ReactiveCocoa入门到实战

第三周 ReactiveCocoa高级操作

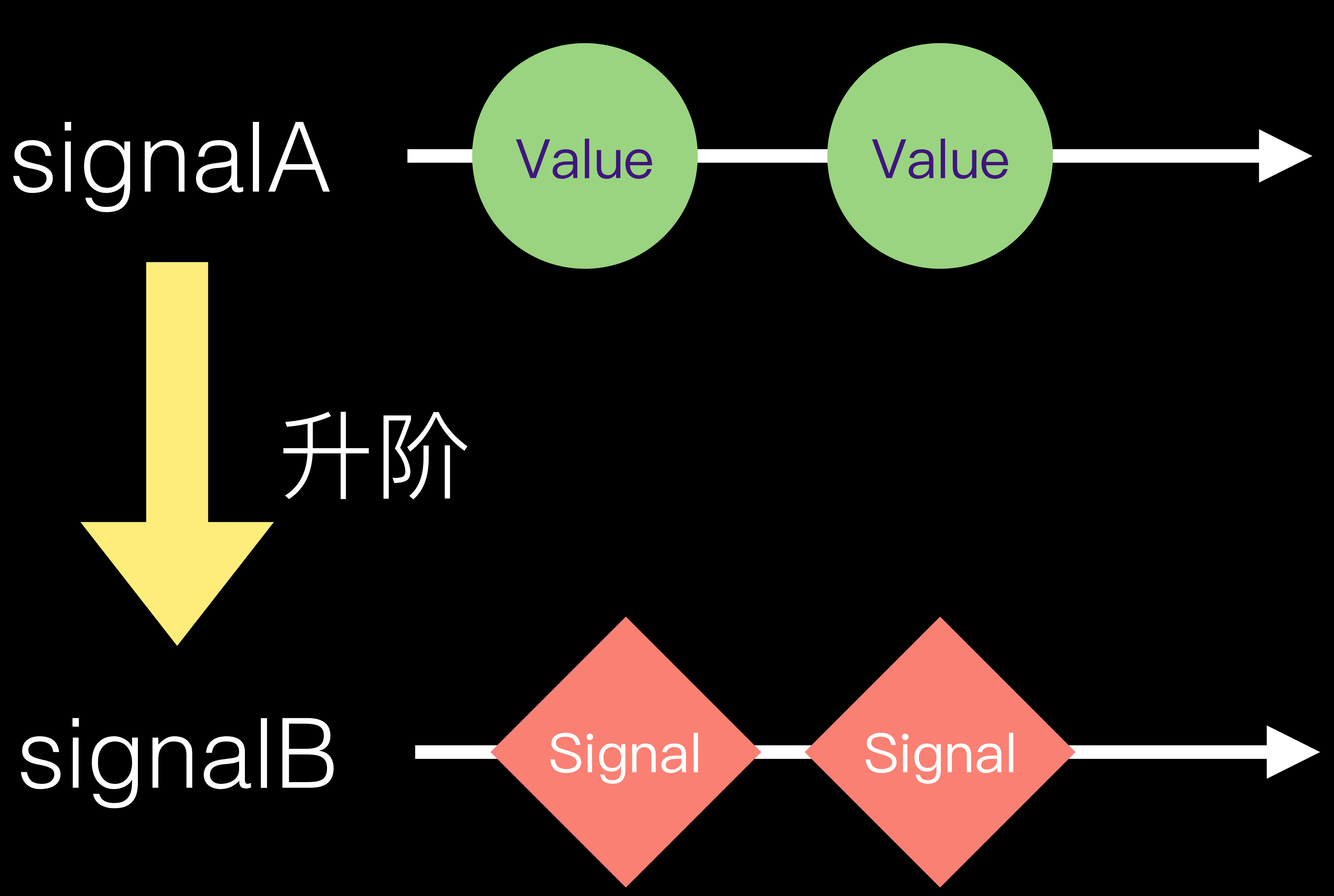
内容大纲

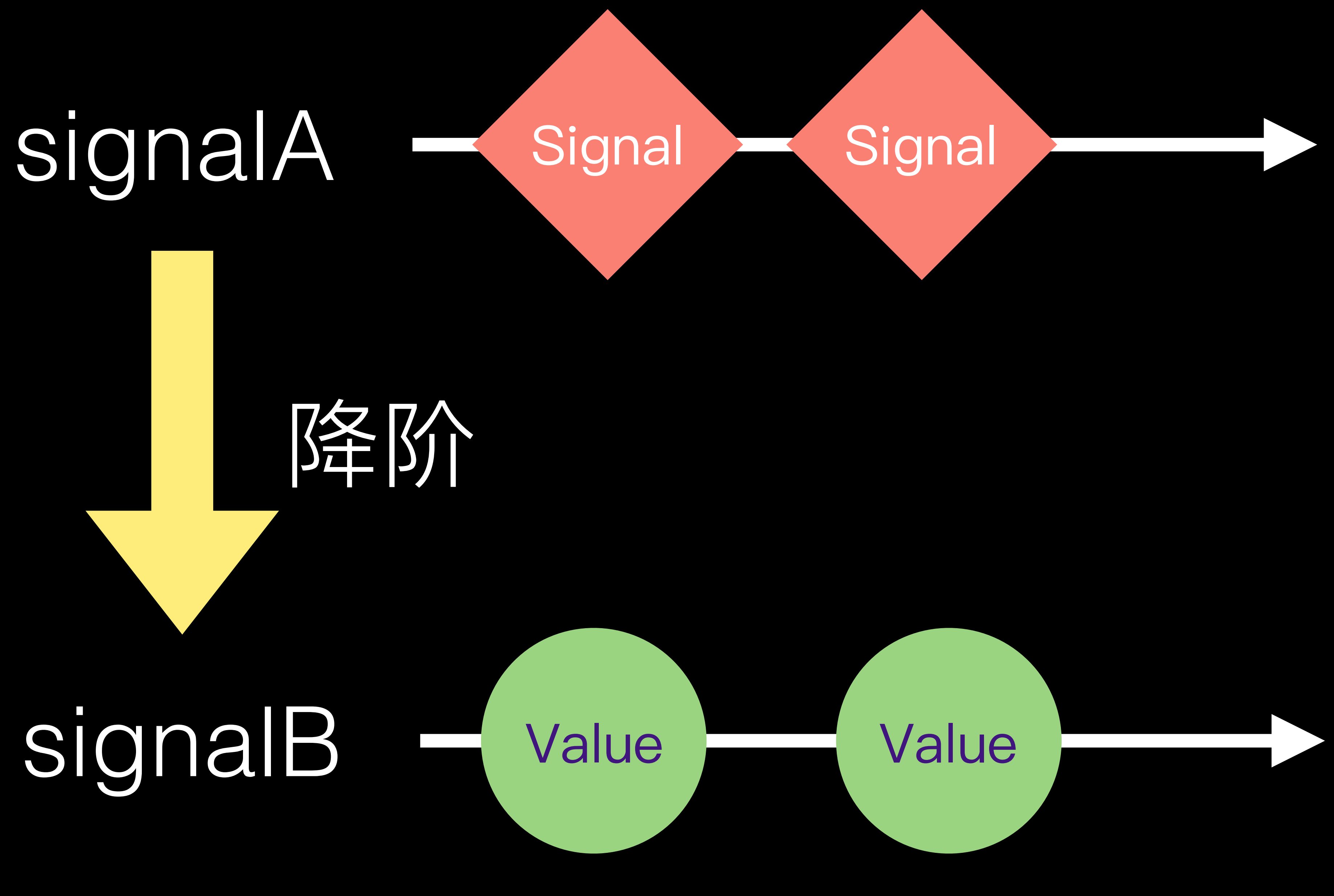
- 信号高阶操作

信号高阶操作

信号高阶操作

升阶和降阶





信号高阶操作

创造高阶信号

```
RACSignal *signal = [RACSignal return:@1];

RACSignal *signalHighOrder = [RACSignal return:signal];
RACSignal *anotherSignal = [signal map:^id(id value) {
    return [RACSignal return:value];
