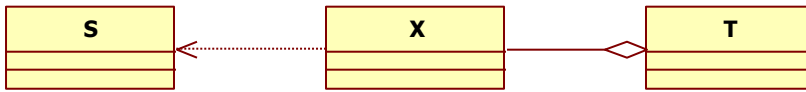
Rule #43:

The following case is equivalent to Rule #101.



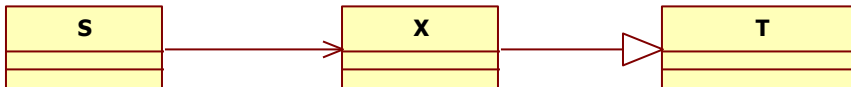
Rule #44:

The following case is equivalent to Rule #90.



Rule #45:

Input:



Proposed output by Egyed:

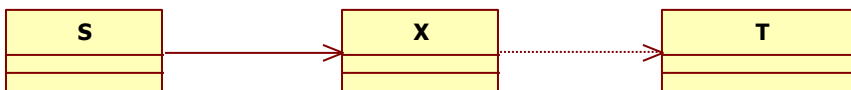


Decision:

Not a correct reasoning. Since X is a specialization of T, we cannot claim that there is also an association between S and T. Otherwise, we would have an association between S and T in the original diagram.

Rule #46:

Input:



Proposed output by Egyed:



Decision:

We cannot regard his proposed output valid for every case because of the reason explained in Rule #24. If we replace the association between S and T with a dependency, then we will have nothing but the same case indicated in Rule #24. Therefore, we can claim the same reasoning here as well.

Rule #47:

Input:



Proposed output by Egyed:



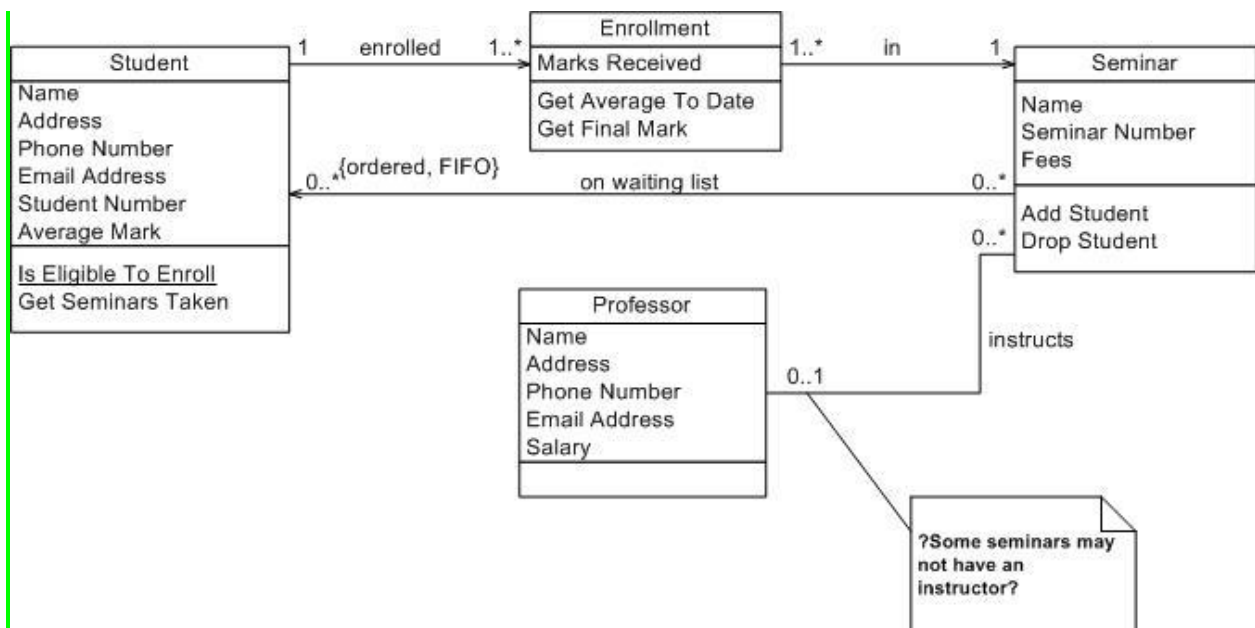
Decision:

This is again pretty interesting and we cannot generalize.

