arXiv:2304.03089v1 [cs.SC] 6 Apr 2023

结合上下文无关文法和 OWL 本体进行序列分类

NicolasLazzari[1], [3][0000−0002−1601−7689], Andrea Poltronieri [2][0000−0003−3848−7574]和 Valentina Presutti[1⋆][0000−0002−9380−5160]

[1] LILEC，博洛尼亚大学，意大利 nicolas.lazzari2@studio.unibo.it, valentina.presutti@unibo.it[2] 计算机科学与工程系，博洛尼亚大学，意大利 andrea.poltronieri2@unibo.it[3] 计算机科学系，比萨大学，意大利

摘要。 本文介绍了一种将上下文无关文法形式化为 OWL 的模式，以及该模式在序列分类中的应用。本文从计算复杂度和实际适用性两个方面，将所提出的方法与现有方法进行了比较，并给出了音乐领域的示例。

关键词： 序列分类 · 上下文无关文法 · 本体 · 音乐

1 引言

乔姆斯基在20世纪50年代引入形式文法 [7], ，尤其是上下文无关文法（上下文无关文法），推动了自然语言处理领域的大量研究。基于统计语言建模的方法已基本取代形式文法，不过由于许多定义域和任务都能从其应用中受益，该领域的研究仍然具有相关性。例如，形式文法的一个重要应用涉及高级编程语言。通过高效的解析过程 [1], ，机器级指令可以抽象为人类可读指令。从理论上讲，每个可以抽象为<样式id='3'>符号序列</样式>的问题都可以用形式文法建模，这使其成为进行<样式 id='5'>序列分类</样式>的合适工具，而序列分类正是本文关注的任务。在生物学领域，RNA二级结构的分类已使用上下文无关文法完成 [9, 15, 35]。类似地，在音乐领域，上下文无关文法被用于对不同类型的和声序列和旋律序列进行分类 [3, 24, 36, 39, 41, 42]。

背景。 一种语言 是由序列组成的集合，其中每个序列都根据符号的有限集 [8] 定义。一种文法可以解释为语言的一个函数，其定义域为一个符号集，值域为一个序列集。定义形式文法是一项复杂的任务，需要深入了解应用领域，同时具备良好的建模技能，以获得

⋆ 字母顺序

一种可高效解析的文法。CFG 可以在少于 $O(n^3)$ [43],with n 字符串长度的时间内完成解析。然而，当符号具有歧义语义并需要额外属性来消除歧义时，解析一个序列是 NP完全的 [14]。歧义符号是一个常见问题，例如自然语言中的多义词或音乐中的减和弦。可以通过使用复杂的记法来缓解这一问题，例如使用 SMILES [44] 来表示生物学领域中的分子，或使用 Harte [18] 来表示音乐和弦。不过，这些记法要么难以解释，需要借助额外工具将其转换回人类可理解的格式，要么会造成信息丢失。

我们通过提出一种结合上下文无关文法（CFG）解析器与 OWL 本体的混合方法，解决序列分类问题。我们定义了一种模式，通过基于描述逻辑对上下文无关文法作出全新定义，对其进行形式化。我们还定义了一组算法，可基于该模式生成 OWL 本体，并支持将序列中的符号与其所属的类对齐。我们的方法依据本体中定义的分类体系识别子序列。我们认为，这一方案具有重要的实践潜力，因为它能够基于语义网知识表示进行序列分类，从而支持将上下文无关文法与 Web 本体及知识图谱相连接。

本研究的贡献可概括如下：

– 基于描述逻辑定义一种新颖的上下文无关文法形式化；– 提供将此类形式化转换为 OWL 的算法；– 证明所提出方法的正确性、计算复杂度及其在音乐领域中的适用性。

本文的组织结构如下：第2节概述序列分类的相关研究。第3节介绍上下文无关文法的相关定义，这些定义将在第4节中用于描述在描述逻辑中对上下文无关文法的形式化。第5节介绍我们针对序列分类的方法。第6节中，我们在音乐领域的序列分类任务上评估该方法。最后，第7节总结本文的贡献并讨论未来的发展。

2 相关工作

与本贡献相关的工作包括：序列分类技术、结合文法的序列模型、整合上下文无关文法和语义网技术的方法，以及它们在音乐领域中的应用（参见第6节）。

序列分类 (SQ)是一项预测输入所属类别的任务，其中输入被定义为随时间或空间变化的序列，类别来自预定义的类别集合 [31]。SQ 与多个应用领域相关，例如基因组学研究 [26], 健康

信息学、异常检测和信息检索 [47],自然语言处理（NLP） [6, 28]。在 [47] 中，确定了 SQ的不同方法，例如基于特征的分类、基于序列距离的分类以及支持向量机（SVM）。最先进的方法主要依赖于深度学习（描述逻辑） [5, 30]。SQ与音乐领域相关，因为序列是其核心，例如跨越时间维度的旋律序列和和声序列。应用于音乐的序列分类示例是 [31], ，该方法使用循环神经网络（LST M-RNN）来识别 拉格 。

基于文法的序列模型。 序列与形式文法之间有着密切的关系。一项经典的相关任务（涵盖自然语言处理 [27]到生物信息学 [9]）是文法推断 [49]。文法用于对不同类型的序列进行分类，主要用于分析遗传序列 [15, 35]。文法也应用于音乐分类。最著名的例子是调性音乐生成理论（GTTM） [29], ，它与乔姆斯基的转换文法或生成文法类似。 [7]尽管 GTTM 并未明确提供生成文法规则，但这项工作启发了大量上下文无关规则的形式化，用于描述不同的音乐流派 [24, 39, 41], 、旋律 [3] 和和声 [36, 42]。这些研究与我们的工作相关，因为它们将某些类型序列的特征形式化为规则。

使用 OWL 的序列模型。有一些提案建议使用 OWL 对序列进行分类。然而，在 OWL 中处理序列时，主要挑战之一是以有序方式组织所描述的元素，正如 [12] 中所提出的那样。例如，OWL 推理被用于对基因组数据进行分类 [46]。 [33] 提出了一种分析爵士和弦序列的方法。该系统基于一个本体，通过推理生成分层的爵士序列分析。同样， [45] 展示了两个 OWL 本体 $\mathcal{MEO}$ 和 $\mathcal{SEQ}$, 它们与 CFG 解析器结合，支持序列分类。然而，该方法只能表示 可安全连接 的上下文无关文法，而我们的方法克服了这一限制（参见第 5.3 节）。