}];
```

信号的高阶操作

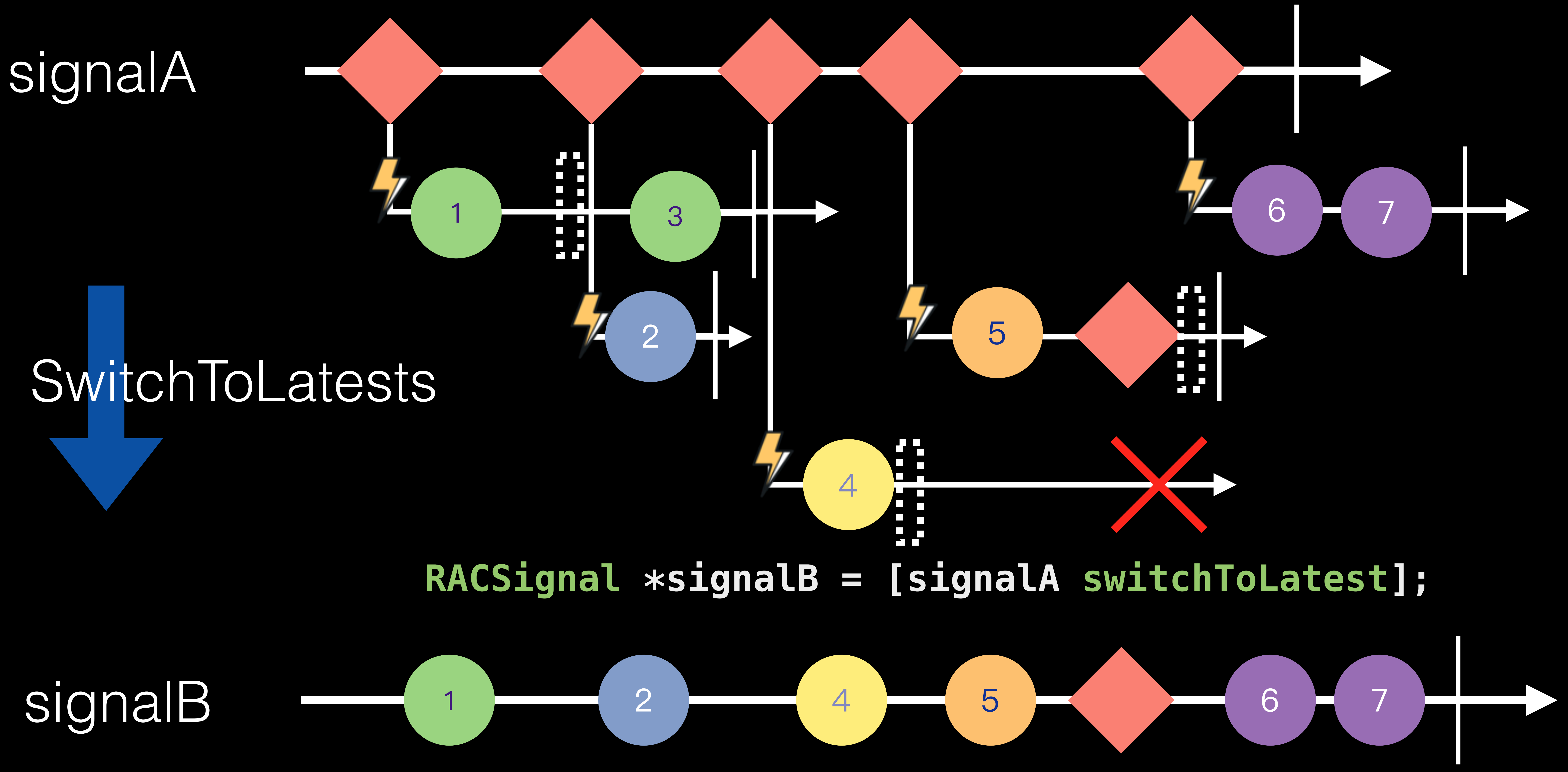
订阅高阶信号

```
RACSignal *signal = @[@1, @2, @3].rac_sequence.signal;
RACSignal *highOrderSignal = [signal map:^id(id value) {
    return [RACSignal return:value];
}];

[highOrderSignal subscribeNext:^(RACSignal *aSignal) {
    [aSignal subscribeNext:^(id x) {
        // get real value here.
    }];
}];
```

