

Funcons-beta: Classes *

The PPlanCompS Project

Classes.cbs | PLAIN | PRETTY

Classes

```
[ Datatype  classes
  Funcon    class
  Funcon    class-instantiator
  Funcon    class-feature-map
  Funcon    class-superclass-name-sequence
  Funcon    class-name-tree
  Funcon    is-subclass-name
  Funcon    class-name-single-inheritance-feature-map ]
```

Datatype classes ::= class($_ : \text{thunks}(\text{references}(\text{objects}))$, $_ : \text{environments}$, $_ : \text{identifiers}^*$)

$\text{class}(\text{Thunk}, \text{Env}, C^*)$ is a class with: * a thunk Thunk for instantiating the class, * an environment Env with the features declared by the class, and * a sequence C^* of names of direct superclasses. $\text{class}(\text{Thunk}, \text{Env})$ is a base class, having no superclasses. $\text{class}(\text{Thunk}, \text{Env}, C)$ is a class with a single superclass.

Class instantiation forces its thunk to compute a reference to an object.

Features are inherited from superclasses. When features with the same name are declared in simultaneously inherited classes, the order of the superclass identifiers in C^* may affect resolution of references to features. Overloading of feature names is supported by using type maps as features.

The class table is represented by binding class names to classes. The class superclass hierarchy is assumed to be acyclic.

```
Funcon  class-instantiator( $\_ : \text{classes}$ ) :  $\Rightarrow \text{thunks}(\text{references}(\text{objects}))$ 
Rule    class-instantiator
          class( $\text{Thunk} : \text{thunks}(\_)$ ,  $\text{Envs} : \text{environments}$ ,  $C^* : \text{identifiers}^*$ )  $\rightsquigarrow$ 
          Thunk
```

```
Funcon  class-feature-map( $\_ : \text{classes}$ ) :  $\Rightarrow \text{environments}$ 
Rule    class-feature-map
          class( $\text{Thunk} : \text{thunks}(\_)$ ,  $\text{Env} : \text{environments}$ ,  $C^* : \text{identifiers}^*$ )  $\rightsquigarrow$ 
          Env
```

*Suggestions for improvement: plancomps@gmail.com.
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.

Funcon `class-superclass-name-sequence(_ : classes) : \Rightarrow identifiers*`

Rule `class-superclass-name-sequence`
`class(Thunk : thunks(_), Env : environments, C* : identifiers*) \rightsquigarrow`
`C*`

Funcon `class-name-tree(_ : identifiers) : \Rightarrow trees(identifiers)`

`class-name-tree` *C* forms a tree where the branches are the class name trees for the superclasses of *C*.

Rule `class-name-tree(C : identifiers) \rightsquigarrow`
`tree(`
`C,`
`interleave-map(`
`class-name-tree given,`
`class-superclass-name-sequence bound-value C))`

Funcon `is-subclass-name(C : identifiers, C' : identifiers) : \Rightarrow booleans`
 `\rightsquigarrow is-in-set(C, {forest-value-sequence class-name-tree C'})`

The result of `is-subclass-name(C, C')` does not depend on the order of the names in `forest-value-sequence class-name-tree C`

Funcon `class-name-single-inheritance-feature-map(C : identifiers) : \Rightarrow environments`
 `\rightsquigarrow map-override interleave-map(`
`class-feature-map bound-value given,`
`single-branching-sequence class-name-tree C)`

For multiple inheritance, different resolution orders can be specified by using different linearisations of the class name tree.