3 预备知识

本节介绍第 4 节中使用的符号和定义，详情请参阅 [20] 。

定义1（上下文无关文法）。上下文无关文法（*CFG*）$G = (V, \Sigma, R, S)$ 由有限个非终结符 V（变量）、一组终结符 Σ （满足 $\Sigma \cap V = \emptyset$）、一组函数 $R \subseteq V \times (V \times R)^*$（产生式）以及一个起始符号 $S \in V$组成。

定义 2（文法的语言）。文法 $G(V, \Sigma, R, S)$的语言定义为 $L(G) = \{w \in \Sigma^* : S \overset{*}{\Rightarrow} w\}$, 其中 $S \overset{*}{\Rightarrow} w$ 表示连续应用产生式

从初始符号 S 开始，这称为推导。

示例*1*（上下文无关文法）。令 $G = (V, \Sigma, R, S)$，其中

$$
\begin{aligned}
V &= \{\text{Expression}, \text{Bit}\} \\
\Sigma &= \{0, 1, +\} \\
R &= \{\text{Expression} \to \text{Expression} + \text{Expression} \mid \text{Bit } 0 \mid \text{Bit } 1 \mid 0 \mid 1, \\
&\qquad \text{Bit} \to \text{Bit } 0 \mid \text{Bit } 1 \mid 0 \mid 1 \ \} \\
S &= \ \ \text{Expression}
\end{aligned}
$$

其中，$X \to X_1 | \cdots | X_n$ 是 $\{X \to X_1, \ \cdots \ X \to X_n\}$的简写

示例1展示了一个用于解析两个二进制数之和的简单文法。根据定义 2，其语言 L 具有以下形式：$L = \{0+0, 0+1, \ 1+0, \ \ 10+0, \ \cdots, \ 11010+10, \ \cdots\}$。

为了表达一种简洁有效的转换方法及其相应证明，我们只考虑乔姆斯基范式文法，如定义 3 所定义。这样便得到如下形式的齐次产生式：

$$A \to BC \quad \text{or} \quad A \to t$$

其中，A，B，$C \in V$和 $t \in \Sigma$。

定义3（乔姆斯基范式）。上下文无关文法处于乔姆斯基范式（乔姆斯基范式）当且仅当函数集合为 $R \subseteq V \times (V \backslash \{S\} \times R)$ [2]。

请注意，这一限制并不会降低表达能力，因为任何上下文无关文法都可以转换为乔姆斯基范式 [20]。例如，示例1转换为乔姆斯基范式后，会得到示例2中的文法。

示例*2*（示例*1 in* 乔姆斯基范式）. 令 $G = (V, \Sigma, R, S)$ 满足

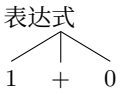
$V = \{$表达式,表达式$_0$, 位, 零, 一, 加$\}$$\Sigma = \{0, 1, +\}$$R = \{$表达式 $\to$ 表达式 $_0$ 表达式 $\mid$位零 $\mid$位 一 $\mid 0 \mid 1$,表达式$_0$ $\to$表达式加,位 $\to$位零 $\mid$位一 $\mid 0 \mid 1$,加 $\to +$,零 $\to 0$,一 $\to 1\}$

根据文法解析语言时，将推导过程可视化为解析树很有用。

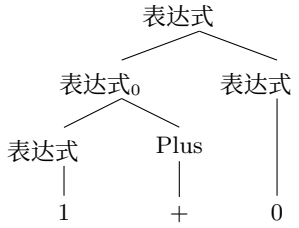
定义4（上下文无关文法的解析树）。一个解析树 T由 $G = (V, \Sigma, R, S)$构成，它是一棵树，其中每个叶节点 $l \in (V \cup \Sigma)$ 以及每个内部节点 $n_i \in V$。给定 $c_1 \cdots c_n$一个内部节点的子节点 n_i，则 $\exists f \in R$满足 f： $n_i \to c_1 \cdots c_n$。 *[20]*

推论 1. 给定 T 上下文无关文法的乔姆斯基范式中的解析树 $\Rightarrow T$ 是一棵二叉树。

推论1源于定义3和4，因为每个产生式要么是一元函数，要么是二元函数。图1a展示了由示例1定义的文法生成的序列 *1+0*的解析树，图1b展示了由示例2定义的文法生成的解析树。



（a）由以下内容生成的解析树
示例2中的文法



(b) 由以下内容生成的解析树
示例2中的文法

图1：从序列 *1+0*获得的解析树

4 使用描述逻辑形式化上下文无关文法

由于 OWL 基于描述逻辑（DL）理论，我们定义了一种基于 DL 的上下文无关文法（乔姆斯基范式）形式化表示，并将其称为 CFG-DL。CFG-DL 基于定义1，其中变量和终结符表示为概念[4]。我们证明了任意上下文无关文法都可以转换为 CFG-DL（参见定理1），且此类转换可以在 $O(n)$ 内完成（参见定理 2）。定理1及其证明依赖于角色化这一概念，它将以如果—那么 [25]形式充当规则的公理形式化。对于每个概念 C，都会创建相应的公理 R_C，并施加限制 $C \equiv R_C.Self$。通过将不同的公理串联起来，可以定义如果—那么 规则。关于更深入的说明，请参阅 Krisnadhi等人 [25]。

定义5（CFG-DL）。一个 *CFG-DL* $G_{DL} = (C_v, R, C_\Sigma, S)$ 由一个概念有限集 C_v、一个概念有限集 C_Σ、一组公理 R以及一个起始概念 $S \in C_v$组成。

定理1. 每个采用乔姆斯基范式的上下文无关文法 G 都可以转换为 *CFG-DL* G_{DL}。

[4] 描述逻辑概念转换为 OWL 类

证明。给定一个处于乔姆斯基范式的上下文无关文法 $G = (V, \Sigma, R, S)$，我们可以如下获得相应的 $G_{DL} = (C'_v, R', C'_\Sigma, S')$：

