

WOODS, W. A. ,1970, Transition network grammars for natural language analysis. In Computational linguistics, vol.13, No.10.

ATN分析算法

詹卫东

<http://ccl.pku.edu.cn/doubtfire>

ATN分析方法

张三是县长派来的
苍蝇是瞎子打死的
主意是董永想出来的

.....

N V N V V 的

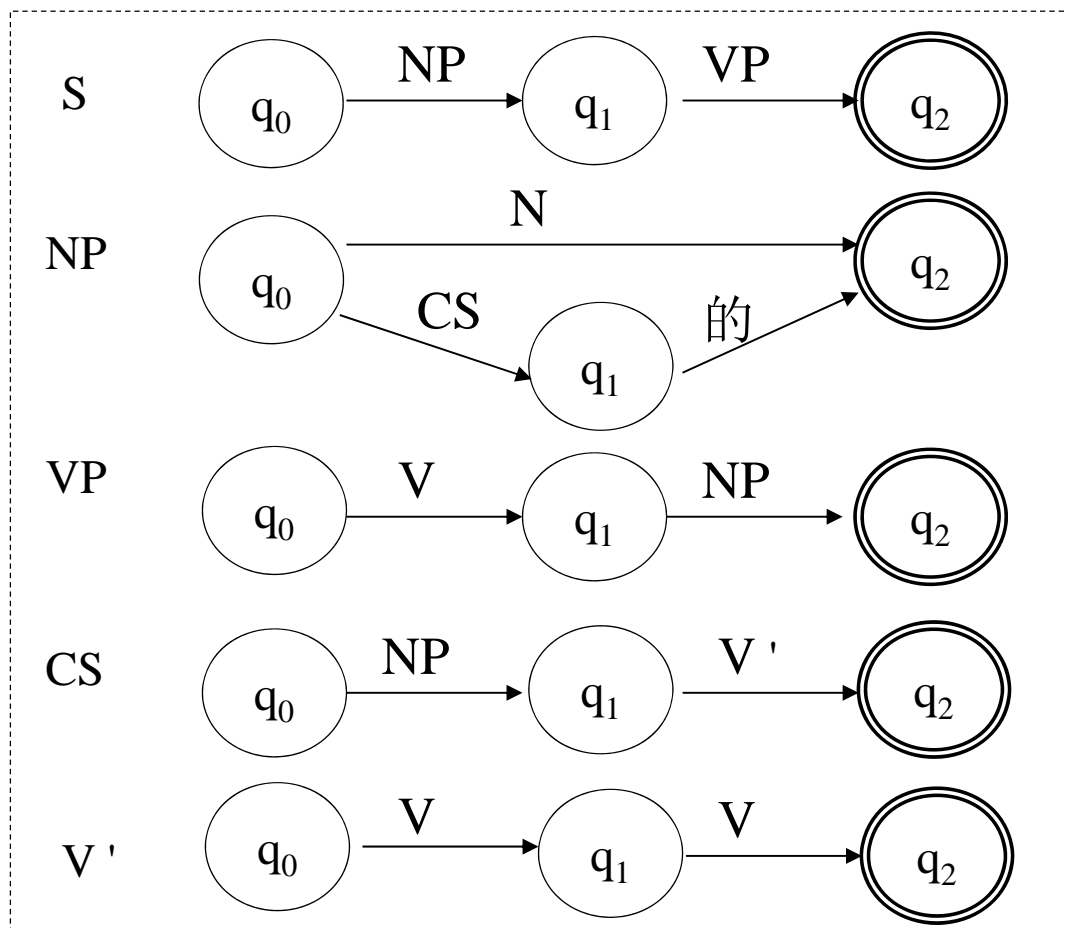
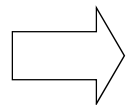
- (1) $S \rightarrow NP VP$
- (2) $NP \rightarrow N$
- (3) $NP \rightarrow CS$ 的
- (4) $CS \rightarrow NP V'$
- (5) $VP \rightarrow V NP$
- (6) $V' \rightarrow V V$

