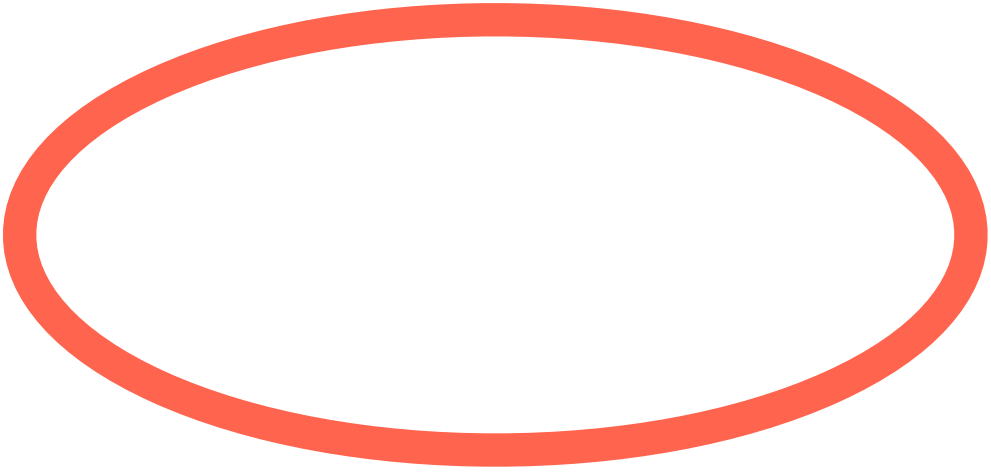
Exception Handling

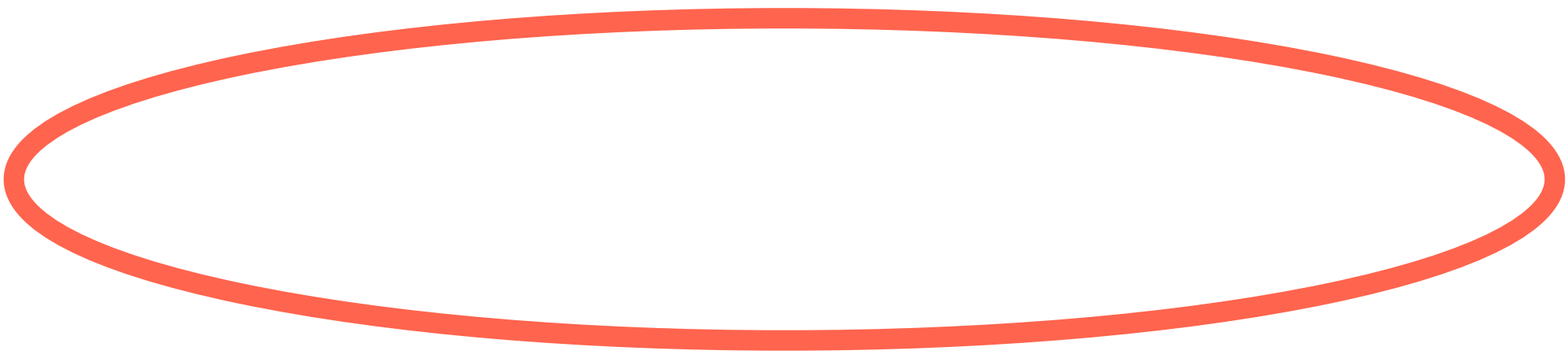
structured concurrency: no coroutine leaks

