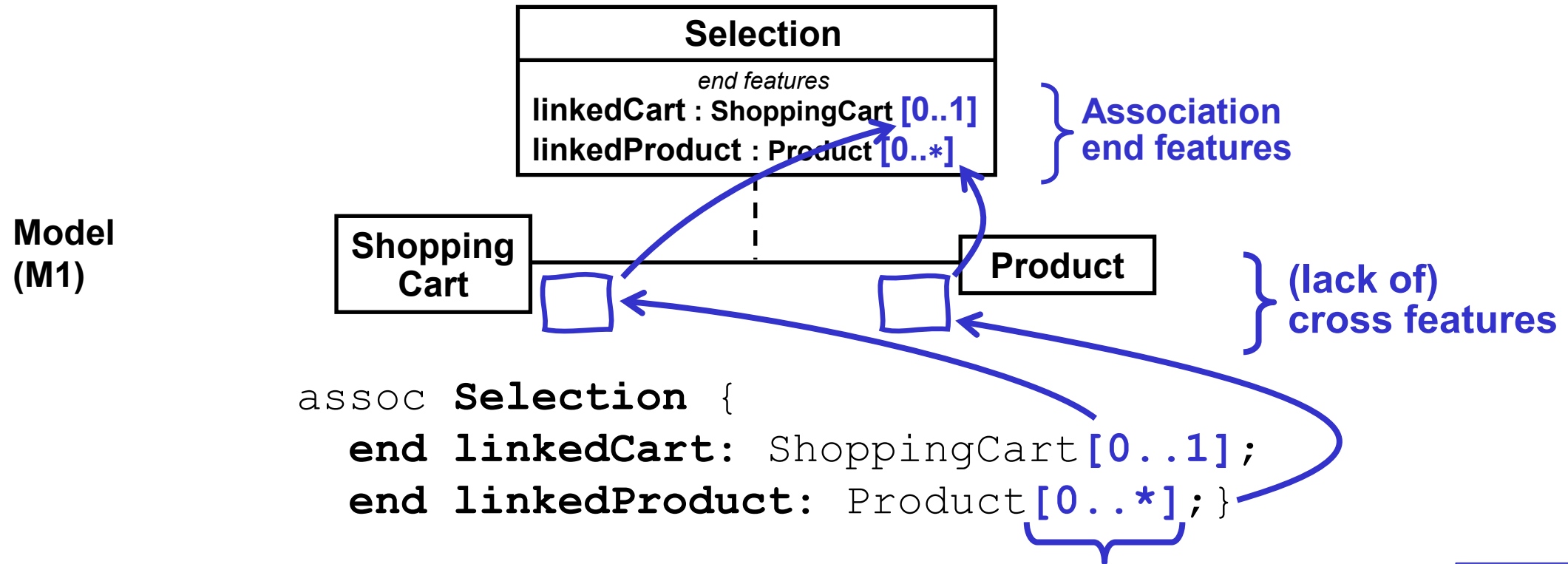# Redundancy in association end multiplicities, ordering, and uniqueness

- **See background in https://issues.omg.org/secure/attachment/19973/sysml-1-assoc-review-10.pdf**

- **See term definitions in KERML-34**

Assoc Syntax (multiplicity …)

Model (M1)

Selection
end features **linkedCart : ShoppingCart** [0..1] **linkedProduct : Product** [0..*]

Association end features

Shopping Cart

Product

(lack of) cross features

```
assoc Selection {
    end linkedCart: ShoppingCart[0..1];
    end linkedProduct: Product[0..*];}
```

- **End multiplicity is about "cross navigation"**
  - **Useful when cross features do not exist**
    - **As in one-way and non-navigable associations.**
  - **But when they exist …**

Same for ordering and uniqueness

25

Assoc Syntax **Redundancy** (multiplicity ...)

Model (M1)

Selection

end features
linkedCart : ShoppingCart [0..1]
linkedProduct : Product [0..*]

} **Association end features**

Shopping Cart

inCart
0..1

selected Product
0..*

Product

} **Cross features**

Binary only

```
class Product { feature inCart: ShoppingCart [0..1]; }
class ShoppingCart { feature selectedProduct: Product[0..*]; }
assoc Selection {
    end linkedCart: ShoppingCart[0..1] subsets linkedProduct.inCart;
    end linkedProct: Product[0..*] subsets linkedCart.selectedProduct;}
```

} **Cross features**

- **Multiplicity redundant and possibly conflicting**
  - between (binary) end and cross features.
  - due to redundancy in **abstract syntax**.

Same for ordering and uniqueness