### 第三十六次2020年9月28日-10月4日

#### 1 Clause Elimination for SAT and QSAT

SAT和QSAT处理的公式大小不同。SAT处理的CNF公式变量子句数目很多，限制了很多理论上在多项式时间上的子句删减技术。QSAT实例通常相对较小，因为QBF的语言可以通过量化实现更简洁的编码。

本文的重点是SAT和QSAT求解的预处理和简化技术。这项工作的一方面是通过在搜索之前和搜索过程中将其他简化技术集成到求解过程中来进一步提高SAT和QSAT求解效率，另一方面是了解不同简化技术之间的关系。特别的，我们专注于开发和分析CNF（针对SAT）和PCNF公式（针对QSAT）的子句消除程序，这是SAT和QSAT求解器的标准输入格式。

子句删除程序：在不改变满足性的情况下，从公式F中迭代删除拥有冗余属性P的子句，直到固定点为止。

然而检测子句在某个属性下是冗余的问题是co-NP完全的。两个众所周知的冗余属性是重言式和包含式。更复杂的冗余属性有阻塞子句。我们基于已有技术的不对称变体引入了一种新的消除程序。在不对称变体中，首先用某些文字扩展CNF中的子句，以保留CNF的可满足性。原始子句被增强子句代替。如果增强子句具有冗余属性，则将其从CNF中删除。否则，将还原原始子句。我们还开发了一系列新颖的涵盖条款消除程序。 对于更一般的QSAT设置的应用程序，我们自然地将CNF级别的程序提升为PCNF，自然而然地，它涉及的程度更大。

#### 2 实验

内存访问错误的问题：Comp中clause为空导致启发式pick变量时出现内存访问错误，修改方法为考虑找二元子句。若整个Comp SAT，Comp的clause和var应该都为空，而报错时clause空var非空，说明一定有二元子句未满足，因此启发式pick时不能只考虑长子句，二元子句也必须考虑进去。

模型数结果不正确的问题：在使用MAX\_LIT启发式和exact方法时，得到的模型数比正确值偏大。经过调试发现，Load\_Instance()中binary clause的size的赋值出现错误，导致部分二元子句丢失，约束减少因此模型数变多。修改过后的模型数结果计算正确。

Long\_ watched\_ list在BCP\_ component中为空的问题？并没有在程序中找到generate\_long\_watch\*，构造Long\_watched\_list的函数我都找了一下。

Load\_Instance()中除了单元子句外的子句都参与构造Long\_watched\_list，Load\_Instance()函数末尾将函数内构造的Long\_watched\_list清空。

Vivification()函数先清空Long\_watched\_list之后重新构造了前两位文字监听的Long\_watched\_list。暂时还没有完全弄清楚Long\_watched\_list是否存在问题。但目前模型数结果已经正确，估计问题不大。