

G_3 逻辑中的弱合取范式^{*})

章衡^{1,3} 张明义² 杨本娟¹

(贵州大学信息工程学院 贵阳 550003)¹ (贵州科学院 贵阳 550002)²

(贵州民族学院经济管理学院 贵阳 550025)³

摘要 本文为 G_3 逻辑提出一种类似于经典逻辑中合取范式的弱合取范式,并给出两种范式化简算法:一种是通过公式刻画反模型的语义方法;另一种为基于重写翻译的语法方法。文章最后证明,在 G_3 逻辑中对任意公式做弱合取范式化简不存在多项式算法。

关键词 弱合取范式, G_3 逻辑, 范式化简, 计算复杂性

Weak Conjunctive Normal Form for G_3 Logic

ZHANG Heng^{1,3} ZHANG Ming-Yi² YANG Ben-Juan¹

(School of Information Engineering, Guizhou University, Guiyang 550003)¹ (Guizhou Academy of Sciences, Guiyang 550002)²

(School of Economic Administration, Guizhou University for Nationalities, Guiyang 550025)³

Abstract In this paper, we put forward a weak conjunctive normal form for G_3 logic, which is similar to the conjunctive normal form for classical propositional logic. We also give two algorithms to reduce a formula to the weak conjunctive normal form, where one is based on characterizing counter models in model-based-formulas, and the other is based on rewriting formulas according to transformation rules. In the end, we show that there is no polynomial time algorithm to reduce an arbitrary formula to the weak conjunctive normal form.

Keywords Weak conjunctive normal form, G_3 logic, Normal form reduction, Computational complexity

1 引言

Gödel 逻辑是由著名逻辑学家 Gödel 于 1932 年提出的一类多值逻辑,用来证明直觉主义逻辑不存在有穷多值语义^[5]。近年来, Gödel 逻辑已逐渐成为计算机科学研究的一个热点,主要体现在:以 Gödel 逻辑作为基础的模糊逻辑正被广泛地研究;被用来刻画逻辑程序设计与非单调推理中的稳定模型语义^[6];与直觉主义逻辑之间的紧密关系使之可能成为并行计算的一种逻辑分析工具^[1]。

G_3 逻辑,又称 HT (here-and-there) 逻辑,是一种 3 值的 Gödel 逻辑,在众多 Gödel 逻辑中极具典型性。Lifschitz 等人证明了两个逻辑程序在稳定模型语义下强等价当且仅当这两个逻辑程序对应的公式在 G_3 逻辑中等价^[7]。因此,我们对 G_3 逻辑中范式的研究不但对基于 G_3 逻辑的自动定理证明、自动推理与逻辑程序设计具有重要的意义,同时相应结果也可在强等价意义下应用到稳定模型语义下的逻辑程序设计。

本文第 2 节介绍 G_3 逻辑的语法、语义,给出相关性质;第 3 节定义弱合取范式,并给出其在 G_3 逻辑中存在性的一种构造性证明;第 4 节给出基于语法重写翻译的一种范式生成算法;第 5 节分析范式生成问题的计算复杂性。

2 G_3 逻辑

本文使用的命题语言 \mathcal{L} 由命题变元集 $Var \mathcal{L}$ 、命题常元 \perp 及联结词 \wedge 、 \vee 和 \rightarrow 按标准方式构成。我们令 $For \mathcal{L}$ 表示所有命题公式的集合。对于 $Var \mathcal{L}$ 的任何子集 Σ ,我们说 φ 为 Σ 上的命题公式当且仅当 $\varphi \in For \mathcal{L}$,且该公式中出现的命

题变元均属于 Σ 。根据习惯,我们令 $\neg\varphi$ 表示 $\varphi \rightarrow \perp$, \top 表示表示 $\perp \rightarrow \perp$ 。通常地,我们称 \perp 、 \top 或命题变元为原子公式,并用小写拉丁字母表示原子公式,小写希腊字母表示一般命题公式。

G_3 逻辑的语义主要有两种形式:一种是传统的 3 值语义,另一种是 Kripke 语义。这里给出 Kripke 语义的一种简化版本。称二元组 $\langle X, Y \rangle$ 为 G_3 逻辑的一个 Kripke 解释(以下简称 G_3 解释)当且仅当 $X \subseteq Y \subseteq Var \mathcal{L}$ 。对任一 $\Sigma \subseteq Var \mathcal{L}$,称 $\langle X, Y \rangle$ 为 Σ 上的 G_3 解释当且仅当 $X \subseteq Y \subseteq \Sigma$ 。 G_3 解释与公式之间的可满足关系 \models 通过下述方式递归定义:

- 1° $\langle X, Y \rangle \models p$, 当 $p \in X$
- 2° $\langle X, Y \rangle \not\models \perp$
- 3° $\langle X, Y \rangle \models \varphi \wedge \psi$, 当 $\langle X, Y \rangle \models \varphi$ 且 $\langle X, Y \rangle \models \psi$
- 4° $\langle X, Y \rangle \models \varphi \vee \psi$, 当 $\langle X, Y \rangle \models \varphi$ 或 $\langle X, Y \rangle \models \psi$
- 5° $\langle X, Y \rangle \models \varphi \rightarrow \psi$, 当
 - (i) $\langle X, Y \rangle \not\models \varphi$ 或 $\langle X, Y \rangle \models \psi$, 且
 - (ii) $Y \models \varphi \rightarrow \psi$.

