

Синтаксический анализ

Лекция 7

Майоров Владимир Дмитриевич

24 октября 2023

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

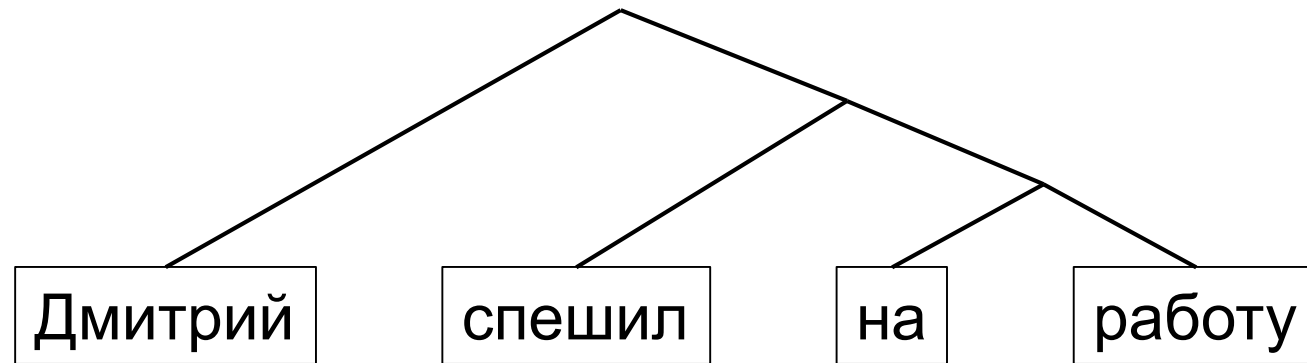
- Глагол сочетается с наречием

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

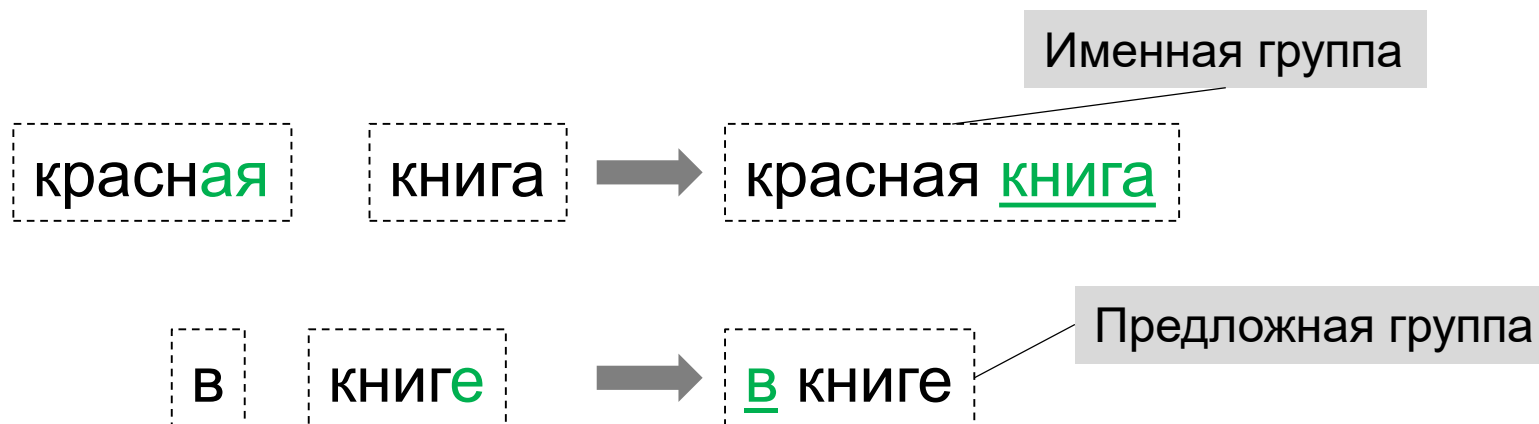
в книга → не словосочетание

в книге → в книге

Синтаксические правила

В зависимости от главного слова в словосочетании выделяют

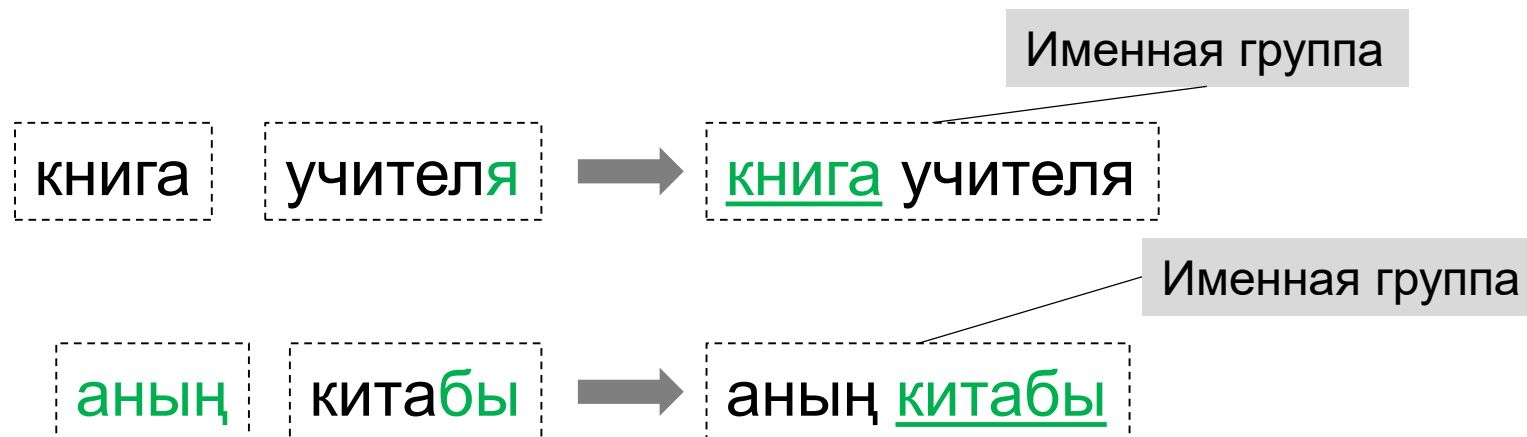
- Именные группы (главное – существительное)
- Группа прилагательного
- Наречная группа
- Предложная группа
- Глагольная группа
- Предложение



Синтаксические правила

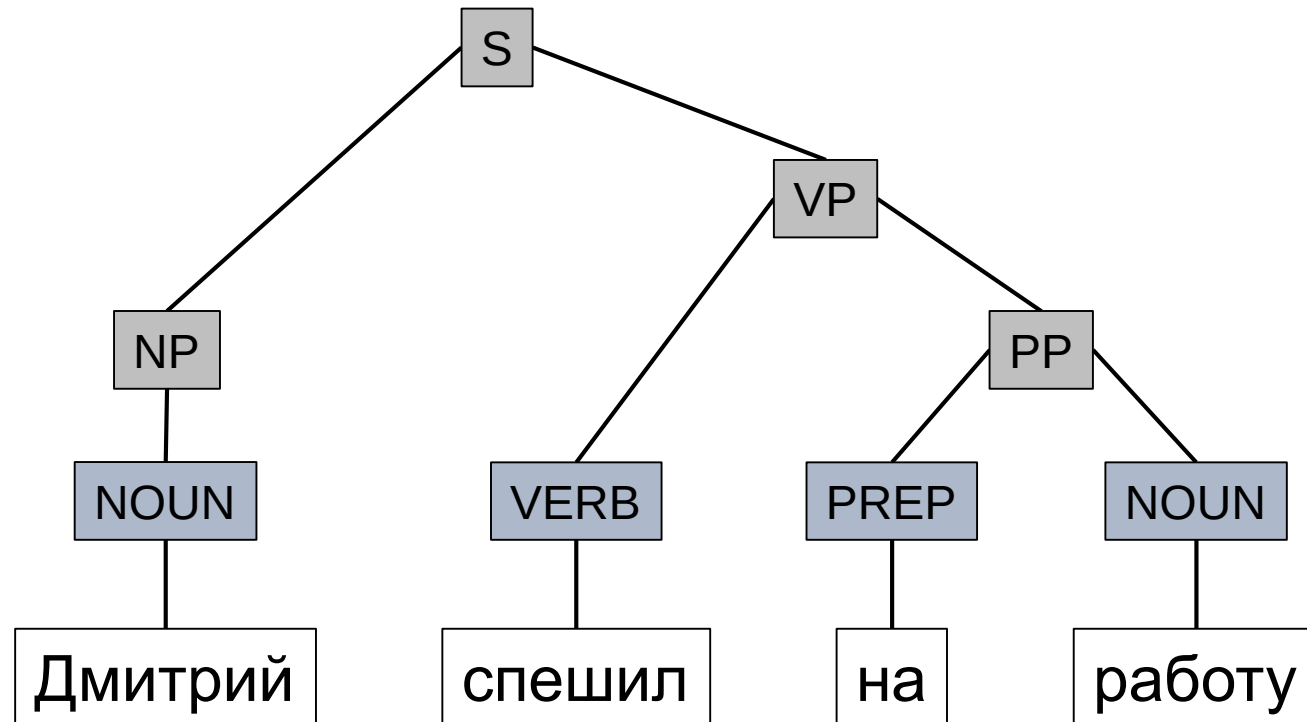
В зависимости от главного слова в словосочетании выделяют

- 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

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

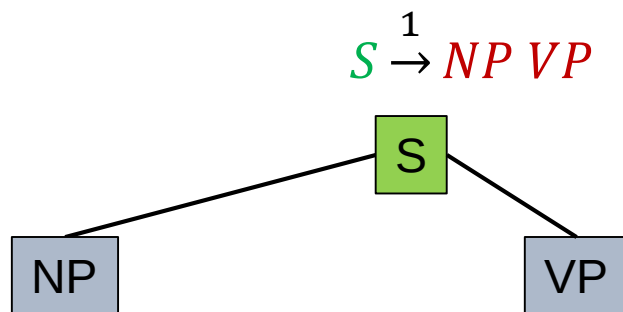
$N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$

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. *Det* → *the*
15. *Prep* → *from*

Формальная грамматика

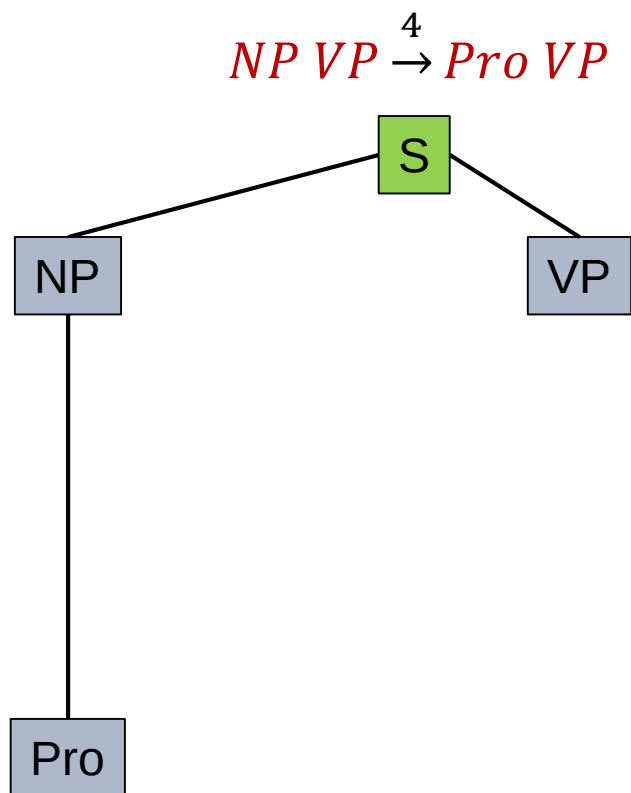
 $\Sigma = \{booked, flight, Moscow, I, a, from\}$ $N = \{S, VP, NP, PP, Verb, Noun, Pro, Det, Prep\}$ 

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. $Det \rightarrow the$
15. $Prep \rightarrow from$

Формальная грамматика

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

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

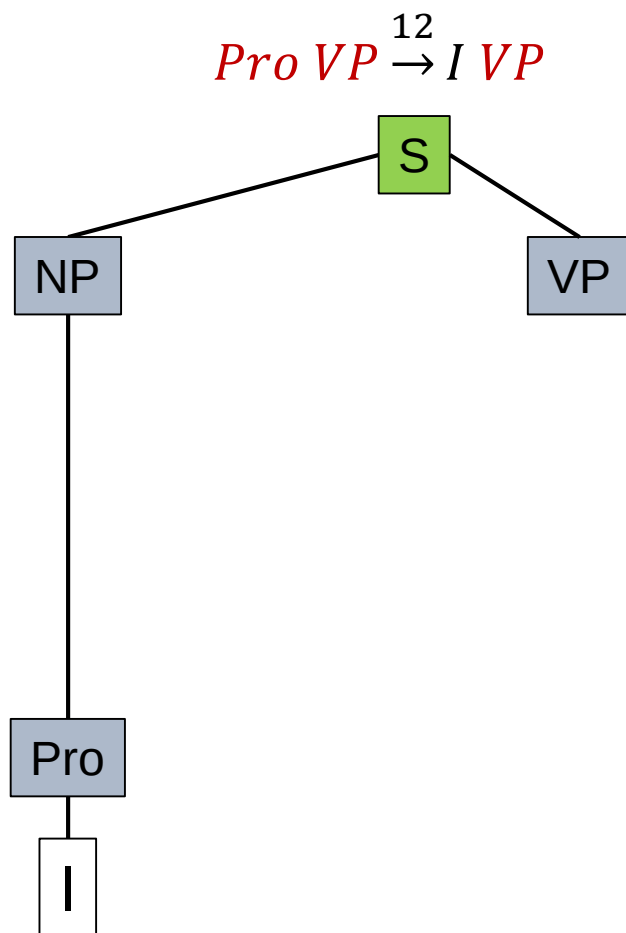


1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$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

Формальная грамматика

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

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

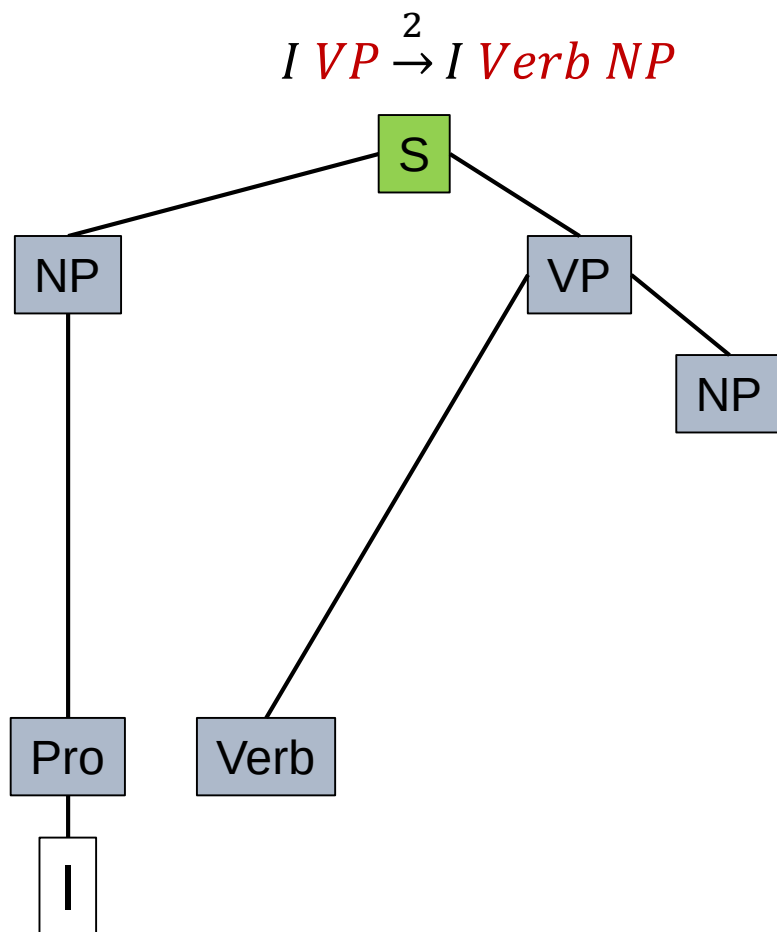


1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$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

Формальная грамматика

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

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



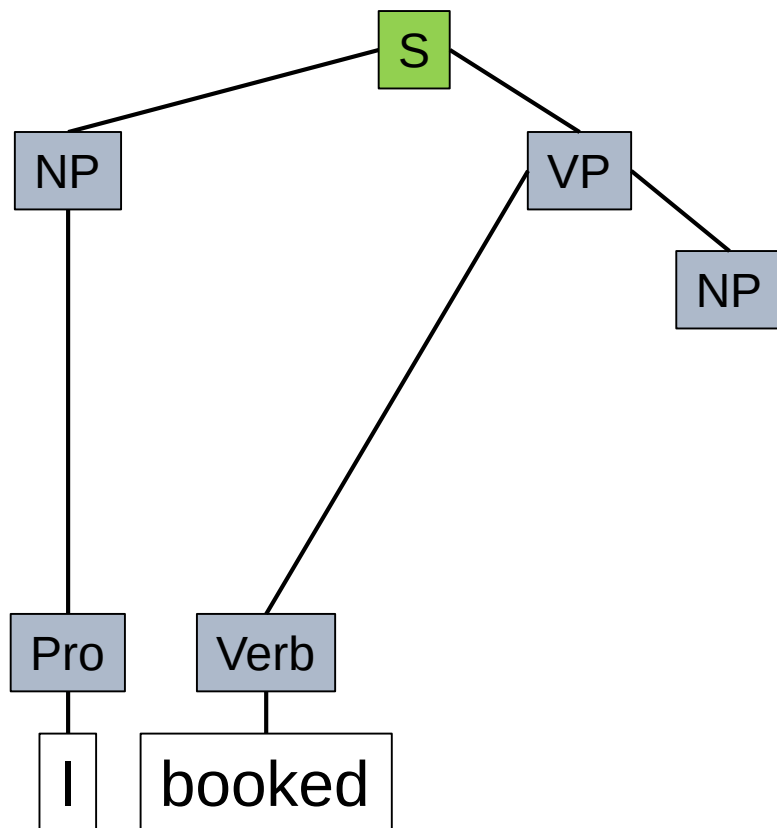
1. $S \rightarrow NP \ VP$
2. $VP \rightarrow Verb \ NP$
3. $VP \rightarrow Verb \ NP \ PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det \ NP$
6. $NP \rightarrow Noun \ PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep \ Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun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

Формальная грамматика

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

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

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



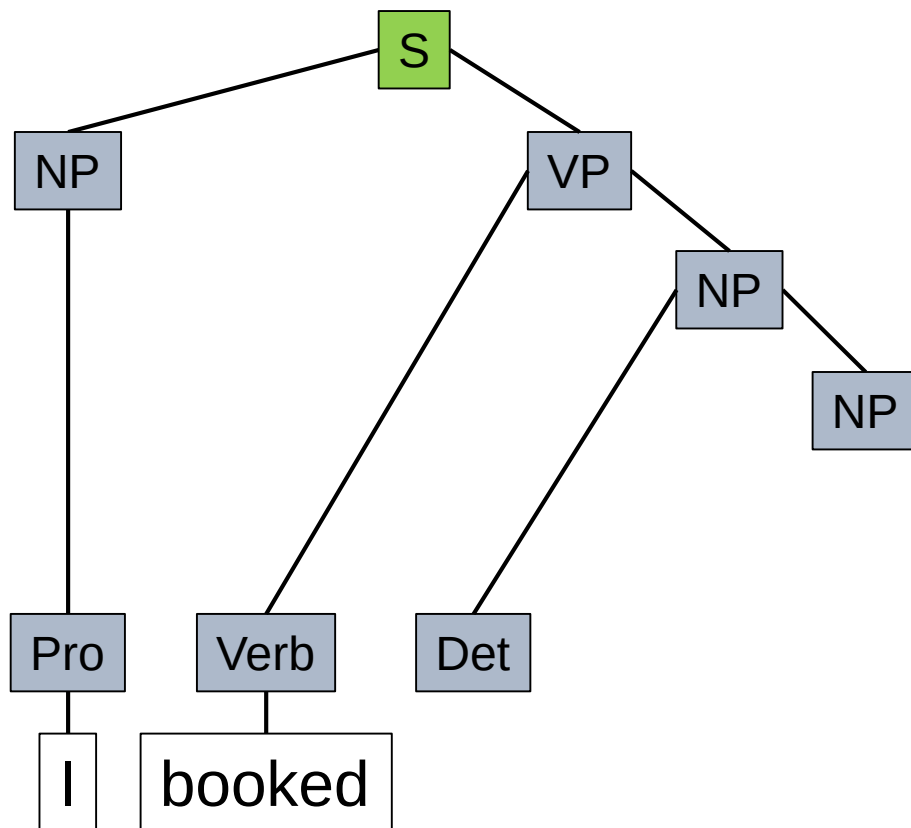
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $\underline{Verb} \rightarrow \underline{booked}$
10. $Noun \rightarrow flight$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

Формальная грамматика

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

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

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



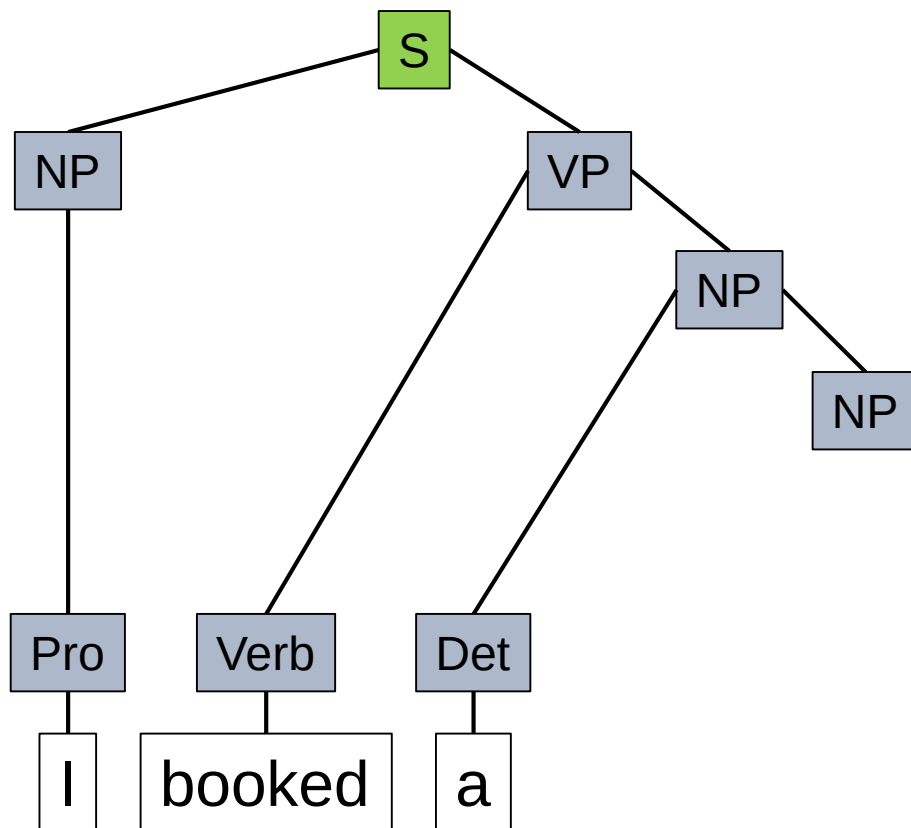
1. $S \rightarrow NP\ VP$
2. $VP \rightarrow Verb\ NP$
3. $VP \rightarrow Verb\ NP\ PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det\ NP$
6. $NP \rightarrow Noun\ PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep\ Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow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

Формальная грамматика

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

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

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



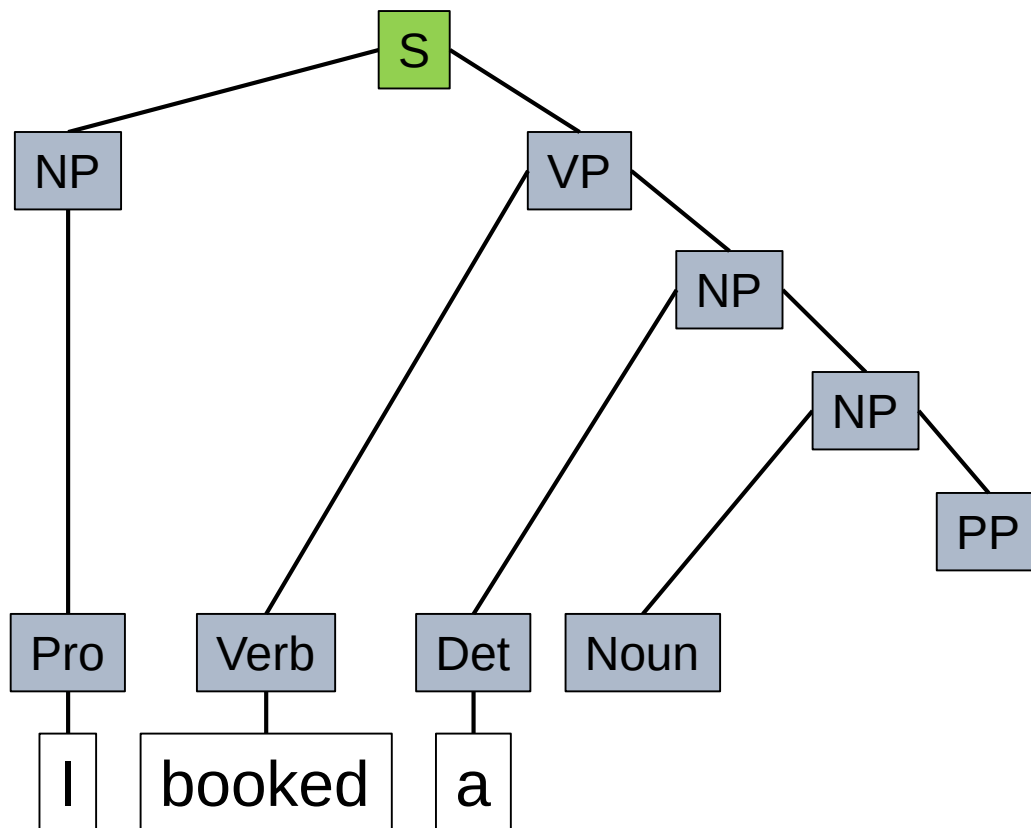
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$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

Формальная грамматика

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

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

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



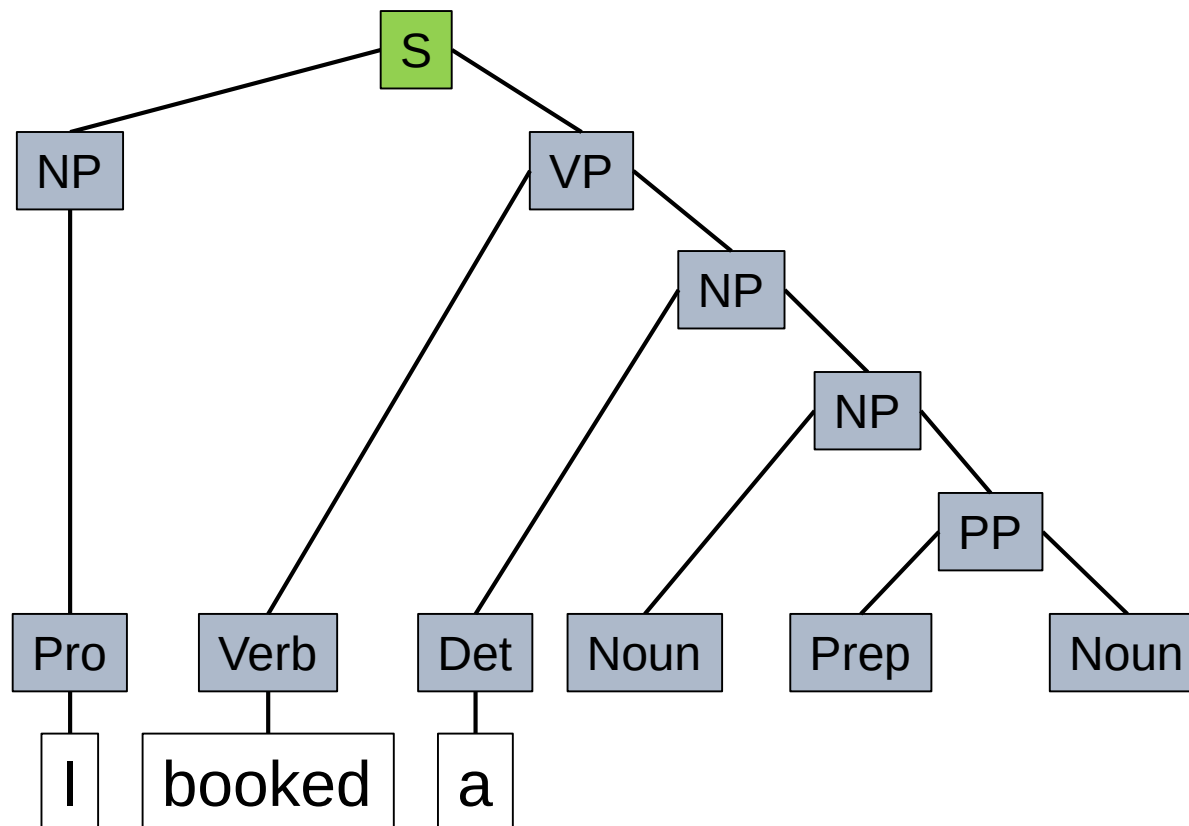
1. $S \rightarrow NP \ VP$
2. $VP \rightarrow Verb \ NP$
3. $VP \rightarrow Verb \ NP \ PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det \ NP$
6. $NP \rightarrow Noun \ PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep \ Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun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

