### 第三十五次2020年8月17日-8月23日

#### 1 SAT求解器

在本文中，我们以冲突驱动学习的方式展示了一个小型，完整而有效的SAT求解器，这是CH AFF所证明的。 我们旨在提供有关实现的详细信息，以使读者能够在很短的时间内构造自己的求解器。 这将使SAT解算器的用户可以对当前最先进的SAT技术进行领域特定的扩展或改编，以满足特定应用领域的需求。 提出的求解器就是为此而设计的，除其他外，还包括一种用于添加任意布尔约束的机制。 它还支持通过增量SAT接口有效地解决一系列相关的SAT问题。

1.介绍：

SAT求解器在各种应用中的使用正在逐步发展。 随着如何有效地将问题编码为SAT的见识不断增加，SAT求解器已成功解决了越来越多的问题域。 对于电子设计自动化（EDA）行业而言，这尤其正确[BC 99，Lar92]。 当前最先进的求解器可以适应+满足这些问题域的特定特征，从而进一步放大了成功[AR 02，ES03]。

但是，即使对问题域和现代SAT技术都有透彻的了解，修改现有的求解器也可能是耗时的过程，需要进入一万行软件包的内部工作。同样，从头开始编写求解器通常意味着要花费大量时间重新发现正确而有效的求解器的复杂细节。问题是，尽管现代SAT求解器中使用的技术已有充分的文献记载，但是实现所需的细节尚未得到足够的介绍。

在2002年秋天，作者实施了求解器SATZOO和SAT-N IK。为了充分了解现代SAT解决方案所需的实现技巧，有必要参考先前实现的源代码。我们发现，其中包含的材料可以更容易访问，这对于SAT社区是理想的。因此，本文的主要目标是弥合现有的SAT技术描述与其实际实现之间的差距。

我们将通过基于冲突驱动的回溯[MS96]的思想，以及受监视的文字和动态变量排序[MZ01]，介绍最小的SAT求解器MIN ISAT的代码来实现此目的。 MIN ISAT的原始C ++源代码（可从http://www.cs.chalmers.se/~een下载）在600行以下（不包括注释），并且是重新考虑和简化了SATZOO和SATN设计的结果。不牺牲效率的IK。我们将以熟悉C ++或Java的任何人都可以访问的方式呈现代码的所有相关部分。

所提供的代码包括一个增量SAT接口，它可以解决一系列相关问题，并可能带来巨大的效率提升[ES03]。通过提供一种定义布尔变量的任意约束的机制，我们还概括了SAT问题表述的表达能力。

从本文的文档中，我们希望您可以用自己喜欢的语言实现一个新的SAT求解器，或者从网上获取MIN ISAT的C ++版本并开始对其进行修改以包含新的有趣的想法。

2. 应用程序接口：

我们首先介绍MIN ISAT的外部接口，用户应用程序可以使用该接口指定和解决SAT问题。 假定您具有有关SAT的基本知识（例如，参见[MS96]）。 变量，文字和向量的类型var，lit和Vec分别在第4节中进行了详细说明。

标准冲突驱动的SAT求解器可以表示子句（具有两个或更多文字）和赋值。 尽管可以将作业视为单元条款，但对它们进行了特殊处理，最好将其视为单独的信息类型。 标准求解器使用的唯一推理机制是单位传播。 一旦子句在当前分配下成为单位（除一个以外的所有文字均为假），就会声明剩余的未绑定文字，从而可能使更多子句成为单位。 该过程将继续进行，直到无法传播更多信息为止。