

Advanced Compiler - Homework 2 Report

許峻源
111062649

1. How to run my program ?

我稍微修改makefile, 把所有testcase的檔名, 不包含檔案類別, 放在變數 ll:= ..., 然後要被執行的testcase 紀錄在target, 每次要改變要執行的case時, 只要改target 那項就好。

執行步驟:

1. make
會把所有的.ll file 和 .so 都建立好
2. make run
執行target 中的testcase

```
LLVM_CONFIG=../llvm_build/bin/llvm-config

CXX=`$(LLVM_CONFIG) --bindir`/clang
CXXFLAGS=`$(LLVM_CONFIG) --cppflags` -fPIC -fno-rtti
LDFLAGS=`$(LLVM_CONFIG) --ldflags`
IRFLAGS=-Xclang -disable-00-optnone -fno-discard-value-names -S -emit-llvm
OPT=`$(LLVM_CONFIG) --bindir`/opt

ll:=icpp icpp2 icpp3 foo # the testcase list. If there is any new testcase, just put the name of the new testcase here

target:=foo.ll # the one sent to run. modify the target, when wanting to change to another testcase

.PHONY: all test run clean
all: hw2.so test

test: ${ll} #make all testcases
${ll}: *.c
    $(CXX) $(IRFLAGS) -o $@.ll $@.c

hw2.so: hw2.cpp
    $(CXX) -shared -o $@ $(CXXFLAGS) $(LDFLAGS)

run: ${target} hw2.so #run the target .ll file with hw2.so
    $(OPT) -disable-output -load-pass-plugin=./hw2.so -passes=hw2 $<

clean:
    rm -f *.o *.ll *.so
```

2. Output

- icpp.c

```
S1:-----
TREF:{ }
TGEN:{ p }
DEP:{
}
TDEF:{(p, S1)}
TEQUIV:{ (*p,x)}

S2:-----
TREF:{ }
TGEN:{ pp }
DEP:{
}
TDEF:{(p, S1), (pp, S2)}
TEQUIV:{ (*p,x), (*pp,p), (**pp,x)}

S3:-----
TREF:{ pp }
TGEN:{ p,*pp }
DEP:{
    pp: S2--->S3
    p: S1-0->S3
}
TDEF:{(p, S3), (pp, S2), (*pp, S3)}
TEQUIV:{ (*p,y), (*pp,p), (**pp,y)}

S4:-----
TREF:{ p,*pp }
TGEN:{ y,*p }
DEP:{
    p: S3--->S4
    *pp: S3--->S4
}
TDEF:{(p, S3), (pp, S2), (*pp, S3), (y, S4), (*p, S4)}
TEQUIV:{ (*p,y), (*pp,p), (**pp,y)}
```

- icpp2.c

```
S1:-----
TREF:{ }
TGEN:{ p }
DEP:{
}
TDEF:{(p, S1)}
TEQUIV:{ (*p,x)}

S2:-----
TREF:{ }
TGEN:{ pp }
DEP:{
}
TDEF:{(p, S1), (pp, S2)}
TEQUIV:{ (*p,x), (*pp,p), (**pp,x)}

S3:-----
TREF:{ pp }
TGEN:{ p,*pp }
DEP:{
    pp: S2--->S3
    p: S1-0->S3
}
TDEF:{(p, S3), (pp, S2), (*pp, S3)}
TEQUIV:{ (*p,y), (*pp,p), (**pp,y)}

S4:-----
TREF:{ p,*pp }
TGEN:{ y,*p }
DEP:{
    p: S3--->S4
    *pp: S3--->S4
}
TDEF:{(p, S3), (pp, S2), (*pp, S3), (y, S4), (*p, S4)}
TEQUIV:{ (*p,y), (*pp,p), (**pp,y)}
```

```

S5:-----
TREF:{ y,p,pp,*p,*pp,**pp }
TGEN:{ y,**pp }
DEP:{
    p: S3--->S5
    pp: S2--->S5
    *pp: S3--->S5
    y: S4--->S5
    *p: S4--->S5
    y: S4-0->S5
}
TDEF:{(p, S3), (pp, S2), (*pp, S3), (y, S5), (*p, S4), (**pp, S5)}
TEQUIV:{ (*p,y), (*pp,p), (**pp,y)}