CLAIM 1:

If X is not an association class, then we cannot claim that there exists a transitive association between S and T.

Ex:



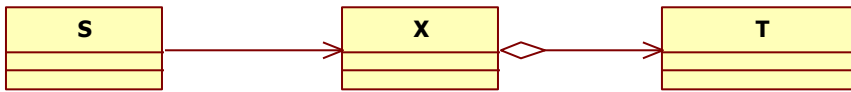
In this example, Enrollment is an association class so if we remove it, then we can create an association between Student and Seminar.

CLAIM 2:

In order to create an association between S and T, associations between S and X and X and T must be same kind. In other words, if the association between S and X is created due to the data-driven view (created to navigate from objects of S to objects of X), then the association between X and T should also be data-driven type. If the association between S and X is created because objects of S need to interact with objects of X (behavior-driven), then the association between X and T should also be created because of the same reason and there must exist a method call chain starting from S and ending in T.

Rule #48:

Input:



Proposed output by Egyed:



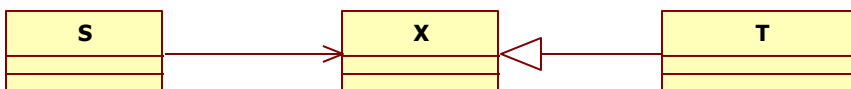
Decision:

If the association between S and X is created to navigate from objects in S to objects in X, then output is valid (Object navigation or data-driven view of associations). If the association between S and X is created because of the interaction between X and S, then we cannot always regard the output above as correct, because the association between X and T is a data-driven type.

Rule #49:

The following case is equivalent to Rule #7.

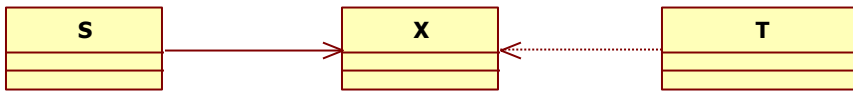
Input:



Rule #50:

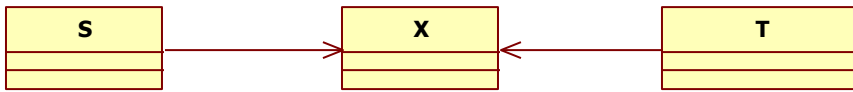
The following case is equivalent to Rule #29.

Input:



Rule #51:

Input:



Proposed output by Egyed:

NO TRANSITIVE RELATIONSHIP

Decision:

Simply true.

Rule #52:

Input:



Proposed output by Egyed:

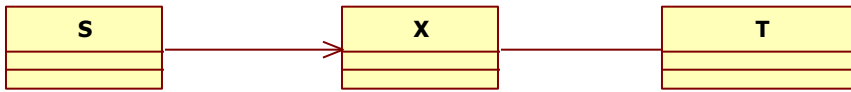
NO TRANSITIVE RELATIONSHIP

Decision:

Simply true

Rule #53:

Input:



Proposed output by Egyed:

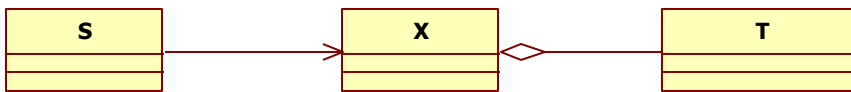


Decision:

The reasoning in Rule #47 is also valid in this rule as well.

Rule #54:

Input:



Proposed output by Egyed:

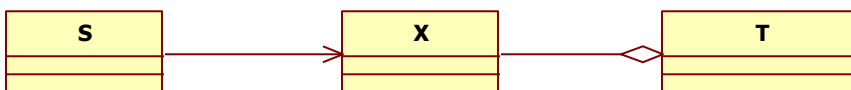


Decision:

The reasoning in Rule #48 is valid.

Rule #55:

Input:



Proposed output by Egyed:



Decision:

If the association between S and X is created to navigate from objects in X to objects in T, then output is valid (Object navigation or data-driven view of associations). If the association between S and X is because the interaction between S and X, then we cannot regard the output above as correct.

