

ОСНОВЫ ОБРАБОТКИ ТЕКСТОВ

Лекция #6:
Синтаксический анализ

Лектор: м.н.с. ИСП РАН Майоров Владимир Дмитриевич

Синтаксис

- Предложение – это единица языка, которая представляет собой грамматически организованное соединение слов, обладающее смысловой законченностью.
- Грамматика – раздел лингвистики, который изучает закономерности построения правильных осмысленных речевых отрезков (словоформ, словосочетаний, предложений, текстов).
- Синтаксис — раздел лингвистики, изучающий и моделирующий правила, по которым образуются единицы, более крупные, чем слово, а именно словосочетания и предложения.

Синтаксические правила

- Существительное сочетается с прилагательным

быстрый бег → быстрый бег

быстро бег → не словосочетание

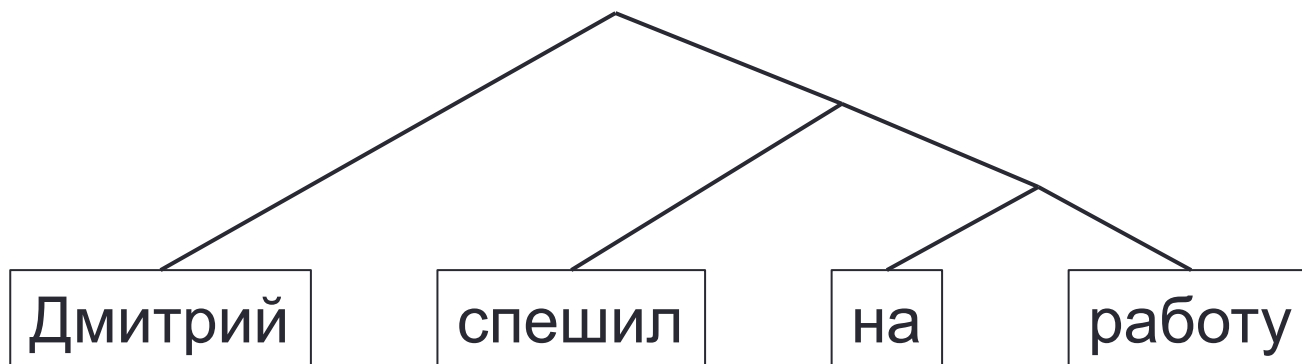
- Глагол сочетается с наречием

бегать быстрый → не словосочетание

бегать быстро → бегать быстро

Синтаксическая структура

- Представление предложения в виде вложенных составляющих.
- Составляющая (фраза, синтаксическая группа) – структурная единица предложения, составленная из более тесно связанных друг с другом составляющих меньшего размера.



Синтаксические правила

- Существительное управляет прилагательным

красная книга → красная книга

красный книга → не словосочетание

- Предлог управляет существительным

в книга → не словосочетание

в книге → в книге

Синтаксические правила

- В зависимости от главного слова в словосочетании выделяют
 - Именные группы (главное – существительное)
 - Группа прилагательного
 - Наречная группа
 - Предложная группа
 - Глагольная группа
 - Предложение



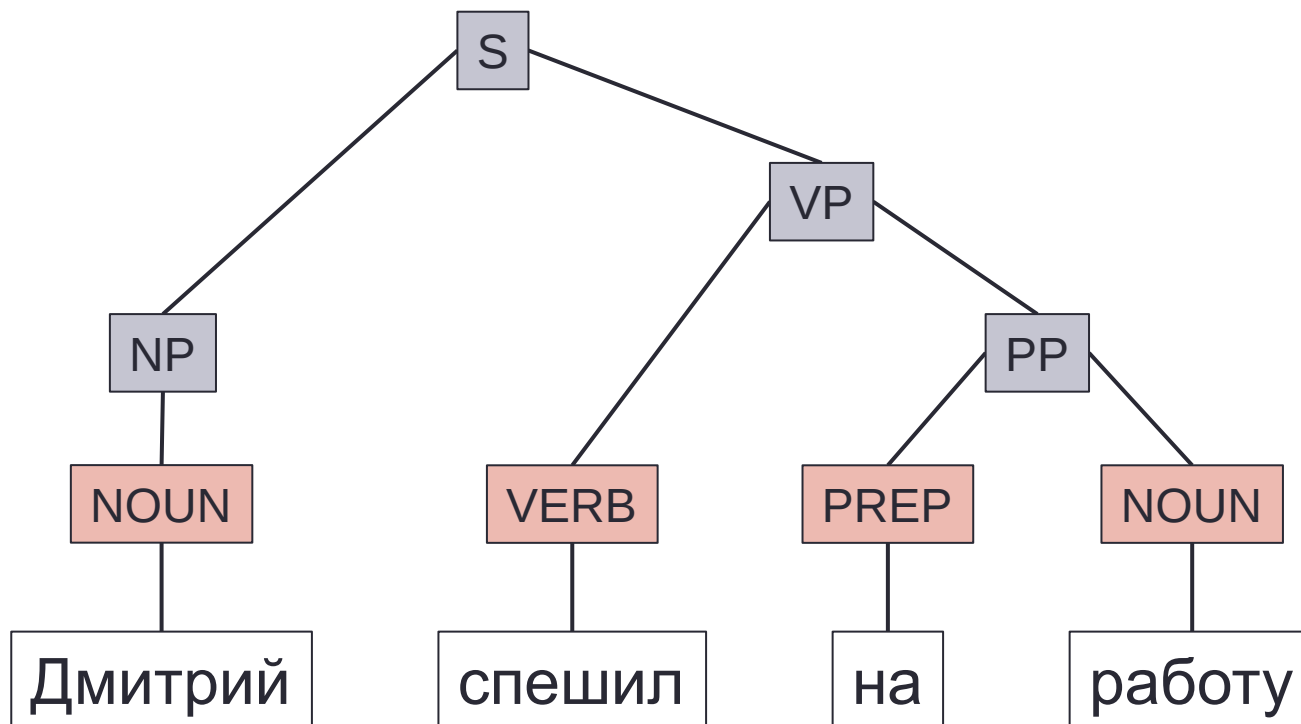
Синтаксические правила

- В зависимости от главного слова в словосочетании выделяют
 - Именные группы (главное – существительное)
 - Группа прилагательного
 - Наречная группа
 - Предложная группа
 - Глагольная группа
 - Предложение



Синтаксическая структура

- Каждой составляющей назначается тип в зависимости от главного слова



Формальная грамматика

- Способ описания формального языка.
- Формальная грамматика состоит из:
 - Σ — множество терминальных символов
 - N — множество нетерминальных символов
 - P — набор правил вывода $\alpha \rightarrow \beta$, где
 - α — последовательность символов из $\Sigma \cup N$, хотя бы один из N
 - β — последовательность символов из $\Sigma \cup N$
 - S — начальный символ (из N)

Формальная грамматика

- P — набор правил вывода $\alpha \rightarrow \beta$, где
 - α — последовательность символов из $\Sigma \cup N$, хотя бы один из N
 - β — последовательность символов из $\Sigma \cup N$
- Иерархия Хомского
 - тип 0. неограниченные грамматики
любые правила
 - тип 1. контекстно-зависимые грамматики
правила вида $c_1 A c_2 \rightarrow c_1 \beta c_2$
 - тип 2. контекстно-свободные грамматики
правила вида $A \rightarrow \beta$
 - тип 3. регулярные грамматики
правила вида $A \rightarrow a$, $A \rightarrow aB$ или $A \rightarrow \varepsilon$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

S

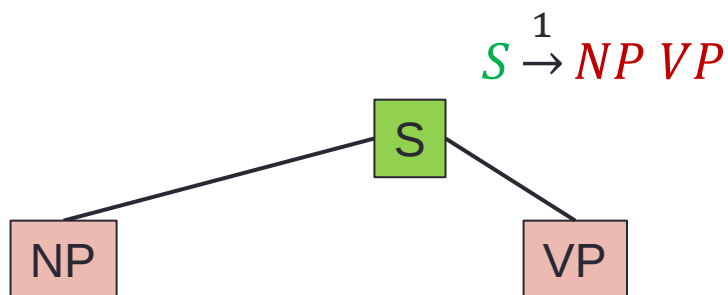
S

1. *S* → *NP VP*
2. *VP* → *Verb NP*
3. *VP* → *Verb NP PP*
4. *NP* → *Pro*
5. *NP* → *Det NP*
6. *NP* → *Noun PP*
7. *NP* → *Noun*
8. *PP* → *Prep Noun*
9. *Verb* → *booked*
10. *Noun* → *flight*
11. *Noun* → *Moscow*
12. *Pro* → *I*
13. *Det* → *a*
14. *Prep* → *from*

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$



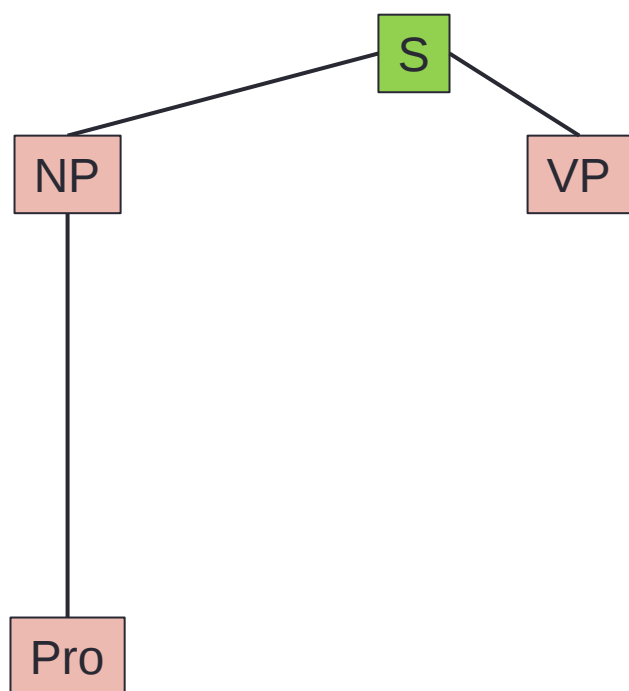
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

$NP VP \xrightarrow{4} Pro VP$

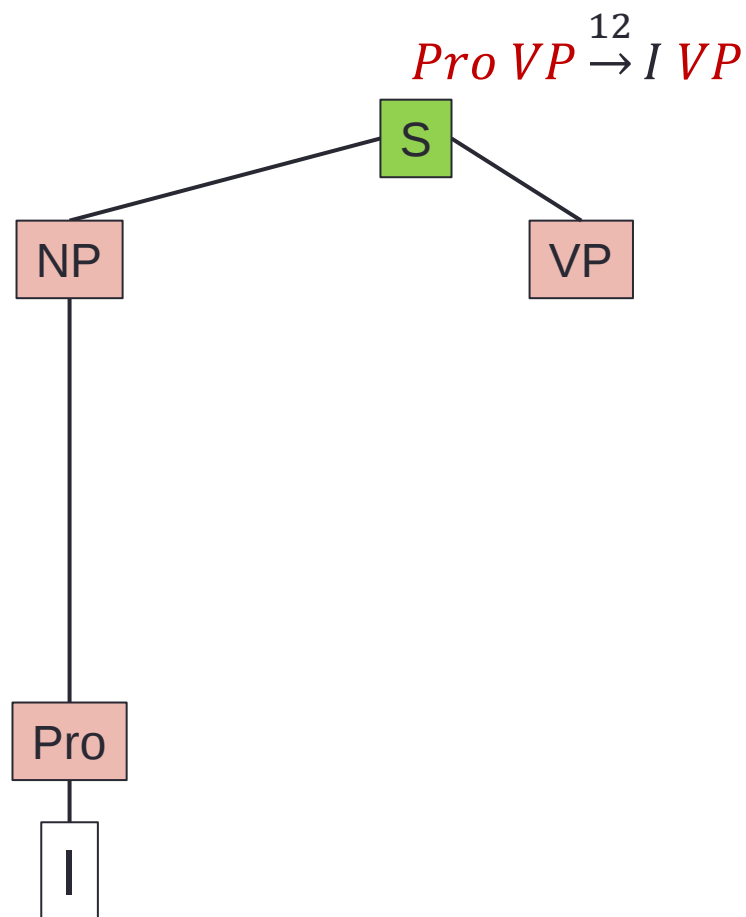


1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$



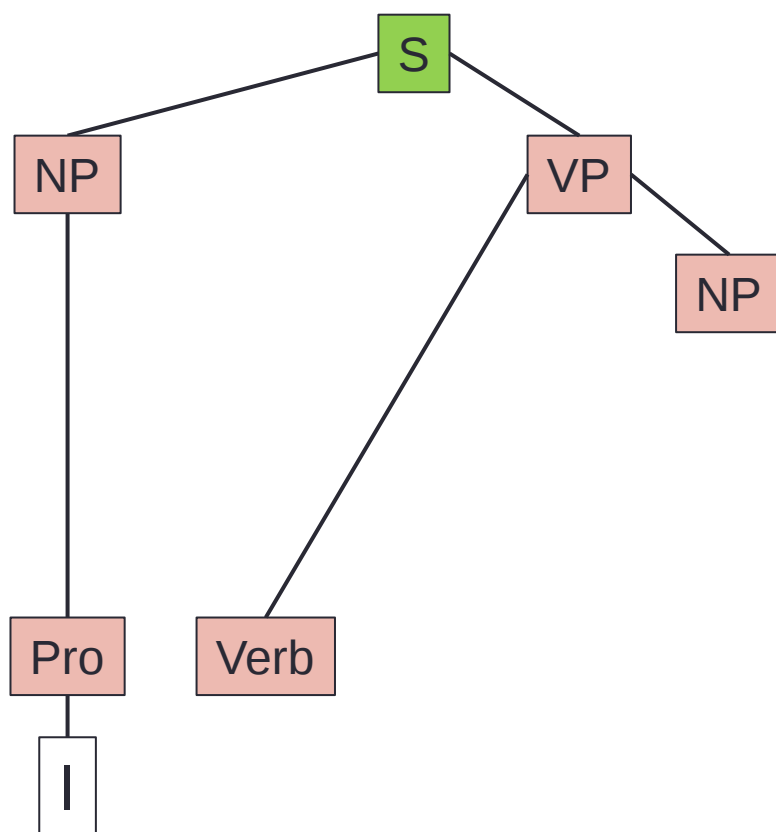
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

$I VP \xrightarrow{2} I Verb NP$



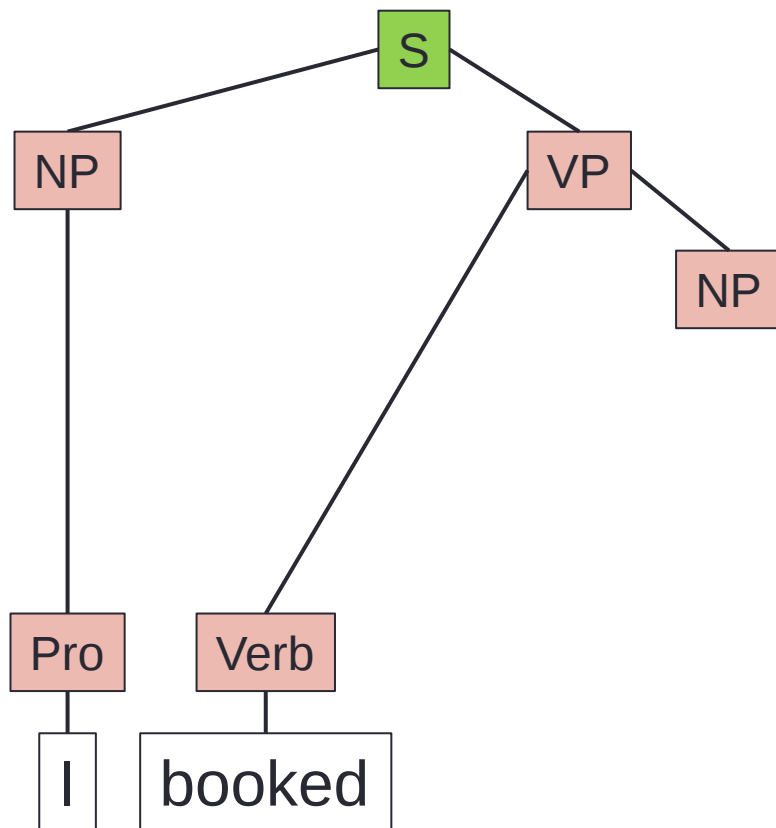
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

$I \text{ Verb } NP \xrightarrow{9} I \text{ booked } NP$



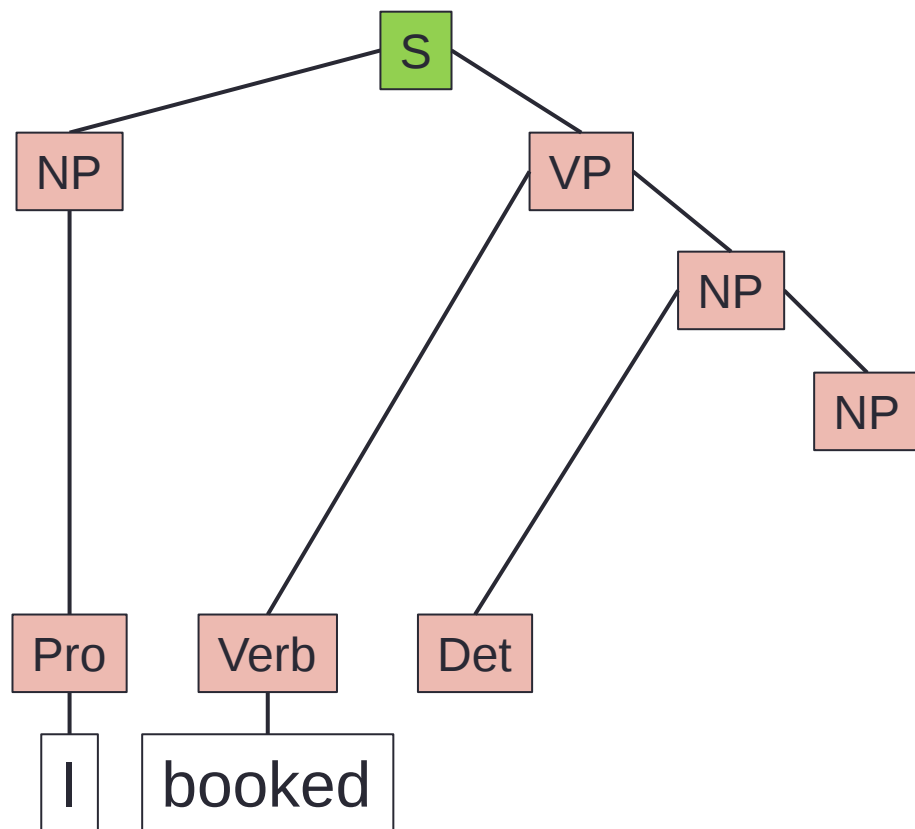
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $\underline{Verb} \rightarrow \underline{booked}$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

$I \text{ booked } NP \xrightarrow{5} I \text{ booked } Det \text{ NP}$



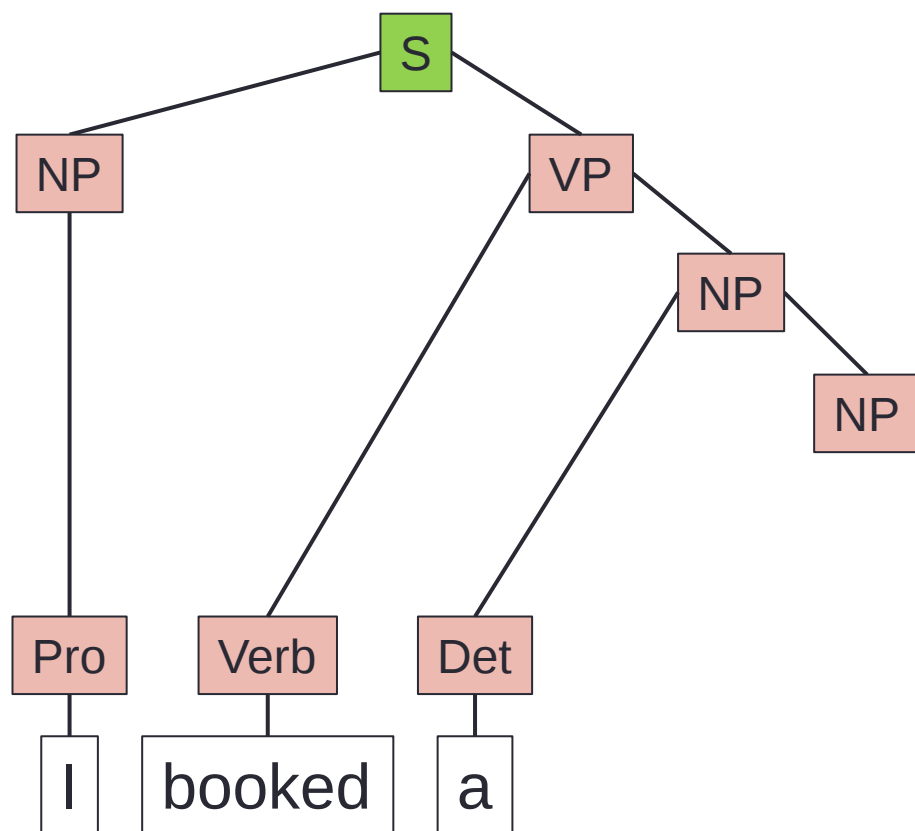
1. $S \rightarrow NP \text{ VP}$
2. $VP \rightarrow Verb \text{ NP}$
3. $VP \rightarrow Verb \text{ NP PP}$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det \text{ NP}$
6. $NP \rightarrow Noun \text{ PP}$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep \text{ Noun}$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

$I \text{ booked } Det \text{ } NP \xrightarrow{13} I \text{ booked } a \text{ } NP$



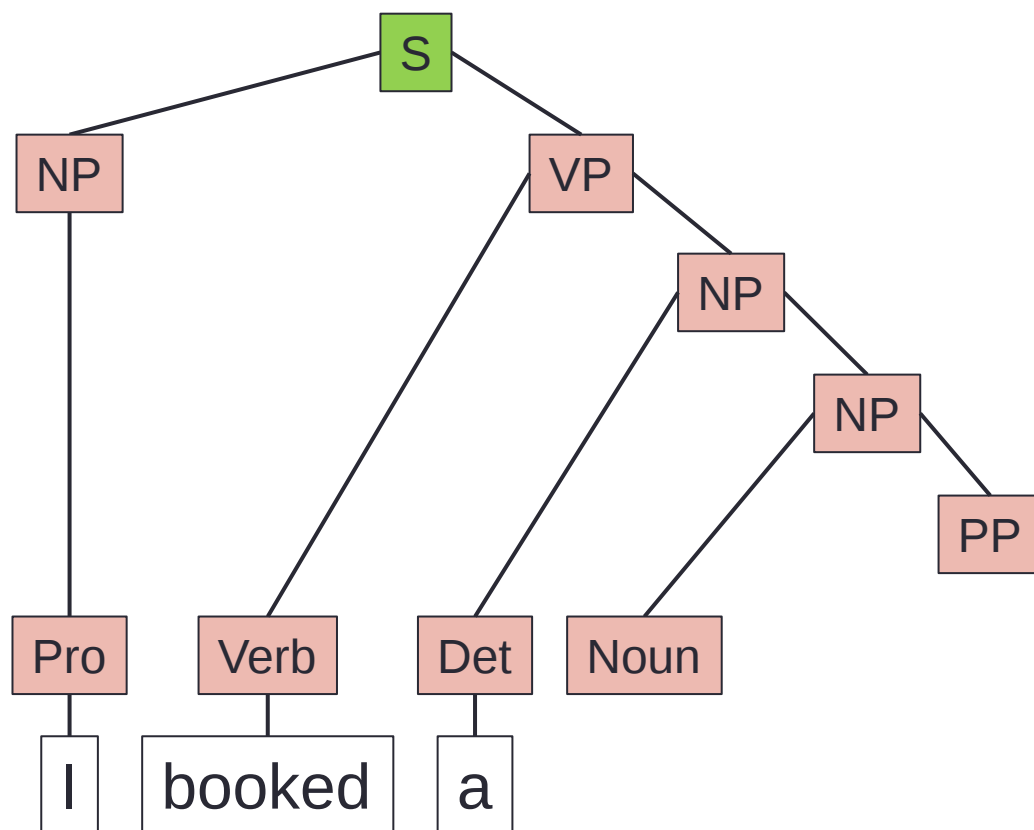
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

$I\ booked\ a\ NP \xrightarrow{6} I\ booked\ a\ Noun\ PP$



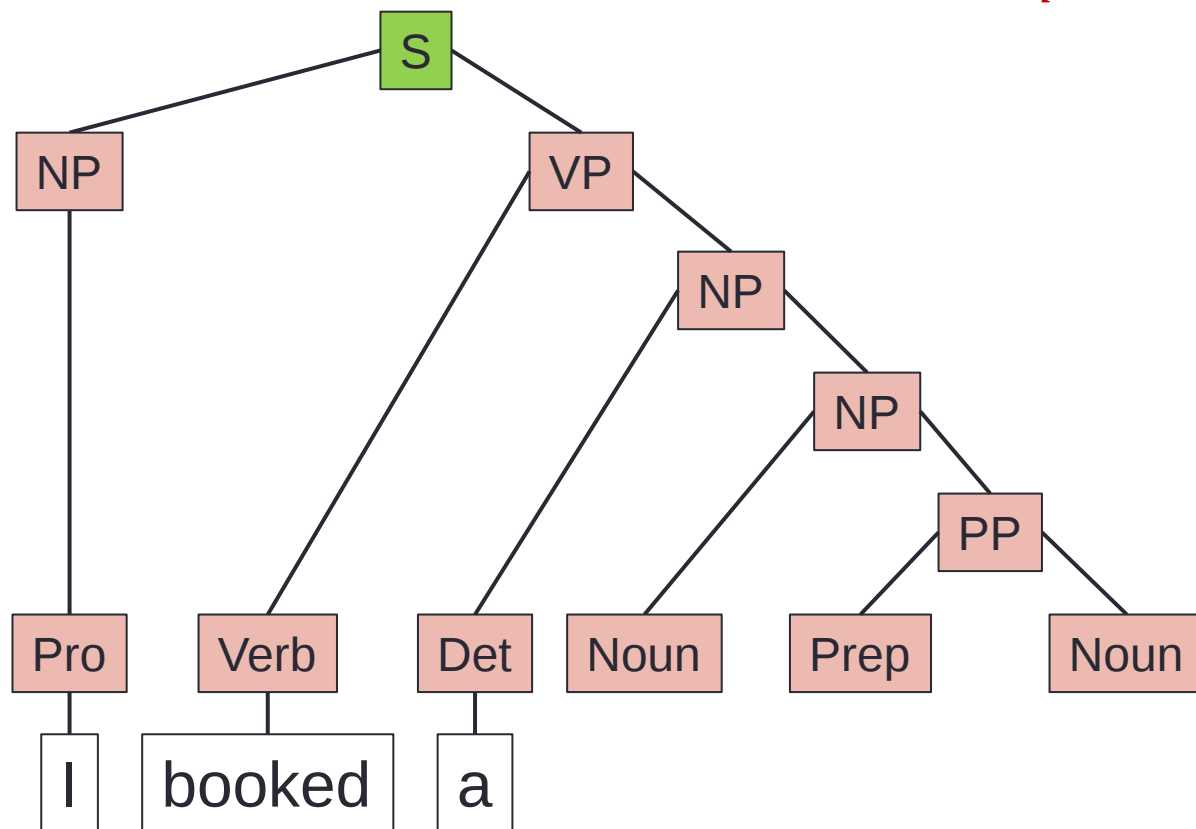
1. $S \rightarrow NP\ VP$
2. $VP \rightarrow Verb\ NP$
3. $VP \rightarrow Verb\ NP\ PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det\ NP$
6. $NP \rightarrow Noun\ PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep\ Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