信号的高阶操作

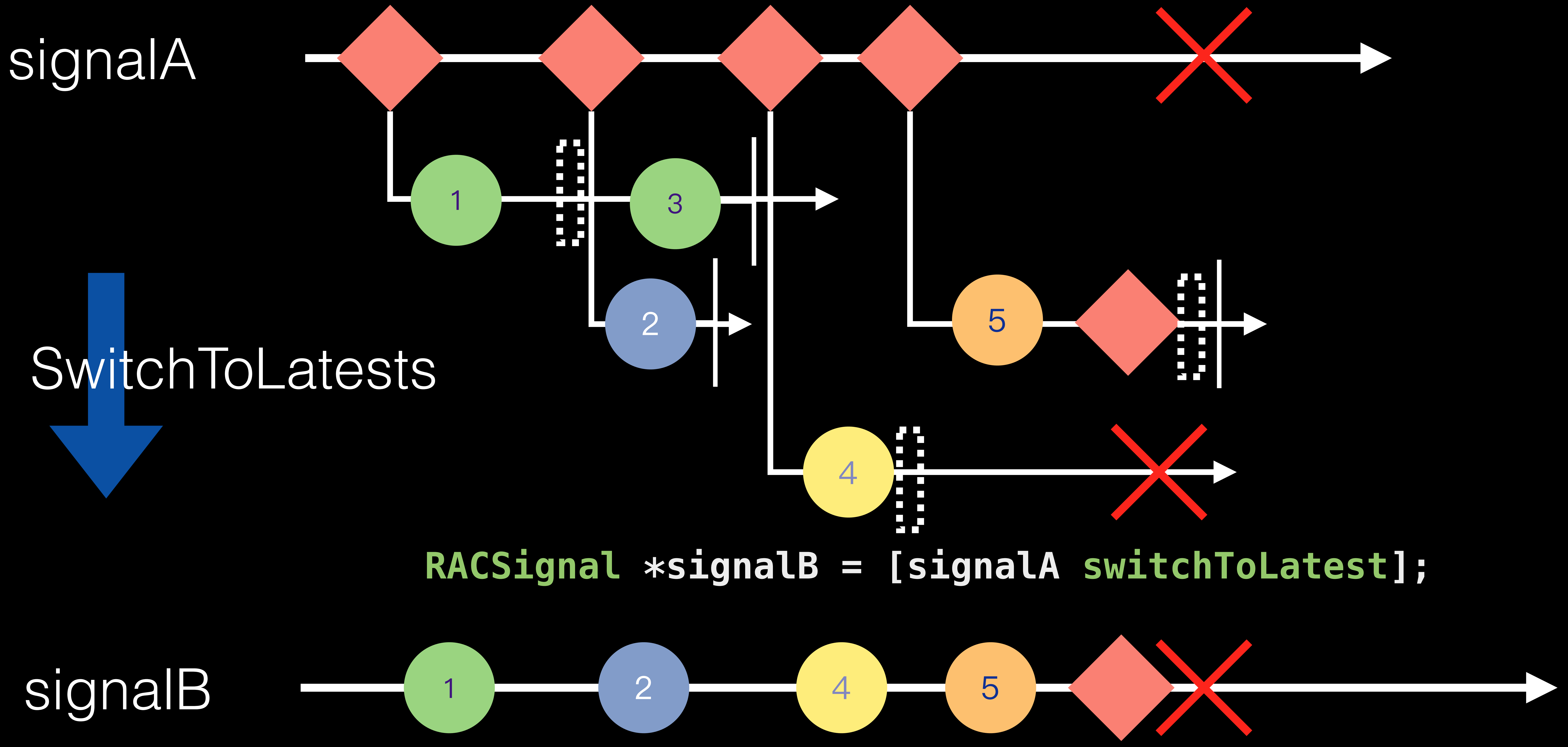
降阶操作——SwitchToLatests



```
RACSignal *signalB = [signalA switchToLatest];
```

信号的高阶操作

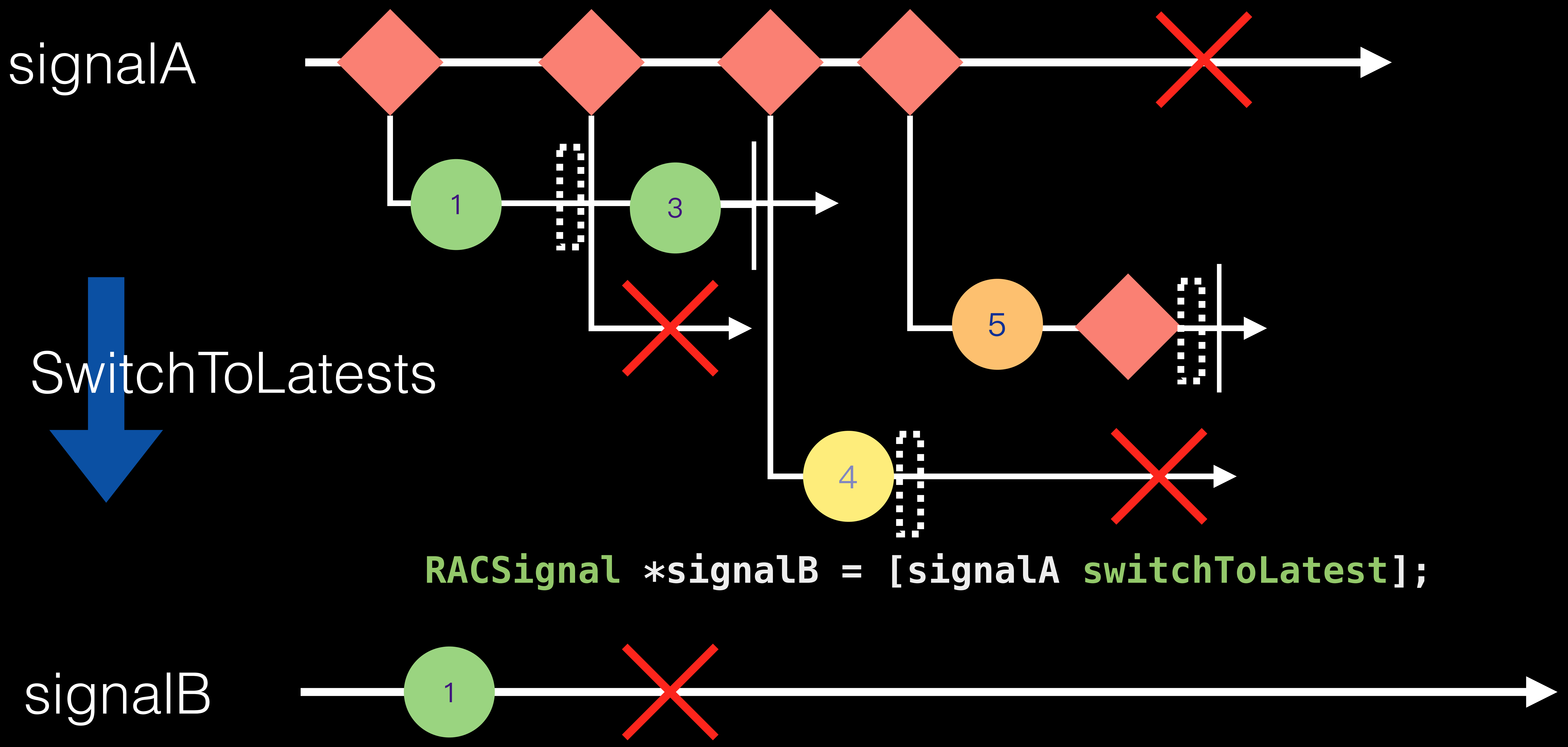
降阶操作——SwitchToLatests



```
RACSignal *signalB = [signalA switchToLatest];
```

