



:: LazySet

LazySet
 ↑
 UnionSet → ~~AbstractNonConvexSet~~

AbstractNonConvexSet
 ↑
 UnionSet

AbstractConvexSet
 ←
 AbstractNonConvexSet

Linear Map

<: AbstractConvexSet

"Trait"-based approach.

$\text{isconvextype}(\text{Type} \{ \text{LinearMap} \} N, \text{ST}, \text{NM}, \text{MT} \{ \}) = \dots$
 $= \text{isconvextype}(\text{ST})$.

```
function foo(X::LinearMap)
  if isconvextype(X)
    ----
  else
    ---
  end
end
```

```
abstract type IsConvexOrNot end
struct ConvexSet <: IsConvexOrNot end
struct NonConvexSet <: IsConvexOrNot end

-foo(X, ::ConvexSet) = ----
-foo(X, ::NonConvexSet) = ----
foo(X) = -foo(X, isconvextype(typeof(X)))
```

Cuando Julia recibe un valor X por primera vez, compila la función _foo adecuada dependiendo de si X es convex o no. La segunda vez que se llama esa función, ya tiene compilado el código para el tipo de X. En otras palabras, el dispatch ocurre en tiempo de compilación.