$I\ booked\ a\ Noun\ PP \xrightarrow{8} I\ booked\ a\ Noun\ Prep\ Noun$



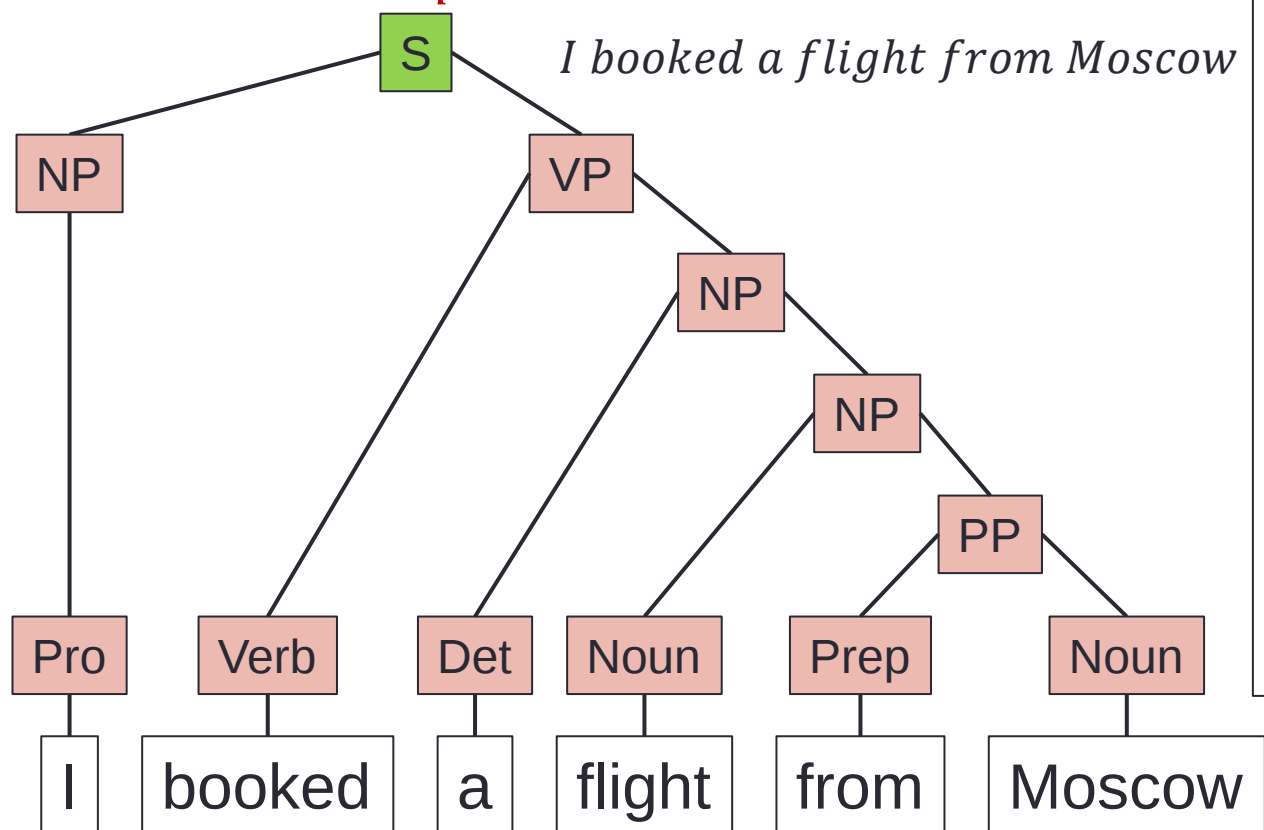
1. $S \rightarrow NP\ VP$
2. $VP \rightarrow Verb\ NP$
3. $VP \rightarrow Verb\ NP\ PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det\ NP$
6. $NP \rightarrow Noun\ PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep\ Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Prep \rightarrow from$

Формальная грамматика

$\Sigma = \{booked, flight, Moscow, I, a, from\}$

$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

I booked a Noun Prep Noun $\xrightarrow{10} \xrightarrow{14} \xrightarrow{11}$

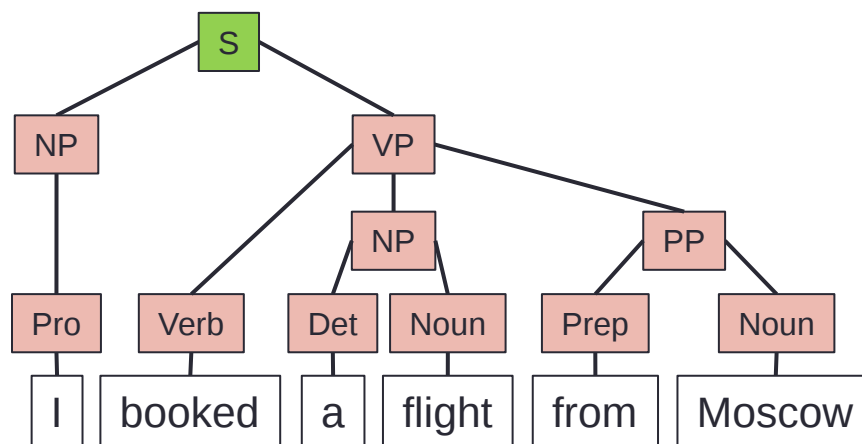
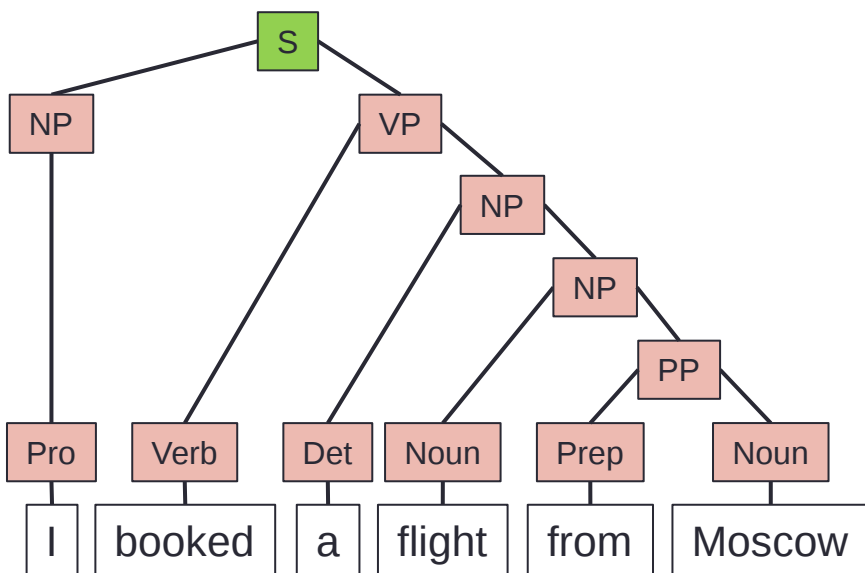


1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $\underline{Noun} \rightarrow \underline{flight}$
11. $\underline{Noun} \rightarrow \underline{Moscow}$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $\underline{Prep} \rightarrow \underline{from}$

Разбор предложения

- На входе: последовательность символов из Σ (слова)
- На выходе:
 - Синтаксическое дерево
 - Последовательность правил для генерации входной строки
- Методы построения дерева:
 - Сверху-вниз
Из начального символа S , вывести входную строку
 - Снизу-вверх
Из входной строки вывести начальный символ S

Неоднозначность языка



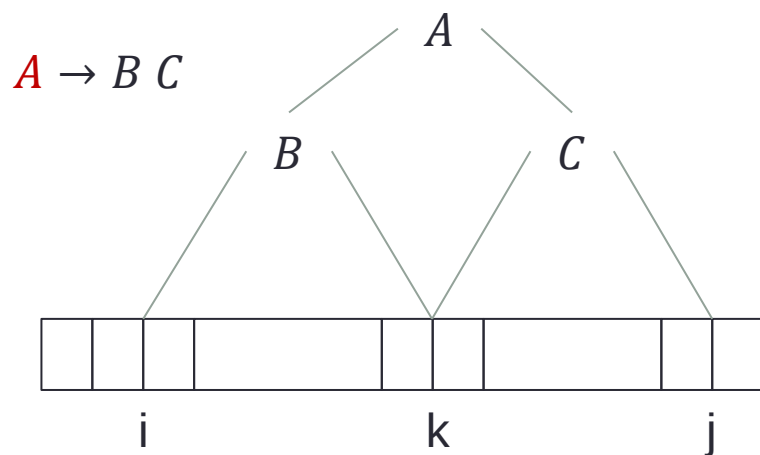
- | | |
|--------------------------------|-------------------------------|
| 1. $S \rightarrow NP VP$ | 9. $Verb \rightarrow booked$ |
| 2. $VP \rightarrow Verb NP$ | 10. $Noun \rightarrow flight$ |
| 3. $VP \rightarrow Verb NP PP$ | 11. $Noun \rightarrow Moscow$ |
| 4. $NP \rightarrow Pro$ | 12. $Pro \rightarrow I$ |
| 5. $NP \rightarrow Det NP$ | 13. $Det \rightarrow a$ |
| 6. $NP \rightarrow Noun PP$ | 14. $Prep \rightarrow from$ |
| 7. $NP \rightarrow Noun$ | |
| 8. $PP \rightarrow Prep Noun$ | |

Алгоритм СҮК

- Для работы необходима КС грамматика в нормальной форме Хомского (CNF)
 - Все правила должны иметь вид: $A \rightarrow BC$ или $A \rightarrow \beta$
 - Любая КС грамматика может быть преобразована в CNF (кроме правила $S \rightarrow \varepsilon$)
- Алгоритм позволяет определить выводимо ли предложение в заданной КС грамматике
- При небольшой модификации алгоритм строит все возможные разборы предложения

Алгоритм СҮК

- Идея алгоритма заключается в рассмотрении всех подстрок исходной строки
 - Для каждой подстроки определить, можно ли вывести ее в заданной грамматике, зная какие подстроки меньшего размера можно вывести в этой грамматике



Алгоритм СҮК

- Дана грамматика в CNF, N нетерминальных символов
- Дана строка w длины n
- Зададим трехмерный массив $d \in B^{N \times n \times n}$,
 $d[A][i][j] = 1$, если из A можно вывести строку $w[i..j]$
- Решаем задачу динамическим программированием
 - Будем рассматривать все подстроки длины m для $m = \overline{1..n}$
 - Для каждой подстроки $[i..j]$ длины m рассмотрим все разбиения $[i..k][k+1..j]$
 - Для каждого правила $A \rightarrow B C$, если
 $d[B, i, k] = 1$ и $d[C, k + 1, j] = 1$, то $d[A, i, j] = 1$

Алгоритм СҮК

6						
5						
4						
3						
2						
1	<i>NP</i>	<i>Verb</i>	<i>Det</i>	<i>Noun NP</i>	<i>Prep</i>	<i>Noun NP</i>
	I	booked	a	flight	from	Moscow

1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК

6						
5						
4						
3						
2						
1						
	I	booked	a	flight	from	Moscow

NP

Verb

Det

*Noun
NP*

Prep

*Noun
NP*

1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК

6						
5						
4						
3						
2						
1	NP	Verb	Det	Noun NP	Prep	Noun NP
	I	booked	a	flight	from	Moscow

1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК

6						
5						
4						
3						
2			<i>NP</i>			
1	<i>NP</i>	<i>Verb</i>	<i>Det</i>	<i>Noun NP</i>	<i>Prep</i>	<i>Noun NP</i>
	I	booked	a	flight	from	Moscow

1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК

6						
5						
4						
3						
2			<i>NP</i>			
1	<i>NP</i>	<i>Verb</i>	<i>Det</i>	<i>Noun NP</i>	<i>Prep</i>	<i>Noun NP</i>
	I	booked	a	flight	from	Moscow

1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК

6						
5						
4						
3						
2			<i>NP</i>		<i>PP</i>	
1	<i>NP</i>	<i>Verb</i>	<i>Det</i>	<i>Noun NP</i>	<i>Prep</i>	<i>Noun NP</i>
	I	booked	a	flight	from	Moscow

1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СУК

6						
5						
4						
3						
2						
1						
	I	booked	a	flight	from	Moscow
	NP	Verb	Det	Noun NP	Prep	Noun NP
	NP	Verb	NP	PP		
	NP	Verb	NP			
	NP	Verb				
	NP	Verb				

1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СУК

6						
5						
4						
3						
2						
1						
	I	booked	a	flight	from	Moscow

Phrase structure labels (in red italics):

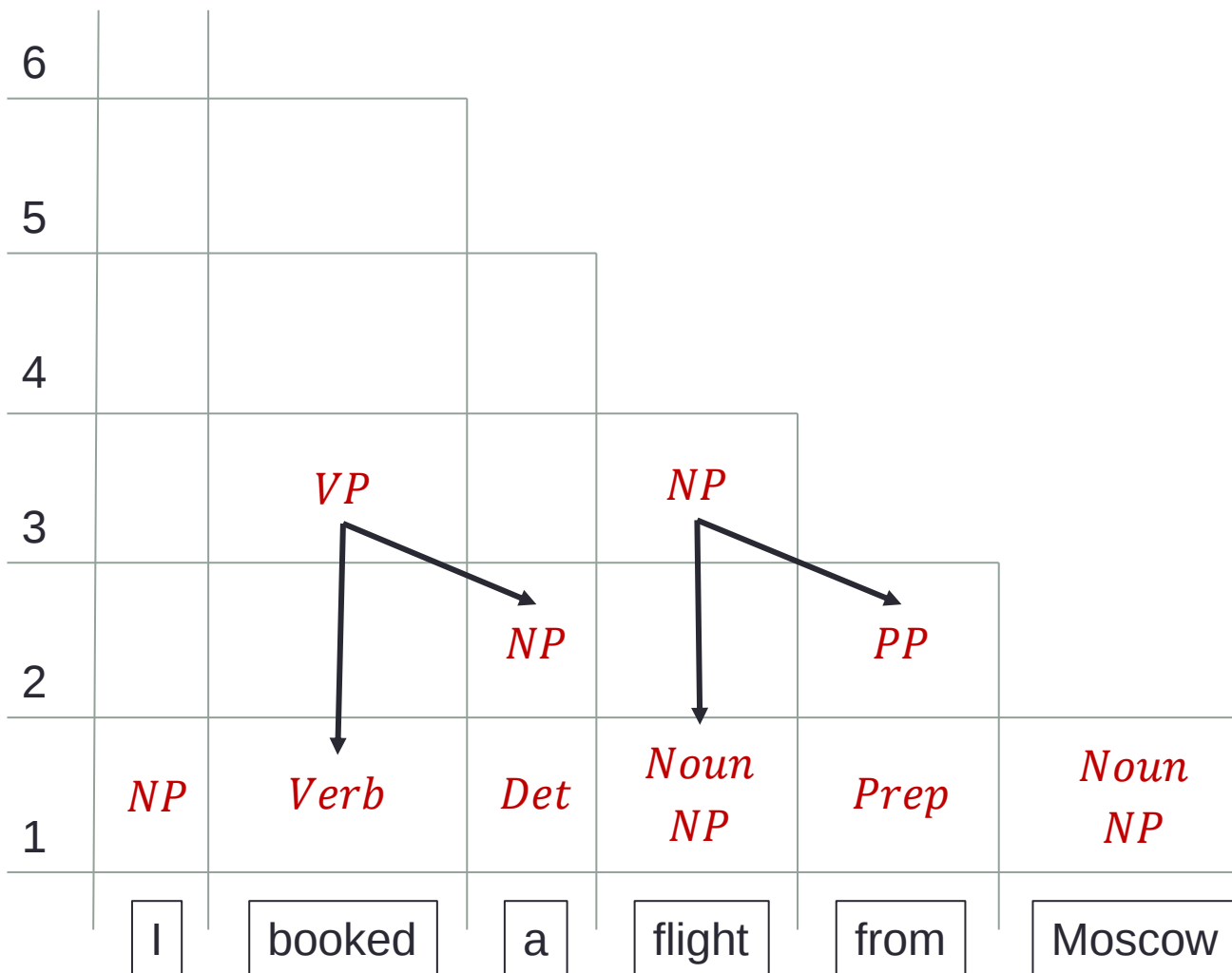
- Row 1: *NP* (above I), *Verb* (above booked), *Det* (above a), *Noun NP* (above flight), *Prep* (above from), *Noun NP* (above Moscow)

Diagram features:

- A red square highlights the cell at row 3, column 1.
- Two arrows originate from the red square:
 - One arrow points down to the *NP* label above 'I'.
 - Another arrow points down and to the right to the *Det* label above 'a'.

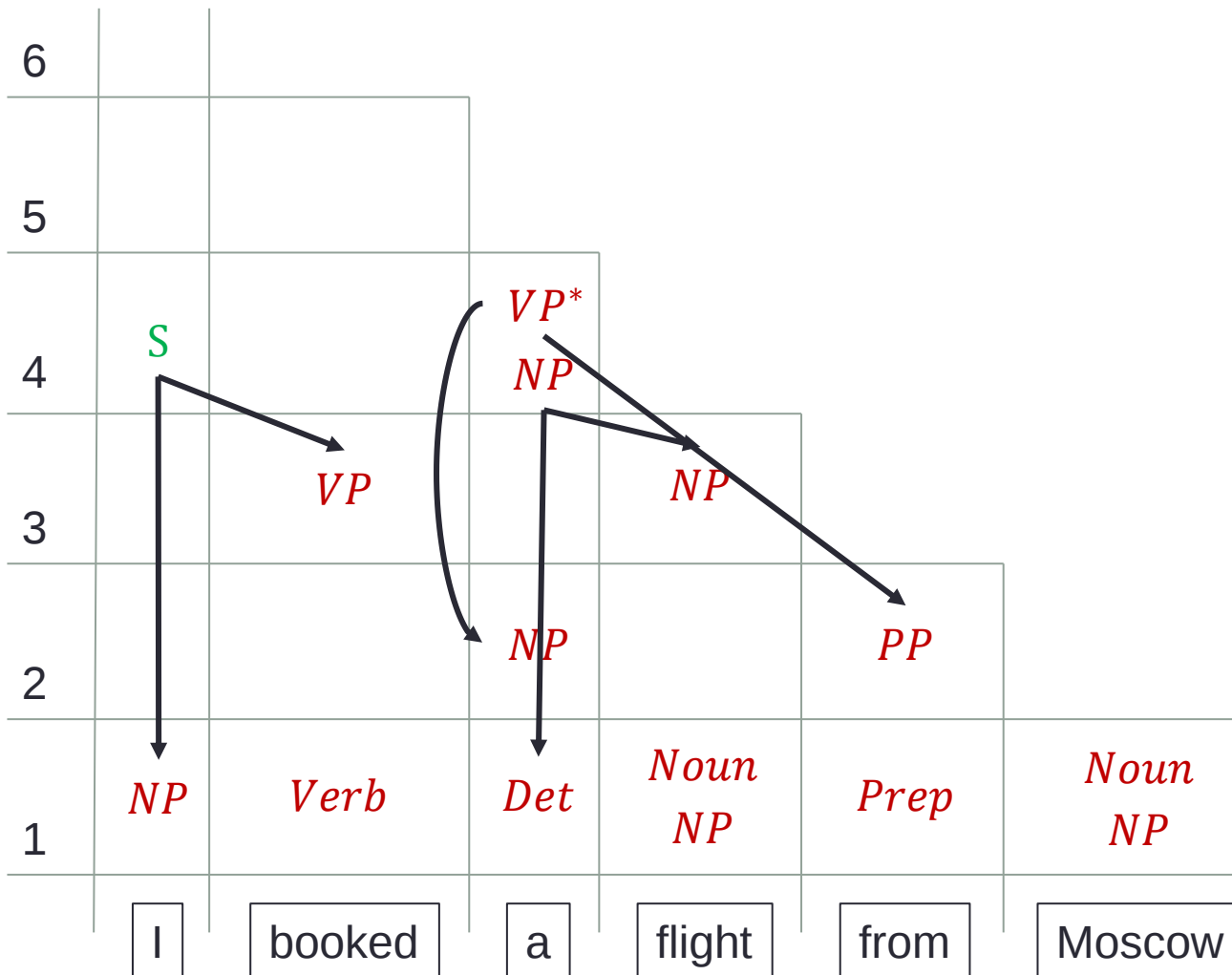
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК



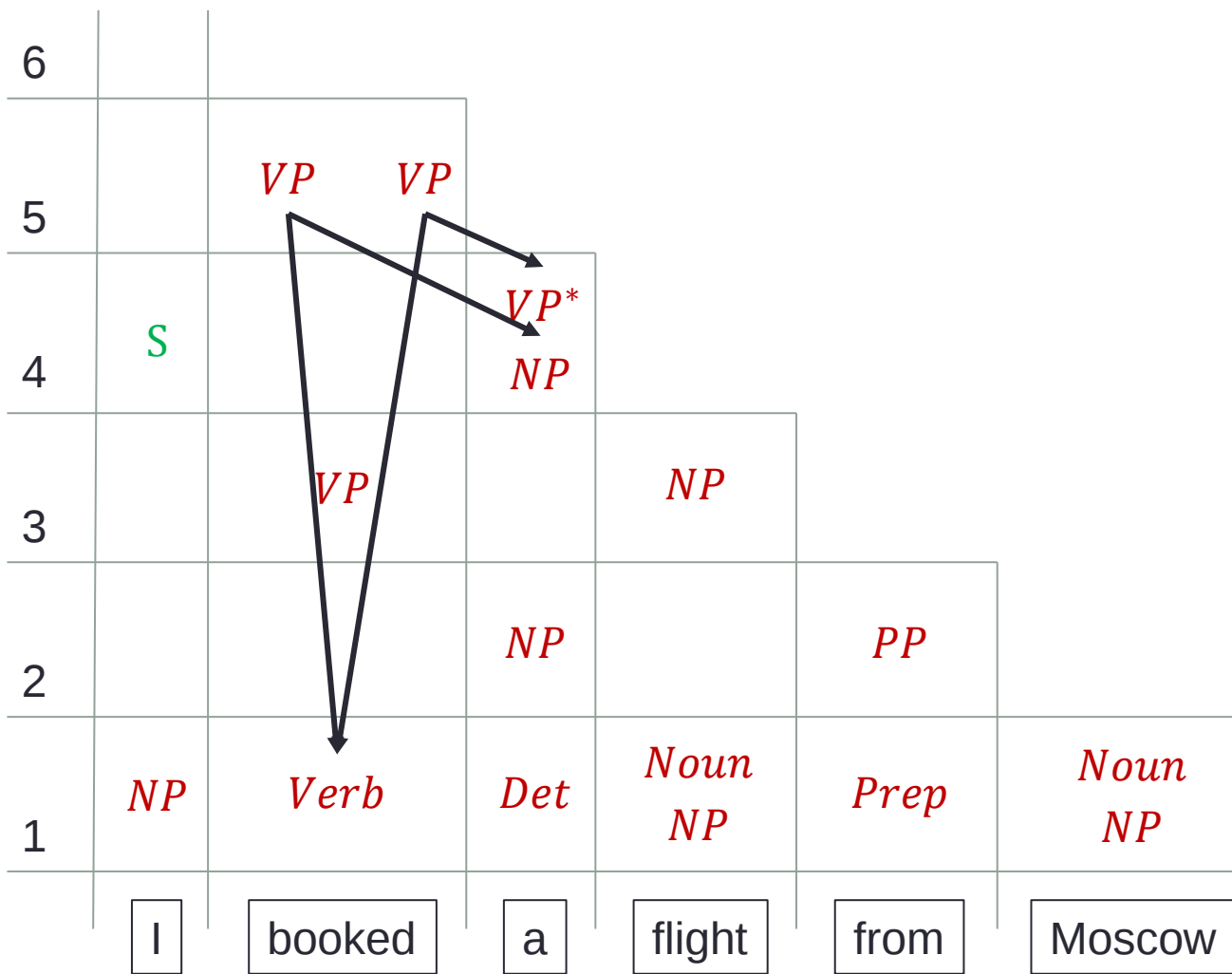
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК



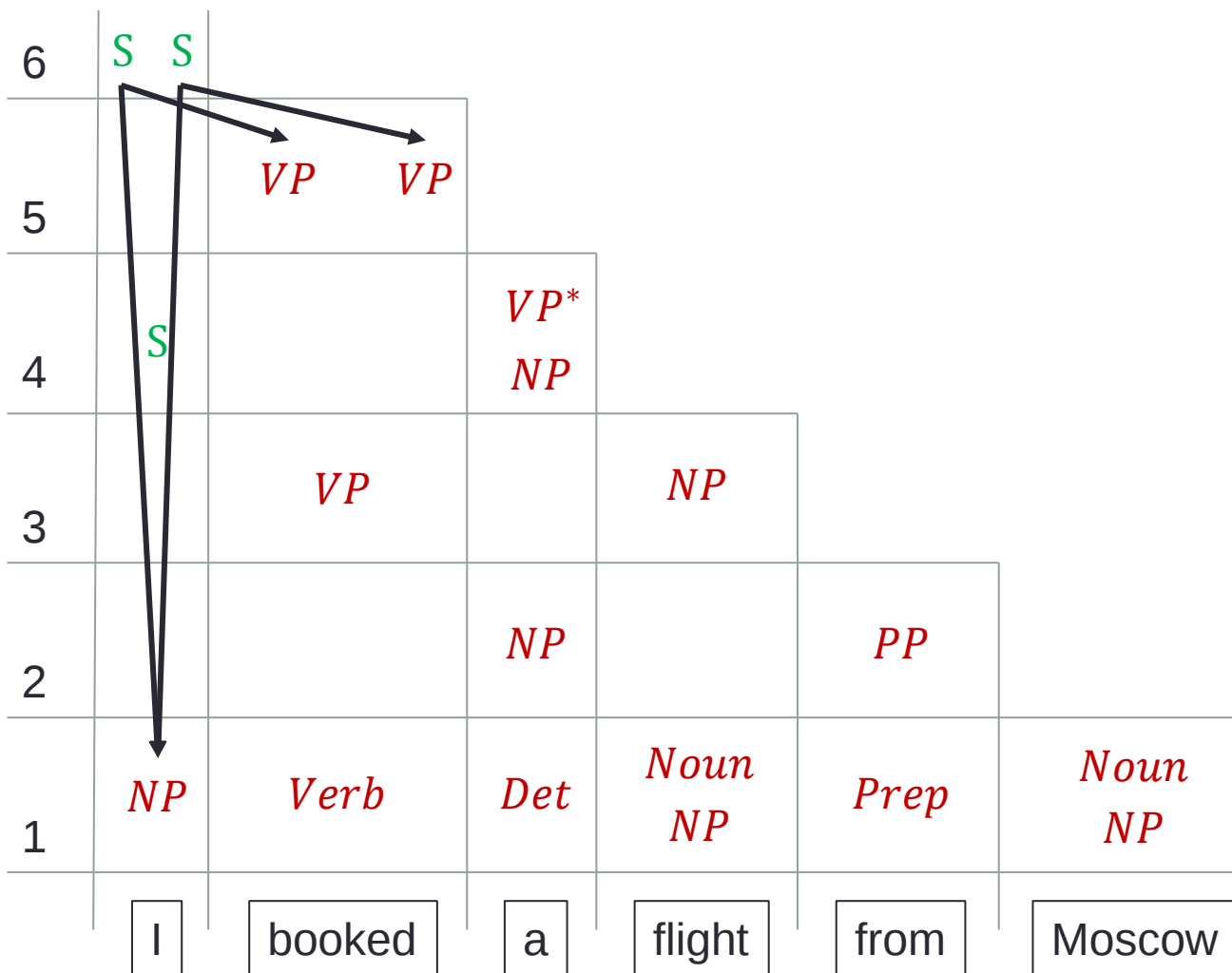
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК



1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Алгоритм СҮК



1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $PP \rightarrow Prep Noun$
8. $NP \rightarrow I$
9. $NP \rightarrow flight$
10. $NP \rightarrow Moscow$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Det \rightarrow a$
15. $Prep \rightarrow from$

Выбор лучшего дерева

- Зададим каждому правилу вероятность его применения
- Будем оценивать вероятность разбора, как произведение вероятностей правил, участвующих в нем

