59

https://godbolt.org/z/Gbh75aadj

```
struct node
{
    node* prev;
    node* next;
};

static_assert(! is_send_v<node>);
```

```cpp
struct type
{
    type()
    {
        auto n = std::make_shared<node>();

        [[maybe_unused]] auto this_capturing = [this] { run(); };
        static_assert(! is_send_v<decltype(this_capturing)>);

        [[maybe_unused]] auto this_n_capturing = [this, n] { run(); };
        static_assert(! is_send_v<decltype(this_n_capturing)>);

        [[maybe_unused]] auto n_ref_capturing = [&n] {};
        static_assert(! is_send_v<decltype(n_ref_capturing)>);

        [[maybe_unused]] auto n_val_capturing = [n] {};
        static_assert(! is_send_v<decltype(n_val_capturing)>);
    }

    void run() {}
};
```

```
[[maybe_unused]] auto non_capturing = [] (int) {};
static_assert(is_send_v<decltype(non_capturing)>);

int i = 0;
[[maybe_unused]] auto val_capturing = [i] (int) {};
static_assert(is_send_v<decltype(val_capturing)>);

[[maybe_unused]] auto ref_capturing = [&i] (int) {};
static_assert(! is_send_v<decltype(ref_capturing)>);
```

```cpp
struct node
{
    node* prev;
    node* next;
};

static_assert(! is_send_v<node>);
```

```cpp
[[maybe_unused]] auto non_capturing = [] (int) {};
static_assert(is_send_v<decltype(non_capturing)>);

int i = 0;
[[maybe_unused]] auto val_capturing = [i] (int) {};
static_assert(is_send_v<decltype(val_capturing)>);

[[maybe_unused]] auto ref_capturing = [&i] (int) {};
static_assert(! is_send_v<decltype(ref_capturing)>);
```

```cpp
struct type
{
    type()
    {
        auto n = std::make_shared<node>();

        [[maybe_unused]] auto this_capturing = [this] { run(); };
        static_assert(! is_send_v<decltype(this_capturing)>);

        [[maybe_unused]] auto this_n_capturing = [this, n] { run(); };
        static_assert(! is_send_v<decltype(this_n_capturing)>);

        [[maybe_unused]] auto n_ref_capturing = [&n] {};
        static_assert(! is_send_v<decltype(n_ref_capturing)>);

        [[maybe_unused]] auto n_val_capturing = [n] {};
        static_assert(! is_send_v<decltype(n_val_capturing)>);
    }

    void run() {}
};
```

Problems: Summary

- ~~Nested pointers~~
- ~~`this` pointers~~
- Global pointers
- Leaked pointers