当 $\Gamma \subseteq For \mathcal{L}$,则我们定义 $\langle X, Y \rangle \models \Gamma$ 当且仅当对任意 $\varphi \in \Gamma$,均有 $\langle X, Y \rangle \models \varphi$ 。

对任意 G_3 解释 $\langle X, Y \rangle$ 和任意命题公式 φ ,若 $\langle X, Y \rangle \models \varphi$,则称 $\langle X, Y \rangle$ 为 φ 的模型,否则称 $\langle X, Y \rangle$ 为 φ 的反模型。类似地,对任意 G_3 解释 $\langle X, Y \rangle$ 和任意 $\Gamma \subseteq For \mathcal{L}$,若 $\langle X, Y \rangle \models \Gamma$,则称 $\langle X, Y \rangle$ 为 Γ 的模型,否则称 $\langle X, Y \rangle$ 为 Γ 的反模型。我们说公式(或公式集) φ 与 ψ 在 G_3 中逻辑等价当且仅当它们有相同的模型,记 $\varphi \equiv_{G_3} \psi$ 。公式 φ 是有效的当且仅当任何 G_3 解释都是 φ 的模型。

^{*} 本课题得到国家自然科学基金(60573009)和贵州省长基金 2005(212)资助。章衡 硕士生,研究方向为计算机科学逻辑;张明义 研究员,博士生导师,研究方向为非经典逻辑、计算机科学理论、人工智能;杨本娟 硕士生,研究方向为计算机科学理论。

事实 (F1) $\langle X, Y \rangle \models \varphi$ 当且仅当 $X \models \varphi$;

(F2) 若 $\langle X, Y \rangle \models \varphi$, 则 $Y \models \varphi$;

(F3) $\langle X, Y \rangle \models \neg \varphi$ 当且仅当 $Y \models \neg \varphi$.

关于 G_3 的更多内容请参考文[4]。下面我们证明两个后文中需要用到的命题。

命题 1 设 Σ 为任一有穷的命题变元集合, 则共有 $3^{|\Sigma|}$ 个不同的 Σ 上的 G_3 解释。

证明: 令 $\langle X, Y \rangle$ 表示任意 G_3 解释, 则对 Σ 中的任一命题变元, 要么属于 X , 要么不属于 X 但属于 Y , 要么既不属于 X 也不属于 Y , 即共有 3 种选择, 故 Σ 上不同 G_3 解释的个数为 $3^{|\Sigma|}$ 。□

命题 2 设 $Y \subseteq \text{Var } \mathcal{L}$, φ 为任意命题公式, 且 $\langle Y, Y \rangle \not\models \varphi$, 则对所有 $X \subseteq Y$, 均有 $\langle X, Y \rangle \not\models \varphi$ 。

证明: 不妨反设 $\langle X, Y \rangle \models \varphi$ 。根据事实(F2), 立得 $Y \models \varphi$ 。但又由事实(F1), 从题设 $\langle Y, Y \rangle \not\models \varphi$ 有 $Y \not\models \varphi$ 。两者矛盾, 故有 $\langle X, Y \rangle \not\models \varphi$ 。□

3 弱合取范式

本节我们先给出弱合取范式的定义, 然后给出该范式在 G_3 逻辑存在性的一种构造性证明。

定义 1 称一个命题公式为弱文字当且仅当该公式属于以下 4 种形式之一: $p, \neg p, \neg \neg p$ 或 $p \rightarrow q$, 其中 p, q 为任意命题变元; 称有穷个弱文字的析取式为弱合取子式; 称有穷个弱合取子式的合取式为弱合取范式。

注释 1 事实上, 弱合取范式是在经典合取范式文字集中加入双否定与蕴含文字得到的一种扩展。

下面我们通过一种类似于经典命题逻辑中使用真值表生成合取范式的构造性方法来证明 G_3 逻辑可归约到弱合取范式。

定义 2 设 Σ 为 $\text{Var } \mathcal{L}$ 的有穷子集, $\langle X, Y \rangle$ 为 Σ 上的任意 G_3 解释。我们令 $C_{\langle X, Y \rangle}$ 表示如下公式:

$$\left(\bigvee_{a \in Y-X} a \right) \vee \left(\bigvee_{b \in Y} \neg b \right) \vee \left(\bigvee_{c \in \Sigma-Y} \neg \neg c \right) \vee \left(\bigvee_{\substack{d \in X \\ e \in Y-X}} (d \rightarrow e) \right)$$

命题 3 设 Σ 为 $\text{Var } \mathcal{L}$ 的有穷子集, $\langle X, Y \rangle$ 与 $\langle X', Y' \rangle$ 是 Σ 上两任意 G_3 解释, 则 $\langle X', Y' \rangle \models C_{\langle X, Y \rangle}$ 当且仅当下面条件均成立:

- (1) 若 $X \neq Y$, 则 $X' = X$ 且 $Y' = Y$;
- (2) 若 $X = Y$, 则 $X' \subseteq Y$ 且 $Y' = Y$ 。

证明: $1^\circ \Rightarrow$

用反证法。我们先讨论 Y' 。不妨反设 $Y' \neq Y$, 则可能存在于两种情形:

情形 1 $\exists p \in Y'$ 使得 $p \notin Y$ 。显然, $p \in \Sigma - Y$ 且 $Y' \models \neg \neg p$ 。由事实(F3), 有 $\langle X', Y' \rangle \models \neg \neg p$ 。因此 $\langle X', Y' \rangle \models C_{\langle X, Y \rangle}$, 与题设 $\langle X', Y' \rangle \not\models C_{\langle X, Y \rangle}$ 矛盾。

情形 2 $\exists p \in Y$ 使得 $p \notin Y'$ 。此时, 容易证明 $\langle X', Y' \rangle \models \neg p$ 。因此, 亦有 $\langle X', Y' \rangle \models C_{\langle X, Y \rangle}$; 与题设 $\langle X', Y' \rangle \not\models C_{\langle X, Y \rangle}$ 矛盾。