1. $S \rightarrow NP VP$	1.0
2. $VP \rightarrow Verb NP$	0.90
3. $VP \rightarrow Verb NP PP$	0.10
4. $NP \rightarrow Pro$	0.2
5. $NP \rightarrow Det NP$	0.3
6. $NP \rightarrow Noun PP$	0.1
7. $NP \rightarrow Noun$	0.4
8. $PP \rightarrow Prep Noun$	1.0
9. $Verb \rightarrow booked$	1.0
10. $Noun \rightarrow flight$	0.3
11. $Noun \rightarrow Moscow$	0.7

Построение грамматики

```
(S
  (NP
    (I Pro))
  (VP
    (booked Verb)
    (NP
      (a Det)
      (NP
        (flight Noun)
        (PP
          (from Prep)
          (Moscow Noun))))))
```

Построение грамматики

(S

1. $S \rightarrow NP VP$

(NP

(I Pro))

(VP

(booked Verb)

(NP

(a Det)

(NP

(flight Noun)

(PP

(from Prep)

(Moscow Noun))))))

Построение грамматики

(S

(NP

(I Pro))

(VP

(booked Verb)

(NP

(a Det)

(NP

(flight Noun)

(PP

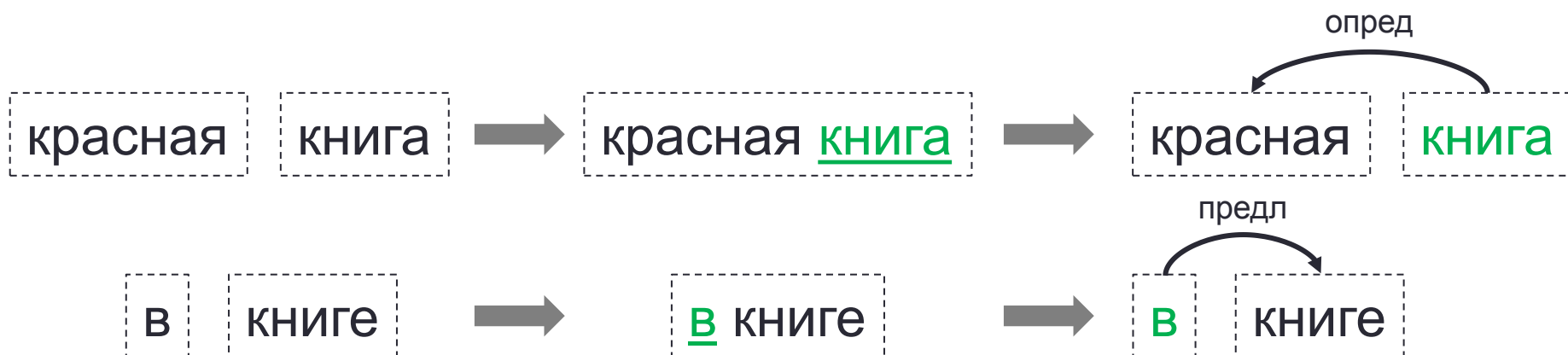
(from Prep)

(Moscow Noun))))))

1. $S \rightarrow NP VP$
2. $NP \rightarrow Pro$

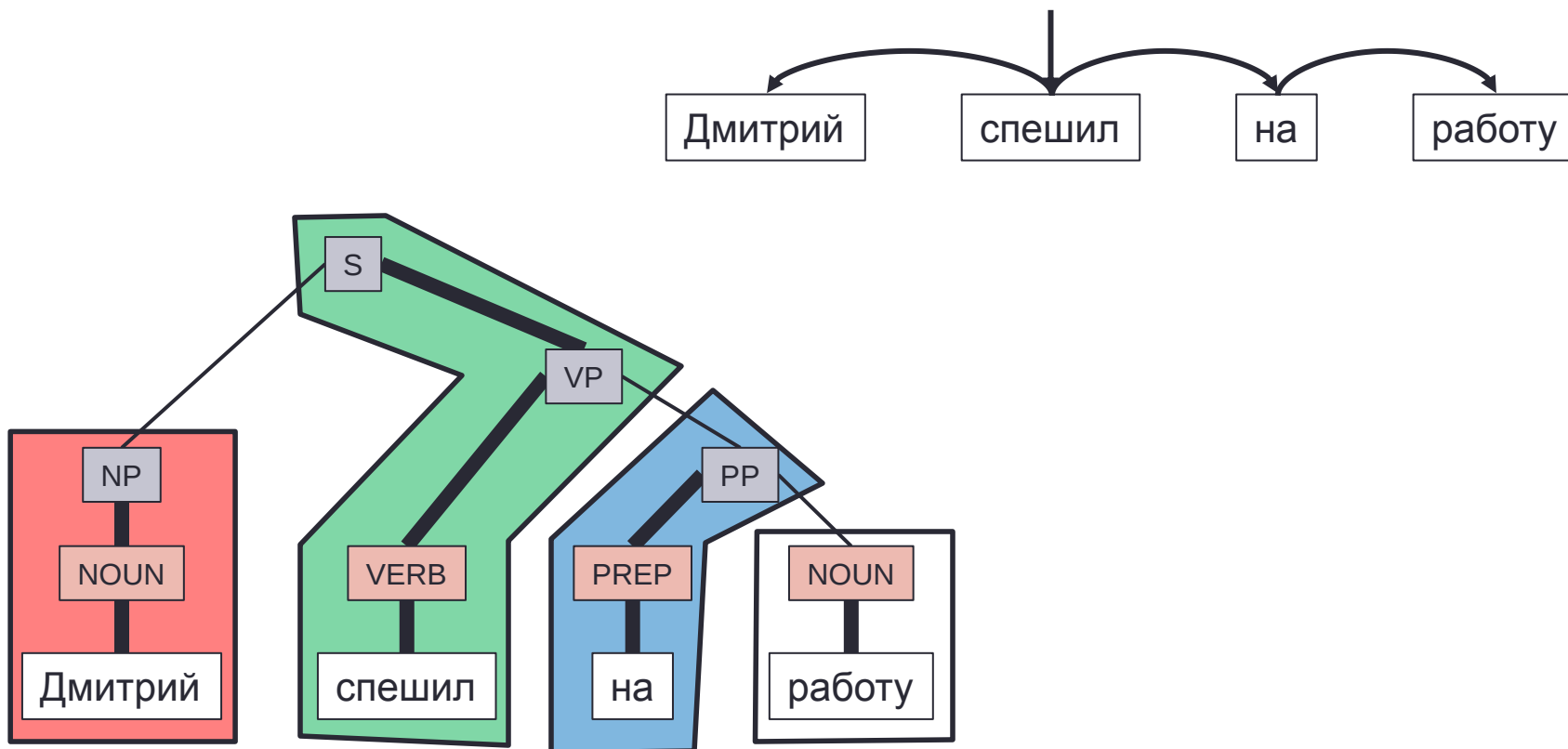
Грамматика зависимостей

- Формальная модель описания структуры словосочетаний (предложений)
- Описывается в виде иерархии слов, между которыми установлено бинарное отношение зависимости
- Различают типы зависимостей (соответствуют типам синтаксических правил)



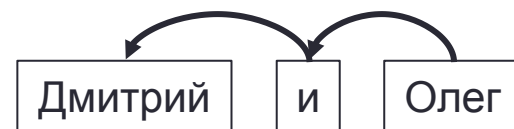
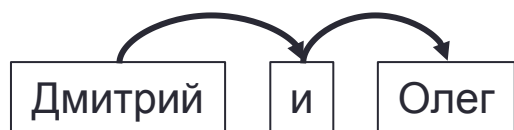
Грамматика зависимостей

- Дерево составляющих может быть преобразовано в дерево зависимостей

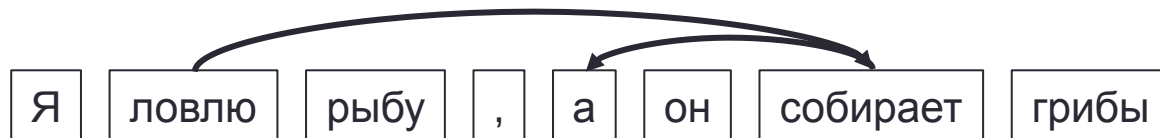


Грамматика зависимостей

- Не все синтаксические отношения подчинительные
 - Сочинительная группа



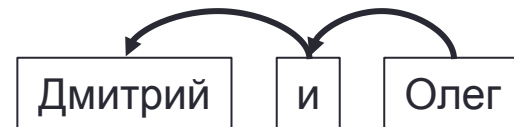
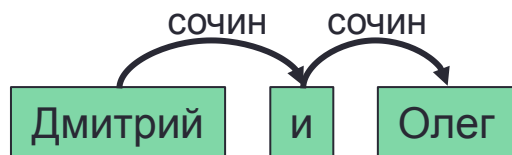
- Сложносочиненные предложения



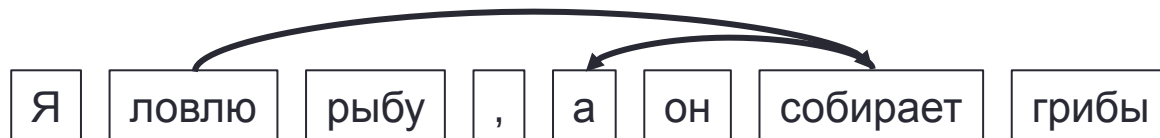
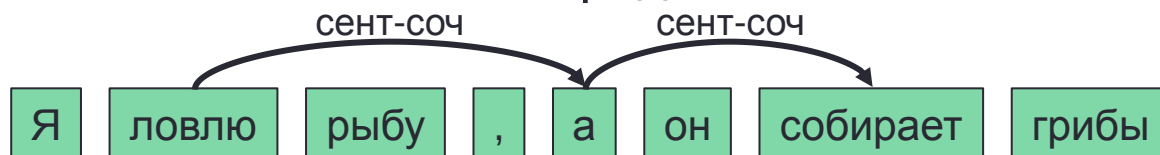
Грамматика зависимостей

- Не все синтаксические отношения подчинительные

- Сочинительная группа

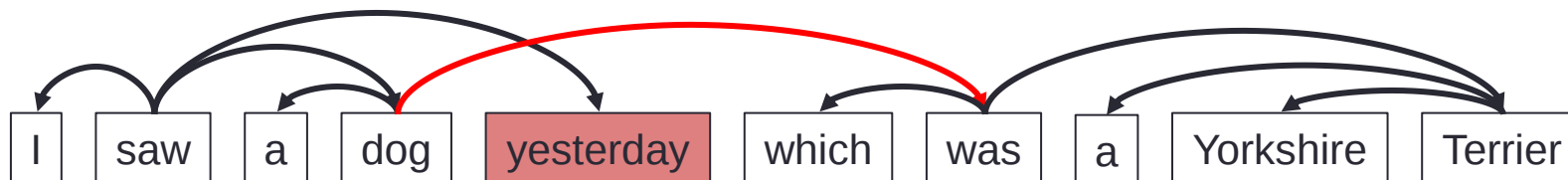


- Сложносочиненные предложения



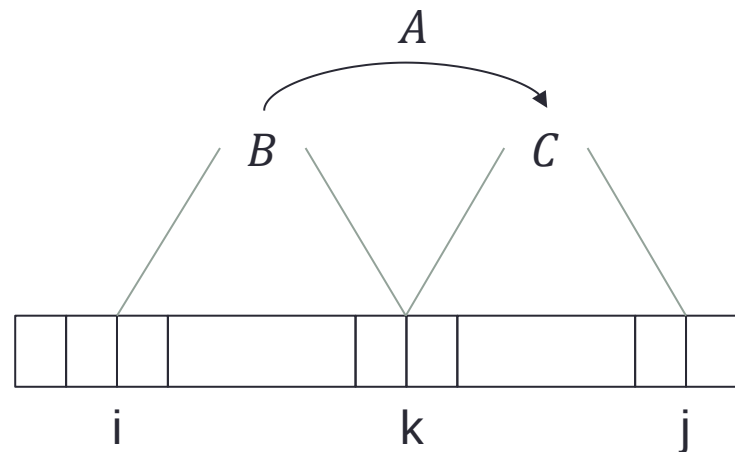
Проективное синтаксическое дерево

- Дерево зависимостей является проективным, если
 - Для каждой дуги в дереве существует направленный путь от главного слова до каждого из слов, расположенных между главным и зависимым словом этой дуги
 - $(i, l, j) \Rightarrow \forall k \in [\min(i, j), \max(i, j)], i \rightarrow^* k$



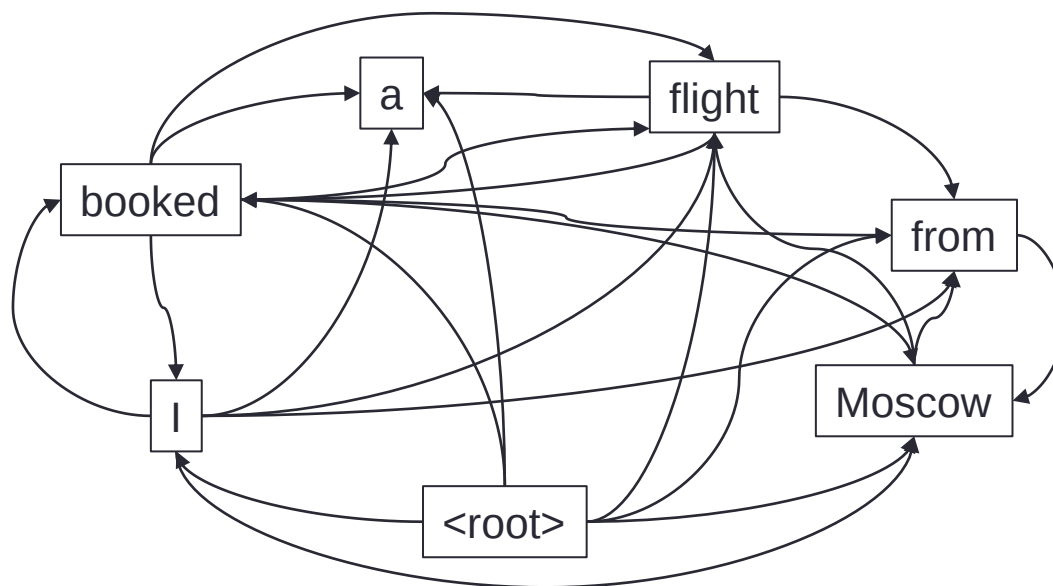
Методы динамического программирования

- Некоторое обобщение алгоритма СКУ для грамматик зависимостей
 - Строится дерево разбора предложения снизу-вверх
 - На каждом этапе рассматриваются подстроки исходной строки
 - Если подстрока состоит из двух деревьев, оценивается левая и правая дуга между ними:
 - $score(A) = score(B) + score(C) + score(arc)$
- Варианты
 - Алгоритм Коллинса. Сложность $O(n^5)$
 - Алгоритм Эйснера. Сложность $O(n^3)$



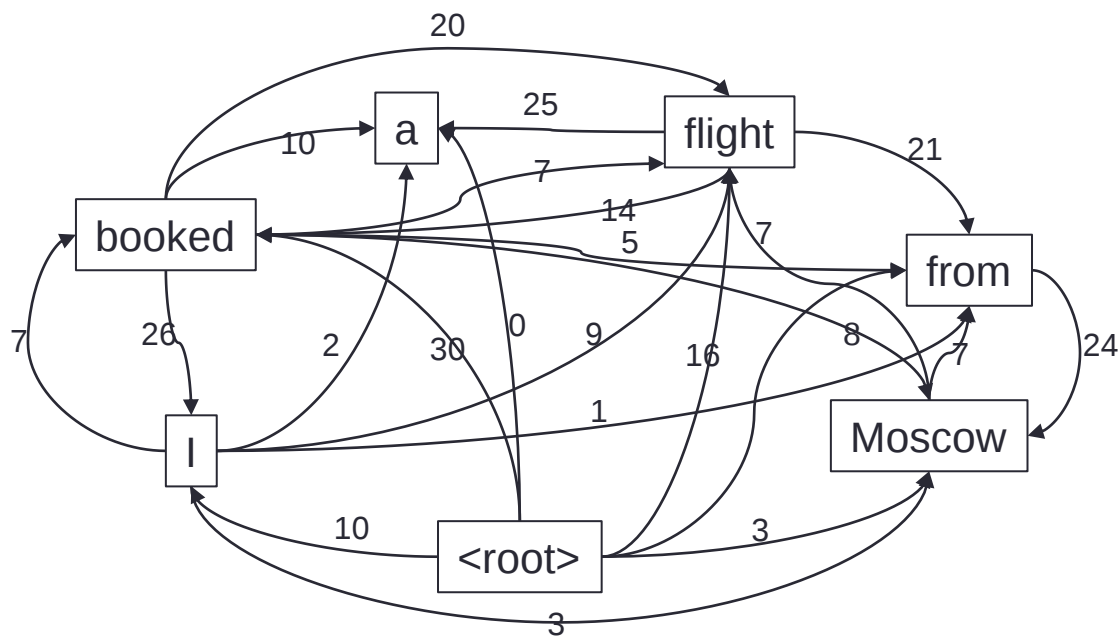
Графовые методы

- Предложение представляется в виде взвешенного полного графа (ориентированного)
 - Вершины графа – слова + root
 - Дуги графа – отношения между словами
- Цель
 - Найти подграф, являющийся деревом зависимостей



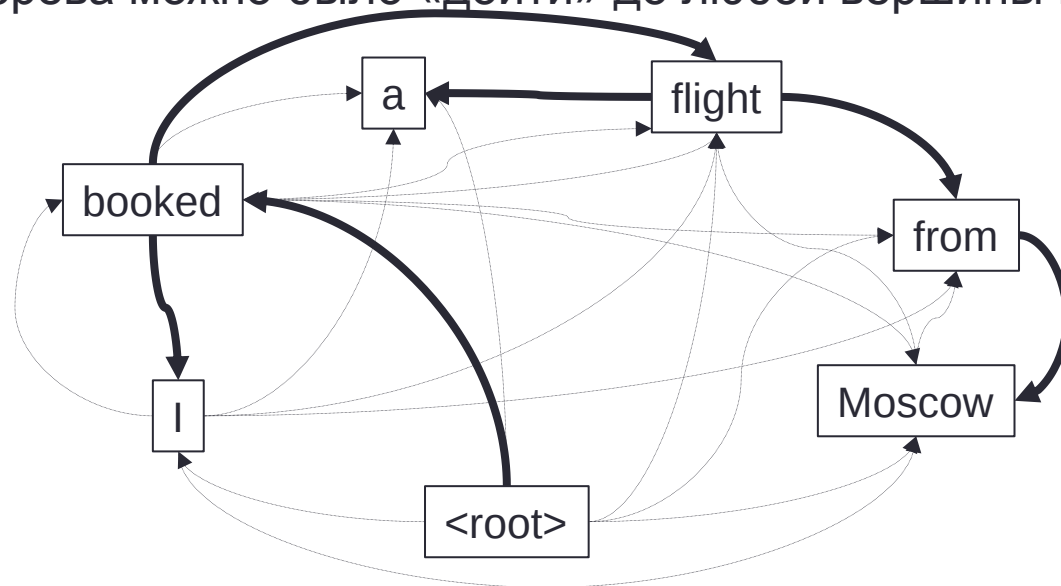
Графовые методы

- Для каждой дуги (i, j) вычисляется ее вес $s(i, j)$
- Вес подграфа в графе $s(x, y) = \sum_{(i,j) \in y} s(i, j)$, где
 - x – вершины подграфа,
 - y – дуги подграфа



Максимальное покрывающее дерево (MST)

- Задача синтаксического анализа формулируется как задача нахождения максимального покрывающего дерева (MST) в связанном взвешенном графе
 - Дерево называется покрывающим, если
 - Включает все вершины графа
 - Содержит минимальное количество дуг из графа, так, чтобы из корня дерева можно было «дойти» до любой вершины графа



Transition-based parsing

- Система переходов состоит из
 - C – множество конфигураций
 - T – множество переходов. $t: C \rightarrow C$
 - c_s – функция инициализации
 - $C_t \subseteq C$ – множество конечных конфигураций
- Конфигурация состоит из:
 - Σ – стек
 - B – буфер
 - A – множество зависимостей (i, l, j)
- Последовательность переходов для последовательности x :
 - $c_0 = c_s(x)$
 - $c_i = t(c_{i-1}), i = \overline{1..m}$
 - $c_m \in C_t$

Transition-based parsing

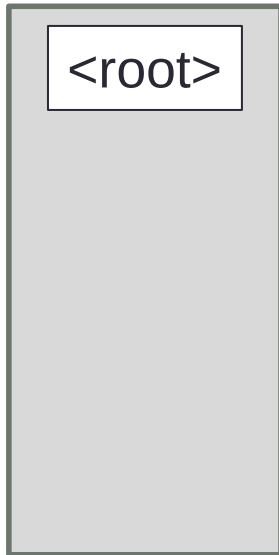
- Алгоритм
 - Получить начальную конфигурацию $c = c_s(x)$
 - Пока $c \notin C_t$
 - Выбрать переход $t = \underset{t}{\operatorname{argmax}} \operatorname{Score}(c, t)$
 - Выполнить переход $c = t(c)$
 - Извлечь множество зависимостей из $c.A$

Arc-standard

- Инициализация
 - $c_s(x = x_1 \dots x_n) = ([0], [1, \dots, n], \emptyset)$
- Конечное состояние
 - $C_t = \{c \in C \mid c = ([0], [], A)\}$
- Переходы
 - (*Shift*) $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$
 - (*LeftArc_l*) $([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$, если $i \neq 0$
 - (*RightArc_l*) $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

