



北京大学

硕士学位论文

题目： 基于多值语义的不一致度量

姓 名： 肖国辉

学 号： 10701072

院 系： 北京大学 数学科学学院

专 业： 应用数学

研究方向： 计算机软件与理论

导师姓名： 林作铨 教授

二〇一〇年五月

版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以任何方式传播。否则，引起有碍作者著作权之问题，将可能承担法律责任。

摘 要

度量知识库的不一致程度可以帮助我们处理矛盾。目前存在着多种不一致度量的定义，其中一类定义基于多值语义。基于不同的多值语义（包括三值语义、四值语义、极小不一致语义和准经典语义），又可以定义不同的度量。本文仔细分析了这些定义之间的关系，发现基于三值语义、四值语义和极小不一致语义的不一致度量是等价的，但是跟基于准经典语义的度量不同。然后，我们研究了不一致度量的计算问题。计算复杂度分析显示，计算基于多值语义的度量通常是不易计算的。为了解决这个问题，本文给出了两个归约算法，把基于四值语义（三值语义，极小不一致语义）和准经典语义的度量归约为部分极大可满足问题。部分极大可满足问题是已经有一些高效的求解器可用，我们实现了这两个归约算法，并在一些基准测试数据集上，使用不同的求解器进行了实验，初步的实验结果表明了算法的有效性，可以度量大规模的知识库。

关键词：不一致度量，多值语义，超协调逻辑，可满足问题

Inconsistency Measurement under Multi-Valued Semantics

Guohui Xiao (Applied Mathematics)

Supervised by **Professor Zuoquan Lin**

Abstract

Measuring the inconsistency degree of a knowledge base can help us to deal with inconsistencies. Several inconsistency measures have been given under different multi-valued semantics, including 4-valued semantics, 3-valued semantics, minimal inconsistent semantics and quasi classical semantics. In this paper, we first carefully analyze the relationship among these inconsistency measures by showing that the inconsistency degrees under 4-valued semantics, 3-value semantics, minimal inconsistent semantics are the same, but different from the one based on quasi classical semantics. We then consider the computation of these inconsistency measures and show that computing inconsistency measurement under multi-valued semantics is usually intractable. To tackle this problem, we propose two novel algorithms that respectively reduce the problems of computing inconsistency degrees under 4-valued semantics (3-valued semantics, minimal inconsistent semantics) and under quasi classical semantics into the partial Max-SAT problems. We implement these algorithms and do experiments on some benchmark data sets. The preliminary but encouraging experimental results show that our approach is efficient to handle large knowledge bases.

Keywords: inconsistency measure, multi-valued semantics, paraconsistent logic, satisfiability problem

目 录

第一章 绪论	1
1.1 研究背景	1
1.1.1 超协调推理	1
1.1.2 不一致度量	3
1.2 问题陈述	3
1.3 主要工作	4
1.4 文章结构	5
第二章 不一致度量与可满足性问题	7
2.1 不一致度量	7
2.1.1 基于多值语义的不一致度量	7
2.1.2 多值语义	8
2.2 可满足性问题	12
2.2.1 可满足性问题	12
2.2.2 极大可满足性问题	14
2.2.3 带权的极大可满足问题	14
2.2.4 部分极大可满足性问题	15
第三章 不一致度量的计算	19
3.1 基于不同多值语义的度量定义之间的关系	19
3.2 不一致度量问题的复杂度分析	21
3.3 归约算法	23
3.3.1 归约基于四值语义的度量到部分极大可满足性问题	23

3.3.2	归约基于准经典语义的度量到部分极大可满足性问题	26
3.4	相关工作	31
第四章	算法实现和测评	33
4.1	算法实现	33
4.1.1	程序基本结构	33
4.1.2	归约不一致度量问题为部分极大可满足问题	34
4.1.3	调用部分极大可满足问题求解器计算不一致度	36
4.2	性能测评	37
第五章	总结和展望	43
5.1	总结	43
5.2	展望	43
致 谢		53

第一章 绪论

1.1 研究背景

在知识表示和推理领域，不一致处理是一个传统的研究问题。由于知识库的质量问题、不同知识库的合并等原因，实际中用到的知识库通常含有不一致信息。基于经典逻辑，在不一致的知识库时出现平凡性问题，使推理的结果失去意义。所以，需要其他的方法来解决或克服这一个问题，现在不一致处理仍然是一个研究的热点问题。

1.1.1 超协调推理

针对经典逻辑的平凡性问题，很多学者提出了多种超协调逻辑。与经典逻辑不同，当知识库出现矛盾的，超协调逻辑并不会推导出所有结论，这样更加符合人们日常推理的习惯。

形式化地，在一个超协调逻辑的推理关系 \models_p 中，存在着任意公式 α 和 β ，使得 $\{\alpha, \neg\alpha\} \not\models_p \beta$ 。

主要有两种实现超协调性的方法：超协调的模型论和超协调的证明论[1]。

从模型论看，在经典逻辑中， $S \models \alpha$ 的含义是， α 在 S 的所有模型中为真。由于没有模型满足 S ，所以对任意的 α 都有 $S \models \alpha$ 。如果对 S 的部分模型做一个优先的选择，使得若 α 在 S 的优先模型中为真，则 $S \models_p \alpha$ ，这时候就可以避免平凡推理。称这种定义优先模型来实现非平凡推理的途径为优先语义。优先语义可以基于经典逻辑，从模型论直接刻画超协调性。优先语义同样可以基于多值逻辑，直接在模型中承认矛盾，并采用优先模型向经典的模型论逼近，从而实现超协调的推理。正如[2]所提到的，优先语义可以作为超协调逻辑的统一基础。

从证明论来看，为了达到超协调性，各种超协调逻辑必须减弱自身的推理能力，即一些在经典逻辑中有效的推理规则在超协调逻辑中可能无效。通过分析经

典逻辑产生平凡性 ($\{A, \neg A\} \vdash B$) 的原因可以知道, 在一个超协调推理系统中, 下面三个推理规则必须至少有一个不成立[3, 4, 5, 6]:

1. 等价律: $A \wedge \neg A \vdash B \wedge \neg B$ 。
2. 传递律: 若 $A \vdash B$ 且 $B \vdash C$, 则 $A \vdash C$ 。
3. 合取律: 若 $S \vdash A$ 且 $S \vdash B$, 则 $S \vdash A \wedge B$ 。

故从证明论看, 各种超协调逻辑系统基本上可以分成下面三种途径:

1. 抛弃等价律的超协调逻辑系统称为相干系统。
2. 抛弃传递律 (相当于抛弃了分离规则) 的系统称为正强系统。
3. 抛弃合取律的系统称为非合系统。

按照采用的技术, 超协调系统也可以分为下面几类:

1. 弱化公理系统。这类系统采用和经典逻辑相同的语法形式, 通过弱化经典逻辑中的公理系统和推理规则来达到超协调的目的[7]。它们属于从证明论的角度实现超协调性。
2. 准经典逻辑系统。这类系统保留经典逻辑的语言, 修改或限制经典逻辑的推理方法, 实现超协调推理, 如文献[8, 9, 10, 11]的工作, 也属于证明论的范畴。
3. 多值逻辑[12, 13, 14, 15, 16]。这类系统引入表示矛盾的真值, 使得自身具有描述矛盾的能力, 这样即使是包含矛盾的前提也会存在多值模型, 从而可以进行超协调推理。标记逻辑 (Annotated Logic[17, 18, 19]) 也可以归入这一类。最早的多值系统是三值逻辑, 它有多种语义[20, 21]。为了表示计算机中的信息, Belnap等人提出了四值逻辑[13, 12]。这种做法其实也属于模型论的范畴。
4. 变换系统。这类系统首先将包含矛盾的前提变换到不同的公式, 从而把矛盾分成不相干的两部分, 这样包含矛盾的推理就被转化为经典逻辑的推理。最后通过某种技术对经典逻辑的推理结果给出解释[22, 23, 24, 25, 26, 27]。

1.1.2 不一致度量

近来, 量化一个不协调知识库中不协调信息量的研究引起越来越多的关注。这是因为, 虽然出现不一致信息时, 经典逻辑就会发现平凡性推理的问题。但是超协调逻辑的研究表明了, 不同程度的不一致对推理的结果有不同的影响。不一致度量, 可以帮助我们采取不同的策略来解决[28], 即是该忽略还是该消除。其次, 它也被应用到不同背景中[29], 如软件需求[30, 31, 32], 新闻报告[28], 整形约束[33], 语义万维网本体[34], 和电子商务协议[35]。进一步地, 度量矛盾知识库的不协调信息量可以为消除不协调性方法提供更细粒度的上下文信息[36, 37, 38],

度量不协调度的不同方法基于不同的看待承载不协调性的最小单元的观点[37]。基于语法的方法将不协调的最小单元看成是知识库的一个公式, 例如考虑度量最大协调公式子集[39]。基于语义的方法将不协调的最小单元看成是构成知识库的命题原子, 例如基于某种超协调语义下的取矛盾真值的命题原子的个数[40, 36, 37, 33, 41])。现在也有一些工作试图把两者结合起来, 例如来考虑每一个公式对于整体的不一致度的贡献[38]。本文中我们集中在使用多值语义度量不协调程度, 属于后者语义方法。我们关注的主要是如何定义和计算描述逻辑知识库的不协调度, 以便给用户理解和处理其不协调知识库的信息。

1.2 问题陈述

本文中关注基于多值语义的一类不一致度量。在这一类中, 基于不同的多值语义可以给出不同的定义, 例如三值语义 (3-valued Semantics) [42]、四值语义(4-valued Semantics)[12]、极小不一致语义 (LP_m Semantics) [43]和准经典语义 (Quasi Classical Semantics) [44]。这些度量在本文中分别记作 ID_3 , ID_4 , ID_{LP_m} 和 ID_Q , 其中ID的含义为不一致度 (Inconsistency Degree)。这些定义分别在不同的文献中提出, ID_4 出现在文献[28, 45]中, ID_3 出现在文献[40], ID_{LP_m} 出现在文献[46], ID_Q 出现在文献[36]中。现在的研究主要集中于某种具体的度量的应用, 但是并没有工作来系统研究它们之间的关系。

如果想要把不一致度量实际应用, 我们需要有效的算法来快速计算不一致

度。在我们之前的工作中[45]，我们已经证明了对于一个给定的命题知识库 K ，计算其基于四值语义的不一致度 (ID_4) 是难解的。遵循近似推理的方法[47]，我们在[45]中给出了一个任意时间近似算法，可以逐渐的逼近实际的不一致度。但是这个算法在大的知识库上的表现并不理想，收敛的速度比较慢。

不一致度量的计算可以看作是一个复杂的搜索问题。对于这样的问题，一种方法是直接给出一个算法，使用启发式搜索和剪枝策略。我们之前的近似算法[45]就是基于这一思路，但是效果并不理想。本文遵循另一外思路，即把不一致度的计算归约到可满足问题。

1.3 主要工作

本文的工作围绕着基于多值语义的不确定度量的理论和算法，主要工作如下：

1. 我们首先研究基于不同的多值语义的度量之间的关系。我们发现，对于给定的知识库，基于三值、四值和极小不一致语义的度量是等价的，而小于或等于基于准经典语义的度量。这样就意味着，今后我们只需要考虑 ID_4 和 ID_Q 就足够了。
2. 我们先给出不一致度量计算相关问题的复杂度分析。基于之前的工作[45]和不同度量间的关系，我们把原有的 ID_4 相关复杂度结论扩展到 ID_3, ID_4 和 ID_Q 。这些结论说明了，不一致度量的计算的相关问题都是NP-难解的。
3. 我们提出了把不一致度计算归约为已有的可满足问题。部分可满足问题 (Partial Max-SAT) 已经有了很多高效的求解器。不一致度的计算可以很自然的归约到部分可满足问题，针对 ID_4 和 ID_Q ，我们给出了两个高效的归约算法。
4. 使用现有的部分可满足问题求解器，我们实现了这两个算法。初步的实验结果说明了我们的方法可以有效度量大的知识库的不一致度。与原有的方法相比，有着明显的优势。

1.4 文章结构

本文后面的结构如下：

- 在第二章中，我们回顾了基于多值语义的不一致度量的定义和可满足性相关问题。
- 在第三章中，首先我们证明了 ID_4 ， ID_3 ， ID_{LP_m} 的等价关系。然后给出了计算不一致度量相关问题的复杂度分析。最后，给出了两个归约算法分别计算 ID_4 和 ID_Q 。
- 在第四章中，我们实现了上述的两个算法，并在一些数据集上做了测试。初步的实验结果说明了算法的有效性。然后我们比较了相关工作。
- 在第五章，我们总结了全文，并且展望了未来的工作。

第二章 不一致度量与可满足性问题

本文中，我们使用小写字母 p, q, r 来表示命题变元，有时也加上下标。一个文字（literal）为一个命题变元 p 或其否定 $\neg p$ 。变元集 $\mathcal{A} = \{p_1, \dots, p_n\}$ 上的命题语言 $\mathcal{L}_{\mathcal{A}}$ 是由 \mathcal{A} 中的变元构成的所有的命题公式的集合。一个 \mathcal{A} 上的命题知识库 K 是一个由 \mathcal{A} 中的命题变元构成的命题公式的集合，即 $\mathcal{L}_{\mathcal{A}}$ 的子集。本文只考虑元素个数有限的知识库。