Rule #56:

Input:



Proposed output by Egyed:

NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #57:

Input:



Proposed output by Egyed:

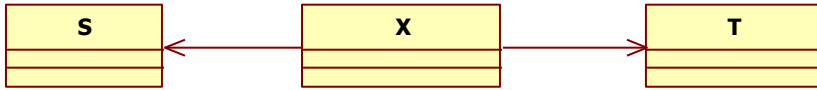
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #58:

Input:



Proposed output by Egyed:

NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #59:

Input:



Proposed output by Egyed:

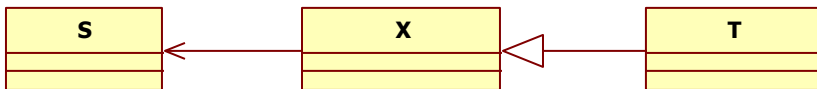
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

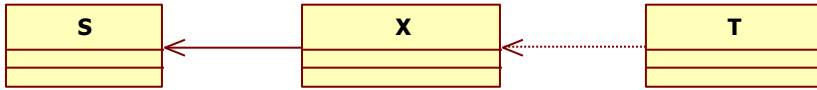
Rule #60:

The following case is equivalent to Rule #3.



Rule #61:

The following case is equivalent to Rule #25.



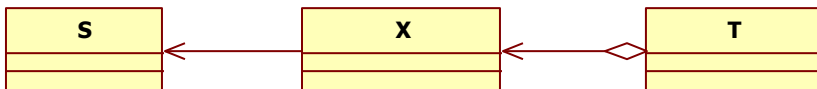
Rule #62:

The following case is equivalent to Rule #47.



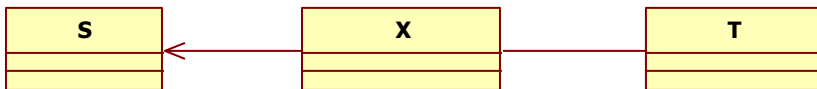
Rule #63:

The following case is equivalent to Rule #69.



Rule #64:

The following case is equivalent to Rule #113.



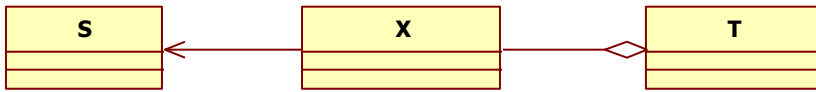
Rule #65:

The following case is equivalent to Rule #102.



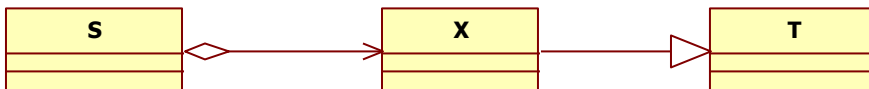
Rule #66:

The following case is equivalent to Rule #91.



Rule #67:

Input:



Proposed output by Egyed:

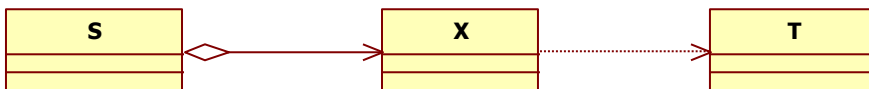


Decision:

In my opinion, it is not true, because X is a specialization of T and the aggregation only exists between S and X. If we propagate the aggregation, we obtain an invalid semantic. In other words, if we claim that S also contains T, then it implies that it is a container of every instance in T that include every instances in subclasses of T as well.

Rule #68:

Input:



Proposed output by Egyed:

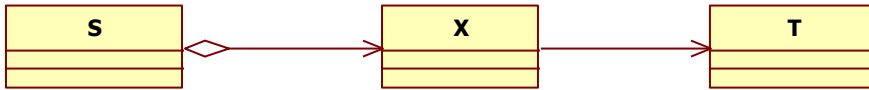


Decision:

If the dependency between X and T affects the aggregation between S and X, then we can claim that there exists a dependency between S and T. Otherwise, we cannot claim the output above.

Rule #69:

Input:



Proposed output by Egyed:

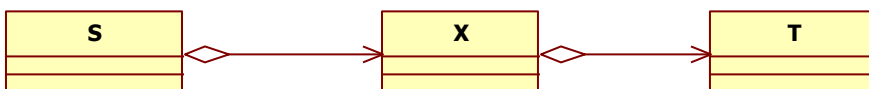


Decision:

If the association between X and T is created to navigate from objects in X to objects in T, then output is valid (Object navigation or data-driven view of associations). If the association between X and T is created because of the interaction between X and T, then we cannot always regard the output above as correct.