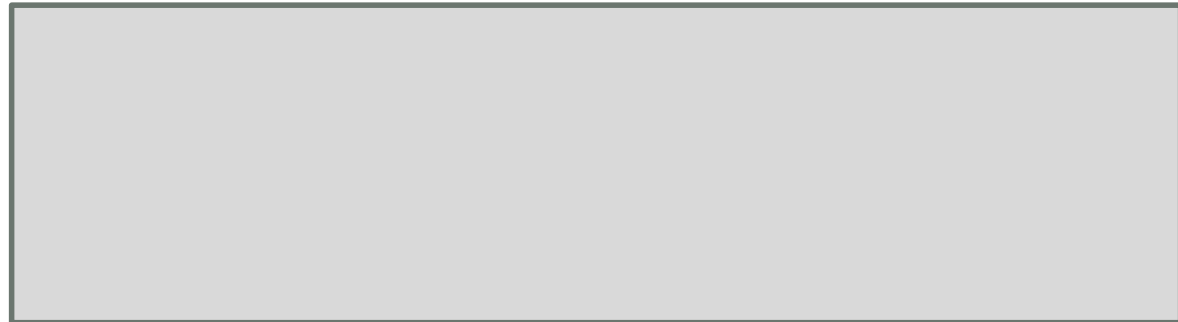
Стек



Буфер

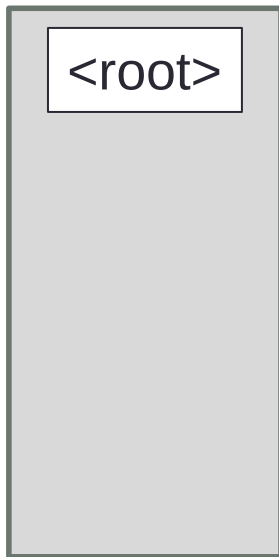


Зависимости



Arc-standard

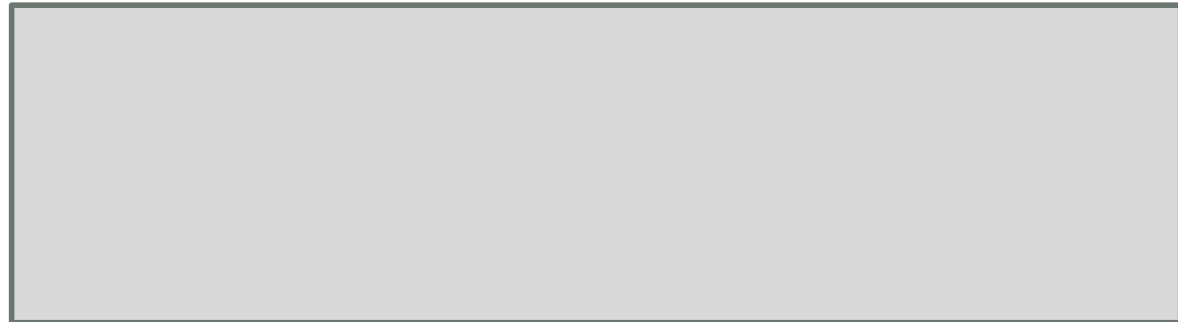
Стек



Буфер



Зависимости

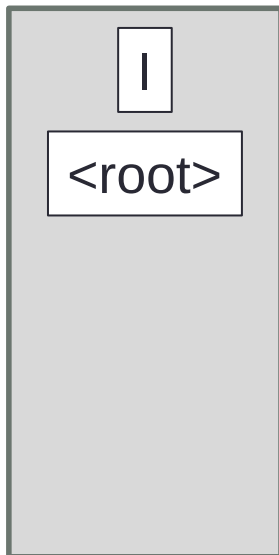


Shift



Arc-standard

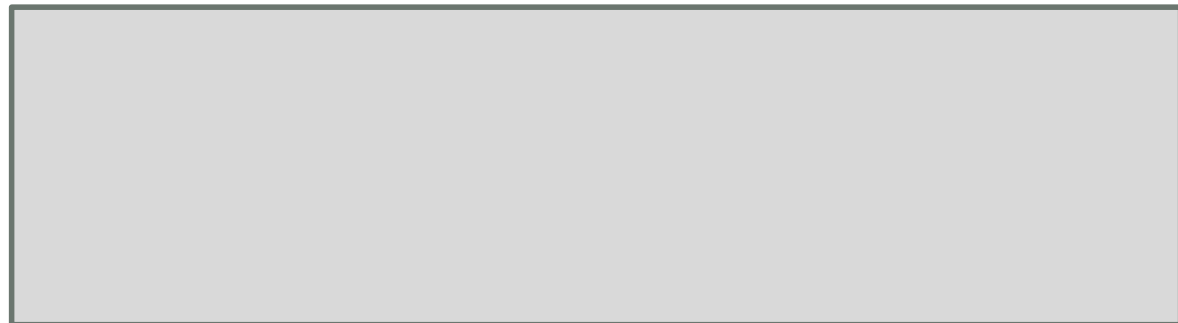
Стек



Буфер

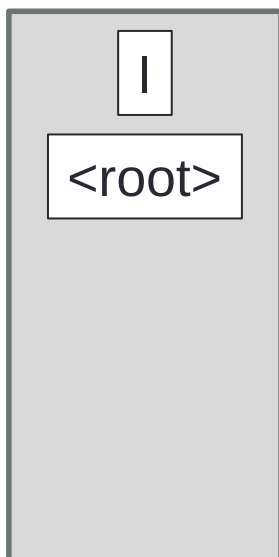


Зависимости

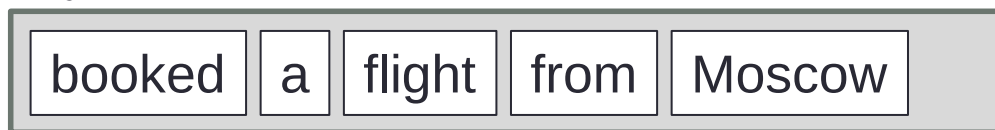


Arc-standard

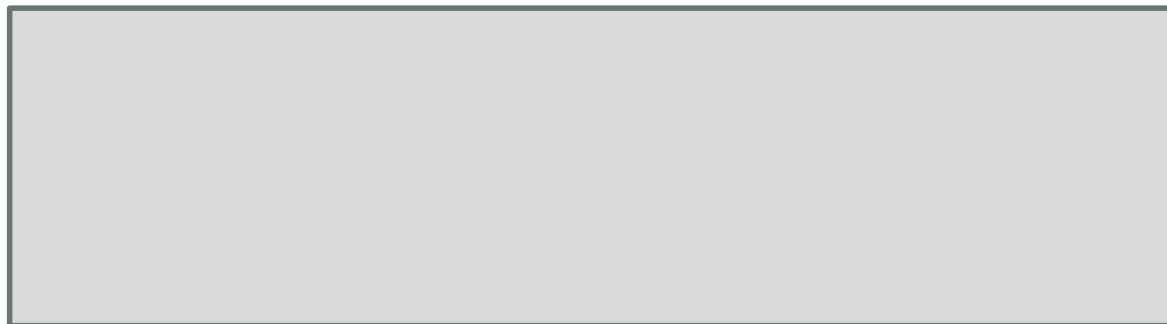
Стек



Буфер



Зависимости

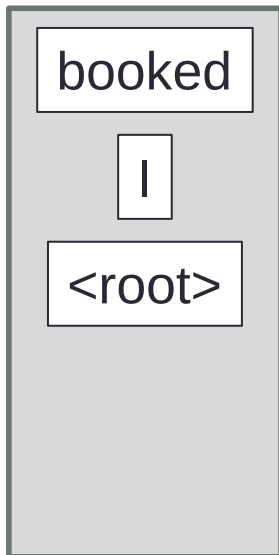


Shift



Arc-standard

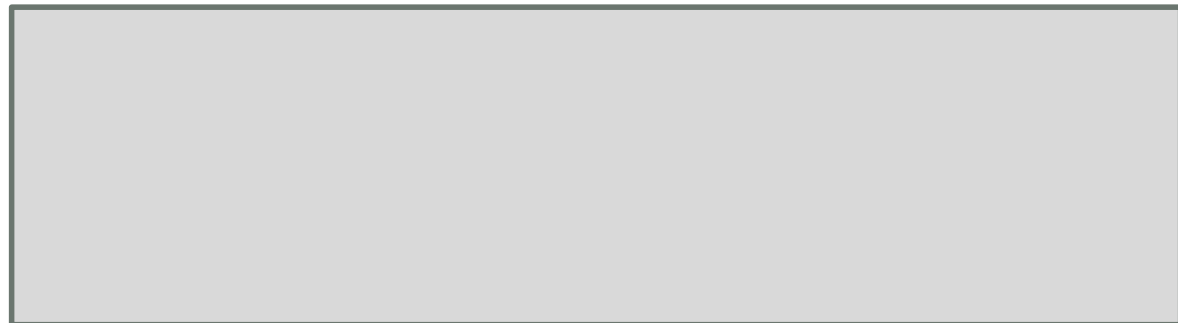
Стек



Буфер

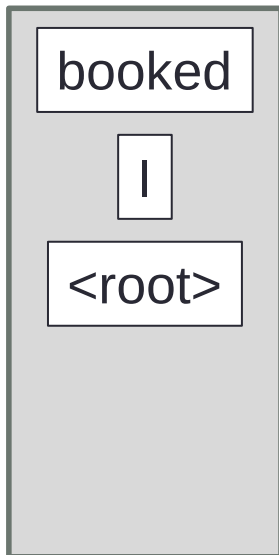


Зависимости



Arc-standard

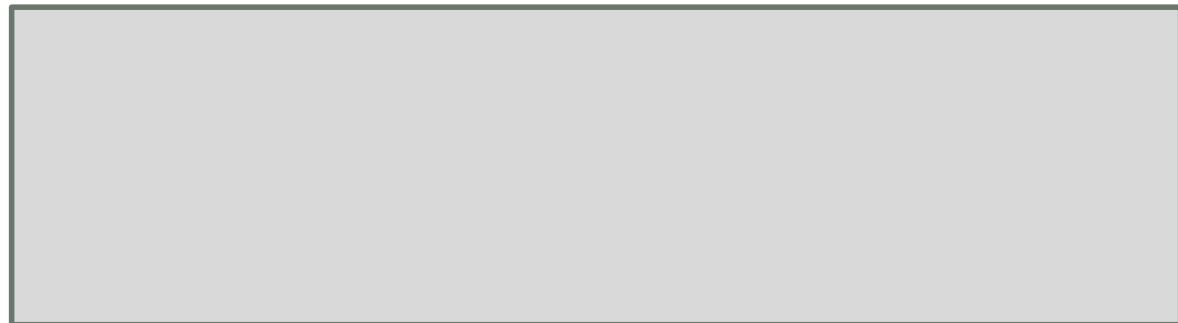
Стек



Буфер



Зависимости

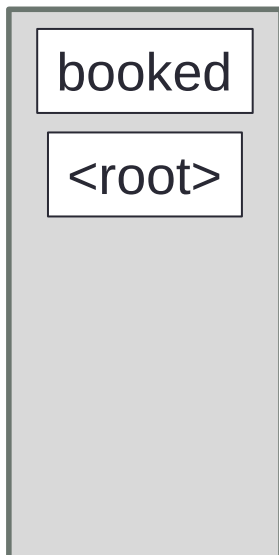


LeftArc_{subj}

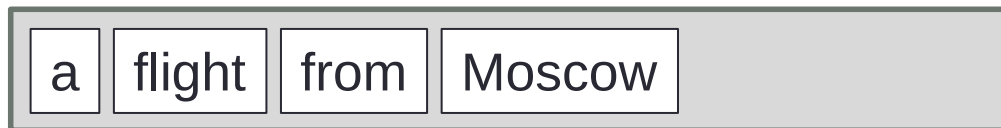


Arc-standard

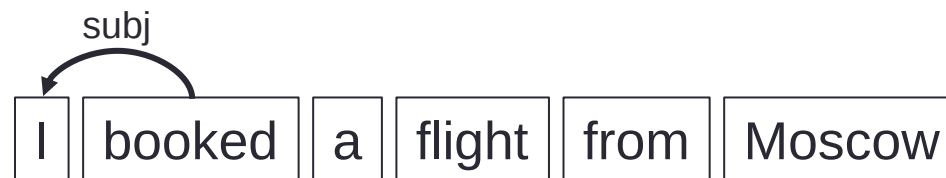
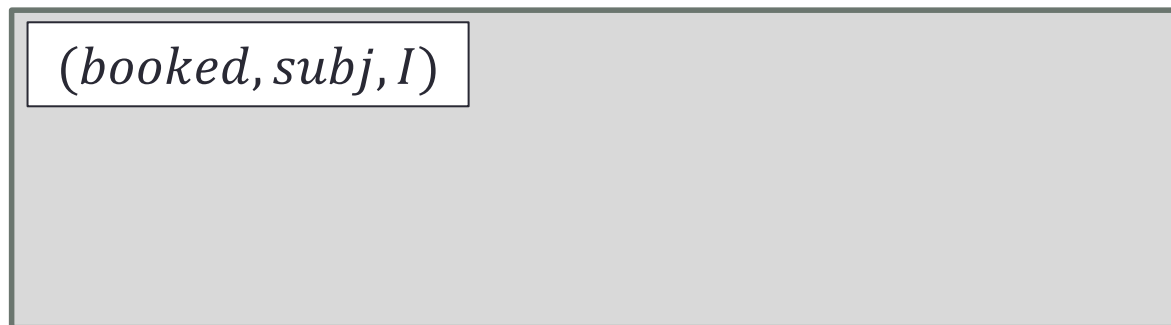
Стек