Формальная грамматика

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

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

I booked a Noun PP ⁸ $\rightarrow$ *I booked a Noun Prep Noun*



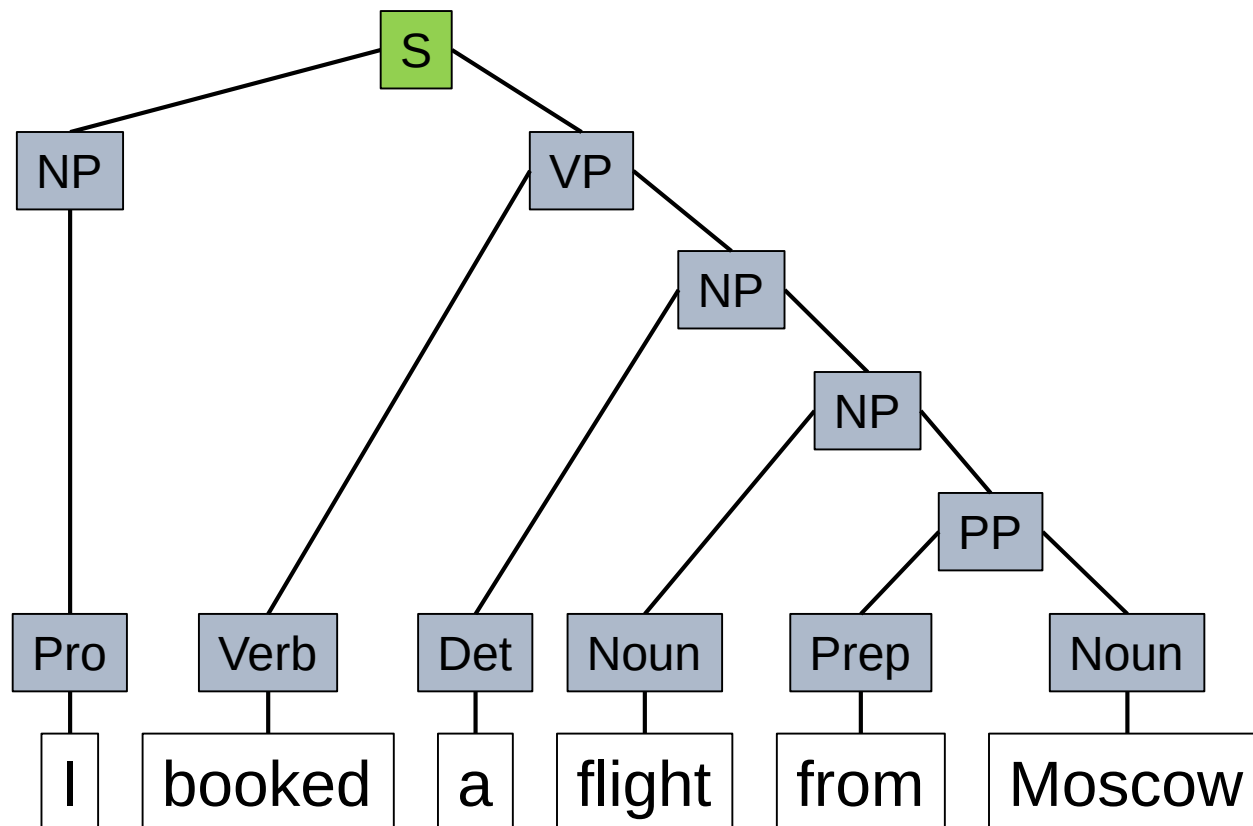
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb NP PP$
4. $NP \rightarrow Pro$
5. $NP \rightarrow Det NP$
6. $NP \rightarrow Noun PP$
7. $NP \rightarrow Noun$
8. $PP \rightarrow Prep Noun$
9. $Verb \rightarrow booked$
10. $Noun \rightarrow flight$
11. $Noun \rightarrow Moscow$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

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^{10 15 11} → → I booked a flight from Moscow



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. $Noun \rightarrow \underline{flight}$
11. $Noun \rightarrow \underline{Moscow}$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Det \rightarrow the$
15. $Prep \rightarrow \underline{from}$

Построение грамматики

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

Построение грамматики

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

1. $S \rightarrow NP VP$

Построение грамматики

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

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

Построение грамматики

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

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

Построение грамматики

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

1. $S \rightarrow NP VP$
2. $NP \rightarrow Pro$
3. $Pro \rightarrow I$
4. $VP \rightarrow Verb NP$

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

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

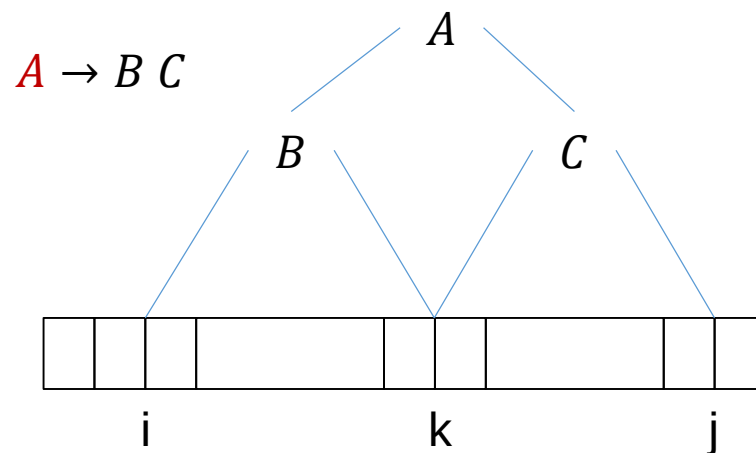
Алгоритм СΥΚ

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. $Det \rightarrow the$
15. $Prep \rightarrow from$

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

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

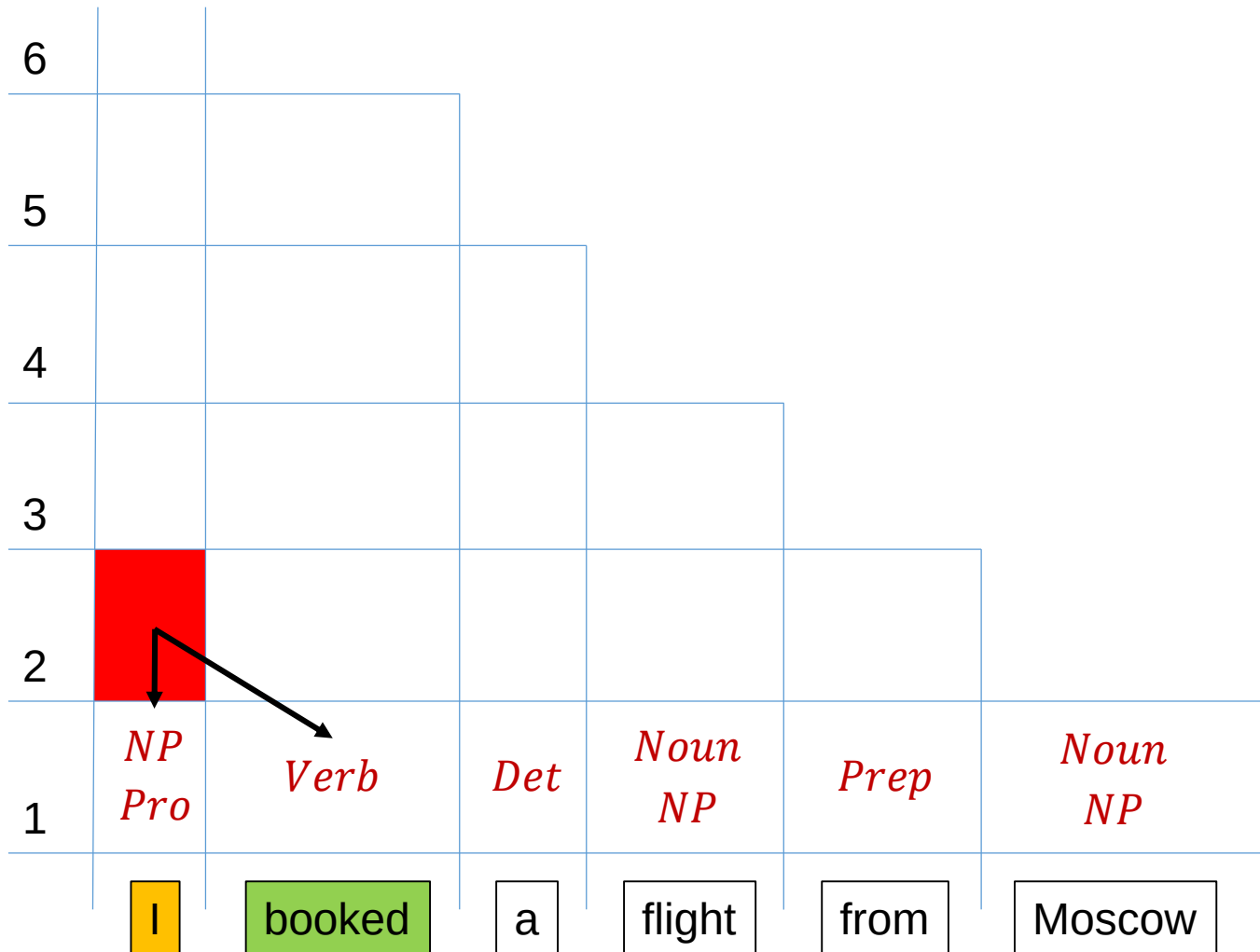


Алгоритм СҮК

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

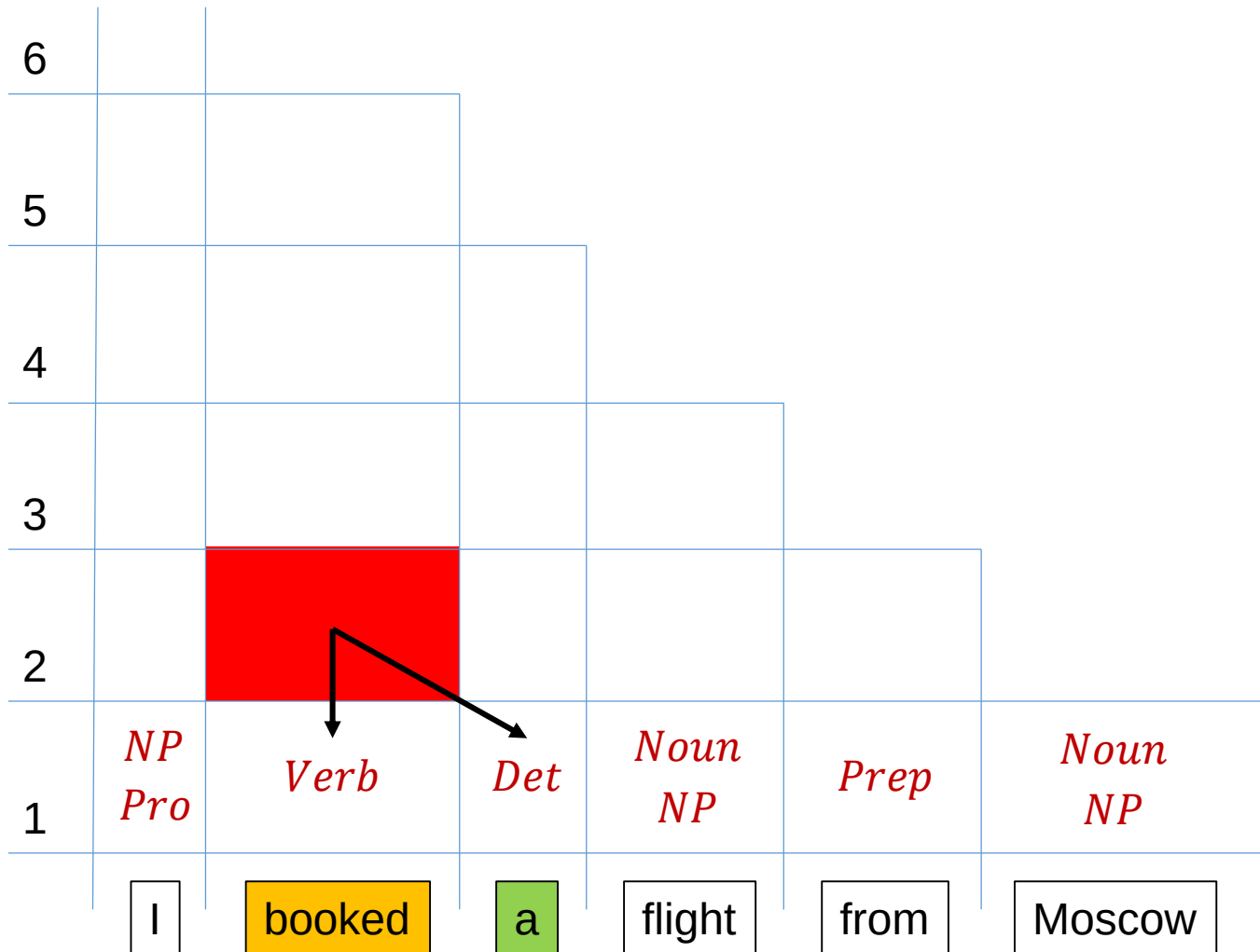
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК



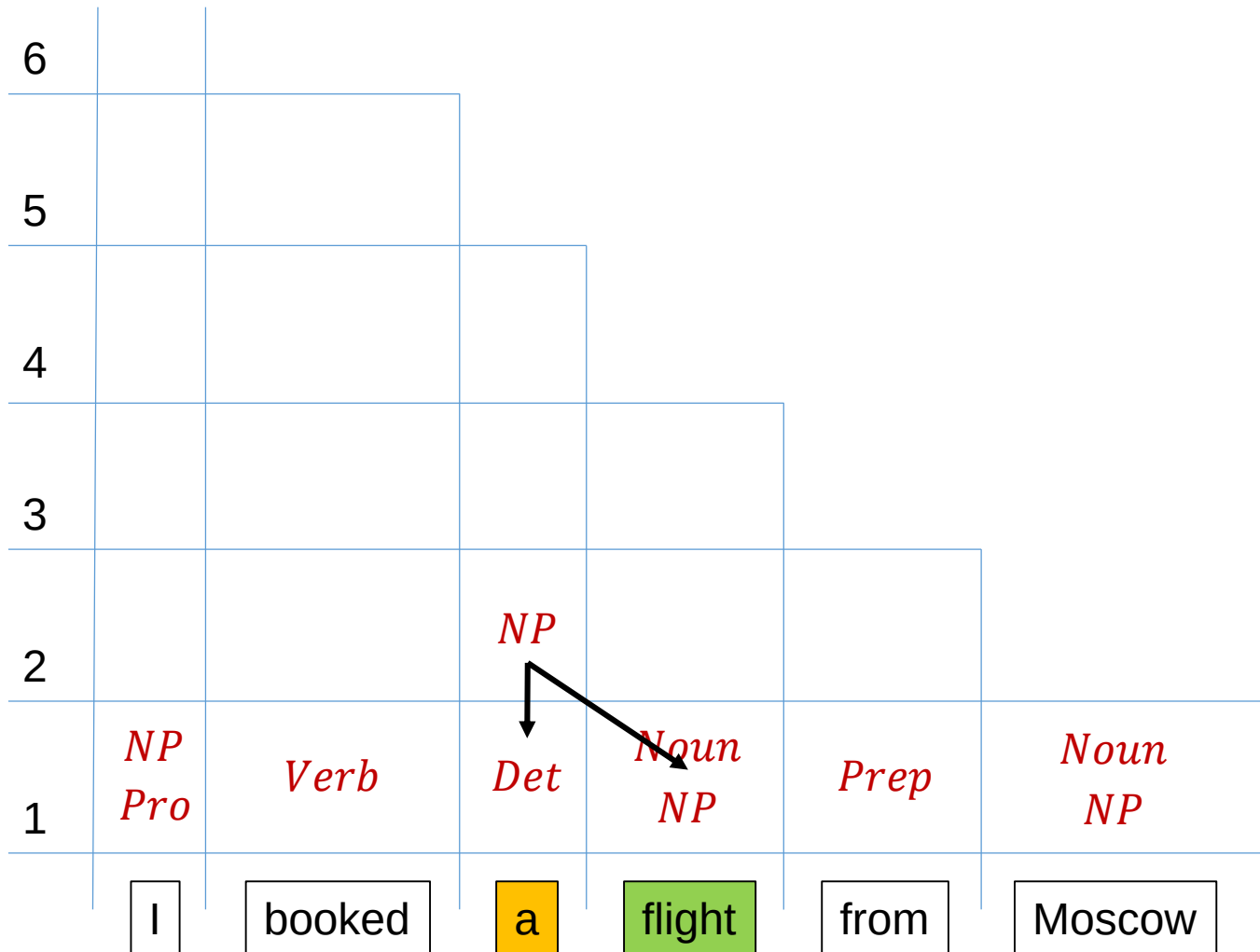
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК



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

Алгоритм СҮК



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

Алгоритм СҮК

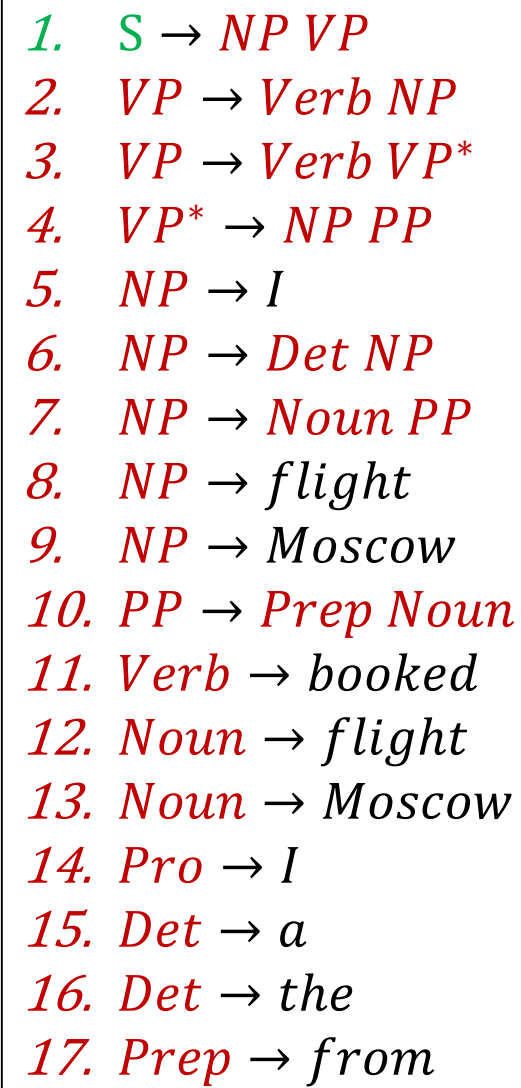
6						
5						
4						
3						
2			NP			
1	NP Pro	Verb	Det	Noun NP	Prep	Noun NP
	I	booked	a	flight	from	Moscow

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

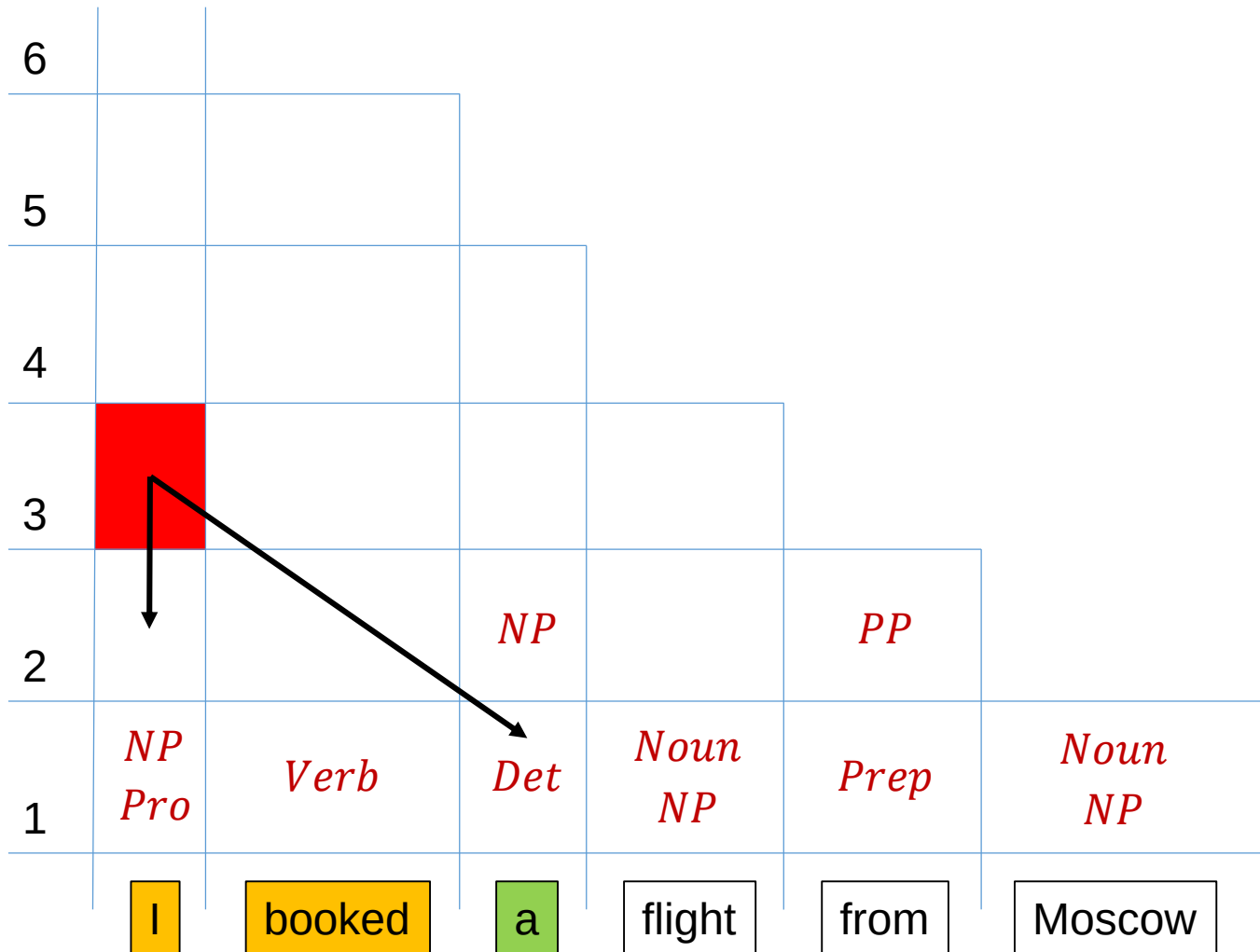
Алгоритм СҮК

6						
5						
4						
3						
2			<i>NP</i>		<i>PP</i>	
1	<i>NP</i> <i>Pro</i>	<i>Verb</i>	<i>Det</i>	<i>Noun</i> <i>NP</i>	<i>Prep</i>	<i>Noun</i> <i>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. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$
12. $Noun \rightarrow flight$
13. $Noun \rightarrow Moscow$
14. $Pro \rightarrow I$
15. $Det \rightarrow a$
16. $Det \rightarrow the$
17. $Prep \rightarrow from$

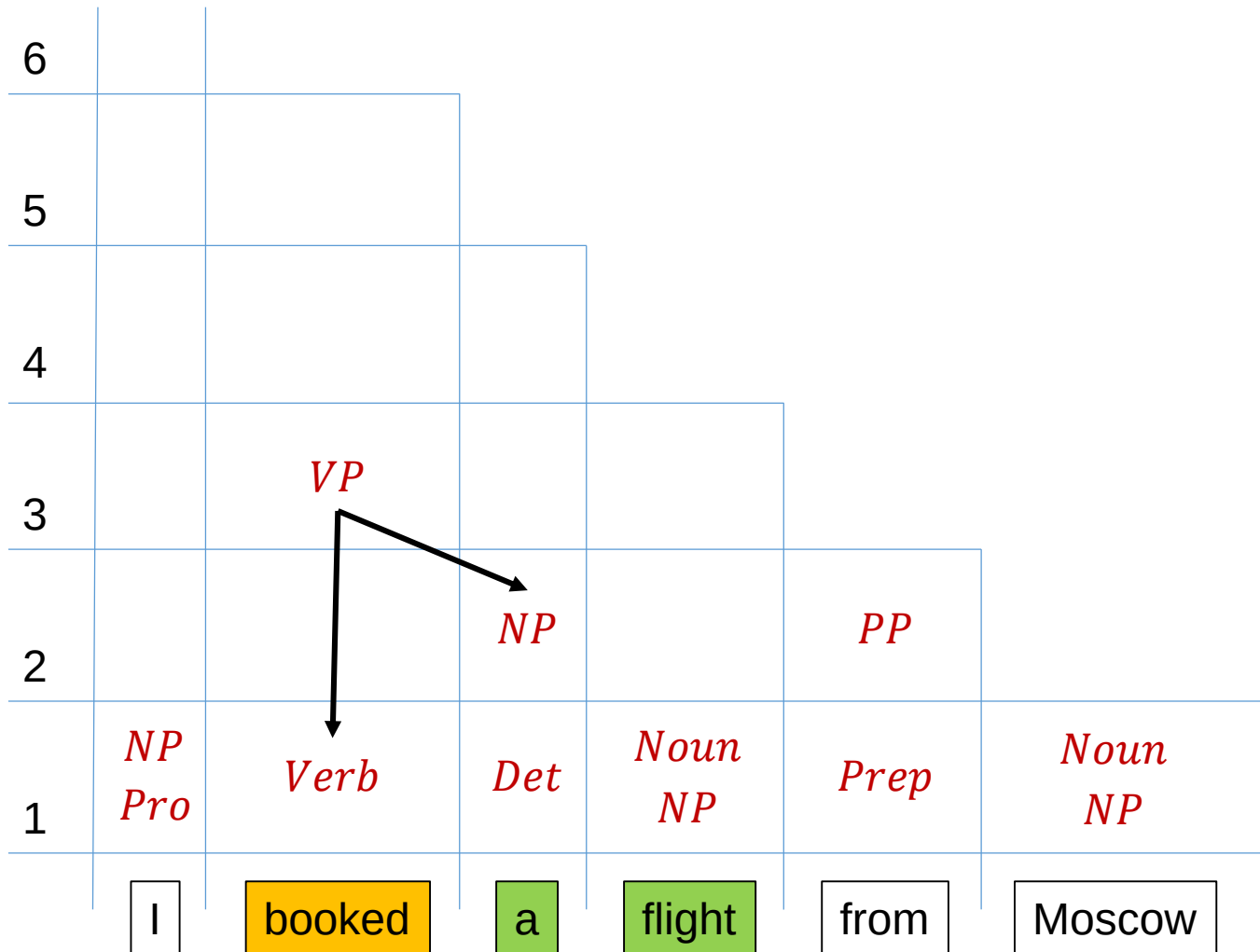


Алгоритм СҮК



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

Алгоритм СҮК



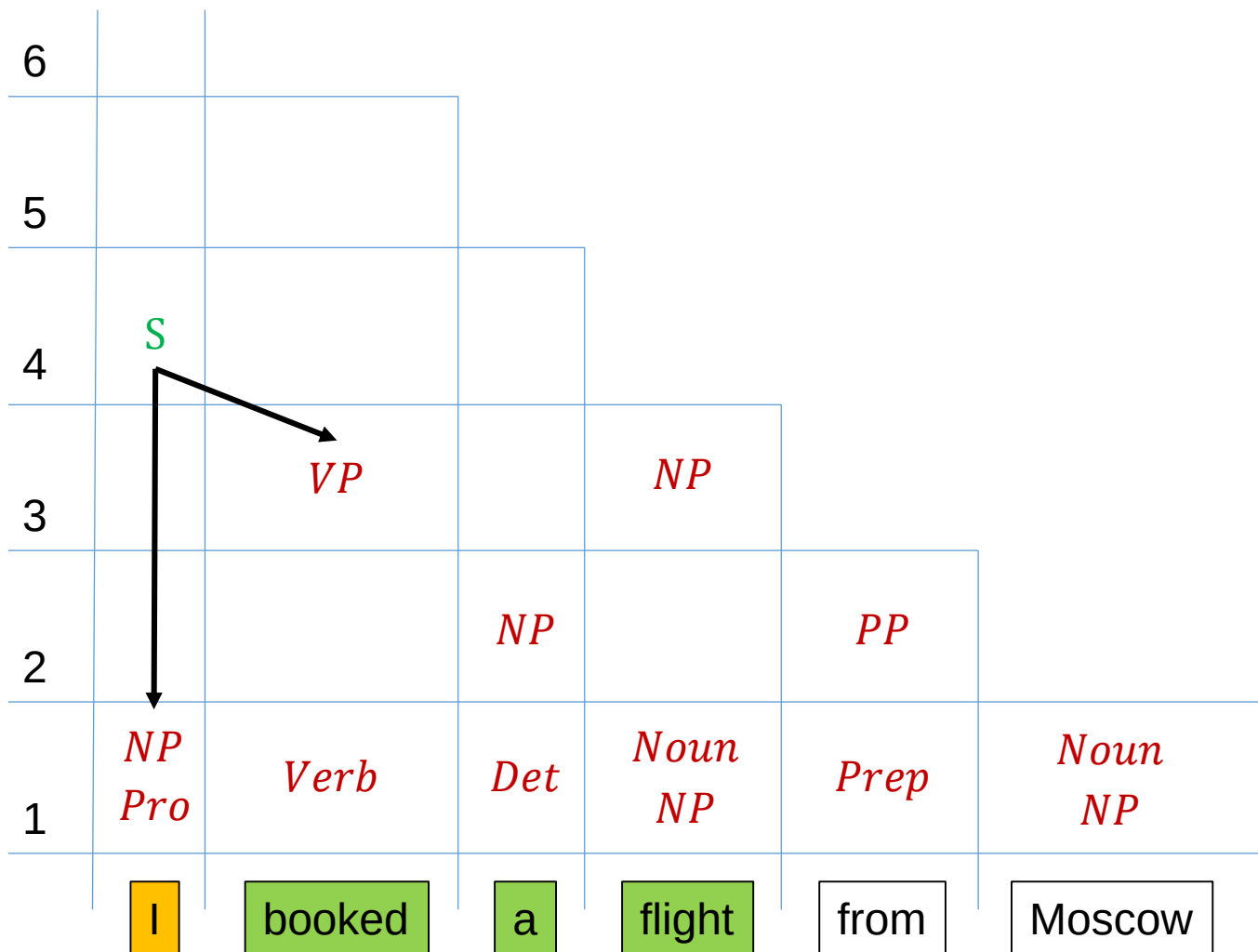
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК

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

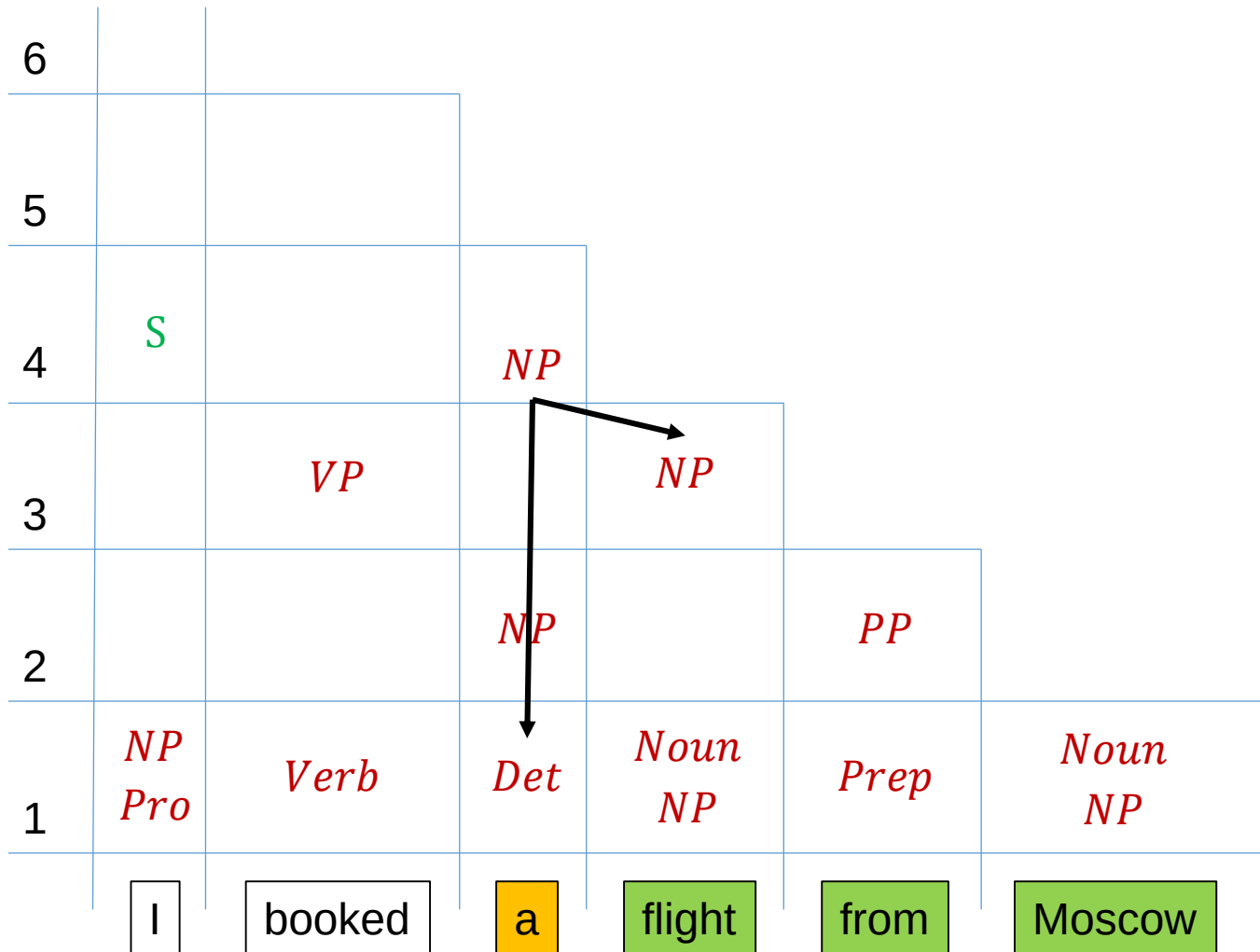
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК



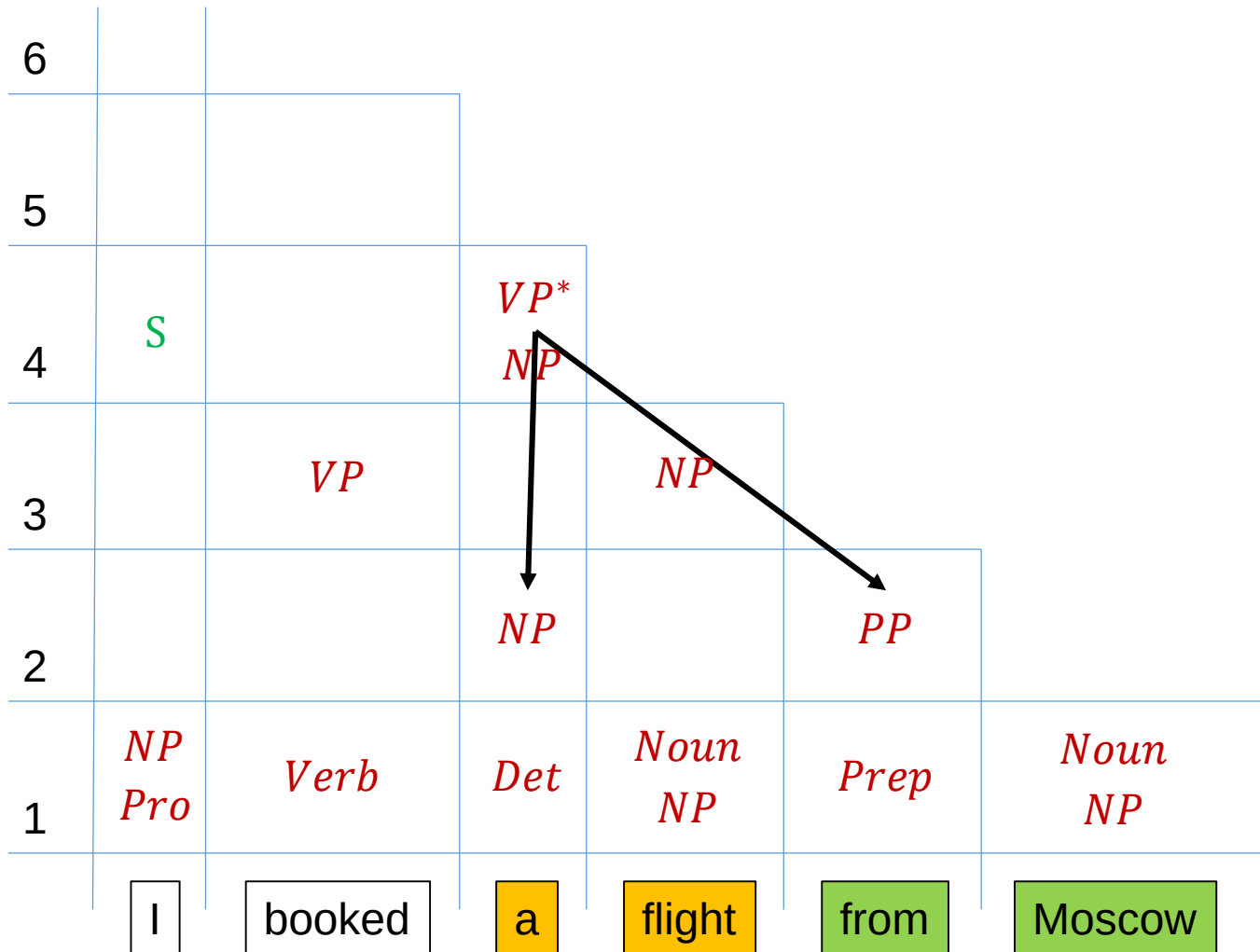
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК



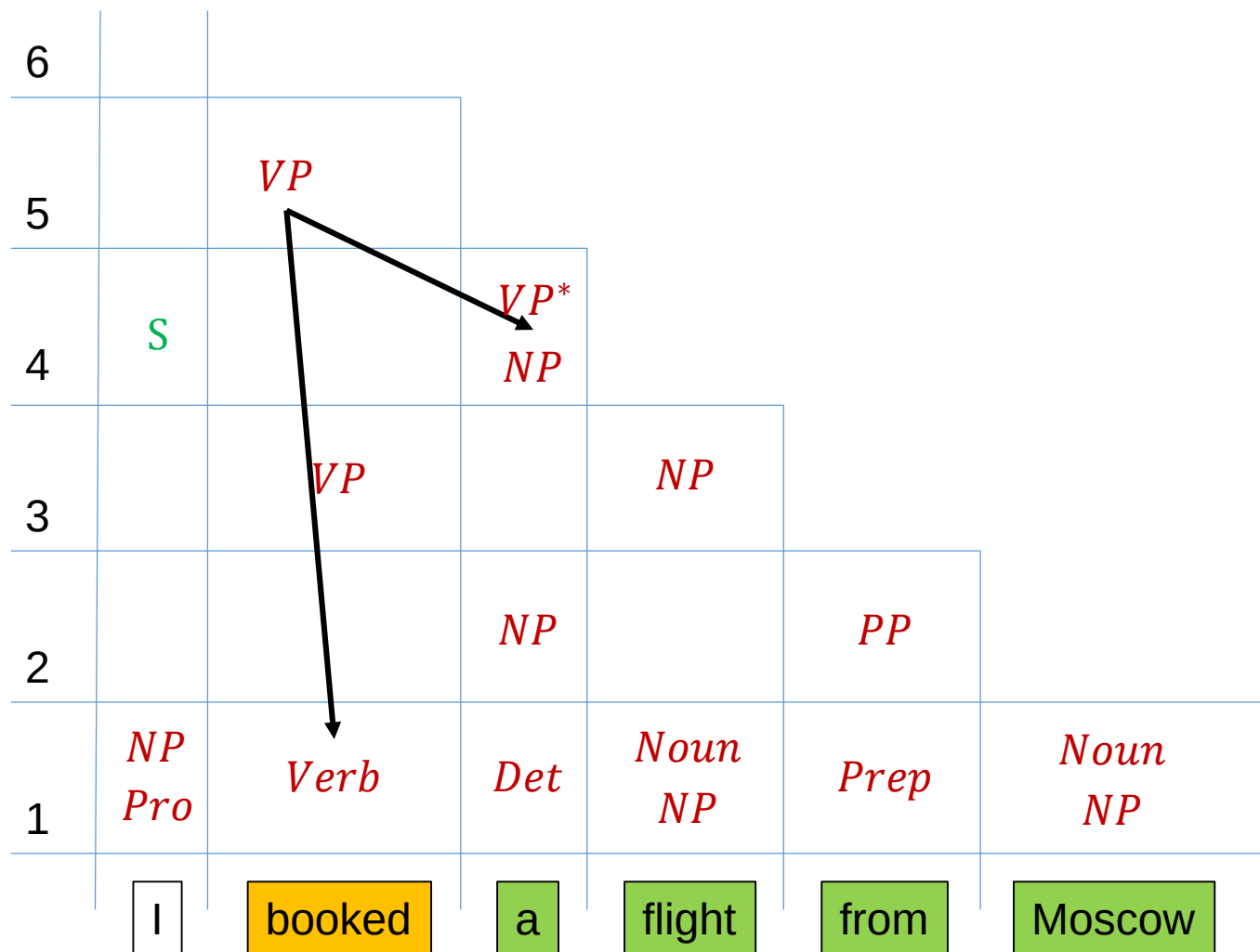
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм CYK



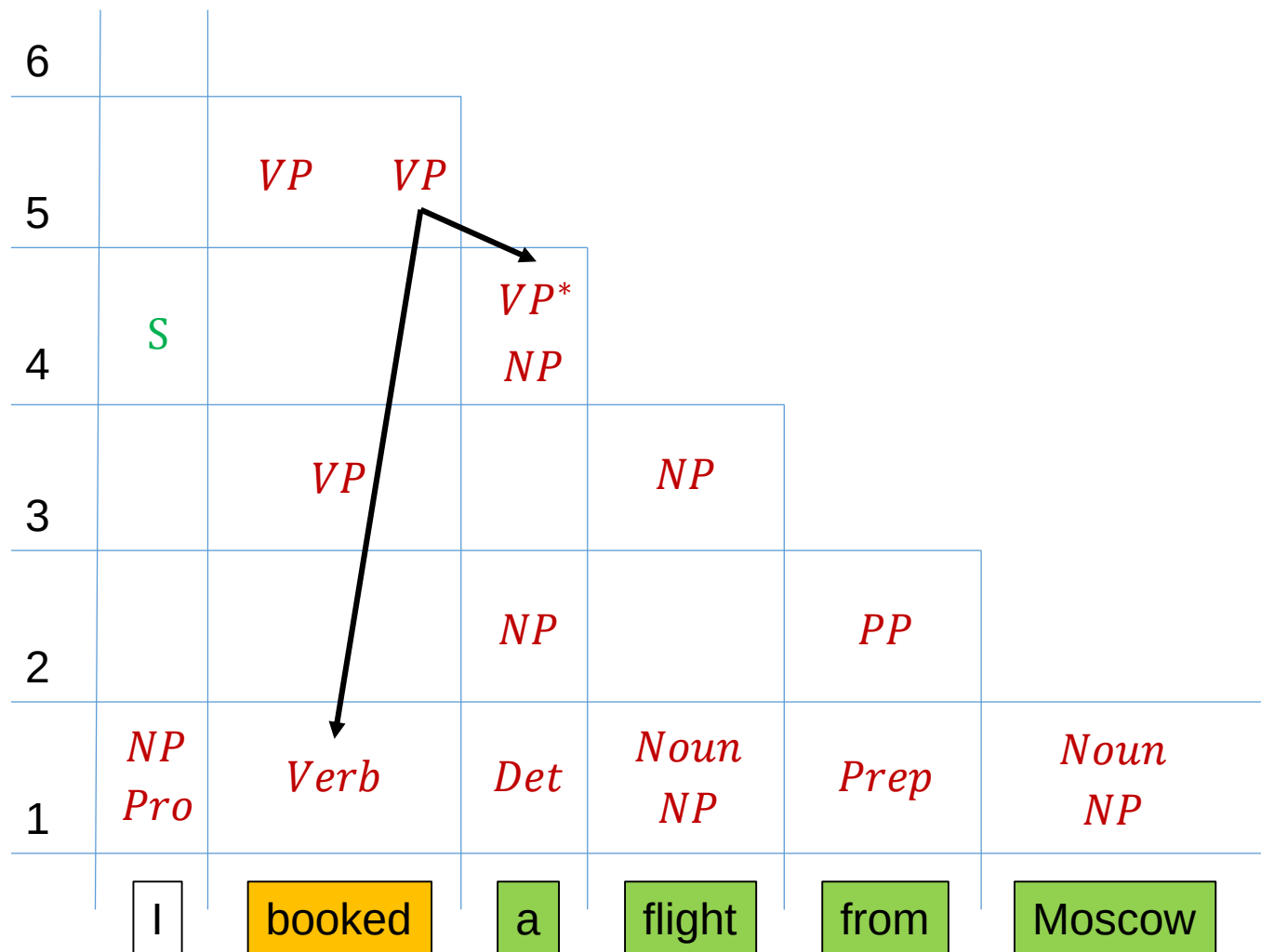
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК



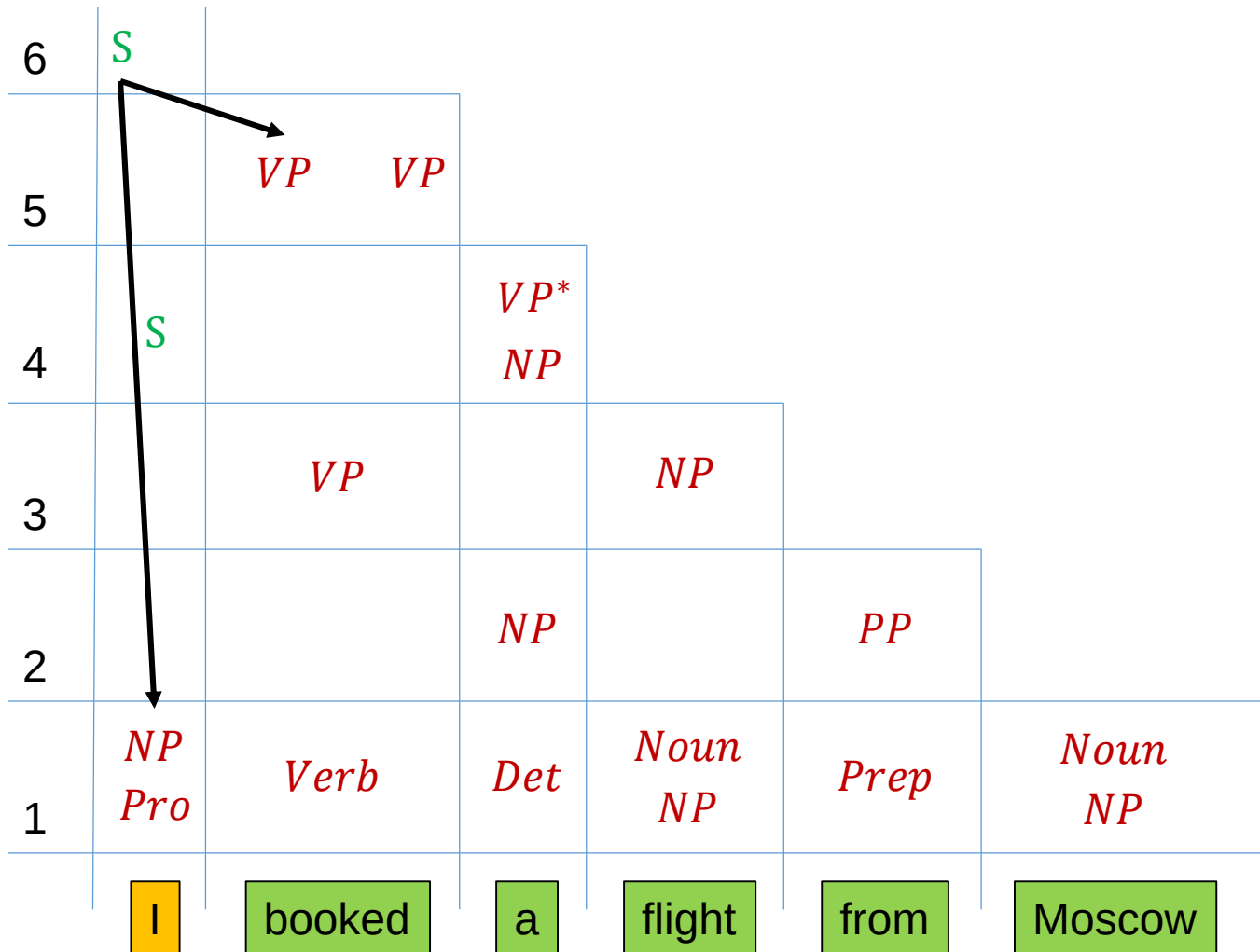
1. *S* → *NP VP*
2. *VP* → *Verb NP*
3. *VP* → *Verb VP**
4. *VP** → *NP PP*
5. *NP* → *I*
6. *NP* → *Det NP*
7. *NP* → *Noun PP*
8. *NP* → *flight*
9. *NP* → *Moscow*
10. *PP* → *Prep Noun*
11. *Verb* → *booked*
12. *Noun* → *flight*
13. *Noun* → *Moscow*
14. *Pro* → *I*
15. *Det* → *a*
16. *Det* →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

Алгоритм СҮК



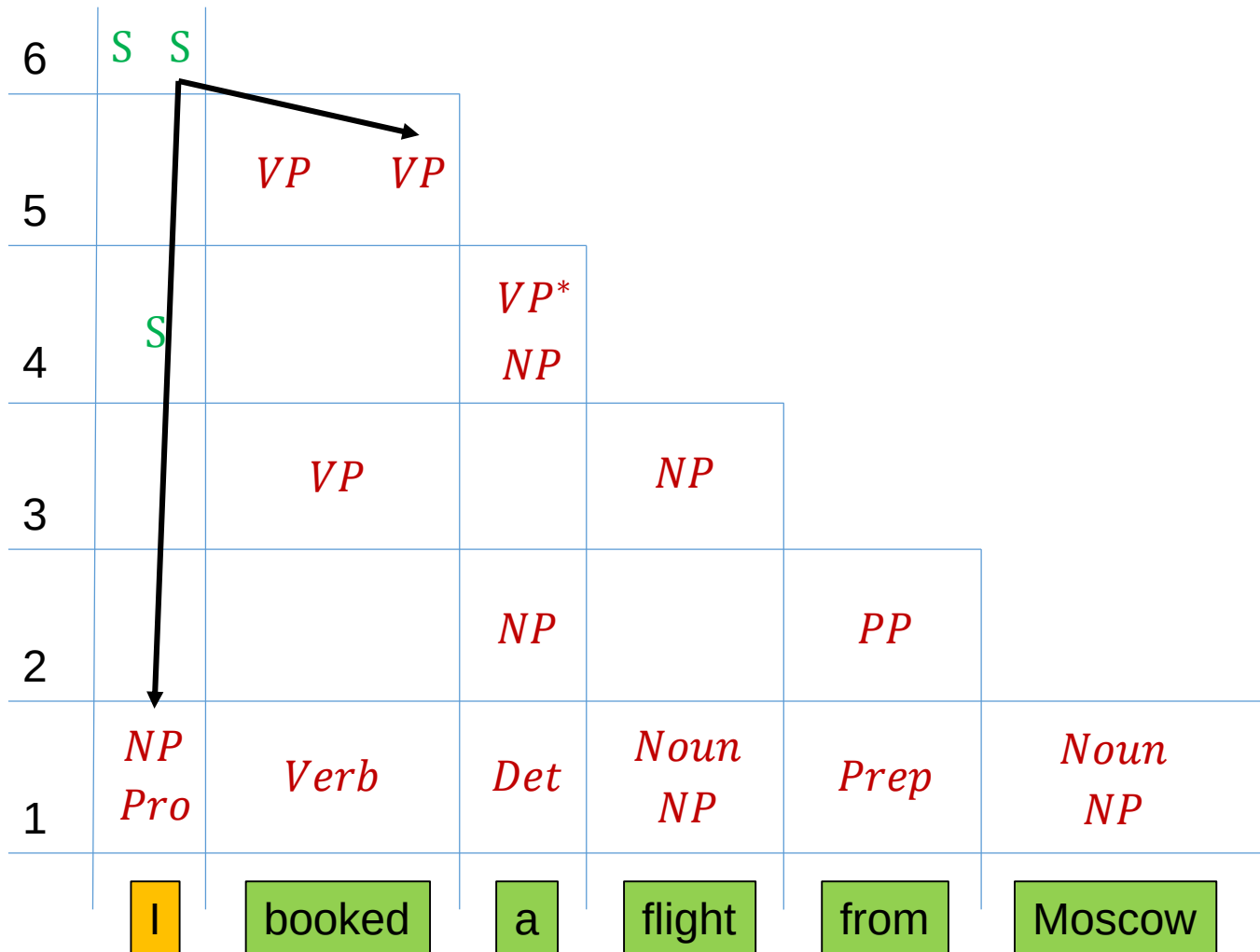
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК



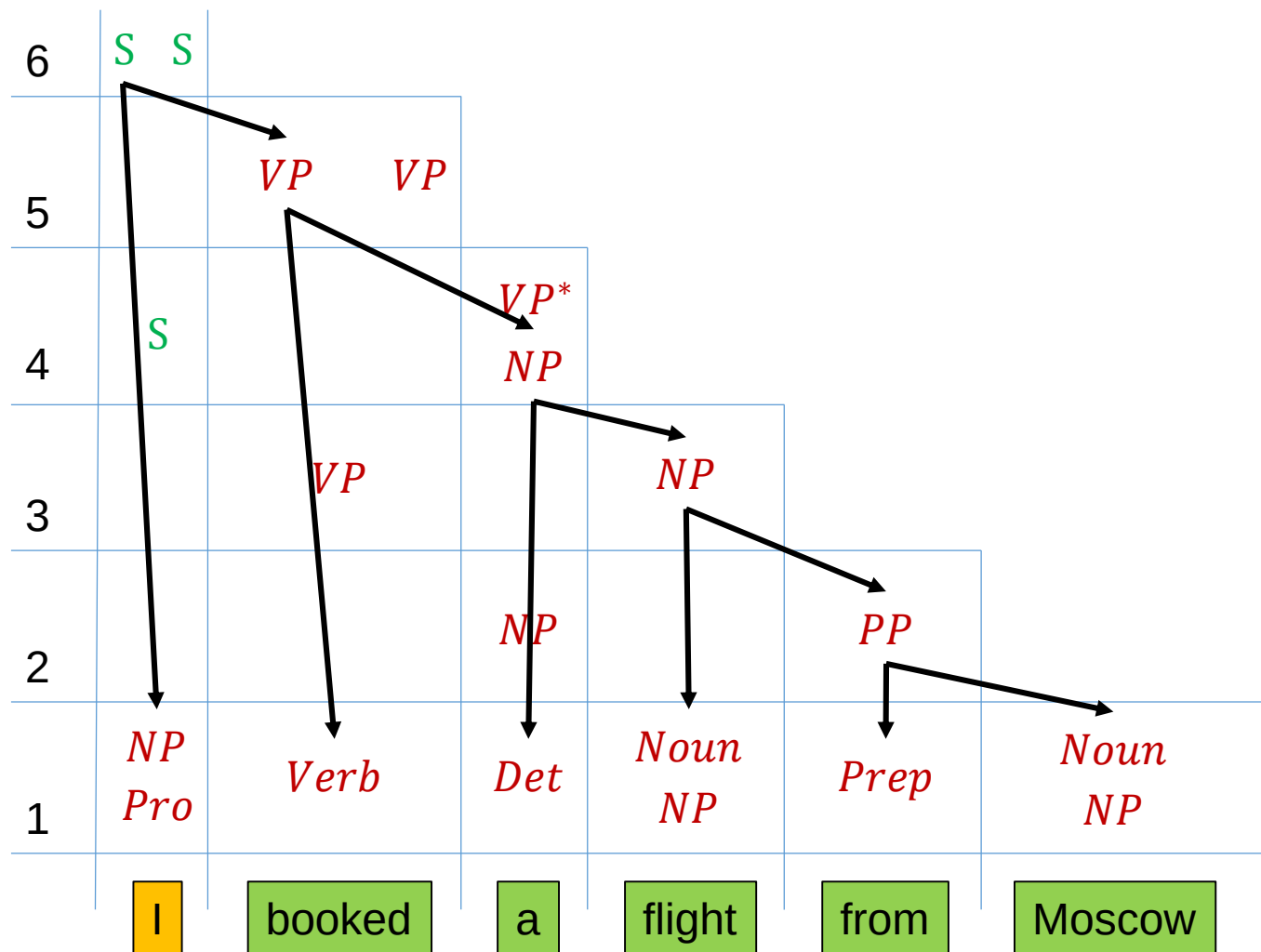
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК



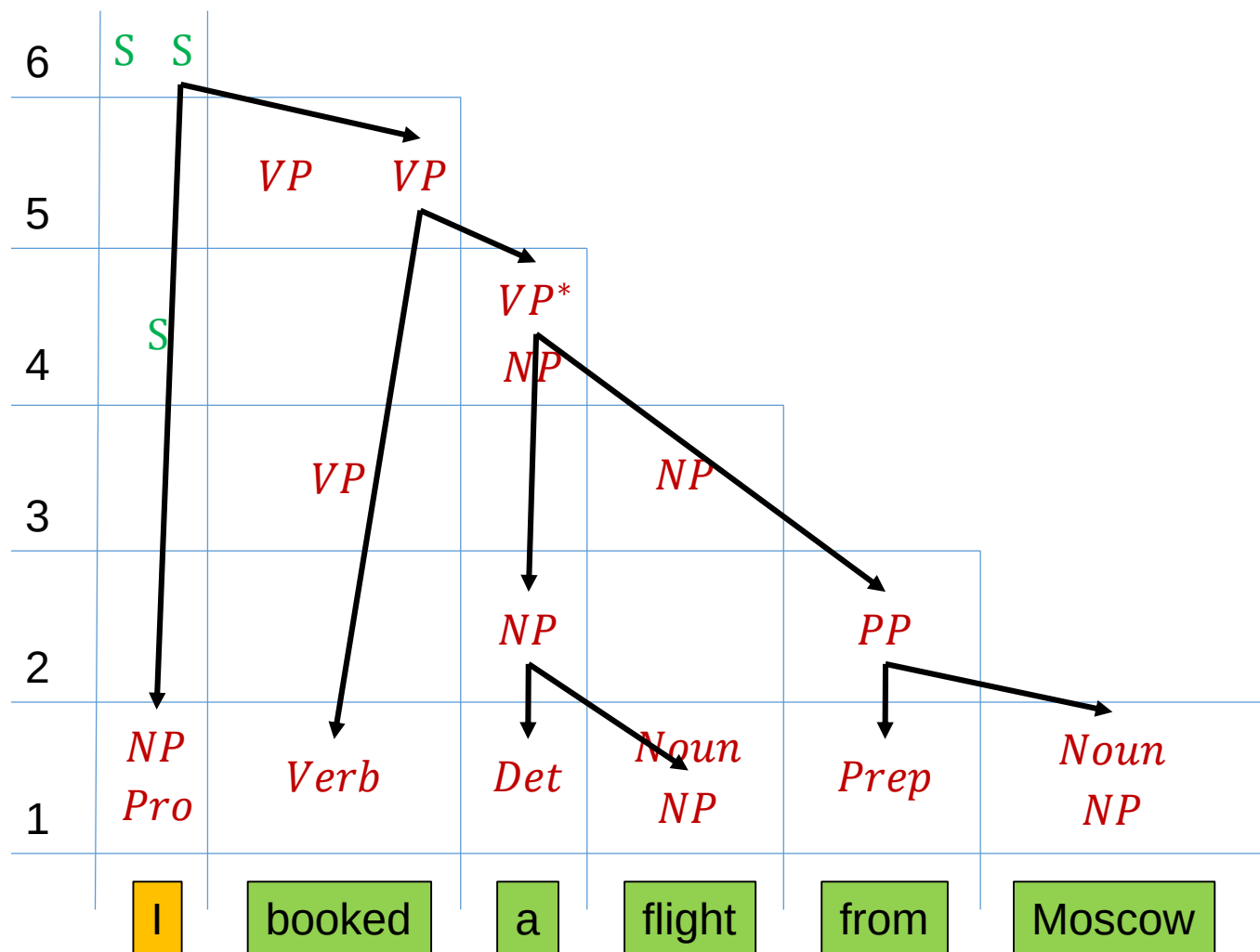
1. $S \rightarrow NP VP$
2. $VP \rightarrow Verb NP$
3. $VP \rightarrow Verb VP^*$
4. $VP^* \rightarrow NP PP$
5. $NP \rightarrow I$
6. $NP \rightarrow Det NP$
7. $NP \rightarrow Noun PP$
8. $NP \rightarrow flight$
9. $NP \rightarrow Moscow$
10. $PP \rightarrow Prep Noun$
11. $Verb \rightarrow booked$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

Алгоритм СҮК



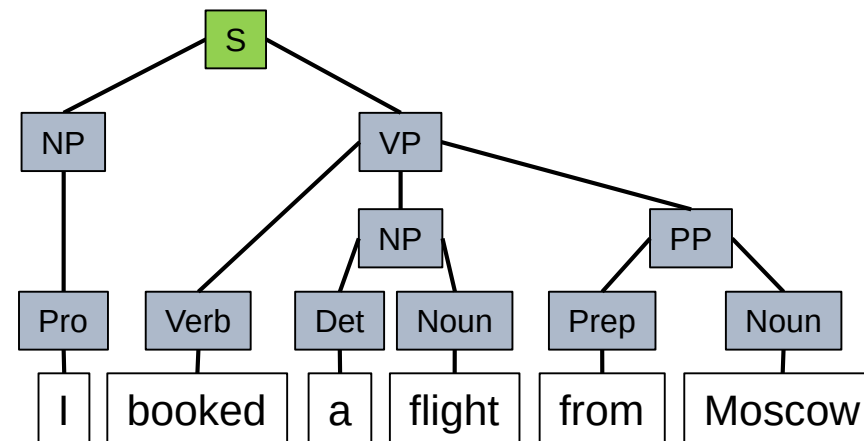
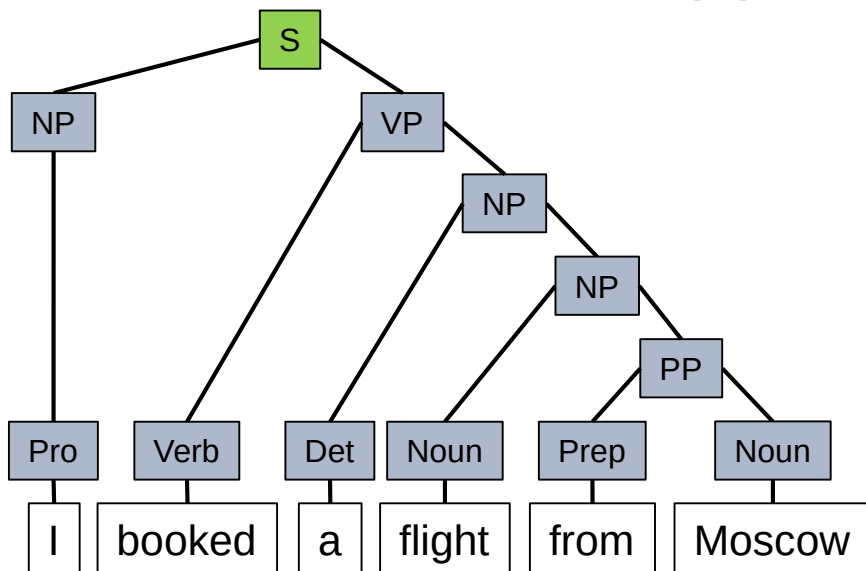
1. *S* → *NP VP*
2. *VP* → *Verb NP*
3. *VP* → *Verb VP**
4. *VP** → *NP PP*
5. *NP* → *I*
6. *NP* → *Det NP*
7. *NP* → *Noun PP*
8. *NP* → *flight*
9. *NP* → *Moscow*
10. *PP* → *Prep Noun*
11. *Verb* → *booked*
12. *Noun* → *flight*
13. *Noun* → *Moscow*
14. *Pro* → *I*
15. *Det* → *a*
16. *Det* →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

Алгоритм СҮК



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

Неоднозначность языка



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. $Noun \rightarrow flight$

10. $Verb \rightarrow booked$
11. $Noun \rightarrow Moscow$
12. $Pro \rightarrow I$
13. $Det \rightarrow a$
14. $Det \rightarrow the$
15. $Prep \rightarrow from$

Выбор лучшего дерева

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

$$p(tree) = \prod_i p(\alpha_i \rightarrow \beta_i)$$

$$p(\alpha \rightarrow \beta) = \frac{Count(\alpha \rightarrow \beta)}{Count(\alpha)}$$

$$t^*(s) = \arg \max_{t \in T(s)} p(t)$$

Выбор лучшего дерева

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

$$p(\text{tree}) = \prod_i p(\alpha_i \rightarrow \beta_i)$$

$$p(\alpha \rightarrow \beta) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)}$$

$$t^*(s) = \arg \max_{t \in T(s)} p(t)$$

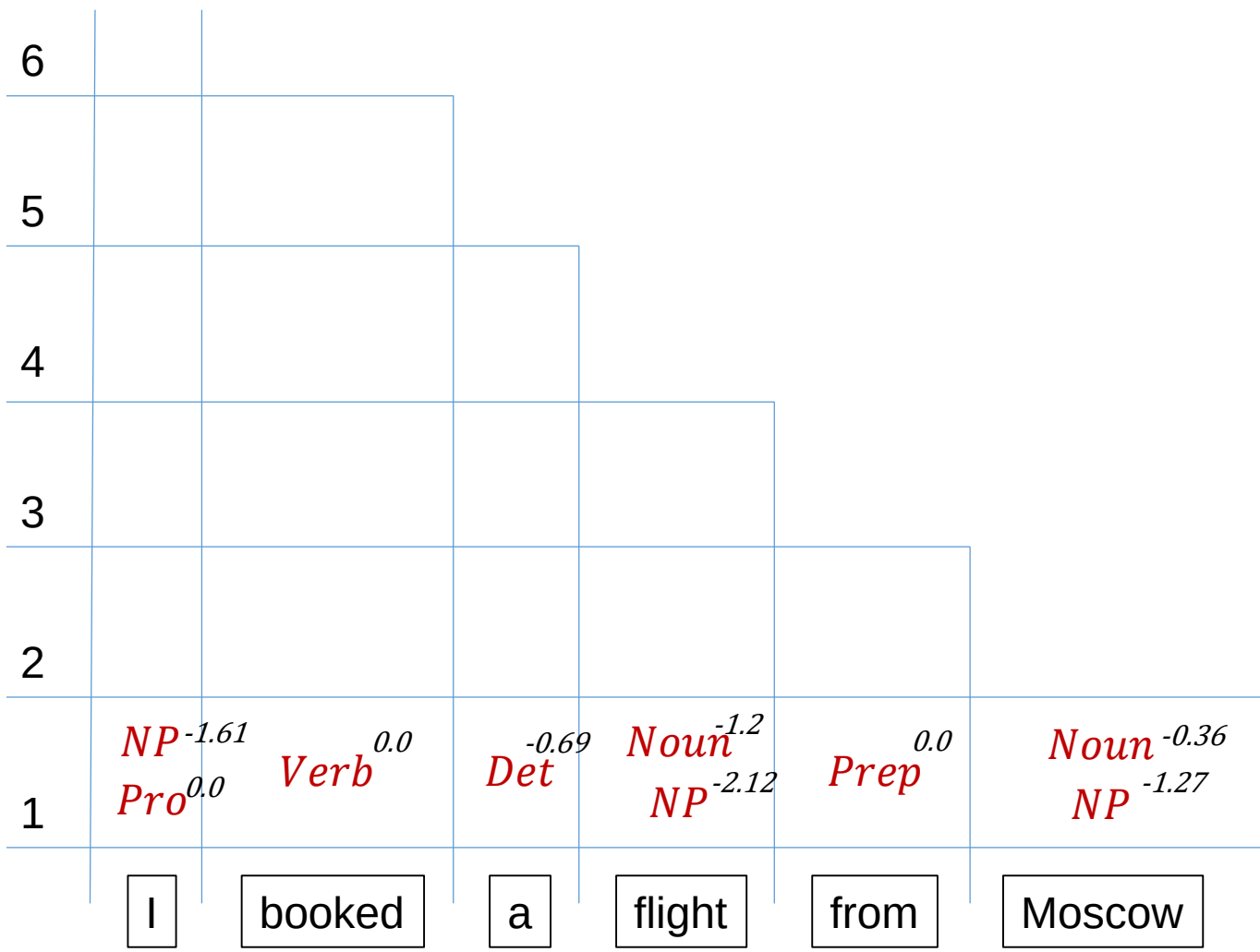
1. <i>S</i> → <i>NP VP</i>	1.0
2. <i>VP</i> → <i>Verb NP</i>	0.9
3. <i>VP</i> → <i>Verb NP PP</i>	0.1
4. <i>NP</i> → <i>Pro</i>	0.2
5. <i>NP</i> → <i>Det NP</i>	0.3
6. <i>NP</i> → <i>Noun PP</i>	0.1
7. <i>NP</i> → <i>Noun</i>	0.4
8. <i>PP</i> → <i>Prep Noun</i>	1.0
9. <i>Verb</i> → <i>booked</i>	1.0
10. <i>Noun</i> → <i>flight</i>	0.3
11. <i>Noun</i> → <i>Moscow</i>	0.7

Алгоритм СΥΚ

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

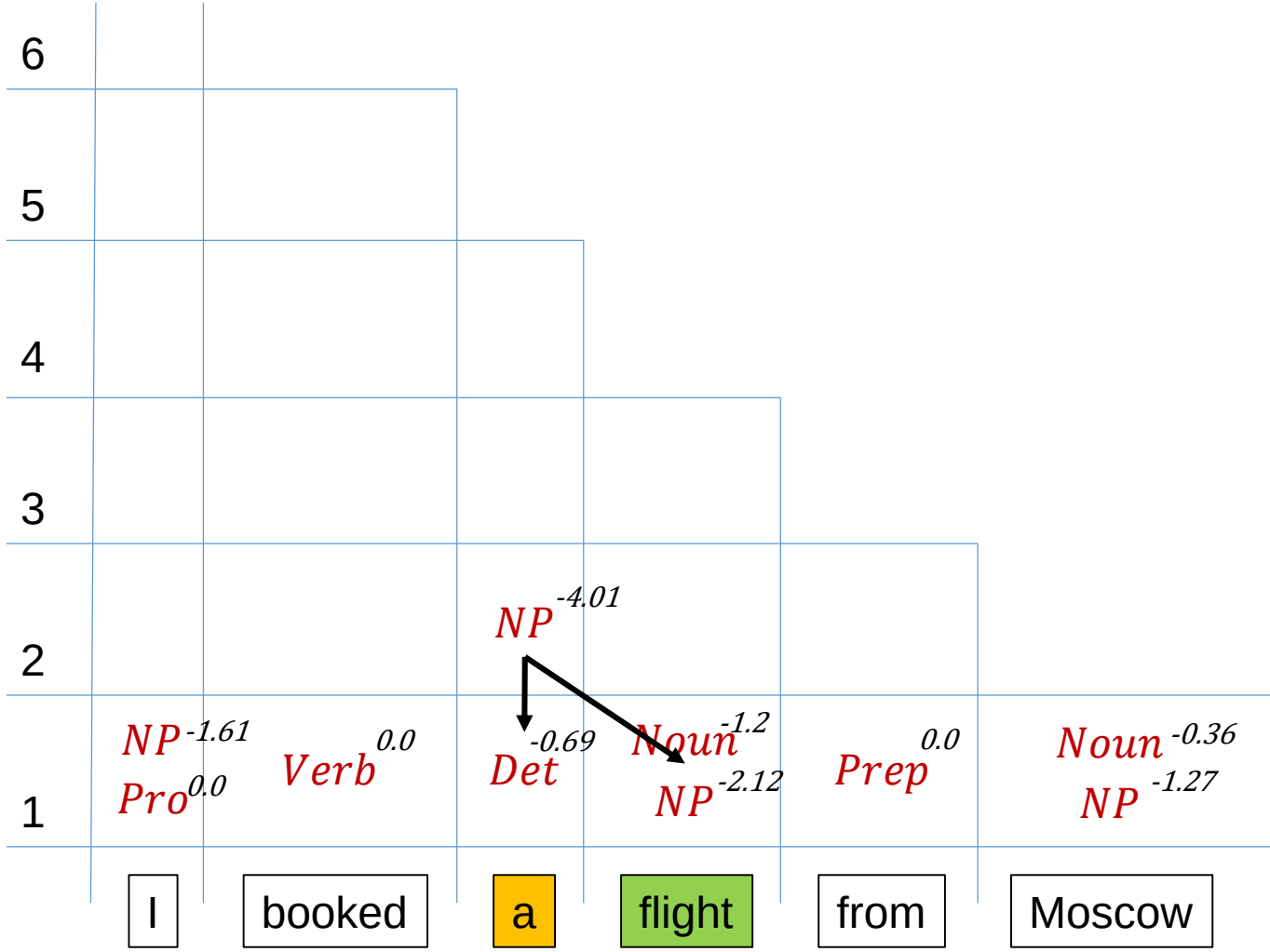
1.	<i>S</i> → <i>NP VP</i>	1.0
2.	<i>VP</i> → <i>Verb NP</i>	0.9
3.	<i>VP</i> → <i>Verb VP*</i>	0.1
4.	<i>VP*</i> → <i>NP PP</i>	1.0
5.	<i>NP</i> → <i>I</i>	0.2
6.	<i>NP</i> → <i>Det NP</i>	0.3
7.	<i>NP</i> → <i>Noun PP</i>	0.1
8.	<i>NP</i> → <i>flight</i>	0.12
9.	<i>NP</i> → <i>Moscow</i>	0.28
10.	<i>PP</i> → <i>Prep Noun</i>	1.0
11.	<i>Verb</i> → <i>booked</i>	1.0
12.	<i>Noun</i> → <i>flight</i>	0.3
13.	<i>Noun</i> → <i>Moscow</i>	0.7
14.	<i>Pro</i> → <i>I</i>	1.0
15.	<i>Det</i> → <i>a</i>	0.5
16.	<i>Det</i> → <i>the</i>	0.5
17.	<i>Prep</i> → <i>from</i>	1.0

Алгоритм СҮК



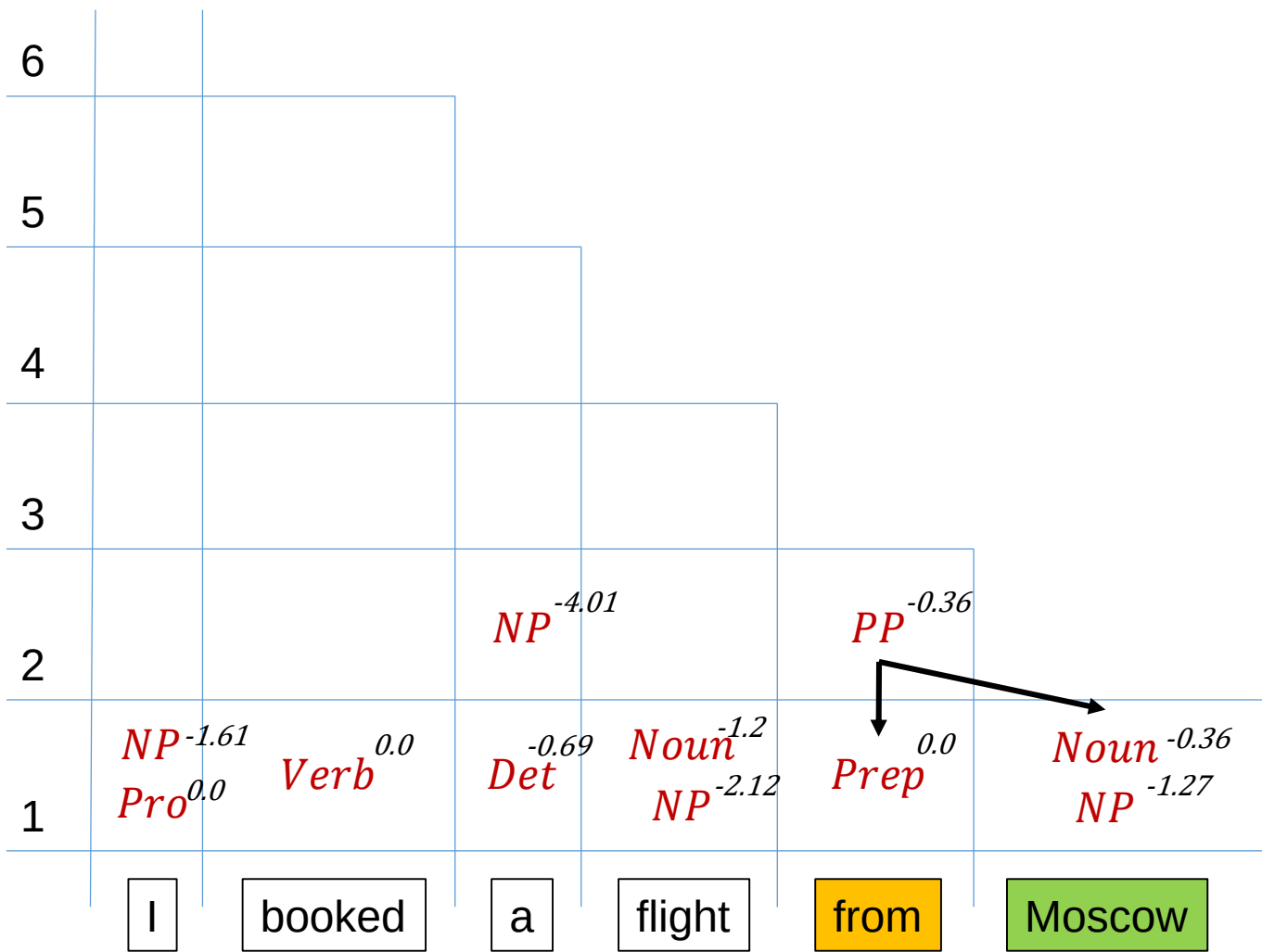
1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0

Алгоритм СҮК



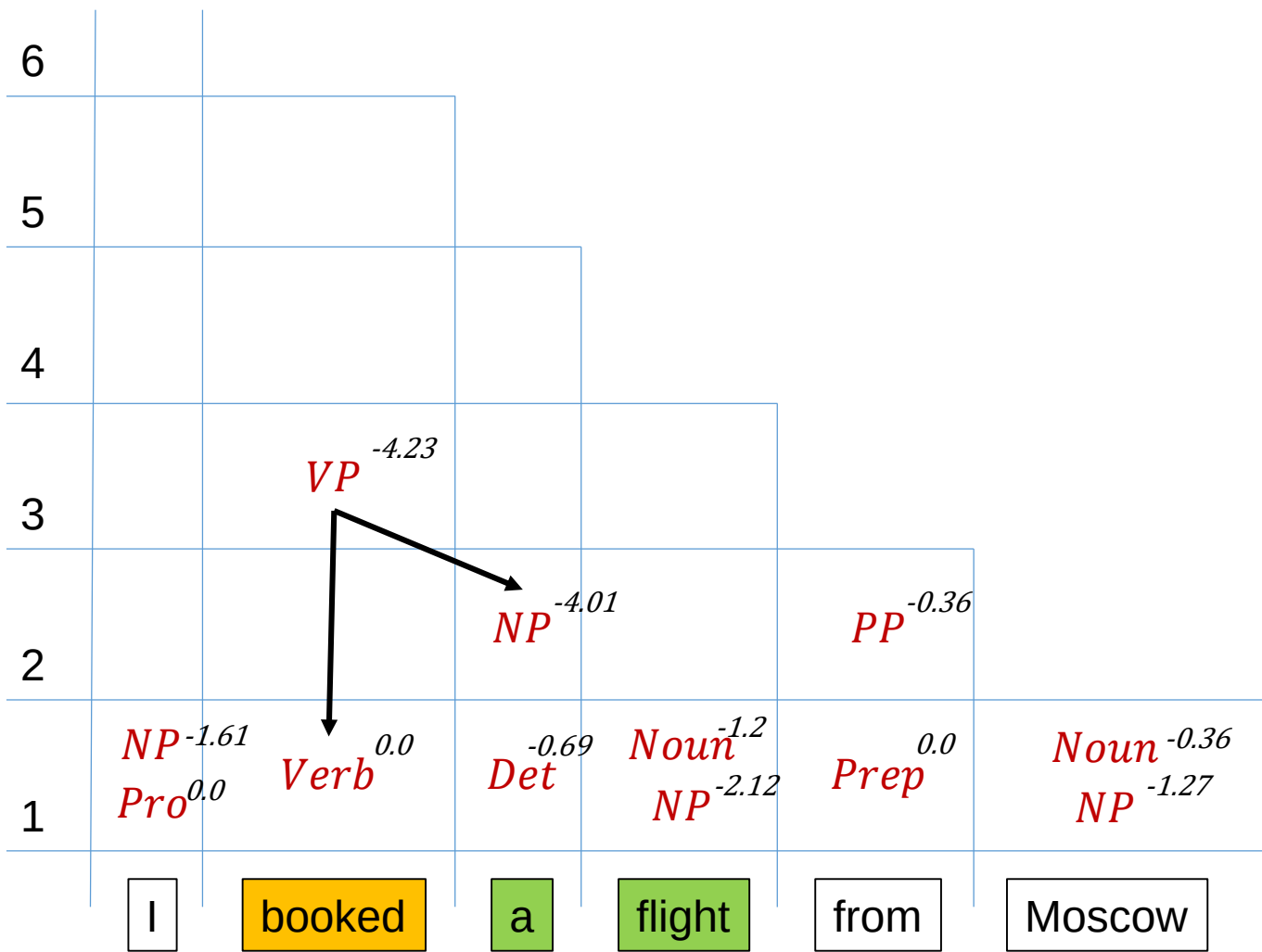
1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0

Алгоритм СҮК



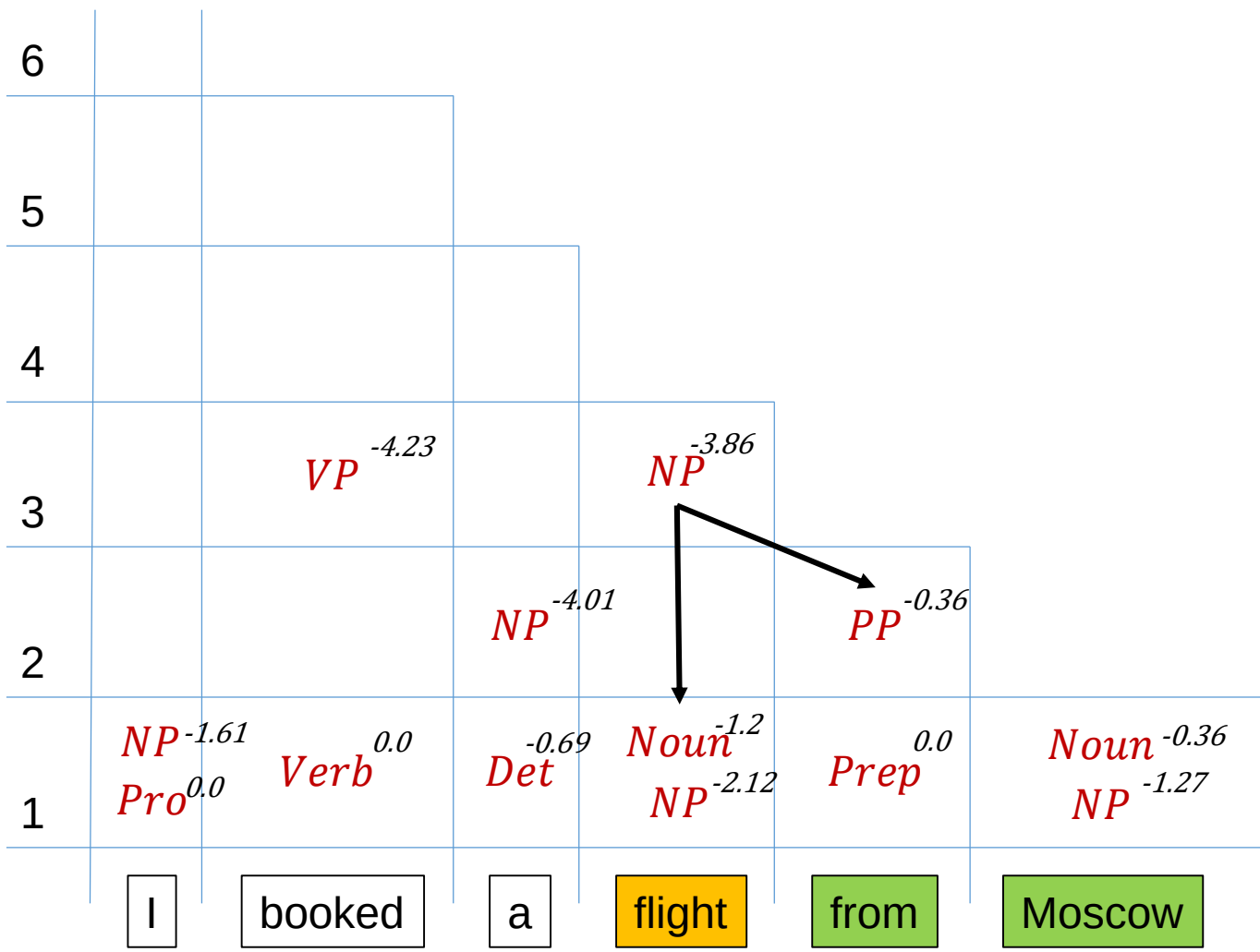
1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0