1. $\forall v \in V$ 令 C_v 为一个概念，使得 $C_v \sqsubseteq C'_v \Longrightarrow \forall v \in V \ \exists C_v \sqsubseteq C'_v$，其中 C_v 是变量V所对应的概念。2. $\forall t \in \Sigma$ 令 C_t 为一个概念，使得 $C_t \sqsubseteq C'_\Sigma \Longrightarrow \forall t \in \Sigma \ \exists C_t \sqsubseteq C'_\Sigma$，其中 C_t 是终结符 t所对应的概念。3. 令 $f \in R$。根据定义 3，f 属于以下类型之一：(a) $R \to AB$，使得 $R \in V$、A、$B \in V \cup \Sigma$。(b) $R \to t$，使得 $R \in V$和 $t \in \Sigma$；两种情况分别可在描述逻辑中表示如下：(a) i. 令 $C_R \sqsubseteq C'_v$、$C_A \sqsubseteq C'_v$、$C_B \sqsubseteq C'_v$分别为步骤 1 中定义的R、A、B所对应的概念 ii. 令 R_R、R_A、R_B为概念 C_R、C_A、C_B的角色化 [25]，使得 $C_R \equiv \exists R_R.Self$、$C_A \equiv \exists R_A.Self$以及 $C_B \equiv \exists R_B.Self$。iii. 令 R_{next}为角色，使得 $C_1 \circ R_{next} \circ C_2$具有语义含义：$C_1$ 在序列 C_2中作为下一个元素，其中 $C_1 \sqsubseteq C'$ 和 $C_2 \sqsubseteq C'$iv. 令 $V_1 \equiv \exists R_1.Self$ 和 $V_2 \equiv \exists R_2.Self$ 为角色，使得 $R_A \circ R_{next} \circ R_B \sqsubseteq R_1$和 $R_B \circ R_{next}^{-1} \circ R_A \sqsubseteq R_2$。$\xrightarrow{3\ a\ i\ 3\ a\ ii\ 3\ a\ iii\ 3\ a\ iv} R \to AB \Longleftrightarrow$ ( )、( )、( )、( ) $(C_A \sqcap V_1) \sqcup (C_B \sqcap V_2) \sqsubseteq C_R$。(b) 令 $C_t \sqsubseteq C'_\Sigma$ 为步骤 2 中定义的终结符 t 的概念，令$C_R \sqsubseteq C'_V$ 为步骤 1 中定义的变量 R 的概念$\Rightarrow R \to t \Longleftrightarrow C_t \sqsubseteq C_r \xrightarrow{3a\ 3b} f \in R'$、$\forall f \in R$。4. $\xrightarrow{1} \exists C_s \sqsubseteq C'_V$，其中 C_s 是与 S相对应的概念，如步骤 1 中所定义。$\xrightarrow{1\ 2\ 3\ 4} G_{DL} \equiv G'$、、、。

■

定理 2。 *CFG* $G = (V, \Sigma, R, S)$ 与 *CFG-DL* $G_{DL} = (C'_v, R', C'_\Sigma, S')$ 之间的转换可以在 $O(n)$内完成，具体而言为 $O(|V| + |\Sigma| + |R|)$。

证明。 由定理1的证明可知，我们只需对 V、Σ 和 R 中的每个元素最多遍历一次。■

我们指出，在定义5中，终结符被建模为概念，并且与变量处于同一层级。乍一看，将终结符表示为个体似乎更直观。但这会彻底改变序列中元素的语义。以示例1中文法的语言中的序列 *10+11* 为例。终结符*1*出现了三次，但它们本质上是不同的实体：终结符的第一次出现

1 既具有句法特征，也有相对于整个序列的位置特征。如果我们将每个终结符表示为个体，那么该终结符在序列中的每次出现都会被表示为同一个个体。这会使整个序列的语义失效，并导致形式化结果错误。为了解决这个问题，我们需要恰当地定义如何在描述逻辑中表示序列。我们通过将定义2调整为适用于CFG-DL来实现这一点。

定义6（CFG-DL 的语言）。设 $G = (C, R, C_\Sigma, S)$ 为一个 *CFG-DL*，我们将 $L(G)$ 定义为满足以下条件的序列集合 s：给定 N 序列中的元素数量 s，$s \equiv (C_1 \sqcap \exists R_{next}.C_2) \sqcap \cdots \sqcap (C_{N-1} \sqcap \exists R_{next}.C_N)$，其中 R_{next} 是定理*1*证明中步骤*3(a)iii*定义的角色，且 $C_t \sqsubseteq C_\Sigma, t \in [1, N]$。

5 使用 CFG-DL 进行序列分类

CFG-DL 可以用 OWL 表示，因为 OWL2 直接语义基于描述逻辑 [22]。我们根据定义5以及定理1的相应构造性证明4设计了一种算法。算法1将上下文无关文法转换为 OWL，而无需生成任何中间 CFG-DL。遵循构造性证明4，也可以定义一种类似的算法，将上下文无关文法转换为 CFG-DL。

三元组采用曼彻斯特语法编写 [21]。我们用符号 ▶ 表示需要创建的 OWL 三元组。通常，我们使用 R_{next} 作为在定理 1 证明的步骤 3(a)iii 中定义的角色R_{next} 。只要是函数型属性，任意 OWL 属性都可以用作 R_{next} ，例如来自序列：直接前驱 属性的序列 本体设计模式 [19]。算法 1 的复杂度同样为 $O(n)$：与定理 2 中的考量类似，我们只需遍历 V、Σ 和 R 中的每个元素，且最多遍历一次。

通过对 owl:Thing 使用存在量词限制，实现对类 V_1 和 V_2 的角色化。这样可以避免由于算法后续使用属性链而创建非简单属性，并允许使用 Hermit [16] 或 Pellet [37]等推理器。