Буфер

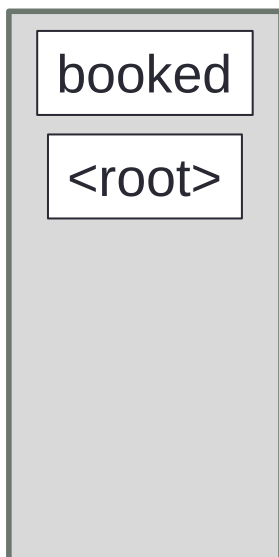


Зависимости

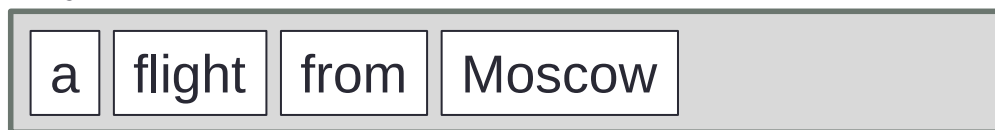


Arc-standard

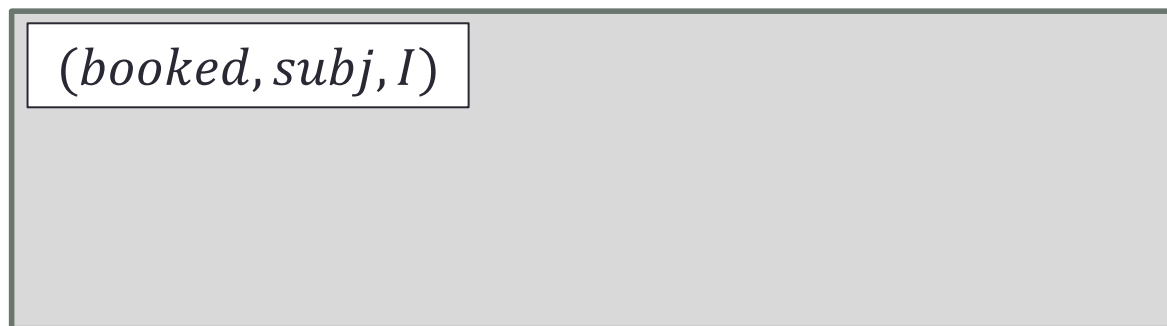
Стек



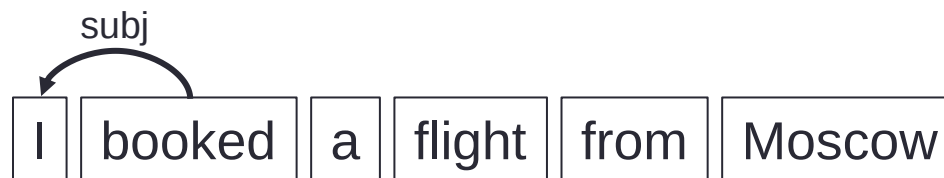
Буфер



Зависимости

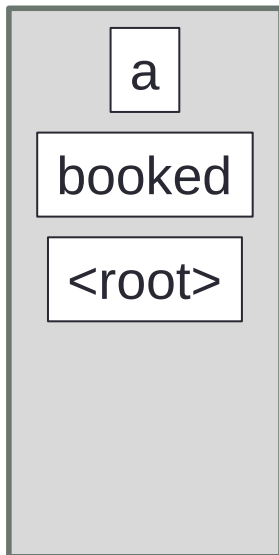


Shift



Arc-standard

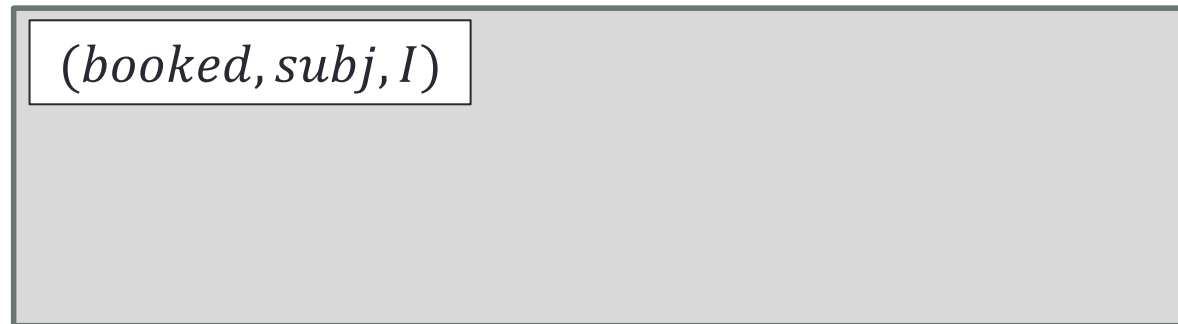
Стек



Буфер

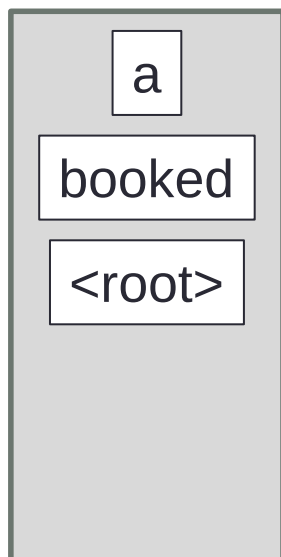


Зависимости



Arc-standard

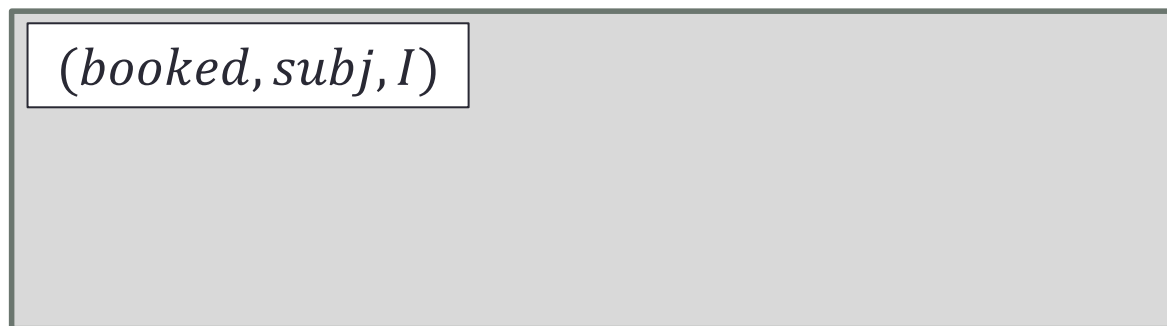
Стек



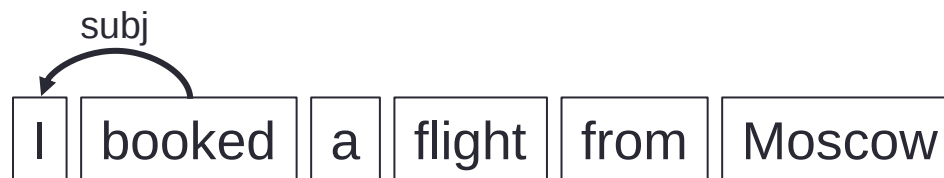
Буфер



Зависимости

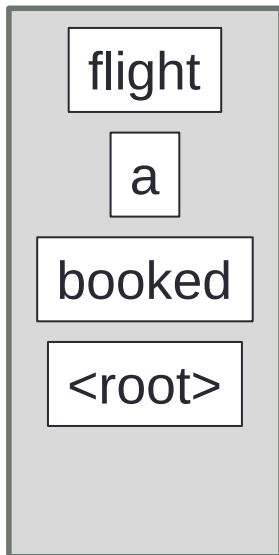


Shift



Arc-standard

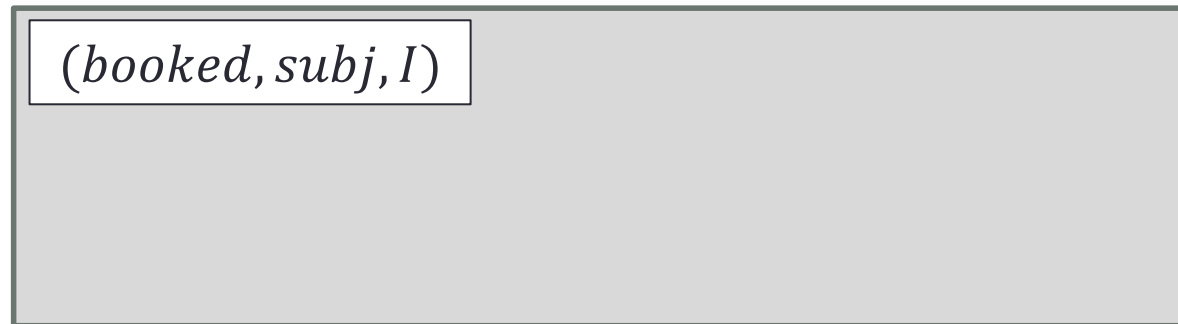
Стек



Буфер

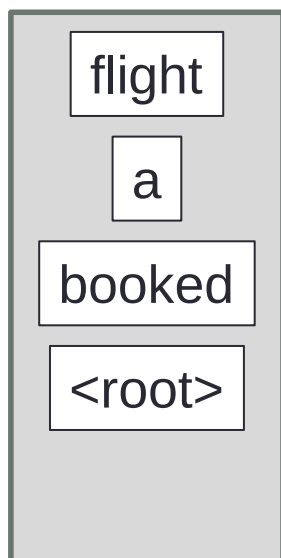


Зависимости



Arc-standard

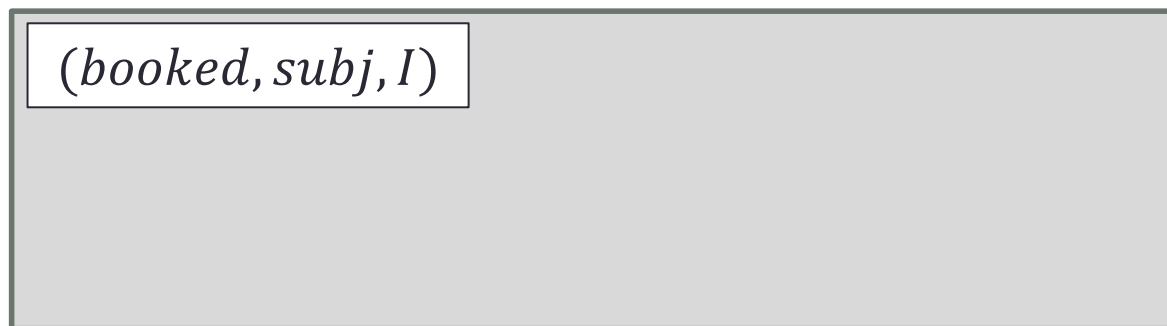
Стек



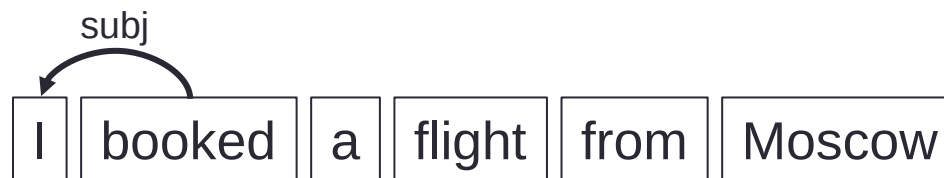
Буфер



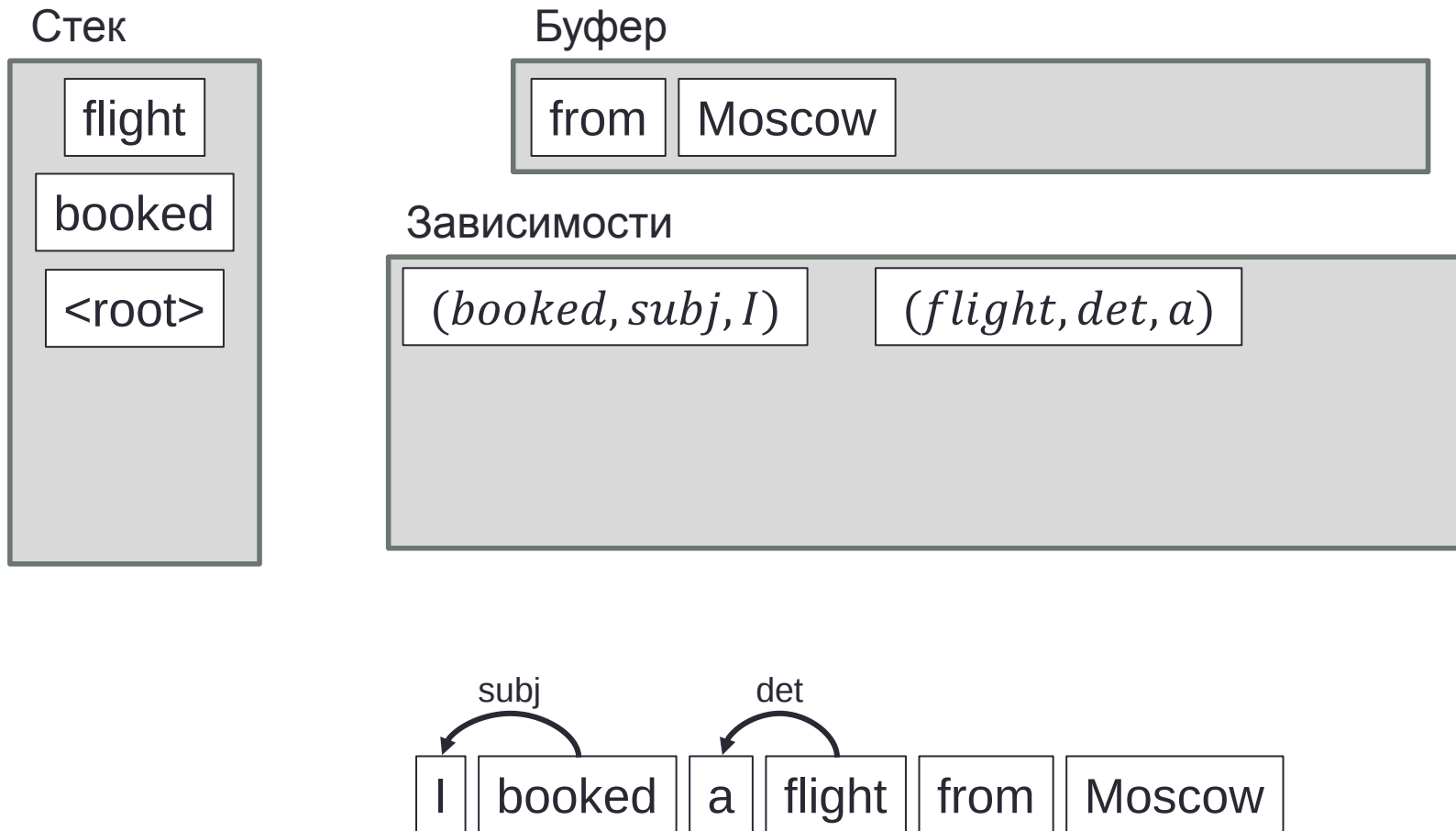
Зависимости



LeftArc_{det}

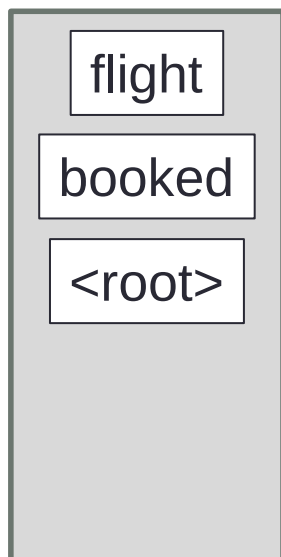


Arc-standard

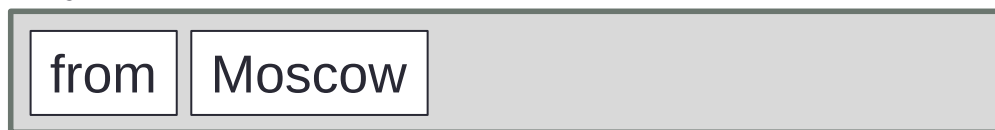


Arc-standard

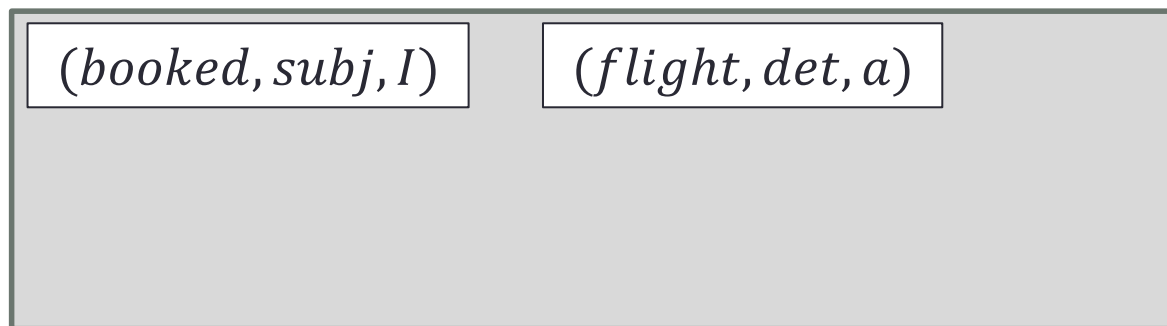
Стек



Буфер



Зависимости

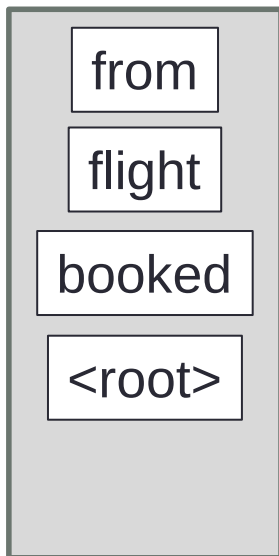


Shift

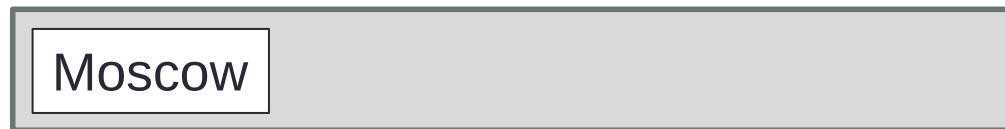


Arc-standard

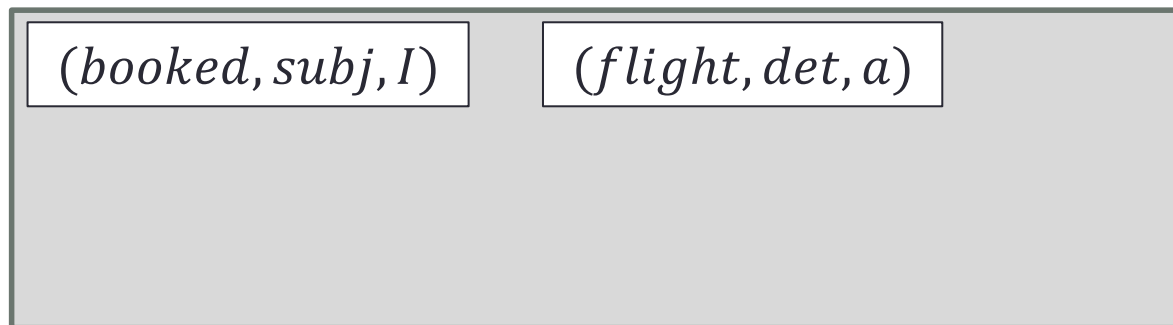
Стек



Буфер

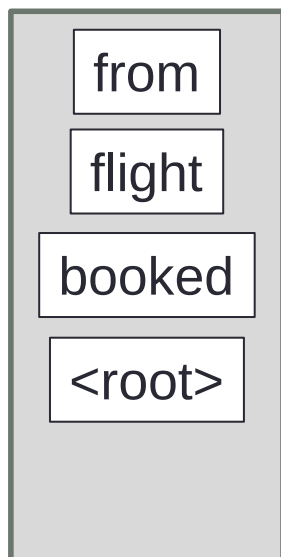


ЗАВИСИМОСТИ

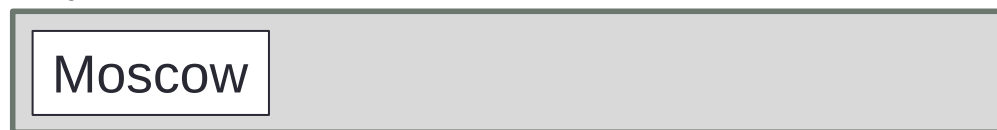


Arc-standard

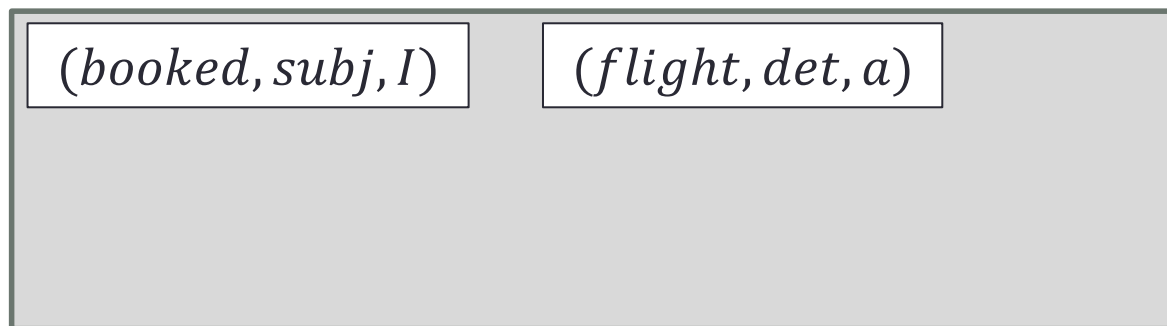
Стек



Буфер



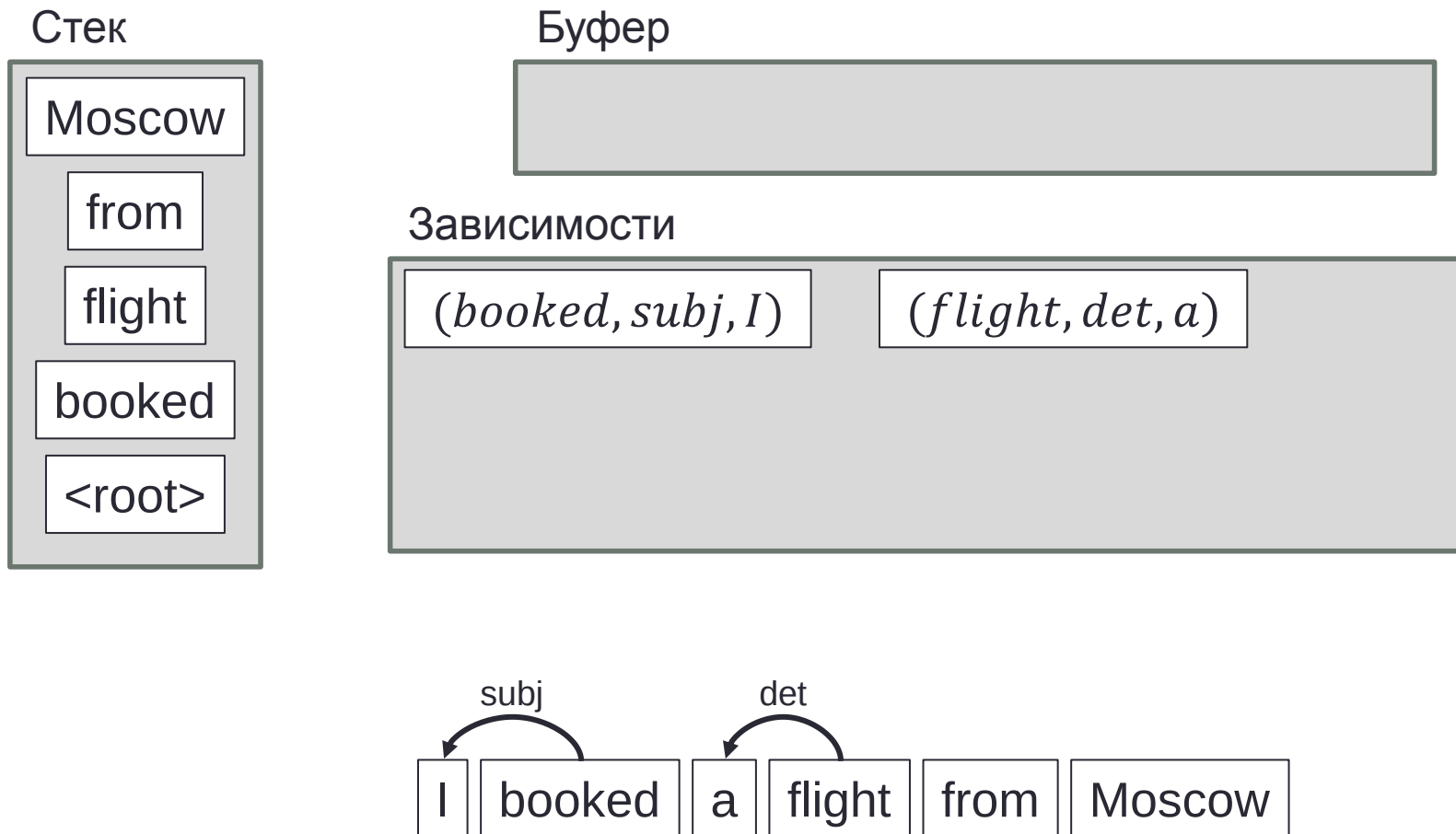
ЗАВИСИМОСТИ



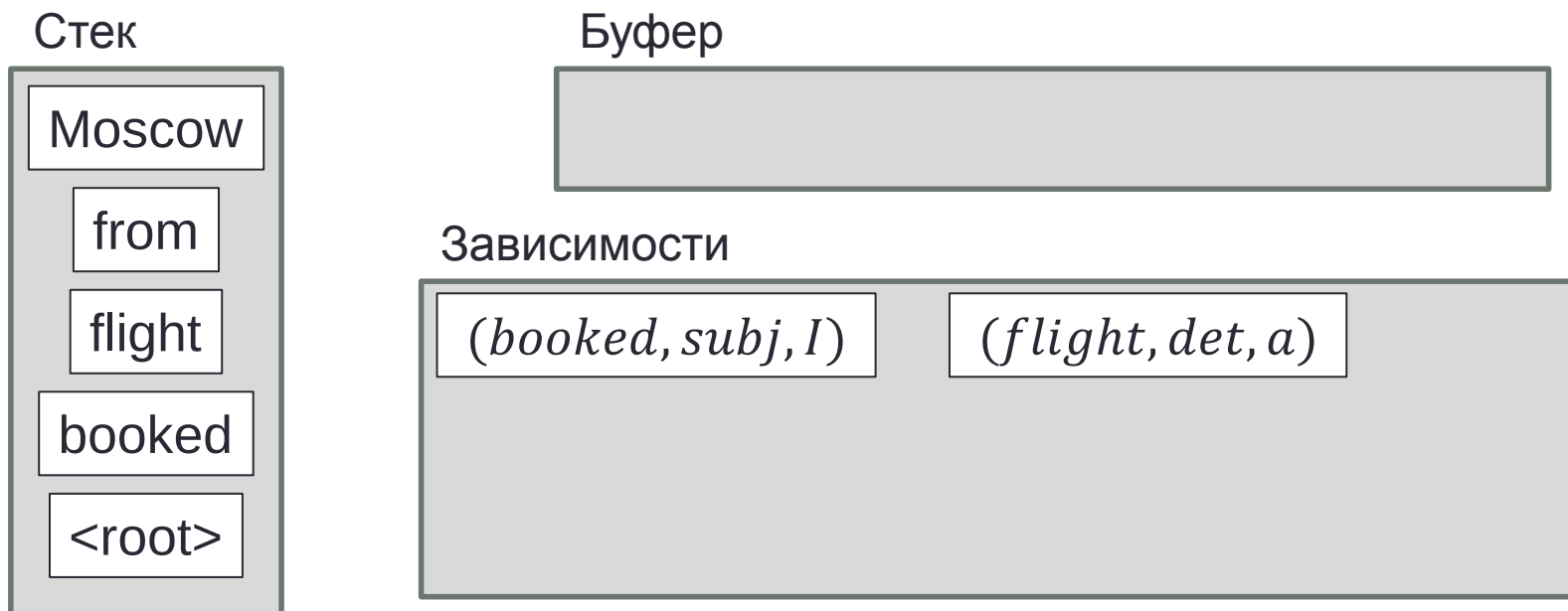
Shift



Arc-standard

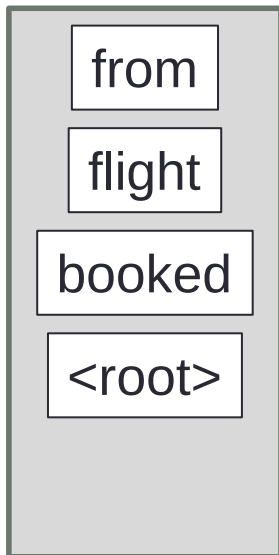


Arc-standard

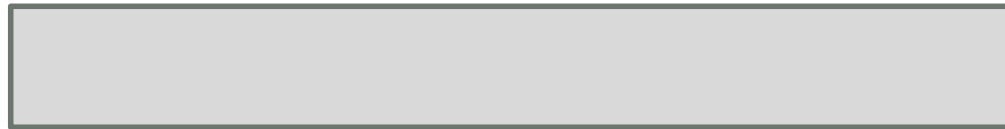


Arc-standard

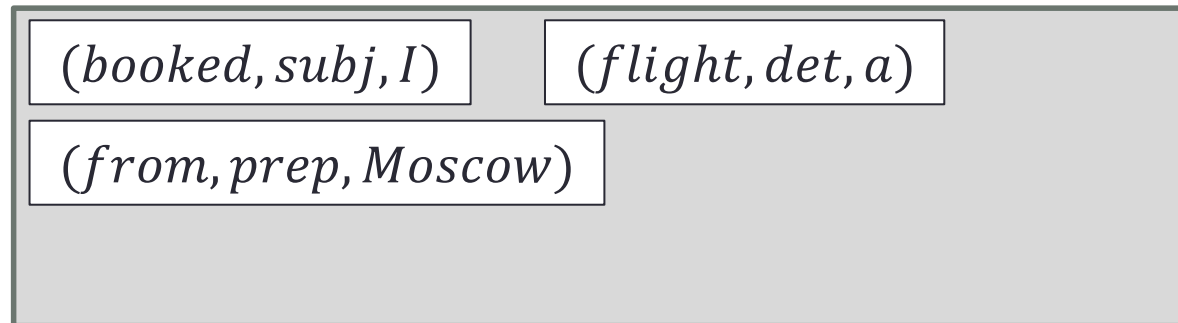
Стек



Буфер

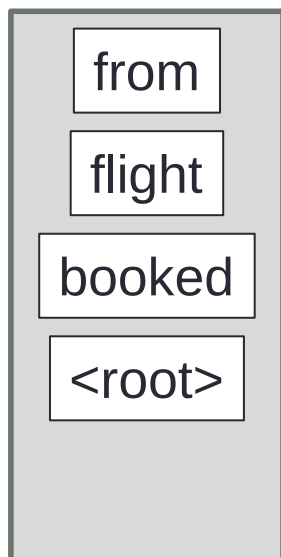


ЗАВИСИМОСТИ

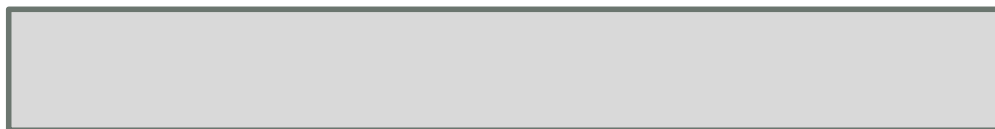


Arc-standard

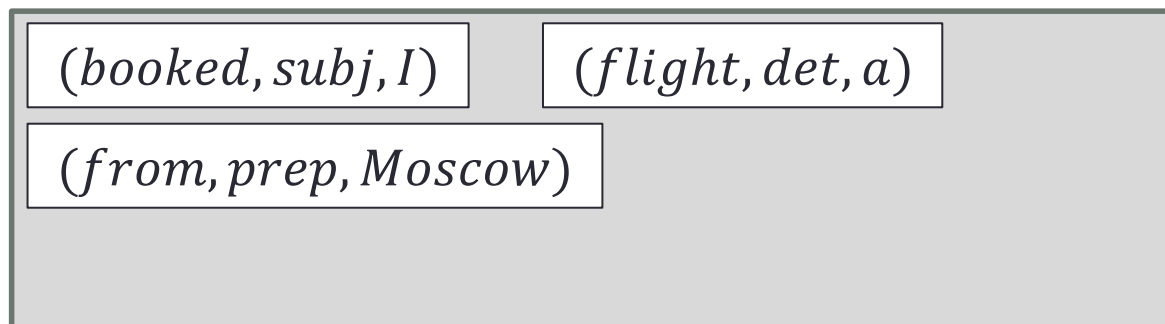
Стек



Буфер



ЗАВИСИМОСТИ

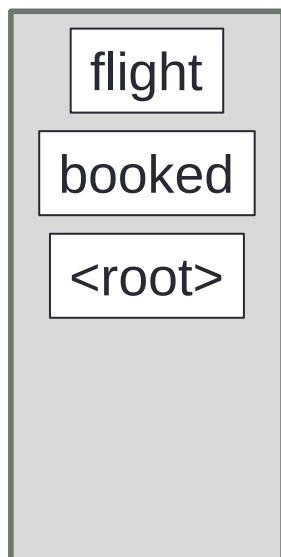


RightArc_{pmod}

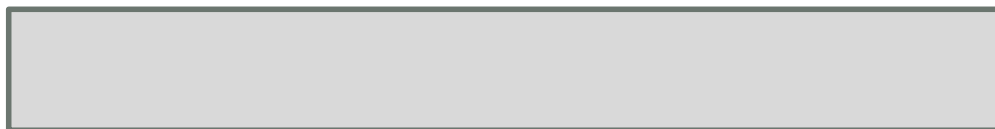


Arc-standard

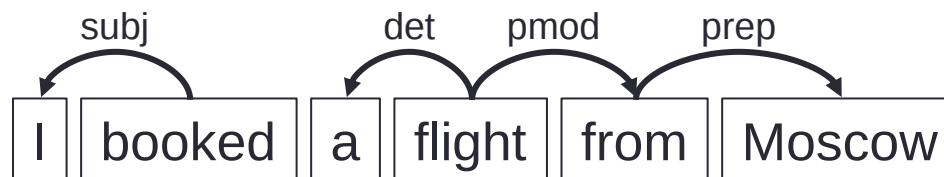
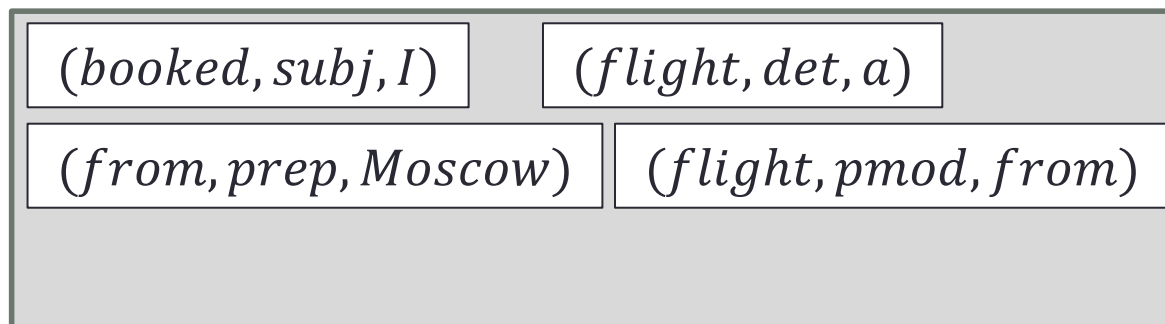
Стек



Буфер

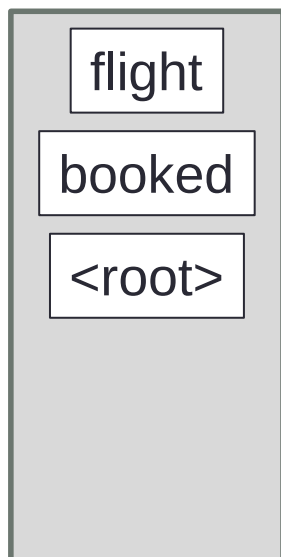


ЗАВИСИМОСТИ

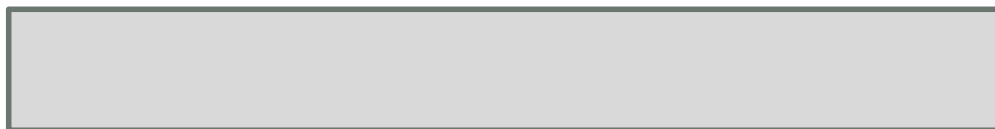


Arc-standard

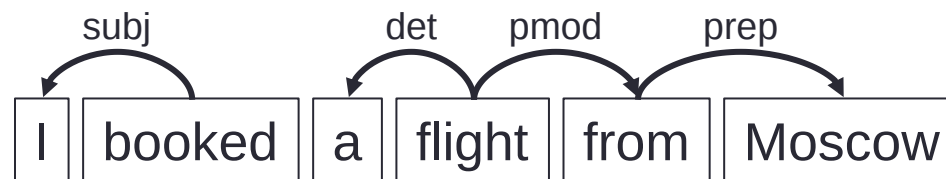
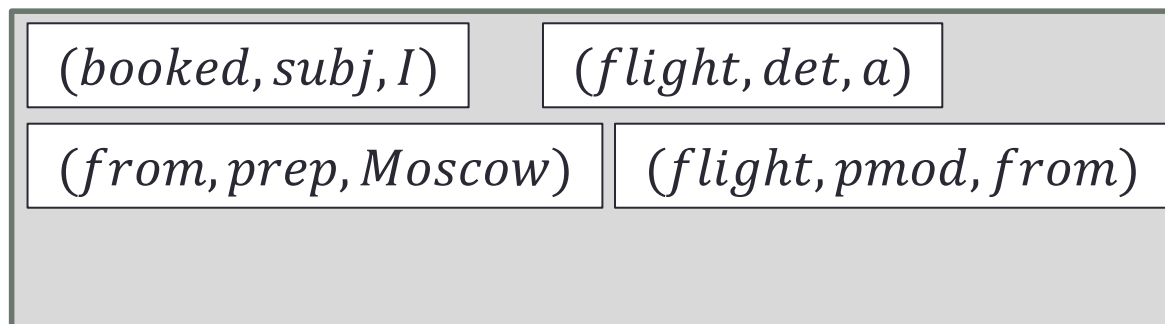
Стек



Буфер



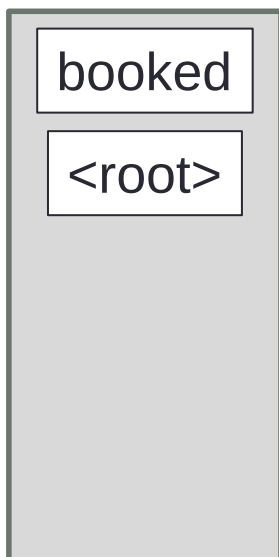
ЗАВИСИМОСТИ



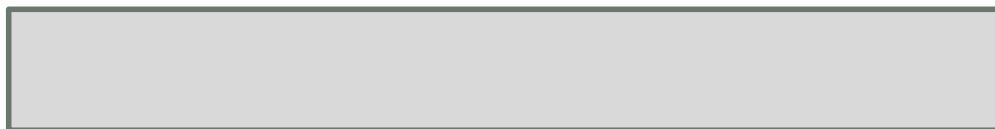
RightArc_{dobj}

Arc-standard

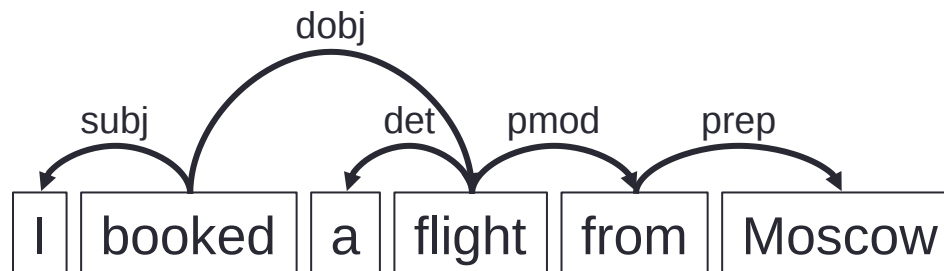
Стек



Буфер

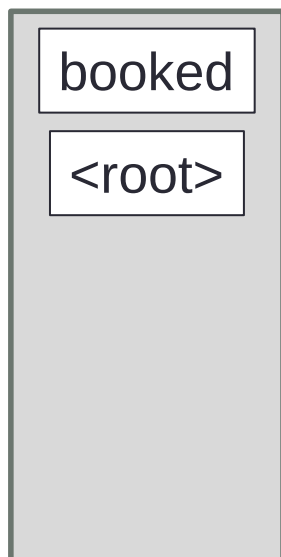


ЗАВИСИМОСТИ

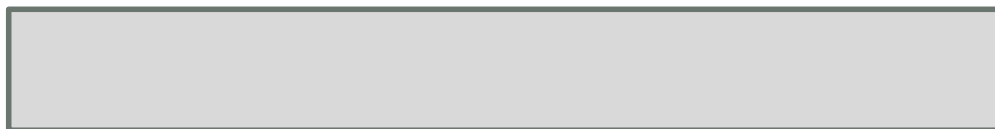


Arc-standard

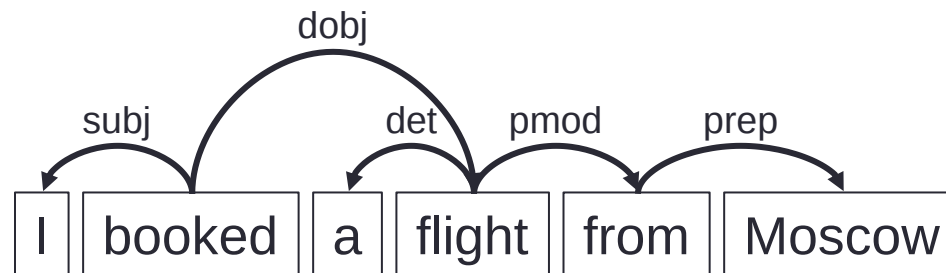
Стек



Буфер



ЗАВИСИМОСТИ



RightArc_{root}

Arc-standard

Стек

<root>

Буфер

ЗАВИСИМОСТИ

(booked, subj, I)

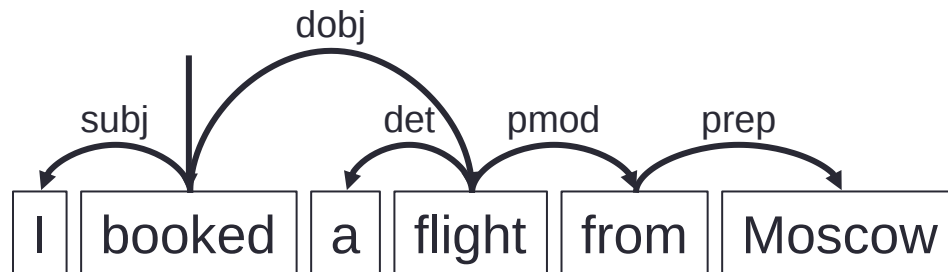
(flight, det, a)

(from, prep, Moscow)

(flight, pmod, from)

(booked, dobj, flight)

(root, root, booked)



Arc-standard

Стек

<root>

Буфер

ЗАВИСИМОСТИ

(booked, subj, I)

(flight, det, a)

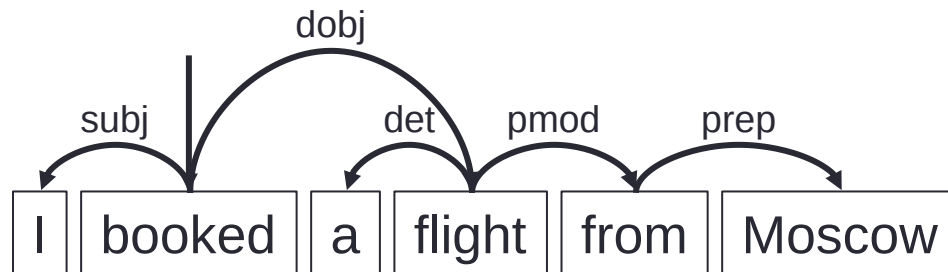
(from, prep, Moscow)

(flight, pmod, from)

(booked, dobj, flight)

(root, root, booked)

DONE!!!



Arc-eager

- Инициализация

- $c_s(x = x_1 \dots x_n) = ([0], [1, \dots, n], \emptyset)$

- Конечное состояние

- $C_t = \{c \in C \mid c = (\Sigma, [], A)\}$

- Переходы

- (*Shift*) $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

- (*LeftArc_l*) $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\}),$
если $i \neq 0$ и $\nexists k: (k, *, i) \in A$

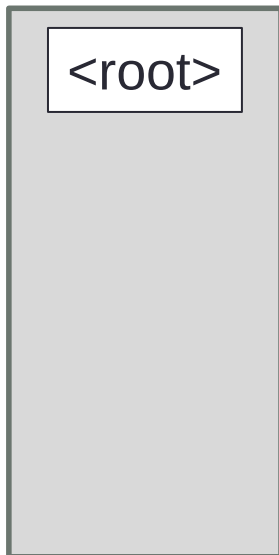
- (*RightArc_l*) $([\sigma|i], [j|\beta], A) \rightarrow ([\sigma|i]j, \beta, A \cup \{(i, l, j)\})$

