Writing LLVM Passes (2)
Using Alive2 to Check Tests

SWPP

April 8th

Juneyoung Lee

What Did We Do Last Week?

- Implement a simple constant folding pass
- Q: What do we need to write more complex passes?
- I’ll introduce two examples:
    1. instmatch.cpp: How to use Matchers?
    2. fillundef.cpp: How to smartly transform the program?

When You're Using STL...

- Be careful about:

A. Out-of-bounds access

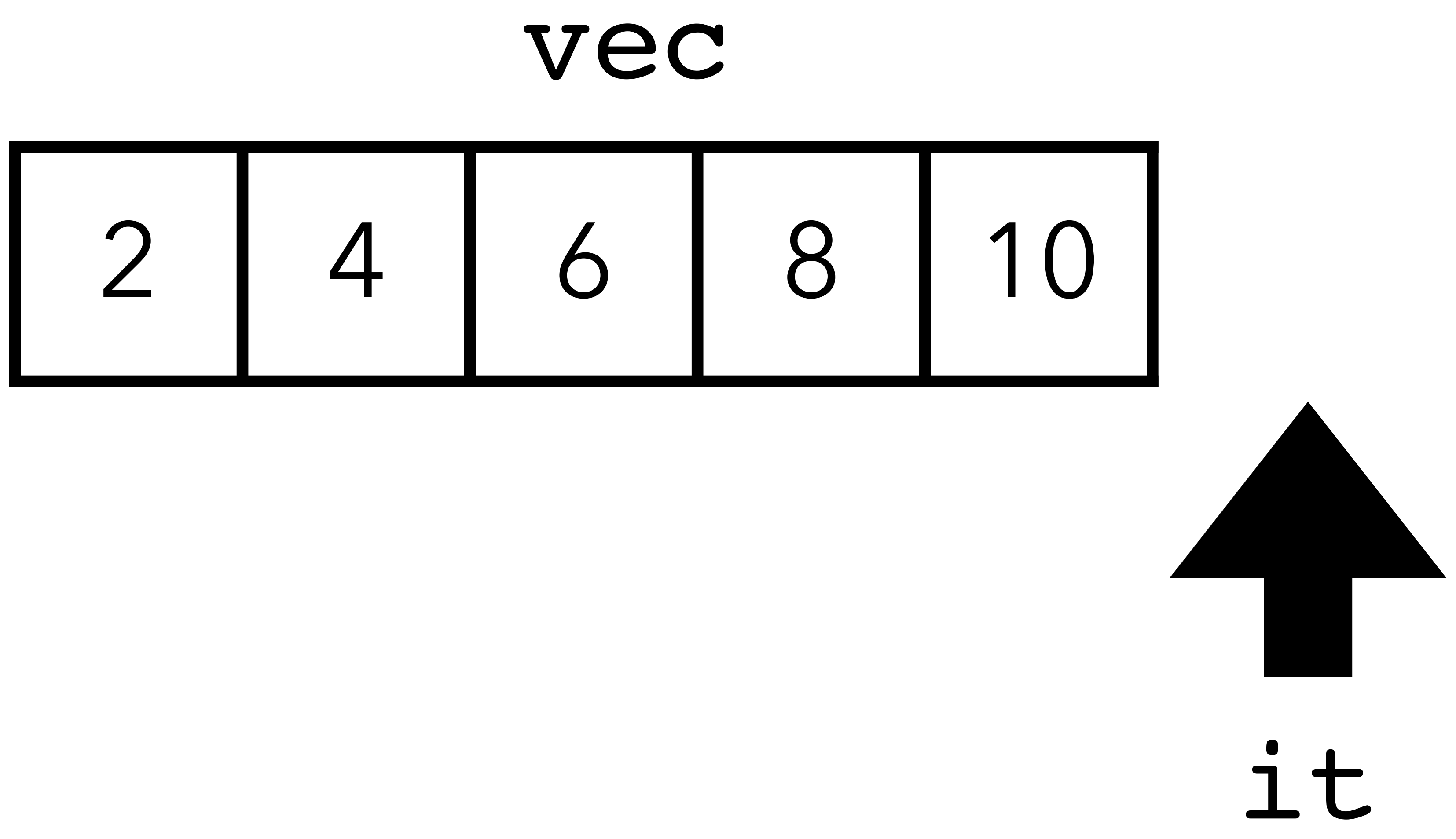
B. Use-after-free

C. Passing a container by value

A. Out-of-bounds access

```
auto it = std::find(vec.begin(), vec.end(), 5);
*it = 6;
```

```
auto it = std::find(vec.begin(), vec.end(), 5);
if (it != vec.end())
  *it = 6;
```



B. Use-after-free

```
auto it = std::find(vec.begin(), vec.end(), 5);
if (it != vec.end())
  vec.erase(it);

cout << “after 5: “;
for (; it != vec.end(); ++it)
  cout << *it << “ ”;
```

‘it’ is invalidated!

```
auto it = std::find(vec.begin(), vec.end(), 5);
if (it != vec.end())
  it = vec.erase(it);

cout << “after 5: “;
for (; it != vec.end(); ++it)
  cout << *it << “ ”;
```

C. Passing a container by value

```
vector<int> v = {1, 2, 3, 4};
pushBackFive(v);

void pushBackFive(vector<int> v) {
  v.push_back(5);
}
```

```
vector<int> v = {1, 2, 3, 4};
pushBackFive(v);

void pushBackFive(vector<int> &v) {
  v.push_back(5);
}
```

```
vector<int> v = {1, 2, 3, 4};
printAll(v);

void printAll(vector<int> v) {
  for (int i : v)
    cout << i << endl;
}
```

```
vector<int> v = {1, 2, 3, 4};
printAll(v);

void printAll(const vector<int> &v) {
  for (int i : v)
    cout << i << endl;
}
```

Using Alive2 to Check Tests

- We learned how to use FileCheck last week
- However, it does syntactic checks only. :(
- Let's see how to use Alive2!
  - Web: https://alive2.llvm.org/ce/z/vKMMgb , https://alive2.llvm.org/ce/z/jLdff9
  - Terminal: alive-tv <src.ll> <tgt.ll>

How to Build?

1. Compile new LLVM using llvm-alive2.json
   - Please git-pull llvmscript
   - Use examples/llvm-alive2.json to clone & build
   - It will take shorter time (hopefully...)
2. Install z3 using apt (Ubuntu) / brew (Mac)
3. git clone https://github.com/AliveToolkit/alive2.git
4. mkdir build ; cd build;
5. cmake -GNinja -DCMAKE_PREFIX_PATH=my-llvm-alive2-releaseassert/build
   -DBUILD_TV=1 -DCMAKE_BUILD_TYPE=Release ..