Refactoring Laws for Colorful Alloy

The refactored laws of our colorful Alloy is present in the form of an equation of two Alloy template. we can apply a refactoring whenever either side template is matched by a piece of Alloy code and when it satisfies the preconditions. A matching is an assignment of all the variables in the left/right hand side templates to concrete values from the source annotated Alloy code.

Restriction of Colorful Alloy:

ⓐ ∩ ⓑ = ⊥ , which means there's no common product in ⓐand ⓑ.

The meaning of Symbols:

p denotes a set of signatures, predicates, or facts,p may contain feature annotations;
ds denotes a set of Field declarations, ds may contain features;
ⓐ, ⓑ, ⓒ denote a set of features combination that can be marked on the Alloy elements;
ⓚ represent a concrete feature k is selected, while ⓚ means the de-selection of k.
r represent a Field relation.
Law 1,2,3,**??** are rules for signatures.

Law 1 (Remove abstract Attribute) *[1]*

ⓐ **abstract sig** A { ds} ⓐ ⓐ **sig** A_1 **extends** A { ds_1 } ⓐ ... ⓐ **sig** A_n **extends** A {ds_n } ⓐ	==	ⓐ **sig** A { ds} ⓐ ⓐ **sig** A_1 **extends** A { ds_1 }ⓐ ... ⓐ **sig** A_n **extends** A {ds_n }ⓐ ⓐ **fact** { A= A_1+ ...+ A_n } ⓐ

Law 2 (Remove the Multiplicity of Sig) *[1]*

ⓐ m **sig** A { ds } ⓐ	==	ⓐ **sig** A { ds } ⓐ ⓐ **fact** { m A }ⓐ

where $m \in \{lone, one, some\}$;

If we have signatures with abstract and multiplicity, we can always remove them with law 1 and law 2 before the refactoring .

Law 3 (Introduce generalization) *[1]*

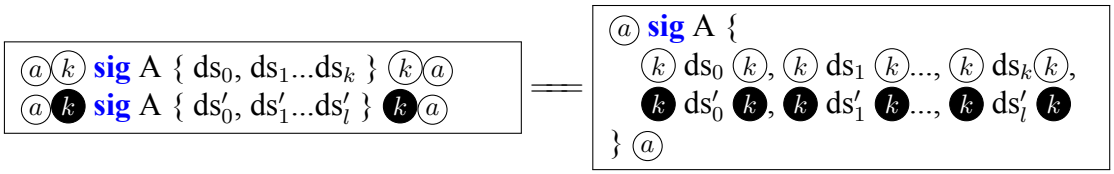
ⓐ **sig** A { ds} ⓐ	==	ⓐ **abstract sig** U { } ⓐ ⓐ **sig** A **extends** U { ds } ⓐ

→ no paragraphs and field declarations named U ;
← U does not appear in paragraphs and Field declarations.

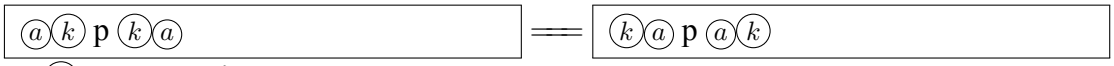
Law 4 (Separate feature annotations)

ⓐ p ⓐ	==	ⓐⓚ p ⓚⓐ ⓐⓚp ⓚⓐ

Law 5 (Separate feature annotations For Field)



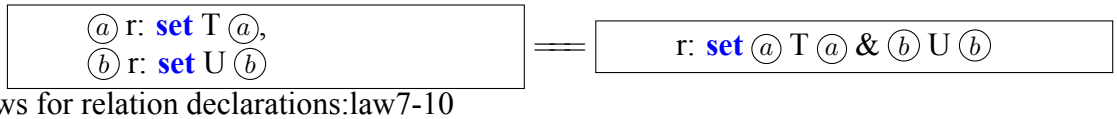
Law 6 (Reorder) *put features in the right order to merge*



$\leftrightarrow$ ⓚnot appear in ⓐ

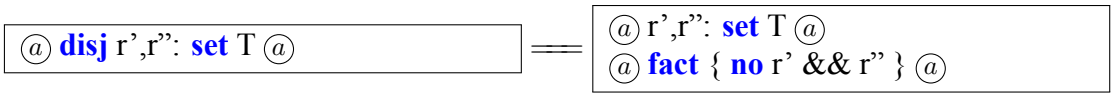
our implementation does not care about the order of the feature

Law 7 (Combine Building Expressions)



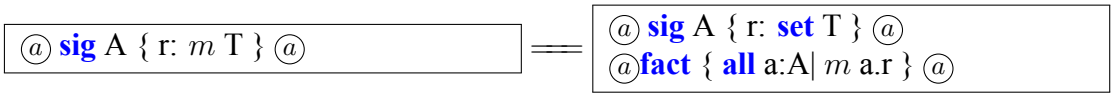
Laws for relation declarations:law7-10

Law 8 (Remove disj qualifier)



Law 9 (Separate Relation Declarations) *[1]*

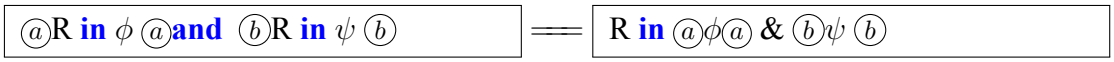
Law 10 (Remove Relation Qualifier) *[1]*



also apply to binary expressions such as (some-> one)

For expressions:

Law 11 ("and" expression)



References

[1] R. Gheyi. A refinement theory for alloy. 2007.