Summary

- Sigs such as ❶❷ one sig Ready, Selected extends State {}❷❶ need to be separate.

  But have difficulty in implementation. The position of the two sig are the same. How to do the Implementation ? calculate and compare all the positions before merge?

- Models without Feature Model get the original easily. The orders of the model is not matters when do the composition merge.
- Models with Feature Model are much more complex to merge. In our models, we can get the original model in the end.
  For example. Original model: ①category : some Category①,

  After merge the sigs and Fields, we get
  ①③❺ category : some Category…
  ①❸④ category : some Category…
  ①④⑤ category : some Category..
  ((⑤ implies ①) and (⑤ implies ④)) -→ ①

  Remove features based on Feature Model,

  ①③❺ category : some Category…
  ①❸④ category : some Category…
  ① category : some Category①
  Need a rule to delete first two model?

  Original model: ⑤sig Label {}⑤

  ❶❷❸❹⑤❻ sig Label {}…
  ①❷❸❹⑤❻ sig Label {}…
  ❶②❸❹⑤❻ sig Label {}…
  ❶❷③❹⑤❻ sig Label {}…
  ①②❸❹⑤❻ sig Label {}…
  ①❷③❹⑤❻ sig Label {}…
  ❶❷③④⑤❻ sig Label {}…
  ①❷③④⑤❻ sig Label {}…
  ❶❷③④⑤⑥ sig Label {}…
  ①❷③④⑤⑥ sig Label {}…

  After Merge sigs, we get
  ❸❹⑤❻ sig Label {}…
  ❷③④⑤ sig Label {}…

  Remove features based on Feature Model.
  ❸❹⑤❻ (⑥❹some none❹⑥)-> ❸❹⑤(④❸some none❸④) -> ❸⑤
  ❷③④⑤ (②③some none③②) -> ③④⑤(④❸some none❸④)-> ③⑤

Easily to do by hand.

- The order of the composition does not change the result model, but can make the process much complex.

Merge sigs by hand.

1. Vending machine

(Features:① No free drinks ② Multiple selection, number of products: 4)

(product❶❷)

```
❶❷ abstract sig State {}❷❶
❶❷ one sig Ready, Selected extends State {}❷❶

❶❷ sig Product {}❷❶

❶❷sig Time {
        state : one State,
        selection : lone Product,
        stock : Product -> one Int,
}❷❶
```

(product①❷)

```
①❷abstract sig State {}❷①
①❷one sig Ready, Selected extends State {}❷①
①❷one sig Served extends State {}❷①

①❷sig Product {
        price : one Int
}❷①

①❷sig Time {
        state : one State,
        selection : lone Product,
        stock : Product -> one Int,
        coins : one Int
}❷①
```

Step 1: separate signature declarations (how to implement?)

Merged model

```
❶❷ abstract sig State {}❷❶
❶❷ one sig Ready extends State {}❷❶
❶❷ one sig Selected extends State {}❷❶
❶❷ sig Product {}❷❶
❶❷sig Time {
        state : one State,
        selection : lone Product,
        stock : Product -> one Int,
}❷❶

①❷abstract sig State {}❷①
①❷one sig Ready extends State {}❷①
①❷one sig Selected extends State {}❷①
①❷one sig Served extends State {}❷①

①❷sig Product {price : one Int}❷①

①❷sig Time {
        state : one State,
        selection : lone Product,
        stock : Product -> one Int,
        coins : one Int
}❷①
```

Step 2: merge signatures (the order of sigs matters)

(Must merge parent signature first, otherwise, throw error when resolve sub signature. For example,

```
❶❷ abstract sig State {}❷❶
①❷abstract sig State {}❷①,
❷ one sig Ready extends State {}❷)
```

Merged model
❷ abstract sig State {}❷
❷ one sig Ready extends State {}❷
❷ one sig Selected extends State {}❷