Алгоритм СҮК



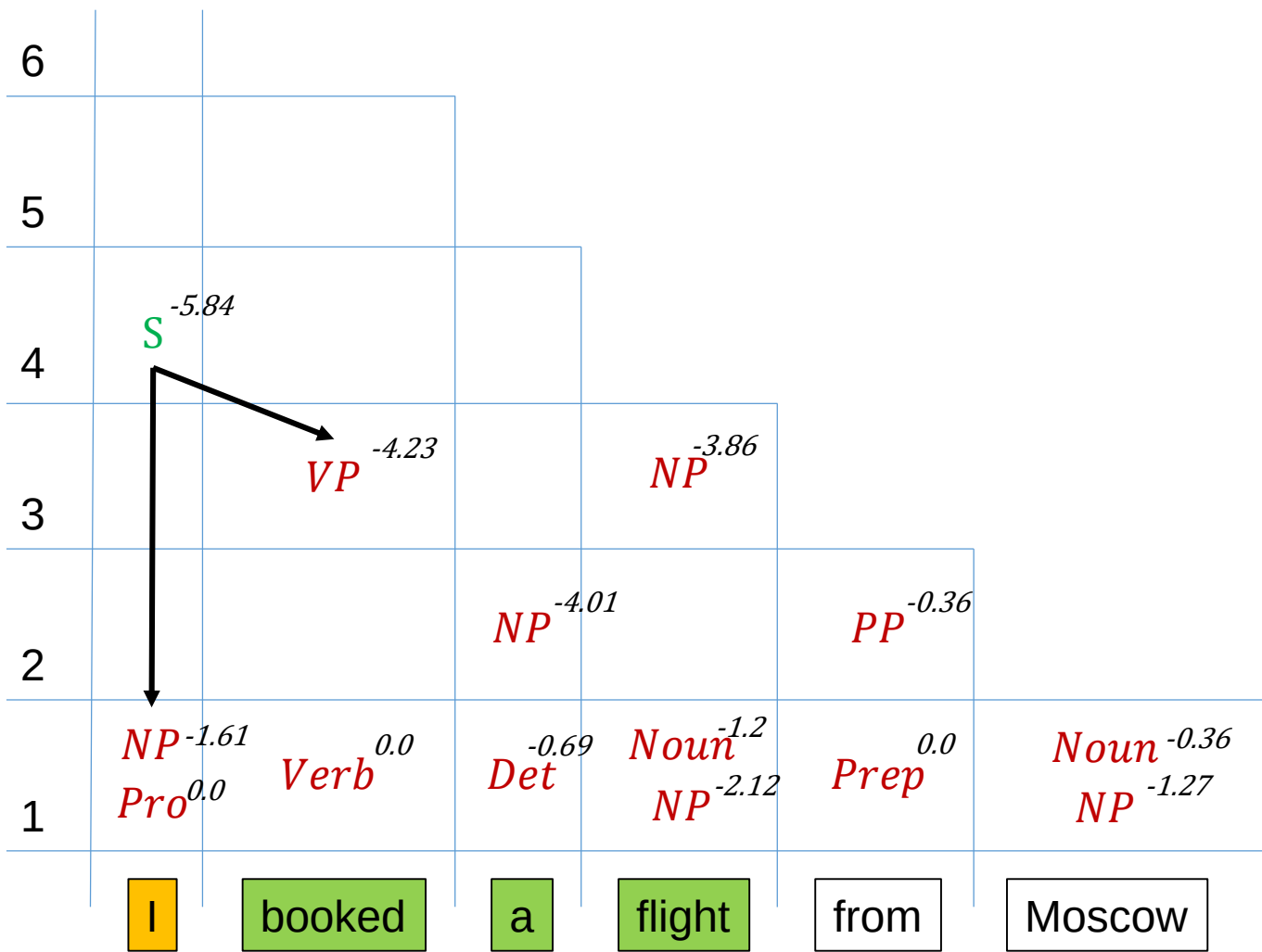
1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0

Алгоритм СҮК



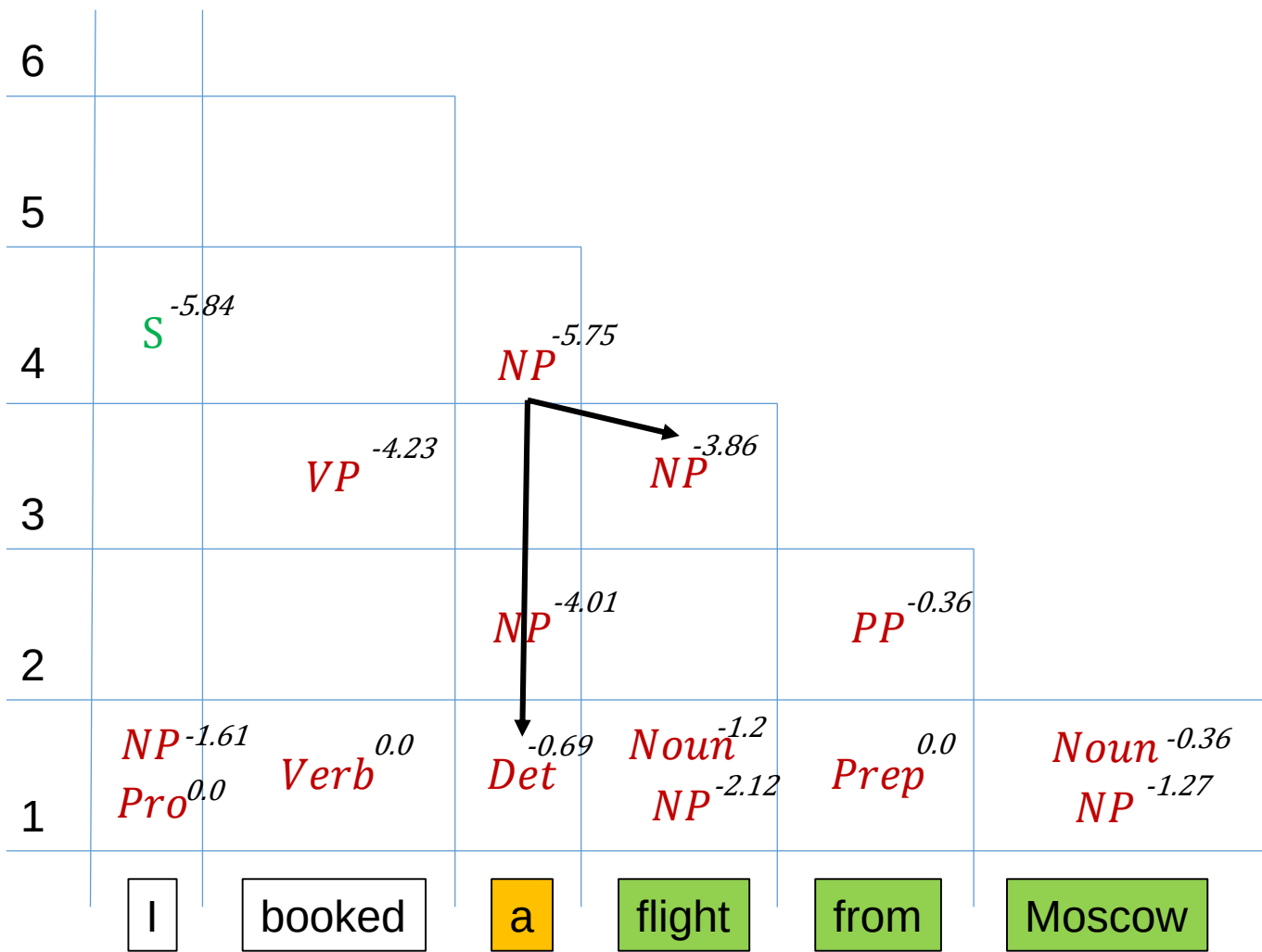
1.	<i>S</i> → <i>NP VP</i>	1.0
2.	<i>VP</i> → <i>Verb NP</i>	0.9
3.	<i>VP</i> → <i>Verb VP*</i>	0.1
4.	<i>VP*</i> → <i>NP PP</i>	1.0
5.	<i>NP</i> → <i>I</i>	0.2
6.	<i>NP</i> → <i>Det NP</i>	0.3
7.	<i>NP</i> → <i>Noun PP</i>	0.1
8.	<i>NP</i> → <i>flight</i>	0.12
9.	<i>NP</i> → <i>Moscow</i>	0.28
10.	<i>PP</i> → <i>Prep Noun</i>	1.0
11.	<i>Verb</i> → <i>booked</i>	1.0
12.	<i>Noun</i> → <i>flight</i>	0.3
13.	<i>Noun</i> → <i>Moscow</i>	0.7
14.	<i>Pro</i> → <i>I</i>	1.0
15.	<i>Det</i> → <i>a</i>	0.5
16.	<i>Det</i> → <i>the</i>	0.5
17.	<i>Prep</i> → <i>from</i>	1.0

Алгоритм СҮК



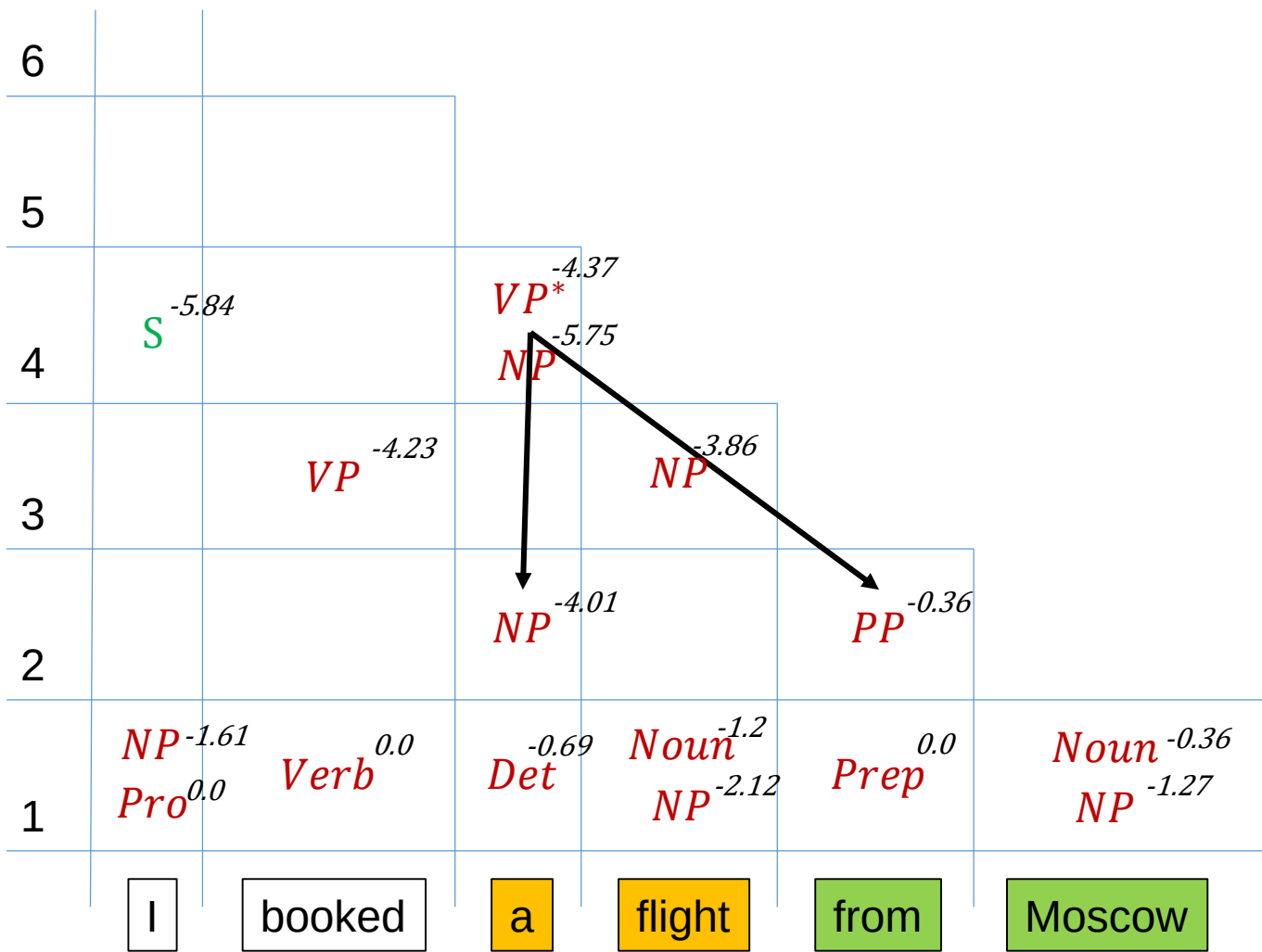
1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0

Алгоритм СҮК



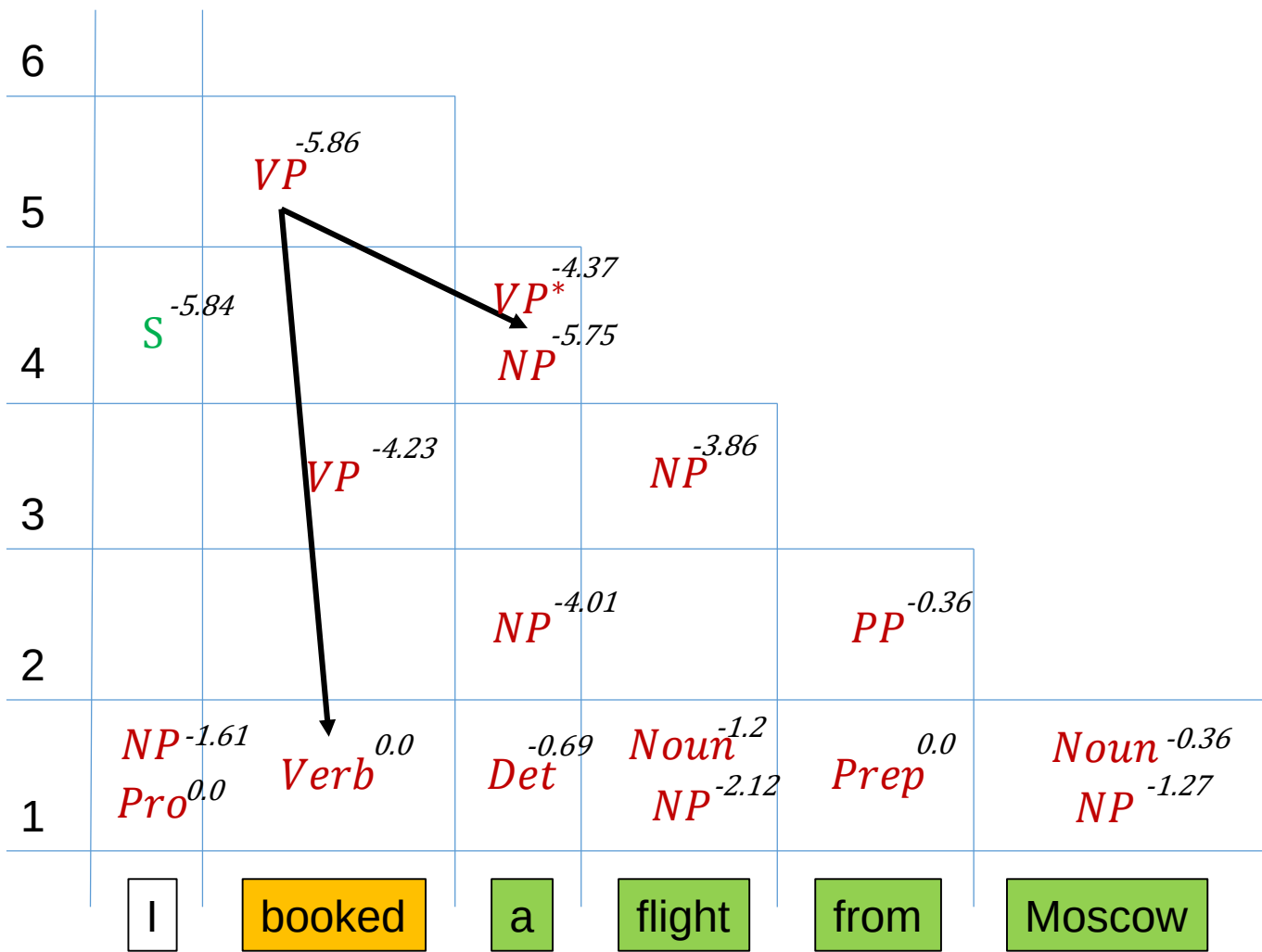
1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0

Алгоритм СΥΚ



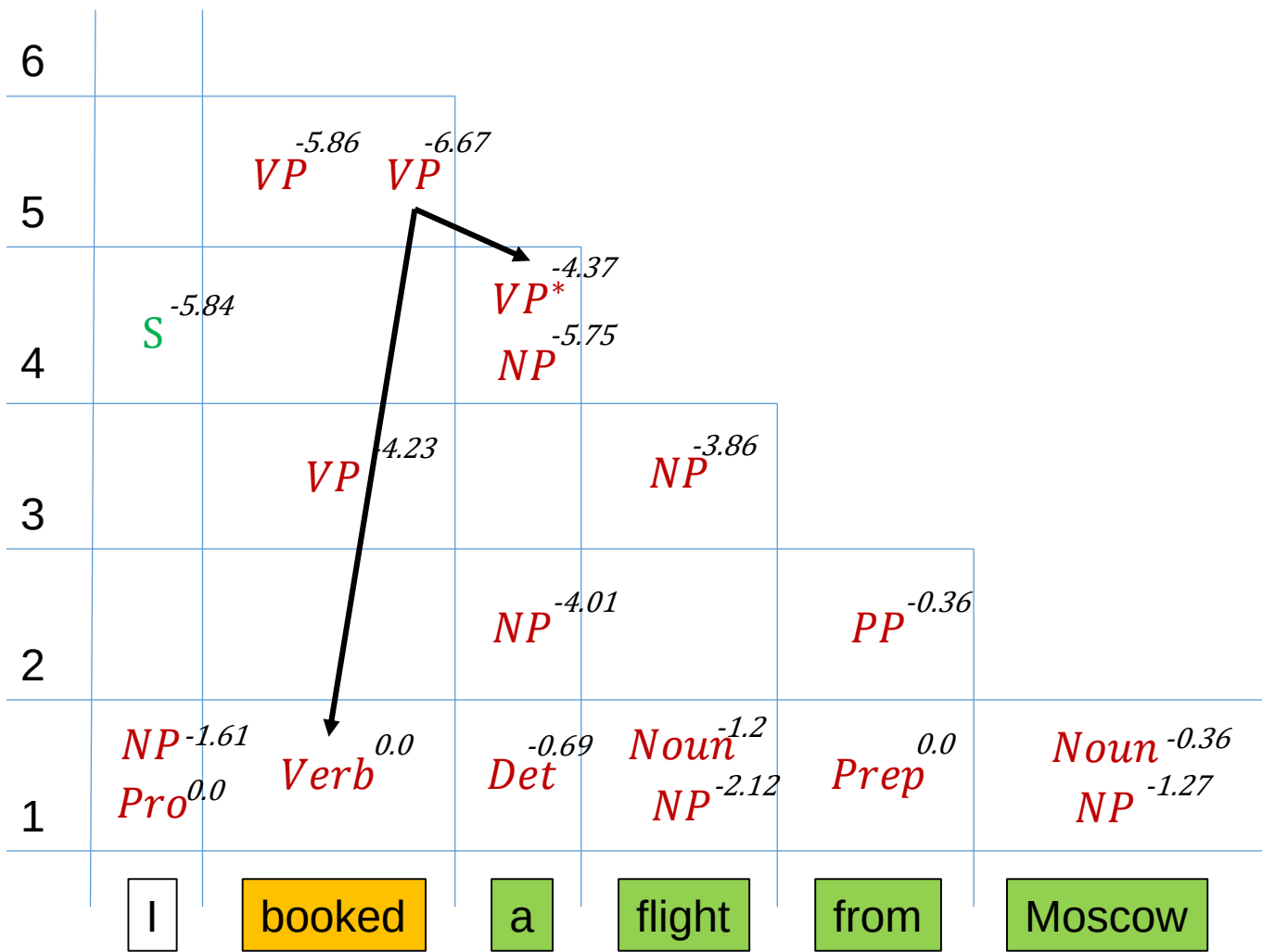
1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0

Алгоритм СΥΚ

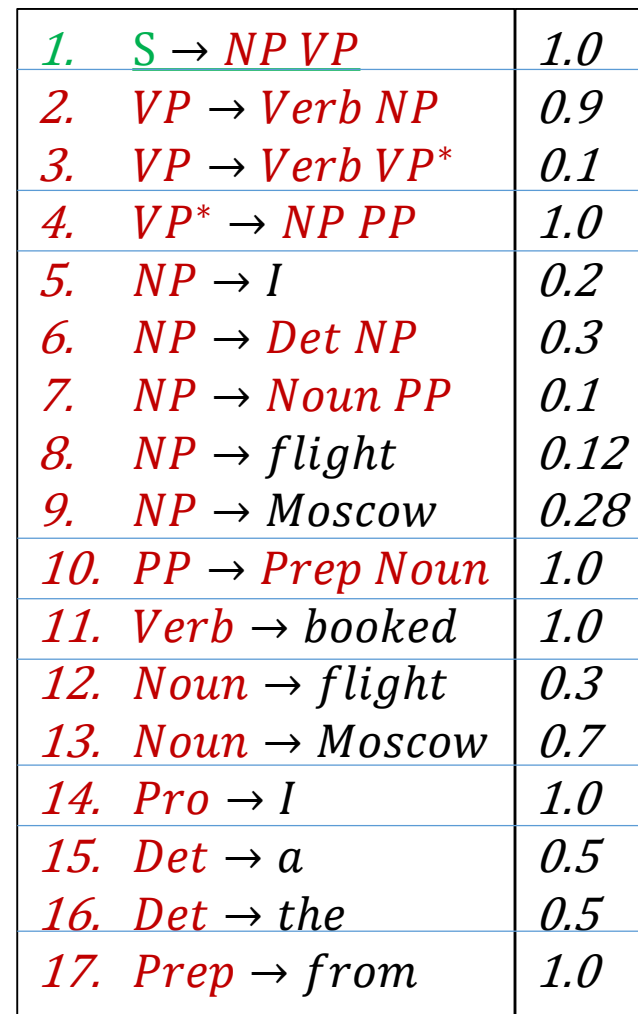


1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0

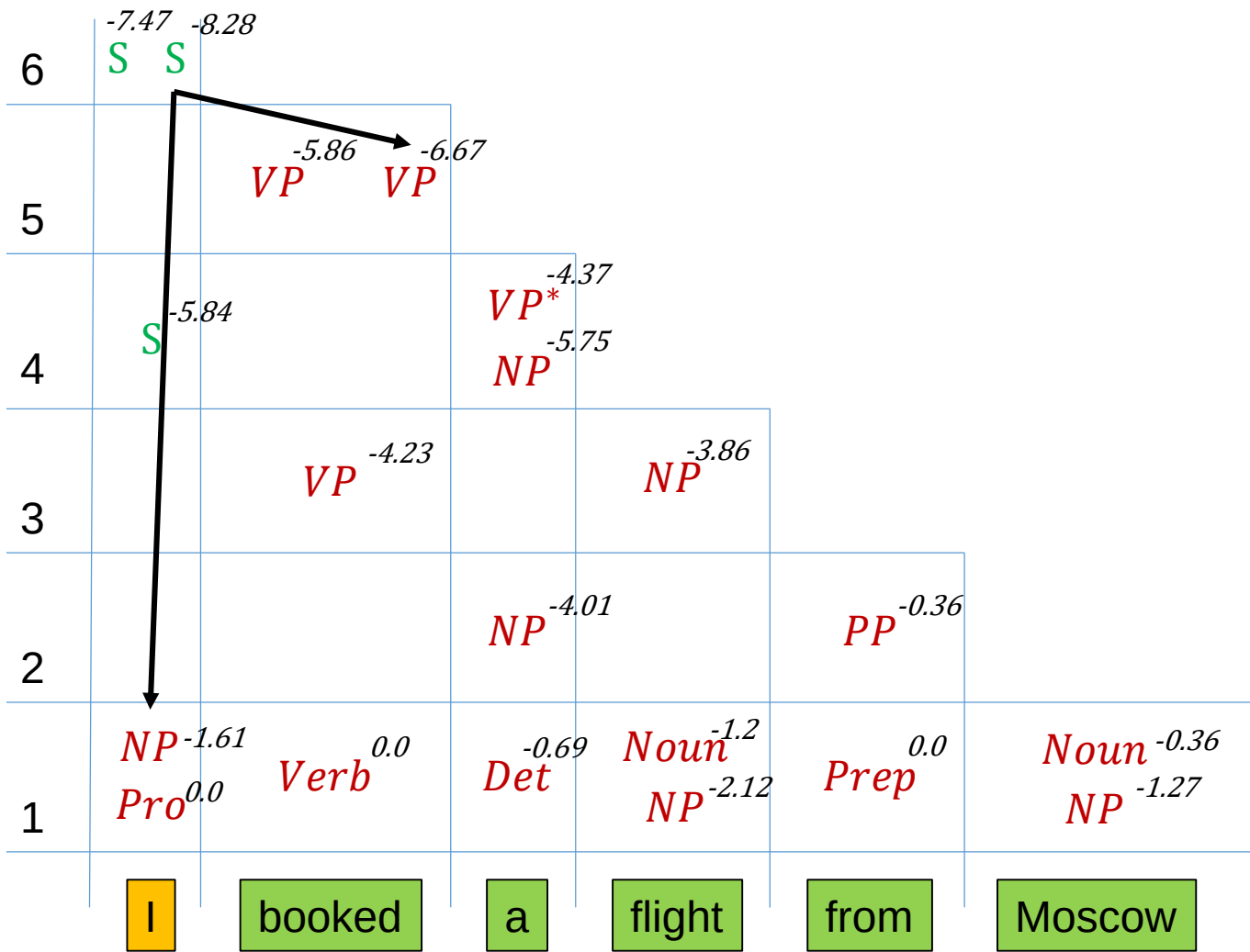
Алгоритм СΥΚ



1.	$S \rightarrow NP VP$	1.0
2.	$VP \rightarrow Verb NP$	0.9
3.	$VP \rightarrow Verb VP^*$	0.1
4.	$VP^* \rightarrow NP PP$	1.0
5.	$NP \rightarrow I$	0.2
6.	$NP \rightarrow Det NP$	0.3
7.	$NP \rightarrow Noun PP$	0.1
8.	$NP \rightarrow flight$	0.12
9.	$NP \rightarrow Moscow$	0.28
10.	$PP \rightarrow Prep Noun$	1.0
11.	$Verb \rightarrow booked$	1.0
12.	$Noun \rightarrow flight$	0.3
13.	$Noun \rightarrow Moscow$	0.7
14.	$Pro \rightarrow I$	1.0
15.	$Det \rightarrow a$	0.5
16.	$Det \rightarrow the$	0.5
17.	$Prep \rightarrow from$	1.0



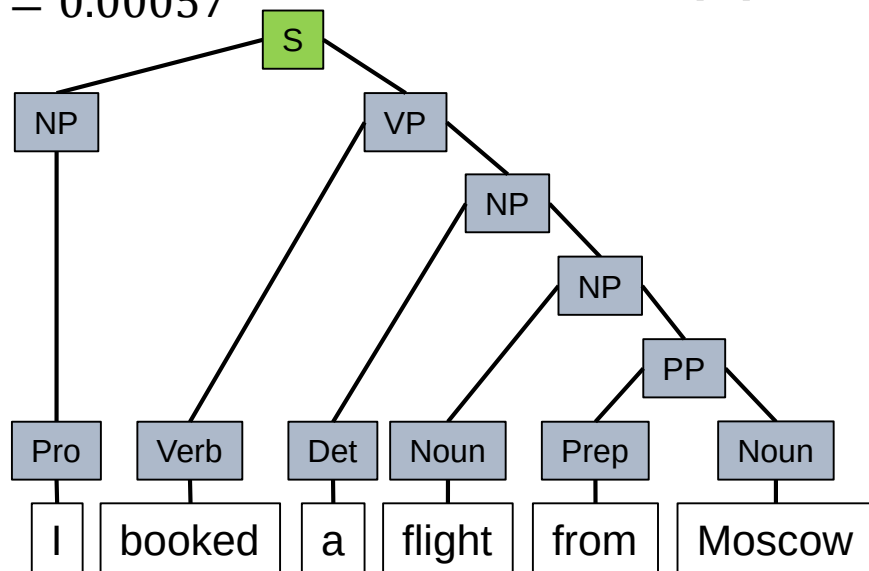
Алгоритм СҮК



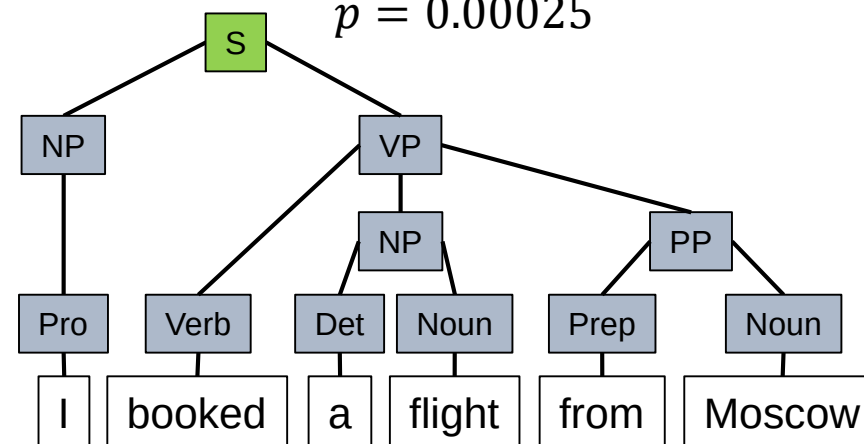
1.	<i>S</i> → <i>NP VP</i>	1.0
2.	<i>VP</i> → <i>Verb NP</i>	0.9
3.	<i>VP</i> → <i>Verb VP*</i>	0.1
4.	<i>VP*</i> → <i>NP PP</i>	1.0
5.	<i>NP</i> → <i>I</i>	0.2
6.	<i>NP</i> → <i>Det NP</i>	0.3
7.	<i>NP</i> → <i>Noun PP</i>	0.1
8.	<i>NP</i> → <i>flight</i>	0.12
9.	<i>NP</i> → <i>Moscow</i>	0.28
10.	<i>PP</i> → <i>Prep Noun</i>	1.0
11.	<i>Verb</i> → <i>booked</i>	1.0
12.	<i>Noun</i> → <i>flight</i>	0.3
13.	<i>Noun</i> → <i>Moscow</i>	0.7
14.	<i>Pro</i> → <i>I</i>	1.0
15.	<i>Det</i> → <i>a</i>	0.5
16.	<i>Det</i> → <i>the</i>	0.5
17.	<i>Prep</i> → <i>from</i>	1.0