序列必须经过转换，才能用于通过算法1获得的本体中。算法2给出了一个在$O(n)$中执行此类转换的算法。该算法基于定义6。与算法1类似，我们使用符号 ▶ 以曼彻斯特语法表示三元组，并使用 R_{next} 作为定理1证明的步骤3(a)iii 中定义的角色 R_{next}。

图2展示了示例2中的文法，该文法通过算法1转换为OWL中的CFG-DL，并用于解析使用算法2转换的序列 *1 + 0*。我们可以看到，整个序列被正确分类为表达式类，并且Expression 0和Expression被分类为彼此的子类。事实上，Expression 0和Expression是等价的。这可以从对示例1执行规范化过程后得到示例2的结果中观察到：*Expression0* 变量被引入以获得

算法1 OWL 中的上下文无关文法

要求： $G = (V, \Sigma, R, S)$
▶对象属性：R_1
▶对象属性：R_2
▶ 类：V_1 等价于：R_1 某个
▶ 类：V_2 等价于：R_2 some
对于 $v \in V$，**执行**
▶对象属性：R_v
▶类：C_v 等价于：R_v some Self
结束循环
对于 $t \in \Sigma$，**执行**
▶对象属性：R
▶类：C_t 等价于：R_t 某个 Self
结束 for
对于 $r \in R$ **执行**
如果 r 的类型为 $R \to AB$ **则**
with C_R, C_A, C_B *being the respective concepts of* R, A, B
with R_A, R_B *being the respective rolification of* A, B
▶ 对象属性：R_1 子属性链：R_A o R_{next} o R_B
▶ 对象属性：R_2 子属性链：R_B o inverse(R_{next}) o R_A
▶ (C_A 且 V_1) 或 (C_B 且 V_2) 子类：C_R
否则如果 r 属于类型 $R \to t$ **则**
with C_R, C_t *being the respective concepts of* R, t
▶ 类：C_t 子类：C_R
结束 if
结束 for

算法2 CFG-DL 中的 OWL 序列

要求： $G = (V, \Sigma, R, S)$
要求： $s \subseteq \Sigma^*$ 要表示的序列
要求： N 序列的长度 s
对 $i \in [1, N-1]$ **执行**
$s_i \leftarrow s[i]$
$s_n \leftarrow s[i+1]$
with C_i, C_n *being the respective concepts of the terminals* s_i, s_n
▶ 个体：s_n 类型：C_n
▶ 个体：s_i 类型：C_i 事实：R_{next} s_n
结束循环

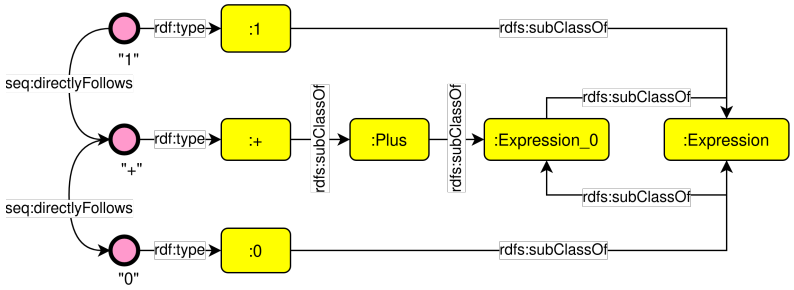


图2：由示例2中的文法解析的序列 *1+0* ，在 OWL 中表示为 CFG-DL

按 CNF 的要求，表达式 → 表达式 + 表达式 是一个二元投影。如果将每个 表达式 *0* 都替换为其右侧部分（表达式 + 表达式），则可得到一个等价但不符合 CNF 的文法。图3中的整体模式可以推广到转换为 OWL 的每个上下文无关文法。

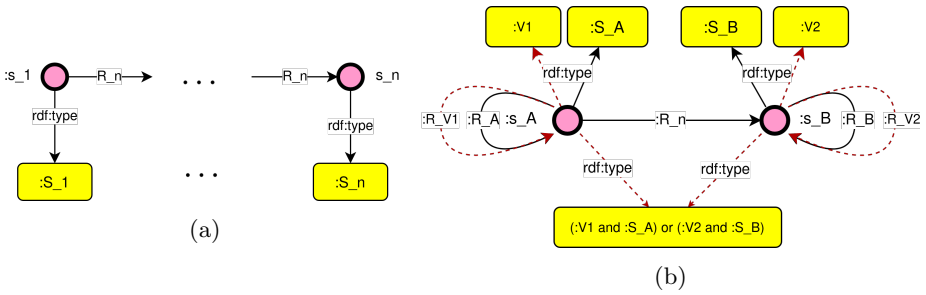


图3：算法1（3a）和算法2（3b）中针对 $R \to A\ B$类型规则的一般本体模式。红色箭头由推理器创建，这是由于属性链和一般公理的定义。$R \to t$ 类型规则的模式是一个简单的包含关系。

5.1 计算复杂度

算法1生成的本体属于DL $\mathcal{SROIEL}$，而DL $\mathcal{SROIEL}$包含于OWL 2 DL [25] 中。其自动推理的复杂度为指数级（非确定性指数时间） [23]。通过*DL*推理器解析序列，与使用外部解析器相比，复杂度过高：即使在二义性文法的情况下，解析上下文无关文法的复杂度也为 $O(n^3)$ [43] [13]。如果不使用逆属性，生成的本体属于DL $\mathcal{SROEL}$，处于OWL-EL之内，且可在

多项式时间内求解 [23]。为降低整体复杂度，我们提出一种结合上下文无关文法解析器和OWL推理的混合方法，以执行序列分类。

5.2 将上下文无关文法解析器与基于 OWL 的推理相结合

我们主张，将上下文无关文法转换为CFG-DL，除了是一种有趣的理论方法之外，也构成了一种相关的实用方法，用于执行能够受益于显式知识表示的自动序列分类，并使用OWL 本体。在实践中，给定一个上下文无关文法 G，使用 G的解析器识别序列 $s \in L(G)$后，可以使用算法3将所得的解析树转换为实例化由算法1产生的 OWL 本体 O ，用于 G。

