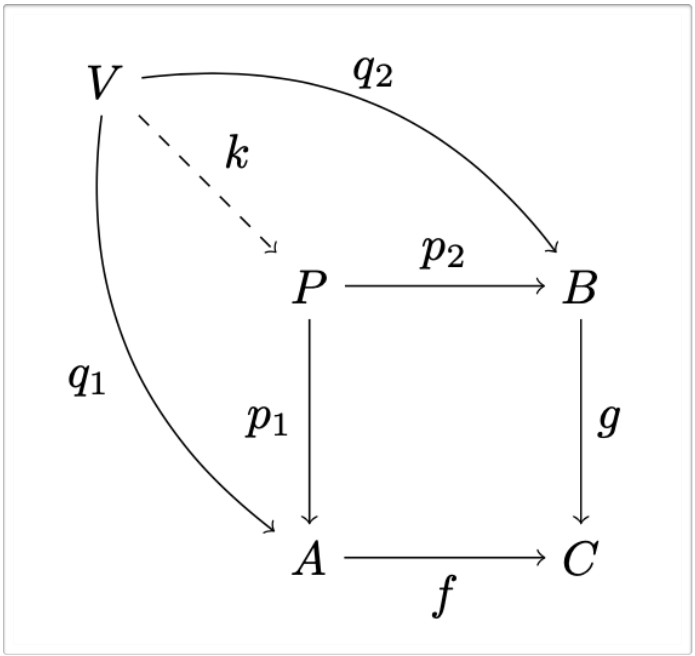
Category Theory for Computer Science

EECS 598, Winter 2022

Instructor: Max New

```haskell
instance Functor (Cont r) where
  fmap f m = Cont $ \k ->
    runCont m (k . f)


instance Monad (Cont r) where
  return = Cont $ \k -> k x
  m >>= mk = Cont $ \k ->
             runCont m $ \x ->
             mk x k
```



Category theory is a branch of mathematics developed to axiomatize structures of composition and formalize common constructions across disparate mathematical disciplines. Despite its highly abstract origins, category theory has become a common tool in the study of programming languages, with applications to type systems, functional programming, proof assistants and probabilistic programming languages. However, much introductory material on category theory assumes a great deal of mathematical background, making it difficult to learn for many computer scientists.

This course will serve as an introduction to category theory for computer science students, especially those interested in applications to formal logic and programming languages. Tentative topics include categories, functors, natural transformations, adjunctions, universal properties, monads, multicategories, and functorial semantics and functional programming applications. We will focus on the formal relationship between syntax and semantics as a unifying theme.

Prerequisites: Graduate standing or instructor permission. Mathematical maturity required.