子句（clause） $\gamma = l_i \vee \dots \vee l_n$ 为一些命题文字的析取。一个公式满足合取范式（Conjunctive Normal Form, CNF），如果它是一些子句的合取 $\gamma_1 \wedge \dots \wedge \gamma_k$ ，通常表示为一个子句集 $K = \{\gamma_1, \dots, \gamma_k\}$ 。

对于集合 S ，我们用 $|S|$ 来表示 S 内的元素个数，用 $\wp(S)$ 来表示 S 的幂集，即 S 的所有子集构成的集合。

2.1 不一致度量

2.1.1 基于多值语义的不一致度量

基于不同的多值语义可以定义不同的不一致度量。比较重要的有基于四值语义（4-valued semantics）的[28, 45]，基于三值语义（3-valued semantics）的[40]，基于 LP_m 语义的[46]和基于准经典语义（Quasi Classical Semantics 或 Q-Semantics）的度量[36]。上述语义全都使用了一个额外的真值 B 来表示矛盾信息。

为了区别于多值语义，在本文中，我们称原来的二值语义为经典语义，并用 \models 来表示经典蕴含关系。同时我们用 \models_i （ $i = 3, 4, Q, LP_m$ ）来表示多值蕴含关系。

为了统一讨论不同的定义，我们给出基于多值语义的度量一致定义如下：

定义2.1. 若 \mathcal{I} 是命题知识库 K 在 i -语义（ $i = 3, 4, LP_m, Q$ ）下的一个模型，则知

知识库 K 相对于模型 \mathcal{I} 的不一致度, 记作 $ID_i(K, \mathcal{I})$, 定义为:

$$ID_i(K, \mathcal{I}) = \frac{|\{p \in Var(K) \mid p^{\mathcal{I}} = B\}|}{|Var(K)|},$$

其中分子内的 $\{p \in Var(K) \mid p^{\mathcal{I}} = B\}$ 称作 \mathcal{I} 相对于 K 的冲突集, 记为 $Conflict(K, \mathcal{I})$ 。

知识库 K 在 i -语义下的不一致度, 记作 $ID_i(K)$, 定义为

$$ID_i(K) = \min_{\mathcal{I} \models_i K} ID_i(K, \mathcal{I}),$$

其中 $\mathcal{I} \models_i K$ 表示 \mathcal{I} 是 K 在 i -语义下的一个模型。

下面我们分别给出几个不同的多值语义的介绍。

2.1.2 多值语义

2.1.2.1 四值语义

不同于经典语义的两个真值, 四值语义[12, 48]的真值集合有四个元素: *true*, *false*, *unknown* 和 *both*, 分别记作 t, f, N, B 。其中真值 N 用来表示不完全信息, 真值 B 表示矛盾信息。

这四个真值和序关系 \preceq 一起构成了一个格: $\mathbf{FOUR} = (\{t, f, B, N\}, \preceq)$, 其中 $f \preceq N \preceq t, f \preceq B \preceq t, N \not\preceq B, B \not\preceq N$ 。

在四值语义下, 连接词 \vee 和 \wedge 按照序关系 \preceq 下的上界和下界来分别定义。同时一元运算符 \neg 定义为 $\neg t = f, \neg f = t, \neg B = B, \neg N = N$ 。

在 \mathbf{FOUR} 中, $\{t, B\}$ 含有成真信息。一个四值解释 \mathcal{I} 是知识库 K 的一个四值模型, 记作 $\mathcal{I} \models_4 K$, 当且仅当对每一个公式 $\phi \in K, \phi^{\mathcal{I}} \in \{t, B\}$ 都成立。一个知识库如果有四值模型, 那么它是四值可满足的。一个知识库 K 四值蕴含公式 φ , 记作 $K \models_4 \varphi$, 当且仅当每一个 K 的四值模型都是 φ 的四值模型。

例2.2. 给定知识库 $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$ 。考虑 K 的三个模型 $\mathcal{I}_1, \mathcal{I}_2$ 和 \mathcal{I}_3

如下:

$$\begin{aligned} p^{\mathcal{I}_1} &= t, q^{\mathcal{I}_1} = B, r^{\mathcal{I}_1} = f, s^{\mathcal{I}_1} = t, u^{\mathcal{I}_1} = N; \\ p^{\mathcal{I}_2} &= B, q^{\mathcal{I}_2} = f, r^{\mathcal{I}_2} = B, s^{\mathcal{I}_2} = t, u^{\mathcal{I}_2} = N; \\ p^{\mathcal{I}_3} &= B, q^{\mathcal{I}_3} = B, r^{\mathcal{I}_3} = B, s^{\mathcal{I}_3} = t, u^{\mathcal{I}_3} = N. \end{aligned}$$

显然, $ID_4(K, \mathcal{I}_1) = \frac{1}{5}$, $ID_4(K, \mathcal{I}_2) = \frac{2}{5}$, $ID_4(K, \mathcal{I}_3) = \frac{3}{5}$ 。同时我们知道由于 K 不是二值可满足的, K 的每一个模型至少有一个矛盾值 B , 所以 $ID_4(K) = \frac{1}{5}$ 。

2.1.2.2 三值语义

顾名思义, 三值语义下有三个真值 t, f, B 。三值语义[42]可以看作是一种带了限制的四值语义, 即 \mathcal{I} 是知识库 K 的三值模型, 当且仅当 \mathcal{I} 是 K 的四值模型, 并且对于每个命题变元 $p \in \mathcal{A}$, 都有 $p^{\mathcal{I}} \neq N$ 。同样的, 我们也可以定义三值蕴含关系 \models_3 。

例2.3. 知识库 $K = \{p, \neg q, \neg p \vee q, r \vee s\}$, 三个四值解释

$$\begin{aligned} \mathcal{I}_1 : p^{\mathcal{I}_1} &= B, q^{\mathcal{I}_1} = f, r^{\mathcal{I}_1} = t, s^{\mathcal{I}_1} = t; \\ \mathcal{I}_2 : p^{\mathcal{I}_2} &= B, q^{\mathcal{I}_2} = B, r^{\mathcal{I}_2} = t, s^{\mathcal{I}_2} = t; \\ \mathcal{I}_3 : p^{\mathcal{I}_3} &= B, q^{\mathcal{I}_3} = B, r^{\mathcal{I}_3} = t, s^{\mathcal{I}_3} = N. \end{aligned}$$

其中 \mathcal{I}_1 和 \mathcal{I}_2 也是三值模型, 但是 \mathcal{I}_3 不是三值模型, 因为 $s^{\mathcal{I}_3} = N$ 。

而且 $ID_3(K, \mathcal{I}_1) = \frac{1}{4}$, $ID_3(K, \mathcal{I}_2) = \frac{2}{4}$ 。

同时 $ID_3(K) = \frac{1}{4}$ 。

2.1.2.3 极小不一致语义

极小不一致模型是一类模型, 按照某些优先序原则, 只选择最小的解释作为模型。这里介绍 LP_m [43] 作为一个极小不一致模型的代表。 LP_m 模型是“极经典”的三值模型。一个三值模型是一个 LP_m 模型, 当且仅当没有其它的三值模型含有更少的矛盾。形式化地, $\mathcal{I} \models_{LP_m} \alpha$ 当且仅当 $\mathcal{I} \models_3 \alpha$, 而且没有其他的 α 的三值模型 \mathcal{J} 使得 $\{p \mid p^{\mathcal{J}} = B\} \subsetneq \{p \mid p^{\mathcal{I}} = B\}$ 成立。

例2.4. 知识库 $K = \{p, \neg q, \neg p \vee q, r \vee s\}$, 三个四值模型:

$$\begin{aligned}\mathcal{I}_1 : p^{\mathcal{I}_1} = B, q^{\mathcal{I}_1} = f, r^{\mathcal{I}_1} = t, s^{\mathcal{I}_1} = t; \\ \mathcal{I}_2 : p^{\mathcal{I}_2} = B, q^{\mathcal{I}_2} = B, r^{\mathcal{I}_2} = t, s^{\mathcal{I}_2} = t; \\ \mathcal{I}_3 : p^{\mathcal{I}_3} = B, q^{\mathcal{I}_3} = B, r^{\mathcal{I}_3} = t, s^{\mathcal{I}_3} = N.\end{aligned}$$

其中 $\mathcal{I}_1, \mathcal{I}_2$ 也是三值模型, 由于 \mathcal{I}_3 不是三值模型, 也不可能是 LP_m 模型。

比较 \mathcal{I}_1 和 \mathcal{I}_2 , 可以看到 $\{p \mid p^{\mathcal{I}_1} = B\} \subsetneq \{p \mid p^{\mathcal{I}_2} = B\}$, 所以 \mathcal{I}_2 也不可能是 LP_m 模型。

同时我们知道由于 K 不是二值可满足的, K 的每一个模型至少会有一个矛盾值 B , 所以不可能有模型比 \mathcal{I}_1 含有的矛盾值更少, 从而 \mathcal{I}_1 是一个 LP_m 模型。同时 $ID_{LP_m}(K) = \frac{1}{4}$ 。

2.1.2.4 准经典语义

给定知识库 K 的命题变元集合 \mathcal{A} , 命题集 \mathcal{A}^\pm 定义为

$$\mathcal{A}^\pm = \{+p, -p \mid p \in \mathcal{A}\}.$$

任意的 $I \in \wp(\mathcal{A}^\pm)$ 称作 K 的准经典解释, 其中 $\wp(\mathcal{A}^\pm)$ 是 \mathcal{A}^\pm 的幂集。

给定一些文字 l_1, \dots, l_n , 子句 $l_1 \vee \dots \vee l_n$ 关于 l_i 的聚焦(focus), 记作 $\otimes(l_1 \vee \dots \vee l_n, l_i)$, 定义为

$$\otimes(l_1 \vee \dots \vee l_n, l_i) = l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_n.$$

对于知识库 K , 准经典模型 (Q-model) 定义如下:

定义2.5 (准经典模型[49]). 假定 p 是一个命题变元, $\gamma_1, \dots, \gamma_m$ 是一些子句,

l_1, \dots, l_n 为文字, 可满足关系 $\models_Q \subseteq \wp(\mathcal{A}^+) \times \mathcal{L}_{\mathcal{A}}$ 定义如下:

$$\begin{aligned}
\mathcal{I} \models_Q p & \quad \text{iff} \quad +p \in \mathcal{I} \\
\mathcal{I} \models_Q \neg p & \quad \text{iff} \quad -p \in \mathcal{I} \\
\mathcal{I} \models_Q l_1 \vee \dots \vee l_n & \quad \text{iff} \quad [\mathcal{I} \models_Q l_1 \text{ 或者 } \dots \text{ 或者 } \mathcal{I} \models_Q l_n] \\
& \quad \text{且 [对所有的 } i, \text{ 如果 } \mathcal{I} \not\models_Q \neg l_i \\
& \quad \text{则 } \mathcal{I} \models_Q \otimes(l_1 \vee \dots \vee l_n, l_i)] \\
\mathcal{I} \models_Q \{\gamma_1, \dots, \gamma_m\} & \quad \text{iff} \quad \mathcal{I} \models_Q \gamma_i (1 \leq i \leq m).
\end{aligned}$$

对于任意形式的知识库 K , \mathcal{I} 是 K 的准经典模型当且仅当 \mathcal{I} 是 $CNF(K)$ 的准经典模型, 其中 $CNF(K)$ 是 K 通过经典变换得的合取范式, 即:

$$\begin{aligned}
\mathcal{I} \models_Q \neg\neg\alpha \vee \beta & \quad \text{iff} \quad \mathcal{I} \models_Q \alpha \vee \beta \\
\mathcal{I} \models_Q \neg(\alpha \wedge \beta) \vee \gamma & \quad \text{iff} \quad \mathcal{I} \models_Q (\neg\alpha \vee \neg\beta) \vee \gamma \\
\mathcal{I} \models_Q \neg(\alpha \vee \beta) \vee \gamma & \quad \text{iff} \quad \mathcal{I} \models_Q (\neg\alpha \wedge \neg\beta) \vee \gamma \\
\mathcal{I} \models_Q \alpha \wedge (\beta \vee \gamma) & \quad \text{iff} \quad \mathcal{I} \models_Q (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \\
\mathcal{I} \models_Q \alpha \vee (\beta \wedge \gamma) & \quad \text{iff} \quad \mathcal{I} \models_Q (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \\
\mathcal{I} \models_Q (\alpha \rightarrow \beta) \vee \gamma & \quad \text{iff} \quad \mathcal{I} \models_Q \neg\alpha \vee \beta \vee \gamma \\
\mathcal{I} \models_Q \neg(\alpha \rightarrow \beta) \vee \gamma & \quad \text{iff} \quad \mathcal{I} \models_Q (\alpha \wedge \neg\beta) \vee \gamma
\end{aligned}$$

类似于Belnap的四值逻辑, 准经典模型也可以看作是从命题变元集 \mathcal{A} 到真值集合 $\{B, t, f, N\}$ 的一个映射:

$$p^{\mathcal{I}} = \begin{cases} t & \text{iff } +p \in \mathcal{I} \text{ 且 } -p \notin \mathcal{I}; \\ f & \text{iff } +p \notin \mathcal{I} \text{ 且 } -p \in \mathcal{I}; \\ B & \text{iff } +p \in \mathcal{I} \text{ 且 } -p \in \mathcal{I}; \\ N & \text{iff } +p \notin \mathcal{I} \text{ 且 } -p \notin \mathcal{I}. \end{cases}$$

这样按照定义2.1的方式, 我们可以定义基于准经典语义的度量 ID_Q 。

例2.6 (例2.2 续). $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$. 考虑下面的三个四值模型 \mathcal{I}_1 , \mathcal{I}_2 和 \mathcal{I}_3 :

$$\begin{aligned} p^{\mathcal{I}_1} &= t, q^{\mathcal{I}_1} = B, r^{\mathcal{I}_1} = f, s^{\mathcal{I}_1} = t, u^{\mathcal{I}_1} = N; \\ p^{\mathcal{I}_2} &= B, q^{\mathcal{I}_2} = f, r^{\mathcal{I}_2} = B, s^{\mathcal{I}_2} = N, u^{\mathcal{I}_2} = t; \\ p^{\mathcal{I}_3} &= B, q^{\mathcal{I}_3} = B, r^{\mathcal{I}_3} = B, s^{\mathcal{I}_3} = t, u^{\mathcal{I}_3} = N. \end{aligned}$$

根据定义2.5, \mathcal{I}_1 和 \mathcal{I}_2 不是 K 的准经典模型, 尽管它们是 K 的四值模型, 而 \mathcal{I}_3 是一个准经典模型, 而且我们有 $\text{ID}_Q(K) = \text{ID}_Q(K, \mathcal{I}_3) = \frac{3}{5}$.

