### 第四十次2020年11月2日-11月8日

#### 代码修改

**LoadInstance添加代码:**

else if ( clause.Size() == 2 ) {

Add\_Binary\_Clause\_Naive( clause[0], clause[1] );

//1104

if (clause.Exact()){

Add\_Binary\_Clause\_Naive( ~clause[0], ~clause[1] );

}

for ( i = 0; i < cnf.Num\_Clauses(); i++ ) {

Clause &clause = cnf[i];

if(clause.Exact())

{

breakLoop1:

continue;

int num\_bi = (clause.Size() - 1) \* (clause.Size()) / 2;

for (unsigned x = 0; x < clause.Size() - 1; x++) {

for (unsigned y = x + 1; y < clause.Size(); y++) {

unsigned m = ~(clause[x]);

unsigned n = ~(clause[y]);

if (\_binary\_clauses[m][n]) {

num\_bi--;

} else {

goto breakLoop1;

}

}

}

if (num\_bi == 0) {

clause.SetExact(true);

}

}

}

**BCP添加代码:**

Literal lit2 = \_dec\_stack[start];

if(true)

{

vector<unsigned> & \_watched = \_long\_watched\_lists[lit2];

for ( i = 0, size = \_watched.size(); i < size; ) {

Clause &clause = \_long\_clauses[\_watched[i]];

if ( clause.Size() < 3 ) {

cerr << "ERROR[Solver::BCP]: \_long\_clauses[" << \_watched[i] << "] = " << \_long\_clauses[\_watched[i]] << endl;

}

assert(clause.Size() >= 3);

if(clause.Exact())

{

for(int j = 0;j < clause.Size(); j++)

{

Literal li = clause[j];

if(li == lit2)

continue;

if(Lit\_Undecided( li ))

{

Assign( li, Reason( lit2 ) );

}

if(Lit\_SAT(li))

{

\_big\_learnt[1] =~li;

return Reason( lit );

}

}

}

}

}

**MAX\_LIT添加代码:**

case MAX\_LIT:

for (int i = 0; i < \_long\_clauses.size(); i++) {

Clause tempClause = \_long\_clauses[i].Copy();

tempClause.Sort(tempSorter);

for (int j = tempClause.Size() - 1; j >= 0; j--) {

\_heur\_sorted\_lits[tempCounter++] = tempClause[j];

}

}

break;

**总结:**

目前，LoadInstance、BCP添加的代码经过调试没有问题，MAX\_LIT添加代码还没有完全改完，运行时报错。