信号的高阶操作

降阶操作——SwitchToLatests



```
RACSignal *signalB = [signalA switchToLatest];
```

信号的高阶操作

示例

```
RACSignal *autoRunButtonClickSignal = [self.autoRunBtn
    rac_signalForControlEvents:UIControlEventTouchUpInside];
RACSignal *oneStepButtonClickSignal = [self.oneStepBtn
    rac_signalForControlEvents:UIControlEventTouchUpInside];

RACSignal *idSignal = [RACSignal return:nil];
RACSignal *timerSignal = [RACSignal interval:1
                                 onScheduler:[RACScheduler
                                              mainThreadScheduler]];

autoRunButtonClickSignal = [autoRunButtonClickSignal mapReplace:idSignal];
oneStepButtonClickSignal = [oneStepButtonClickSignal mapReplace:timerSignal];

RACSignal *controlSignal = [autoRunButtonClickSignal merge:
                            oneStepButtonClickSignal];
controlSignal = [controlSignal switchToLatest];
```

信号的高阶操作

示例

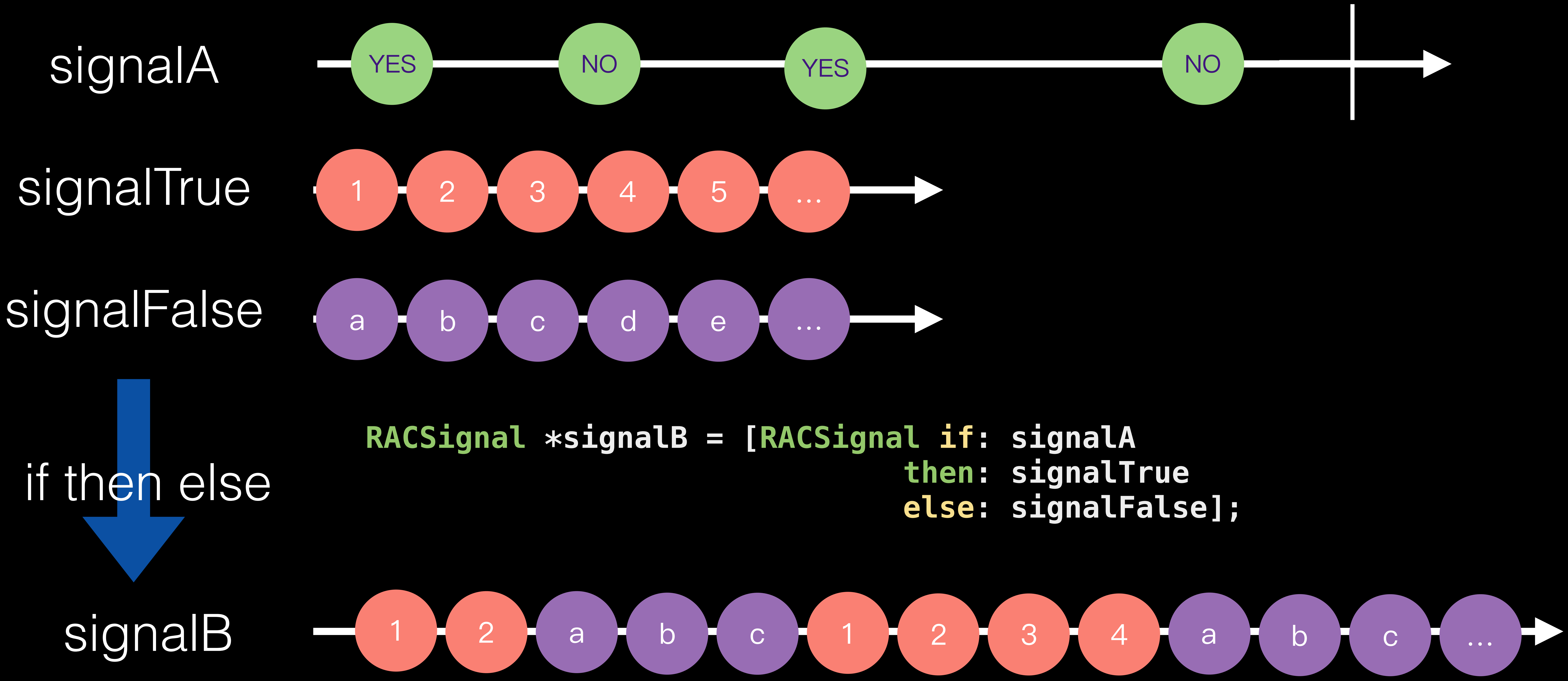
```
RACSignal *searchTextSignal = [self.searchTextField rac_textSignal];

RACSignal *requestSignals = [searchTextSignal map:^id(NSString *searchText) {
    NSString *urlString = [NSString stringWithFormat:
        @"http://xxxx.xxx.xxx/?q=%@", searchText];
    NSURL *url = [NSURL URLWithString:urlString];
    NSURLRequest *request = [NSURLRequest requestWithURL:url];
    return [NSURLConnection rac_sendAsynchronousRequest:request];
}];

requestSignals = [requestSignals switchToLatest];
```

信号的高阶操作

降阶操作——If/then/else



```
RACSignal *signalB = [RACSignal if: signalA
                              then: signalTrue
                              else: signalFalse];
```

