A value with a custom `Drop` implementation may be dropped during const-eval.

Erroneous code example:

```
enum DropType {
    A,
}

impl Drop for DropType {
    fn drop(&mut self) {}
}

struct Foo {
    field1: DropType,
}

static FOO: Foo = Foo { field1: (DropType::A, DropType::A).1 }; // error!
```

The problem here is that if the given type or one of its fields implements the `Drop` trait, this `Drop` implementation cannot be called within a const context since it may run arbitrary, non-const-checked code. To prevent this issue, ensure all values with a custom `Drop` implementation escape the initializer.

```
enum DropType {
    A,
}

impl Drop for DropType {
    fn drop(&mut self) {}
}

struct Foo {
    field1: DropType,
}

static FOO: Foo = Foo { field1: DropType::A }; // We initialize all fields
                                               // by hand.
```