ATN : Augmented Transition Network

RTN : Recursive Transition Network

从CFG到递归转移网络 (RTN)

- (1) $S \rightarrow NP VP$
- (2) $NP \rightarrow N$
- (3) $NP \rightarrow CS \text{ 的}$
- (4) $CS \rightarrow NP V'$
- (5) $VP \rightarrow V NP$
- (6) $V' \rightarrow V V$



RTN算法描述 —— 基本概念

子网名称	S, NP, VP, CS, V' (S是个特殊的子网)
状态节点	q_0, q_1, q_2 (其中 q_0 是开始状态, q_2 是结束状态)
出边	从当前状态向下一个状态转移的弧 (NP, VP, V, ...)
待分析字符串	$w_1w_2w_3\dots$ (主意是董永想出来的, 张三是县长派来的)
递归栈	记录来自哪个子网, 以及回到上层子网时应处的状态
当前状态	\langle 当前子网, 当前状态节点 \rangle \langle 当前出边, 后续状态节点 \rangle

对过去加以保留

对将来进行预测

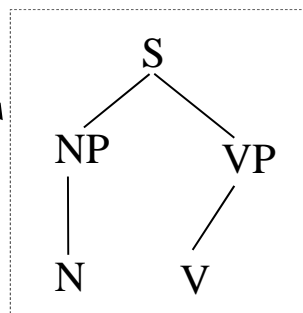
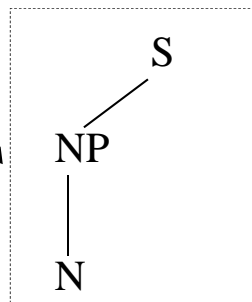
RTN算法描述 (top-down)

1. 初始化: 当前状态为 $\langle S, q_0 \rangle$, 字符串指针指向待分析字符串第一个字符, 递归栈清空。
2. 2.1 如果当前状态节点不是终止状态:
把当前状态节点出边指针指向第一个出边,
 - 2.1.1 **如果当前出边的标记为终结符**,
比较该标记与当前字符串移动指针所指字符
 - 2.1.1.1 如果相等, 预测得到验证, 构造子树, 将当前状态节点设为当前出边的后续状态节点, 同时修改当前出边;
 - 2.1.1.2 如果不相等, 出边指针指向下一个出边。如果不存在下一个出边并且存在回溯点, 则回溯, 否则分析失败。
 - 2.1.2 **如果当前出边的标记为非终结符**, 把当前子网名称及当前节点出边的后续状态压栈, 并把当前状态节点设为当前子网的开始状态, 同时修改当前出边及后续状态。
 - 2.1.3 **如果当前出边有多重选择**, 需要设置恢复断点, 保存递归栈、待分析字符串、当前ATN状态以及出边列表状态的全部内容, 以便回溯时使用。
- 2.2. 如果当前状态节点是终止状态但不是子网S的终止状态:
将递归栈退栈, 并将当前状态设置为当前栈顶的状态, 转入步骤2继续执行。
- 2.3. 如果当前状态节点是终止状态并且是子网S的终止状态:
 - 2.3.1 若递归栈已空且待分析字符串已空, 则分析成功, 结束;
 - 2.3.2 否则, 若存在回溯点, 回溯。
 - 2.3.3 若不存在回溯点, 分析失败。

RTN算法过程示例-1

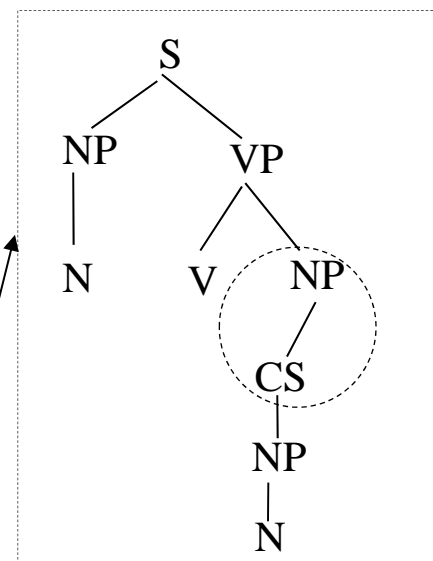
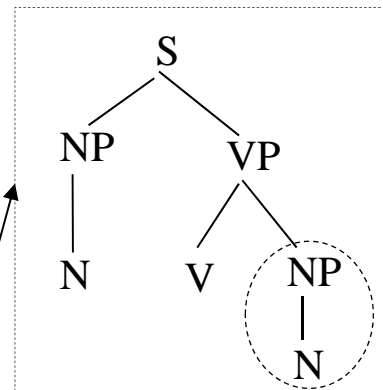
记录回溯点,即当前选择的出边序号,回溯时+1