信号的高阶操作

If/then/else 本质&类似操作

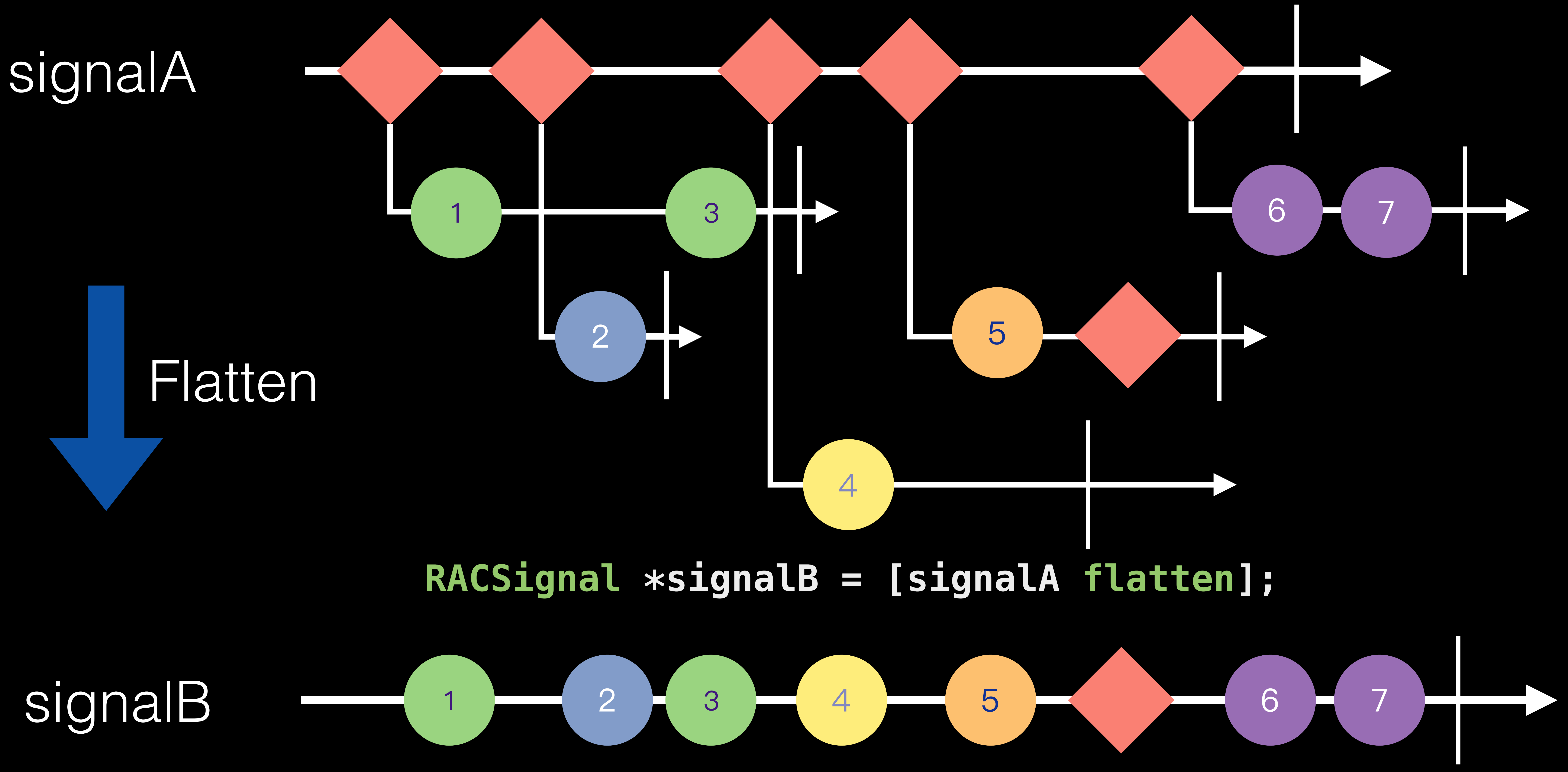
```
+ (RACSignal *)if:(RACSignal *)boolSignal
             then:(RACSignal *)trueSignal
             else:(RACSignal *)falseSignal {

  return [[boolSignal
   map:^(NSNumber *value) {
    return (value.boolValue ? trueSignal : falseSignal);
   }]
   switchToLatest];
}
```

```
+ (RACSignal *)switch:(RACSignal *)signal
                cases:(NSDictionary *)cases
              default:(RACSignal *)defaultSignal;
```