❷ sig Product { ①price : one Int①}❷
❷sig Time {
❶state : one State❶,
❶selection : lone Product❶,
❶stock : Product -> one Int❶,
①state : one State①,
①selection : lone Product①,
①stock : Product -> one Int①,
①coins : one Int①
}❷
①❷one sig Served extends State {}❷①

Step 3: merge Fields

Merged model
❷ abstract sig State {}❷
❷ one sig Ready extends State {}❷
❷ one sig Selected extends State {}❷
❷ sig Product { ①price : one Int①}❷

❷sig Time {
state : one State ,
stock : Product -> one Int ,
selection : lone Product ,
①coins : one Int①
}❷

①❷one sig Served extends State {}❷①

Step 4: Add product ❶② (nothing can be merge)

Merged model
❷ abstract sig State {}❷
❷ one sig Ready extends State {}❷
❷ one sig Selected extends State {}❷
❷ sig Product { ①price : one Int①}❷

❷sig Time {
state : one State ,
stock : Product -> one Int ,
selection : lone Product ,
①coins : one Int①
}❷
①❷one sig Served extends State {}❷①
❶②abstract sig State {}②❶

product ❶②
❶②one sig Ready extends State {}②❶
❶②one sig Selected extends State {}②❶

❶②sig Product {

}②❶

❶②sig Time {
state : one State,
quantity : Product -> one Int,
stock : Product -> one Int,

}②❶

Step 5: Add product ①②

Merged model
❷ abstract sig State {}❷
❷ one sig Ready extends State {}❷

❷ one sig Selected extends State {}❷
❷ sig Product { ①price : one Int①}❷
❷sig Time {

```
        state : one State ,
        stock : Product -> one Int ,
         selection : lone Product ,
        ①coins : one Int①
}❷
```

```
①❷one sig Served extends State {}❷①
❶②abstract sig State {}②❶
❶②one sig Ready extends State {}②❶
❶②one sig Selected extends State {}②❶

❶②sig Product {}②❶
❶②sig Time {
        state : one State,
```

Step 6: merge signatures

```
        Merged model
❷ abstract sig State {}❷
❷ one sig Ready extends State {}❷
❷ one sig Selected extends State {}❷
❷ sig Product {
        ①price : one Int①
        }❷
❷sig Time {
        state : one State ,
        stock : Product -> one Int ,
         selection : lone Product ,
        ①coins : one Int①
        }❷
①one sig Served extends State {}①

②abstract sig State {}②
②one sig Ready extends State {}②
②one sig Selected extends State {}②

②sig Product { ①price : one Int① }②
②sig Time {
        ❶state : one State❶,
        ❶quantity : Product -> one Int❶,
        ❶stock : Product -> one Int❶,
        ①state : one State①,
        ①quantity : Product -> one Int①,
        ①stock : Product -> one Int①,
        ①coins : one Int①

}②
```

Step 6: merge signatures generate in step 5

```
        quantity : Product -> one Int,
        stock : Product -> one Int,

}②❶
```

```
        product ①②
①②abstract sig State {}②①
①②one sig Ready extends State {}②①
①②one sig Selected extends State {}②①
①②one sig Served extends State {}②①

①②sig Product {
        price : one Int
}②①

①②sig Time {
        state : one State,
        quantity : Product -> one Int,
        stock : Product -> one Int,
        coins : one Int
}②①
```

```
        Merged model
abstract sig State {}
 one sig Ready extends State {}
 one sig Selected extends State {}
sig Product {
        ❷①price : one Int①❷,
        ②①price : one Int①②
        }
sig Time {
        ❷state : one State ❷,
        ❷stock : Product -> one Int ❷,
        ❷selection : lone Product ❷,
        ❷①coins : one Int①❷,
        ②❶state : one State❶②,
        ②❶quantity : Product -> one Int❶②,
        ②❶stock : Product -> one Int❶②,
        ②①state : one State①②,
        ②①quantity : Product -> one Int①②,
        ②①stock : Product -> one Int①②,
        ②①coins : one Int①②
        }
①one sig Served extends State {}①
```

Step 6: merge fields

```
        Merged model
abstract sig State {}
 one sig Ready extends State {}
 one sig Selected extends State {}
sig Product {
        ①price : one Int①,
        }
sig Time {
        state : one State ,
        stock : Product -> one Int ,
        ❷selection : lone Product ❷,
        ①coins : one Int①,
        ②quantity : Product -> one Int②,
        }
①one sig Served extends State {}①
```

The result is the same with the origin one. The order not matters when do composition merge.
Get the same result when do 4 products together.

