A pattern for a struct fails to specify a sub-pattern for every one of the struct's fields.

Erroneous code example:

```
struct Dog {
    name: String,
    age: u32,
}

let d = Dog { name: "Rusty".to_string(), age: 8 };

// This is incorrect.
match d {
    Dog { age: x } => {}
}
```

To fix this error, ensure that each field from the struct's definition is mentioned in the pattern, or use `..` to ignore unwanted fields. Example:

```
struct Dog {
    name: String,
    age: u32,
}

let d = Dog { name: "Rusty".to_string(), age: 8 };

match d {
    Dog { name: ref n, age: x } => {}
}

// This is also correct (ignore unused fields).
match d {
    Dog { age: x, .. } => {}
}
```