2.2 可满足性问题

2.2.1 可满足性问题

判断一个CNF的知识库的可满足性 (Satisfiability) 的问题称作可满足性问题 (SAT问题)。众所周知, SAT问题是NP-完全的。尽管SAT问题是难解的, 但是现在已经有了很多的高度优化的求解器, 可以处理很大规模的输入, 并且已经广泛应用于工业领域。

可满足问题 (satisfiability problem) 的实例为一些子句 (clause) 的合取, 子句为由变元或变元的否定组成的析取式。如果我们用 p_i 来表示命题变元, 一个可满足问题的例子如下:

$$(p_1 \vee p_3 \vee \neg p_4) \wedge (p_4) \wedge (p_2 \vee \neg p_3)$$

其中 \vee 为析取连接词, \wedge 为合取连接词, 而 $\neg p_i$ 表示 p_i 的否定。

给定变元 p_1, p_2, \dots, p_n 上的子句集 $\{\gamma_1, \gamma_2, \dots, \gamma_m\}$, 则可满足问题为判定公式

$$\gamma_1 \wedge \gamma_2 \wedge \dots \wedge \gamma_m$$

是否可满足, 即是否存在一个变元赋值, 使得上面的公式成真。也就是说, 使其中的每一个子句 γ_j 都成真。

2.2.1.1 CNF 文件格式

最常用的 CNF 实例的文件格式为 Dimacs 格式[50]，其为纯文本格式，扩展名为 “.cnf”，结构如下：

1. **导言区** 导言包含关于实例的基本信息。这些信息由一些行组成，每一行由一个字符开始，决定了这一行的类型。这些类型包括：

- (a) **注释行** 注释是一些给用户给读的信息，会被SAT求解器忽略。注释行出现在导言区的最前面，每一行以一个小写的字母 **c** 开始，例如：

```
c This is an example of a comment line.
```

- (b) **问题行** 每一个CNF文件仅有一个问题行。这一行包含了本实例的基本信息。形式如下：

```
p FORMAT VARIABLES CLAUSES
```

小写字母 **p** 表示这一个问题描述行。FORMAT表示文件格式的类型，对于 CNF文件，应为 “.cnf”。VARIABLES为一个整数 m ，表示变元的个数。CLAUSES也是一个整数 n ，表示子句的个数。这一行必须出现的导言区的最后。

2. **子句区** 问题描述行后就是子句区。在子句区中，每一个变元表示为一个整数1到 n 之前的整数。每一个子句表示为一个整数序列，中间以空白字符分隔。正的整数 i 表示一个正的文字 i ；负的整数 $-i$ 表示一个负文字，即 i 的否定。每一个子句以一个0结束。

例2.7. 考虑一个CNF的实例

$$(p_1 \vee p_3 \vee \neg p_4) \wedge (p_4) \wedge (p_2 \vee \neg p_3)$$

一个对应的CNF文件可以为：

```

c Example CNF format file
c
p cnf 4 3
1 3 -4 0
4 0
2 -3 0

```

2.2.2 极大可满足性问题

极大可满足性问题是可满足性问题的最优化版本，目标为寻找一个经典二值赋值，满足尽可能多的子句。即给定一个子句集 K ，寻找解释 $\hat{\mathcal{I}}$ ，使得

$$\hat{\mathcal{I}} = \arg \max_{\mathcal{I}} |\{\gamma \mid \gamma \in K, \mathcal{I} \models \gamma\}|.$$

2.2.3 带权的极大可满足问题

如果给每一个子句加上权重，极大可满足问题就变成了加权极大可满足问题 (Weighted Max-SAT Problem)。一个加权的子句定义为一个二元组 $\langle w, \gamma \rangle$ ，其中 w 是正整数，表示子句 γ 的权重。

给定一个带权的子句集 $W = \{\langle w_i, \gamma_i \rangle \mid i = 1, \dots, n\}$ ，极大可满足问题的目标为寻找一个二值赋值，使得所满足的子句的权重和最大。即寻找解释 $\hat{\mathcal{I}}$ ，使得

$$\hat{\mathcal{I}} = \arg \max_{\mathcal{I}} \left\{ \sum w \mid \langle w, \gamma \rangle \in W, \mathcal{I} \models \gamma \right\}$$

2.2.3.1 WCNF 文件格式

带权的极大可满足问题的实例最常使用的格式是 Dimacs 格式[51]，其结构与 CNF 基本相同：

1. 导言区 文件的基本信息

- (a) 注释行 注释信息。
- (b) 问题行 格式为“p wcnf nbvar nbclauses”，与 CNF 格式的差别在第二个 FORMAT 上，对于 WCNF 文件，这里一定是 “wcnf”。

2. **带权子句区** 每一行由一个整数的序列组成，表示一个带权的子句。第一个整数表示这个子句的权重 w ，满足 $1 \leq w \leq 2^{20}$ 。接下来的整数序列表示组成这个子句的文字。最后以一个0结束。例如 $15 \ 1 \ -2 \ 4 \ 0$ 可以表示一个带权的子句 $\langle 15, p_{15} \vee p_1 \vee \neg p_2 \vee p_4 \rangle$ 。

例2.8. 给定一个带权的极大可满足问题

$$W = \{\langle 10, p_1 \vee \neg p_2 \rangle, \langle 3, \neg p_1 \vee p_2 \vee \neg p_3 \rangle, \langle 8, \neg p_3 \vee p_2 \rangle, \langle 5, p_1 \vee p_3 \rangle\}$$

则 W 可以表示为下面的 *WCNF* 文件：

```

c
c comments Weighted Max-SAT
c
p wcnf 3 4
10 1 -2 0
3 -1 2 -3 0
8 -3 2 0
5 1 3 0

```

2.2.4 部分极大可满足性问题

作为SAT问题的一个扩展，部分可满足问题(partial maximum satisfiability problem, Partial Max-SAT)近年来成了一个研究热点。一个部分 Max-SAT 问题形如 $P = (H, S)$ ，其中 H 是一个子句集，称作硬部分； S 是另外一个子句集，称作软部分。目标为找一个经典二值赋值，使得在满足所有 H 中的硬子句的前提下，尽可能多的满足 S 的软子句。也就是说寻找解释 $\hat{\mathcal{I}}$ ，使得

$$\hat{\mathcal{I}} = \arg \max_{\mathcal{I}} |\{\gamma \mid \gamma \in S, \mathcal{I} \models \gamma, \mathcal{I} \models H\}|$$

事实上，部分极大可满足性问题是一类特殊的带权的极大可满足问题，如果合理的分配权值，给硬子句的权值足够大，那么部分极大可满足性问题，可以转化为一个带权的极大可满足问题。

定理2.9. 给定一个部分极大可满足问题 $P = (H, S)$, 其对应的带权的极大可满足问题定义为:

$$W(P) = \{ \langle |S| + 1, \gamma \rangle \mid \gamma \in H \} \\ \cup \{ \langle 1, \gamma \rangle \mid \gamma \in S \}$$

则 \hat{I} 是 P 的最优解, 当且仅当它也是 $W(P)$ 的最优解。

2.2.4.1 WCNF 格式

极大部分部分可满足的文件格式也是 Dimacs 格式[51]:

1. 导言区 文件的基本信息

(a) 注释行 注释信息。

(b) 问题行 格式为“p wcnf nbvar nbclauses top”, 其中与带权的部分可满足问题的CNF格式的差别在于最后的top。top是一个整数, 我们假定它总大于不可满足的软子句的数量之和。在实际使用中, 根据定理2.9, 其值可设为软子句之个数加1。

2. 子句区 每一行由一个整数的序列组成, 表示一个带权的子句, 第一个整数表示权重, 余下的整数表示子句。软子句的权重设为1, 硬子句的权值设为top。

例2.10. 给定部分极大可满足问题 $P = (H, S)$, 其中 $H = \{p_1 \vee \neg p_2 \vee p_4, \neg p_1 \vee \neg p_2 \vee p_3\}$, $S = \{\neg p_2 \vee \neg p_4, \neg p_3 \vee p_2, p_1 \vee p_3\}$ 。则 P 可表示为以下的 WCNF 文件:

```
c
c comments Partial Max-SAT
c
p wcnf 4 5 15
15 1 -2 4 0
15 -1 -2 3 0
1 -2 -4 0
1 -3 2 0
1 1 3 0
```

当前已有的一些部分Max-SAT求解器，例如 SAT4j MaxSAT [52], MSUnCore [53] 和Clone [54, 55]都是高度优化的。他们的性能已经在第三届¹ and第四届² MaxSAT比赛中得到证明。

下面简单介绍一下这几个求解器：

1. SAT4j[52]。SAT4j是一个用Java实现的SAT求解器，其支持可满足问题、部分极大可满足问题和带权的极大可满足问题。SAT4j目前已经随Eclipse³一起发布，并且也在Linux的发行版Ubuntu⁴的软件仓库中。SAT4j可以作为一个Java库来使用，作为一个黑盒可以嵌入到其他的Java程序中。总体来说，SAT4j是一个性能和稳定性都比较好的求解器，而且有比较好的团队支持。
2. MSUnCore[53]。MSUNCORE (Maximum Satisfiability with UNSatisfiable COREs) 是一个基于不可满足核的求解器，其中实现几个不同的Max-SAT算法，有能力处理大型的实例。同时它看作是一种分枝限界算法。在2008年的MaxSAT竞赛中，msu1.2获得了工业MaxSAT组的第一名和工业部分MaxSAT组的第二名。
3. Clone [54, 55]。Clone是一个分枝限界的求解器，同时使用了一种称作 d-DNNF 的易解语言。一旦实例为 d-DNNF 形式，便可以有效的在多项式时间内求解。如果不是，Clone会通过分枝的办法设法把实例变成 d-DNNF 形式。

¹<http://www.maxsat.udl.cat/08/>

²<http://www.maxsat.udl.cat/09/>

³<http://www.eclipse.org>

⁴<http://www.ubuntu.com>

第三章 不一致度量的计算

3.1 基于不同多值语义的度量定义之间的关系

本节分析基于不同多值语义的度量定义之间的关系，最终的结论是 ID_3 , ID_4 和 ID_{LP_m} 是等价的，但是与 ID_Q 不同。

命题3.1. 令 K 是一个命题知识库，则 $ID_3(K) = ID_4(K)$ 。

证明. (1) 首先证明 $ID_4(K) \leq ID_3(K)$:

如果 $\mathcal{I} \models_3 K$, 则 $\mathcal{I} \models_4 K$ 。

$$\begin{aligned} ID_4(K) &= \min \{ID_4(K, \mathcal{I}) \mid \mathcal{I} \models_4 K\} \\ &\leq \min \{ID_4(K, \mathcal{I}) \mid \mathcal{I} \models_3 K\} \\ &= \min \{ID_3(K, \mathcal{I}) \mid \mathcal{I} \models_3 K\} \\ &= ID_3(K). \end{aligned}$$

(2) 接下来证明 $ID_4(K) \geq ID_3(K)$:

任给 K 的一个四值解释 \mathcal{I} , 我们可以定义一个三值解释 \mathcal{I}' 如下

$$p^{\mathcal{I}'} = \begin{cases} p^{\mathcal{I}} & \text{if } p^{\mathcal{I}} \neq N \\ t & \text{if } p^{\mathcal{I}} = N \end{cases}$$