Rule #70:

Input:



Proposed output by Egyed:

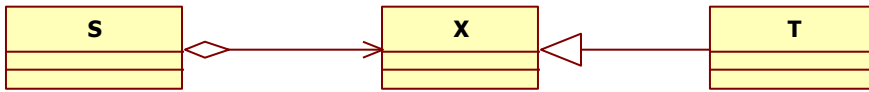


Decision:

Correct.

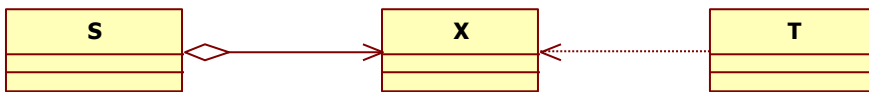
Rule #71:

The following rule is equivalent to Rule # 8.



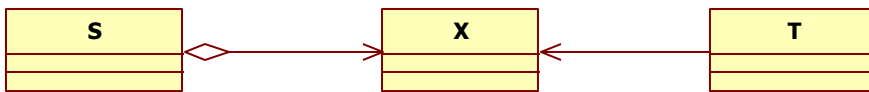
Rule #72:

The following rule is equivalent to Rule # 30.



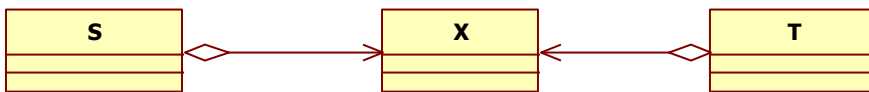
Rule #73:

The following rule is equivalent to Rule # 52.



Rule #74:

Input:



Proposed output by Egyed:

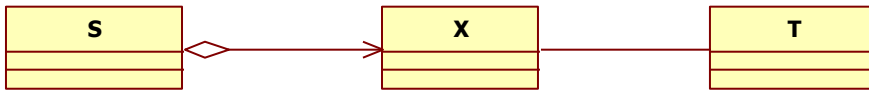
NO TRANSITIVE OUTPUT

Decision:

Correct

Rule #75:

Input:



Proposed output by Egyed:

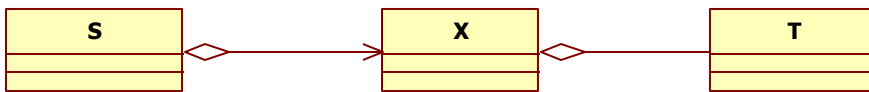


Decision:

Same reasoning made in Rule #69 is valid for this case as well.

Rule #76:

Input:



Proposed output by Egyed:

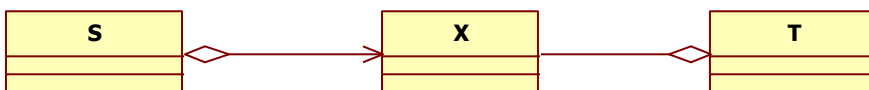


Decision:

Correct.

Rule #77:

Input:



Proposed output by Egyed:

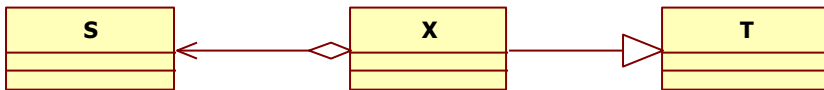


Decision:

Correct, because the association between S and X and the association between X and T are data-driven type.

Rule #78:

Input:



Proposed output by Eged:

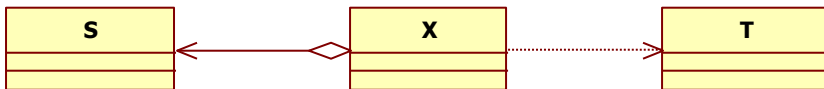
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #79:

Input:



Proposed output by Eged:

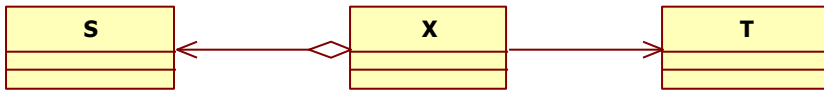
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #80:

Input:



Proposed output by Egyed:

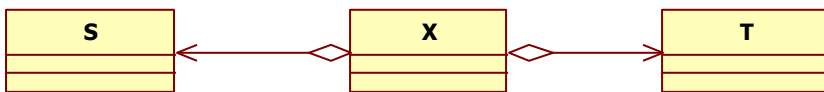
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

Rule #81:

Input:



Proposed output by Egyed:

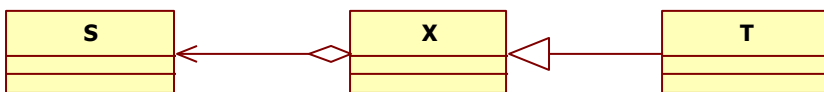
NO TRANSITIVE RELATIONSHIP

Decision:

Correct.

