4.1 Syntax

简要回顾词性的含义后，介绍了成分（constituent）及其基本性质：

- Constituents are continuous
- Constituents are non-crossing
  - if two constituents share one word, then one of them must completely contain the other.
- Each word is a constituent

有一些简洁的方法来验证句子中某几个单词是否构成成分，例如“coordination” test（可以用并列连词连接的两个成分是同类型的成分）、“pronoun” test（可以用代词来替换某成分）等。

接下来本节都在介绍利用上下文无关语法生成句子的过程（与形式语言课程中相同，因此这里只简要记录），这样的句子可以满足基本的语法。给定语法（即生成式）和句子，我们可以还原其生成过程（语法树）。一个句子可能对应不同的语法树，这时句子是有歧义的。

注意生成式中，|表示并列可以简化表达，括号用于表示可选成分。此外，语法可能形成循环、嵌套等情况。

4.2 Introduction to Parsing

由计算机语言（如 C 语言）引入，计算机语言的最大特点就是没有歧义。与计算机语言（Programming Languages）相比，人类语言（Human Languages）的主要不同有：

- No types for words. 每个单词的类型不能自动获取；
- No brackets around phrases. 短语没有明确的范围；
- Ambiguity（Words, Parses），词汇层面和语法分析层面都可能歧义；
- Implied information，句子中经常暗含背景信息。

分析句子常利用上下文无关语法（Context-free Grammars）。它是一个四元组(N, Σ,R,S)：

- N: non-terminal symbols，非终止符号（常代表句子成分）；
- Σ: terminal symbols (disjoint from N)，终止符号（常是单词）；
- R: rules ($A \rightarrow \beta$), where β is a string from ($\Sigma \cup N$)*，转换规则；
- S: start symbol from N，起始符号。

4.3 Classic parsing methods

语法分析的一种经典方式是将其看作搜索问题。该过程需要考虑两类约束，即：

- From the input sentence
- From the grammar

前者是指搜索过程要满足输入句子，后者指搜索过程要满足给定语法。

常见的搜索过程有自上而下（Top-down）和自下而上（Bottom-up）两类。前者是从语法的起始变元（一般表示为 S）展开搜索，为了生成完整的句子需要大量搜素。后者是从终止字符（即单词）展开搜索，为了得到完整的分析树（即回溯到 S）需要大量搜索。

Shift-reduce Parsing 是一种自下而上的算法，涉及两种操作：

1. Shift 操作，将一个终止字符（即单词）压入栈，此时该单词从语句中移出；
2. Reduce 操作，将栈顶满足语法右侧的一个或几个字符弹出，替换为语法左侧的字符。

该算法成功结束仅当满足两个条件：所有单词从语句中移出、栈中只含有起始变元 S。

CKY (Cocke-Kasami-Younger)算法也是一种自下而上的算法，它需要语法是经过标准化的语法（即二元语法，the only things that are allowed to have is a non-terminal going to two non-terminal, or a non-terminal going to a terminal），也即满足乔姆斯基范式（Chomsky Normal Form）。该算法执行的示意图如图 4.3.1。

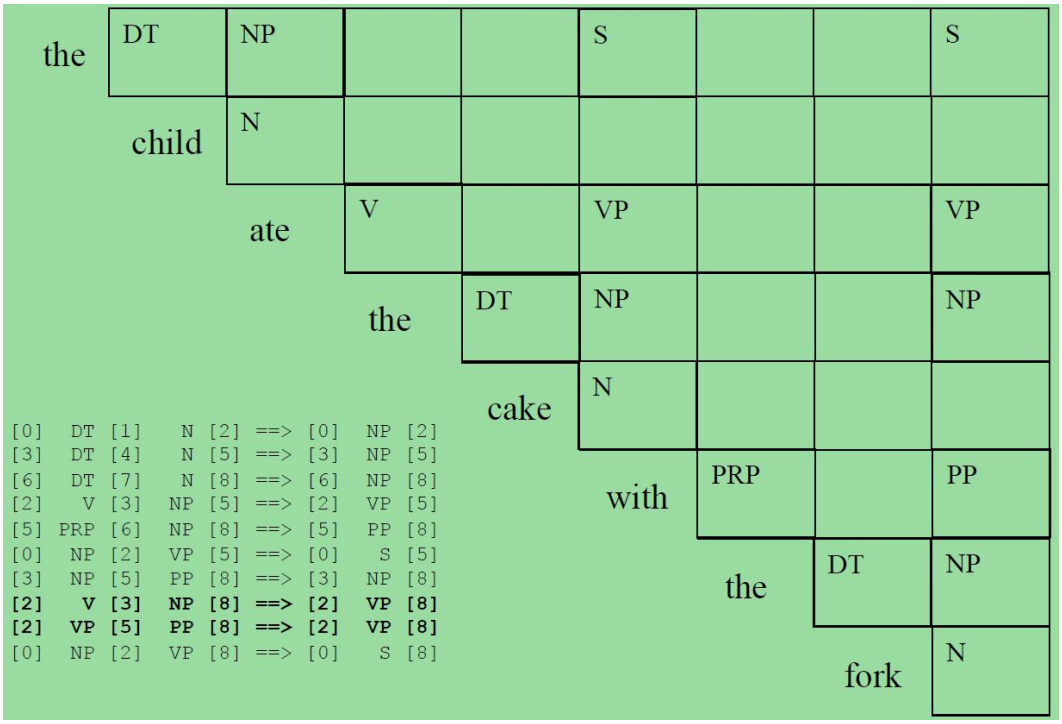

图 4.3.1

CKY 算法中表格单元的数量级为 $O(n^2)$，其中 n 是句子长度。为了找到某个单一的分析树，算法的复杂度是 $O(n^3)$。但若需要找到所有可能的分析树，则复杂度是指数级的，某种意义上这是显然的，因为理论上一个句子的可能分析树本身就是指数级的（the number of parses can be exponential）。