故若 $\langle X', Y' \rangle \not\models C_{\langle X, Y \rangle}$, 则必有 $Y' = Y$ 。

接下来讨论 X' 。类似地, 我们反设 $\exists p \in X'$ 使得 $p \notin X$ 。显然 $\langle X', Y' \rangle \models p$ 。又由 $X' \subseteq Y' = Y$, 得 $p \in Y - X$ 。故 $\langle X', Y' \rangle \models C_{\langle X, Y \rangle}$, 与题设矛盾。因此, 必有 $X' \subseteq X$ 成立。

特别地, 当 $X \neq Y$ 时, 我们证明 $X' = X$ 。为得到这一结论, 反设 $\exists p \in X$ 使得 $p \notin X'$ 。由 $X \neq Y$ 与 $X \subseteq Y$, 有 $Y - X \neq \emptyset$ 。从 $Y - X$ 中任取 q , 由 $\langle X', Y' \rangle \not\models C_{\langle X, Y \rangle}$, 立得 $\langle X', Y' \rangle \not\models p \rightarrow q$ 。另一方面, 由 $q \in Y - X$ 和 $Y' = Y$ 有 $Y' \models p \rightarrow q$ 。又由 $p \notin X'$ 有 $\langle X', Y' \rangle \not\models p$, 从而 $\langle X', Y' \rangle \models p \rightarrow q$, 与前面结论矛盾。故当 $X \neq Y$ 时, $X' = X$ 成立。

$2^\circ \Leftarrow$
容易证明 $\langle X, Y \rangle \not\models C_{\langle X, Y \rangle}$, 故如果 $X' = X$ 且 $Y' = Y$, 则立有 $\langle X', Y' \rangle \not\models C_{\langle X, Y \rangle}$ 。特别地, 当 $X = Y = Y'$ 时, 有 $\langle Y', Y' \rangle \not\models C_{\langle X, Y \rangle}$ 。根据命题 2, 若 $X' \subseteq Y$, 则 $\langle X', Y' \rangle \not\models C_{\langle X, Y \rangle}$ 。□

定理 1 设 Σ 为 $\text{Var } \mathcal{L}$ 的有穷子集, Γ 为 Σ 上的任一命题公式集合, 并令

$$\mathfrak{F}(\Gamma) = \bigwedge_{\langle X, Y \rangle: \langle X, Y \rangle \not\models \Gamma} C_{\langle X, Y \rangle}$$

则 $\mathfrak{F}(\Gamma)$ 与 Γ 在 G_3 逻辑中逻辑等价。

证明: 若 $\langle X', Y' \rangle$ 为 $\mathfrak{F}(\Gamma)$ 的模型, 则 $\langle X', Y' \rangle$ 必是 Γ 的模型。否则, 根据 $\mathfrak{F}(\Gamma)$ 的定义, $C_{\langle X', Y' \rangle}$ 是 $\mathfrak{F}(\Gamma)$ 的合取项。由命题 3, 总有 $\langle X', Y' \rangle \not\models C_{\langle X', Y' \rangle}$, 这与假设 $\langle X', Y' \rangle$ 为 $\mathfrak{F}(\Gamma)$ 的模型矛盾。

反过来, 若 $\langle X', Y' \rangle$ 非 $\mathfrak{F}(\Gamma)$ 的模型, 则必存在 Γ 的一个反模型 $\langle X, Y \rangle$ 使得 $\langle X', Y' \rangle \not\models C_{\langle X, Y \rangle}$ 。此时, 由命题 3, 若 $X \neq Y$, 则 $X' = X$ 且 $Y' = Y$, 即 $\langle X', Y' \rangle$ 是 Γ 的反模型; 若 $X = Y$, 则 $X' \subseteq Y$ 且 $Y' = Y$, 由此立得 $\langle Y', Y' \rangle$ 不是 Γ 的模型, 进而根据命题 2, 有 $\langle X', Y' \rangle$ 也不可能为 Γ 的模型。

故 $\mathfrak{F}(\Gamma)$ 与 Γ 在 G_3 逻辑中逻辑等价。□

注释 2 (1) 定理 1 说明, 对任一命题公式, 都存在与之在 G_3 逻辑中等价的弱合取范式。或者简单地说, G_3 逻辑可归约到弱合取范式。

(2) 命题 3 与定理 1 给出了一种从反模型生成范式的语义方法。给定含 n 个命题变元的集合 Σ 和 Σ 上的公式 φ , 根据命题 1, 该公式最多有 3^n 个反模型。因此, $\mathfrak{F}(\{\varphi\})$ 最多可能含 3^n 个弱合取子式。

例 1 令 $\Sigma = \{a, b\}$, 则 $\neg a \rightarrow \neg b$ 共有两个 Σ 上的反模型: $\langle \emptyset, \{b\} \rangle$ 与 $\langle \{b\}, \{b\} \rangle$ 。根据命题 3, 我们可以构造得到:

$$C_{\langle \emptyset, \{b\} \rangle} = b \vee \neg b \vee \neg \neg a,$$

$$C_{\langle \{b\}, \{b\} \rangle} = \neg b \vee \neg \neg a$$

则 $\mathfrak{F}(\{\neg a \rightarrow \neg b\}) = (C_{\langle \emptyset, \{b\} \rangle}) \wedge (C_{\langle \{b\}, \{b\} \rangle})$ 。容易验证 $\neg a \rightarrow \neg b \equiv_{G_3} (C_{\langle \emptyset, \{b\} \rangle}) \wedge (C_{\langle \{b\}, \{b\} \rangle})$ 。

例 2 考虑公式 $(\neg a \rightarrow b) \rightarrow (a \rightarrow c)$ 。我们令 $\Sigma = \{a, b, c\}$, 下面给出反模型及生成子式列表:

反模型 $\langle X, Y \rangle$	生成弱合取子式 $C_{\langle X, Y \rangle}$
$\langle \emptyset, \{a\} \rangle$	$a \vee \neg a \vee \neg \neg b \vee \neg \neg c$
$\langle \emptyset, \{a, b\} \rangle$	$a \vee b \vee \neg a \vee \neg b \vee \neg \neg c$
$\langle \{b\}, \{a, b\} \rangle$	$a \vee \neg a \vee \neg b \vee \neg \neg c \vee (b \rightarrow a)$
$\langle \{a\}, \{a\} \rangle$	$\neg a \vee \neg \neg b \vee \neg \neg c$
$\langle \{a\}, \{a, c\} \rangle$	$c \vee \neg a \vee \neg c \vee \neg \neg b \vee (a \rightarrow c)$
$\langle \{a\}, \{a, b\} \rangle$	$b \vee \neg a \vee \neg b \vee \neg \neg c \vee (a \rightarrow b)$
$\langle \{a\}, \{a, b, c\} \rangle$	$b \vee c \vee \neg a \vee \neg b \vee \neg c$
	$\vee (a \rightarrow b) \vee (a \rightarrow c)$
$\langle \{a, b\}, \{a, b\} \rangle$	$\neg a \vee \neg b \vee \neg \neg c$
$\langle \{a, b\}, \{a, b, c\} \rangle$	$c \vee \neg a \vee \neg b \vee \neg c$
	$\vee (a \rightarrow c) \vee (b \rightarrow c)$

对以上子式取合取, 即得到所要的范式。

4 范式的语法翻译

上一节中, 我们给出了一种从反模型构造范式的方法。不幸的是, 对于由 n 个命题变元构成的公式, 我们必须逐一检

查 3^n 个 G_3 解释是否为待化简公式的反模型。若是,则生成对应子式。显然,这种方法非常低效,构造出的范式也不够简练。本节我们从语法途径给出一种基于重写的范式化简算法,并证明该算法的正确性。

我们定义一个重写系统 T 来实现这一范式化简算法。 T 由两部分组成:准弱合取子式集 S 与重写规则集 R 。我们将 S 中的准弱合取子式看作一个析取式,其析取项可为除析取式之外的任意合法公式。由于关于析取与合取的交换律与结合律在 G_3 逻辑中均成立,因此,我们总是可以将一个准弱合取子式写成 $\lambda_1 \vee \dots \vee \lambda_m \vee \lambda_{m+1} \vee \dots \vee \lambda_n$ 的形式,其中, $0 \leq m \leq n, n \geq 1$, 且当 $1 \leq i \leq m$ 时 λ_i 为非析取式,当 $m+1 \leq i \leq n$ 时 λ_i 为弱文字。在下面的重写规则中,我们令 $\varphi = \lambda_1, \alpha = \lambda_2 \vee \dots \vee \lambda_n$,即将准弱合取子式写成 $\varphi \vee \alpha$ 的形式;特别地,当 $n=1$ 时,我们省略规则中的“ $\vee \alpha$ ”部分。

下面描述使用 T 化简命题公式 ξ 的过程。我们令 S_i 表示系统第 i 步后得到的合取子式集。特别地,我们规定初始子式集 $S_0 = \{\xi\}$ 。第 $i+1$ 步重写按以下步骤进行:(1)从 S_i 中选择一个须进一步化简的子式 $\varphi \vee \alpha$;(2)将 S_i 除该子式以外的其它子式加入 S_{i+1} ;(3)从 R 中选取重写规则应用到 $\varphi \vee \alpha$,得到一个或若干个新的子式,并将这些新子式加入到 S_{i+1} 。如此迭代进行,直到 n 步后对所有子式都没有任何重写规则可应用,则化简结束。此时,对 S_n 各子式取合取式,即得到化简后的范式。

重写规则的选取主要依据选定子式中 φ 的公式结构。若 φ 为否定式,则有以下四种形式需要进行重写:(1) $\neg\neg\psi$;(2) $\neg(\psi \wedge \chi)$;(3) $\neg(\psi \vee \chi)$;(4) $\neg(\psi \rightarrow \chi)$ 。其中对第(1)种形式需进一步细分成四种形式: $\neg\neg\neg\psi, \neg\neg\neg(\psi \wedge \chi), \neg\neg\neg(\psi \vee \chi)$ 和 $\neg\neg\neg(\psi \rightarrow \chi)$,因此,我们有下面七条规则:

$$\neg\neg\neg\psi \vee \alpha \Rightarrow \neg\neg\psi \vee \alpha \quad R1$$

$$\neg\neg\neg(\psi \wedge \chi) \vee \alpha \Rightarrow \begin{cases} \neg\neg\neg\psi \vee \alpha \\ \neg\neg\neg\chi \vee \alpha \end{cases} \quad R2$$

$$\neg\neg\neg(\psi \vee \chi) \vee \alpha \Rightarrow \neg\neg\neg\psi \vee \neg\neg\neg\chi \vee \alpha \quad R3$$

$$\neg\neg\neg(\psi \rightarrow \chi) \vee \alpha \Rightarrow \neg\psi \vee \neg\neg\neg\chi \vee \alpha \quad R4$$

$$\neg(\psi \wedge \chi) \vee \alpha \Rightarrow \neg\psi \vee \neg\chi \vee \alpha \quad R5$$

$$\neg(\psi \vee \chi) \vee \alpha \Rightarrow \begin{cases} \neg\psi \vee \alpha \\ \neg\chi \vee \alpha \end{cases} \quad R6$$

$$\neg(\psi \rightarrow \chi) \vee \alpha \Rightarrow \begin{cases} \neg\neg\neg\psi \vee \alpha \\ \neg\chi \vee \alpha \end{cases} \quad R7$$

若 φ 为合取式,则直接应用下面的分配律:

$$(\psi \wedge \chi) \vee \alpha \Rightarrow \begin{cases} \psi \vee \alpha \\ \chi \vee \alpha \end{cases} \quad R8$$