2. Grandpa

(Features: ① Bible② Marriage③ Forbid incest, FM: -- ③ requires ②)

```
        Product ❶❷❸
❶❷❸abstract sig Person {
        parents: set Person
}❸❷❶
❶❷❸sig Man extends Person {}❸❷❶
❶❷❸sig Woman extends Person {}❸❷❶

fact FeatureModel {
        -- ③ requires ②
        ③❷some none❷③
}
```

```
        Product ①❷❸
①❷❸abstract sig Person {
        parents: set Person
}❸❷①
①❷❸sig Man extends Person {} ❸❷①
①❷❸sig  Woman extends Person {}❸❷①

①❷❸one sig Eve extends Woman {}❸❷①
①❷❸one sig Adam extends Man {}❸❷①
```

Step 1, merge sigs

result model
❷❸abstract sig Person{
❶parents: set Person❶,
①parents: set Person①
}❸❷
❷❸sig Man extends Person{ }❸❷
❷❸sig Woman extends Person{ }❸❷

①❷❸one sig Eve extends Woman {}❸❷①
①❷❸one sig Adam extends Man {}❸❷①

Step 2: merge fields

result model
❷❸abstract sig Person{
parents: set Person
}❸❷
❷❸sig Man extends Person{ }❸❷
❷❸sig Woman extends Person{ }❸❷

①❷❸one sig Eve extends Woman {}❸❷①
①❷❸one sig Adam extends Man {}❸❷①

Step 3: add product ❶②❸ (nothing can be merge)

merged model
❷❸abstract sig Person{
parents: set Person
}❸❷
❷❸sig Man extends Person{ }❸❷
❷❸sig Woman extends Person{ }❸❷

①❷❸one sig Eve extends Woman {}❸❷①
①❷❸one sig Adam extends Man {}❸❷①

product ❶②❸
❶②❸abstract sig Person {
spouse: lone Person,
parents: set Person
}❸②❶
❶②❸sig Man extends Person{}❸②❶
❶②❸sig Woman extends Person {}❸②❶

Step 4: Add product ①②❸

merged model
❷❸abstract sig Person{
parents: set Person
}❸❷
❷❸sig Man extends Person{ }❸❷
❷❸sig Woman extends Person{ }❸❷
①❷❸one sig Eve extends Woman {}❸❷①
①❷❸one sig Adam extends Man {}❸❷①

❶②❸abstract sig Person {
spouse: lone Person,
parents: set Person
}❸②❶
❶②❸sig Man extends Person{}❸②❶
❶②❸sig Woman extends Person {}❸②❶

product ①②❸
①②❸ abstract sig Person {
spouse: lone Person,
parents: set Person
} ❸②①
①②❸ sig Man extends Person{} ❸②①
①②❸ sig Woman extends Person {} ❸②①

①②❸ one sig Eve extends Woman {} ❸②①
①②❸ one sig Adam extends Man {} ❸②①

Step 5: merge sigs

result model
❸sig Man extends Person{ }❸
❸sig Woman extends Person{ }❸
①❸one sig Eve extends Woman{ }❸①
❸abstract sig Person{
①②spouse: lone Person②①,
①②parents: set Person②①,
❶②spouse: lone Person②❶,
❶②parents: set Person②❶,
❷parents: set Person❷
}❸
①❸one sig Adam extends Man{ }❸①