易见, 如果 $\mathcal{I} \models_4 K$, 则 $\mathcal{I}' \models_3 K$ 。而且 $\{p \mid p^{\mathcal{I}} = B\} = \{p \mid p^{\mathcal{I}'} = B\}$, 即 $ID_4(K, \mathcal{I}) = ID_3(K, \mathcal{I}')$ 。所以根据 $ID_4(K)$ 和 $ID_3(K)$ 的定义, 可得 $ID_4(K) \geq ID_3(K)$ 。

综上所述, $ID_4(K) = ID_3(K) = \frac{1}{5}$ 。 □

例3.2 (例2.6 续). $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$. 考虑 K 的四值模型 \mathcal{I}_1 :

$$p^{\mathcal{I}_1} = t, q^{\mathcal{I}_1} = B, r^{\mathcal{I}_1} = f, s^{\mathcal{I}_1} = t, u^{\mathcal{I}_1} = N.$$

通过把 $u^{\mathcal{I}_1}$ 从 N 改为 t , 得到了 K 的一个三值模型 \mathcal{I}'_1 :

$$p^{\mathcal{I}'_1} = t, q^{\mathcal{I}'_1} = B, r^{\mathcal{I}'_1} = f, s^{\mathcal{I}'_1} = t, u^{\mathcal{I}'_1} = t.$$

易见, $ID_4(K, \mathcal{I}_1) = ID_3(K, \mathcal{I}'_1)$ 。

命题3.3. 令 K 是一个命题知识库, 则 $ID_{LP_m}(K) = ID_3(K)$ 。

证明. (1) $ID_3(K) \leq ID_{LP_m}(K)$ 。

因为 LP_m 模型一定也是三值模型, 所以易见 $ID_3(K) \leq ID_{LP_m}(K)$ 。

(2) $ID_3(K) \geq ID_{LP_m}(K)$ 。

反设 $ID_3(K) < ID_{LP_m}(K)$, 那么一定存在两个三值解释 $\mathcal{I}_0, \mathcal{J}_0$, 使得 $\mathcal{I}_0 \models_3 K$, $\mathcal{J}_0 \models_{LP_m} K$, $ID_3(K, \mathcal{I}_0) = ID_3(K)$ 且 $ID_{LP_m}(K, \mathcal{J}_0) = ID_{LP_m}(K)$, 所以 $ID_3(K, \mathcal{I}_0) < ID_{LP_m}(K, \mathcal{J}_0)$ 。于是 $|\{p \mid p^{\mathcal{I}_0} = B\}| < |\{p \mid p^{\mathcal{J}_0} = B\}|$, 并且根据 \mathcal{I}_0 的定义, 没有其他的三值模型含有更少的矛盾, 所以 \mathcal{I}_0 是 K 的一个 LP_m 模型。这样 $ID_{LP_m}(K, \mathcal{I}_0) < ID_{LP_m}(K, \mathcal{J}_0)$, 与假设矛盾。 \square

例3.4. $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$ 。再次考虑例3.2中定义三值模型 \mathcal{I}'_1 。则 \mathcal{I}'_1 也是 K 的一个 LP_m 模型, 因为 $\mathcal{I}'_1 \models_3 K$ 而且其他的三值解释 \mathcal{J} 满足 $\{p \mid p^{\mathcal{J}} = B\} \subseteq \{p \mid p^{\mathcal{I}'_1} = B\}$ 只能是经典的二值解释, 这是不可能满足 K 的。所以 $ID_{LP_m}(K) = \min_{\mathcal{I} \models_{LP_m} K} ID_{LP_m}(K, \mathcal{I}) = ID_{LP_m}(K, \mathcal{I}'_1) = 1/5 = ID_3(K)$ 。

命题3.5. $ID_4(K) \leq ID_Q(K)$ 。

证明. 因为每个准经典模型一定也是四值模型, 所以结论显然成立。 \square

在例2.6中, 我们已经看到 $ID_4(K) = \frac{1}{5} < \frac{3}{5} = ID_Q(K)$ 。所以 $ID_4(K)$ 可以严格小于 $ID_Q(K)$ 。

综合上面的讨论, 根据命题3.1, 3.3, 和3.5, 我们可以得到下面的定理:

定理3.6. 对于给定的命题知识库 K , $ID_3(K) = ID_{LP_m}(K) = ID_4(K) \leq ID_Q(K)$ 。

3.2 不一致度量问题的复杂度分析

不针对一般的算法, 我们来研究不一致度量问题本身的复杂度如何。在[45]中, 我们已经讨论了 ID_4 相关的复杂度分析, 这些结果可以并行的扩展到其他度量上。

首先我们来定义在 i -语义($i = 3, 4, LP_m, Q$)下的不一致度计算问题:

- $ID_{i,\leq d}$ (相应的 $ID_{i,<d}, ID_{i,\geq d}, ID_{i,>d}$): 给定一个命题知识库 K 和一个数 $d \in [0, 1]$, 判断 $ID_i(K) \leq d$ (相应的 $ID_i(K) < d, ID_i(K) \geq d, ID_i(K) > d$) 是否成立。
- EXACT- ID_i : 给定知识库 K 和一个数 $d \in [0, 1]$, 判断 $ID_i(K) = d$ 是否成立
- ID_i : 给定知识库 K , 计算 $ID_i(K)$ 。

显然, 有两个肯定的平凡实例 $ID_{i,\leq 1}, ID_{i,\geq 0}$ 和两个否定的平凡实例 $ID_{i,<0}, ID_{i,>1}$ 。

对于一般的情况, 下面的几个定理说明了这些问题的复杂度。

定理3.7. $ID_{i,\leq d}$ 和 $ID_{i,<d}$ ($i = 3, 4, LP_m, Q$) 是 **NP**-完全的; $ID_{i,\geq d}$ 和 $ID_{i,>d}$ ($i = 3, 4, LP_m, Q$) 是 **coNP**-完全的。

证明. 我们分别来证明.、。

$ID_{i,\leq d}$ 是 **NP-完全的:**

证明 $ID_{i,\leq d}$ ($i = 3, 4, Q$) 是 **NP** 问题, 可以使用下面的不确定算法:

1. 猜一个 $Var(K)$ 上的 i -解释 \mathcal{I} ;
2. 检查 \mathcal{I} 是 K 的一个 i -模型, 并且 $\frac{|Conflict(\mathcal{I})|}{|Var(K)|} \leq d$ 。这可以在确定的多项式时间内完成。

$ID_{LP_m,\leq d}$ 是 **NP** 问题, 由定理3.6 的结论 $ID_{LP_m}(K) = ID_4(K)$ 可得。

NP-难的部分可以把由经典二值语义下的可满足性归约到我们的问题。归约为 K 是经典二值可满足的，当且仅当 $ID_i(K) \leq 0$ 成立。由 ID_i 的定义，这个归约的正确性是显然的。由于二值可满足是**NP**-完全的，所以 $ID_{i,\leq d}$ 是**NP**-难的。

$ID_{i,<d}$ 是NP**-完全的:**

跟 $ID_{i,\leq d}$ 的类似, $ID_{i,<d}$ 是**NP**问题是显然成立的。 **NP**-难的部分也可以由二值可满足的归约得到。 K 是二值可满足的，当且仅当 $ID_i(K) < \frac{1}{2\lceil \text{Var}(K) \rceil}$ ，根据 $ID_i(K)$ 的定义, 对于一个不一致的知识库, $ID_i(K)$ 的最小可能值为 $\frac{1}{\lceil \text{Var}(K) \rceil}$ 。

$ID_{i,\geq d}$ 和 $ID_{i,>d}$ 是coNP**-完全的:**

这是因为 $ID_{i,\geq d}$ 是 $ID_{i,<d}$ 的互补问题, $ID_{i,>d}$ 是 $ID_{i,\leq d}$ 的互补问题。

□

定理3.8. $EXACT-ID_i$ ($i = 3, 4, LP_m, Q$) 是**DP**-完全的¹。

证明. 为了证明在 $EXACT-ID_i$ 是**DP**问题, 必须要构造两个语言 $L_1 \in \mathbf{NP}$ 和 $L_2 \in \mathbf{coNP}$, 使得所有的 $EXACT-ID_i$ 的成真实例都在恰好在 $L_1 \cap L_2$ 中。这只需要令 $L_1 = \{K \mid ID_i(K) \leq d\}$ 和 $L_2 = \{K \mid ID_i(K) \geq d\}$ 即可。

为了证明完全性, 令 $L = L_1 \cap L_2$ 为**DP**中的任意一个语言。我们必须证明 L 可以归约到 $EXACT-ID_i$ 。为此, 由于 $ID_{i,\leq}$ 是**NP**-完全的, $ID_{i,\geq}$ 是**coNP**-完全的, 存在从 L_1 到 $ID_{i,\leq}$ 的归约 R_1 和 L_2 到 $ID_{i,\geq}$ 的归约 R_2 。因此, 从 L 到 $EXACT-ID_i$ 的归约 R 可以定义为, 对任意的一个实例 x : $R(x) = (R_1(x), R_2(x))$ 。 $R(x)$ 是 $EXACT-ID_i$ 的肯定实例, 当且仅当 $R_1(x)$ 是 $ID_{i,\leq}$ 的肯定实例, 且 $R_2(x)$ 是 $ID_{i,\geq}$ 的肯定实例, 也就等价于 $x \in L$ 。 □

由于 $ID_4(K) = ID_3(K) = ID_{LP_m}(K)$, 计算 ID_4 的复杂度[45]可以扩展为下面的定理。

定理3.9. ID_i ($i=3, 4, LP_m$) 是**FP^{NP[log n]}**-完全的²。

¹一个语言 L 在**DP** [56]中当且仅当有两个语言 $L_1 \in \mathbf{NP}$ 和 $L_2 \in \mathbf{coNP}$ 使得 $L = L_1 \cap L_2$ 。

²复杂度**FP^{NP[log n]}**定义为所有的可以由一个多项式Oracle SAT图灵机来判定, 而且其查询SAT的次数为 $\mathcal{O}(\log |x|)$ 。 **FP^{NP[log n]}**是对应的函数复杂度类。

ID_Q 的函数复杂度还没有完全解决。尽管如此, 由于Exact- ID_Q 是DP-完全的3.8, 我们可以确定这个问题也是难解的, 并且与复杂度与 ID_4 相同。

3.3 归约算法

在前面的一节复杂度分析中, 已经看到计算不一致度通常是难解的。在这一节中, 我们给出两个新颖的算法, 可以把计算不一致度量的问题归约为部分Max-SAT问题, 这样我们就可以充分利用现有的部分Max-SAT求解器的性能。

本节中, 不失一般性, 假定所有的知识库都是CNF的, 即一个子句集, 因为所有的知识库都可以等价地变换到CNF形式。根据定理3.6, 我们只需考虑 ID_4 和 ID_Q 的计算。

3.3.1 归约基于四值语义的度量到部分极大可满足性问题

给定一个知识库 $K = \{\gamma_i \mid i = 1, \dots, n\}$, 在 K 上的四值推理可以用在 $4(K)$ 上的二值推理来模拟, 其定义如下[57]:

定义3.10. 给定一个知识库 $K = \{\gamma_i \mid i = 1, \dots, n\}$, 命题变元 $p \in \text{Var}(K)$, $4(\cdot)$ 是一个从子句(集)到子句(集)的映射, 递归定义如下:

$$\begin{aligned} 4(\{\gamma_1, \gamma_2, \dots, \gamma_n\}) &= \{4(\gamma_1), 4(\gamma_2), \dots, 4(\gamma_n)\}; \\ 4(l_1 \vee \dots \vee l_k) &= 4(l_1) \vee \dots \vee 4(l_k); \\ 4(p) &= +p; \\ 4(\neg p) &= -p. \end{aligned}$$

在这里, $4(K)$ 是变元集 $\{+p, -p \mid p \in \text{Var}(K)\}$ 上的知识库。显然, 从 K 计算 $4(K)$ 可以在线性时间内完成。

一个变元集 \mathcal{A} 上的四值解释 \mathcal{I} 也可以视为 \mathcal{A}^\pm 上的二值解释, 其对应关系为:

$$\begin{aligned}
p^{\mathcal{I}} = B & \text{ iff } +p^{\mathcal{I}} = t \text{ 且 } -p^{\mathcal{I}} = t; \\
p^{\mathcal{I}} = f & \text{ iff } +p^{\mathcal{I}} = f \text{ 且 } -p^{\mathcal{I}} = t; \\
p^{\mathcal{I}} = t & \text{ iff } +p^{\mathcal{I}} = t \text{ 且 } -p^{\mathcal{I}} = f; \\
p^{\mathcal{I}} = N & \text{ iff } +p^{\mathcal{I}} = f \text{ 且 } -p^{\mathcal{I}} = f.
\end{aligned}$$

在下文中，我们会同时使用这两种看法。

定理3.11. [57] 给定一个知识库 K 和一个四值解释 \mathcal{I} ，我们有 $\mathcal{I} \models_4 K$ 当且仅当 $\mathcal{I} \models 4(K)$ 。