信号的高阶操作

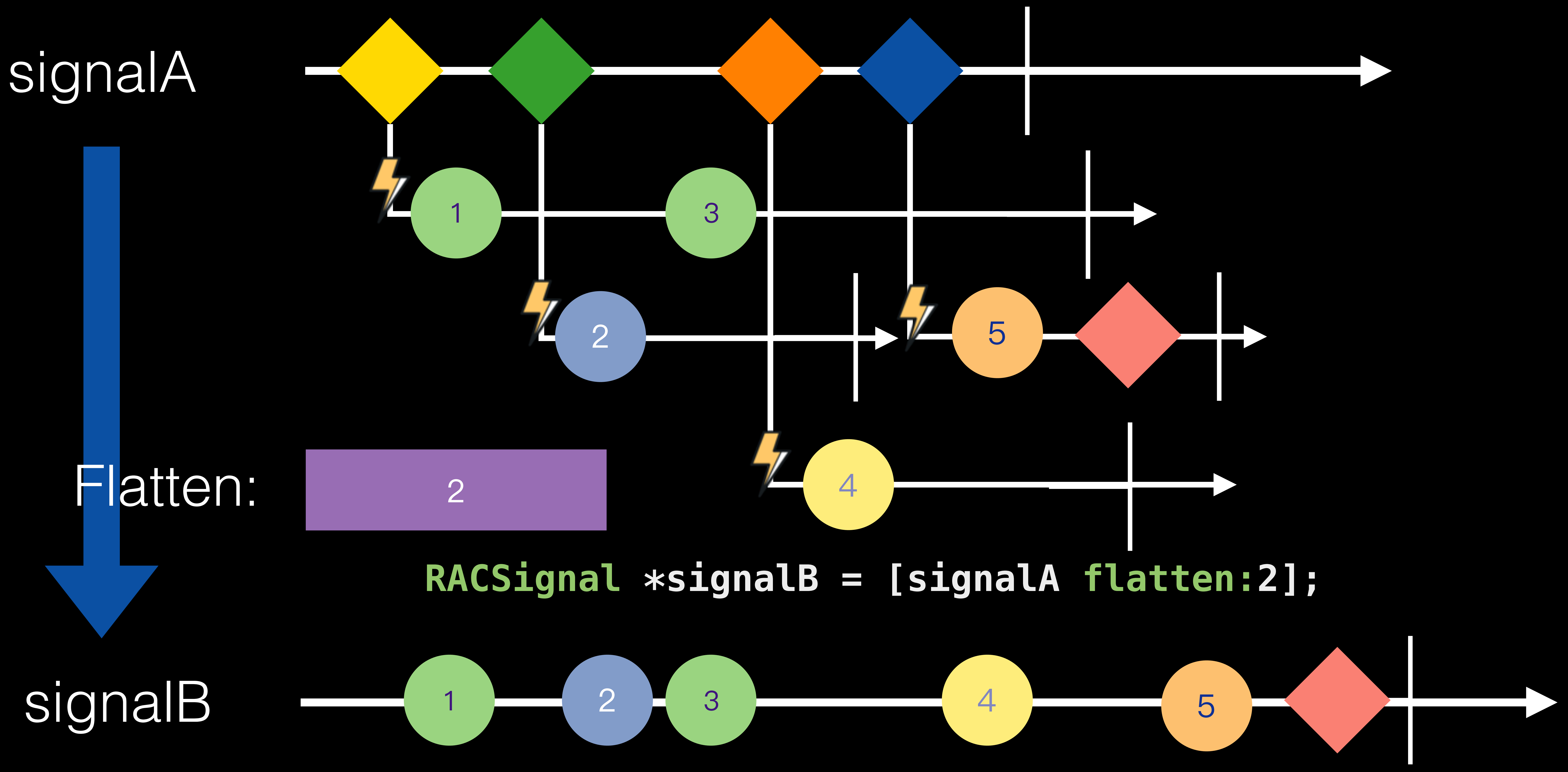
降阶操作——Flatten



```
RACSignal *signalB = [signalA flatten];
```

信号的高阶操作

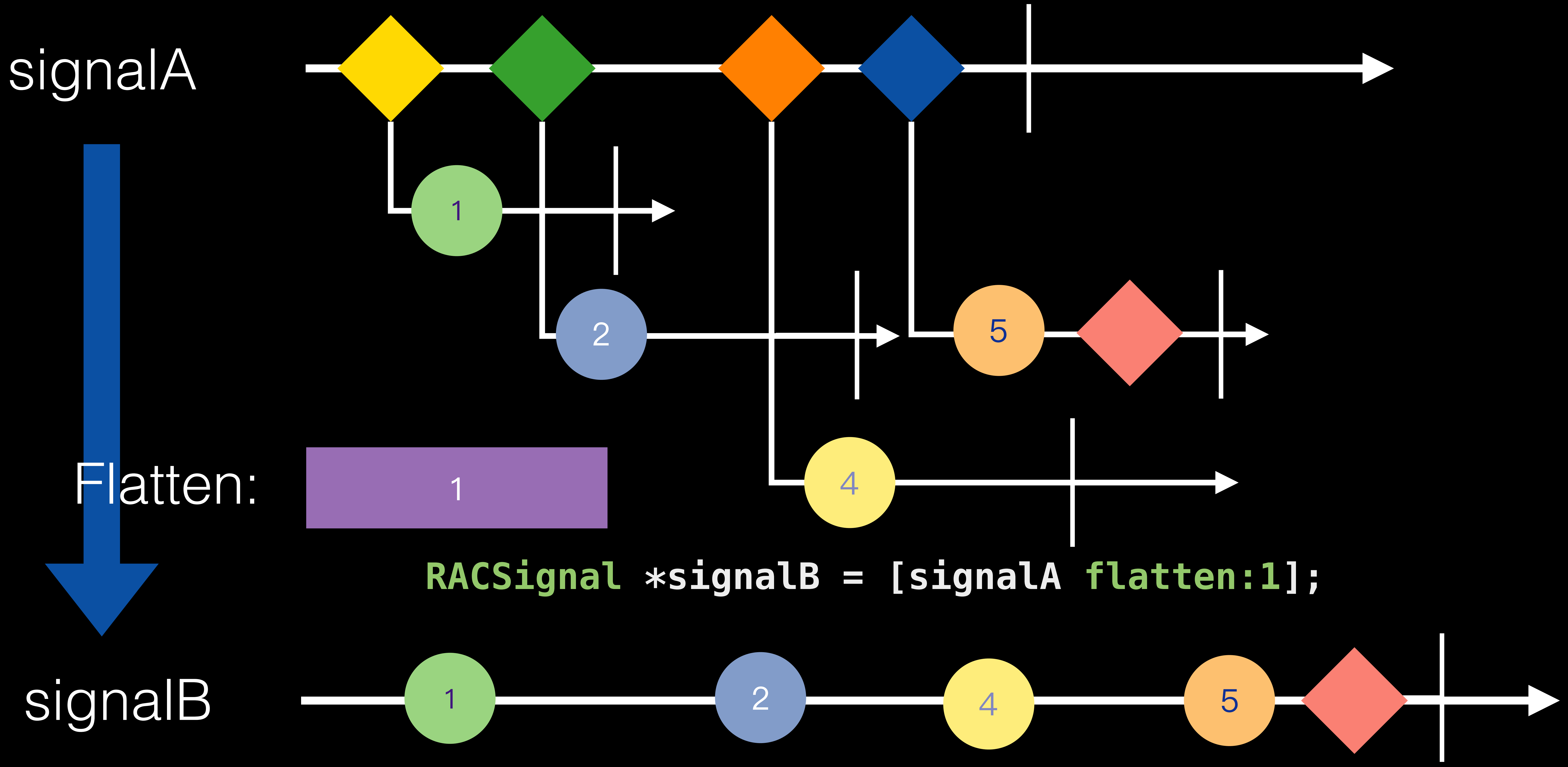
降阶操作——Flatten



```
RACSignal *signalB = [signalA flatten:2];
```

信号的高阶操作

降阶操作——Flatten



```
RACSignal *signalB = [signalA flatten:1];
```

信号的高阶操作

Flatten:1的思考

- 返回a信号、b信号的高阶信号flatten:1
- a信号 concat: b信号
- - (RACSignal *)concat;
- 思考concat与1秒延时信号

```
RACSignal *signal = @[@1, @3, @7,
                      @9, @8].rac_sequence.signal;

RACSignal *timerSignal = [[signal map:^id(id value) {
    return [[RACSignal return:value] delay:1];
}] concat];
```