Step 6: Merge fields, delete duplicated fields

result model
❸sig Man extends Person{ }❸
❸sig Woman extends Person{ }❸
①❸one sig Eve extends Woman{ }❸①
❸abstract sig Person{
②spouse: lone Person② ,
parents: set Person
}❸
①❸one sig Adam extends Man{ }❸①

Step 7: Add product ❶②③(nothing can be merge)

merged model
❸sig Man extends Person{ }❸
❸sig Woman extends Person{ }❸
①❸one sig Eve extends Woman{ }❸①
❸abstract sig Person{
②spouse: lone Person② ,
parents: set Person
}❸
①❸one sig Adam extends Man{ }❸①

Product❶②③
❶②③ abstract sig Person {
spouse: lone Person,
parents: set Person
} ③②❶
❶②③ sig Man extends Person{} ③②❶
❶②③ sig Woman extends Person {} ③②❶

Step 7: Add product ①②③

merged model
❸sig Man extends Person{ }❸
❸sig Woman extends Person{ }❸
①❸one sig Eve extends Woman{ }❸①
❸abstract sig Person{
②spouse: lone Person② ,
parents: set Person
}❸
①❸one sig Adam extends Man{ }❸①
❶②③ abstract sig Person {
spouse: lone Person,
parents: set Person
} ③②❶
❶②③ sig Man extends Person{} ③②❶
❶②③ sig Woman extends Person {} ③②❶

Product ①②③
①②③abstract sig Person {
spouse: lone Person,
parents: set Person
}③②①
①②③sig Man extends Person{}③②①
①②③sig Woman extends Person {}③②①

①②③one sig Eve extends Woman {}③②①
①②③one sig Adam extends Man {}③②①

Step 8: merge sigs

Result model
❸sig Man extends Person{ }❸
❸sig Woman extends Person{ }❸
①❸one sig Eve extends Woman{ }❸①
❸abstract sig Person{
②spouse: lone Person② ,
parents: set Person
}❸
①❸one sig Adam extends Man{ }❸①
②③abstract sig Person{
❶spouse: lone Person❶,
❶parents: set Person❶,
①spouse: lone Person①,
①parents: set Person①
}③②
②③sig Man extends Person{ }③②
②③sig Woman extends Person {}③②

①②③one sig Eve extends Woman {}③②①
①②③one sig Adam extends Man {}③②①

Step 8: merge Fields

Result model
❸sig Man extends Person{ }❸
❸sig Woman extends Person{ }❸

①❸one sig Eve extends Woman{ }❸①

❸abstract sig Person{
②spouse: lone Person② ,
parents: set Person
}❸

①❸one sig Adam extends Man{ }❸①
②③abstract sig Person{
parents: set Person ,
spouse: lone Person ,
}③②

②③sig Man extends Person{ }③②
②③sig Woman extends Person {}③②

①②③one sig Eve extends Woman {}③②①
①②③one sig Adam extends Man {}③②①

Step 8: Remove redundant Features based on FM
fact FeatureModel {
-- ③ requires ②
③❷some none❷③
}

Result model
❸sig Man extends Person{ }❸
❸sig Woman extends Person{ }❸

①❸one sig Eve extends Woman{ }❸①

❸abstract sig Person{
②spouse: lone Person② ,
parents: set Person
}❸

①❸one sig Adam extends Man{ }❸①
③abstract sig Person{
parents: set Person ,
spouse: lone Person ,
}③

③sig Man extends Person{ }③
③sig Woman extends Person {}③
①③one sig Eve extends Woman {}③①
①③one sig Adam extends Man {}③①

Step 9: Merge Sigs

Result model
sig Man extends Person{ }
sig Woman extends Person{ }

①one sig Eve extends Woman{ }①

①one sig Adam extends Man{ }①

abstract sig Person{
③spouse: lone Person③,
③parents: set Person③,
②❸spouse: lone Person❸②,
❸parents: set Person❸
}