例3.12. 令 $K = \{\neg p, p \vee q, \neg q, r\}$ ，则 $4(K) = \{-p, +p \vee +q, -q, +r\}$ 。考虑解释 $\mathcal{I}_1 = \{+p, -p, -q, +r\}$ 。它可以看作是一个 $\{p, q, r\}$ 上的四值解释：

$$p^{\mathcal{I}_1} = B, q^{\mathcal{I}_1} = f, r^{\mathcal{I}_1} = t.$$

同时 \mathcal{I}_1 也可以看作 $\{+p, -p, +q, -q, +r, -r\}$ 上的二值解释：

$$\begin{aligned}
(+p)^{\mathcal{I}_1} = t, (-p)^{\mathcal{I}_1} = t, (+q)^{\mathcal{I}_1} = f, \\
(-q)^{\mathcal{I}_1} = t, (+r)^{\mathcal{I}_1} = t, (-r)^{\mathcal{I}_1} = f.
\end{aligned}$$

易见 $\mathcal{I}_1 \models_4 K$ 且 $\mathcal{I}_1 \models 4(K)$ 。

推论3.13. 给定 \mathcal{A} 上的知识库 K ，则 K 在四值语义下的不一致度可以用 \mathcal{A}^+ 上的二值语义来计算：

$$\begin{aligned}
ID_4(K, \mathcal{I}) &= \frac{|\{p \mid p \in \text{Var}(K), +p^{\mathcal{I}} = t \wedge -p^{\mathcal{I}} = t\}|}{|\text{Var}(K)|}; \\
ID_4(K) &= \min_{\mathcal{I} \models 4(K)} ID_4(K, \mathcal{I}).
\end{aligned}$$

注意到

$$\begin{aligned}
& \min_{\mathcal{I} \models 4(K)} |\{p \mid p \in \text{Var}(K), +p^{\mathcal{I}} = t \wedge -p^{\mathcal{I}} = t\}| \\
&= \min_{\mathcal{I} \models 4(K)} |\{p \mid p \in \text{Var}(K), (\neg +p \vee \neg -p)^{\mathcal{I}} = f\}| \\
&= \max_{\mathcal{I} \models 4(K)} |\{p \mid p \in \text{Var}(K), (\neg +p \vee \neg -p)^{\mathcal{I}} = t\}|.
\end{aligned}$$

这提示我们可以使用部分Max-SAT问题求解器来计算基于四值语义的不一致度。考虑下面的部分Max-SAT问题实例：

定义3.14. 给定知识库 $K = \{\gamma_1, \dots, \gamma_n\}$, $\text{Var}(K) = \{p_1, \dots, p_m\}$, 则相对应的计算基于四值语义的不一致度 $\text{ID}_4(K)$ 的部分Max-SAT问题, 记作 $P_4(K) = (H_4(K), S_4(K))$, 定义为:

$$H_4(K) = \{4(\gamma) \mid \gamma \in K\};$$

$$S_4(K) = \{\neg + p \vee \neg - p \mid p \in \text{Var}(K)\}.$$

推论3.15. 若 \mathcal{I} 是部分Max-SAT问题 $P_4(K)$ 的一个最优解。令 $b(K, \mathcal{I}) = |\{p \mid +p^{\mathcal{I}} = t \wedge -p^{\mathcal{I}} = t\}|$, $m(K, \mathcal{I}) = |\text{Var}(K)|$ 。则 $\text{ID}_4(K) = b(K, \mathcal{I})/m(K)$ 。

证明. 根据定义2.1 和事实 $p^{\mathcal{I}} = B$ iff $+p^{\mathcal{I}} = t$ 且 $-p^{\mathcal{I}} = t$, 这个推论显然成立。 \square

由推论3.15, 可以得到下面的算法。

Algorithm 1 使用部分Max-SAT求解器计算 ID_4

```

1: procedure  $\text{ID}_4(K)$ 
2:    $P \leftarrow \{\}$ 
3:    $m \leftarrow |\text{Var}(K)|$ 
4:   for all Clause  $\gamma \in K$  do
5:      $P.addHardClause(4(\gamma))$ 
6:   end for
7:   for all Variable  $p \in \text{Var}(K)$  do
8:      $P.addSoftClause(\neg + p \vee \neg - p)$ 
9:   end for
10:   $\mathcal{I} \leftarrow \text{PartialMaxSATSolver}(P)$ 
11:   $b = |\{p \mid +p^{\mathcal{I}} = t \wedge -p^{\mathcal{I}} = t\}|$ 
12:  return  $b/m$ 
13: end procedure

```

算法首先在每4行到第9行, 生成 $P_4(K)$, 然后在第10行调用一个部分Max-SAT 求解器得到一个最优解。最后在每11到12行, 通过定理3.15, 得到最终的不一致度。

推论3.16 (算法1 的正确性). 对于给定的知识库 K , 算法1 对于计算基于四值语义的度量是可靠完全的。即, $Algorithm1(K) = ID_4(K)$, 其中 $Algorithm1(K)$ 是算法1 对输入 K 的输出。

证明. 由推论3.15, 结论易见。 □

下面的例子进一步解释了算法1。

例3.17. 令 $K = \{p \vee q, \neg p, \neg q, r\}$, 则 $4(K) = \{+p \vee +q, -p, -q, +r\}$ 。然后, 根据定义3.14, $P_4(K) = (H_4(K), S_4(K))$, 其中

$$\begin{aligned} H_4(K) &= \{+p \vee +q, -p, -q, +r\}, \\ S_4(K) &= \{\neg +p \vee \neg -p, \neg +q \vee \neg -q, \neg +r \vee \neg -r\}. \end{aligned}$$

对于 $P_4(K)$, 通过调用部分Max-SAT 问题求解器, 我们可以得到一个最优解 \mathcal{I}_0 :

$$\begin{aligned} +p^{\mathcal{I}_0} &= t, -p^{\mathcal{I}_0} = t, +q^{\mathcal{I}_0} = f, \\ -q^{\mathcal{I}_0} &= t, +r^{\mathcal{I}_0} = t, -r^{\mathcal{I}_0} = f. \end{aligned}$$

对应的 K 的四值模型为 \mathcal{I}_0 :

$$p^{\mathcal{I}_0} = B, q^{\mathcal{I}_0} = f, r^{\mathcal{I}_0} = t,$$

于是由算法1可知 $ID_4(K) = 1/3$ 。

3.3.2 归约基于准经典语义的度量到部分极大可满足性问题

在第3.1节中, 已经看到基于准经典语义的度量与基于四值语义的度量是不同的。在本节中, 我们来研究计算基于准经典语义的度量。

首先, 与四值语义相同, 准经典语义下的推理也可以归约到二值语义。

为了简化符号, 对于每一个文字 l , 我们定义:

$$\begin{aligned} +l &= +p \quad \text{if } l = p, & +l &= -p \quad \text{if } l = \neg p, \\ -l &= -p \quad \text{if } l = p, & -l &= +p \quad \text{if } l = \neg p. \end{aligned}$$

定义3.18 (准经典变换[58]). 给定一个CNF的知识库 $K = \{\gamma_1, \dots, \gamma_n\}$, K 的准经典变的定义如下:

$$\begin{aligned} Q(\{\gamma_1, \dots, \gamma_n\}) &= \{Q(\gamma_1), \dots, Q(\gamma_n)\}, \\ Q(l_1 \vee \dots \vee l_n) &= \bigvee_{i=1}^n (+l_i \wedge \neg -l_i) \vee \bigwedge_{i=1}^n (+l_i \wedge -l_i). \end{aligned}$$

定理3.19. [58] 令 K 是一个命题知识库, \mathcal{I} 是一个准经典解释。则

$$\mathcal{I} \models_Q K \quad \text{当且仅当} \quad \mathcal{I} \models Q(K).$$

例3.20. 令 $K = \{\neg p, p \vee q, \neg q, r\}$ 。则 $4(K) = \{-p, +p \vee +q, -q, +r\}$, 而 $Q(K) = \{-p, (+p \wedge \neg -p) \vee (q \wedge \neg -q) \vee (+p \wedge -p \wedge +q \wedge -q), -q, +r\}$ 。这说明 $4(K)$ 通常于 $Q(K)$ 不同的。

按照下面的推论, 可以用经典语义来计算 ID_Q 。

推论3.21. 给定命题变元集 \mathcal{A} 上的知识库 K , 则 K 在准经典语义下的不一致可以在一个 \mathcal{A}^+ 上的知识库用经典语义计算:

$$\begin{aligned} ID_Q(K, \mathcal{I}) &= \frac{|\{p \in \text{Var}(K) \mid +p^{\mathcal{I}} = t \wedge -p^{\mathcal{I}} = t\}|}{|\text{Var}(K)|}; \\ ID_Q(K) &= \min_{\mathcal{I} \models Q(K)} ID_Q(K, \mathcal{I}). \end{aligned}$$

证明. 同推论3.15的证明。 \square

与4(\cdot) 相比, 变换 $Q(\cdot)$ 不能保持CNF的形式。所以 $Q(l_1 \vee \dots \vee l_n)$ 不能直接作为用于部分Max-SAT求解器的输入。另一方面, 直接使用分配律将 $Q(l_1 \vee \dots \vee l_n)$ 变换为CNF, 生成的公式大小是原公式的指数倍。为了解决这个问题, 我们采用了[59]中的技术, 通过引入新的变元来保持可满足性:

$$\begin{aligned} y_i &:= +l_i \wedge \neg -l_i, i = 1, \dots, n; \\ z &:= \bigwedge_{i=1}^n (+l_i \wedge -l_i). \end{aligned}$$

这样, 我们就可以定义变换 $Q'(\cdot)$:

$$\begin{aligned} Q'(\{\gamma_1, \dots, \gamma_n\}) &= \{Q'(\gamma_1), \dots, Q'(\gamma_n)\} \\ Q'(l_1 \vee \dots \vee l_n) &= \left(\bigvee_{i=1}^n y_i \vee z \right) \wedge \bigwedge_{i=1}^n (\neg y_i \vee +l_i) \\ &\quad \wedge \bigwedge_{i=1}^n (\neg y_i \vee \neg -l_i) \\ &\quad \wedge \bigwedge_{i=1}^n (\neg z \vee +l_i) \wedge \bigwedge_{i=1}^n (\neg z \vee -l_i). \end{aligned}$$

$Q'(\cdot)$ 把每一个长度为 n 的子句变换为了 $4n + 1$ 个子句。易见, $Q'(p) \equiv +p$ 且 $Q'(\neg p) \equiv -p$.

下面的命题说明我们可以在 $Q'(K)$ 不仅保持了可满足性, 而且可以用来计算 $ID_Q(K)$:

命题3.22. 给定命题知识库 K , 我们有

$$ID_Q(K) = \frac{\min_{I \models Q'(K)} |\{p \in Var(K) \mid +p^I = t, -p^I = t\}|}{|Var(K)|}.$$

证明. 命题变元集 $\{+p, -p \mid p \in \text{Var}(K)\}$ 上的一个解释 \mathcal{I} , 使得 $\mathcal{I} \models Q(K)$, 我们可以把 \mathcal{I} 扩展为 \mathcal{I}' , 定义在变元集 $\{+p, -p \mid p \in \text{Var}(K)\} \cup \{y_i\} \cup \{z\}$ 上, 使得 $\mathcal{I}' \models Q'(K)$:

$$\begin{aligned} y_i^{\mathcal{I}'} &= (+l_i \wedge \neg -l_i)^{\mathcal{I}}, i = 1, \dots, n; \\ z^{\mathcal{I}'} &= (\wedge_{i=1}^n (+l_i \wedge -l_i))^{\mathcal{I}}. \end{aligned}$$

另一方面, 如果 $\mathcal{I} \models Q'(K)$, 那么 \mathcal{I} 也可以看作 $Q(K)$ 的一个解释, 而且有 $\mathcal{I} \models Q(K)$ 。

所以

$$\begin{aligned} &\{p \mid p \in \text{Var}(K), +p^{\mathcal{I}} = t, -p^{\mathcal{I}} = t, \mathcal{I} \models Q'(K)\} \\ &= \{p \mid p \in \text{Var}(K), +p^{\mathcal{I}} = t, -p^{\mathcal{I}} = t, \mathcal{I} \models Q(K)\} \end{aligned}$$

由推论3.21 知, 结论成立。 \square

定义3.23. 给定命题知识库 $K = \{\gamma_1, \dots, \gamma_n\}$, 其对应的用于计算 ID_Q 的部分 Max-SAT 问题 $P_Q(K) = (H_Q(K), S_Q(K))$ 定义如下:

$$\begin{aligned} H_Q(K) &= \{Q'(\gamma) \mid \gamma \in K\}; \\ S_Q(K) &= \{\neg +p \vee \neg -p \mid p \in \text{Var}(K)\}. \end{aligned}$$

与推论3.15类似, 下面的推论把计算基于准经典语义的不确定的度量归约到了部分 Max-SAT 问题。

推论3.24. 给定知识库 K , 若 \mathcal{I} 是部分 Max-SAT 问题 $P_Q(K)$ 的一个最优解。令 $b(K, \mathcal{I}) = |\{p \mid +p^{\mathcal{I}} = t \wedge -p^{\mathcal{I}} = t\}|$, $m(K) = |\text{Var}(K)|$, 则 $\text{ID}_Q(K) = b(K, \mathcal{I})/m(K)$ 。