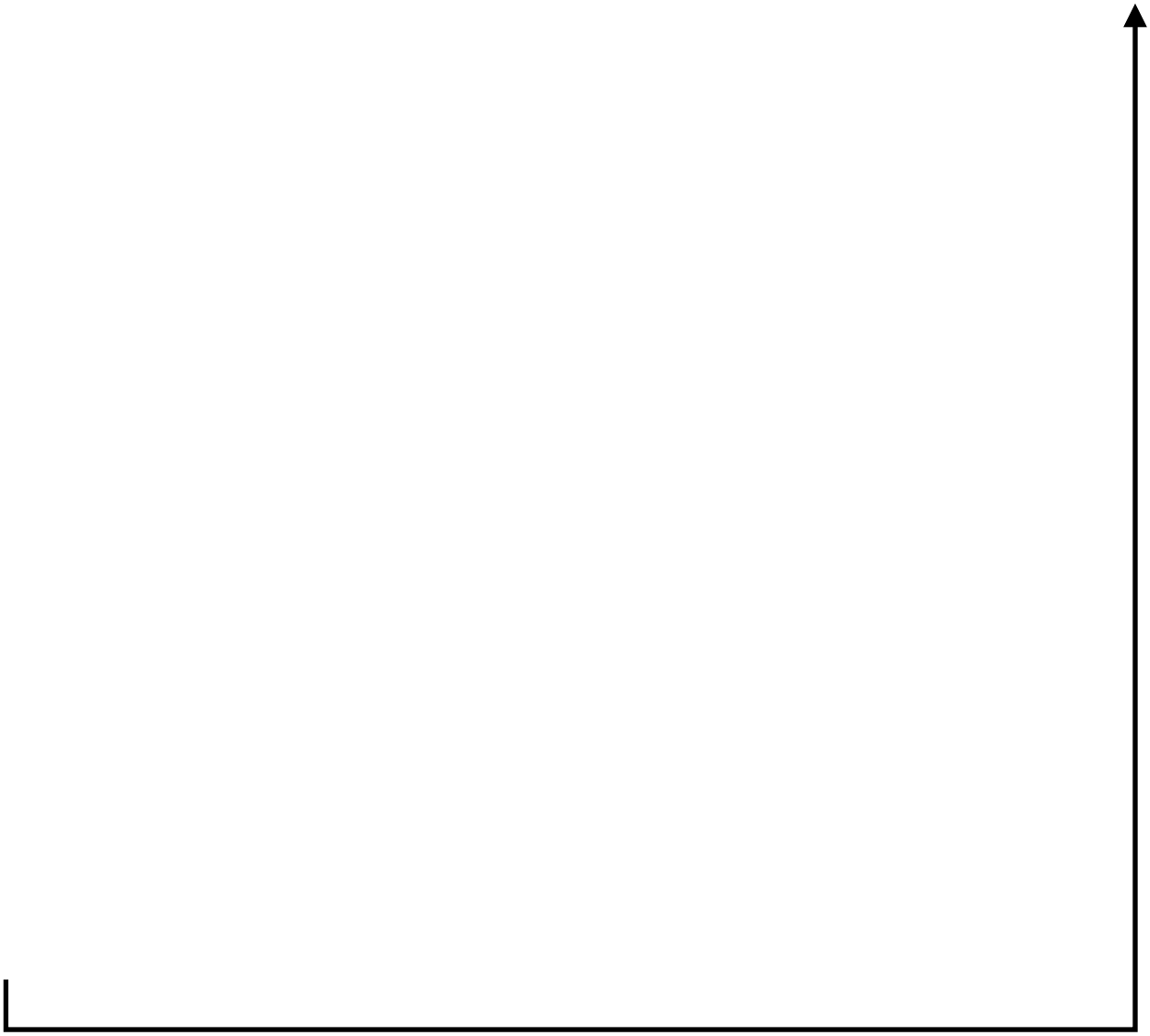
38

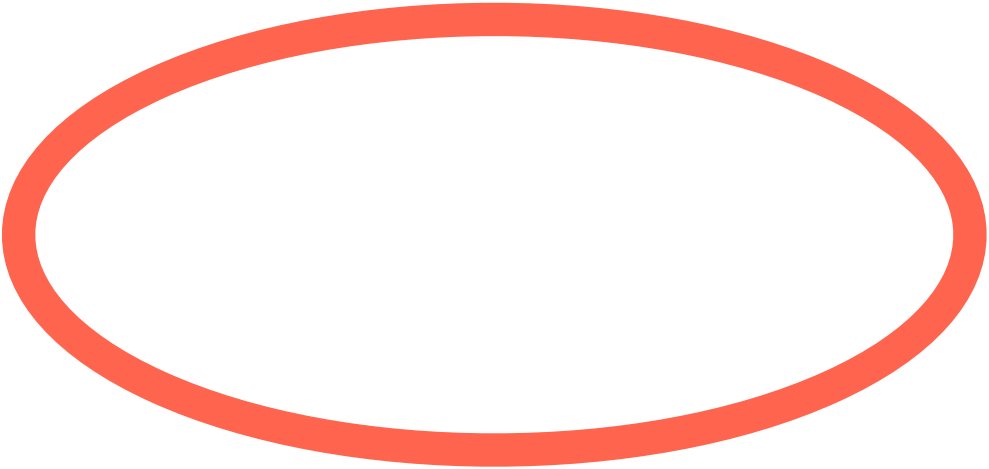
```
suspend fun bakePretzels(): List<FinishedPretzel> = coroutineScope {
    val oven = async { preheatOven(ColdOven) }
    val dough = async { prepareDough() }
    val uncookedPretzels = List(5) { async { shapePretzel(dough.await()) } }
    val bakedPretzels = async { bake(oven.await(), uncookedPretzels.awaitAll()) }
    val topping = async { prepareTopping() }
    bakedPretzels.await().map { finishPretzel(it, topping.await()) }
}
```

```
throw IllegalStateException("Broken Oven!")
```