Step 10: Merge fields,
sig Man extends Person{ }
sig Woman extends Person{ }
①one sig Eve extends Woman{ }①
①one sig Adam extends Man{ }①

abstract sig Person{
③spouse: lone Person③,
②❸spouse: lone Person❸②,
parents: set Person
}

Step 11: Add features according to FM.
sig Man extends Person{ }
sig Woman extends Person{ }
①one sig Eve extends Woman{ }①
①one sig Adam extends Man{ }①

abstract sig Person{
parents: set Person ,
③②spouse: lone Person②③,

```
    ②❸spouse: lone Person❸②,
    }
sig Man extends Person{ }
```

Step 12: merge Field.

```
sig Man extends Person{ }
sig Woman extends Person{ }
①one sig Eve extends Woman{ }①
①one sig Adam extends Man{ }①

abstract sig Person{
    ② spouse: lone Person ②,
     parents: set Person
    }
```

The result is the same with the origin one. The order not matters when do composition merge. Get the same result when do 6 products together.

3. Alloy4fun (Features 4, FM ④ requires ③)

Sig Link{} exist in base model, omitted.

Variant ❶❷❸❹

```
❶❷❸❹sig Model {
        public : lone Link,
}❹❸❷❶
```

Variant ①❷❸❹

```
①❷❸❹sig Model {
        derivationOf : lone Model,
        public : lone Link,
}❹❸❷①
```

Step 1: merge sig

Result model

```
❷❸❹sig Model {
        ❶public : lone Link❶,
        ①derivationOf : lone Model①,
        ①public : lone Link①,

}❹❸❷
```

Step 2: merge Field

Result model

```
❷❸❹sig Model {
         public : lone Link ,
        ①derivationOf : lone Model①
}❹❸❷
```

Step 3: Add Product ❶②❸❹,❶❷③❹,①②❸❹

Merged model

```
❷❸❹sig Model {
         public : lone Link ,
        ①derivationOf : lone Model①
}❹❸❷
```

Variant ❶②❸❹

```
❶②❸❹sig Model {
        public : lone Link,
        secret : lone Link,

}❹❸②❶
```

❶②❸❹sig Secret in Model {}❹❸②❶

Variant❶❷③❹
❶❷③❹sig Model {
public : lone Link,
command : lone Command
}❹③❷❶

❶❷③❹sig Command {}❹③❷❶

Step 4: merge sigs

Merged model
❷❸❹sig Model {
public : lone Link ,
①derivationOf : lone Model①
}❹❸❷

②❸❹sig Model {
❶public : lone Link❶,
❶secret : lone Link❶,
①derivationOf : lone Model①,
①public : lone Link①,
①secret : lone Link①,
}❹❸②

②❸❹sig Secret in Model {}❹❸②

❶❷③❹sig Model {
public : lone Link,
command : lone Command
}❹③❷❶

❶❷③❹sig Command {}❹③❷❶

Step 5 : Merge sig Model

❸❹sig Model {
❷ public : lone Link❷ ,
❷①derivationOf : lone Model①❷
②❶public : lone Link❶②,
❶②secret : lone Link②❶,
②①derivationOf : lone Model①②,
②①public : lone Link①②,
①②secret : lone Link②①,

}❹❸

②❸❹sig Secret in Model {}❹❸②
❶❷③❹sig Model {
public : lone Link,
command : lone Command
}❹③❷❶
❶❷③❹sig Command {}❹③❷❶

Step 6: Merge Field

❸❹sig Model {

Variant ①②❸❹
①②❸❹sig Model {
derivationOf : lone Model,
public : lone Link,
secret : lone Link,
}❹❸②①

①②❸❹sig Secret in Model {}❹❸②①

public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
}❹❸

②❸❹sig Secret in Model {}❹❸②
❶❷③❹sig Model {
public : lone Link,
command : lone Command
}❹③❷❶
❶❷③❹sig Command {}❹③❷❶

Step 7: Add product ①❷③❹

❸❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
}❹❸

②❸❹sig Secret in Model {}❹❸②
❶❷③❹sig Model {
public : lone Link,
command : lone Command
}❹③❷❶
❶❷③❹sig Command {}❹③❷❶