证明. 同推论3.15的证明。 \square

例3.25. 令 $K = \{\neg p, p \vee q, \neg q, r\}$. 则 $P_Q(K) = (H_Q(K), S_Q(K))$, $H_Q(K) = Q'(K) = \{Q'(\neg p), Q'(p \vee q), Q'(\neg q), Q'(r)\}$, 其中 $Q'(\neg p) = -p$, $Q'(\neg q) = -q$, $Q'(r) = +r$,

$$\begin{aligned} Q'(p \vee q) &= (y_p \vee y_q \vee z) \wedge (\neg y_p \vee +p) \wedge (\neg y_p \vee \neg -p) \\ &\quad \wedge (\neg y_q \vee +q) \wedge (\neg y_q \vee \neg -q) \wedge (\neg z \vee +p) \\ &\quad \wedge (\neg z \vee -p) \wedge (\neg z \vee +q) \wedge (\neg z \vee -q). \end{aligned}$$

而 $S_Q(K) = \{\neg + p \vee \neg - p, \neg + q \vee \neg - q, \neg + r \vee \neg - r\}$ 。

$P_Q(K)$ 的一个最优解 \mathcal{I}_0 为：

$$\begin{aligned} +p^{\mathcal{I}_0} = t, -p^{\mathcal{I}_0} = t, +q^{\mathcal{I}_0} = t, -q^{\mathcal{I}_0} = t, +r^{\mathcal{I}_0} = t, -r^{\mathcal{I}_0} = f, \\ y_p^{\mathcal{I}_0} = f, y_q^{\mathcal{I}_0} = f, z^{\mathcal{I}_0} = t. \end{aligned}$$

所以由推论3.24, 得 $ID_Q(K) = \frac{2}{3}$ 。

由推论3.24, 可以得到算法2。其中命题原子 y_i 和 z 是由变换 $Q'(\cdot)$ 引入的。

推论3.26 (算法2的正确性). 给定知识库 K , 对于计算基于准经典语义的度量, 算法2是可靠完全的. 即 $Algorithm2(K) = ID_Q(K)$, 其中 $Algorithm2(K)$ 为算法2对于输入 K 的输出。

证明. 由推论3.24 易得. □

3.4 相关工作

基于多值语义的不一致度量, 已经有了很多工作, 本文中用到的度量来自于文献[40, 36, 37, 46, 34, 41, 60]。这些文献都是提出了某种度量或者是讨论了某种度量的应用, 但是并没有一个工作系统地分析不同的度量之间的数量关系。在本文中, 我们发现基于三值语义、四值语义和极小不一致语义的度量其实是等价的, 同时不同于基于准经典语义的度量。这一发现将上面提到的很多工作可以联系在一起。

关于基于多值语义的不一致度的计算, 相关的工作并不多。[34, 45]中给出了基于四值语义的不一致度量的算法和问题复杂度分析, 但是其中给出算法的效果并不理想。其中中的算法相当于一个搜索算法, 在一个非常大的搜索空间内寻找目标。而设计一个有效的搜索策略是不容易的。本文使用另外一个思路, 提出的归约算法可以把不一致度量计算问题归约为一个极大部分可满足问题, 这样就可

Algorithm 2 用部分Max-SAT求解器来计算 ID_Q

```

1: procedure  $ID_Q(K)$ 
2:    $P \leftarrow \{\}$ 
3:    $m \leftarrow |Var(K)|$ 
4:   for all Clause  $\gamma = \{l_1, \dots, l_n\} \in K$  do
5:      $P.addHardClause(y_1 \vee \dots \vee y_n \vee z)$ 
6:     for  $i = 1$  to  $n$  do
7:        $P.addHardClause(\neg y_i \vee +l_i)$ 
8:        $P.addHardClause(\neg y_i \vee \neg -l_i)$ 
9:        $P.addHardClause(\neg z \vee +l_i)$ 
10:       $P.addHardClause(\neg z \vee -l_i)$ 
11:     end for
12:   end for
13:   for all  $p \in Var(K)$  do
14:      $P.addSoftClause(\neg +p \vee \neg -p)$ 
15:   end for
16:    $\mathcal{I} \leftarrow \text{PartialMaxSATSolver}(P)$ 
17:    $b = |\{p \mid +p^{\mathcal{I}} = t \wedge -p^{\mathcal{I}} = t\}|$ 
18:   return  $b/m$ 
19: end procedure

```

以利用现有的多种高效的极大部分可满足问题的求解器，于是本文提出的算法可以更有效地处理大规模的输入。

本文的工作是针对于命题逻辑的。文献[41]中讨论了定义在一阶逻辑上的不一致度量，但是这个度量不是定义为一个实数，而是定义为了一个序列，这样便很难应用，更无法谈及算法和复杂度。我们可以考虑研究一阶逻辑的某个子集，在其上仍然可以定义不一致度量为一个实数，而且可以使用类似于本文的方法来计算。文献[60]已经在 DL-Lite 上定义了一个不一致度量。

第四章 算法实现和测评

4.1 算法实现

我们实现了第3.3节的提出两个算法，程序的名字称为 IDM (Inconsistency Degree Measurer)。

4.1.1 程序基本结构

IDM 程序主要有两个部分：

- (1) 把一个CNF知识库进行 $\mathcal{I}(\cdot)$ 和 $\mathcal{Q}(\cdot)$ 变换。这一部分是用Java语言来完成的，使用的版本为jdk 1.6¹。对CNF文件的解析使用了SAT4j MaxSAT [52]类库中的功能。
- (2) 调用一个外部的部分Max-SAT求解器来计算不一致度。这一部分使用Python语言来完成，版本为2.6²。Python的脚本调用部分Max-SAT求解器，并计算出不一致度量。目前使用的求解器有SAT4j MaxSAT [52], MsUncore [53] 和Clone [54]这三个，可以很容易地扩展使用其他的求解器上。

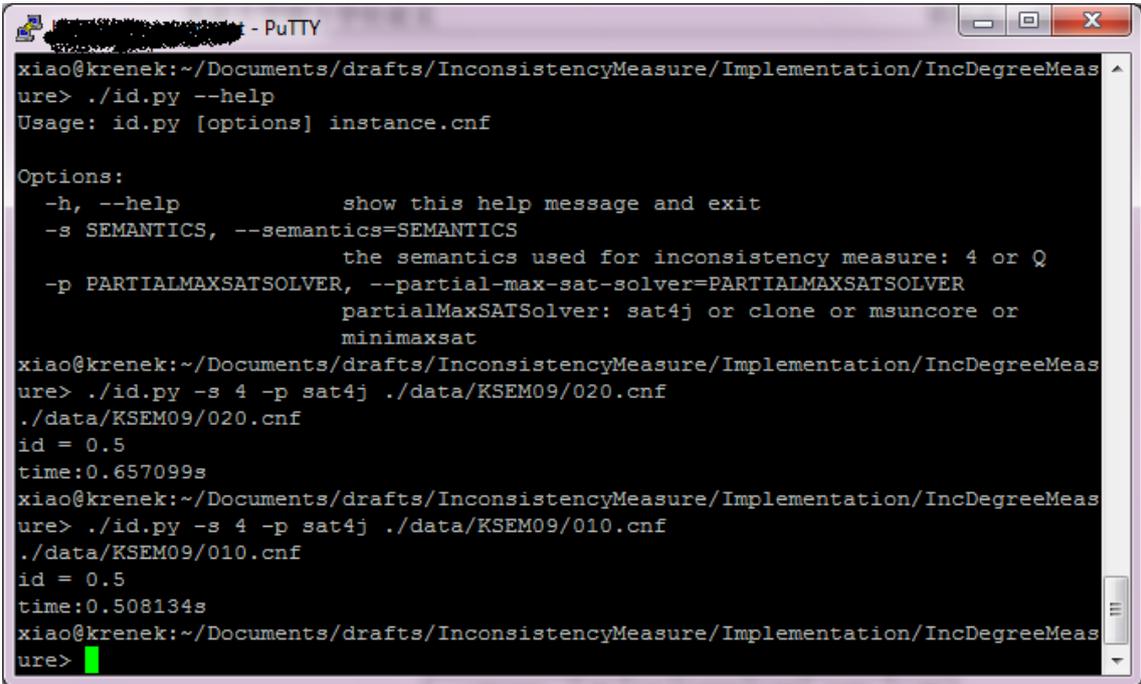
主程序脚本id.py 有三个参数：

- (1) `--semantics`, 简写作`-s`。选择使用的语义。可选项：`4`和`Q`，分别表示四值语义和准经典语义，默认为`4`；
- (2) `--partial-max-sat-solver`, 简写作`-p`。选择使用的部分Max-SAT求解器。可选项：`sat4j`, `clone` 和`msuncore`。默认为`sat4j`。
- (3) 输入文件名，使用 Dimacs 格式[50]，不可省略。

图4.1是程序的一个演示。

¹<http://java.sun.com/>

²<http://www.python.org/>



```

xiao@krenek:~/Documents/drafts/InconsistencyMeasure/Implementation/IncDegreeMeas
ure> ./id.py --help
Usage: id.py [options] instance.cnf

Options:
  -h, --help                show this help message and exit
  -s SEMANTICS, --semantics=SEMANTICS
                           the semantics used for inconsistency measure: 4 or Q
  -p PARTIALMAXSATSOLVER, --partial-max-sat-solver=PARTIALMAXSATSOLVER
                           partialMaxSATsolver: sat4j or clone or msuncore or
                           minimaxsat
xiao@krenek:~/Documents/drafts/InconsistencyMeasure/Implementation/IncDegreeMeas
ure> ./id.py -s 4 -p sat4j ./data/KSEM09/020.cnf
./data/KSEM09/020.cnf
id = 0.5
time:0.657099s
xiao@krenek:~/Documents/drafts/InconsistencyMeasure/Implementation/IncDegreeMeas
ure> ./id.py -s 4 -p sat4j ./data/KSEM09/010.cnf
./data/KSEM09/010.cnf
id = 0.5
time:0.508134s
xiao@krenek:~/Documents/drafts/InconsistencyMeasure/Implementation/IncDegreeMeas
ure>

```

图 4.1: 程序演示

4.1.2 归约不一致度量问题为部分极大可满足问题

变换模块的功能是把输入的 CNF 文件按照算法1或算法2进行相应的变换，转化为部分极大可满足问题，并保存在 WCNF 文件中。

其中要特别注意对于算法1和2中新引入的变元的编码。在本程序中，对一个变元 p ，如果它的编码为 i ，其相应的在变换后的 WCNF 文件中，新引入的变元的编码定义如下：

$$y_p \rightarrow 4i - 3$$

$$y_{\neg p} \rightarrow 4i - 2$$

$$+p \rightarrow 4i - 1$$

$$-p \rightarrow 4i$$

$$z_k \rightarrow 4 \times nVars + k$$

例4.1. 继续考虑例3.17中的实例 $K = \{p \vee q, \neg p, \neg q, r\}$ 。把 p, q, r 分别编码为 1, 2, 3, 则输入的 *CNF* 文件为:

```
p cnf 3 4
1 2 0
-1 0
-2 0
3 0
```

经过 $4(\cdot)$ 变换, 变换为 $P_4(K) = (H_4(K), S_4(K))$, 其中

$$H_4(K) = \{+p \vee +q, -p, -q, +r\},$$

$$S_4(K) = \{\neg +p \vee \neg -p, \neg +q \vee \neg -q, \neg +r \vee \neg -r\}.$$

表示为以下 *WCNF* 文件:

```
c A temporal KB for measuring ID
p wcnf 19 7 4
4 3 7 0
4 4 0
4 8 0
4 11 0
1 -3 -4 0
1 -7 -8 0
1 -11 -12 0
```

如果使用 $Q'(\dots)$ 变换, 则 $P_Q(K) = (H_Q(K), S_Q(K))$, $H_Q(K) = Q'(K) = \{Q'(\neg p), Q'(p \vee q), Q'(\neg q), Q'(r)\}$, 其中 $Q'(\neg p) = -p$, $Q'(\neg q) = -q$, $Q'(r) = +r$,

$$\begin{aligned} Q'(p \vee q) = & (y_p \vee y_q \vee z) \wedge (\neg y_p \vee +p) \wedge (\neg y_p \vee \neg -p) \\ & \wedge (\neg y_q \vee +q) \wedge (\neg y_q \vee \neg -q) \wedge (\neg z \vee +p) \\ & \wedge (\neg z \vee -p) \wedge (\neg z \vee +q) \wedge (\neg z \vee -q). \end{aligned}$$

而 $S_Q(K) = \{\neg +p \vee \neg -p, \neg +q \vee \neg -q, \neg +r \vee \neg -r\}$ 。

可以编码为以下的 *WCNF* 文件:

```
c A temporal KB for measuring ID
p wcnf 39 27 4
4 1 5 13 0
4 -1 3 0
4 -1 -4 0
4 -13 3 0
4 -13 4 0
4 -5 7 0
4 -5 -8 0
4 -13 7 0
4 -13 8 0
4 2 14 0
4 -2 4 0
4 -2 -3 0
4 -14 4 0
4 -14 3 0
4 6 15 0
4 -6 8 0
4 -6 -7 0
4 -15 8 0
4 -15 7 0
4 9 16 0
4 -9 11 0
4 -9 -12 0
4 -16 11 0
4 -16 12 0
1 -3 -4 0
1 -7 -8 0
1 -11 -12 0
```