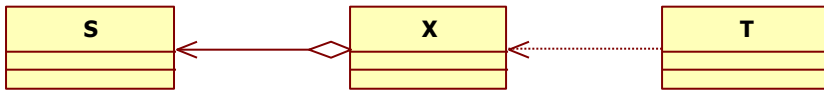
Rule #82:

The following case is equivalent to Rule #4.



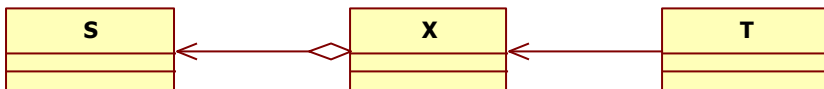
Rule #83:

The following case is equivalent to Rule #26.



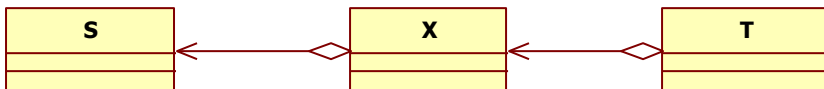
Rule #84:

The following case is equivalent to Rule #48.



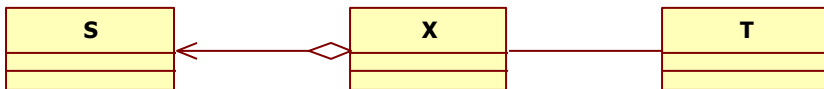
Rule #85:

The following case is equivalent to Rule #70.



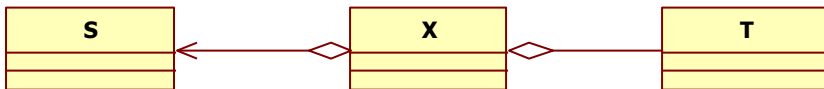
Rule #86:

The following case is equivalent to Rule #114.



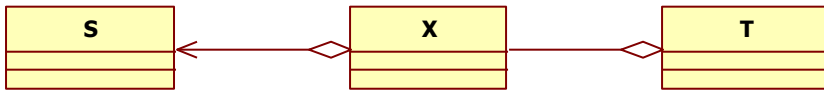
Rule #87:

The following case is equivalent to Rule #103.



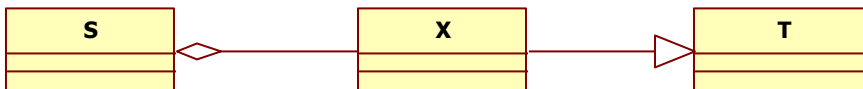
Rule #88:

The following case is equivalent to Rule #92.



Rule #89:

Input:



Proposed output by Egyed:

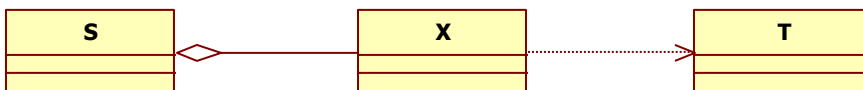


Decision:

Not correct due to the same reason we discussed before. X is a specialization of T and the aggregation between S and X is a specialized and cannot be propagated.

Rule #90:

Input:



Proposed output by Egyed:

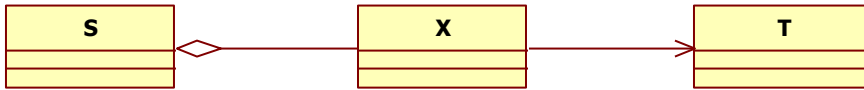


Decision:

If the dependency between X and T affects the aggregation between S and X, then we can claim that there exists a dependency between S and T. Otherwise, we cannot claim the output above.