Product ①❷③❹
①❷③❹sig Model {
①derivationOf : lone Model①,
public : lone Link,
③command : lone Command③
}❹③❷①

①❷③❹sig Command {}❹③❷①

Step 8: Merge sig

❸❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
}❹❸

②❸❹sig Secret in Model {}❹❸②
❷③❹sig Model {
❶public : lone Link❶,
❶command : lone Command❶,
①derivationOf : lone Model①,
①public : lone Link①,
①command : lone Command①
}❹③❷
❷③❹sig Command {}❹③❷

Step 9: Merge Field

❸❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
}❹❸

②❸❹sig Secret in Model {}❹❸②
❷③❹sig Model {
public : lone Link,
command : lone Command,
①derivationOf : lone Model①,
}❹③❷

❷③❹sig Command {}❹③❷

Step 10 : Add product

Merged model

❸❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
}❹❸

②❸❹sig Secret in Model {}❹❸②

❷③❹sig Model {
public : lone Link,
command : lone Command,
①derivationOf : lone Model①,
}❹③❷

❷③❹sig Command {}❹③❷

product ❶②③❹

❶②③❹sig Model {
public : lone Link,
②secret : lone Link②,
③command : lone Command③
}❹③②❶

❶②③❹sig Secret in Model {}❹③②❶

❶②③❹sig Command {}❹③②❶

Product❶❷③④

❶❷③④sig Model {
public : lone Link,
③command : lone Command③
}④③❷❶

Merge sigs

Merged model

❸❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
}❹❸

②❸❹sig Secret in Model {}❹❸②

❷③❹sig Model {
public : lone Link,
command : lone Command,
①derivationOf : lone Model①,
}❹③❷

❷③❹sig Command {}❹③❷

②③❹sig Model {
❶public : lone Link❶,
❶②secret : lone Link②❶,
❶③command : lone Command③❶,
①derivationOf : lone Model①,
①public : lone Link①,
①②secret : lone Link②①,
①③command : lone Command③①
}❹③②

❶❷③④sig Command {}④③❷❶

❶❷③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link
}④③❷❶

Product ①②③❹

①②③❹sig Model {
①derivationOf : lone Model①,
public : lone Link,
②secret : lone Link②,
③command : lone Command③
}❹③②①

①②③❹sig Secret in Model {}❹③②①

①②③❹sig Command {}❹③②①

②③❹sig Secret in Model {}❹③②
②③❹sig Command {}❹③②

❶❷③④sig Model {
public : lone Link,
③command : lone Command③
}④③❷❶

❶❷③④sig Command {}④③❷❶
❶❷③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link
}④③❷❶
Merge sig Model

Merged model
❹sig Model {
❸public : lone Link❸,
❸①derivationOf : lone Model①❸,
❸②secret : lone Link②❸,
③❷public : lone Link❷③,
③❷command : lone Command❷③,
③❷①derivationOf : lone Model①❷③,
③②❶public : lone Link❶②③,
③❶②secret : lone Link②❶③,
②❶③command : lone Command③❶②,
③②①derivationOf : lone Model①②③,
③②①public : lone Link①②③,
③①②secret : lone Link②①③,
②①③command : lone Command③①②

}❹❸

②❹sig Secret in Model {}❹②
③❹sig Command {}❹③
❶❷③④sig Model {
public : lone Link,
③command : lone Command③
}④③❷❶

❶❷③④sig Command {}④③❷❶
❶❷③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link
}④③❷❶

Merge Field

Merged model
❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
③command : lone Command③,
}❹

②❹sig Secret in Model {}❹②

③❹sig Command {}❹③
❶❷③④sig Model {
public : lone Link,
③command : lone Command③
}④③❷❶

❶❷③④sig Command {}④③❷❶
❶❷③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link
}④③❷❶

Add variant --①❷③④

Merged model
❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
③command : lone Command③,
}❹