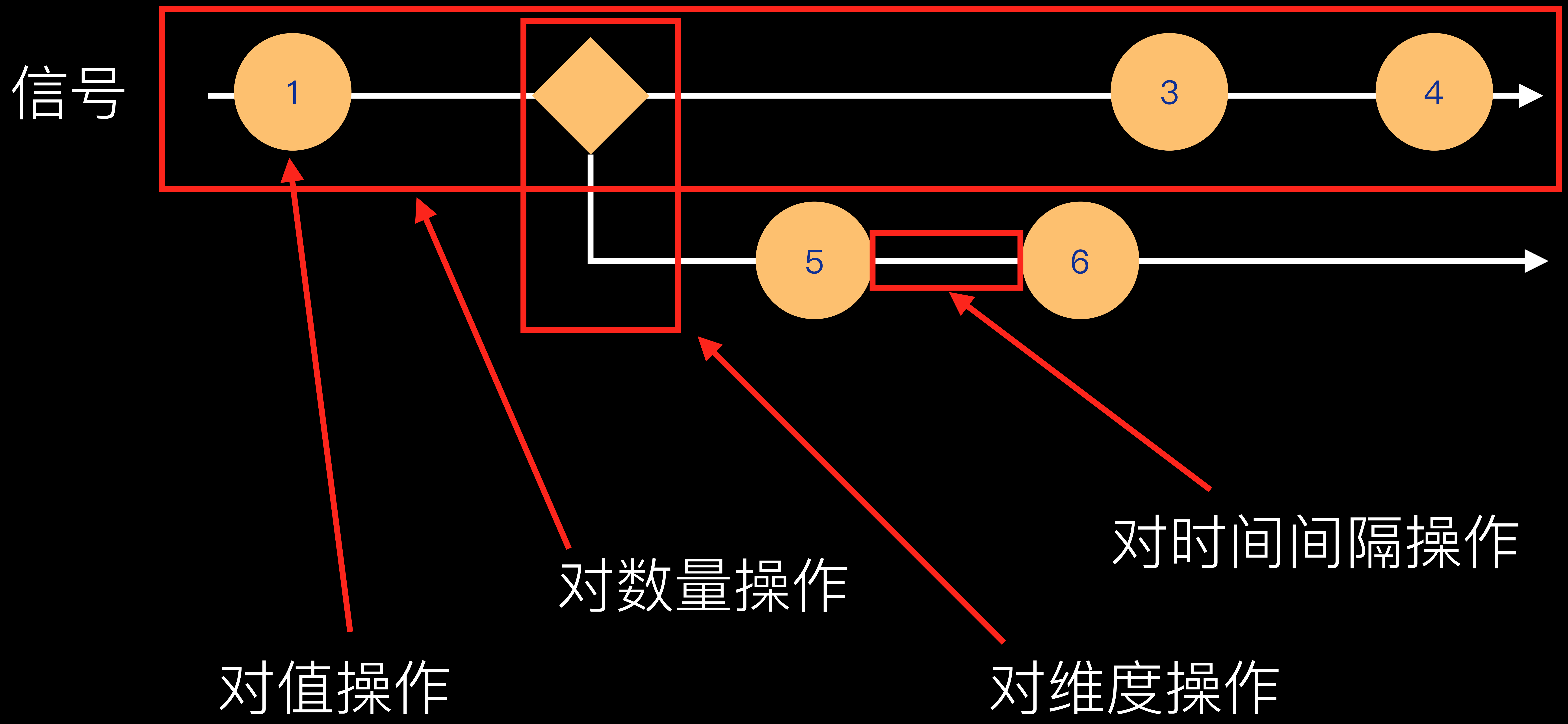
信号的高阶操作

降阶操作总结

- SwitchToLatests
- Flatten（Merge）
- Concat（Flatten:1）

信号的高阶操作

思考



思考如下操作

-改变某一个值的个数

-把值/错误/停止互换

-缩短/拉长时间间隔

信号的高阶操作

思考

改变某个值的个数

```
RACSignal *signal = @[@1, @2, @3].rac_sequence.signal;

RACSignal *mappedSignal = [[signal map:^id(NSNumber *value) {
    return [[[RACSignal return:value] repeat]
            take:value.integerValue];
}] flatten];
```

信号的高阶操作

思考

将一个值改为一个错误

```
RACSignal *signal = @[@1, @2, @3, @0].rac_sequence.signal;

RACSignal *mappedSignal = [[signal map:^id(NSNumber *value) {
    if (value.integerValue == 0) {
        return [RACSignal error:[NSError errorWithDomain:@"0"
                                                    code:0
                                                userInfo:nil]];
    } else {
        return [RACSignal return:value];
    }
}] flatten];
```

信号的高阶操作

思考

改变值的时间间隔

```
RACSignal *signal = @[@"♪5", @"♬1", @"♬2", @"♬3", @"♩4"]
                        .rac_sequence
                        .signal;
NSDictionary *toneLengthMap = @{@"♩": @0.5,
                                @"♪": @0.25,
                                @"♬": @0.125};
RACSignal *mappedSignal = [[signal map:^id(NSString *value) {
    NSString *tone = [value substringFromIndex:1];
    NSString *length = [value substringToIndex:1];
    NSNumber *toneValue = @(tone.integerValue);
    NSNumber *toneLength = toneLengthMap[length];
    return [[RACSignal return:toneValue]
            concat:[[RACSignal empty]
                    delay: toneLength.doubleValue]];
}] concat];
```

信号的高阶操作

思考

Map → Flatten

FlattenMap

信号的高阶操作

FlattenMap的重要性

- 可以用FlattenMap实现很多的信号转换
- 支持串行异步操作（类似Promise）
- 满足Monad部分定义（bind和return才完全满足）

信号的高阶操作

FlattenMap的重要性

```
RACSignal *flatten = [signal flattenMap:^RACStream *(RACSignal *value) {
    return value;
}];

RACSignal *map = [signal flattenMap:^RACStream *(id value) {
    id anotherValue = value; // map here!
    return [RACSignal return: anotherValue];
}];

RACSignal *filter = [signal flattenMap:^RACStream *(id value) {
    BOOL filter = (value == nil); // filter here!
    return filter ? [RACSignal empty] : [RACSignal return:value];
}];
```