Rule #91:

Input:



Proposed output by Egyed:

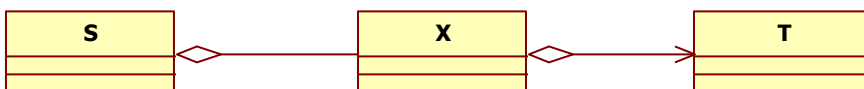


Decision:

If the association between X and T is created to navigate from objects in X to objects in T, then output is valid (Object navigation or data-driven view of associations). If the association between X and T is created because the interaction between X and T, then we cannot always regard the output above as correct, because the association between S and X is data-driven type.

Rule #92:

Input:



Proposed output by Egyed:

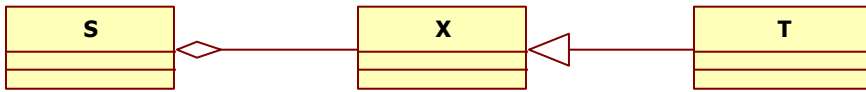


Decision:

Correct.

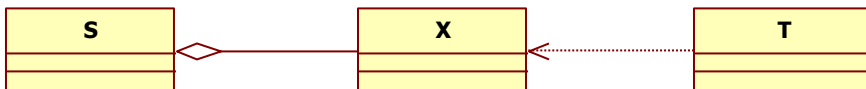
Rule #93:

The following case is equivalent to Rule #11.



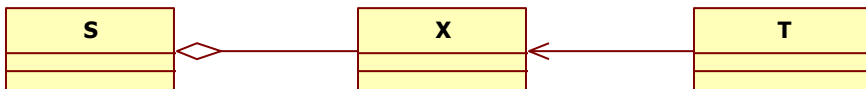
Rule #94:

The following case is equivalent to Rule #33.



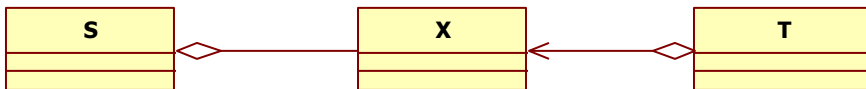
Rule #95:

The following case is equivalent to Rule #55.



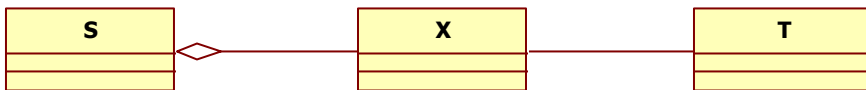
Rule #96:

The following case is equivalent to Rule #77.



Rule #97:

Input:



Proposed output by Egyed:

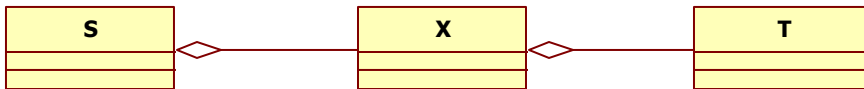


Decision:

If the association between X and T is created to navigate from objects in X to objects in T, then output is valid (Object navigation or data-driven view of associations). If the association between X and T is created because of the interaction between X and T, then we cannot always regard the output above as correct, because the association between S and X is a data-driven type.

Rule #98:

Input:



Proposed output by Egyed:

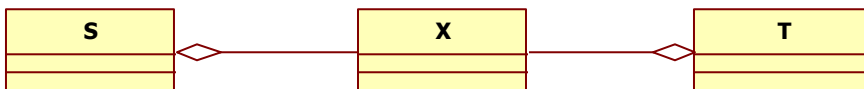


Decision:

Correct.

Rule #99:

Input:



Proposed output by Egyed:

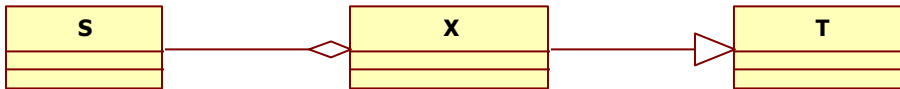


Decision:

Correct, because both associations between S and X and X and T are data-driven type.

Rule #100:

Input:



Proposed output by Egyed:

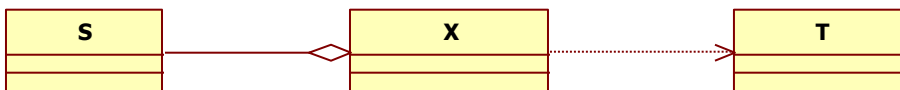


Decision:

Not correct, because X is a specialization of T.