②❹sig Secret in Model {}❹②
③❹sig Command {}❹③
❶❷③④sig Model {
public : lone Link,
③command : lone Command③
}④③❷❶

❶❷③④sig Command {}④③❷❶
❶❷③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link
}④③❷❶

Merge sig

Result model
❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
③command : lone Command③,
}❹

②❹sig Secret in Model {}❹②
③❹sig Command {}❹③
❷③④sig Model {
❶public : lone Link❶,
❶③command : lone Command③❶,
①derivationOf : lone Model①,
①public : lone Link①,
①③command : lone Command③①

}④③❷

❷③④sig Command {}④③❷
❷③④sig Instance {

variant --①❷③④
①❷③④sig Model {
①derivationOf : lone Model①,
public : lone Link,
③command : lone Command③
}④③❷①
①❷③④sig Command {}④③❷①

①❷③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link
}④③❷①

```
        ❶instanceOf : one Command❶,
        ❶model : set Model❶,
        ❶link : one Link❶,
        ①instanceOf : one Command①,
        ①model : set Model①,
        ①link : one Link①
}④③❷
```

Merge Field

Result model

```
❹sig Model {
         public : lone Link,
        ①derivationOf : lone Model①,
        ②secret : lone Link②,
        ③command : lone Command③,
}❹

②❹sig Secret in Model {}❹②
③❹sig Command {}❹③
❷③④sig Model {
        public : lone Link,
        ③command : lone Command③,
        ①derivationOf : lone Model①,
}④③❷

❷③④sig Command {}④③❷
❷③④sig Instance {
        instanceOf : one Command,
        model : set Model,
        link : one Link,
}④③❷
```

Add product

merged model

```
❹sig Model {
         public : lone Link,
        ①derivationOf : lone Model①,
        ②secret : lone Link②,
        ③command : lone Command③,
}❹
②❹sig Secret in Model {}❹②
③❹sig Command {}❹③
❷③④sig Model {
        public : lone Link,
        ③command : lone Command③,
        ①derivationOf : lone Model①,
}④③❷

❷③④sig Command {}④③❷
❷③④sig Instance {
        instanceOf : one Command,
        model : set Model,
        link : one Link,
}④③❷
```

Product ❶②③④

```
❶②③④sig Model {
        public : lone Link,
        ②secret : lone Link②,
        ③command : lone Command③
}④③②❶
❶②③④sig Secret in Model {}④③②❶
❶②③④sig Command {}④③②❶
❶②③④sig Instance {
        instanceOf : one Command,
        model : set Model,
        link : one Link
```

}④③②❶

Product ①②③④
①②③④sig Model {
①derivationOf : lone Model①,
public : lone Link,
②secret : lone Link②,
③command : lone Command③

Merge sig

merged model
❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
③command : lone Command③,
}❹
②❹sig Secret in Model {}❹②
③❹sig Command {}❹③
❷③④sig Model {
public : lone Link,
③command : lone Command③,
①derivationOf : lone Model①,
}④③❷

❷③④sig Command {}④③❷
❷③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link,
}④③❷

②③④sig Model {
❶public : lone Link❶,
❶②secret : lone Link②❶,
❶③command : lone Command③❶,
①derivationOf : lone Model①,
①public : lone Link①,
①②secret : lone Link②①,
①③command : lone Command③①

}④③②
②③④sig Secret in Model {}④③②
②③④sig Command {}④③②
②③④sig Instance {
❶instanceOf : one Command❶,
❶model : set Model❶,
❶link : one Link❶,
①instanceOf : one Command①,
①model : set Model①,
①link : one Link①

