1. Concepts

Definition 5 (*Activable* and *inactivable transformation rule*). Given a feature model $\boldsymbol{FM}$, a TDL^c program $\boldsymbol{P}$ about $\boldsymbol{FM}$, and a transformation rule $\boldsymbol{r}$ in $\boldsymbol{P}$, $\boldsymbol{r}$ is called an *activable* transformation rule, *iff* there is a valid configuration of $\boldsymbol{FM}$, for which, the guard of $\boldsymbol{r}$ is evaluated to be *true*; otherwise, $\boldsymbol{r}$ is called an *inactivable* transformation rule.

Definition 1 (*Max-Normal Guard*). Given a feature model $\boldsymbol{FM}$, a TDL^c program $\boldsymbol{P}$ about $\boldsymbol{FM}$, a transformation rule $\boldsymbol{r}$ in $\boldsymbol{P}$, and the guard $\boldsymbol{g_r}$ of $\boldsymbol{r}$, suppose $\boldsymbol{r}$ is activable, the *max-normal guard* of $\boldsymbol{g_r}$, denoted as $\boldsymbol{max}(\boldsymbol{g_r})$, is a conjunction of *literals* (features or negations of features in $\boldsymbol{FM}$) that satisfies all of the following two properties:

- $\forall f \in FM \cdot (Cst \wedge g_r \rightarrow f) \leftrightarrow (f \in L(\max(g_r)))$, and
- $\forall f \in FM \cdot (Cst \wedge g_r \rightarrow \neg f) \leftrightarrow (\neg f \in L(\max(g_r)))$.

Here and also in the following, Cst denotes the conjunction of all the constraints in FM; given a guard g, $L(g)$ denotes the *literal set* of g (that is, the set that exactly includes all the literals in g), and $F(g)$ denotes the *feature set* of g (that is, the set that exactly includes all the features appearing in g). For example, given a guard $g = f_1 \wedge f_2 \wedge \neg f_3$, $L(g)$ denotes the set $\{f_1, f_2, \neg f_3\}$ and $F(g)$ denotes the set $\{f_1, f_2, f_3\}$.

2. Criteria

Criterion 4 (*Identifier uniqueness*). Any identifier appearing in a TDL^c program should appear in exactly one *create* operation in the TDL^c program.

This criterion means that any identifier appearing in a TDL^c program should be unique. Otherwise unacceptable confusion will occur.

Criterion 5 (*Transformation rule activability*). Every transformation rule in a TDL^c program should be *activable*.

If a transformation rule is *inactivable*, it will never be applied for any valid configuration. That is to say, this rule is unnecessary, or there must be semantic errors in the guard of this rule (supposing the feature model is correct in semantic) that cause it to be unnecessary.

Criterion 6 (*Transformation rule completeness*). Given a feature model FM, and a TDL^c program P about FM, every feature in FM should appear as a positive literal in at least one max-normal guard of a transformation rule in P.

This criterion can be understood from two kinds of context. In the context that a feature model is created only to manage commonality and variability of a class diagram, it is a hard criterion: if a feature violates this criterion, then the set of transformation rules must be incomplete, since any feature in such a feature model should be transformed to certain elements in class diagrams. In the context that a feature model is created to manage commonality and variability of all kinds of reusable assets in SPLE, it is a soft criterion: if a feature violate the criterion, it should be ensured that the feature is really irrelevant to the class diagram.