算法3 OWL中的解析树

要求： $G = (V, \Sigma, R, S)$ 采用乔姆斯基范式
要求： $s \subseteq \Sigma^*$ 使用 $s \in L(G)$
要求： T 通过使用 G解析 s 所获得的解析树
确保： T 是一棵二叉树
  对于所有 叶子 l 在 R **中执行**
    with $\overline{C}_l$ *being the concept of the terminal* l
    对所有祖先 a 的 l **执行**
      with $\overline{C}_a$ *being the concept of the variable* a
      ▶ 类: C_l 子类: C_a
    结束 for
  结束 for 循环

算法3基于定义4和定义1。

序列 s 现在可以由DL推理器根据 O 中的类-或任何与 O对齐的其他本体进行分类。第6节通过音乐领域中的一个用例展示了这一过程。

同样的方法可用于转换由神经网络词性标注 [2, 4] 或成分句法分析 [32, 40, 48]等算法生成的解析树。

5.3 与 $\mathcal{SEQ}$ 的比较

[45] 中提出了 $\mathcal{SEQ}$，这是一种用于通过描述逻辑和 OWL 对元素序列建模的本体模式。该方法通过包含关系识别子序列，从而进行序列分类：待分类的序列包含一组模式（子序列），这些模式用于对该序列进行分类。作者展示了该方法只能表示可安全连接的 上下文无关文法。如果一个上下文无关文法的产生式具有 $R \to t_1 \cdots t_n X$这种形式，且 X， $R \in V$，则该文法是可安全连接的

且 $t_1 \cdots t_n \in \Sigma$ [45]。 V和 Σ的定义与定义1相同。这些限制阻碍了自嵌入 文法 [45],的表示，这类文法包含形如 $R \to \alpha R \beta$的产生式，且 $R \in V$ 和α， $\beta \in (V \cup \Sigma)$ [8]。我们的方案通过直接反映产生式的语义克服了这一局限，如定理1的证明4所示。

6 实验

在本节中，我们将方法应用于音乐领域 [5] ，以自动分析和声进行。和声进行被定义为和弦序列，其分析包括评估每个和弦的潜在功能 [33]。传统上，这项工作由受过训练的音乐家完成，因为需要对音乐领域有深入的知识和理解。分析结果是否正确取决于所使用的分类体系以及分析序列时所处的上下文（例如，音乐流派）。

在音乐理论中，和声是一个经过充分研究的领域，已有多种分类体系被提出以完成这项任务 [36]。大多数方法依据西方音乐理论，根据每个和弦的调性功能进行分类，使用上下文无关文法 [11, 36] 或概率上下文无关文法 [17]。这些文法的实现可能存在问题，并依赖不同的技术，例如 [11] 中的 Haskell 数据类型，或 [17] 中对上下文无关文法定义的扩展。在 [24] 中，使用上下文无关文法检测称为乐句砖块的子序列。乐句砖块是和弦序列的类别。它们的组合可定义新的乐句砖块。 [33], 探索了类似的方法，其中乐句砖块（称为惯用型）的定义是通过使用面向对象编程实现的树状层级本体来完成的。

我们的实验基于 [24], 实现的一部分规则，并使用算法1将其转换为 OWL 本体。

6.1 文法子集

[24] 中定义的上下文无关文法可形式化为 $G_k = (V, \Sigma, R, S)$，其中变量集 V 是将从和声进行中提取的子序列集合， Σ 是和弦集合， R 是产生式集合，用于将每个序列映射到相应的和弦集合。起始 S 可分配给一个特殊变量 $V_s \in V$，使得 $\forall\, t \in \Sigma\ \exists f \in R : \ s.t.\ f(V_s) = t$。为获得一个更易处理的示例，我们从完整文法 G_k中提取一个子集：我们只使用足以分析*Blue Bossa*（作者为*Dorham Kenny*）的变量、终结符和产生式。随后，我们扩展文法，加入其他一些不应出现在最终分析中的产生式，以考察本体对基于文法的方法的反映有多准确。对 [24] 的分析与我们的结果相吻合，为该方法的正确性提供了实证依据。

[5] 实验代码可在以下地址获取：https://github.com/n28div/CFGOwl 采用知识共享署名许可协议发布。

开关小调IV Cm → 小调开启 Cm 关闭F小调开启 Cm → C:min | C:minmaj7 | C:min6 | C:min7 关闭 F → F:7 | F | F:maj | F:min | F:min7 | F:minmaj7 | F:dim7悲伤终止 Cm → 悲伤趋近Cm小调开启 Cm|悲伤趋近 Cm 小调开启 Cm|F:7(#11)小调完全终止Cm 悲伤趋近 Cm → D:hdim7G:7 小调完全终止Cm → G:7C:min7 直进终止 Db → 直进 Db Db|直进 DbDb:maj7 直进 Db → Eb:min7 直进 C0| Eb:min7 Ab:7 直进 C 0 → Ab:7 C:7/Bb

图4：文法 $G_{k_1} \subseteq G_k$ 的产生式。这里只列出产生式。终结符集和变量集与产生式中使用的相同。

图4展示了对严格用于分类 $BlueBossa$ 在 [24] 中演奏的规则进行形式化的结果。使用算法1，我们将图4中的文法转换为 OWL。生成的本体包含130条公理。使用算法2，我们将取自 [24] 的$Blue\ Bossa$和弦标注转换为 OWL。生成的本体总共包含29条公理。将这两个本体合并后，我们得到一个总共包含159条公理的最终本体。该本体能够正确解析该序列，如表1所示。

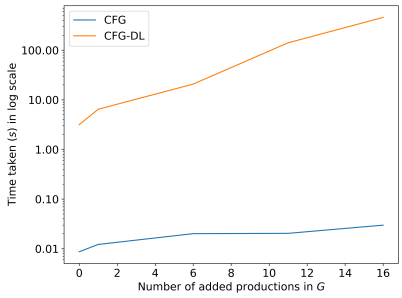
6.2 推理复杂度

如第5.1节所述，使用 CFG-DL 解析序列的计算复杂度为指数级。在图5a中，我们使用第6.1节定义的文法解析歌曲$Blue\ Bossa$。随后，我们逐步添加随机产生式，这些产生式不会影响分类。每次迭代添加5个新产生式，其右部数量从范围 $[1, 10]$ 中随机抽取。每个产生式有 $R \to AB$ 80% 的概率属于一种类型，有 $R \to t$ 20% 的概率属于另一种类型，以反映 $R \to AB$ 产生式出现频率更高这一事实，尤其是在上下文无关文法采用乔姆斯基范式表示时。