若 φ 为蕴含式,则其前件与后件中有一个为否定式、合取式、析取式和蕴含式四者之一时,子式须进行重写,因此共有 8 条关于蕴含式的规则:

$$(\neg\psi \rightarrow \rho) \vee \alpha \Rightarrow \neg\neg\psi \vee \rho \vee \alpha \quad R9$$

$$((\psi \vee \chi) \rightarrow \rho) \vee \alpha \Rightarrow (\psi \rightarrow \rho) \vee (\chi \rightarrow \rho) \vee \alpha \quad R10$$

$$((\psi \vee \chi) \rightarrow \rho) \vee \alpha \Rightarrow \begin{cases} (\psi \rightarrow \rho) \vee \alpha \\ (\chi \rightarrow \rho) \vee \alpha \end{cases} \quad R11$$

$$((\psi \rightarrow \chi) \rightarrow \rho) \vee \alpha \Rightarrow \begin{cases} \neg\neg\psi \vee \rho \vee \alpha \\ (\chi \rightarrow \rho) \vee \alpha \\ \psi \vee \neg\chi \vee \rho \vee \alpha \end{cases} \quad R12$$

$$(\psi \rightarrow \neg\chi) \vee \alpha \Rightarrow \neg\psi \vee \neg\chi \vee \alpha \quad R13$$

$$(\psi \rightarrow (\chi \wedge \rho)) \vee \alpha \Rightarrow \begin{cases} (\psi \rightarrow \chi) \vee \alpha \\ (\psi \rightarrow \rho) \vee \alpha \end{cases} \quad R14$$

$$(\psi \rightarrow (\chi \vee \rho)) \vee \alpha \Rightarrow (\psi \rightarrow \chi) \vee (\psi \rightarrow \rho) \vee \alpha \quad R15$$

$$(\psi \rightarrow (\chi \rightarrow \rho)) \vee \alpha \Rightarrow (\psi \rightarrow \rho) \vee (\chi \rightarrow \rho) \vee \alpha \quad R16$$

以上 16 条规则就构成了 T 的重写规则集 R 。

定理 2 (T 的可靠性与完备性)

(1) T 是可靠的。即若 S_n 是从 S_0 经过 n 步重写得到,则 $\wedge S_n \equiv_{G_3} \wedge S_0$ 。

(2) T 是完备的。即对任意命题公式 ξ , T 总能在有穷步内终止,且 ξ 被化简为弱合取范式。

证明:(1) 容易验证, $R1-R16$ 中任一规则都是可靠的,即规则右部的合取式与规则左部在 G_3 逻辑中等价。由 G_3 逻辑的等值替换性质,若 $0 \leq i < n$, 则 $\wedge S_{i+1} \equiv_{G_3} \wedge S_i$ 。由此立得 $\wedge S_n \equiv_{G_3} \wedge S_0$ 。

(2) 先定义三个权值函数。对任意公式 λ , 令

$$\tau(\lambda) = \begin{cases} 0, & \text{若 } \lambda \in \text{Var } \mathcal{L} \\ \tau(\psi) + 1, & \text{若 } \lambda = \neg\psi \\ \max(\tau(\psi), \tau(\chi)) + 2, & \text{若 } \lambda = \psi * \chi \end{cases}$$

其中 $*$ $\in \{\wedge, \vee, \rightarrow\}$ 。如前所述,我们将一个准弱合取子式写成 $\lambda_1 \vee \dots \vee \lambda_n$, 其中,每一个析取项 λ_i 均不允许为析取式,即保证析取项的最小性。因此,对任意准弱合取子式 φ , 我们将其看作所有最小析取项的集合,并定义

$$\pi(\varphi) = \sum_{\lambda \in \varphi} (3^{\tau(\lambda)} - 1)$$

对于准弱合取子式集 S , 我们定义

$$\gamma(S) = \sum_{\varphi \in S} (3^{\pi(\varphi)} - 1)$$

显然,对任意准弱合取子式的有穷集 S , 都必有 $0 \leq \gamma(S) < \omega$ 。

接下来我们证明:对任意非空准弱合取子式集 S , T 总在 $\gamma(S)$ 步内终止,并且 S 被重写为弱合取子式集。我们通过 $\gamma(S)$ 进行归纳来证明:

当 $\gamma(S) = 0$ 时, S 必为命题变元集,此时,无重写规则可用,故 T 运行 0 步即终止;且 S 本身就是弱合取子式。显然命题成立。

当 $\gamma(S) > 0$ 时,假设对任意准弱合取子式 S' , 如果 $\gamma(S') < \gamma(S)$, 则 T 关于 S' 在 $\gamma(S')$ 步内终止,且 S' 被重写为弱合取子式集。若 S 中所有子式均为弱合取子式,则无需化简, T 立即终止,命题必然成立;否则,我们必能从 S 中选取一个非弱合取子式 σ 进行化简。按前述规定,我们将 σ 写成 $\varphi \vee \alpha$ 的形式,然后根据 φ 的结构进行分析。考虑到文章篇幅,我们仅以 φ 为蕴含式为例作证明。显然 φ 不可能为 $p \rightarrow q$ 形式,否则根据规定, p 本身必为弱合取子式,从而不应该被选取。因此, φ 必具有规则 $R9-R16$ 左部形式之一。这里仅考虑 $R12$ 的情形。此时, $\varphi = ((\psi \rightarrow \chi) \rightarrow \rho) \vee \alpha$, 为方便,我们同时令

$$u_1 =_{\text{def}} \neg\neg\psi \vee \rho \vee \alpha$$

$$u_2 =_{\text{def}} (\chi \rightarrow \rho) \vee \alpha$$

$$u_3 =_{\text{def}} \psi \vee \neg\chi \vee \rho \vee \alpha$$

由 τ 的定义,显然下列不等式成立:

$$\tau(\varphi) \geq \tau(\neg\neg\psi) + 2 \quad \tau(\varphi) \geq \tau(\rho) + 4$$

$$\tau(\varphi) \geq \tau(\chi \rightarrow \rho) + 2 \quad \tau(\varphi) \geq \tau(\psi) + 4$$

$$\tau(\varphi) \geq \tau(\neg\chi) + 3$$

又根据 π 的定义可知:

$$\pi(\sigma) = \pi(\alpha) + 3^{\tau(\varphi)} - 1$$

$$\pi(u_1) = \pi(\alpha) + 3^{\tau(\neg\neg\psi)} + 3^{\tau(\rho)} - 2$$

$$\pi(u_2) = \pi(\alpha) + 3^{\tau(\chi \rightarrow \rho)} - 1$$

$$\pi(u_3) = \pi(\alpha) + 3^{\tau(\psi)} + 3^{\tau(\neg\chi)} + 3^{\tau(\rho)} - 3$$

由此,我们可推断出下列不等式:

$$\pi(\sigma) > \pi(u_1), \pi(\sigma) > \pi(u_2), \pi(\sigma) > \pi(u_3)$$

我们令 $S^* = S - \{\sigma\}$, S' 表示 S 通过 R12 重写 σ 后得到的子式集。显然 $S' = S^* \cup \{u_1, u_2, u_3\}$ 。根据 γ 的定义, 我们有:

$$\gamma(S) = \gamma(S^*) + 3^{\pi(\sigma)} - 1$$

$$\gamma(S') = \gamma(S^*) + 3^{\pi(u_1)} + 3^{\pi(u_2)} + 3^{\pi(u_3)} - 3$$

容易看出, $\gamma(S') < \gamma(S)$ 。由归纳假设, T 关于 S' 在 $\gamma(S')$ 步内终止, 且 S' 可被重写为弱合取子式集 S_n 。因此, T 关于 S 在 $\gamma(S') + 1 \leq \gamma(S)$ 步内终止, 且 S 可被重写为弱合取子式集 S_n 。其余各种情况可类似地证明。

对于初始子句集 $S_0 = \{\xi\}$, 从 γ 的定义容易验证 $\gamma(S_0) < 3^{2\pi(\xi)}$ 。由上述结论, 对任意公式 T 总能在 $3^{2\pi(\xi)}$ 步内终止, 且 ξ 被化简为弱合取范式。 □

例 3 用 T 化简 $\neg a \rightarrow \neg b$:

S_0	$\neg a \rightarrow \neg b$	R9
S_1	$\neg \neg a \vee \neg b$	

例 4 用 T 化简 $(\neg a \rightarrow b) \rightarrow (a \rightarrow c)$:

S_0	$(\neg a \rightarrow b) \rightarrow (a \rightarrow c)$	R12
S_1	$\neg \neg \neg a \vee (a \rightarrow c)$	R1
	$b \rightarrow (a \rightarrow c)$	
	$\neg a \vee \neg b \vee (a \rightarrow c)$	
S_2	$\neg a \vee (a \rightarrow c)$	R16
	$b \rightarrow (a \rightarrow c)$	
	$\neg a \vee \neg b \vee (a \rightarrow c)$	
S_3	$\neg a \vee (a \rightarrow c)$	
	$(b \rightarrow c) \vee (a \rightarrow c)$	
	$\neg a \vee \neg b \vee (a \rightarrow c)$	

注释 3 在定理(2)的证明中, 我们证明了对任意公式 ξ , T 必在 $3^{2\pi(\xi)}$ 步内终止。实际上这是最坏时间复杂度的一个上界估计。这是否意味着我们的语法方法和语义方法比较并无优势呢? 通过例 1 与例 3, 例 2 与例 4 对比可以看出, 语法方法比语义方法明显效率要高, 生成的范式也更简单。读者可以验证, 大多数情况下语法翻译具有语义构造方式无法比拟的优势。当然, 在最坏情况下, 两者具有同样的复杂度。那么, 是否可设计一个更高效的多项式时间算法进行范式化简呢? 下一节的结果将说明: 如果 $P \neq \text{coNP}$, 则这种算法不存在。

5 计算复杂性

本节讨论范式化简问题的复杂性。我们的思路是将 G_3 逻辑中公式的有效性归约到范式化简问题。关于 G_3 逻辑中公式的有效性有如下结论:

命题 4(文[6]定理 5) 判断一个任意公式在 Gödel 逻辑中是否有效这一问题是 coNP 完全的。

在 G_3 逻辑中, 关于 \vee 的交换律与结合律总是成立的, 因此, 对于一个析取式, 我们可以任意改变各析取项的顺序; 又由 $p \equiv_{c_3} \top \rightarrow p$, 故对任一弱合取子式, 我们均可将其等价地写成如下形式:

$$(a_1 \rightarrow b_1) \vee \dots \vee (a_m \rightarrow b_m) \vee \neg \neg \vee c_1 \vee \dots \vee \neg \neg c_n \quad (*)$$

其中 $m, n \geq 0$ 且 $m+n \geq 1, a_i \in \text{Var } \mathcal{L} \cup \{\top\} (i=1 \dots m), b_i \in \text{Var } \mathcal{L} \cup \{\perp\} (i=1 \dots m), c_j \in \text{Var } \mathcal{L} (j=1 \dots n)$ 。令 $A = \{a_1, \dots, a_n\}, B = \{b_1, \dots, b_n\}, C = \{c_1, \dots, c_m\}$ 。特别地, 定义有向图 $G = (\text{Var } \mathcal{L} \cup \{\top, \perp\}, E)$, 其中,

$$E = \{\langle a_i, b_i \rangle : 1 \leq i \leq n\}$$

引理 任一形如(*)的弱合取子式在 G_3 逻辑中有效当且仅当下述条件至少有一个成立

- (1) $A \cap C \neq \emptyset$;
- (2) G 中存在长度大于 2 的路径。

证明: $1^\circ \Rightarrow$

用反证法。不妨反设 $A \cap C = \emptyset$, 且 G 中不存在长度大于 2 的路径。令 $X = (A - B) \cap \text{Var } \mathcal{L}, Y = A \cap \text{Var } \mathcal{L}$ 。显然 $\langle X, Y \rangle$ 是合法的 G_3 解释。对任意 $i \in \{1, \dots, m\}$, 我们分如下九种情况进行分析:

- (a) $a_i = \top, b_i = \perp$;
- (b) $a_i = \top, b_i \neq \perp$ 且 $b_i \in A$;
- (c) $a_i = \top, b_i \neq \perp$ 且 $b_i \notin A$;
- (d) $a_i \neq \top$ 且 $a_i \notin B, b_i = \perp$;
- (e) $a_i \neq \top$ 且 $a_i \notin B, b_i \neq \perp$ 且 $b_i \notin A$;
- (f) $a_i \neq \top$ 且 $a_i \notin B, b_i \neq \perp$ 且 $b_i \in A$;
- (g) $a_i \neq \top$ 且 $a_i \in B, b_i = \perp$;
- (h) $a_i \neq \top$ 且 $a_i \in B, b_i \neq \perp$ 且 $b_i \notin A$;
- (i) $a_i \neq \top$ 且 $a_i \in B, b_i \neq \perp$ 且 $b_i \in A$ 。

容易看出, 在(a)~(f)六种情形中, 我们均能得到 $\langle X, Y \rangle \models a_i$, 且 $\langle X, Y \rangle \not\models b_i$, 由定义立得 $\langle X, Y \rangle \not\models a_i \rightarrow b_i$ 。情形(g)和情形(h)均有 $Y \models a_i$ 和 $Y \not\models b_i$, 故 $Y \not\models a_i \rightarrow b_i$, 由事实(F2), 同样有 $\langle X, Y \rangle \not\models a_i \rightarrow b_i$ 。情形(i)是不可能存在的, 否则由 $a_i \in B$, 存在 a 使得 $\langle a, a_i \rangle \in E$; 由 $b_i \in A$, 存在 b 使得 $\langle b_i, b \rangle \in E$ 。结合 $\langle a_i, b_i \rangle \in E$, 显然 $\langle a, a_i \rangle, \langle a_i, b_i \rangle$ 与 $\langle b_i, b \rangle$ 构成一条长度为 3 的路径。这与假设 G 中不存在长度大于 2 的路径矛盾。

对任意 $c \in C$, 由假设 $A \cap C = \emptyset$ 可知 $c \notin Y$, 故 $Y \models \neg c$, 进而 $Y \not\models \neg \neg c$ 。由事实(F3), 立即可有 $\langle X, Y \rangle \not\models \neg \neg c$ 。由此看出, $\langle X, Y \rangle$ 是该子式的反模型, 这与题设矛盾, 故(1)或(2)必有一个成立。

$2^\circ \Leftarrow$

若 $A \cap C \neq \emptyset$, 则该子式中同时含 $(a \rightarrow b)$ 与 $\neg \neg a$ 两个项。容易验证 $(a \rightarrow b) \vee \neg \neg a$ 在 G_3 逻辑中有效, 故 G 该子式也有效。若 G 中存在长度大于 2 的路径, 则子式中同时含 $(a \rightarrow b), (b \rightarrow c)$ 与 $(c \rightarrow d)$ 三项, 又由 $(a \rightarrow b) \vee (b \rightarrow c) \vee (c \rightarrow d)$ 是 G_3 逻辑中的有效式, 可知该子式必有效。 □

注释 4 判断 $A \cap C \neq \emptyset$ 可在线性时间内完成, 而判定有向图中是否存在长度大于 2 的路径也显然有多项式算法。因此判定一个任意的弱合取子式在 G_3 逻辑中是否有效存在多项式时间算法。

定理 3 若 $P \neq \text{coNP}$, 则在 G_3 逻辑中命题公式的化简问题不存在多项式算法。

证明: 判断命题公式是否有效, 我们可以分两步进行: 先将该公式化简为与之逻辑等价的范式; 然后判断该范式是否有效。由定理 1, G_3 逻辑可归约到弱合取范式, 故该方法可用于任意公式。又根据引理与注释 4, 第二步存在多项式算法。因此, 若第一步也存在多项式算法, 则判断命题公式是否有效也可在多项式时间内完成。但由命题 4, 该问题是 coNP 完全的, 故若 $P \neq \text{coNP}$, 在 G_3 逻辑中命题公式的化简问题不可能存在多项式算法。 □

参考文献

1 Avron A. Hypersequents, Logical Consequence and Intermediate Logics for Concurrency. Ann Math Artificial Intelligence, 1991, 4: 225~248

2 Cabalar P, Pearce D, Valverde A. Reducing Propositional Theories in Equilibrium Logic to Logic Programs. In: 12th Portuguese Conference on Artificial Intelligence, Covilha, 2005

3 Chagrov A, Zakharyashev M. Modal Logic. Oxford; Clarendon Press, 1997

4 Ferraris P, Lifschitz V. Mathematical Foundations of Answer Set Programming. In: We Will Show Them! Essays in Honour of Dov Gabbay, Vol 1. Artemov S, Barringer H, Garcez A, Lamb L, and Woods L, College Publications, 2005. 615~664

5 Gödel K. Zum intuitionistischen Aussagenkalkül. Anz Akad Wiss Wien, 1932, 69; 65~66

6 Hähle R. Complexity of Many-Valued Logics. In: 31st International Symposium on Multiple-Valued Logics, Warsaw, 2001

7 Lifschitz V, Pearce D, Valverde A. Strongly Equivalent Logic Programs. ACM Transactions on Computational Logic, 2001, 2(4); 526~541