Rule #101:

Input:



Proposed output by Egyed:

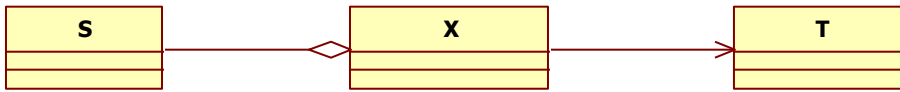


Decision:

If the aggregation between X and S is independent from the dependency between X and T, then the output above is not correct.

Rule #102:

Input:



Proposed output by Egyed:

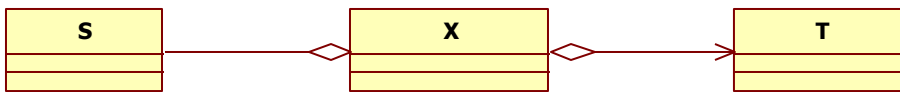


Decision:

If the association between X and T is due to data-driven, then the output above is correct.

Rule #103:

Input:



Proposed output by Egyed:

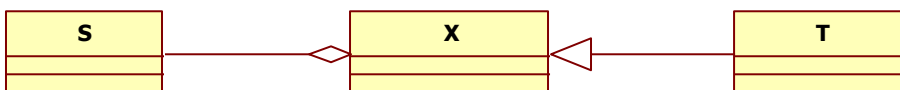


Decision:

Correct and the resulting association is in data-driven type.

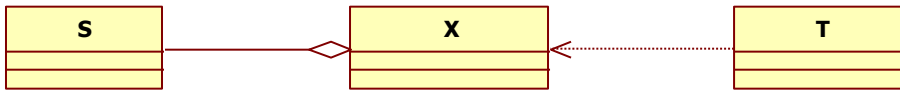
Rule #104:

The following case is equivalent to Rule #10.



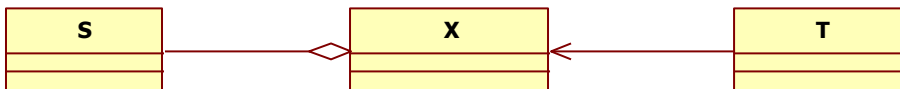
Rule #105:

The following case is equivalent to Rule #32.



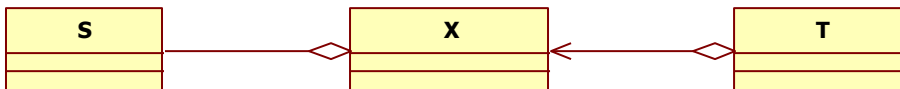
Rule #106:

The following case is equivalent to Rule #54.



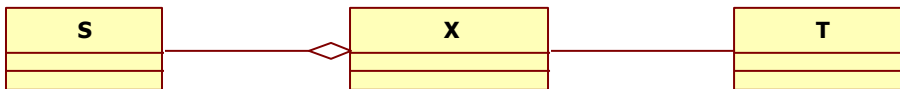
Rule #107:

The following case is equivalent to Rule #76.



Rule #108:

Input:



Proposed output by Egyed:

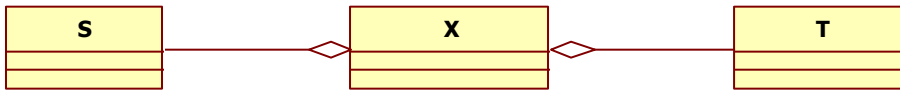


Decision:

If the association between X and T is data-driven, then the output is true. Otherwise, it is not.

Rule #109:

Input:



Proposed output by Egyed:

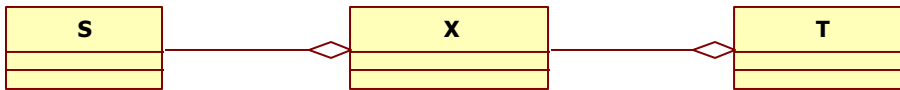


Decision:

Correct.

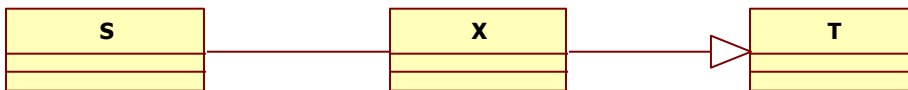
Rule #110:

The following case is equivalent to Rule #98.