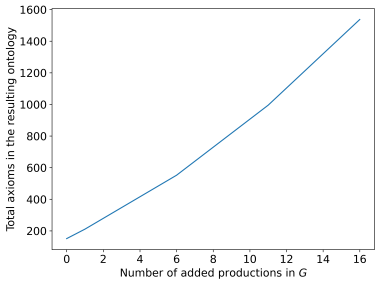
在图 5 中展示了所述实验的实证结果。图 5a 展示了随着向文法中添加产生式，CFG-DL 的时间复杂度呈指数增长。这是由于添加新产生式会导致公理数量按比例增加（图 5b）。采用第 5.2 节所述的混合方法时，计算复杂度要低得多。在图 5a 中，对一个序列进行分类所需的时间显著减少。全部

表 1：使用图 4 的文法对 *Blue Bossa* 的和声进行进行解析的结果。由 [24] 识别出的类以粗体文本表示。

和弦	推断出的类
C:min7	**OnOff小调IV Cm** 变量一 C:min7 小调开启 Cm _
F:min7	**开启小调IV级 Cm** 变量二 F:min7 关闭 F _
D:hdim7	变量一 悲伤趋近 Cm **悲伤终止式 Cm** D:hdim7_
G:7	小调完美终止式 Cm _ 变量二 变量一 G:7 忧伤进行曲 Cm**忧伤终止式 Cm** _
C:minmaj7	C:minmaj7 变量二 小调开启 Cm **忧伤终止式 Cm** _
Eb:min7	变量一 直进 Db _ **直进终止式 Db** _ Eb:min7
Ab:7	变量二 变量一 直进 Db _ Ab:7 **直进终止式 Db**直进 C 0 _ _
Db:maj7	变量二 Db:maj7**直行终止式 Db** _
D:hdim7	变量一 悲伤方式 Cm **悲伤终止式 Cm** D:hdim7_
G:7	小调完美终止式 Cm _ VariableTwo 变量一 G:7 SadApproach CmSadCadence Cm
C:minmaj7	C:minmaj7 VariableTwo 小调On Cm **SadCadence Cm**



(a)随着向文法中添加随机产生式，所用时间（纵轴，对数刻度）的变化。将使用描述逻辑进行解析与在相应上下文无关文法上使用 Earley解析器进行比较。混合方法所用的时间是使用描述逻辑解析的 200%（相差 3 个数量级）。



(b) 本体中的公理总数（y 轴）随文法中随机添加的产生式数量（x 轴）变化的情况。随着文法中添加的产生式越来越多，例如 N，本体中大约会插入 $10N$ 条公理。由于计算复杂度直接取决于本体中的公理数量，因此所得的 CFG-DL 在实际应用中效率低下。

图 5：使用 CFG-DL 解析歌曲 *Blue Bossa* 时计算复杂度的实证结果。

实验使用 Pellet 推理器 [37]，在一台配备 $2.4GHz$ $Inteli5\text{-}6300U$ CPU 和 $8GB$ 内存的设备上，在常规计算负载下执行。

尽管两种方法的结果无法区分，但需要注意的是，如果修改序列，就必须再次执行算法3；而使用算法1生成的 CFG-DL 则能够在无需额外操作的情况下对新元素进行分类。我们计划在未来的工作中探讨这一方面，例如结合算法1和算法3。

6.3 子序列分类

要解读图4和表1中的结果，需要全面理解上下文无关文法。为了提高可解释性，只需扩展算法1生成的本体并提高抽象层次，或者与其他相关本体进行对齐。在图6的示例中，我们可以将结果与特定领域的本体（例如音乐理论本体 [34]）进行对齐。与 [33], 不同，在我们的方法中，文法和本体的定义彼此独立。

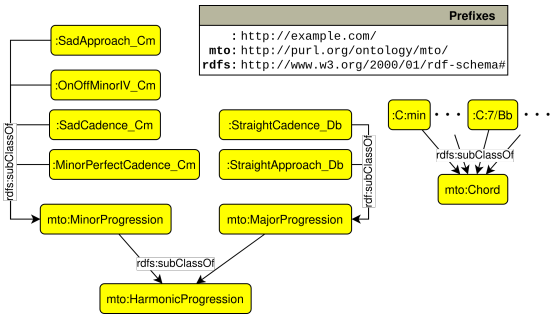


图6：算法1生成的本体所导入的本体。

表2中的分类是使用图6所示的本体得出的。与表1相比，这些结果可以说更易于解读，而且无需更新原有的上下文无关文法。通过使用算法1转换相关文法，可以根据解析过程的结果填充知识图谱。随后，还可以通过对齐其他本体来进行更多分类。例如，要将从大调进行转为小调进行的调式乐段（即从大调和弦进行转为小调和弦进行）分类为X，只需定义如下公理即可

▶((mto:MajorProgression and $\overline{V}_1$) or (mto:MinorProgression and $\overline{V}_2$)) SubClassOf: X

表 2：使用图 4 中的文法并导入图 6 所示的 Music TheoryOntology，对 *Blue Bossa* 的和声进行进行解析的结果。

弦	进展类型
C:min7	小调
F:min7	小调
D:hdim7	小调
G:7	小调
C:minmaj7	小调
Eb:min7	大调
Ab:7	大调
Db:maj	大调
D:hdim7	小调
G:7	小调
C:minmaj7	小调