递归栈内容	待分析字符串	<子网,当前状态>	<出边,后续状态>	回溯
空	N V N V V 的	<S, q ₀ >	<NP, q ₁ >	
→ <S, q ₁ >	N V N V V 的	<NP, q ₀ >	<N, q ₂ >, <CS, q ₁ >	1
<S, q ₁ ><NP, q ₂ >	N V N V V 的		N 预测验证成功	
<S, q ₁ ><NP, q ₂ >	V N V V 的	<NP, q ₂ >	NP 预测验证成功	
<S, q ₁ >	V N V V 的	<S, q ₁ >	<VP, q ₂ >	
<S, q ₂ >	V N V V 的	<VP, q ₀ >	<V, q ₁ >	
<S, q ₂ ><VP, q ₁ >	V N V V 的		V 预测验证成功	



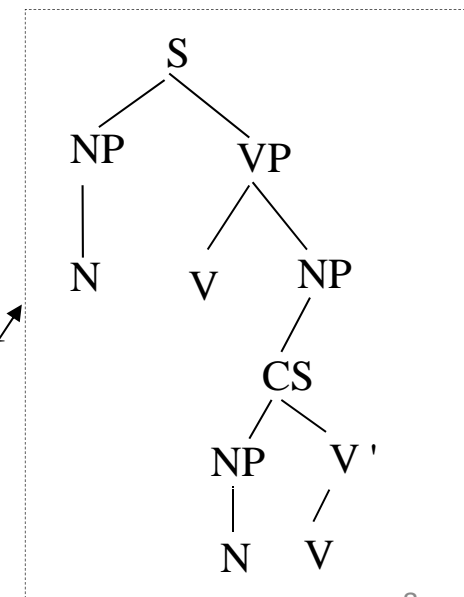
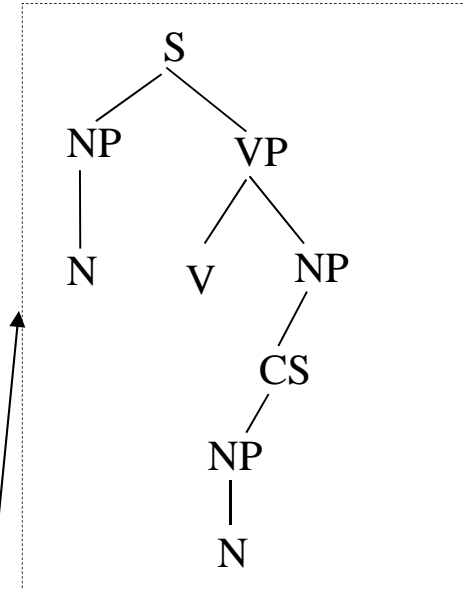
RTN算法过程示例-2

递归栈内容	待分析字符串	<子网,当前状态>	<出边,后续状态>	回溯
<S, q ₂ ><VP, q ₂ >	N V V 的	<NP, q ₀ >	<N, q ₂ >, <CS, q ₁ >	1
<S, q ₂ ><VP, q ₂ > <NP, q ₂ >	N V V 的		N 预测验证成功	
<S, q ₂ ><VP, q ₂ > <NP, q ₂ >	V V 的	<NP, q ₂ >	NP 预测验证成功	
<S, q ₂ ><VP, q ₂ >	V V 的	<VP, q ₂ >	无出边	
<S, q ₂ ><VP, q ₂ >	V V 的	<VP, q ₂ >	回溯 (选择另一条出边)	
<S, q ₂ ><VP, q ₂ >	N V V 的	<NP, q ₀ >	<CS, q ₁ >	2
<S, q ₂ ><VP, q ₂ > <NP, q ₁ >	N V V 的	<CS, q ₀ >	<NP, q ₁ >	
<S, q ₂ ><VP, q ₂ > <NP, q ₁ ><CS, q ₁ >	N V V 的	<NP, q ₀ >	<N, q ₂ >, <CS, q ₁ >	1