8 Pearce D. A New Logical Characterisation of Stable Models and Answer Sets. Non-Monotonic Extensions of Logic Programming, Bad Honnef, 1996

9 Takeuti G, Titani T. Intuitionistic Fuzzy Logic and Intuitionistic Fuzzy Set Theory. Journal of Symbolic Logic, 1984, 49(3); 851~866

(上接第 120 页)

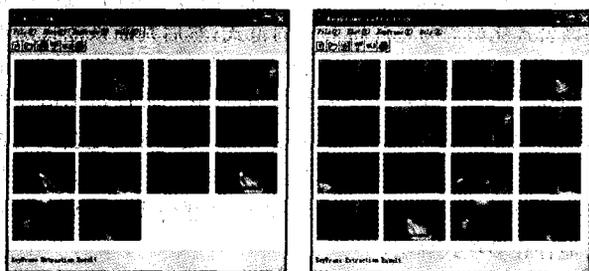
取系统,选用了五个具有不同特点的视频片段组成本系统的实验数据集,包括:爱情片《邂逅》中相聚的片段(XH)、儿童片《绿野仙踪》中庆祝坏女巫死亡的片段(LY)、体育比赛中“篮球比赛”的结尾片段(LQ)、晚间新闻中的反映植树造林片段(XW)和 MTV《赞国歌》中歌伴舞片段(ZJ)。视频序列从 4 千多帧到 2 万多帧,每帧为 352×240 像素,其基本情况如表 1 所示。

表 1 实验数据集

视频片段	帧数	镜头数	本文算法检测出的关键帧数	算法 B 检测出的关键帧数	人工检测的关键帧数
XH	25430	66	85	86	85
LY	17928	73	99	101	99
LQ	9010	29	47	48	47
XW	4558	20	41	41	41
ZJ	7893	23	57	56	55

3.2 实验结果和分析

文[1](记为算法 B)是比较当前帧与所有已提取关键帧之间的差异程度,从而确定当前帧可否作为关键帧,这种方法的结果使得各个关键帧的内容之间最大限度的不同,从而更有代表性。为了更好地评价本文提出的方法,我们和该方法进行比较分析。图 2 是本文算法和算法 B 的对比实验结果,所使用的 352×240 原始图像取自《邂逅》中的一个视频序列共 3542 帧,图 2(a)是带宽 $h(h_c, h_m) = (18, 0.25)$ 本文算法的检测结果,本算法提取了 14 帧关键帧,通过关键帧和字幕文字,我们能直观地和概括地了解这段视频发生了哪些事,有哪些人物。实验结果表明本文算法能够很好地利用帧间的时序关系和相似度来提取关键帧,从而达到较好的效果,其鲁棒性表现在大物体的快速运动上。图 2(b)是阈值为 3 的算法 B 的检测结果,虽然可以获得同本文算法比较接近的效果,但是算法 B 需要进行的阈值选取是点取值,它的微小差别对关键帧提取结果影响很大,阈值的选取难以把握,而本文算法采用的是均值漂移聚类方法,它所需设置的带宽 h 是范围取值,比算法 B 的阈值选取易于控制。



(a) 带宽 $(h_c, h_m) = (18, 0.25)$

(b) 阈值为 3

图 2 关键帧提取的结果

这两种算法提取的关键帧都没有遗漏,但有少量冗余,这是由于光照较强的序列容易出现冗余帧。例如图 3 选用了 MTV 中的一个视频序列共 1879 帧,使用本文算法提取了 12 帧关键帧,显然最后一帧为冗余帧,原因在于舞台灯光的变化,其亮度变化很大,但是场景中人物的变化却由于时间短暂而变化很小(即舞台灯光变化前后的图像帧内容很相似),舞台灯光变化前后分别提取了关键帧,从而产生了 1 帧冗余帧。



图 3 关键帧提取的结果

结论 本文提出了一种基于非参数密度估计聚类关键帧提取的算法,由于采用均值漂移聚类的方法,因此算法能够自动确定类别并且只需用户设置带宽 $h = \{h_c, h_m\}$ 即可。它综合利用了图像的颜色特征和运动特征,从而提高了关键帧提取的有效性。通过一些视频序列的提取实验和实验分析对比,我们认为,本文提出的提取算法对于无强光照射的图像具有很好的提取效果,实验的提取结果与人的视觉具有良好的—致性。当然,该算法还有由于光照较强而产生冗余帧的问题,一个可能的途径是将图像的光照不均的校正技术融入算法中,可以使提取关键帧的有效性得到进一步的提高。

致谢 作者现在在北京交通大学访问学习,此论文是在北京交通大学计算机学院须德教授和郎丛妍博士的指导下完成的,在此一并表示感谢!

参考文献

1 Zhu X Q, Wu X D, Fan J P, et al. Exploring video content structure for hierarchical summarization. Multimedia Systems, 2004, 10(2): 98~115

2 Toklu C, Liou S P. Automatic keyframe selection for content-based video indexing and access. In: Proc. of SPIE. 2000, 3972: 554~563

3 Grabmeier J, Rudolph A. Techniques of cluster algorithms in data mining. Data Mining and Knowledge Discovery, 2002, 6(4): 303~360

4 Kim C, Hwang J N. An integrated scheme for object-based video abstraction. In: Proc. of the ACM Int. Conf. on Multimedia. 2000, 303~311

5 Kin-Wai S, Kin-Man L, Guoping Q. A new key frame representation for video segment retrieval. IEEE Trans. On Circuits and Systems for Video Technology, 2005, 15(9): 1148~1155

6 Comaniciu D, Meer P. Mean Shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell, 2002, 24(5): 603~619