- (*Reduce*) $([\sigma|i], B, A) \rightarrow (\sigma, B, A)$

если $\exists k: (k, *, i) \in A$

Arc-eager

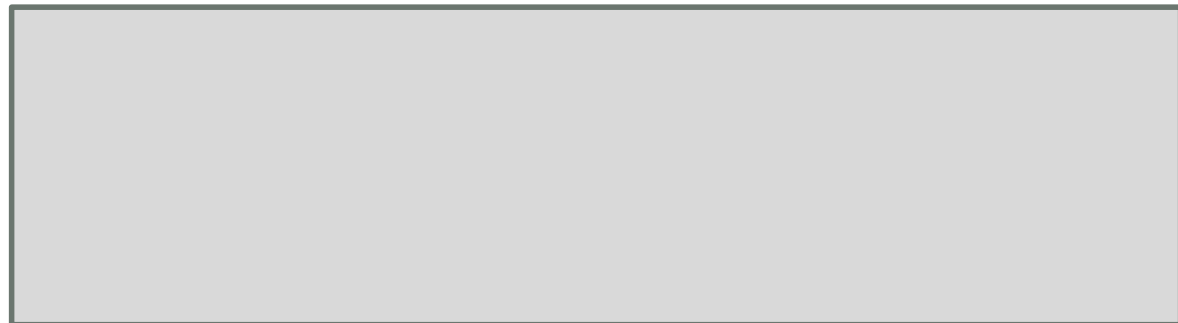
Стек



Буфер

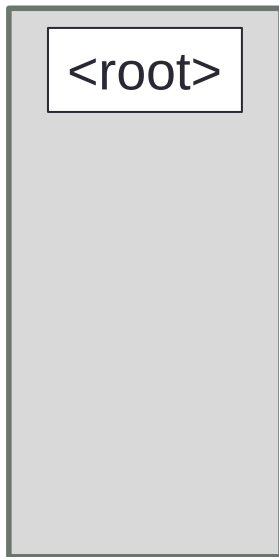


Зависимости



Arc-eager

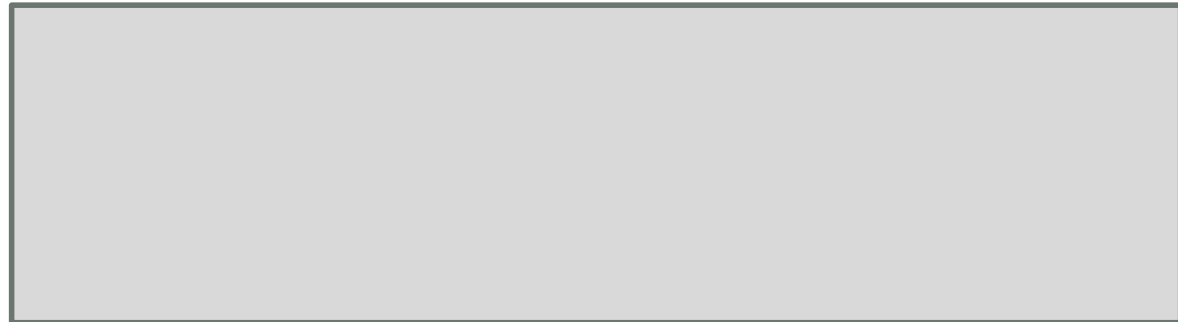
Стек



Буфер



Зависимости

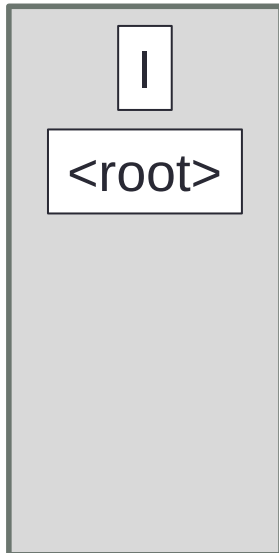


Shift

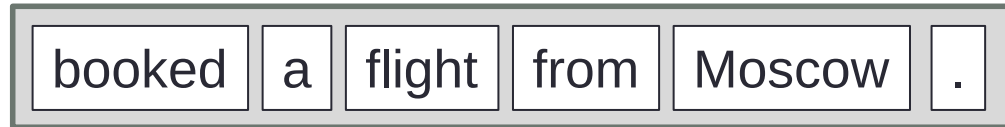


Arc-eager

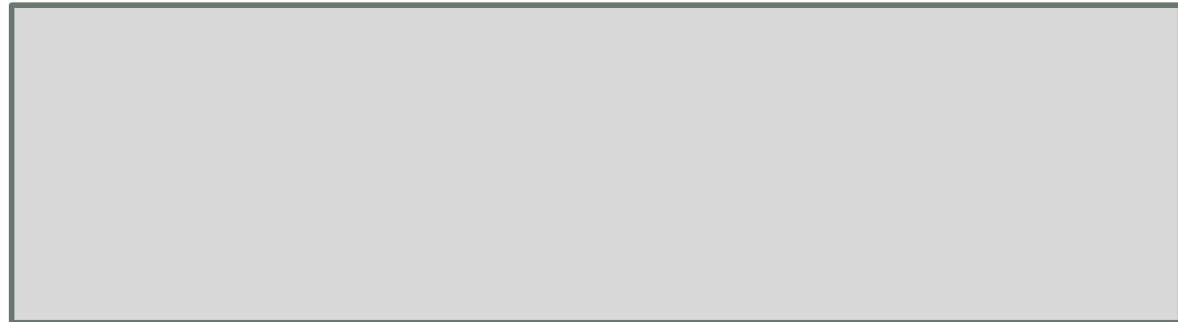
Стек



Буфер

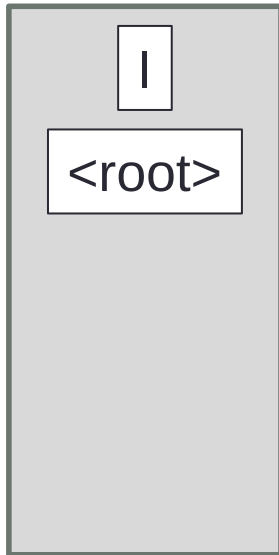


Зависимости

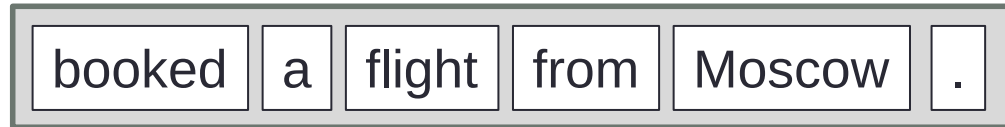


Arc-eager

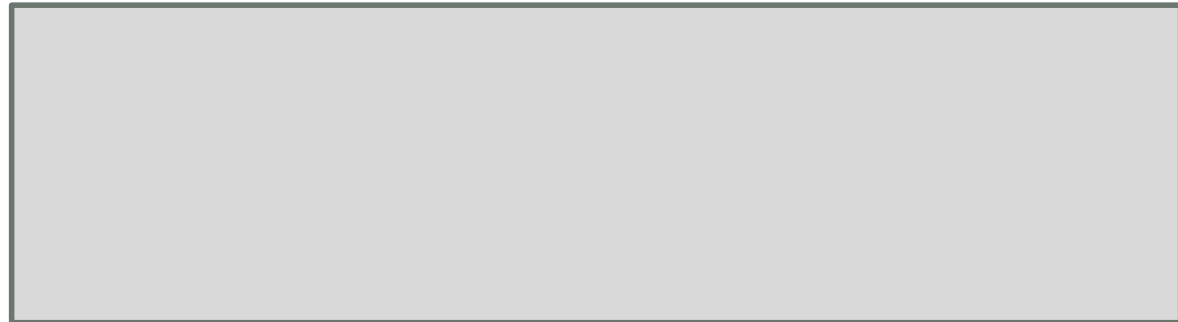
Стек



Буфер



Зависимости

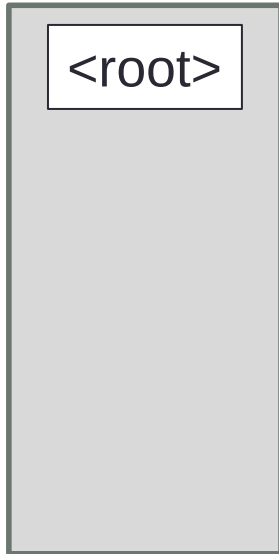


LeftArc_{subj}

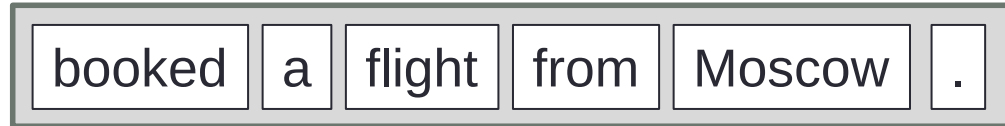


Arc-eager

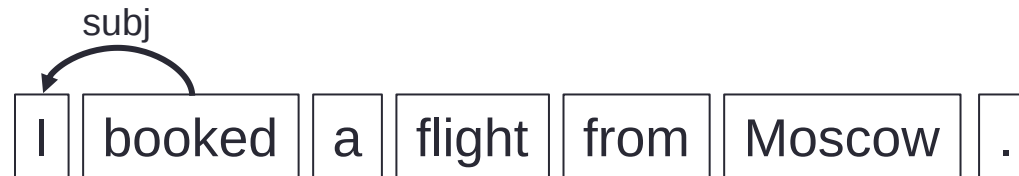
Стек



Буфер

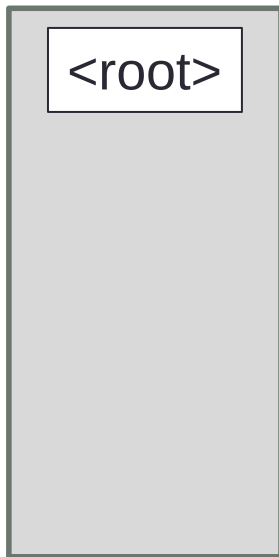


Зависимости

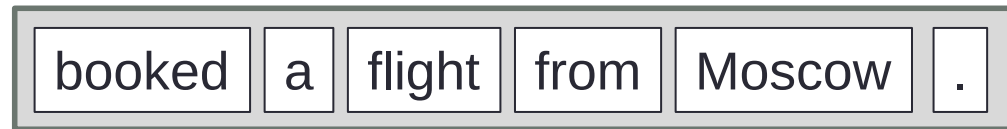


Arc-eager

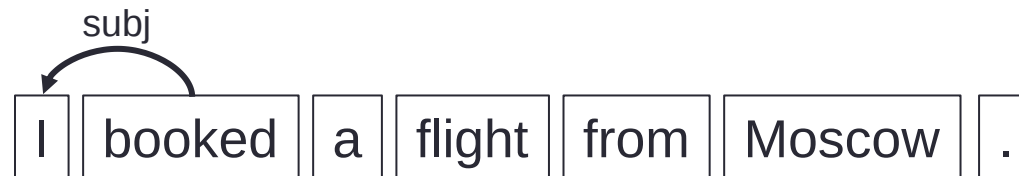
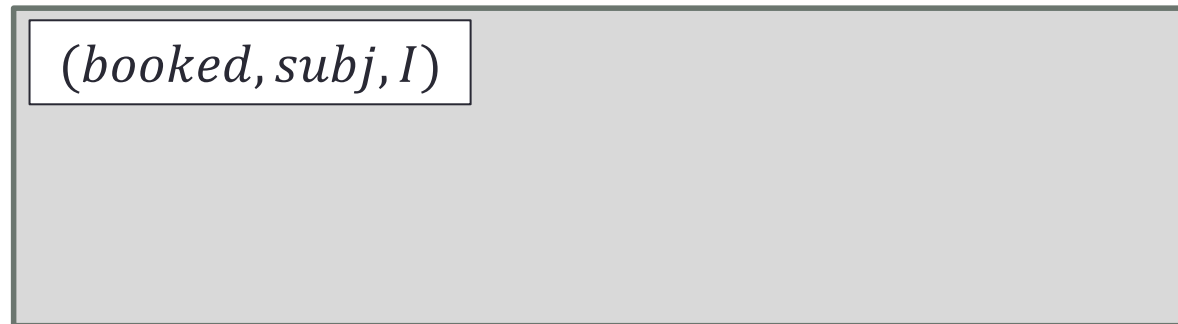
Стек



Буфер

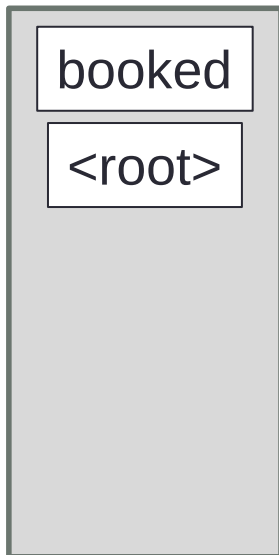


Зависимости



Arc-eager

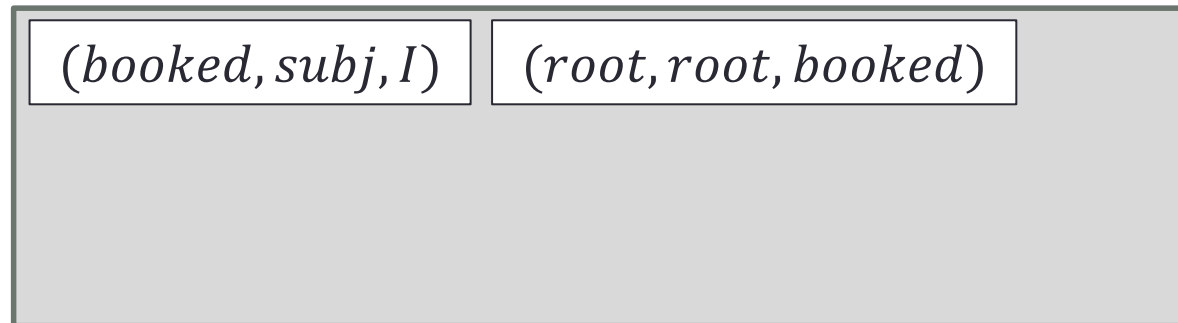
Стек



Буфер

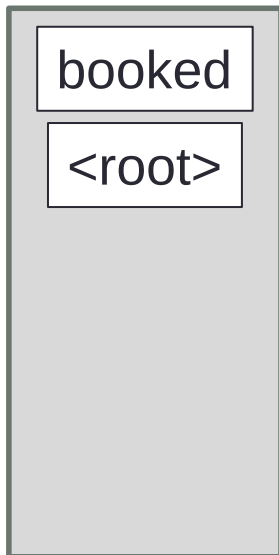


Зависимости



Arc-eager

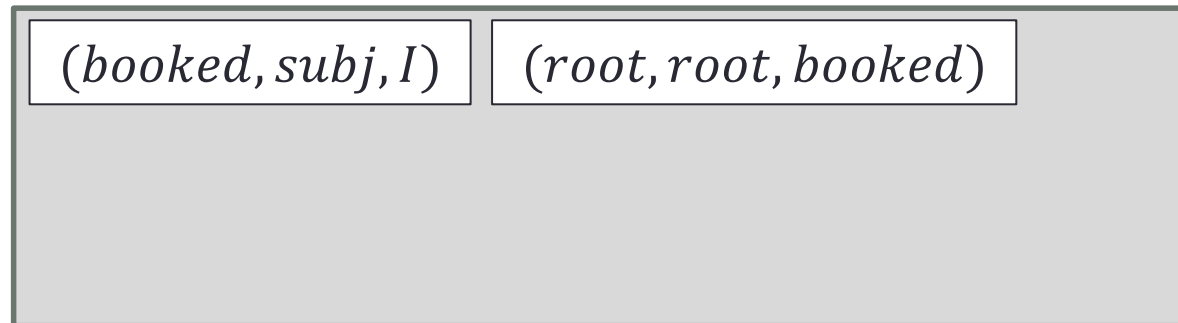
Стек



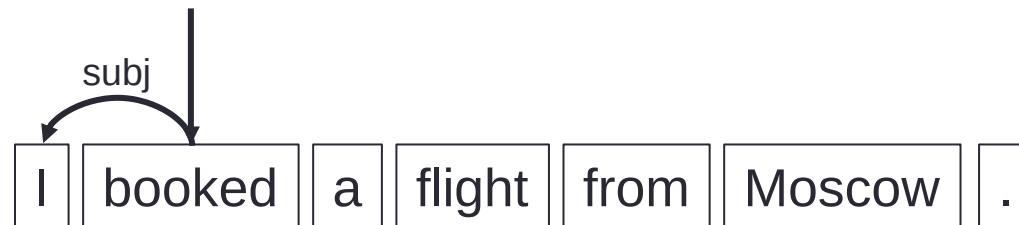
Буфер



Зависимости

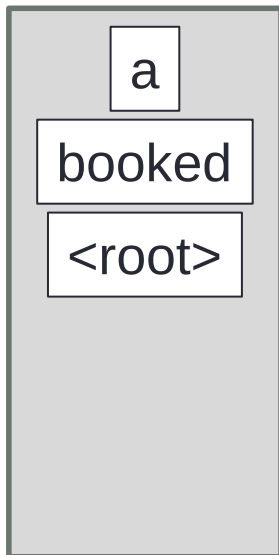


Shift



Arc-eager

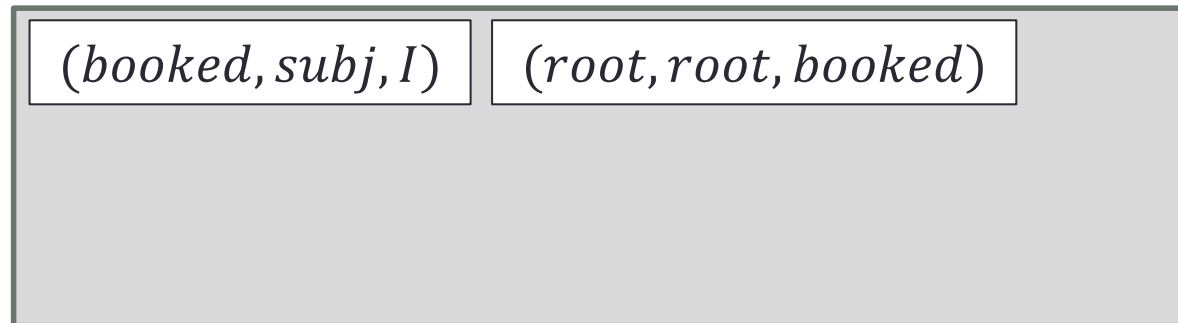
Стек



Буфер

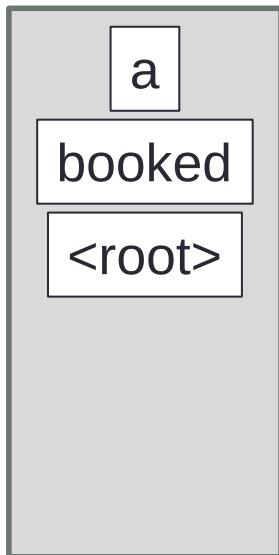


Зависимости



Arc-eager

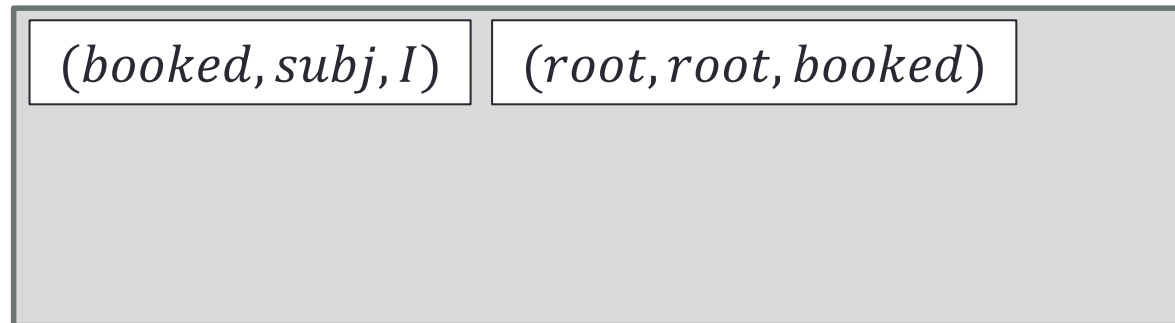
Стек



Буфер



Зависимости

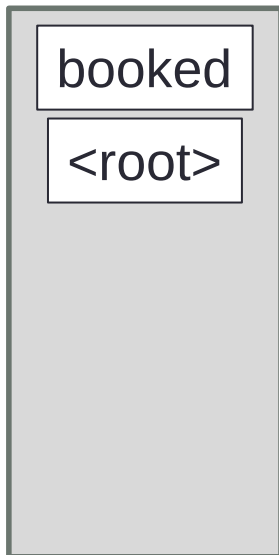


LeftArc_{det}



Arc-eager

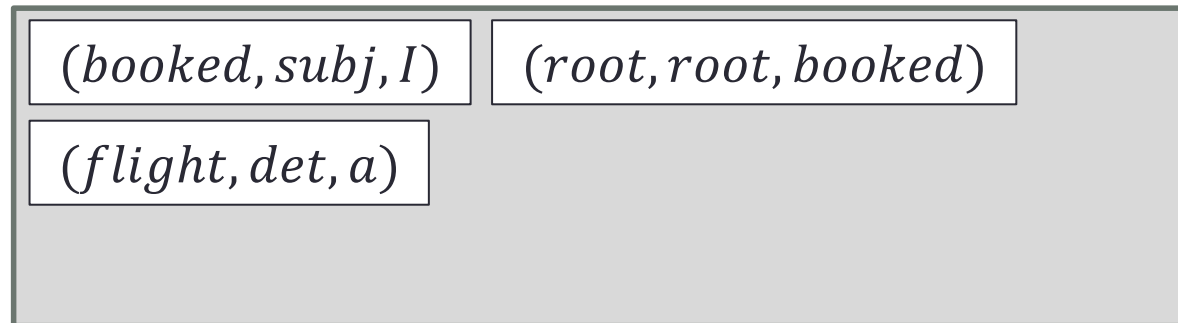
Стек



Буфер

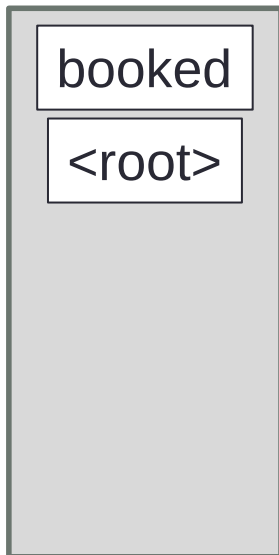


Зависимости



Arc-eager

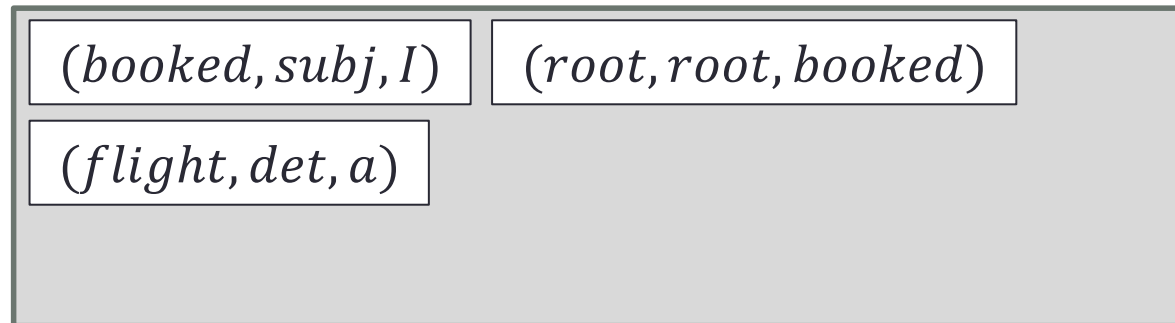
Стек



Буфер



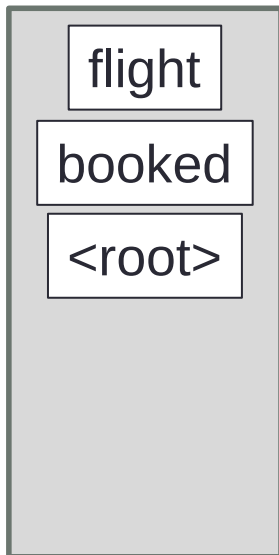
Зависимости



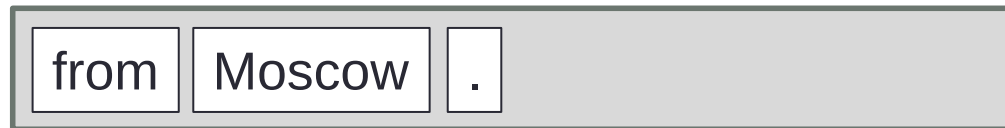
RightArc_{dobj}

Arc-eager

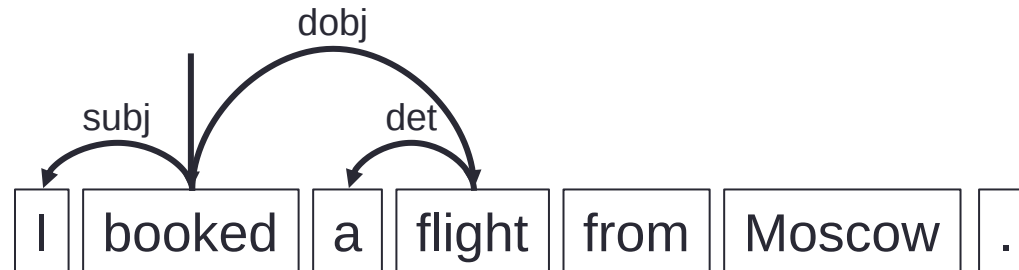
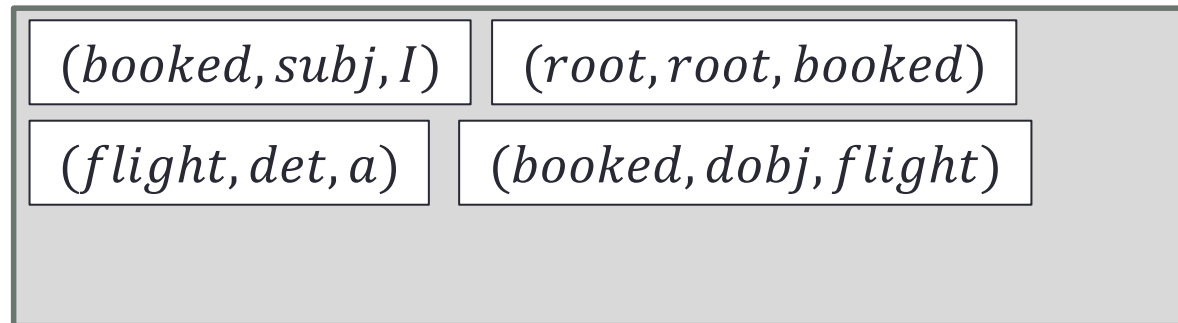
Стек



Буфер

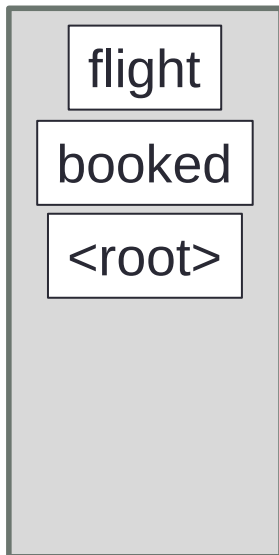


Зависимости

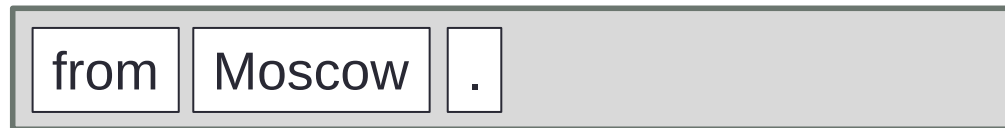


Arc-eager

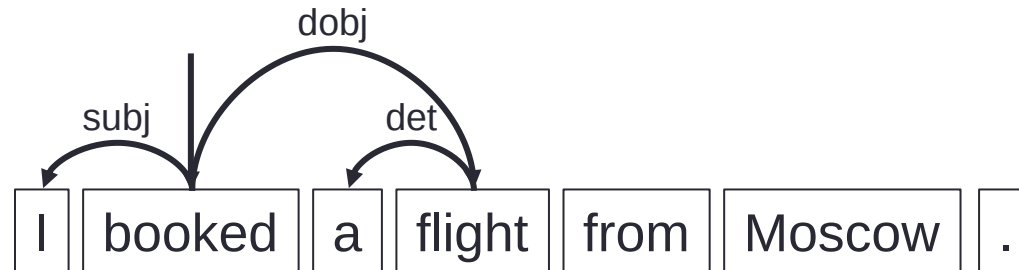
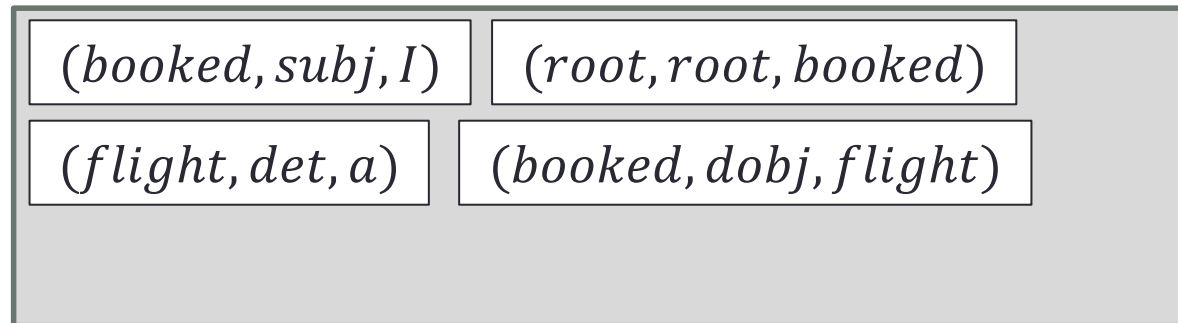
Стек