Неоднозначность языка

$p = 0.00057$



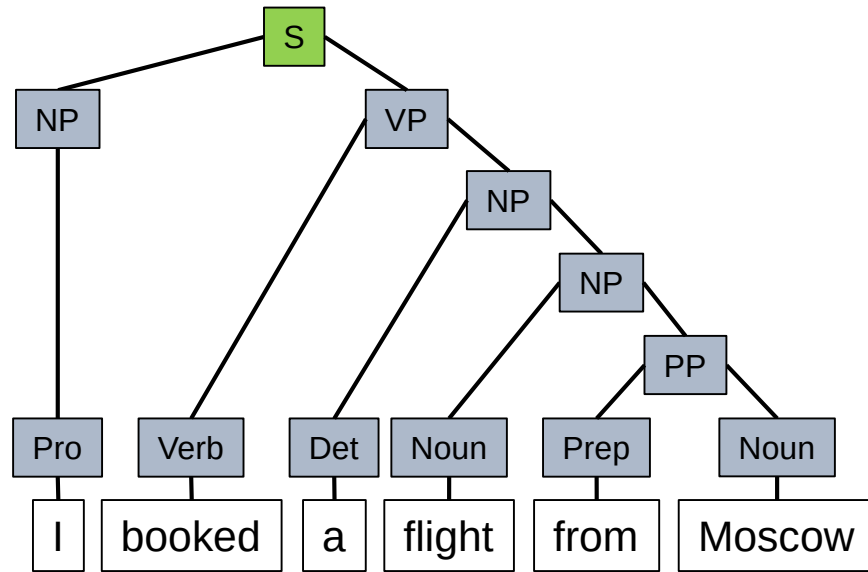
$p = 0.00025$



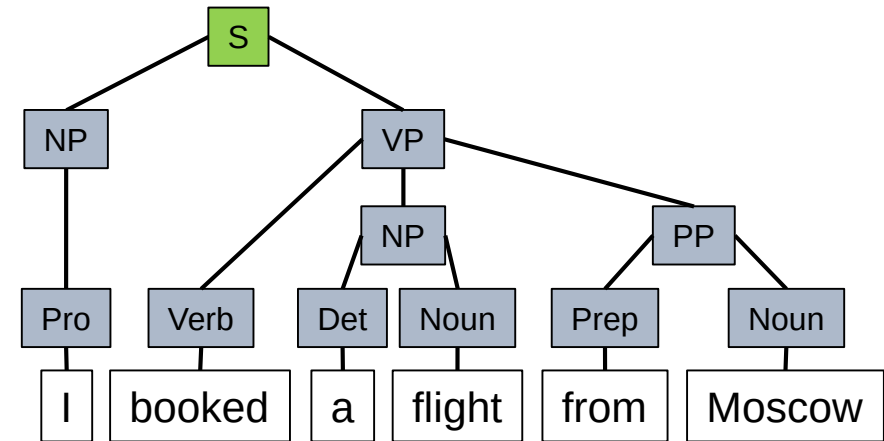
- | | |
|--------------------------------|-------------------------------|
| 1. $S \rightarrow NP VP$ | 10. $Verb \rightarrow booked$ |
| 2. $VP \rightarrow Verb NP$ | 11. $Noun \rightarrow Moscow$ |
| 3. $VP \rightarrow Verb NP PP$ | 12. $Pro \rightarrow I$ |
| 4. $NP \rightarrow Pro$ | 13. $Det \rightarrow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

Оценка качества

Gold:

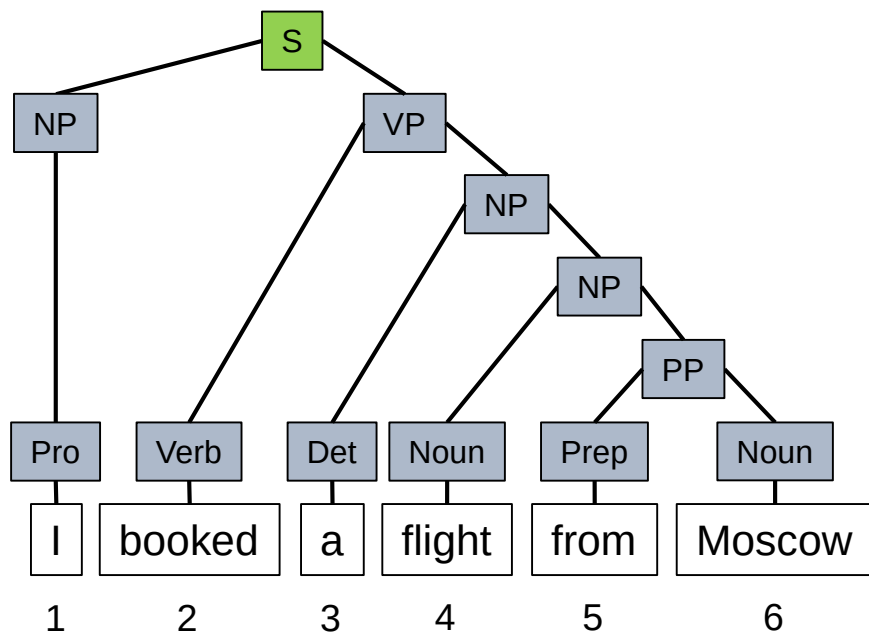


Predicted:

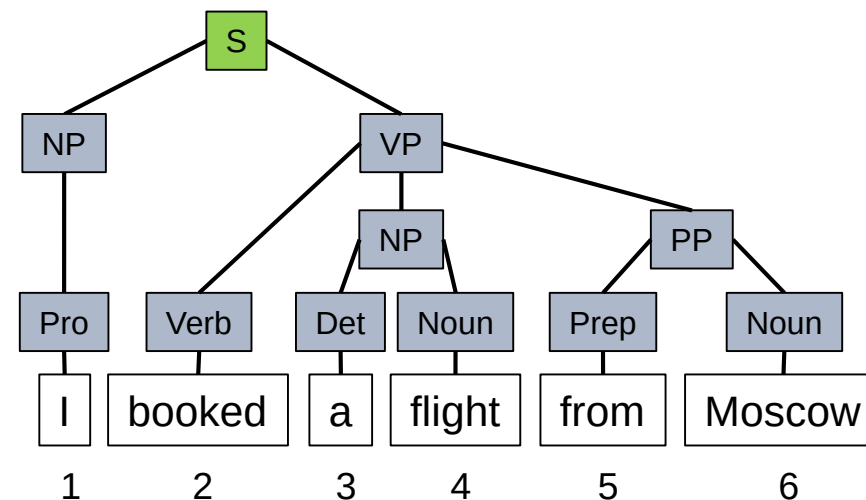


Оценка качества

Gold:



Predicted:



- Gold:
S(1:6), NP(1:1), VP(2:6), NP(3:6), NP(4:6), PP(5:6)
- Predicted:
S(1:6), NP(1:1), VP(2:6), NP(3:4), PP(5:6)

Оценка качества

- $Recall = \frac{|correct|}{|gold|}$;
 - $Precision = \frac{|correct|}{|predicted|}$;
 - $F1 = \frac{2 * Precision * Recall}{Precision + Recall}$;
- Gold:
S(1:6), NP(1:1), VP(2:6), NP(3:6), NP(4:6), PP(5:6)
 - Predicted:
S(1:6), NP(1:1), VP(2:6), NP(3:4), PP(5:6)

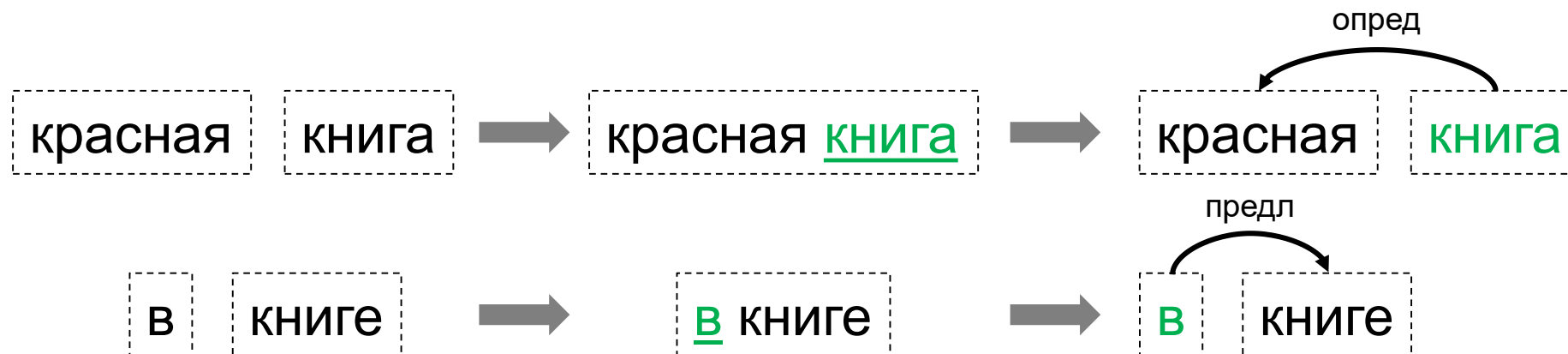
$$Recall = \frac{2}{3}$$

$$Precision = \frac{4}{5}$$

$$F1 = \frac{8}{11}$$

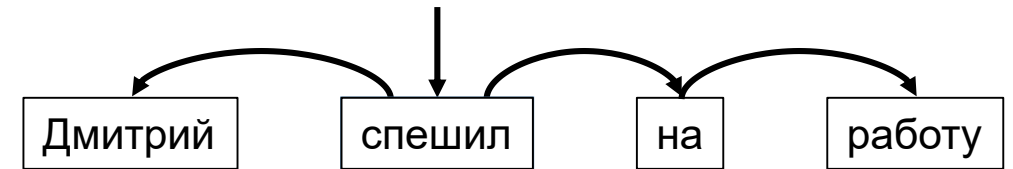
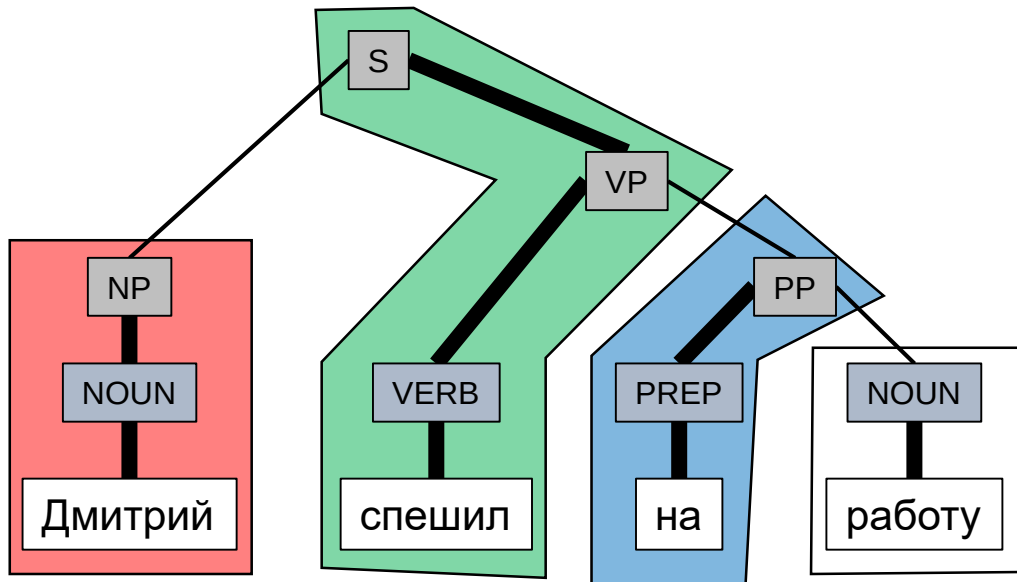
Грамматика зависимостей

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



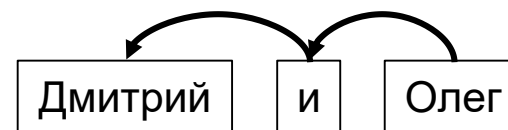
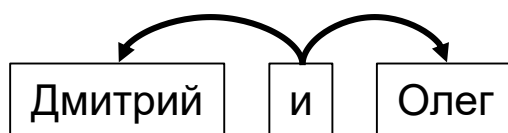
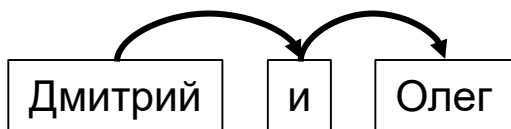
Грамматика зависимостей

- Дерево составляющих может быть преобразовано в дерево зависимостей

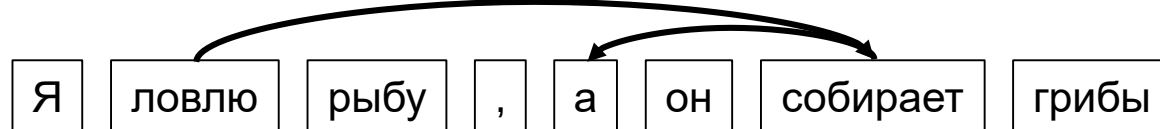
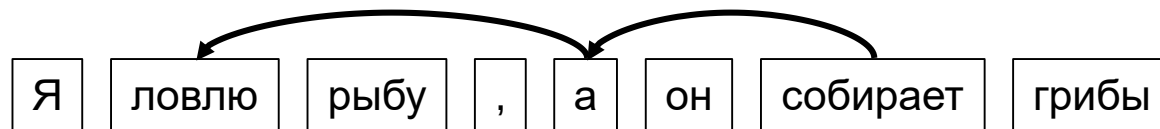
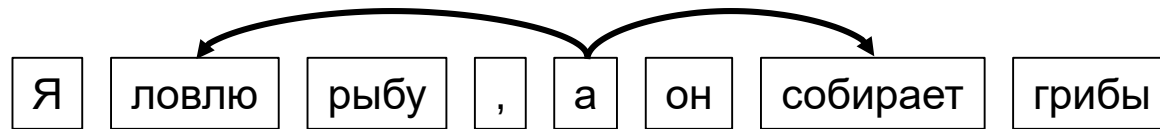
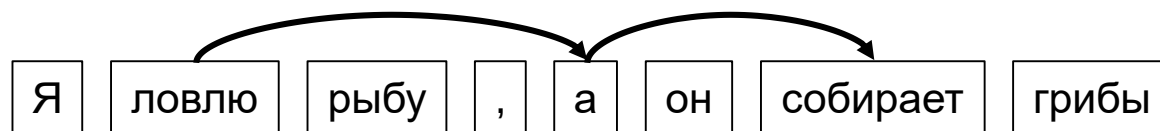


Грамматика зависимостей

- Не все синтаксические отношения подчинительные
 - Сочинительная группа

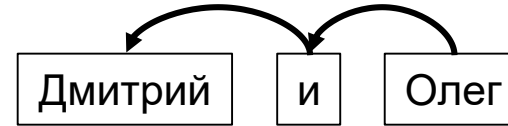
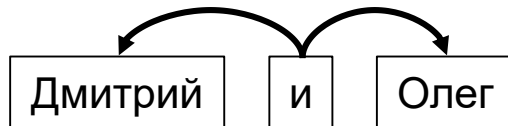
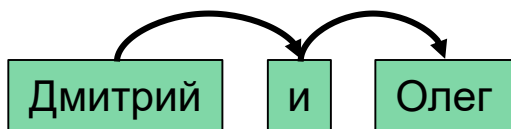


- Сложносочиненные предложения

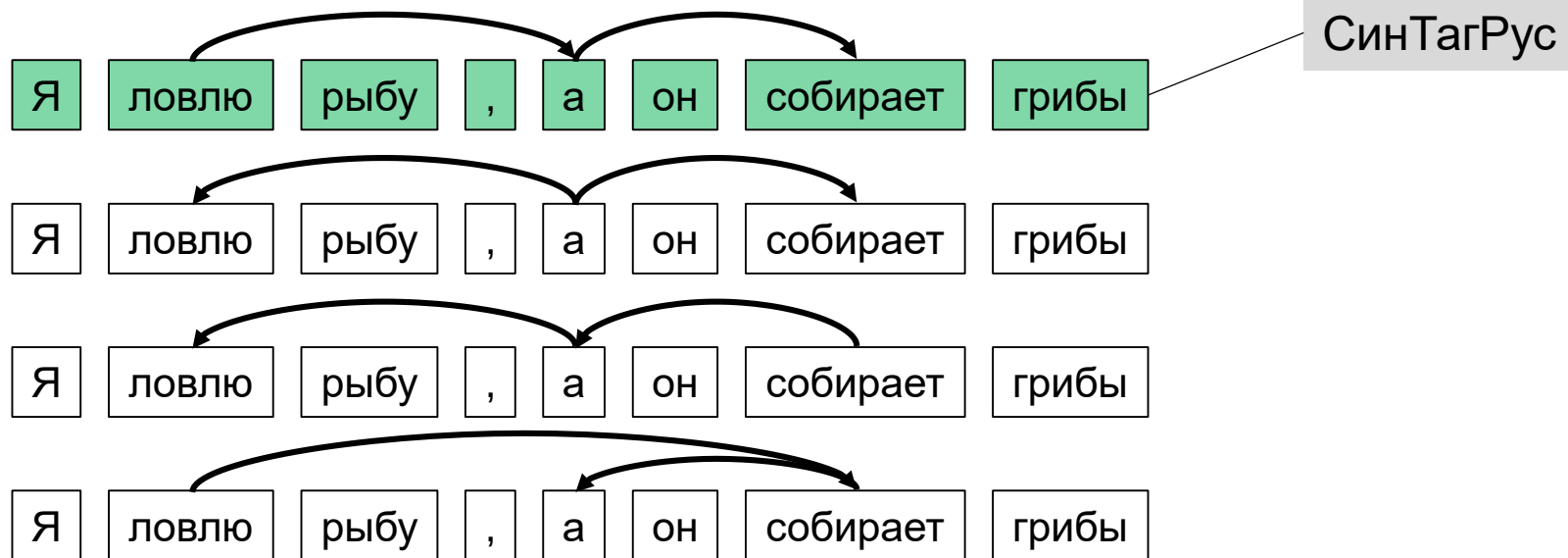


Грамматика зависимостей

- Не все синтаксические отношения подчинительные
 - Сочинительная группа

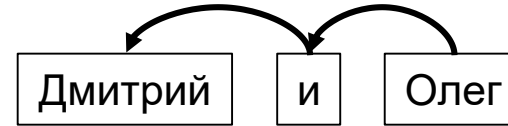
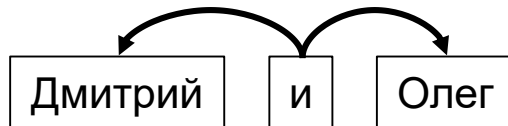
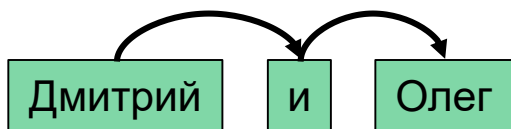


- Сложносочиненные предложения

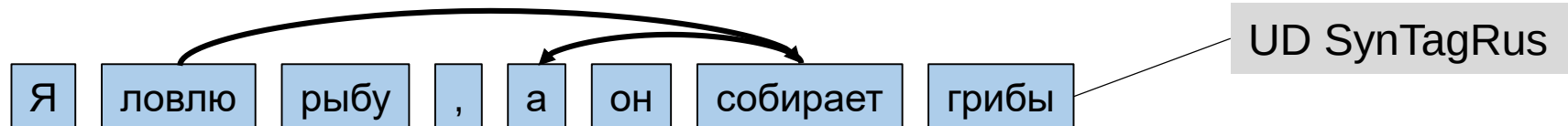
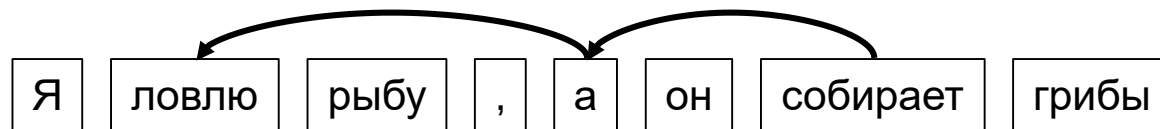
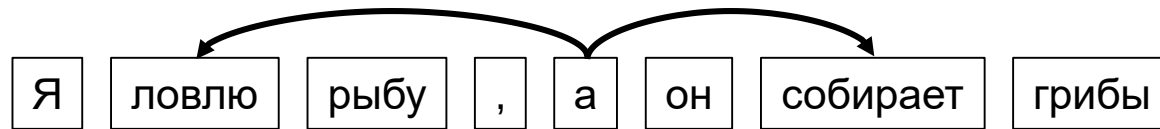
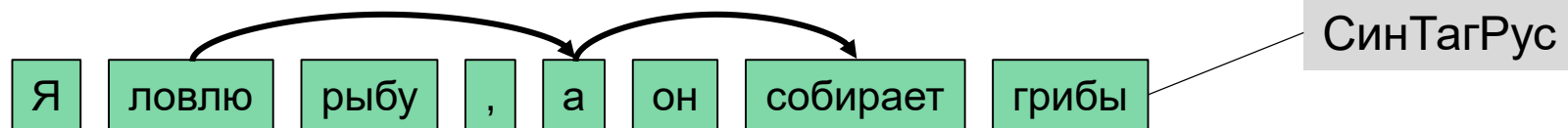


Грамматика зависимостей

- Не все синтаксические отношения подчинительные
 - Сочинительная группа



- Сложносочиненные предложения

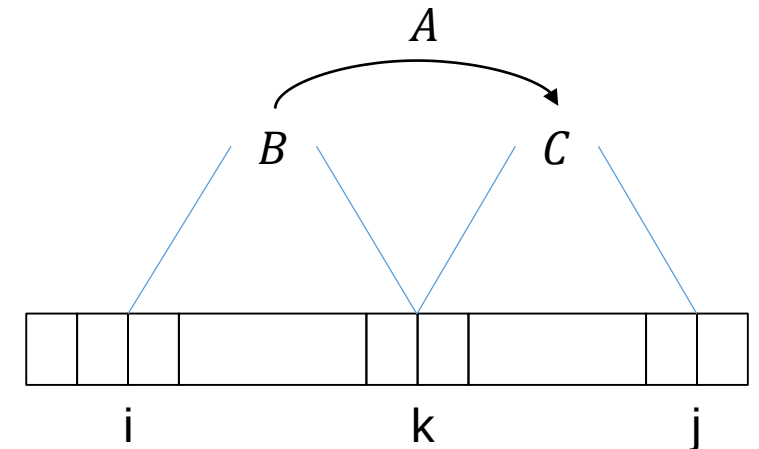


Universal Dependencies

- 141 языков
 - Универсальный набор грамматических категорий
 - Универсальный набор синтаксических отношений
- 245 наборов данных (CoNLL-U формат)
 - Часть размечена вручную
 - Часть сконвертирована из других форматов

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 $O(n^3)$



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

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

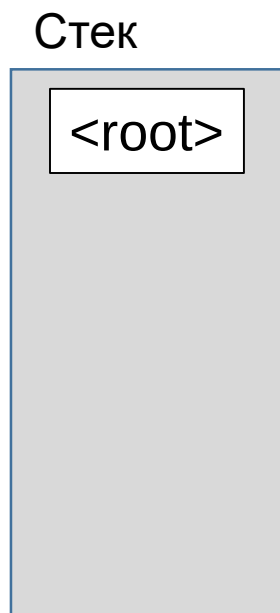
- Переходы

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

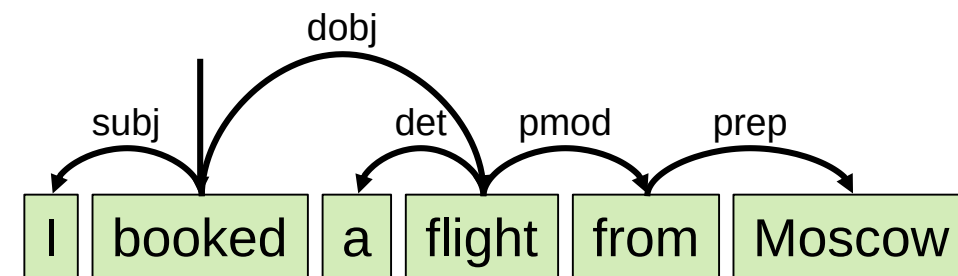
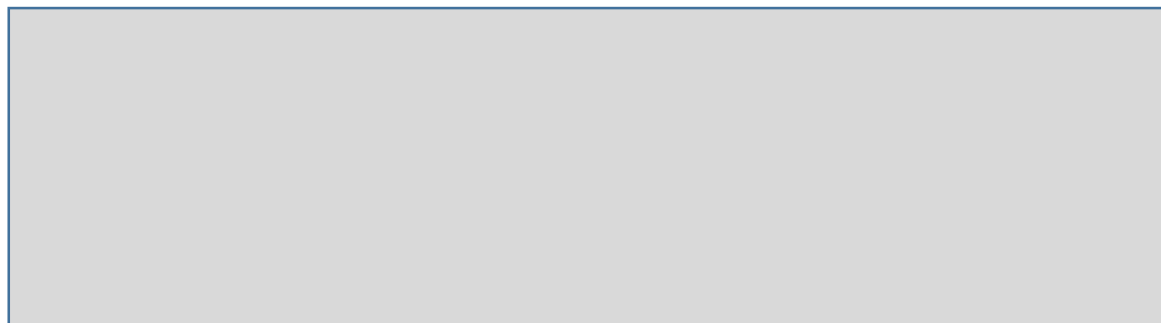
- $(LeftArc_l) \quad ([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\}), \text{ если } i \neq 0$

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

Arc-standard



Зависимости



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

Arc-standard

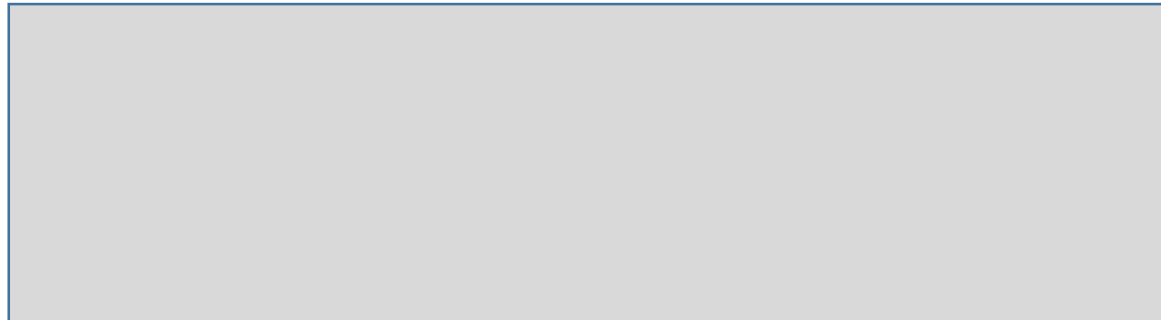
Стек

<root>

Буфер

I booked a flight from Moscow

Зависимости



Shift

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

LeftArc_l

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

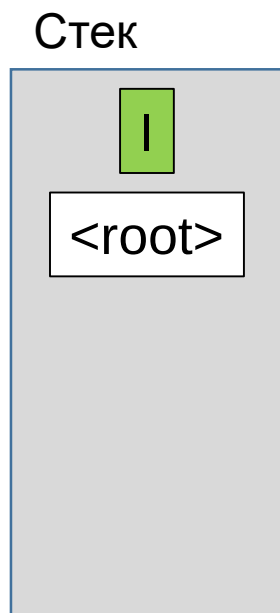
RightArc_l

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

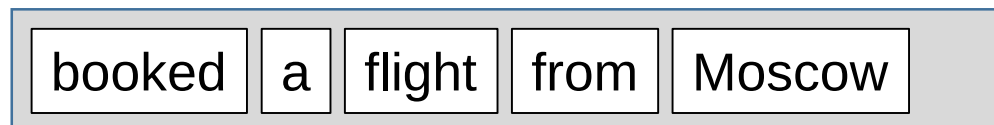
Shift

I booked a flight from Moscow

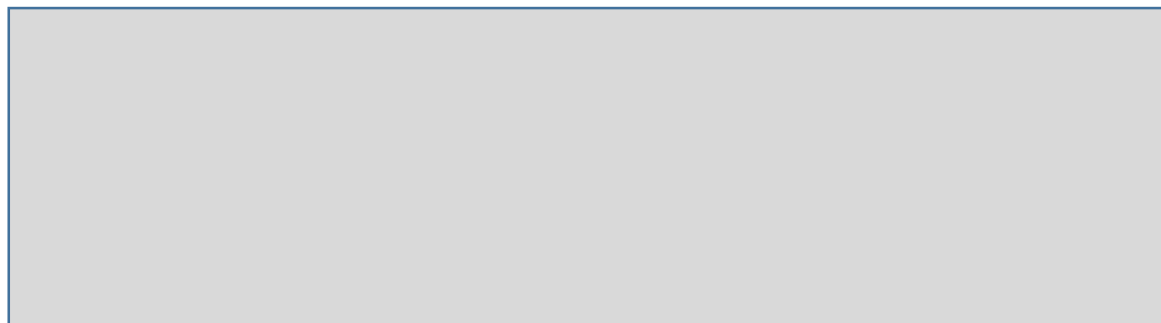
Arc-standard



Буфер



Зависимости



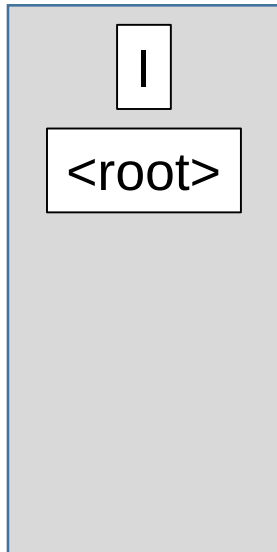
$$\text{Shift} \\ (\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$$

$$\text{LeftArc}_l \\ ([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$$

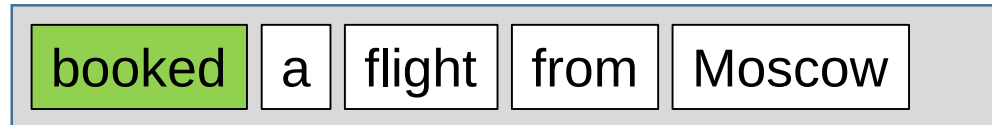
$$\text{RightArc}_l \\ ([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$$