4.1.3 调用部分极大可满足问题求解器计算不一致度

这一部分使用Python的脚本分别调用部分Max-SAT求解器，并计算出不一致度量。目前使用的求解器有SAT4j MaxSAT [52], MsUncore [53] 和Clone [54]这三个，可以很容易地扩展使用其他的求解器上。

例4.2. 把例4.1中生成的WCNF文件，作为输入送入到部分Max-SAT求解器中。对于 $4(K)$ ，调用SAT4j，可以得到如下的输出：

```
c SAT4J: a SATisfiability library for Java (c) 2004-2008 Daniel Le Berre
c This is free software under the dual EPL/GNU LGPL licenses.
```

```

c See www.sat4j.org for details.
.....
.....
c solving .\data\my\m.p.wcnf
c reading problem ...
c ... done. Wall clock time 0.0080s.
.....
.....
s OPTIMUM FOUND
v -1 -2 -3 4 -5 -6 7 8 -9 -10 11 -12 -13 -14 -15 -16 -17 -18 -19 0
c objective function=1
c Total wall clock time (in seconds): 0.023

```

其中最重要的一行为以 v 开头，表示了最优解的集合。根据对应关系，我们关心其中的“-3 4 7 8 11 -12”，它们分别表示了 $+p, -p, +q, -q, +r, -r$ 的赋值，正为取真值，负为取假值。7,8都为正，表示 $+q, -q$ 都取真值，即对应着四值模型中的一个矛盾 $q^T = B$ 。由此可知 $ID_4(K) = 1/3$ 。

4.2 性能测评

为了说明的算法的性能，我们选择了一些测试集来进行测试。

这些实验都是在一台Intel Pentium 4 (3.00GHz) 机器上完成的，物理内存 1G，分配给Java虚拟机的内存为512M，操作系统为 OpenSuse。由于篇幅所限，只把部分测试结果显示在了表4.1，表4.2 和表4.3 中。所有的程序和测试数据都可以在网站下载³。

对于每一个实例，我们设置了时间限制为240秒。如果发生超时，在表格的相应位置上我们标记一个“*”。表格的各列含义分别为：

- “文件名”：测试数据实例的文件名；
- “#V”，“#C”：实例中变元和子句的个数；
- “ ID_4 ”，“ ID_Q ”：实例的基于四值语义的和基于准经典语义的不一致度量；

³<http://www.is.pku.edu.cn/~xgh/id/>

- “任意时间算法”: 文章[45]中的任意时间算法, 得到精确值的时间, 单位为秒;
- “归约算法”: 本文的归约算法在相应的部分 Max- SAT 求解器上的时间, 单位为秒。

表4.1比较了算法1 与[45]中的任意时间算法的性能。任意时间算法[45]可以计算(四值语义下的)不一致度量的上界和下界。如果有足够多的计算资源, 最终将收敛到准确值。

为了对比的需要, 我们使用了与[45]中相同的测试数据:

测试知识库 $K_N = \{p_i, q_j, \neg p_i \vee \neg q_j \mid 1 \leq i, j \leq N\}$, 这里 $N = 1, 2, 5, 7, 10, 20, 50, 100$ 。显然, $|Var(K_N)| = 2N$, $|K_N| = N^2 + 2N$ 。

从表4.1中可知, 当 $N > 10$ 时, 我们的归约算法性能要比任意时间算法有明显优势。同时注意到当 $N > 20$ 时, 任意时间算法已经在时间限制内不能得到精确解, 而我们的归约算法可以很轻松的处理。特别指出, 对于下面的表4.2和表4.3的测试, 任意时间算法无法处理任意一个实例。

实例				任意时间算法	归约算法		
文件名	#V	#C	ID_4		sat4j	msuncore	clone
001.cnf	2	3	0.500	0.001	0.351	0.016	0.566
002.cnf	4	8	0.500	0.003	0.351	0.016	0.571
005.cnf	10	35	0.500	0.268	0.365	0.016	0.635
007.cnf	14	63	0.500	4.477	0.360	0.017	0.732
010.cnf	20	120	0.500	228.754	0.353	0.018	0.960
020.cnf	40	440	0.500	*	0.457	0.031	1.396
050.cnf	100	2600	0.500	*	0.858	0.188	4.209
100.cnf	200	10200	0.500	*	3.513	1.570	17.876

表 4.1: 任意时间算法和归约算法的比较

表4.2给出了算法1 在两组测试数据集上的结果。

实例				归约算法		
文件名	#V	#C	ID ₄	sat4j	msuncore	clone
uuf50-0101	50	218	0.02000	0.396	0.026	1.119
uuf50-0102	50	218	0.02000	0.398	0.020	1.121
uuf50-0103	50	218	0.02000	0.450	0.044	1.142
uuf50-0104	50	218	0.02000	0.397	0.027	1.279
uuf75-011	75	325	0.01330	0.496	0.031	1.379
uuf75-012	75	325	0.01330	0.447	0.030	1.355
uuf75-013	75	325	0.01330	0.443	0.033	1.333
uuf75-014	75	325	0.01333	0.494	0.029	1.372
uuf100-0101	100	430	0.01000	0.545	0.045	1.748
uuf100-0102	100	430	0.01000	0.918	0.053	2.088
uuf100-0103	100	430	0.02000	3.951	2.592	*
C168_FW_SZ_107	1698	5401	0.00059	0.698	0.120	*
C168_FW_SZ_128	1698	5422	0.00059	0.601	0.090	13.191
C168_FW_SZ_41	1698	7489	0.00059	0.849	0.085	11.939

表 4.2: 用归约算法计算ID₄

A组的实例文件名都以“uuf”为前缀,来自于SAT测试数据集 SATLIB⁴。

B组的实例都以“C”为前缀,是一些大型的自动产品配置问题的不可满足数据库[61]。每一个实例都表示对一个产品的配置要求,其中也包含了各种约束。

由于篇幅所限,我们只列出了A组和B组的部分测试结果。

从表4.1可知,几乎所有的实例都能被所有的求解器处理。只有“uuf100-0103”和“C168_FW_SZ_107”这两个实例,在时间限制内,不能被基于Clone的算法1实现进行度量。

表4.3使用与表4.2相同的测试数据集,使用算法2来计算ID_Q。基于SAT4j的实现可以在1秒内处理A组的数据,在9-14秒内处理B组的所有实例。基于MsUncore的

⁴<http://www.satlib.org>

实例				归约算法		
文件名	#V	#C	ID_Q	sat4j	msuncore	clone
uuf50-0101	50	218	1.000	0.445	*	0.428
uuf50-0102	50	218	1.000	0.444	*	0.446
uuf50-0103	50	218	1.000	0.449	*	0.246
uuf50-0104	50	218	1.000	0.494	*	0.433
uuf75-011	75	325	1.000	0.544	*	0.434
uuf75-012	75	325	1.000	0.548	*	0.435
uuf75-013	75	325	1.000	0.455	*	1.338
uuf75-014	75	325	1.000	0.646	*	0.437
uuf100-0101	100	430	1.000	0.709	*	0.478
uuf100-0102	100	430	1.000	0.803	*	0.438
uuf100-0103	100	430	1.000	0.749	*	0.445
C168_FW_SZ_107	1698	5401	0.124	9.269	*	1.487
C168_FW_SZ_128	1698	5422	0.107	9.916	*	0.792
C168_FW_SZ_41	1698	7489	0.117	13.627	*	0.738

表 4.3: 用归约算法计算 ID_Q

实现居然不能处理两组中的任意一个实例。而基于Clone的实现可以在2秒内处理两组的任意实例。

根据以上的实验，基于这些测试数据，我们做出如下的分析。

- 对于大多数的测试数据, 我们的算法都可以在很短的时间内结束, 这说明了我们的方法的有效性。
- 两个算法实现的性能都和内部调用的部分Max-SAT求解器的性能有很大关系。例如, 在我们的实验中, 基于MsUnCore的实现在计算 ID_4 时是最快的, 但是, 再计算 ID_Q 时, 居然处理不了任何一个实例。基于Clone的实现在处理 ID_Q 时是最快的。与其他两个相比, 基于SAT4j的实现不是最快的, 但是可

以处理计算出所有实例的所有度量。

- 我们观察到 ID_4 和 ID_Q 的值通常都是不同的。哪一个更有用，要看具体的应用。在我们的实验中，我们发现计算 ID_4 比计算 ID_Q 更快，这可以通过算法2中使用的变换 $Q'(\cdot)$ 比算法1中实现的变换 $4(\cdot)$ 更复杂来解释。

第五章 总结和展望

5.1 总结

在不同文献中，基于不同的多值语义的不一致度量分别被提出，包括了三值语义、四值语义、极小不一致语义和准经典语义。我们首先研究基于不同的多值语义的度量之间的关系，结果发现，对于给定的知识库，基于三值、四值和 LP_m 语义的度量是等价的，而通常小于基于准经典语义的度量。这就意味着，今后我们只需要考虑 ID_4 和 ID_Q 就足够了。

我们给出了不一致度量计算相关问题的复杂度分析。基于之前的工作[45]和不同度量间的关系，我们把原有的 ID_4 相关复杂度结论扩展到 ID_3 、 ID_4 和 ID_Q 。这些结论说明了，不一致度量的计算的相关问题都是NP-难解的。

我们提出了一个线性时间的归约算法，可以把不一致度计算归约为已有的可满足问题。部分可满足问题（Partial Max-SAT）已经有了很多高效的求解器，而不一致度的计算可以很自然的归约到部分可满足问题，针对 ID_4 和 ID_Q ，我们分别给出了两个高效的归约算法。

使用现有的部分可满足问题求解器，我们实现了这两个算法，并选取了一些数据集进行测试。初步的实验结果说明了我们的方法可以有效度量大的知识库的不一致度，与原有的方法相比，有着明显的优势。

5.2 展望

我们计划把算法增加近似的能力，这样就可以在计算资源有限的情况时，得到最终结果的一个近似。根据SAT竞赛¹的规范,每当部分Max-SAT 求解器得到一个新的最优值时，它就应该及时将其输出。而这个中间结果恰好可以用来计算不一致度的一个上界。

¹www.maxsat.udl.cat/09/index.php?disp=requirements

同时，我们也可以归约到其它的问题来计算不一致度。例如可以归约到伪布尔问题（pseudo boolean problem, 也称0-1 整数规划问题）。在 ID_Q 的归约中，我们看到归约后的部分Max-SAT部分问题可能比原知识库的规模大很多。如果使用伪布尔问题，就可以更紧凑地编码。伪布尔问题已经有了很多成熟的求解器，例如BoolVar[62], PBCLASP[63], SAT4j[52]和SCIP [64]。

另外，由于现在已经存在着很多个部分Max-SAT求解器，而且不同的求解器对不同的实例的性能差别很大。所以我们可以训练一个元框架，使其对于特定的知识库和不一致度量，可以自动的选择最合适的求解器。

我们还考虑把这个方法应用到其他的逻辑系统中，例如描述逻辑（Description Logics）和逻辑程序（Logic Programming）。在描述逻辑上定义不一致度量有定义域选择的问题，如果不能解决这个问题，不一致度量就只能定义成序列。另一方面，很多描述逻辑,如 ALC 有有限模型的性质，于是可以考虑在某个有限模型上来定义不一致度，并进一步研究相关的计算问题。

参考文献

- [1] 林战刚. 缺省理论中的冲突处理. 博士学位论文, 北京大学, 2008.
- [2] 林作铨, 李未. 超协调逻辑的统一基础. 模式识别与人工智能, 8(3):210–216, 1995.
- [3] 林作铨, 李未. 超协调逻辑(I) — 传统超协调逻辑研究. 计算机科学, 21(5):1–8, 1994.
- [4] 林作铨, 李未. 超协调逻辑(II) — 新超协调逻辑研究. 计算机科学, 21(6):1–7, 1994.
- [5] 林作铨, 李未. 超协调逻辑(III) — 超协调性的逻辑基础. 计算机科学, 22(1):1–4, 1995.
- [6] 林作铨, 李未. 超协调逻辑(IV) — 非单调超协调逻辑研究. 计算机科学, 22(1):4–9, 1995.
- [7] da Costa and Newton C. A. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15(4):497–510, 1974.
- [8] Philippe Besnard and Anthony Hunter. Quasi-classical logic: Non-trivializable classical reasoning from inconsistent information. In Christine Froidevaux and Jürg Kohlas, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty, European Conference, ECSQARU'95, Fribourg, Switzerland, July 3-5, 1995, Proceedings*, volume 946 of *Lecture Notes in Computer Science*, pages 44–51. Springer, 1995.
- [9] Anthony Hunter. Reasoning with contradictory information using quasi-classical logic. *J. Log. Comput.*, 10(5):677–703, 2000.