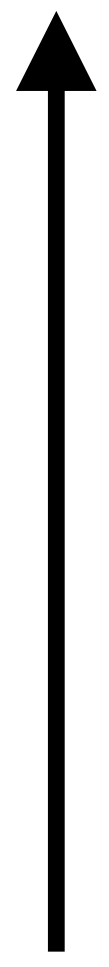









cancel



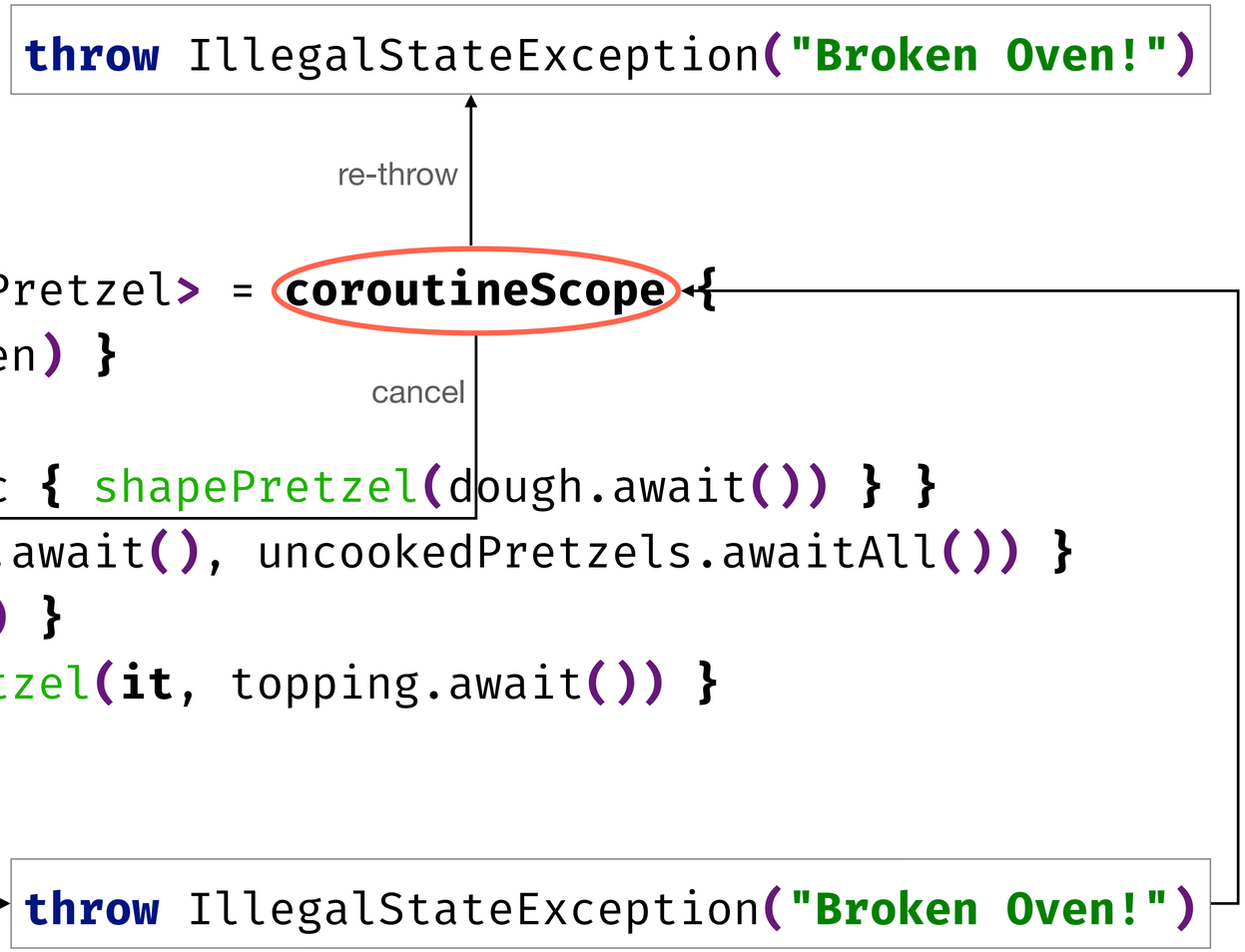
```
throw IllegalStateException("Broken Oven!")
```

re-throw

Exception Handling

structured concurrency: no coroutine leaks

```
suspend fun bakePretzels(): List<FinishedPretzel> = coroutineScope {
    val oven = async { preheatOven(ColdOven) }
    val dough = async { prepareDough() }
    val uncookedPretzels = List(5) { async { shapePretzel(dough.await()) } }
    val bakedPretzels = async { bake(oven.await(), uncookedPretzels.awaitAll()) }
    val topping = async { prepareTopping() }
    bakedPretzels.await().map { finishPretzel(it, topping.await()) }
}
```



Exception Handling