信号的高阶操作

FlattenMap的重要性

```
RACSignal *signal = [RACSignal return:@"http://xx.com/a"];

RACSignal *getSignal = [signal flattenMap: ^RACStream *(NSString *url) {
    NSURLRequest *request = [NSURLRequest requestWithURL:[NSURL URLWithString:url]];
    return [NSURLConnection rac_sendAsynchronousRequest:request];
}];

RACSignal *jsonSignal = [getSignal flattenMap: ^RACStream *(NSData *data) {
    NSError *error = nil;
    id result = [NSJSONSerialization JSONObjectWithData:data options:0 error:&error];
    return error == nil ? [RACSignal return: result] : [RACSignal error: error];
}];

RACSignal *getItemSignal = [jsonSignal flattenMap: ^RACStream *(NSDictionary *value) {
    if (![value isKindOfClass:[NSDictionary class]] || value[@"data.url"] == nil) {
        return [RACSignal error:someError];
    }
    NSURLRequest *anotherRequest = [NSURLRequest requestWithURL:
                                    [NSURL URLWithString:value[@"data.url"]]];
    return [NSURLConnection rac_sendAsynchronousRequest:anotherRequest];
}];
```

信号的高阶操作

FlattenMap与monad、bind

- Functor、Applicative、Monad概念
- FlattenMap符合Monad的bind定义，但是……
    - 无法实现takeUntil:操作
    - 无法做副作用操作例如take:的计数

```
- (instancetype)flattenMap:(RACStream * (^)(id value))block;
- (instancetype)bind:(RACStreamBindBlock (^)(void))block;
typedef RACStream * (^RACStreamBindBlock)(id value, BOOL *stop);
```

信号的高阶操作

bind妙用

```
- (RACSignal *)take:(NSUInteger)count {
  if (count == 0) return [RACSignal empty];

  return [self bind:^{
   __block NSUInteger taken = 0;

   return ^ id (id value, BOOL *stop) {
     if (taken < count) {
       ++taken;
       if (taken == count) *stop = YES;
       return [class return:value];
     } else {
       return nil;
     }
   };
  }];
}
```

信号的高阶操作

bind简单实现和问题

```
- (RACSignal *)bind:(RACStreamBindBlock (^)(void))block;
{
    return [RACSignal createSignal:^RACDisposable *(id<RACSubscriber> subscriber) {
        RACStreamBindBlock bindBlock = block();

        [self subscribeNext:^(id x) {
            BOOL stop = NO;
            RACSignal *signal = (RACSignal *)bindBlock(x, &stop);
            if (signal == nil || stop) { [subscriber sendCompleted];
            } else {
                [signal subscribeNext:^(id x) { [subscriber sendNext:x];
                } error:^(NSError *error) { [subscriber sendError:error];
                } completed:^{ }];
            }
        } error:^(NSError *error) { [subscriber sendError:error];
        } completed:^{ [subscriber sendCompleted]; }];
        return nil;
    }];
}
```

信号的高阶操作

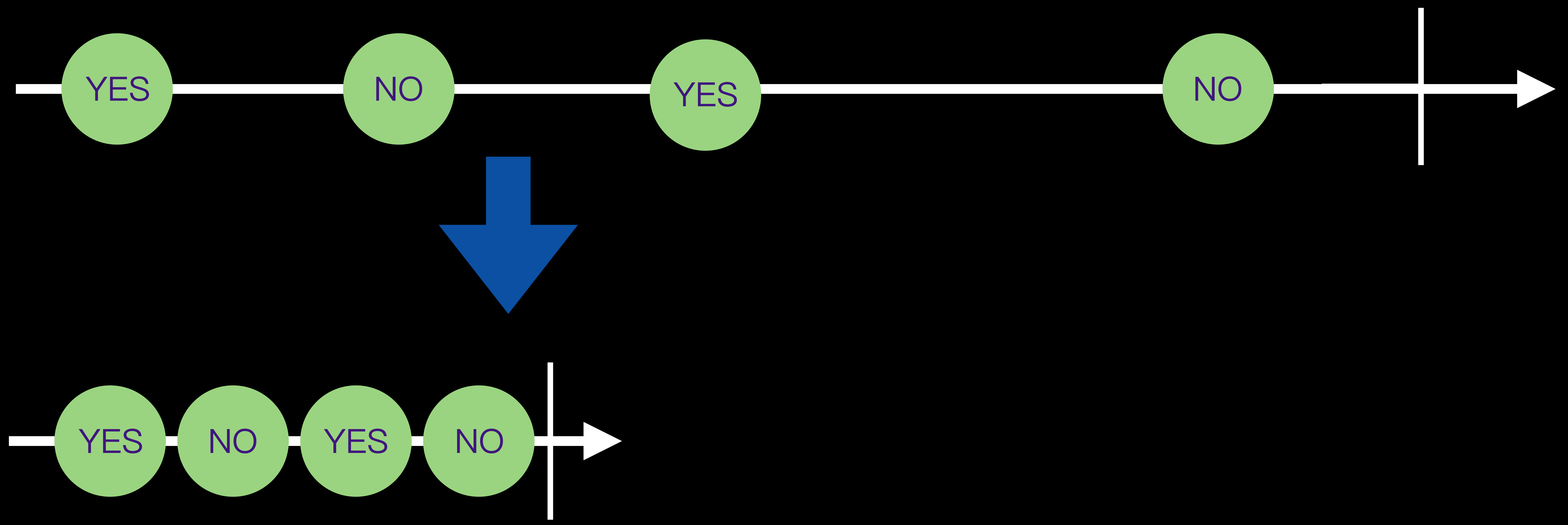
有用的高阶操作

```
- (RACSignal *)try:(BOOL (^)(id value, NSError **errorPtr))tryBlock;
- (RACSignal *)tryMap:(id (^)(id value, NSError **errorPtr))mapBlock;
- (RACSignal *)catch:(RACSignal * (^)(NSError *error))catchBlock;
- (RACSignal *)catchTo:(RACSignal *)signal;
- (RACSignal *)timeout:(NSTimeInterval)interval
           onScheduler:(RACScheduler *)scheduler;
```

信号的高阶操作

扩展问题

我可以缩短值之间的间隔吗?



信号的高阶操作

信号更多的操作

to be continue…