- [10] Anthony Hunter. A semantic tableau version of first-order quasi-classical logic. In Salem Benferhat and Philippe Besnard, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 6th European Conference, ECSQARU 2001, Toulouse, France, September 19-21, 2001, Proceedings*, volume 2143 of *Lecture Notes in Computer Science*, pages 544–555. Springer, 2001.
- [11] Zuoquan Lin and Wei Li. On logic of paradox. In *Proceedings of the 25th IEEE International Symposium on Multipl-Valued Logic*, pages 248–255, 1995.
- [12] N. D. Belnap. A useful four-valued logic. In *Modern uses of multiple-valued logics*, pages 7–73. Reidel, 1977.
- [13] Nuel D. Belnap. How a computer should think. In G. Ryle, editor, *Contemporary Aspects of Philosophy*, pages 30–56. Oriel Press, 1977.
- [14] Ofer Arieli and Arnon Avron. The value of the four values. *Artificial Intelligence*, 102(1):97–141, 1998.
- [15] Matthew L. Ginsberg. Multi-valued logics. In Tom Kehler and Stan Rosenschein, editors, *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 243–247, 1986.
- [16] Matthew L. Ginsberg. Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.
- [17] Michael Kifer and Eliezer L. Lozinskii. A logic for reasoning with inconsistency. *Journal of Automated Reasoning*, 9(2):179–215, October 1992.
- [18] Michael Kifer and V. S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *Journal of Logic Programming*, 12:335–367, 1992.
- [19] K. Thirunarayan and M. Kifer. A theory of nonmonotonic inheritance based on annotated logic. *Artificial Intelligence*, 60(1):23–50, March 1993.

- [20] Anna Radzikowska. A three-valued approach to default logic. *Journal of Applied Non-Classical Logics*, 6(2), 1996.
- [21] Melvin C. Fitting. Kleene's three-valued logics and their children. *Fundamenta Informaticae*, 20(1–3):113–131, 1994.
- [22] Philippe Besnard and Torsten Schaub. A simple signed system for paraconsistent reasoning. In José Júlio Alferes, Luís Moniz Pereira, and Ewa Orłowska, editors, *Proceedings of the European JELIA Workshop (JELIA-96): Logics in Artificial Intelligence*, volume 1126 of *LNAI*, pages 404–416, Berlin, September 30–October 3 1996. Springer.
- [23] Philippe Besnard and Torsten Schaub. Signed systems for paraconsistent reasoning. *Journal of Automated Reasoning*, 20:191–213, 1998.
- [24] Philippe Besnard, Torsten Schaub, Hans Tompits, and Stefan Woltran. Paraconsistent reasoning via quantified boolean formulas, I: Axiomatising signed systems. In Sergio Flesca, Sergio Greco, Nicola Leone, and Giovambattista Ianni, editors, *JELIA*, volume 2424 of *Lecture Notes in Computer Science*, pages 320–331. Springer, 2002.
- [25] Ofer Arieli. Preferential logics for reasoning with graded uncertainty. In Thomas D. Nielsen and Nevin Lianwen Zhang, editors, *ECSQARU*, volume 2711 of *Lecture Notes in Computer Science*, pages 515–527. Springer, 2003.
- [26] Ofer Arieli. Paraconsistent reasoning and preferential entailments by signed quantified boolean formulae. *ACM Trans. Comput. Log.*, 8(3), 2007.
- [27] James J. Lu, Neil V. Murray, and Erik Rosenthal. Signed formulas and annotated logics. In *Proc. 23rd International Symposium on Multiple-Valued Logics*, pages 48–53. IEEE CS Press, Los Alamitos, 1993.
- [28] A. Hunter. How to act on inconsistent news: Ignore, resolve, or reject. *Data & Knowledge Engineering*, 57(3):221–239, 2006.

- [29] 马跃. 语义万维网中的不协调知识处理. 博士毕业论文, 北京大学, 2008.
- [30] K. Mu, Z. Jin, R. Lu, and W. Liu. Measuring inconsistency in requirements specifications. In L. Godo, editor, *Proc. of ECSQARU'05*, volume 3571 of *LNCS*, pages 440–451. Springer, 2005.
- [31] Ana Belén Barragáns Martínez, José J. Pazos Arias, and Ana Fernández Vilas. On measuring levels of inconsistency in multi-perspective requirements specification. In *1st Conference on the Principles of Software Engineering (PRISE)*, pages 21–30, nov 2004.
- [32] Xuefeng Zhu and Zhi Jin. Inconsistency measurement of software requirements specifications: An ontology-based approach. In *ICECCS*, pages 402–410. IEEE Computer Society, 2005.
- [33] John Grant and Anthony Hunter. Measuring inconsistency in knowledgebases. *J. Intell. Inf. Syst.*, 27(2):159–184, 2006.
- [34] Yue Ma, Guilin Qi, Pascal Hitzler, and Zuoquan Lin. Measuring inconsistency for description logics based on paraconsistent semantics. In *Proc. of ECSQARU'07*, pages 30–41. Springer, 2007.
- [35] Qingfeng Chen, Chengqi Zhang, and Shichao Zhang. A verification model for electronic transaction protocols. In Jeffrey Xu Yu, Xuemin Lin, Hongjun Lu, and Yanchun Zhang, editors, *APWeb*, volume 3007 of *Lecture Notes in Computer Science*, pages 824–833. Springer, 2004.
- [36] A. Hunter. Measuring inconsistency in knowledge via quasi-classical models. In *Proc. of AAI'02*, pages 68–73. AAAI Press, 2002.
- [37] A. Hunter and S. Konieczny. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, pages 191–236. Springer, 2005.

- [38] A. Hunter and S. Konieczny. Shapley inconsistency values. In *Proc. of KR'06*, pages 249–259. AAAI Press, 2006.
- [39] K. Knight. Measuring inconsistency. *Journal of Philosophical Logic*, 31(1):77–98, 2002.
- [40] J. Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19(3):435–444, 1978.
- [41] J. Grant and A. Hunter. Analysing inconsistent first-order knowledge bases. *Artificial Intelligence*, 172:1064–1093, 2008.
- [42] Hector J. Levesque. A logic of implicit and explicit belief. In *Proc. of AAAI'84*, pages 198–202. AAAI Press, 1984.
- [43] Graham Priest. Minimally inconsistent LP. *Journal Studia Logica*, 50(2):321–331, 1991.
- [44] Anthony Hunter. Reasoning with contradictory information using quasi-classical logic. *Journal of Logic and Computation*, 10(5):677–703, 2000.
- [45] Yue Ma, Guilin Qi, Guohui Xiao, Pascal Hitzler, and Zuoquan Lin. An anytime algorithm for computing inconsistency measurement. In *Proc. of KSEM'09*, pages 29–40. Springer, 2009.
- [46] J. Grant and A. Hunter. Measuring inconsistency in knowledgebases. *Journal of Intelligent Information Systems*, 27(2):159–184, 2006.
- [47] M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249–310, 1995.
- [48] O. Arieli and A. Avron. The value of the four values. *Artificial Intelligence*, 102:97–141, 1998.

- [49] Philippe Besnard and Anthony Hunter. Quasi-classical logic: Non-trivializable classical reasoning from inconsistent information. In *Proc. of ECSQARU'95*, pages 44–51. Springer, 1995.
- [50] Satisfiability suggested format, May 1993. <ftp://dimacs.rutgers.edu/pub/challenge/satisfiability/doc/satformat.dvi>.
- [51] Maxsat-2007: Benchmarks and solver requirements, 2007. <http://www.maxsat07.udl.es/solver-requirements.html>.
- [52] D. Le Berre. SAT4J: A Satisfiability Library for Java, 2009. <http://www.sat4j.org>.
- [53] Joao Marques-Silva. The msuncore maxsat solver. Technical report, CASL/CSI, University College Dublin, 2009.
- [54] Knot Pipatsrisawat and Adnan Darwiche. Clone: Solving weighted max-sat in a reduced search space. In *Proc. of AI'07*, pages 223–233. Springer, 2007.
- [55] Knot Pipatsrisawat, Akop Palyan, Mark Chavira, Arthur Choi, and Adnan Darwiche. Solving weighted max-sat problems in a reduced search space: A performance analysis. *Journal on Satisfiability Boolean Modeling and Computation (JSAT)*, 4:191–217, 2008.
- [56] C.H. Papadimitriou, editor. *Computational Complexity*. Addison Wesley, 1994.
- [57] Marco Cadoli and Marco Schaerf. On the complexity of entailment in propositional multivalued logics. *Annals of Mathematics and Artificial Intelligence*, 18(1):29–50, 1996.
- [58] Pierre Marquis and Nadège Porquet. Computational aspects of quasi-classical entailment. *Journal of Applied Non-Classical Logics*, 11(3–4):295–312, 2001.
- [59] Matthias Baaz, Uwe Egly, and Alexander Leitsch. *Normal Form Transformations*, chapter 5, pages 273–333. Elsevier Science, 2001.

- [60] Liping Zhou, Houkuan Huang, Guilin Qi, Yue Ma, Zhisheng Huang, and Youli Qu. Measuring inconsistency in DL-Lite ontologies. In *Proc. of WI'09*, pages 349–356. Springer, 2009.
- [61] Carsten Sinz, Andreas Kaiser, and Wolfgang Kuchlin. Formal methods for the validation of automotive product configuration data. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 17(1):75–97, 2003.
- [62] Boolvar/pb, 2009. <http://boolvar.sourceforge.net/>.
- [63] pbclasp, 2010. <http://potassco.sourceforge.net/labs.html>.
- [64] Scip solving constraint integer programs, 2008. <http://scip.zib.de/>.

致 谢

首先我要感谢我的导师林作铨教授。您的渊博的常识和求学的态度对我的学业有着莫大的帮助，同时您的人格魅力也深深感染了我，激励着我不断的前进。

感谢实验室的牟克典老师和所有的同学，我的每一点进步都是与你们的帮助分不开的。在实验室内的讨论氛围中，我学到了很多。

感谢我的父母，你们的生育、养育和教育之恩毕生难报。

最后特别感谢我的女友丁林芳，感谢你一直以来对我的理解和支持。而且本文的核心思想诞生在与你旅行的火车上，永生难忘。

学位论文出版授权声明

本人已经认真阅读《“中国精品学位论文全文数据库”建设章程》，同意将本人的学位论文提交给“中国精品学位论文全文数据库”项目的产品开发与运作方——北京北大方正电子有限公司全文发表，并可按[“中国精品学位论文全文数据库稿酬支付说明”](#)享受相关权益。
同意论文提交后滞后：半年；一年；二年发布。

作者签名： _____
_____年__月__日

导师签名： _____
_____年__月__日

“中国精品学位论文全文数据库”稿酬支付说明

作者信息：姓名：_____ 学号：_____ 所在院系：_____

提交论文类型： 硕士论文， 博士论文

授权作者可以选择报酬方式：

1、唯一授权：本人论文电子版**独家**授权给北大方正，并可选择下列报酬方式（三选一）

(1) 读书卡：硕士论文作者将获得有效期 5 年价值 449 元的方正 Apabi 数字图书馆读书卡 1 张，博士论文作者将获得有效期 10 年价值 669 元的方正 Apabi 数字图书馆读书卡 1 张；

(2) 现金+读书卡：硕士论文作者将获得 30 元现金和有效期 3 年价值 269 元的方正 Apabi 数字图书馆读书卡 1 张，博士论文作者将获得 80 元现金和有效期 5 年价值 449 元的方正 Apabi 数字图书馆读书卡 1 张。

(3) 销售分成：作者每年获得作者本人提交论文销售收入的 10 % 作为著作权使用费。本人同意提供并填写完整、正确的个人信息，并在下列任一个个人信息发生改变时及时通知北大方正（发邮件到 ApabiCEDD@founder.com，或上网 <http://www.apabi.com> 在线提交），若因下列个人信息填写不完整、不正确或未将变化及时通知北大方正而发生的著作权使用费无法按期支付等问题由本人负责。北大方正及项目其它参与方负责对作者的下列个人信息保密：

作者姓名：_____ 开户行：_____ 银行帐号：_____

Email：_____ 手机：_____

2、非唯一授权：本人论文电子版以**非独家**授权方式授权给北大方正，并获得下列报酬：

读书卡：硕士论文作者将获得有效期 3 年价值 269 元的方正 Apabi 数字图书馆读书卡 1 张，博士论文作者将获得有效期 5 年价值 449 元的方正 Apabi 数字图书馆读书卡 1 张。

同时，本人导师将获得方正 Apabi 数字图书馆读书卡 1 张（硕士论文作者的导师的读书卡有效期 5 年，价值 449 元人民币；博士论文作者的导师的读书卡有效期 10 年，价值 669 元人民币）。

北大方正将按照作者选择的论文授权方式和报酬方式，及时支付作者应得的报酬。北大方正将提供数字版权保护技术（DRM）确保合法使用作者论文。同时，北大方正负责为作者提供网上查询论文销售情况及作者个人信息的服务，服务主页：<http://www.apabi.com>。

注：1、本“稿酬支付说明”一式两份，一份由作者本人保存，另一份做为领取稿酬凭证，交送给图书馆。

2、“学位论文授权使用声明”请直接装订在论文影印本的后面。

3、授权作者需提交“学位论文授权使用声明”的复印件、稿酬支付说明原件到图书馆论文采收处，领取稿酬。