Arc-standard

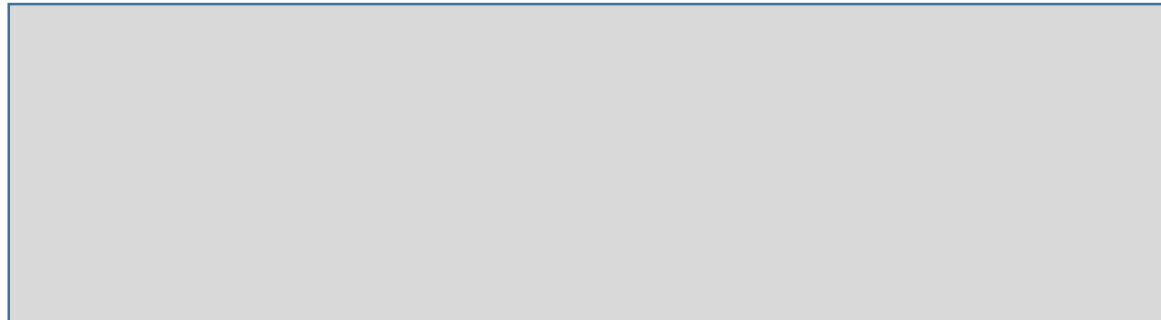
Стек



Буфер



Зависимости



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

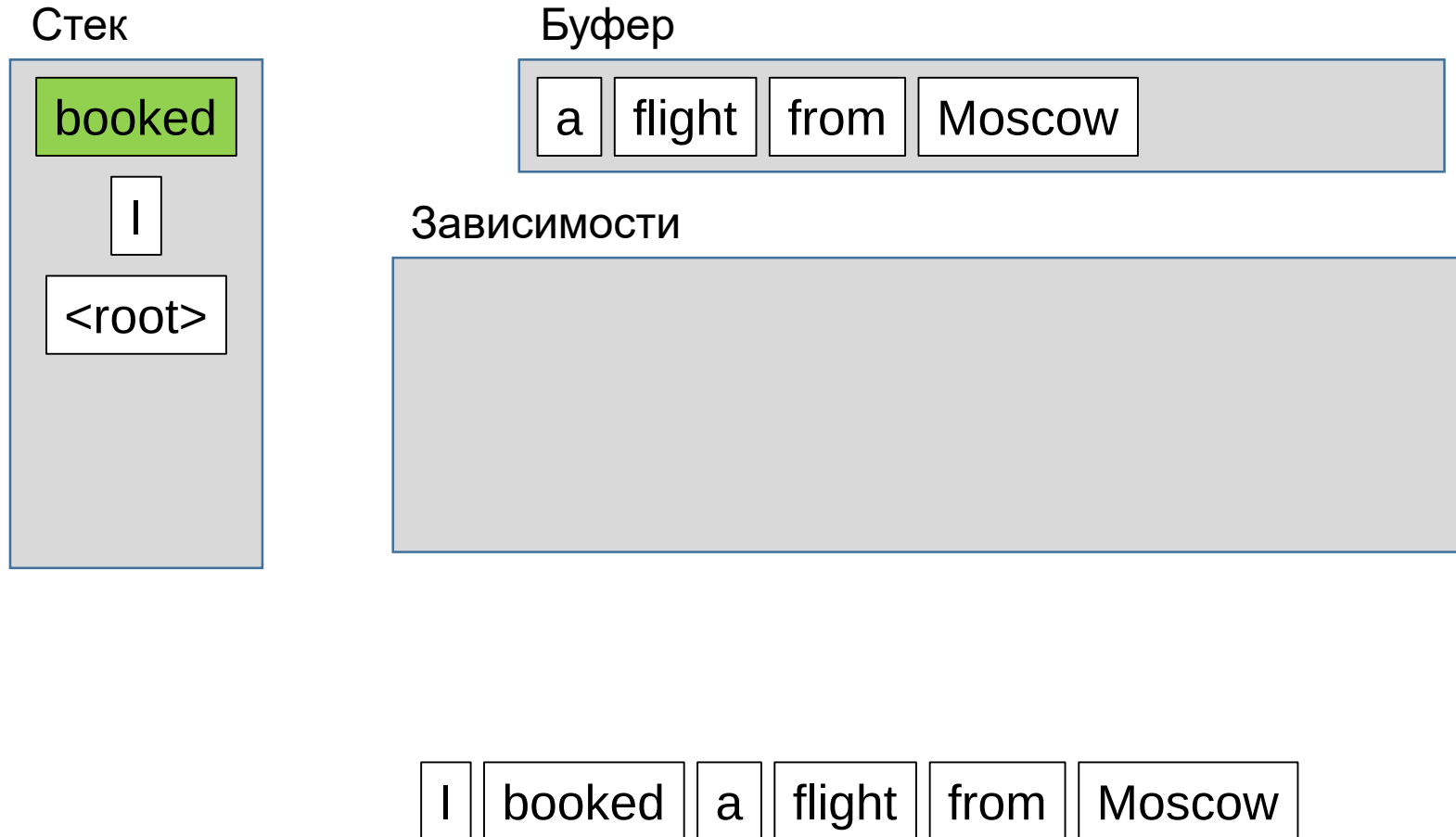
LeftArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$

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

Shift



Arc-standard



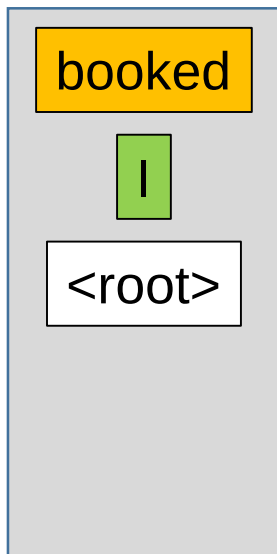
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

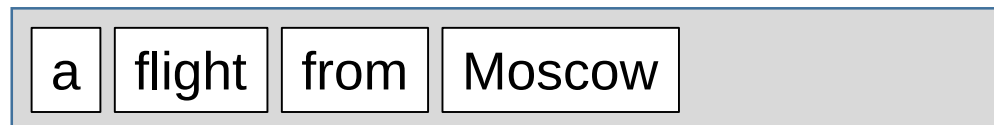
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

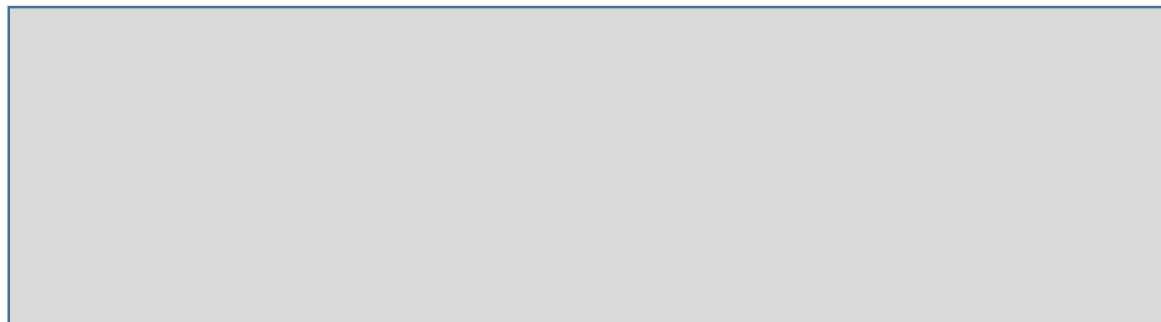
Стек



Буфер



Зависимости



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

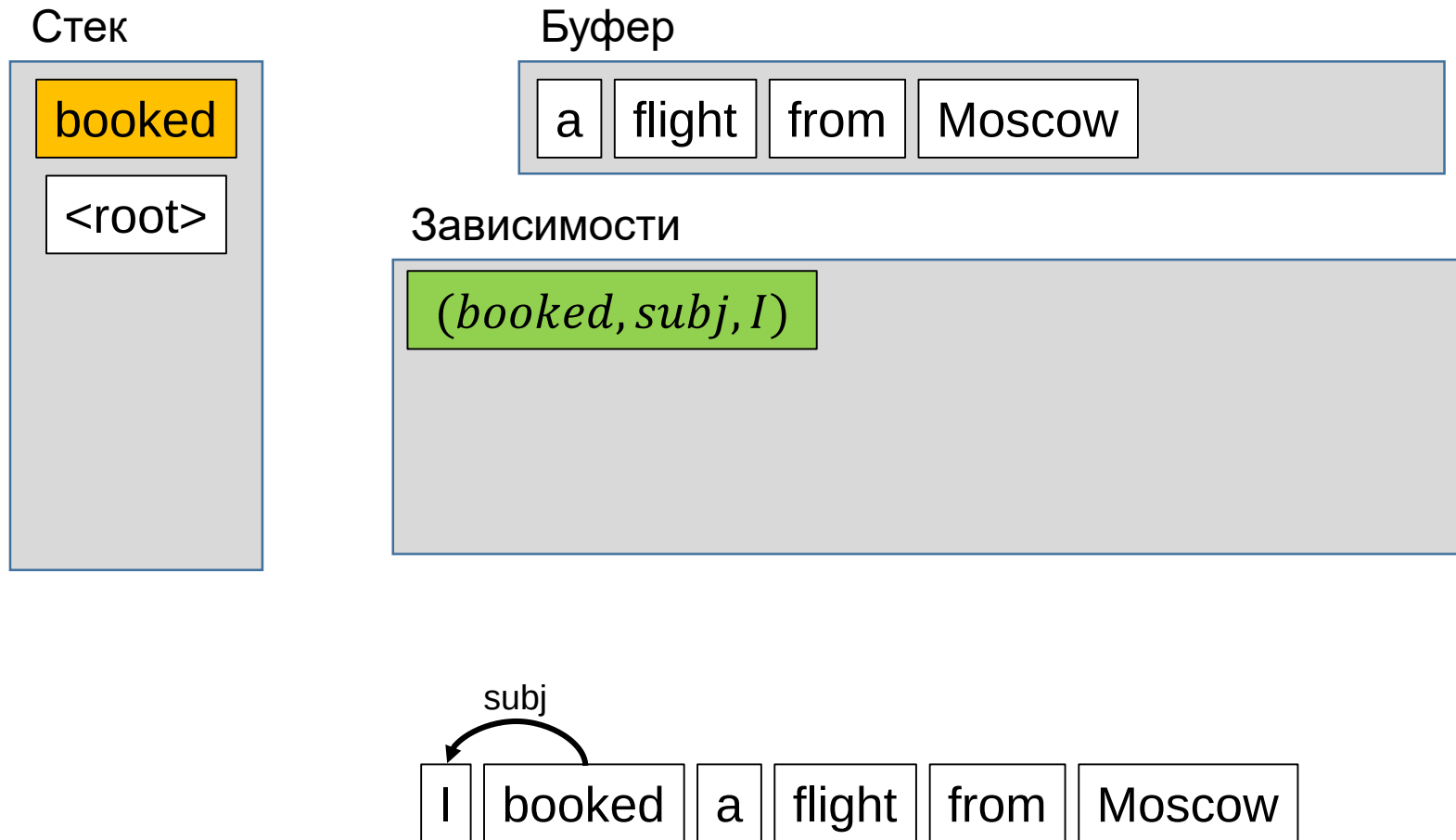
LeftArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$

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



LeftArc_{subj}

Arc-standard



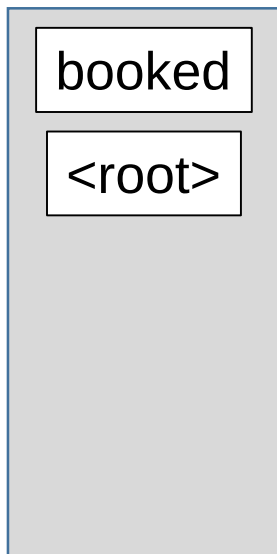
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

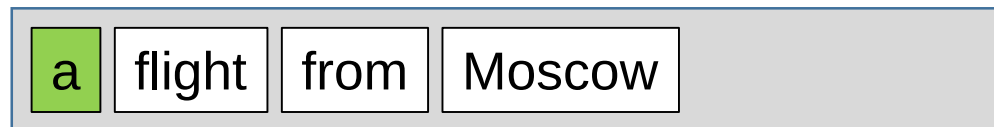
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

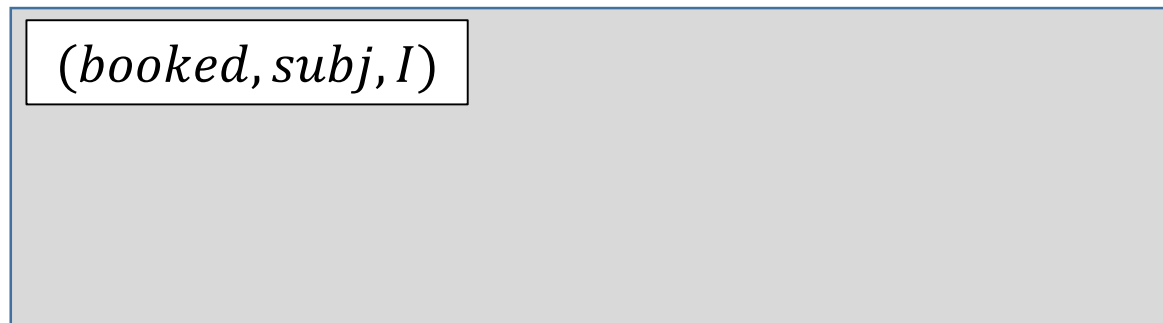
Стек



Буфер



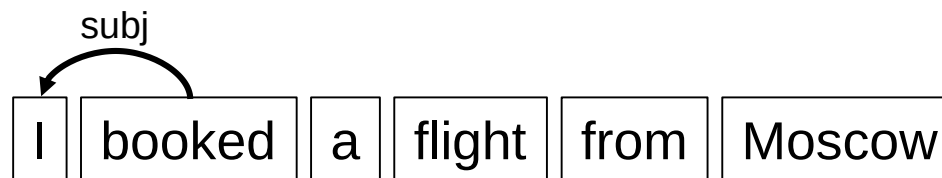
Зависимости



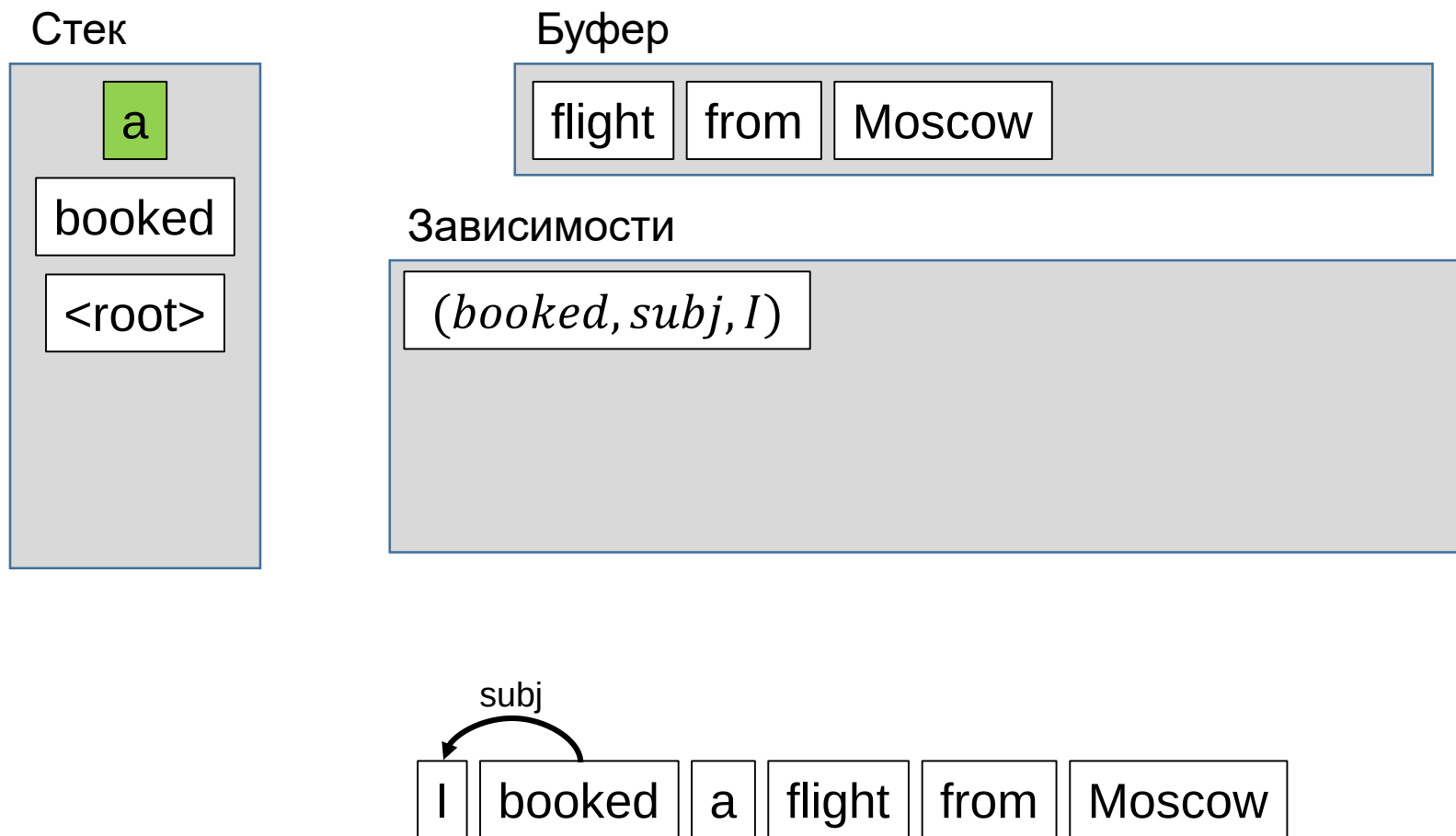
$$\textit{Shift} \\ (\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$$

$$\textit{LeftArc}_l \\ ([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$$

$$\textit{RightArc}_l \\ ([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$$



Arc-standard



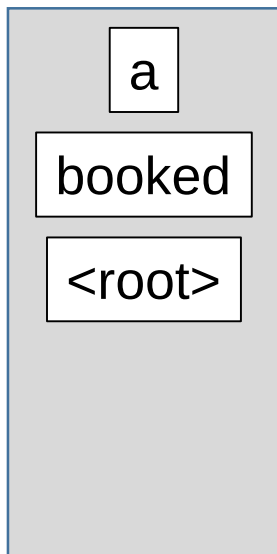
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

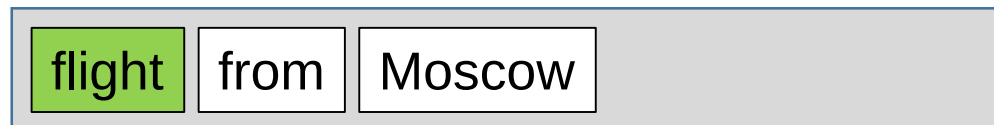
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

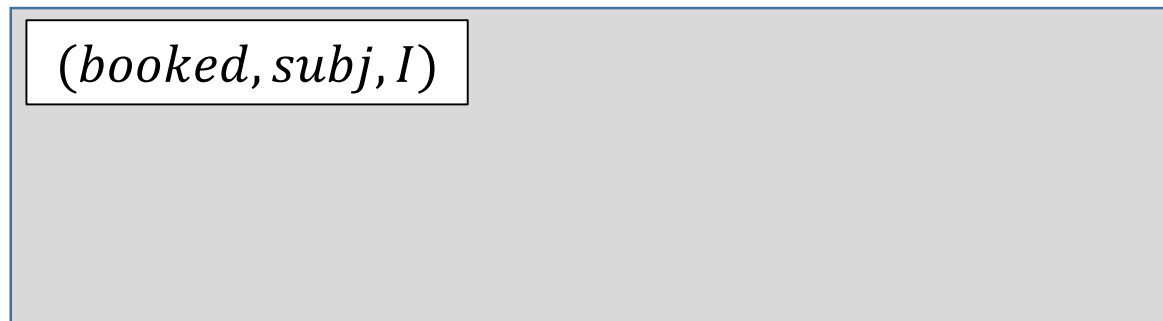
Стек



Буфер



Зависимости

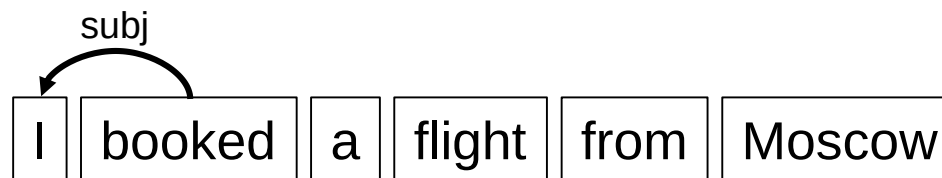


Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

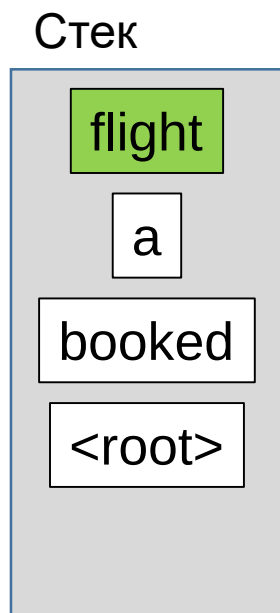
LeftArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$

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

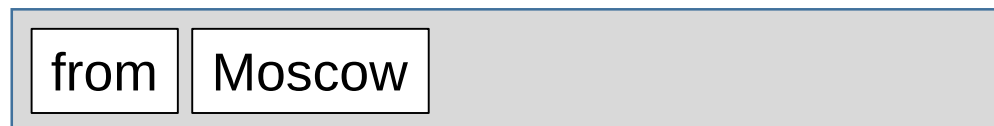
Shift



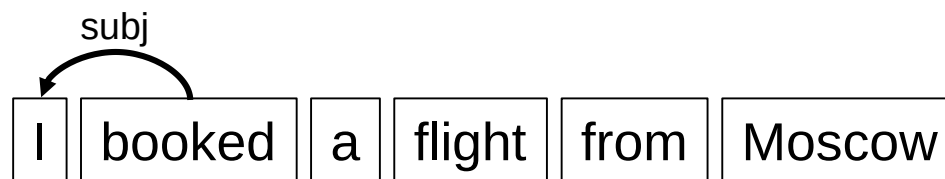
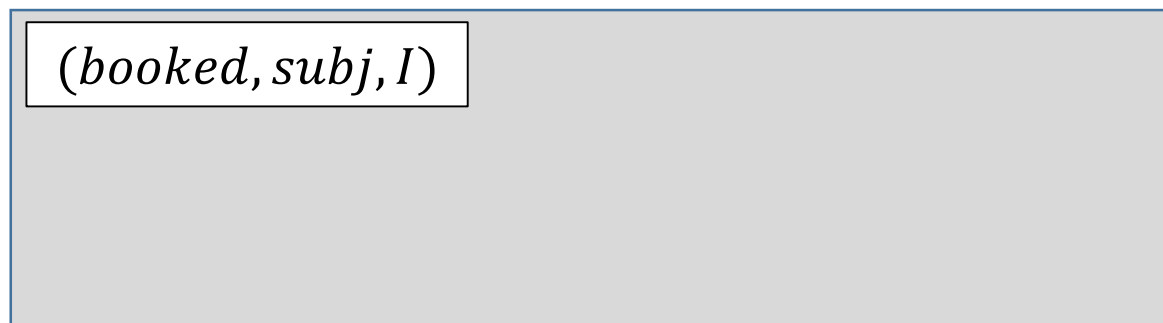
Arc-standard



Буфер



Зависимости



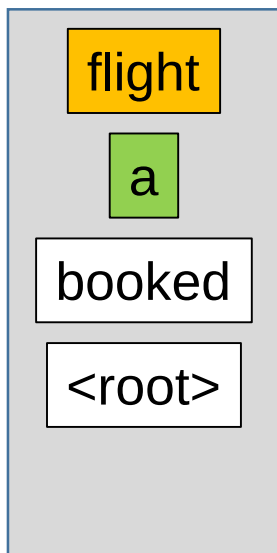
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

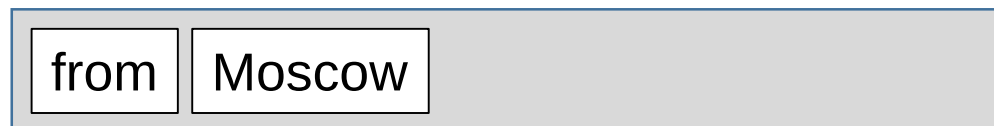
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

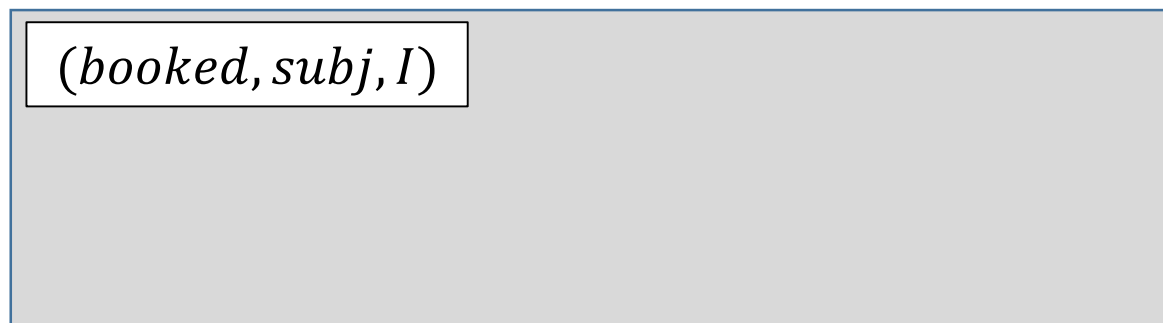
Стек



Буфер



Зависимости



subj



LeftArc_{det}

Shift

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

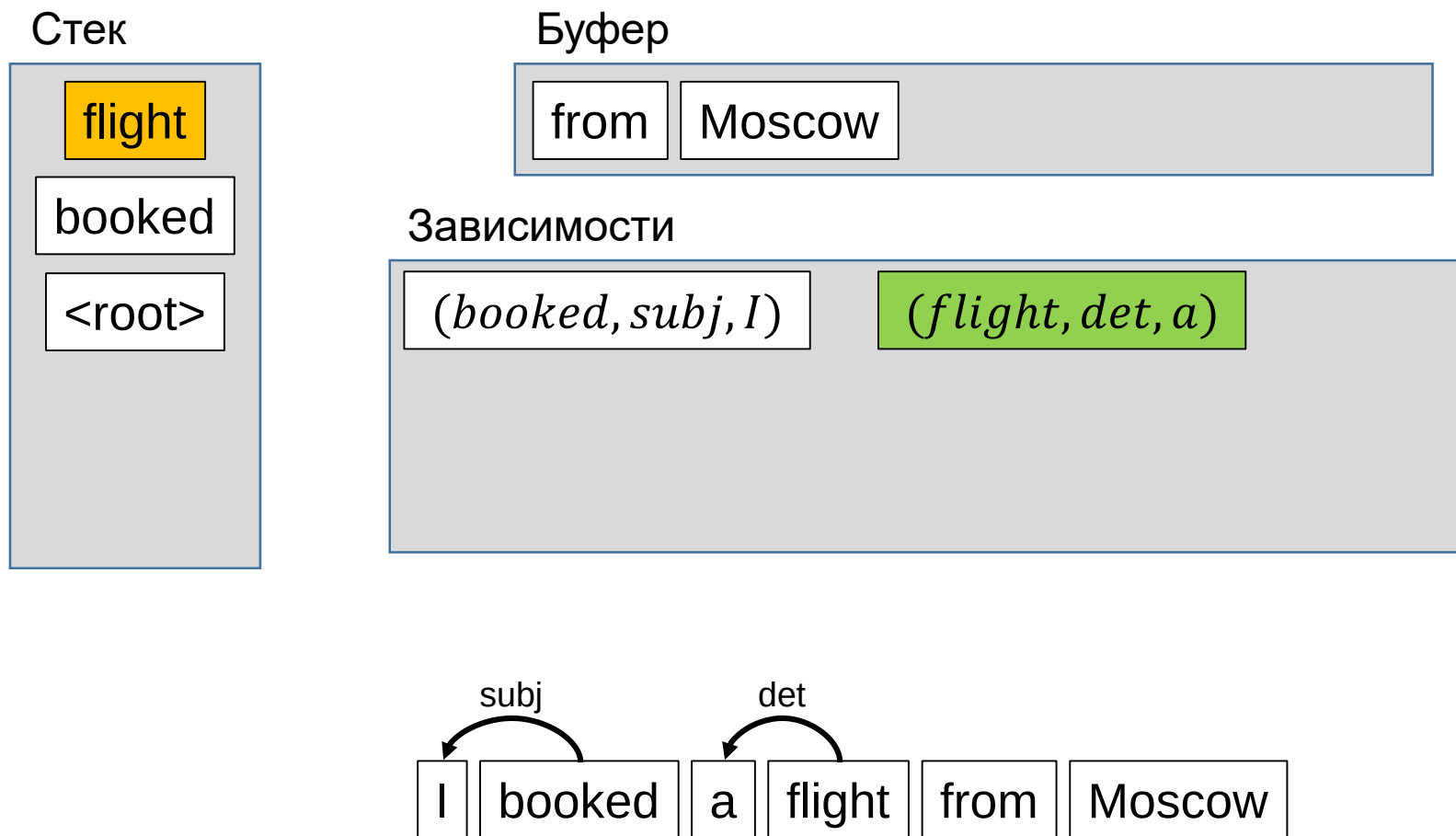
LeftArc_l

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

RightArc_l

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

Arc-standard



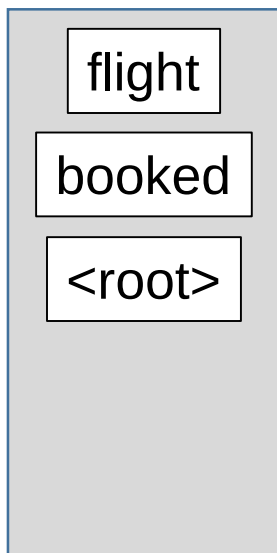
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