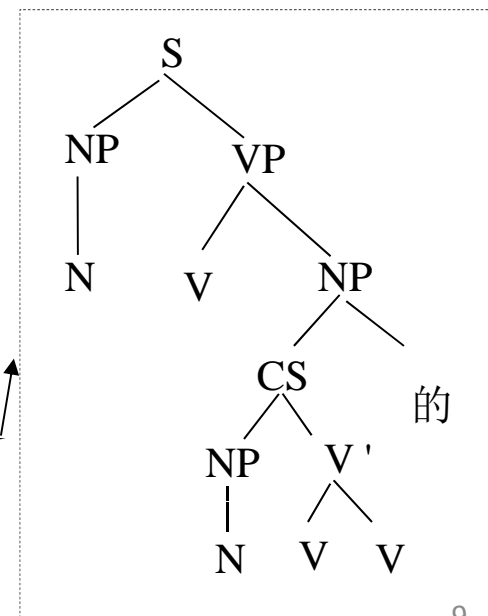
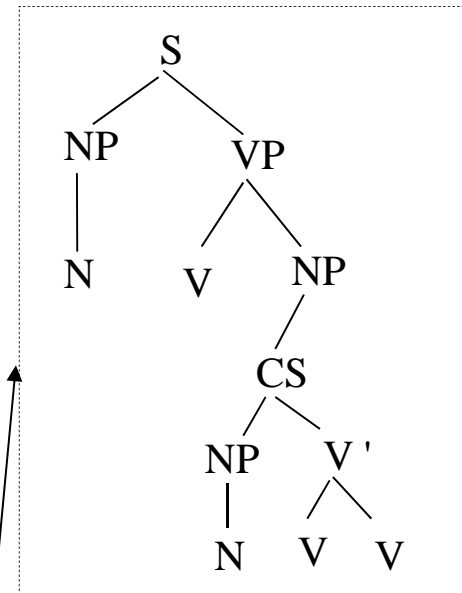
RTN算法过程示例-3

递归栈内容	待分析字符串	<子网,当前状态>	<出边,后续状态>	回溯
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_2 \rangle \langle CS, q_1 \rangle$ $\langle NP, q_1 \rangle$	N V V 的		N 预测验证成功	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_2 \rangle \langle CS, q_1 \rangle$ $\langle NP, q_1 \rangle$	V V 的	$\langle NP, q_0 \rangle$	NP 预测验证成功	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_1 \rangle \langle CS, q_1 \rangle$	V V 的	$\langle CS, q_1 \rangle$	$\langle V', q_2 \rangle$	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_1 \rangle \langle CS, q_2 \rangle$	V V 的	$\langle V', q_0 \rangle$	$\langle V, q_1 \rangle$	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_1 \rangle \langle CS, q_2 \rangle$ $\langle V', q_1 \rangle$	V V 的		V 预测验证成功	



RTN算法过程示例-4

递归栈内容	待分析字符串	<子网,当前状态>	<出边,后续状态>	回溯
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_1 \rangle \langle CS, q_2 \rangle$ $\langle V', q_2 \rangle$	V 的	$\langle V', q_1 \rangle$	$\langle V, q_2 \rangle$	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_1 \rangle \langle CS, q_2 \rangle$ $\langle V', q_2 \rangle$	V 的		V 预测验证成功	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_1 \rangle \langle CS, q_2 \rangle$	的	$\langle CS, q_2 \rangle$	V' 预测验证成功	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_1 \rangle$	的	$\langle NP, q_1 \rangle$	$\langle \text{的}, q_2 \rangle$ CS 预测验证成功	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$ $\langle NP, q_1 \rangle$	的		"的" 预测验证成功	
$\langle S, q_2 \rangle \langle VP, q_2 \rangle$	\$	$\langle NP, q_2 \rangle$	NP 预测验证成功	
$\langle S, q_2 \rangle$	\$	$\langle VP, q_2 \rangle$	VP 预测验证成功	
空	\$	$\langle S, q_2 \rangle$	S 预测验证成功 分析成功结束	