Rule #111:

Input:



Proposed output by Egyed:

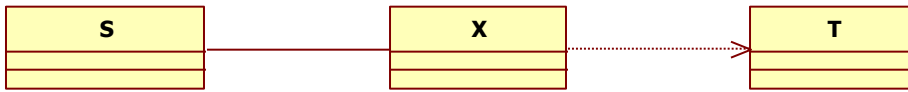


Decision:

Not correct because of the reason I explained for several times in previous rules.

Rule #112:

Input:



Proposed output by Egyed:

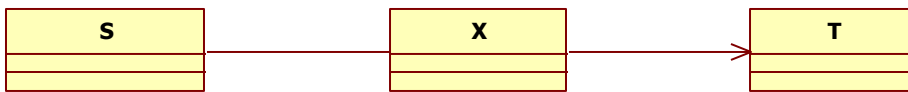


Decision:

Not correct if the association between S and X is independent from the dependency between X and T.

Rule #113:

Input:



Proposed output by Egyed:

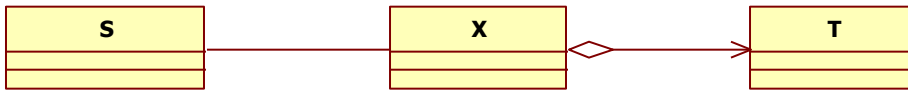


Decision:

If the association between S and X, and the association between X and T are in the same type (i.e. data-driven or behavior driven, and if it is behavior driven, then there must also exist a call chain), then the output above is correct.

Rule #114:

Input:



Proposed output by Egyed:

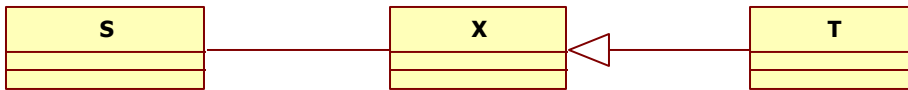


Decision:

If the association between S and X is not data-driven, then the output above is not correct.

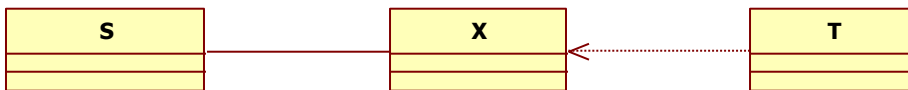
Rule #115:

The following case is equivalent to Rule #9.



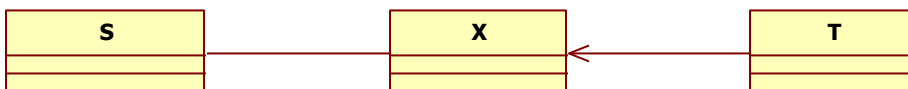
Rule #116:

The following case is equivalent to Rule #31.



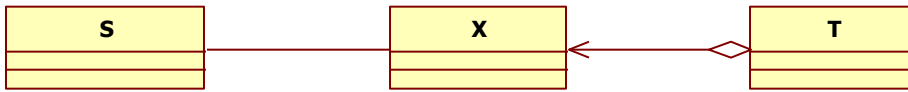
Rule #117:

The following case is equivalent to Rule #53.



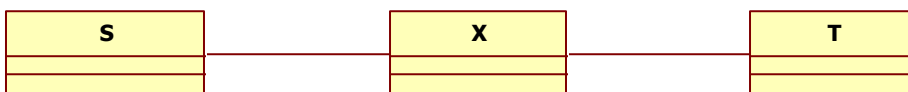
Rule #118:

The following case is equivalent to Rule #75.



Rule #119:

Input:



Proposed output by Egyed:

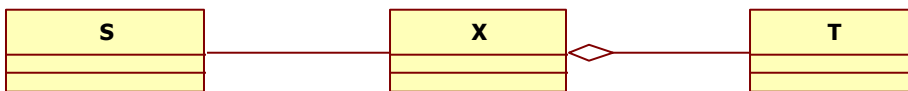


Decision:

If the association between S and X, and the association between X and T are in the same type (i.e. data-driven or behavior driven), then the output above is correct.

Rule #120:

The following case is equivalent to Rule #108.



Rule #121:

The following case is equivalent to Rule #97.