}④③②

Step merge sigs

}④③②①
①②③④sig Secret in Model {}④③②①
①②③④sig Command {}④③②①
①②③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link
}④③②①

merged model
❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
③command : lone Command③,
}❹
②❹sig Secret in Model {}❹②
③sig Command {}③
③④sig Model {
❷public : lone Link❷,
❷③command : lone Command③❷,
❷①derivationOf : lone Model①❷,
②❶public : lone Link❶②,
❶②secret : lone Link②❶,
②❶③command : lone Command③❶②,
②①derivationOf : lone Model①②,
②①public : lone Link①②,
①②secret : lone Link②①,
②①③command : lone Command③①②

}④③❷

③④sig Instance {
❷instanceOf : one Command❷,
❷model : set Model❷,
❷link : one Link❷,
②❶instanceOf : one Command❶②,
②❶model : set Model❶②,
②❶link : one Link❶②,
②①instanceOf : one Command①②,
②①model : set Model①②,
②①link : one Link①②
}④③
②③④sig Secret in Model {}④③②

Merge Fields

merged model
❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
③command : lone Command③,
}❹
②❹sig Secret in Model {}❹②
③sig Command {}③
③④sig Model {
public : lone Link,
③command : lone Command③,
①derivationOf : lone Model①,
②secret : lone Link②,
}④③

③④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link,

}④③
②③④sig Secret in Model {}④③②

Remove Redundant Features based on FM(③④---④)

result model
❹sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
③command : lone Command③,
}❹
②❹sig Secret in Model {}❹②
③sig Command {}③
④sig Model {
public : lone Link,
③command : lone Command③,
①derivationOf : lone Model①,
②secret : lone Link②,
}④

④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link,
}④
②④sig Secret in Model {}④②

Merge sigs(Model, Sercet)

sig Model {
❹ public : lone Link❹,
❹①derivationOf : lone Model①❹,
❹②secret : lone Link②❹,
❹③command : lone Command③❹,
④public : lone Link④,
④③command : lone Command③④,
④①derivationOf : lone Model①④,
④②secret : lone Link②④,

}
②sig Secret in Model {}②
③sig Command {}③

④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link,
}④

Merge Fields

sig Model {
public : lone Link,
①derivationOf : lone Model①,
②secret : lone Link②,
③command : lone Command③
}

②sig Secret in Model {}②
③sig Command {}③

④sig Instance {
instanceOf : one Command,
model : set Model,
link : one Link,
}④

The result is the same with the origin one if we add feature ③ to the Instance Signature based on the Feature Mode. The order not matters when do composition merge.

4. Ecommerce

For example, Sig Image{}
--❶❷❸④❺
--①❷❸④❺
--①②❸④❺
--①❷③④❺
--①❷❸④⑤
--①②③④❺
--①②❸④⑤
--①❷③④⑤
--①②③④⑤

Step 1： Merge sig

①②❸④❺ (② implies ①)
❷❸④❺
①❸④⑤ (⑤ implies ①)
①③④ (③ implies ①)

Step 2： Remove Feature based on FM
②❸④❺
❷❸④❺
❸④⑤
③④

Step 3： Merge sig

❸④❺
❸④⑤
①③④

❸④
③④

Result： ④sig Image {}④

For ❶catalog : one Catalog❶
❶❷❸④❺
❶❷❸④❺

Merge result： ❶❷❸❺ catalog : one Catalog ❶❷❸❺
Remove Features based on FM

For : ①category : some Category①,

After merge
①③❺
①❸④
①④⑤((⑤ implies ①) and (⑤ implies ④)) -→ ①
Remove features

①③❺
①❸④
① ?

Graph(number of Features 6)

For ⑤sig Label {}⑤

❶❷❸❹⑤❻
①❷❸❹⑤❻
❶②❸❹⑤❻
❶❷③❹⑤❻
①②❸❹⑤❻
①❷③❹⑤❻
❶❷③④⑤❻
①❷③④⑤❻
❶❷③④⑤⑥
①❷③④⑤⑥

After Merge
❸❹⑤❻ (⑥❹some none❹⑥)-> ❸❹⑤(④❸some none❸④) -> ❸⑤
❷③④⑤ (②③some none③②) -> ③④⑤(④❸some none❸④)-> ③⑤