Буфер



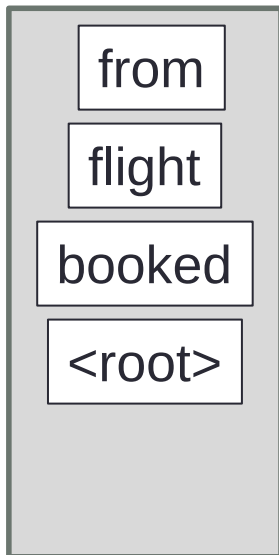
Зависимости



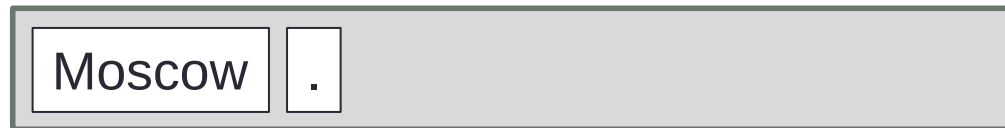
RightArc_{pmo}

Arc-eager

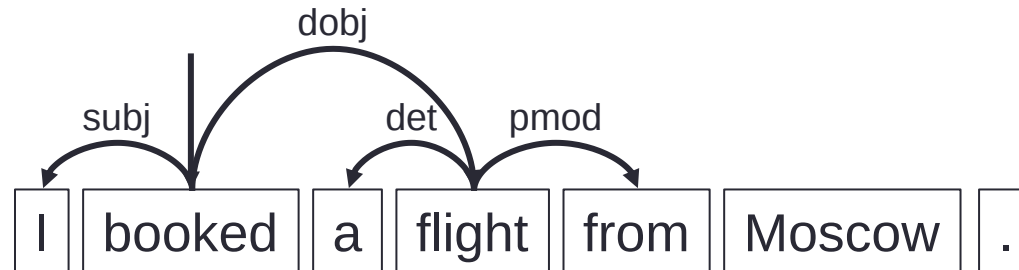
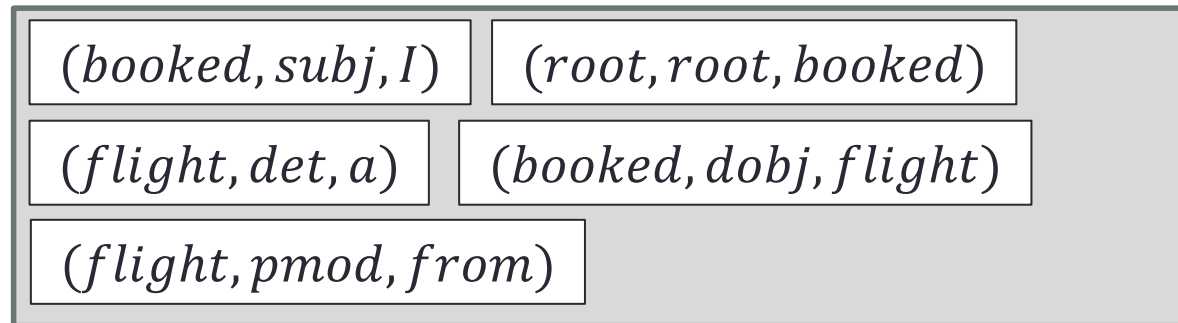
Стек



Буфер

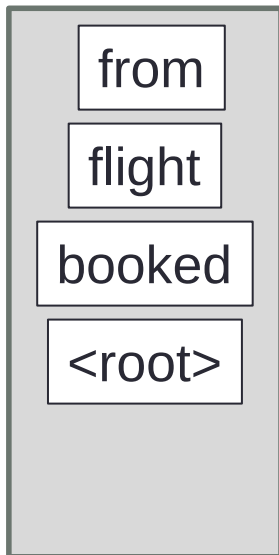


Зависимости

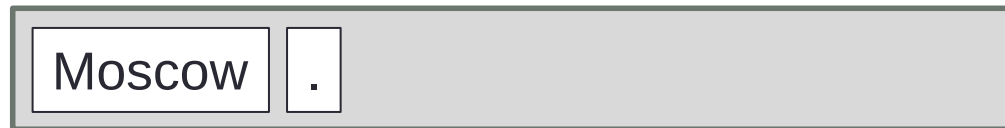


Arc-eager

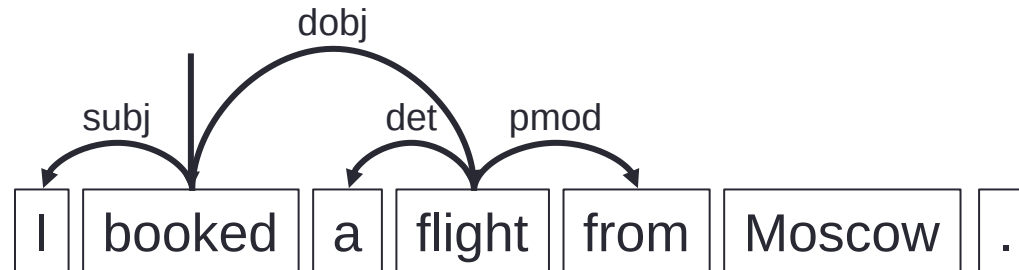
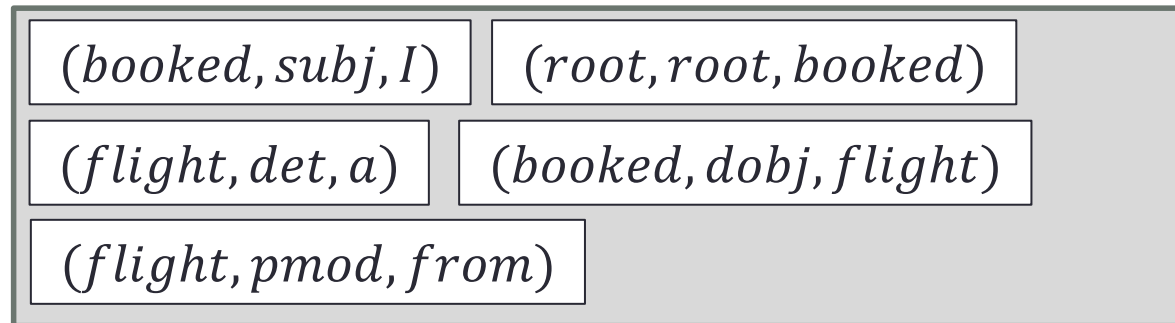
Стек



Буфер



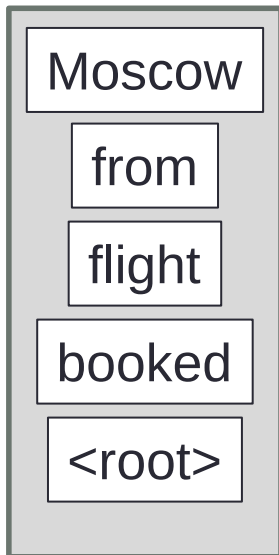
Зависимости



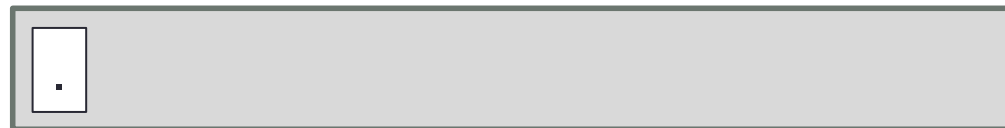
RightArc_{prep}

Arc-eager

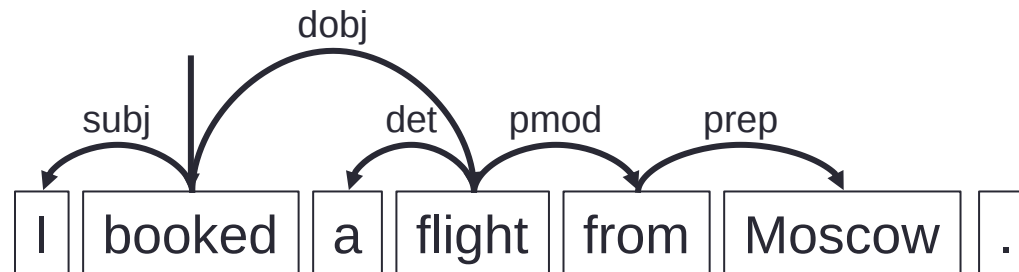
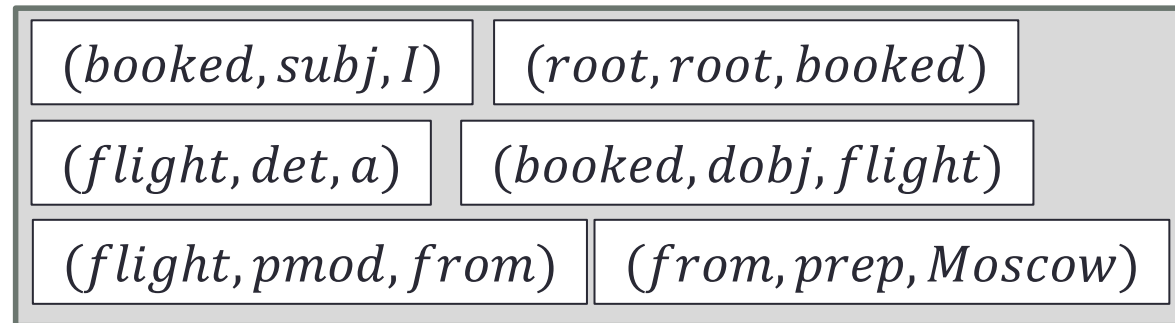
Стек



Буфер

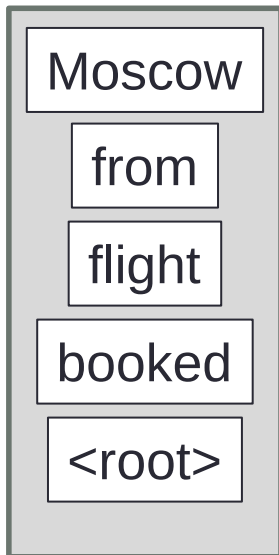


ЗАВИСИМОСТИ

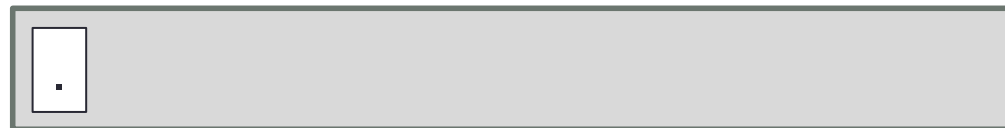


Arc-eager

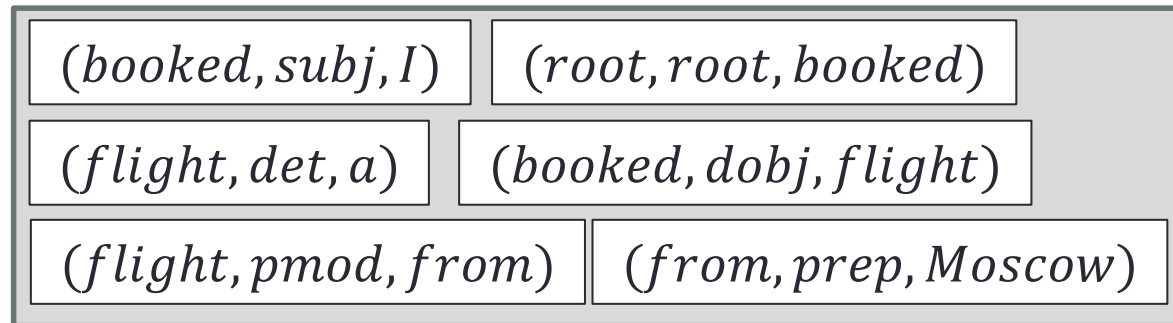
Стек



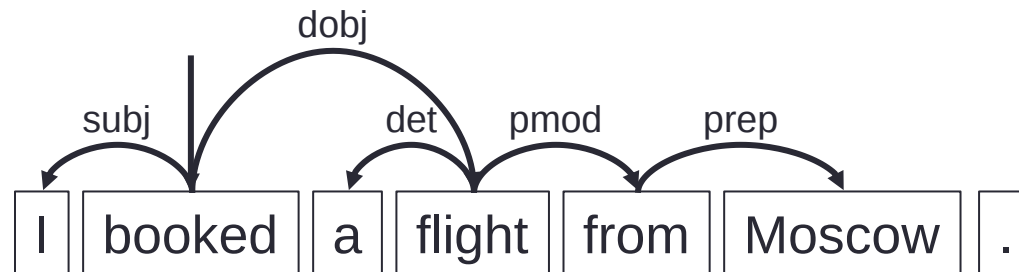
Буфер



ЗАВИСИМОСТИ

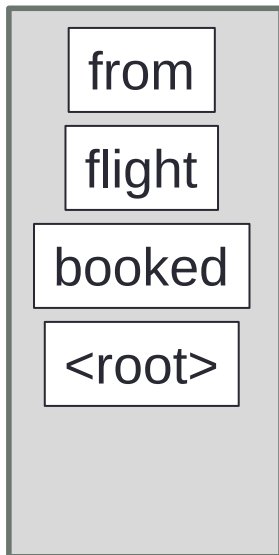


Reduce

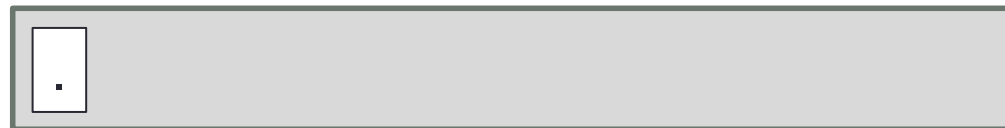


Arc-eager

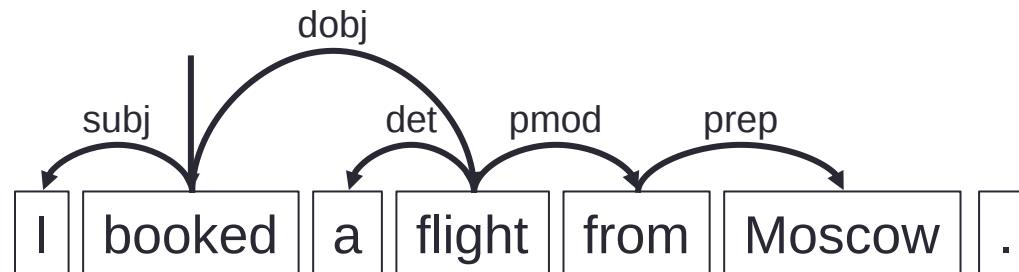
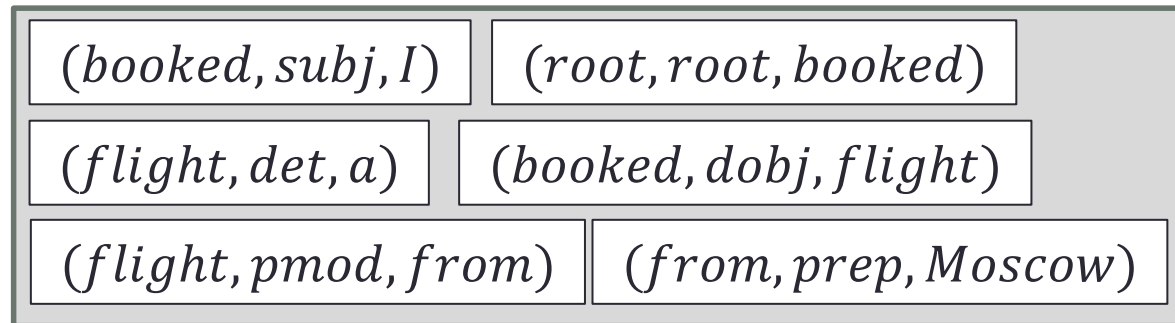
Стек



Буфер

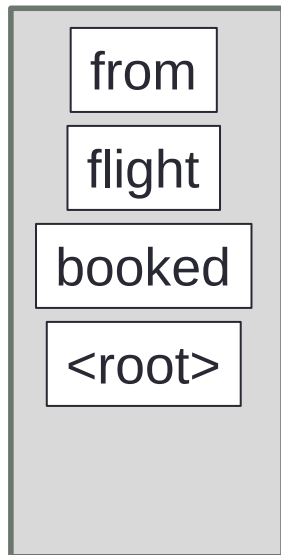


ЗАВИСИМОСТИ

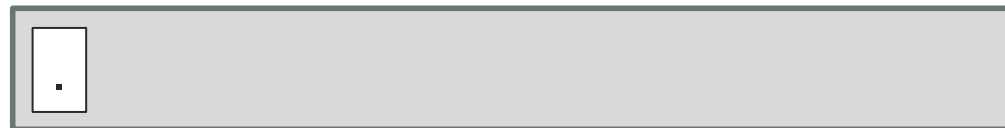


Arc-eager

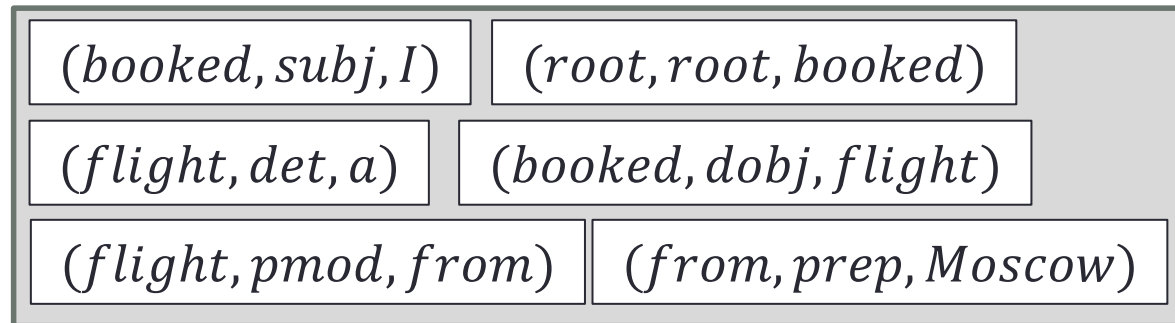
Стек



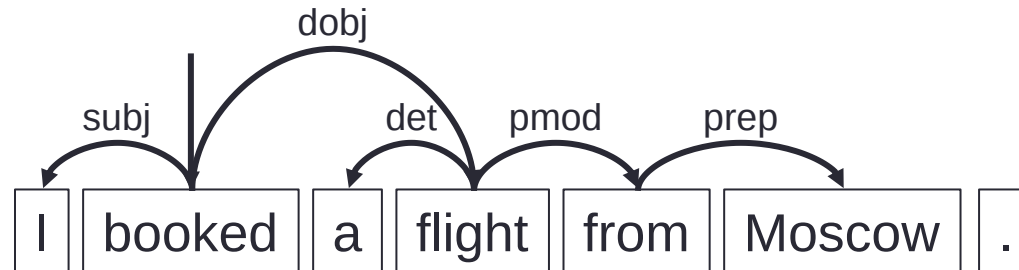
Буфер



ЗАВИСИМОСТИ

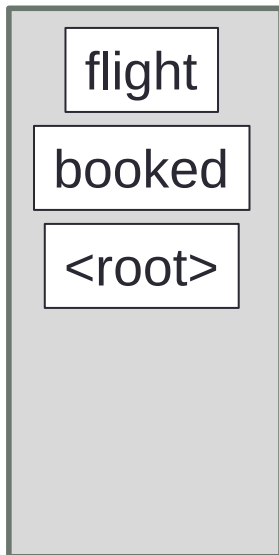


Reduce

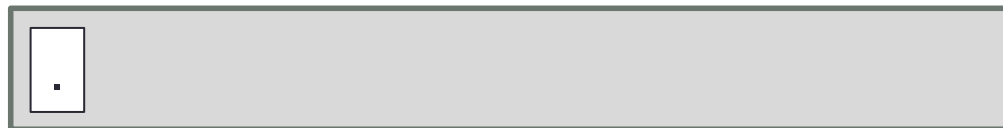


Arc-eager

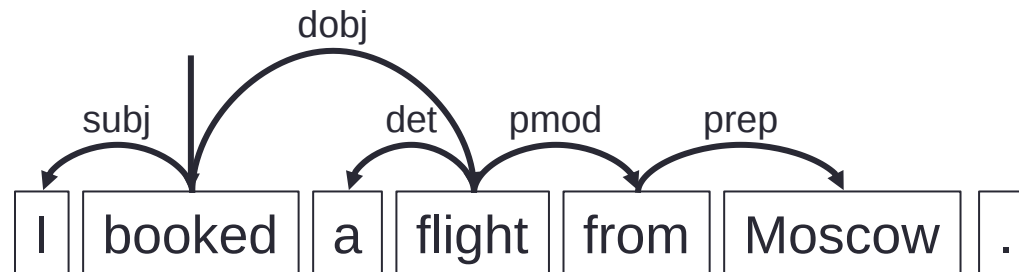
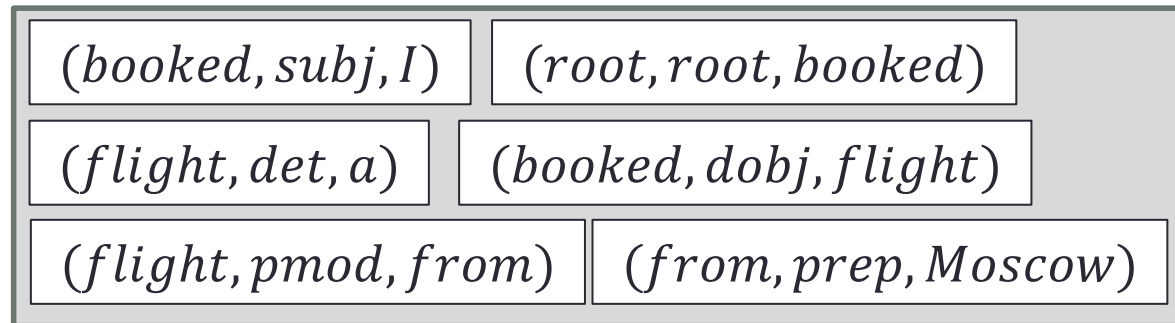
Стек



Буфер

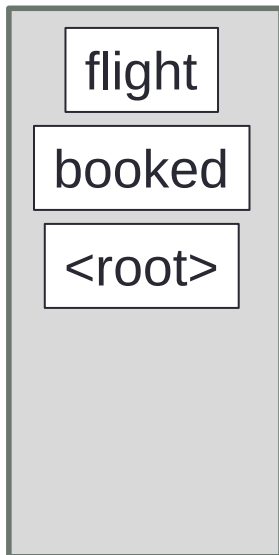


Зависимости



Arc-eager

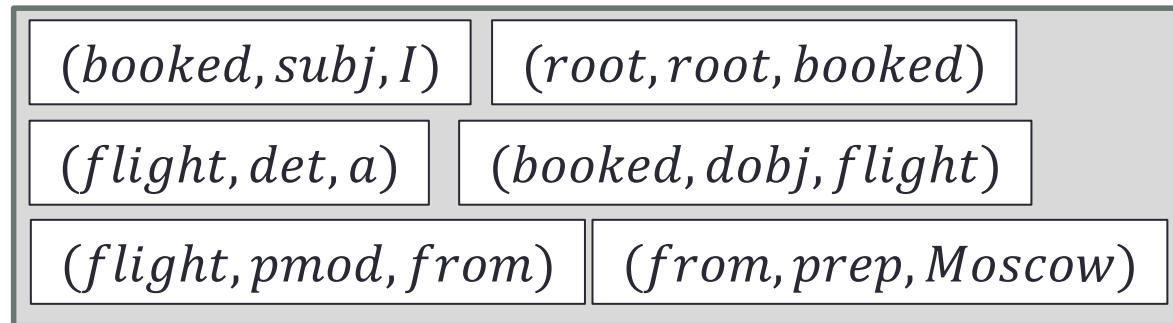
Стек



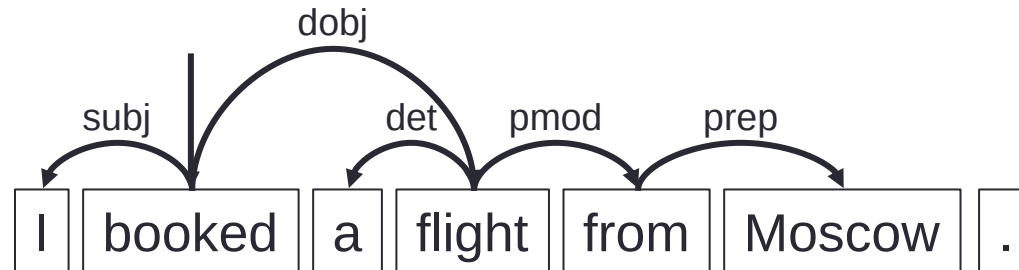
Буфер



ЗАВИСИМОСТИ



Reduce



Arc-eager

Стек

booked

<root>

Буфер

.

ЗАВИСИМОСТИ

(booked, subj, I)

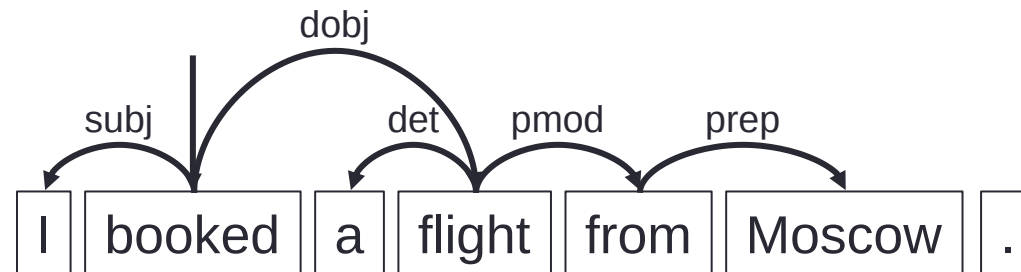
(root, root, booked)

(flight, det, a)

(booked, dobj, flight)

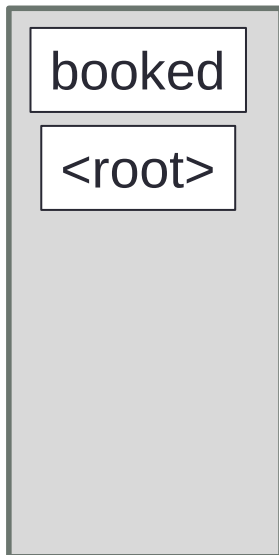
(flight, pmod, from)

(from, prep, Moscow)



Arc-eager

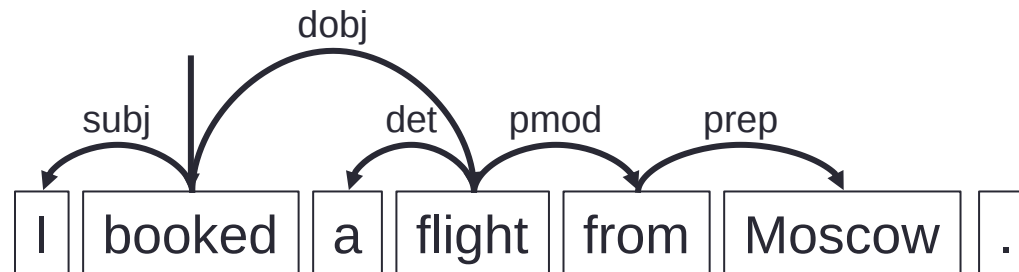
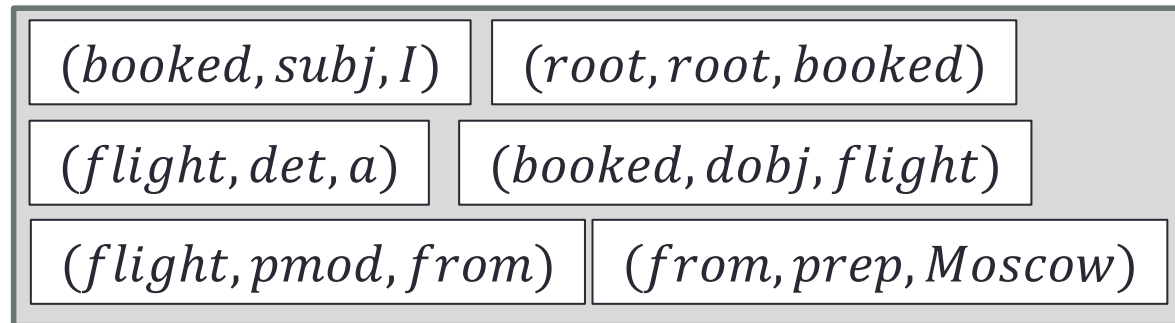
Стек