7 结论

我们提出了一种新方法，使用描述逻辑对乔姆斯基范式下的上下文无关文法进行建模。正如第5.1节和第6.2节中的分析所示，其计算复杂度过高，因此不适合使用 OWL 进行序列解析。然而，如第6.3节所示，该方法能够将基于上下文无关文法的方法与语义网中常用的技术相结合。通过将 OWL 与传统解析算法结合，可以有效地表示和分类序列。这种分类方式可用于计算两个序列之间的相似度等任务。在音乐信息检索领域，推断此类相似度非常有用，因为两个和声进行之间的相似度指标难以定义。然而，同样的方法也可应用于其他使用形式文法对序列进行建模的领域，例如自然语言处理 [27], 生物信息学 [9] 和编程语言 [10]。结合上下文无关文法与 OWL 本体的混合方法，使文法建模过程得以转变：组合范畴文法（CCG）等现有扩展 [38], 旨在透明地兼顾序列的语义与句法方面。可以采用与第4节所述方法类似的方式，以描述逻辑形式化 CCG, 并构建由表达力强大的本体提供语义信息的文法。

致谢本项目获得了欧盟地平线2020研究与创新计划的资助，资助协- 议编号为 No101004746。

参考文献

1.A. V.Ahoand S. C. Johnson。LR句法分析。 *ACM Computing Surveys (CSUR)*，62 99( ): –124，1974。2. A. Akbik,D. Blythe, andR.Vollgraf。用于序列标注的上下文字符串嵌入。在 *Proceedings of the 27th international conference on computational linguistics* 1638 1649, 2018 ，第 –页。

3.M.BaroniandC. Jacoboni。 *Proposal for a grammar of melody : the Bach chorales*。蒙特利尔大学出版社，蒙特利尔，1978。4. B.Bohnet, R. T. McDonald, G.Sim˜oes, D.Andor,E.Pitler, andJ. Maynez。使用基于上下文敏感词元编码的 Meta-BiLSTM 模型进行形态句法标注。在 I.GurevychandY.Miyao，编辑， *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*， *ACL 2018*， *Melbourne, Australia 2018* 2642 2652 ， *July 15-20*，， *Volume 1: Long Papers*，第 – 页。计算语言学协会，2018。 5. R. Carrasco-Davis, G. Cabrera-Vives,F. F¨orster, P. A. Est´evez, P. Huijse, P.Pro-topapas, I. Reyes, J. Mart´ınez-Palomera,and C.Donoso。用于天文事件图像序列分类的深度学习。 *Publications of the Astronomical Society of the Pacific* 131 1004 108006, 2019 ，( ): sep。

6. T. Chen, R. Xu, Y. He,Y.Xia, and X.Wang。使用序列模型学习用于情感分析的用户和产品分布式表示。*Computational Intelligence Magazine* 11 3 34 44, 2016 *IEEE* ，(): –。

7.N. 乔姆斯基。语言描述的三个模型。 *IRE Transactions on Information Theory* 2 3 113 124, 1956 ，( ):–。8.

N. 乔姆斯基。关于文法的某些形式属性。 *Information and Control*，2 2 137 167，1959():–。9. R. Damasevicius。使用文法推断和支持向量机对调控 DNA 序列进行结构分析。 *Neurocomputing*，73(4-6):633–638，2010。 10. C. Z. de Aguiar, R. de AlmeidaFalbo,andV. E. S. Souza。OOC-O：面向对象代码的参考本体。在A. H. F. Laender,B. Pernici, E. Lim, andJ.P.M.deOliveira，编辑， *Conceptual Modeling - 38th International Conference*， *ER 2019*， *Salvador*， *Brazil*， *November 4-7*，2019， *Proceedings*， *Lecture Notes in Computer Science* 的第11788 卷，第13–27 页。Springer，2019。11.W. B. De Haas, J. P. Magalh˜aes, F. Wiering, and R. C. Veltkamp。自动功能和声分析。 *Computer Music Journal*，37(4):37–53，2013。 12. N. Drummond, A. L. Rector, R. Ste vens, G. Moulton, M. Horridge,H.Wang, andJ. Seidenberg。使 OWL井然有序：OWL中的序列模式。在 B.C. Grau, P. Hitzler,C. Shankey, andE. Wallace，编辑， *Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions*， *Athens*， *Georgia*， *USA November 10-11 2006* 216 ，，， *CEUR Workshop Proceedings* 卷。CEUR-WS.org，2006。13.J. Earley。一种高效的上下文无关句法分析算法。 *Communications of the ACM* 13 2 94 102, 1970 ，( ): –。

14.R.C. B. G. Edward Barton and E. S. Ristad。计算复杂度与自然语言。 *Journal of Linguistics*，24(2):573–575，1987。 15.R. Giegerich。随机上下文无关文法简介。 *RNA Sequence, Structure and Function: Computational and Bioinformatic Methods* 85 106, ，，第 –页。2014
16.B.Glimm, I. Horrocks, B.Motik, G. Stoilos,and Z. Wang。HermiT：一种 OWL 2 推理器。 *Journal of Automated Reasoning*，53(3):245–269，2014。

17. D. Harasim、M. Rohrmeier和 T. J.O'Donnell。结合上下文无关文法与OWL 本体的生成式和声语法模型通用解析框架。载于 *ISMIR*，第152–159 页，2018年。18. C. Harte、M. B.Sandler、S.A. Abdallah和E.G´omez。音乐和弦的符号表示：一种用于文本注释的拟议语法。载于 *ISMIR*，第 5 卷，第66–71 页，2005 年。 19. P.Hitzler、A. Gangemi、K. Janowicz、A.Krisnadhi和 V.Presutti，编。 *Ontology Engineering with Ontology Design Patterns - Foundations and Applications*，*Studies on the Semantic Web* 第25 卷。IOS Press，2016 年。 20. J. E. Hopcroft、R.Motwani和J.D. Ullman。自动机理论、语言与计算导论。 *Acm Sigact News*，32(1)：60–65，2001年。 21. M.Horridge、N. Drummond、J. Goodwin、A. L. Rector、R.Stevens 和H.Wang。Manchester OWL 语法。载于 *OWLed*，第216 卷，2006 年。 22. I.Horrocks、B.Parsia和U.Sattler。OWL 2Web本体语言直接语义。*World Wide Web Consortium*，第 42–65 页，2012 年。 23. I. Horrocks和P. F.Patel-Schneider。将 OWL蕴涵归约为描述逻辑可满足性。载于 *International semantic web conference*，第 17–29 页。Springer，2003年。 24. R. Keller、A. Schofield、A. Toman-Yih、Z.Merritt和 J. Elliott。使用惯用分析自动解释爵士和弦进行。*Computer Music Journal* 37 4 54 69, 2013 ，（）：–。

