

Task 2. Feature models, valid products, and their analysis

These exercises let you read a feature model, write it as logic, and check it for modelling mistakes. Draw any diagram on paper or with a tool that you use (*e.g.*, FeatureIDE).

Exercise 2.1. Read a feature model and its products

Figure 1 shows a small feature model for an online store (*E-Shop*). Every product of the store is one valid selection of features.

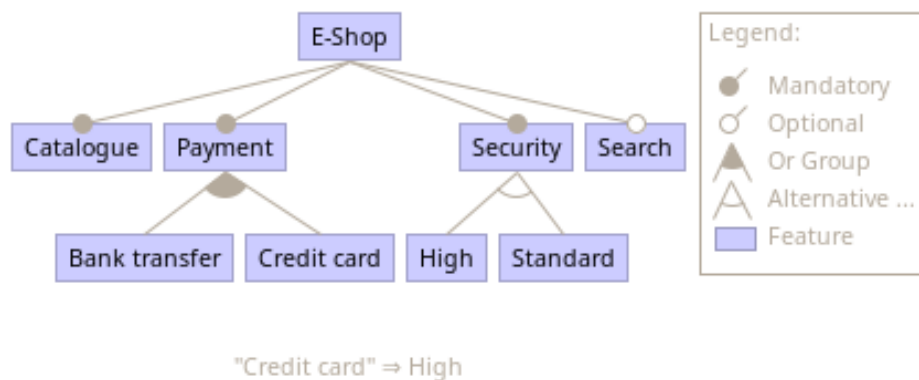


Figure 1: The E-Shop feature model.

1. For each feature, state its relationship to its parent: mandatory, optional, part of an or-group, or part of an alternative group.
2. Write the propositional formula of the model. Use the feature names as Boolean variables and include one part for each relationship (mandatory, optional, the or-group, the alternative group) and one for the cross-tree constraint.
3. How many valid products does the model allow? List them, or show how you counted.
4. Give one **valid** product and one **invalid** product (a selection that breaks a rule). For the invalid one, name the rule it breaks.

Hand in your feature classification, the propositional formula, the number of valid products with your counting, and one valid and one invalid product with the broken rule named.

Exercise 2.2. Find and fix modelling mistakes

A model can be syntactically fine yet still contain mistakes. Starting from the E-Shop model in Figure 1, the team makes two edits:

- they add an optional feature *GiftWrap* under *E-Shop*, with two cross-tree constraints: *GiftWrap* requires *High*, and *GiftWrap* requires *Standard*,
- they keep *Search* optional but add a cross-tree constraint: *Catalogue* requires *Search*.

The two edits are shown in Figure 2.

1. Show that *GiftWrap* can never appear in any valid product. Such a feature is called a **dead feature**. Explain in one or two lines why it is dead.
2. Show that *Search* is now present in every valid product, even though it is drawn as optional (this is sometimes called a *false optional* feature). Explain why.

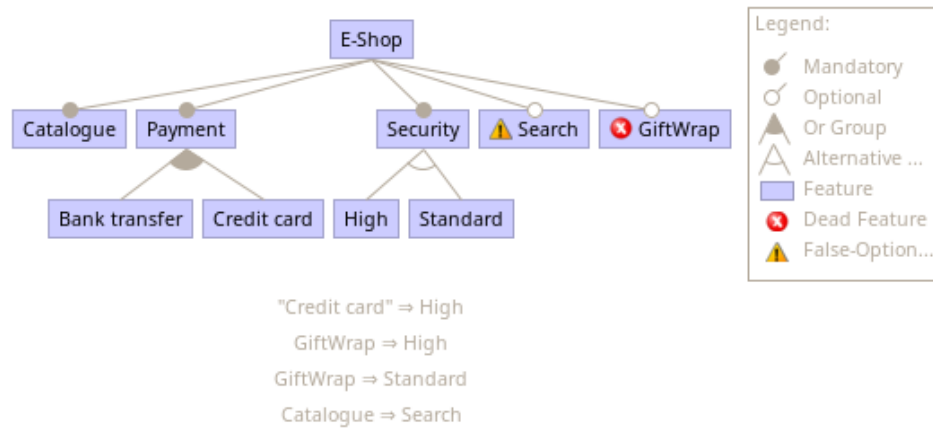


Figure 2: The E-Shop model with the optional *GiftWrap* feature added.

- Propose the smallest change that fixes each mistake so the model matches the intended behaviour: *GiftWrap* should be selectable (say, only with the higher security level) and *Search* should be genuinely optional. After your fixes, how many valid products does the model have?

Hand in your argument that *GiftWrap* is dead, your argument that *Search* is always present, your minimal fix for each mistake, and the number of valid products after the fixes.

Checklist

Before handing in your work, check that you have included the following.

- For Exercise 2.1, the feature classification, the propositional formula, the count of valid products, and one valid and one invalid product.
- For Exercise 2.2, why *GiftWrap* is dead, why *Search* is always present, a minimal fix for each, and the new count of valid products.