Буфер



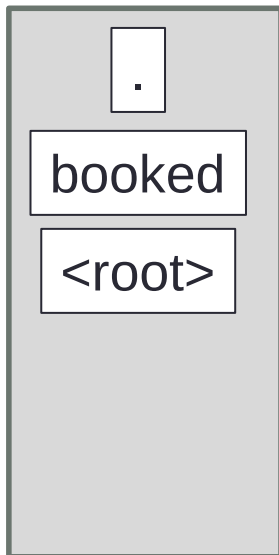
Зависимости



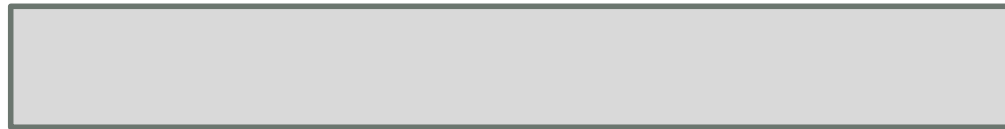
RightArc_{pu}n

Arc-eager

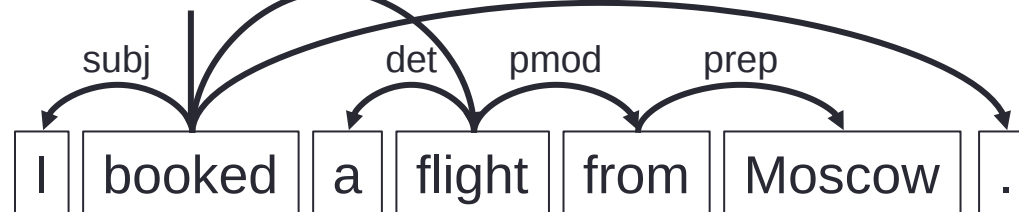
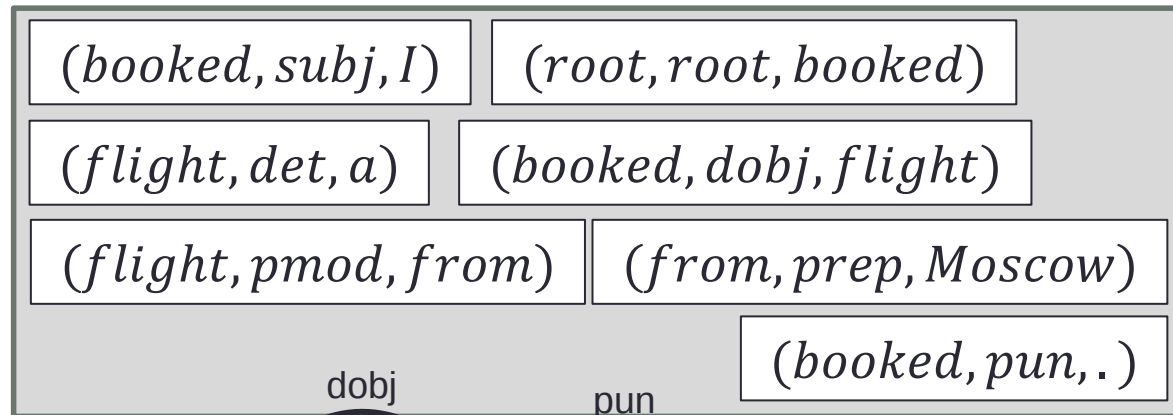
Стек



Буфер

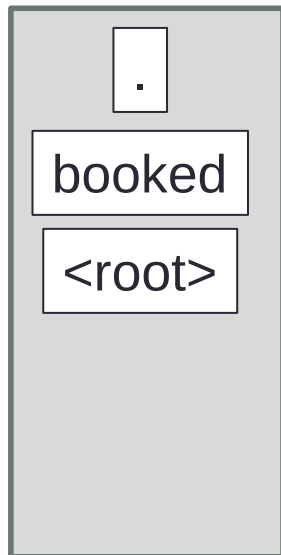


ЗАВИСИМОСТИ

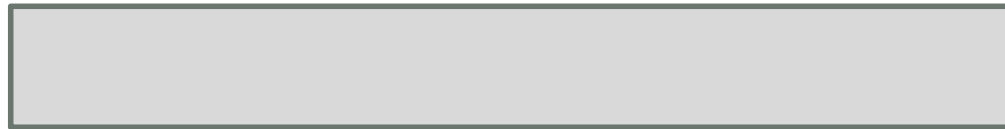


Arc-eager

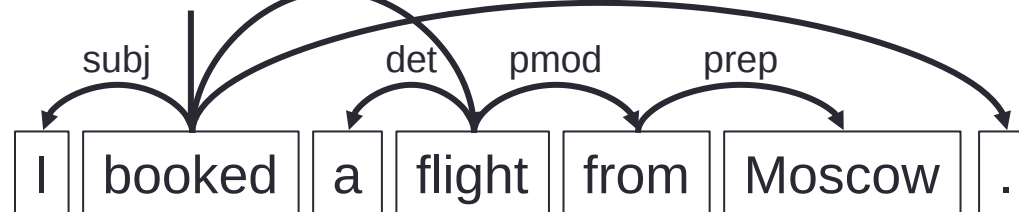
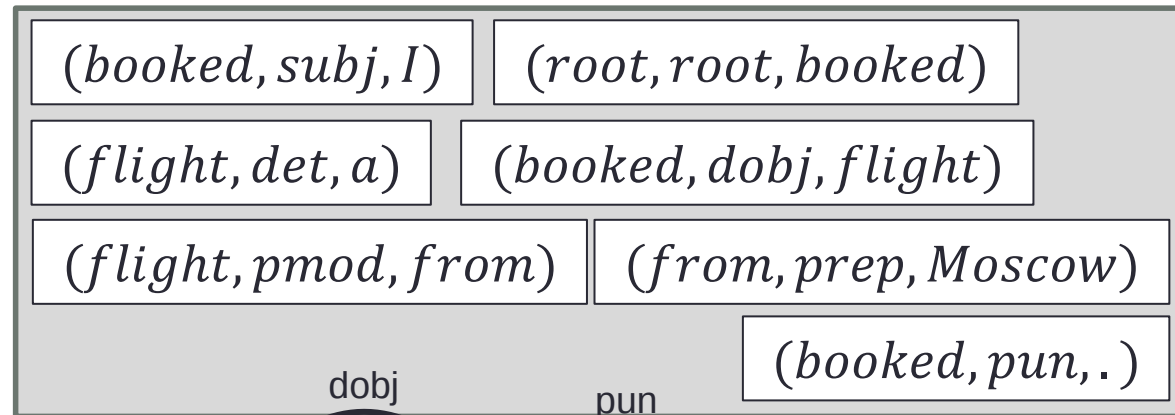
Стек



Буфер



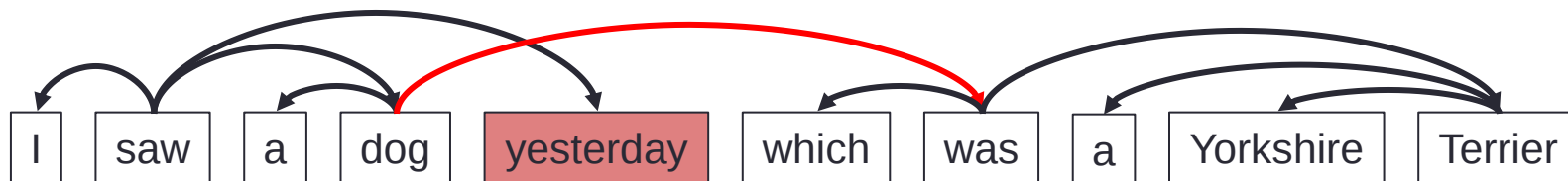
ЗАВИСИМОСТИ



DONE!!!

Проективное синтаксическое дерево

- Дерево зависимостей является проективным, если
 - Для каждой дуги в дереве существует направленный путь от главного слова до каждого из слов, расположенных между главным и зависимым словом этой дуги
 - $(i, l, j) \Rightarrow \forall k \in [\min(i, j), \max(i, j)], i \rightarrow^* k$



Non-projective Transition-based Parsing.

Attardi's system

- Инициализация
 - $c_s(x = x_1 \dots x_n) = ([0], [1, \dots, n], \emptyset)$
- Конечное состояние
 - $C_t = \{c \in C \mid c = ([0], [], A)\}$
- Переходы (для левых зависимостей требуется $i \neq 0$)
 - (*Shift*) $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$
 - (*LeftArc_l*) $([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$
 - (*RightArc_l*) $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$
 - (*LeftArc2_l*) $([\sigma|i|k|j], B, A) \rightarrow ([\sigma|k|j], B, A \cup \{(j, l, i)\})$
 - (*RightArc2_l*) $([\sigma|i|k|j], B, A) \rightarrow ([\sigma|i|k], B, A \cup \{(i, l, j)\})$
 - (*LeftArc3_l*) $([\sigma|i|k_1|k_2|j], B, A) \rightarrow ([\sigma|k_1|k_2|j], B, A \cup \{(j, l, i)\})$
 - (*RightArc3_l*) $([\sigma|i|k_1|k_2|j], B, A) \rightarrow ([\sigma|i|k_1|k_2], B, A \cup \{(i, l, j)\})$

Non-projective Transition-based Parsing.

Online reordering

- Инициализация

- $c_s(x = x_1 \dots x_n) = ([0], [1, \dots, n], \emptyset)$

- Конечное состояние

- $C_t = \{c \in C \mid c = ([0], [], A)\}$

- Переходы

- (*Shift*) $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

- (*LeftArc_l*) $([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$, если $i \neq 0$

- (*RightArc_l*) $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

- (*Swap*) $([\sigma|i|j], \beta, A) \rightarrow ([\sigma|j], [i|\beta], A)$, если $i \neq 0$ и $i < j$

Transition-based Parsing.

Предсказание переходов

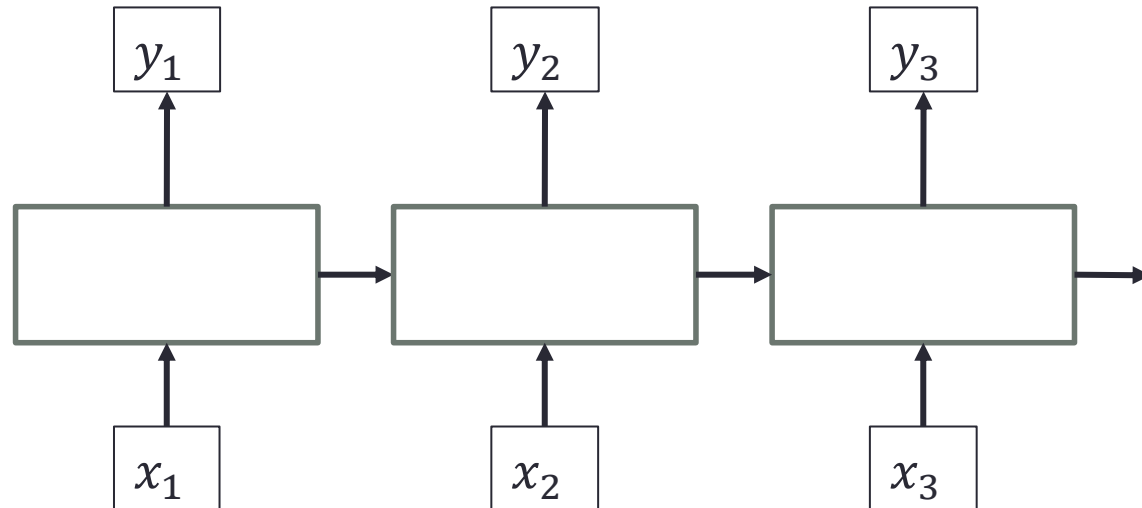
- Задача многоклассовой классификации
 - Объекты классификации: конфигурации парсера
 - Классы: правила перехода
- Методы:
 - Multiclass SVM
 - Naïve Bayes
 - Decision trees
 - Нейронные сети
 - ...

Признаки классификации

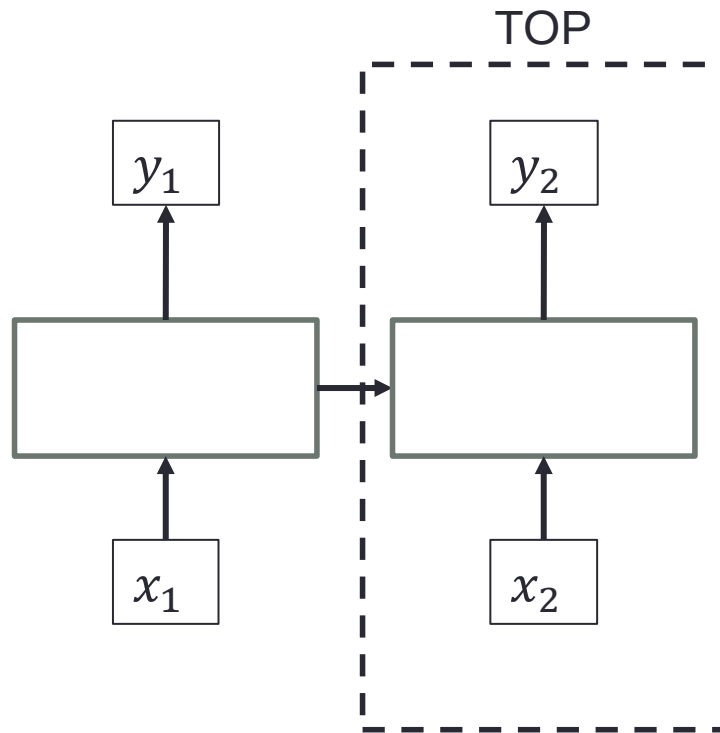
- Признаки для элемента
 - Словоформа
 - Лемма
 - Часть речи
 - Граммемы
 - Тип отношения
- Элементы:
 - i -й элемент стека
 - i -й элемент буфера
 - Левое отношение элемента стека
 - Правое отношение элемента стека

Long Short-Term Memory

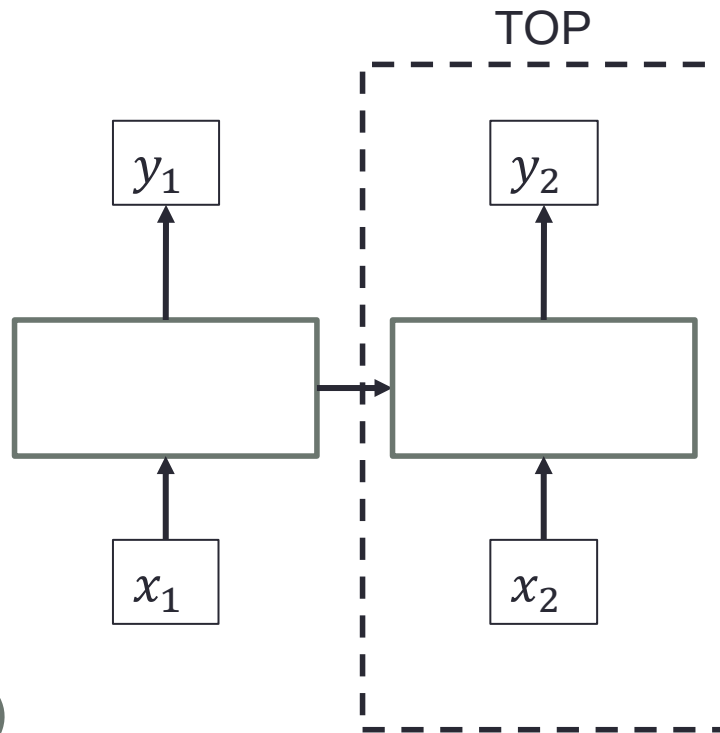
- $i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i)$
- $f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f)$
- $c_t = f_t \odot c_{t-1} i_t + \tanh \odot (W_{cx}x_t + W_{ch}h_{t-1} + b_c)$
- $o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o)$
- $h_t = o_t \odot \tanh(c_t)$
- $y_t = g(h_t)$



Stack Long Short-Term Memory

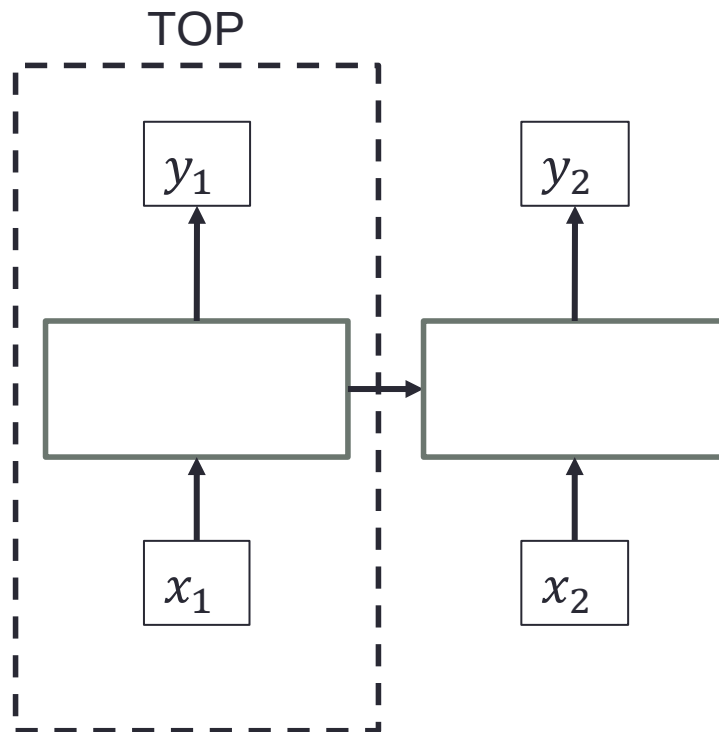


Stack Long Short-Term Memory

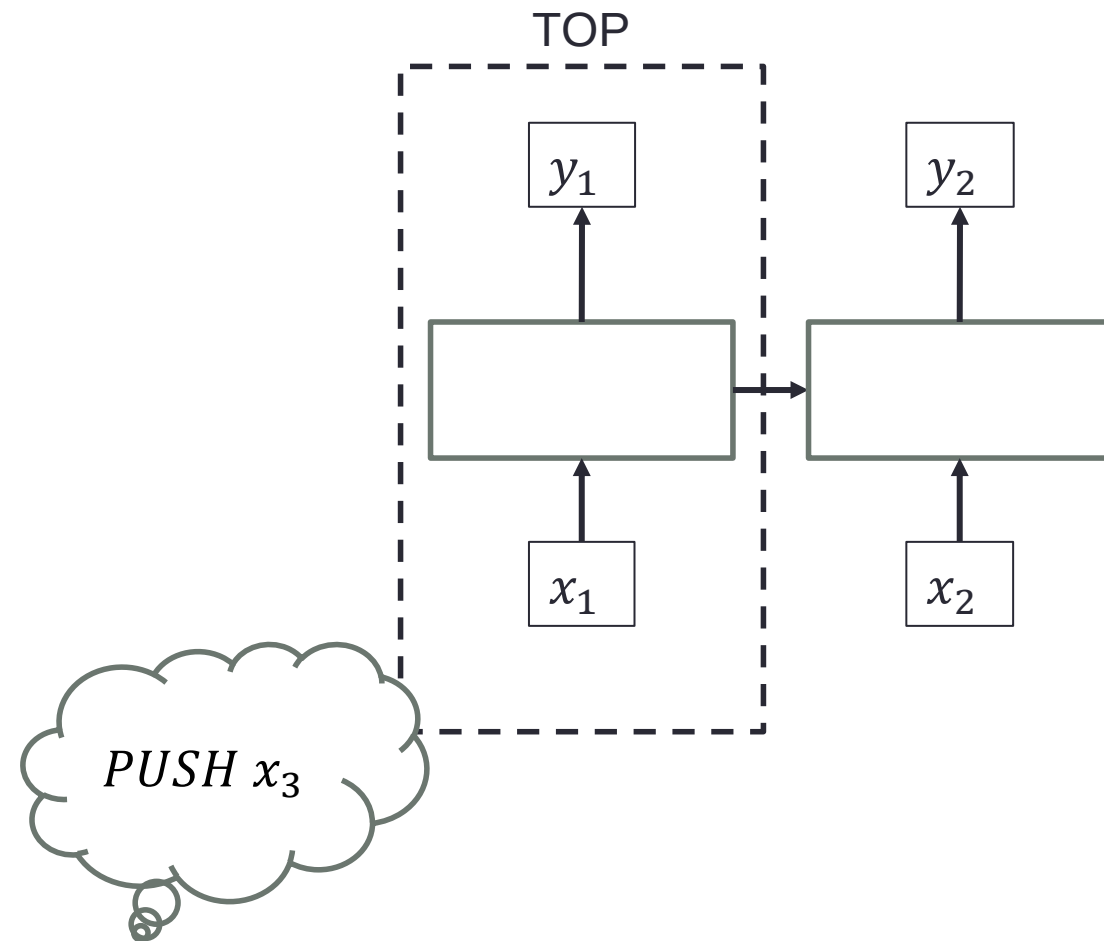


POP

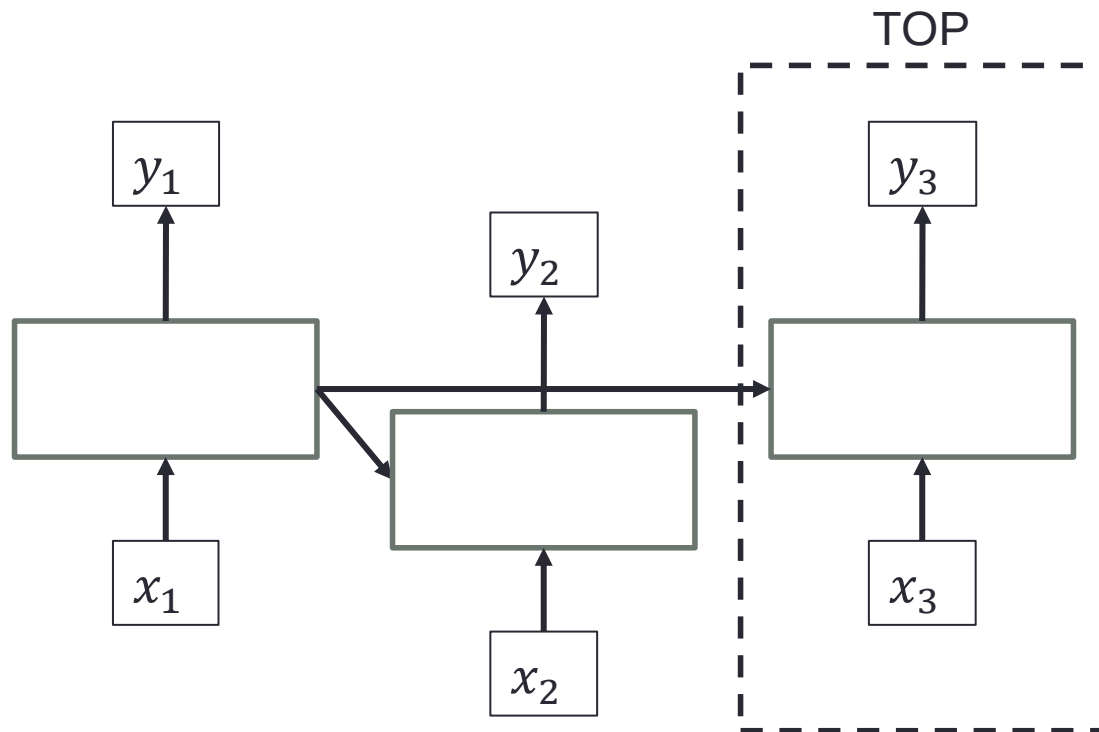
Stack Long Short-Term Memory



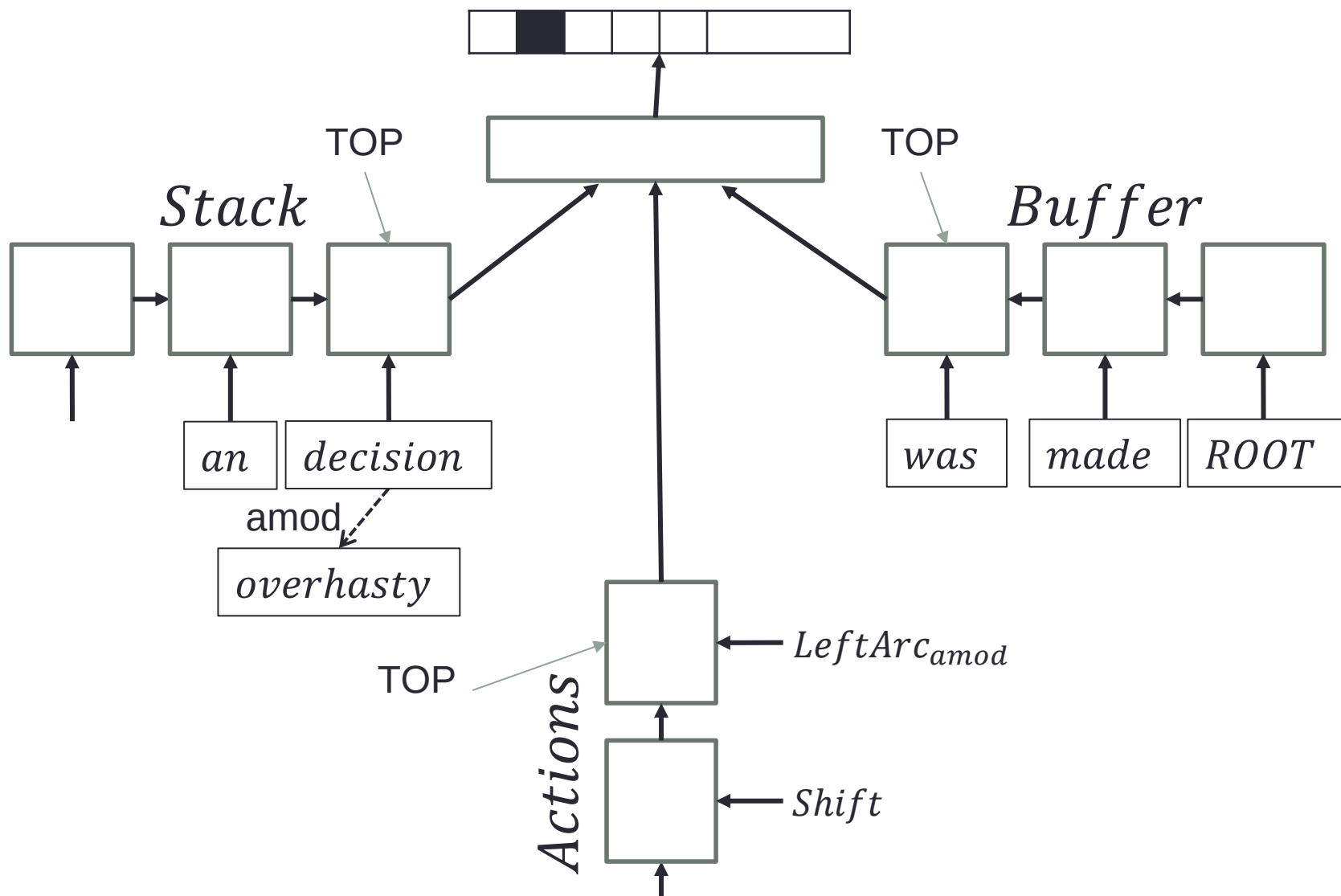
Stack Long Short-Term Memory



Stack Long Short-Term Memory



Stack LSTM Parser



Stack LSTM Parser

- Переходы Arc-standard

- *Shift*

$$(\sigma, [(v_i, i) | \beta], A) \rightarrow ([\sigma | (v_i, i)], \beta, A)$$

- *LeftArc_r*

$$([\sigma | (v_i, i) | (v_j, j)], B, A) \rightarrow ([\sigma | (g_r(v_i, v_j), j)], B, A \cup \{j \xrightarrow{r} i\})$$

- *RightArc_r*

$$([\sigma | (v_i, i) | (v_j, j)], B, A) \rightarrow ([\sigma | (g_r(v_j, v_i), i)], B, A \cup \{i \xrightarrow{r} j\})$$

Оценка качества

- Labeled attachment score (LAS):
 - Отношение количества правильно построенных дуг к общему количеству дуг
- Labeled exact match (LEM):
 - Отношение количества правильно построенных деревьев к общему количеству деревьев
- Unlabeled attachment score/exact match (UAS/UEM):
 - Те же метрики, но дуги рассматриваются без типов

Оценка качества

- Micro-averaged
 - Считаем отношение количества правильных дуг во всем корпусе к количеству дуг в этом корпусе
- Macro-averaged
 - Считаем метрику для каждого предложения и потом усредняем

Следующая лекция

- Машинный перевод
- Лектор: Сысоев Андрей Анатольевич