

std::false\_type

std::conditional\_t< bool(B 1::value), B 1, disjunction< B n... > >

Catch::Detail::disjunction< B 1, B n... >

```
graph BT; A[Catch::Detail::disjunction< B 1, B n... >] --> B[std::false_type]; A --> C[std::conditional_t< bool(B 1::value), B 1, disjunction< B n... > >];
```

The diagram illustrates a relationship between three C++ type definitions. At the bottom is a box for `Catch::Detail::disjunction< B 1, B n... >`. Two arrows originate from its top edge: one points to the `std::false_type` box on the left, and the other points to the `std::conditional_t< bool(B 1::value), B 1, disjunction< B n... > >` box on the right. This suggests that `Catch::Detail::disjunction` is a base or specialization for both `std::false_type` and the `std::conditional_t` expression.