25. A.Krisnadhi 、 F.Maier 和 P. Hitzler。OWL 与规则。载于 *Reasoning Web International Summer School* 382 415 2011 ，第 –页。Springer，。 26.S.Kumar。使用 K-mer 分解和基于软计算的方法进行基因序列分类。载于 T. K. Sharma、C. W. Ahn、O. P.Verma和 B. K. Panigrahi，编，*Soft Computing: Theories and Applications*，第 181–186页，新加坡，2021 年。Springer Singapore。 27. S. Lawrence、C.Giles 和S. Fong。使用循环神经网络进行自然语言文法推断。 *IEEE Transactions on Knowledge and Data Engineering* 12 1 126 140, 2000 ，（）：–。

28. J.Lei、Q. Zhang、J.Wang和H.Luo。基于 BERT 的上下文感知微博情感分析层次化序列分类。载于T. Gedeon、K. W. Wong 和M. Lee，编， *Neural Information Processing*，第376–386 页，Cham，2019 年。SpringerInternational Publishing。29.F. Lerdahl 和 R.Jackendoff。 *A generative theory of tonal music*。The MIT Press，剑桥，马萨诸塞州，1983 年。 30. G. LoBosco 和M.A. Di Gangi。用于DNA 序列分类的深度学习架构。载于 A.Petrosino、V.Loia和 W. Pedrycz，编，*Fuzzy Logic and Soft Computing Applications* 162 171, 2017 ，第 –页，Cham，。SpringerInternational Publishing。 31. S. Madhusudhan 和 G.Chowdhary。Deepsrgm——利用深度学习进行印度古典音乐中的序列分类与排序。载于 A.Flexer、G.Peeters、J. Urbano 和 A. Volk，编，*Proceedings of the 20th International Society for Music Infor-mation Retrieval Conference, ISMIR 2019*，《第 20 届国际音乐信息检索学会会议论文集》，ISMIR2019，第533–540 页。国际音乐信息检索学会，2019年。 32. K.Mrini、F.Dernoncourt、Q. H. Tran、T.Bui、W.Chang和 N.Nakashole。重新思考自注意力：迈向神经解析的可解释性。载于T. Cohn、Y. He 和 Y.Liu，编， *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*，*Findings of ACL*的 EMNLP 2020 卷，第 731–742 页。计算语言学协会，2020 年。

33.F.Pachet。《爵士和弦序列的计算机分析：solar是布鲁斯吗？》，2000。 34.S. M. Rashid、D. DeRoure和D.L. McGuinness。《音乐理论本体》。In *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music* 6 14, 2018 ，页码 – 。 35. E. Rivas和 S.R.Eddy。《RNA的语言：一种包含伪结的形式文法》。 *Bioinform.*，16(4):334–340，2000。 36.M. Rohrmeier。《迈向调性和声的生成句法》。 *Journal of Mathematics and Music* 5 1 35 53, 2011 ，():– 。 37.E. Sirin、B. Parsia、B. C. Grau、A.Kalyanpur和 Y.Katz。《Pellet：一种实用的OWL-DL推理器》。 *Journal of Web Semantics*，5(2):51–53，2007。 38.M. Steedman 和 J. Baldridge。《组合范畴文法》。 *Non-Transformational Syntax: Formal and Explicit Models of Grammar: Wiley Blackwell* 181 224, 2011 ，页码 –。 39. M.J. Steedman。《爵士和弦序列的生成文法》。 *Music Perception: An Interdisciplinary Journal* 2 1 52 77, 1984 ，( ): – 。 40. Y. Tian、Y.Song、F. Xia 和T. Zhang。《使用跨度注意力改进成分句法分析》。InT. Cohn、Y. He和 Y. Liu，编者， *Findings of the Association for Computational Linguistics: EMNLP 2020 Online Event 16-20 November 2020* ， ， ， 2020 *Findings of ACL* 1691 1703 *Findings of the Association for Computational Linguistics: EMNLP 2020 Online Event 16-20 November 2020* 的 EMNLP 卷，页码 –。计算语言学协会，2020。 41. D.Tidhar。《 *A hierarchical and deterministic approach to music grammars and its application to unmeasured preludes* 》。博士论文，柏林工业大学，2005。 42. S. Tojo、Y. Oka 和M. Nishida。《基于 HPSG 的和弦进行分析》。In *Proceedings of the 24th AAAI Conference on Artificial Intelligence and Applications* 06, 305 310, 2006 ，AIA’ 页 – 美国， 。 ACTA Press。43.L.G. Valiant。《少于立方时间内的一般上下文无关识别》。 *Journal of computer and system sciences* 10 2 308 315, 1975 ，( ):– 。44. D.Weininger。《SMILES，一种化学语言和信息系统。1. 方法论和编码规则简介》。 *Journal of chemical information and computer sciences* 28 1 31 36, 1988 ，( ): –。 45.J. Wissmann。《 *Chord Sequence patterns in OWL*》。博士论文，伦敦城市大学，2012。 46. K. Wolstencroft、R. Stevens 和 V. Haarslev。《将 OWL 推理应用于基因组数据》。InC.J. O. Baker和 K.-H. Cheung，编者， *Semantic Web: Revolu-tionizing Knowledge Discovery in the Life Sciences*，页码 225–248，美国马萨诸塞州波士顿，2007。Springer US。47.Z.Xing、J.Pei 和 E. Keogh。《序列分类简要综述》。*Explor. Newsl.* 12 1 40 48, 2010 *SIGKDD* ， ( ): –11月。 48.K.Yang 和J.Deng。《使用图神经网络进行强增量式成分句法分析》。 *Advances in Neural Information Processing Systems* 33 21687 21698, ，: – 。 49. M. Young-Lai。《文法推断》。In L.LIU 和 M.T. ¨OZSU，编者， *Encyclopedia of Database Systems* 1256 1260, 2009 ，页码 – 美国马萨诸塞州波士顿， 。Springer US。 2020