Scala type computation

Scala type computation

- Anonymous type functions:

```
({type F[A] = Either[A, String]})#F
```

- Case distinction performed with type members:

```
sealed trait Nat { type Plus[N<:Nat] <: Nat }
trait Zero extends Nat {
  type Plus[N <: Nat] = N
}
trait Succ[N <: Nat] extends Nat {
  type Plus[M <: Nat] = Succ[N#Plus[M]]
}
```

- We can even define a general type-level fold!