```

● icpp3.c

```

S1:-----
TREF:{ }
TGEN:{ p }
DEP:{
}
TDEF:{(p, S1)}
TEQUIV:{ (*p,x)}

S2:-----
TREF:{ }
TGEN:{ pp }
DEP:{
}
TDEF:{(p, S1), (pp, S2)}
TEQUIV:{ (*p,x), (*pp,p), (**pp,x)}

S3:-----
TREF:{ pp }
TGEN:{ p,*pp }
DEP:{
    pp: S2--->S3
    p: S1-0->S3
}
TDEF:{(p, S3), (pp, S2), (*pp, S3)}
TEQUIV:{ (*p,y), (*pp,p), (**pp,y)}

```

```

S4:-----
TREF:{ p,*pp }
TGEN:{ y,*p }
DEP:{
    p: S3--->S4
    *pp: S3--->S4
}
TDEF:{(p, S3), (pp, S2), (*pp, S3), (y, S4), (*p, S4)}
TEQUIV:{ (*p,y), (*pp,p), (**pp,y)}

S5:-----
TREF:{ p,pp,*pp }
TGEN:{ y,**pp }
DEP:{
    p: S3--->S5
    pp: S2--->S5
    *pp: S3--->S5
    y: S4-0->S5
}
TDEF:{(p, S3), (pp, S2), (*pp, S3), (y, S5), (*p, S4), (**pp, S5)}
TEQUIV:{ (*p,y), (*pp,p), (**pp,y)}

```

- foo.c

```
S1:-----
TREF:{ b,c }
TGEN:{ a }
DEP:{
}
TDEF:{(a, S1)}
TEQUIV:{}

S2:-----
TREF:{ }
TGEN:{ p }
DEP:{
}
TDEF:{(a, S1), (p, S2)}
TEQUIV:{ (*p,y)}

S3:-----
TREF:{ b,c }
TGEN:{ d }
DEP:{
}
TDEF:{(a, S1), (p, S2), (d, S3)}
TEQUIV:{ (*p,y)}
```

```
S4:-----
TREF:{ a,d,x,y,*p }
TGEN:{ f }
DEP:{
    a: S1--->S4
    d: S3--->S4
}
TDEF:{(a, S1), (p, S2), (d, S3), (f, S4)}
TEQUIV:{ (*p,y)}

S5:-----
TREF:{ a,d,x,y,*p }
TGEN:{ g }
DEP:{
    a: S1--->S5
    d: S3--->S5
}
TDEF:{(a, S1), (p, S2), (d, S3), (f, S4), (g, S5)}
TEQUIV:{ (*p,y)}

S6:-----
TREF:{ i,p }
TGEN:{ y,*p }
DEP:{
    p: S2--->S6
}
TDEF:{(a, S1), (p, S2), (d, S3), (f, S4), (g, S5), (y, S6), (*p, S6)}
TEQUIV:{ (*p,y)}
```

```
S7:-----
TREF:{ a,d,x,y,*p }
TGEN:{ h }
DEP:{
    a: S1--->S7
    d: S3--->S7
    y: S6--->S7
    *p: S6--->S7
}
TDEF:{(a, S1), (p, S2), (d, S3), (f, S4), (g, S5), (y, S6), (*p, S6), (h, S7)}
TEQUIV:{ (*p,y)}

S8:-----
TREF:{ y,*p }
TGEN:{ f }
DEP:{
    y: S6--->S8
    *p: S6--->S8
    f: S4->S8
}
TDEF:{(a, S1), (p, S2), (d, S3), (f, S8), (g, S5), (y, S6), (*p, S6), (h, S7)}
TEQUIV:{ (*p,y)}
```

3. Experiment Report

首先依據宣告的type來決定, 哪些變數會是ptr, 接著紀錄這些變數。

用store來區分不同的statement, 用load 的次數來分辨ptr的level, 接著依照每個statement來分別紀錄他們的TGEN, TDEF, TREF, TEQUIV

紀錄完個別statement的set後, 更新完整的TEQUIV, 用array 來紀錄不同的equivalence。接著再更新TREF, TGEN, TDEF。

最後輸出結果。