从RTN到ATN

- 寄存器（register）
- 条件测试（condition）
- 动作（action）

一件衣服 —— * 一件小孩

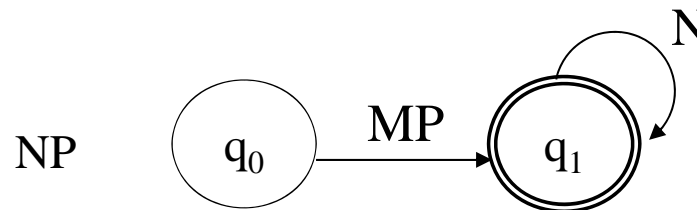
张三打李四 —— 李四被张三打了

一件小孩的衣服

施事：张三

受事：李四

ATN示例



NP: $q_0 - MP - q_1$

动作:

(1) 进入MP网络

(2) 将“量词”寄存器的值置为“MP*”

NP: $q_1 - N - q_1$

条件:

(1) 当前词是名词

(2) 名词寄存器表中的“量词”值为“MP*”

一件衣服

“一件” → “量词”寄存器

“MP*”=“件”

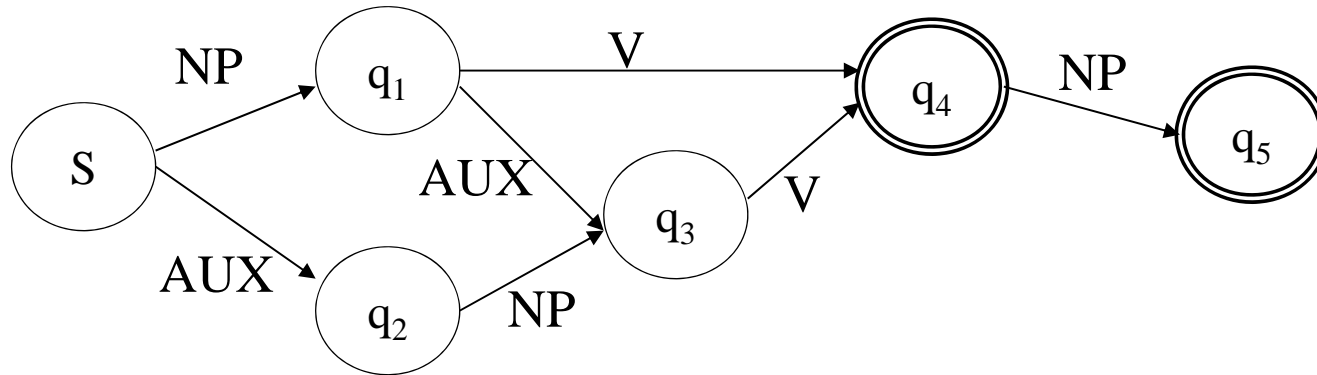
“衣服”的“量词”值为“件”

可以匹配，NP分析成功

一件小孩

分析失败

ATN示例（续）



You can help them NP-AUX-V-NP

Can you help them AUX-NP-V-NP

引自刘开瑛、郭炳炎，1991，P29

ATN的Lisp表示法

```
1 ((S/ (PUSH NP / T
2   (SETR SUBJ *)
3   (SETR TYPE "DCL")
4   (TO q1)
5   (CAT AUX T
6     (SETR AUX *)
7     (SETR TYPE "QUESTION")
8     (TO q2))
9   (q1 (CAT V T
10      (SETR AUX NIL)
11   .....
```