需要指出，为了使用 CKY 算法，往往需要将原始语法转变为二元语法，这会使得最终得到的分析树与原始分析树不同，相应的后处理过程可以转回原始分析树。此外，CKY 算法无法解决语法分析树的歧义问题。

Earley parser 算法是自上而下的算法，没有二元语法的限制，下节介绍。

4.4.1 Earley Parser

对于 Earley Parser 算法并没有给出严格的形式化说明，只是结合一个例子给出了具体的介绍。作为一种动态算法，它按照下述思路进行初始化并迭代：

1. 在所有生成式右侧的最前面标记*，指代该生成式目前匹配到的位置，例如 V -> * 'take'；

2. 从句子中读入一个新的单词，该单词作为终止字符会满足某生成式的右侧，进而可更新该生成式的匹配位置，例如读入’take’后，V -> * 'take'更新为 V -> 'take' *；

3. 此时进行观察，每有某个生成式的右侧得到完全匹配，说明该生成式左侧得以匹配，进而更新与该左侧相关生成式匹配状态，例如步骤 2 中更新为 V -> 'take' *后，V 得以匹配，若之前已有生成式状态 VP -> * V NP，则更新为 VP -> V * NP。这一环节中需要迭代更新所有满足规则的生成式，并记录下这些被更新的生成式便于恢复分析过程；

4. 继续读入新的单词，重复执行步骤 2、3。全部单词读入完成后，若存在以 S 为左侧的生成式匹配完成，则分析成功，根据记录下的更新过程回溯分析过程即可。

4.4.2 Issues with Context-free grammars

上下文无关文法在用于分析语法时存在一些不足：

1. 保持一致性（Agreement）的问题。例如英语中不同的人称、数量、时态、语态等对应不同的形式，要很好地解决该问题可以针对每种具体情况给定专门的生成式规则，但这会引起组合爆炸的问题。

2. Subcategorization Frames。一种例子是不同的动词在不同的情境下具有不同的语法作

用，例如直接作用于宾语、做表语、形成介词短语、不定式等多种情况。这导致在分析过程中需要考虑复杂的可能。

3. 上下文无关的假设存在问题。一种例子是，全体名词短语中各类的比例，与 S 生成或动词短语生成的名词短语各类比例不同，即名词短语类别的分布受上下文的影响。具体数据为 All NPs: 11% NP PP, 9% DT NN, 6% PRP. NPs under S: 9% NP PP, 9% DT NN, 21% PRP. NPs under VP: 23% NP PP, 7% DT NN, 4% PRP.

4.5.1 The Penn Treebank

Penn Treebank 是十分经典的语法分析标注库，包括 40000 条训练语句，2400 条测试语句。句子类型多为 Wall Street Journal news stories 和 some spoken conversations。该数据集将每条语句标记为语法分析书，示例如图 4.5.1。

```
(S
  (SBAR-PRP
    (IN Because)
    (S
      (S
        (NP-SBJ (DT the) (NNP CD))
        (VP
          (VBD had)
          (NP
            (NP (DT an) (JJ effective) (NN yield))
            (PP (IN of) (NP (CD 13.4) (NN %))))
          (SBAR-TMP
            (WHADVP-4 (WRB when))
            (S
              (NP-SBJ-1 (PRP it))
              (VP
                (VBD was)
                (VP
                  (VBN issued)
                  (NP (-NONE- *-1))
                  (PP-TMP (IN in) (NP (CD 1984)))
                  (ADVP-TMP (-NONE- *T*-4)))))))))
                     ...
```

图 4.5.1

该数据集考虑到了一些特殊情况，例如会将省略的句子成分（如主语）用特殊符号填补。

此外，该数据集的一项重要功能是支持按照一定条件查询所需的语句样本，例如可以方便地查询到含有介词短语的训练语句，等等。

该数据集的可用于：

- Statistics about different constituents（成分） and phenomena（现象）
- Training systems
- Evaluating systems
- Multilingual extensions

4.5.2 Parsing evaluation

介绍了语法分析数据集后，自然引入语法分析的评价方法。

常用的评价指标如下：

1. Precision and recall, get the proper constituents. 该指标要求正确地将单词组合为句子成分；

2. Labeled precision and recall, also get the correct non-terminal labels. 在 1 的基础上，还要求所得句子成分的类别是正确的；

3. F1, harmonic mean of precision and recall；

4. Crossing brackets, 针对的是(A (B C)) 与 ((A B) C) 之间的差异；

5. Complete match，完全正确识别的句子的比例；

6. Tagging accuracy，正确标记的单词比例。

另外，计算出算法的性能指标 P(A)后，如何判断该指标是否足够高呢？首先，应当给出参考的指标基准（baselines），该基准可以是利用十分简单的规则（如一律标记为名词）得到的标记结果对应的评价指标，记为 P(E)。定义

κ =(P(A)-P(E))/(1-P(E))

κ 越高，算法效果越好。一般大于 0.7 可认为高，低于 0.4 或 0.3 则认为较低。