

`std::false_type`

`std::conditional_t
< bool(B1::value), B1,
disjunction< Bn... > >`

`Catch::Detail::disjunction
< B1, Bn... >`

```
graph BT; A[Catch::Detail::disjunction< B1, Bn... >] --> B[std::false_type]; A --> C[std::conditional_t< bool(B1::value), B1, disjunction< Bn... > >];
```

The diagram illustrates a relationship between three C++ type aliases. At the bottom is a gray box containing the code `Catch::Detail::disjunction< B1, Bn... >`. Two blue arrows originate from this box: one points to a white box at the top left containing `std::false_type`, and the other points to a white box at the top right containing `std::conditional_t< bool(B1::value), B1, disjunction< Bn... > >`. This suggests that `Catch::Detail::disjunction` is a specialization or alias for `std::false_type` in one context and for `std::conditional_t` in another.