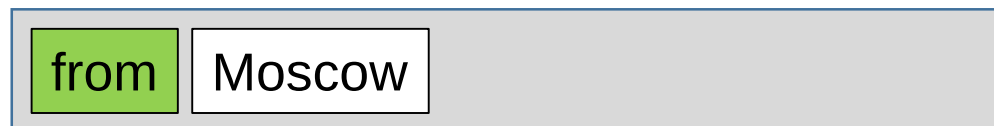
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

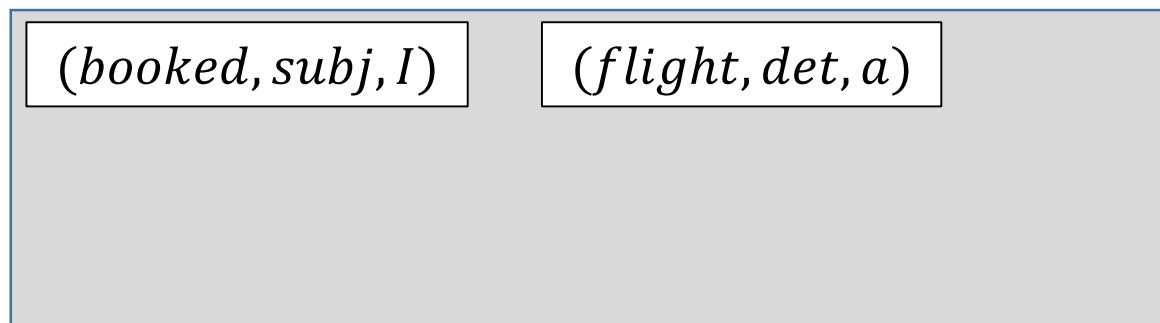
Стек



Буфер



Зависимости

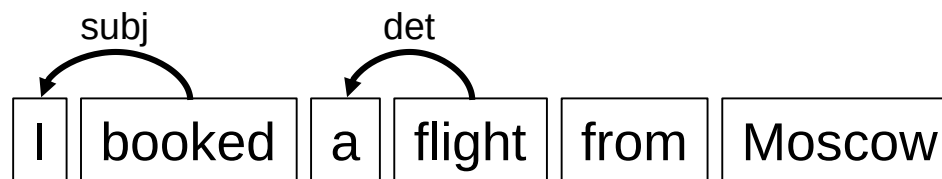


Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

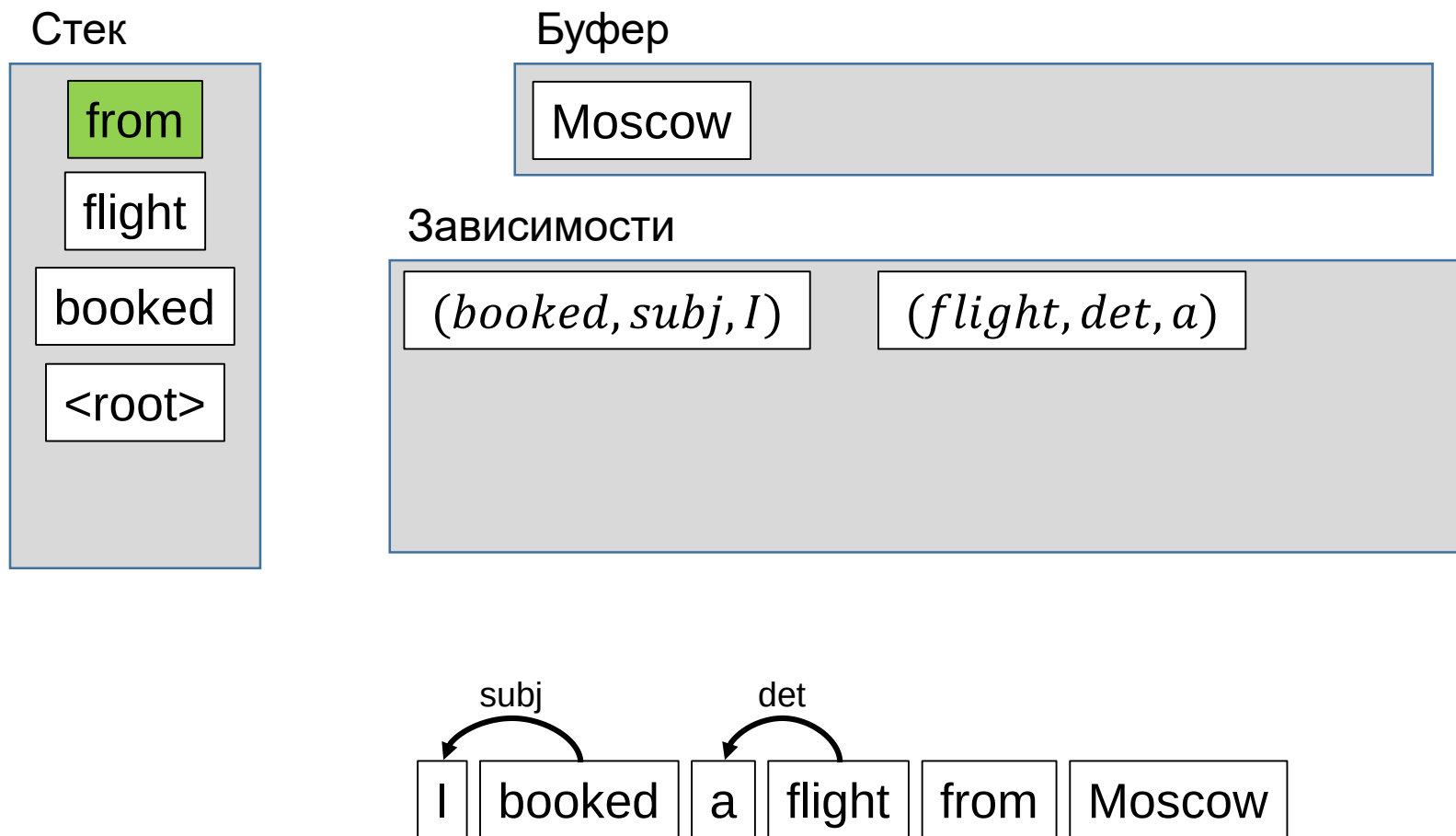
LeftArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$

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

Shift



Arc-standard



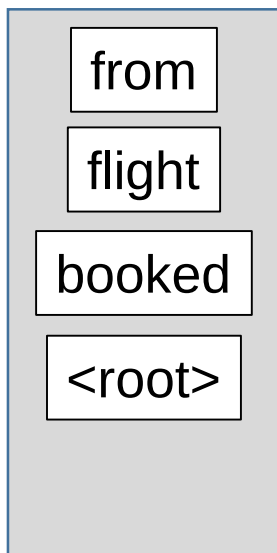
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

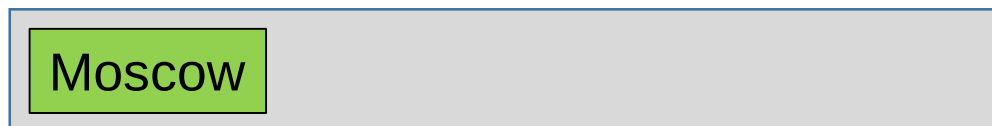
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

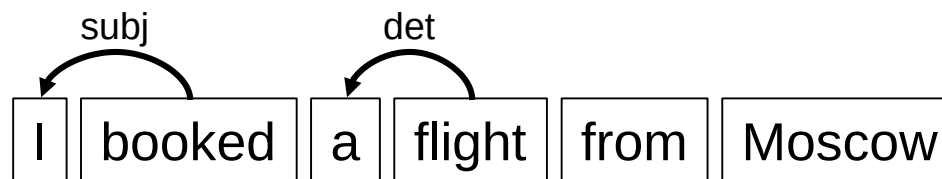
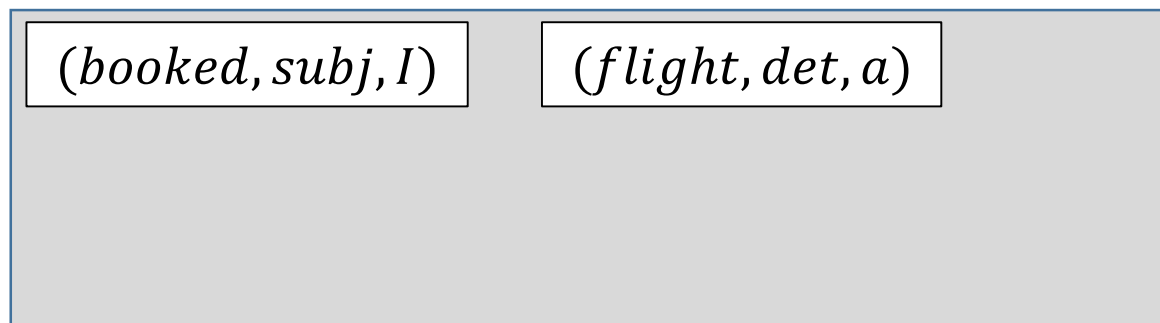
Стек



Буфер



Зависимости



Shift

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

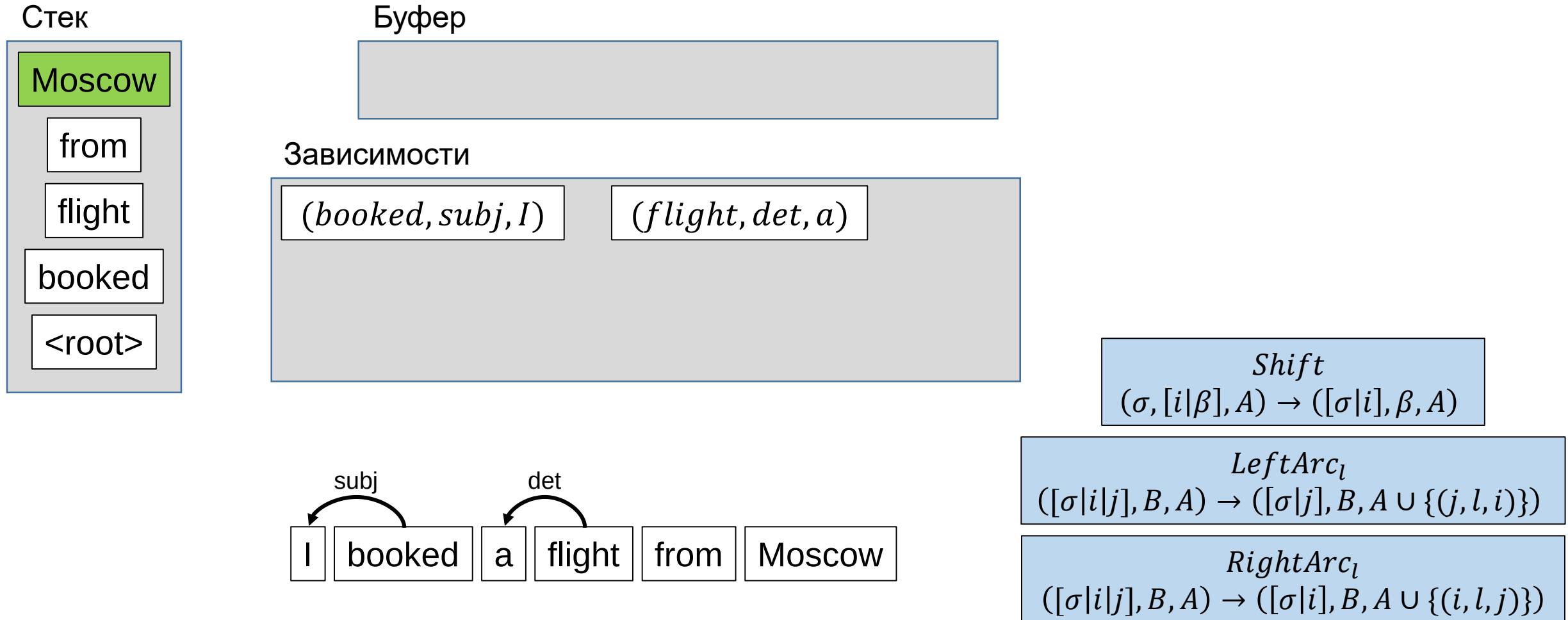
LeftArc_l

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

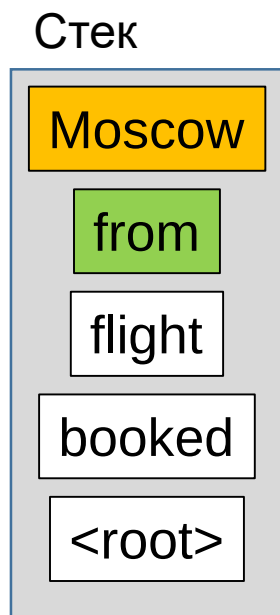
RightArc_l

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

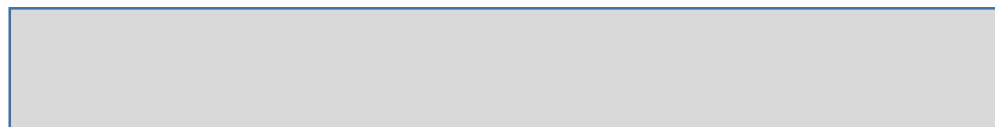
Arc-standard



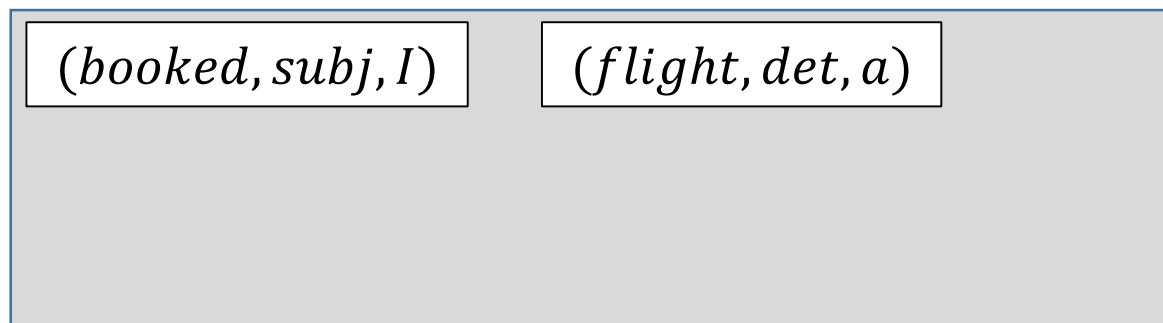
Arc-standard



Буфер



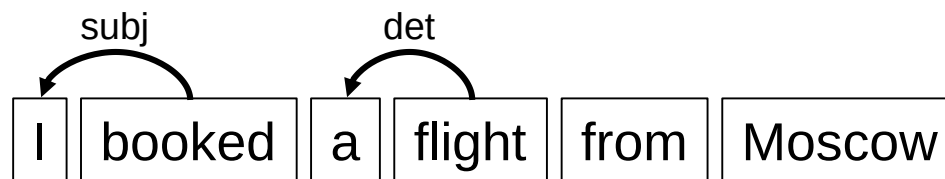
Зависимости



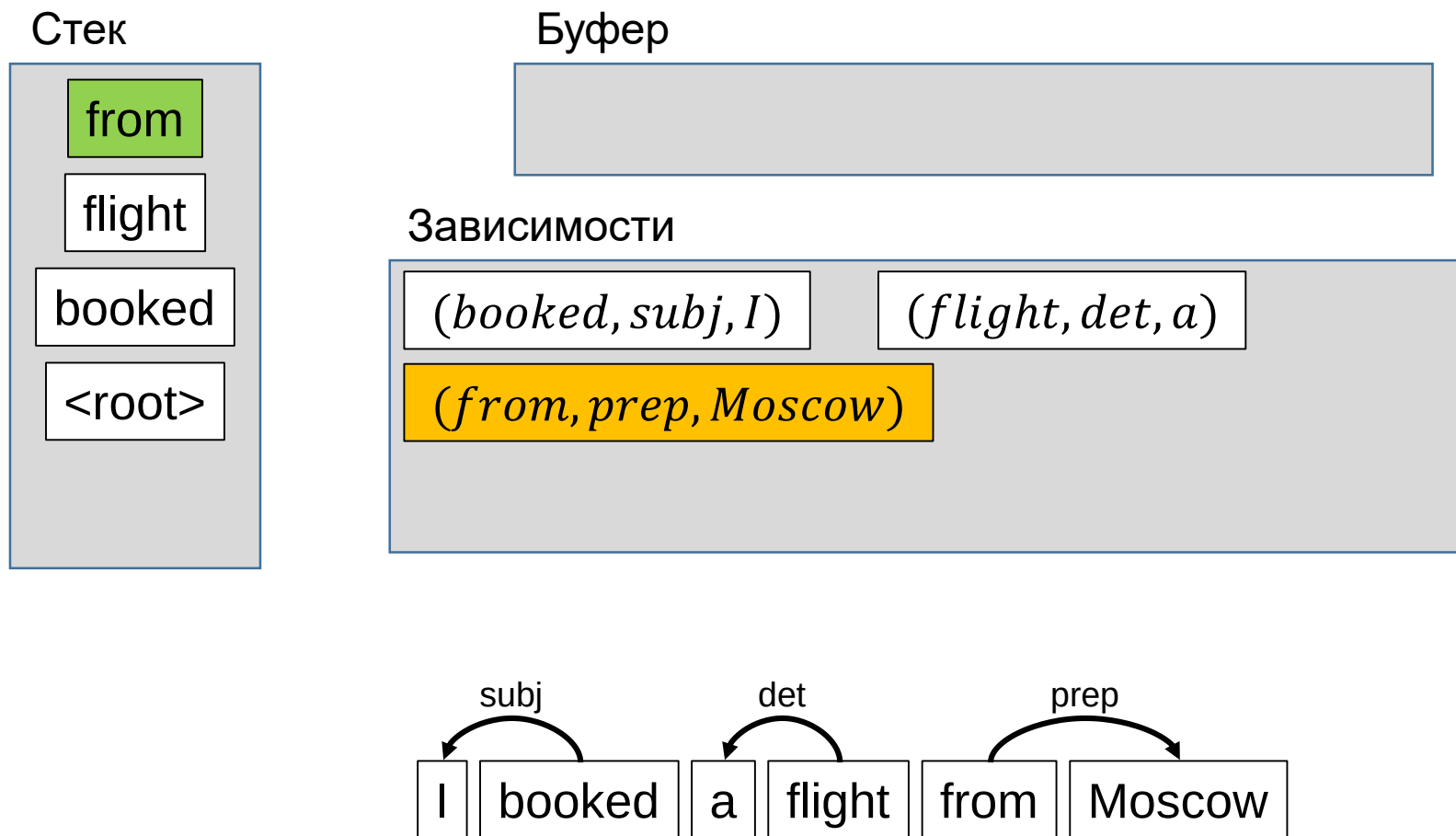
$$\text{Shift} \\ (\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$$

$$\text{LeftArc}_l \\ ([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$$

$$\text{RightArc}_l \\ ([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$$



Arc-standard



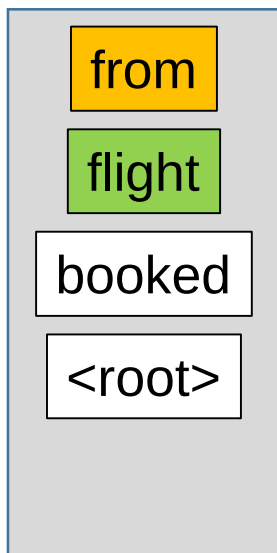
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

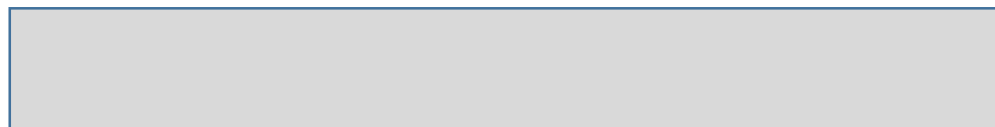
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

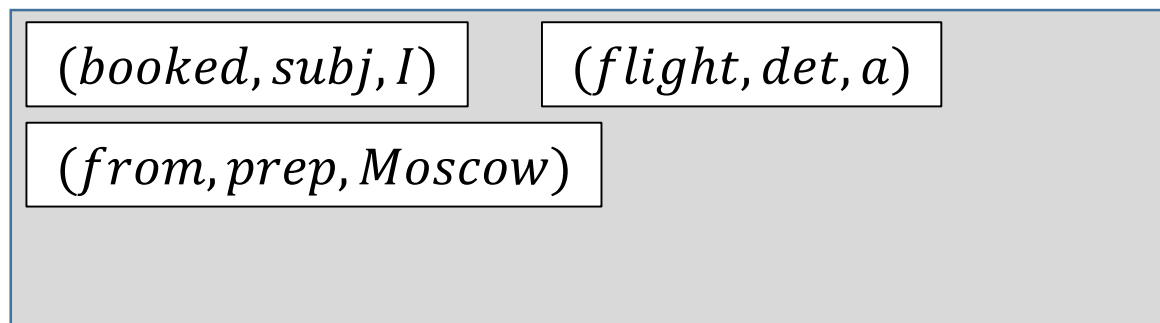
Стек



Буфер



Зависимости



Shift

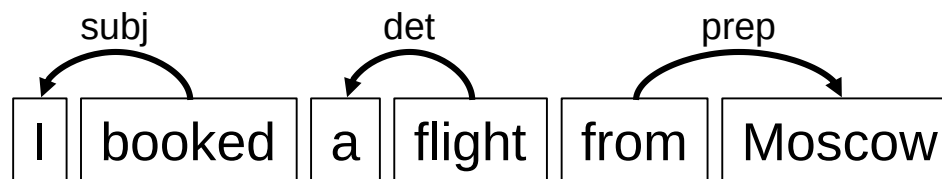
$(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

LeftArc_l

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

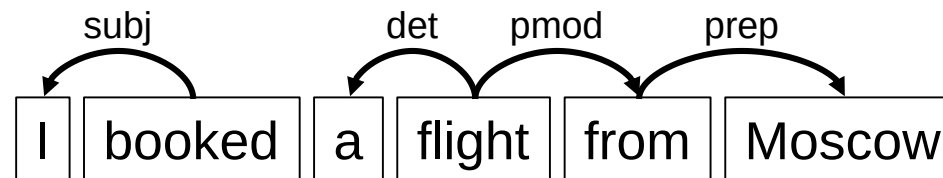
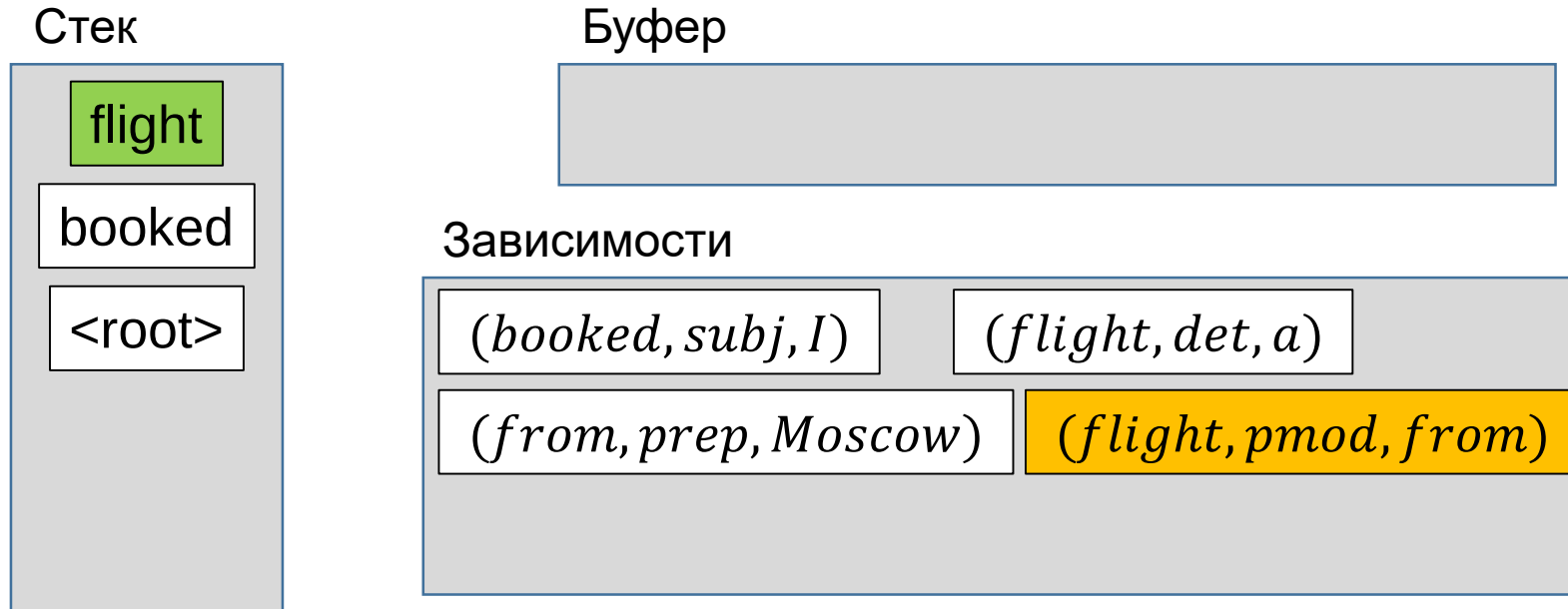
RightArc_l

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



RightArc_{pmod}

Arc-standard



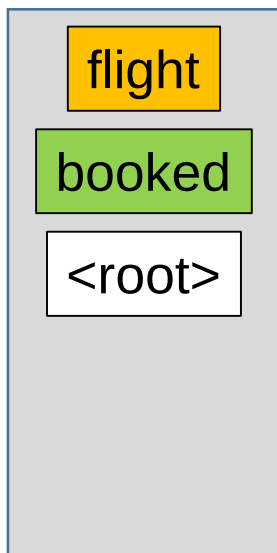
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

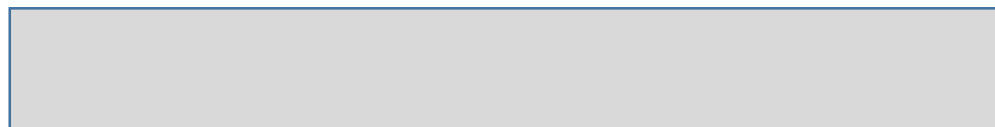
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

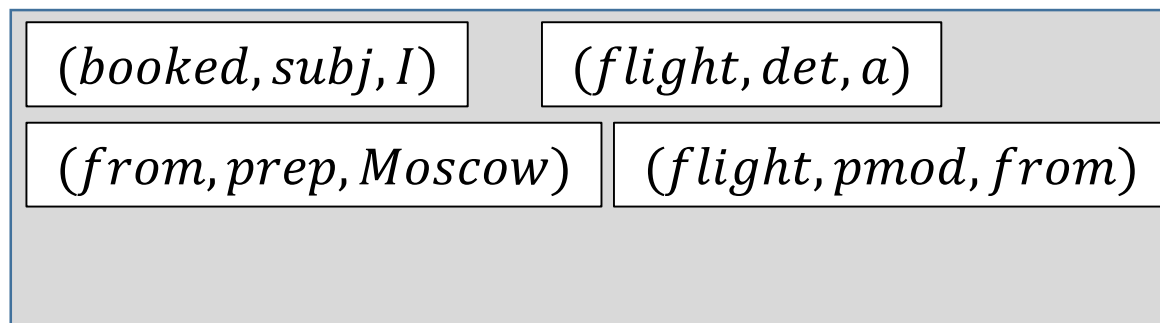
Стек



Буфер



Зависимости



Shift

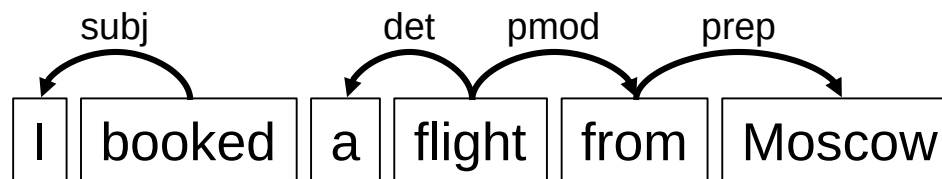
$(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

LeftArc_l

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

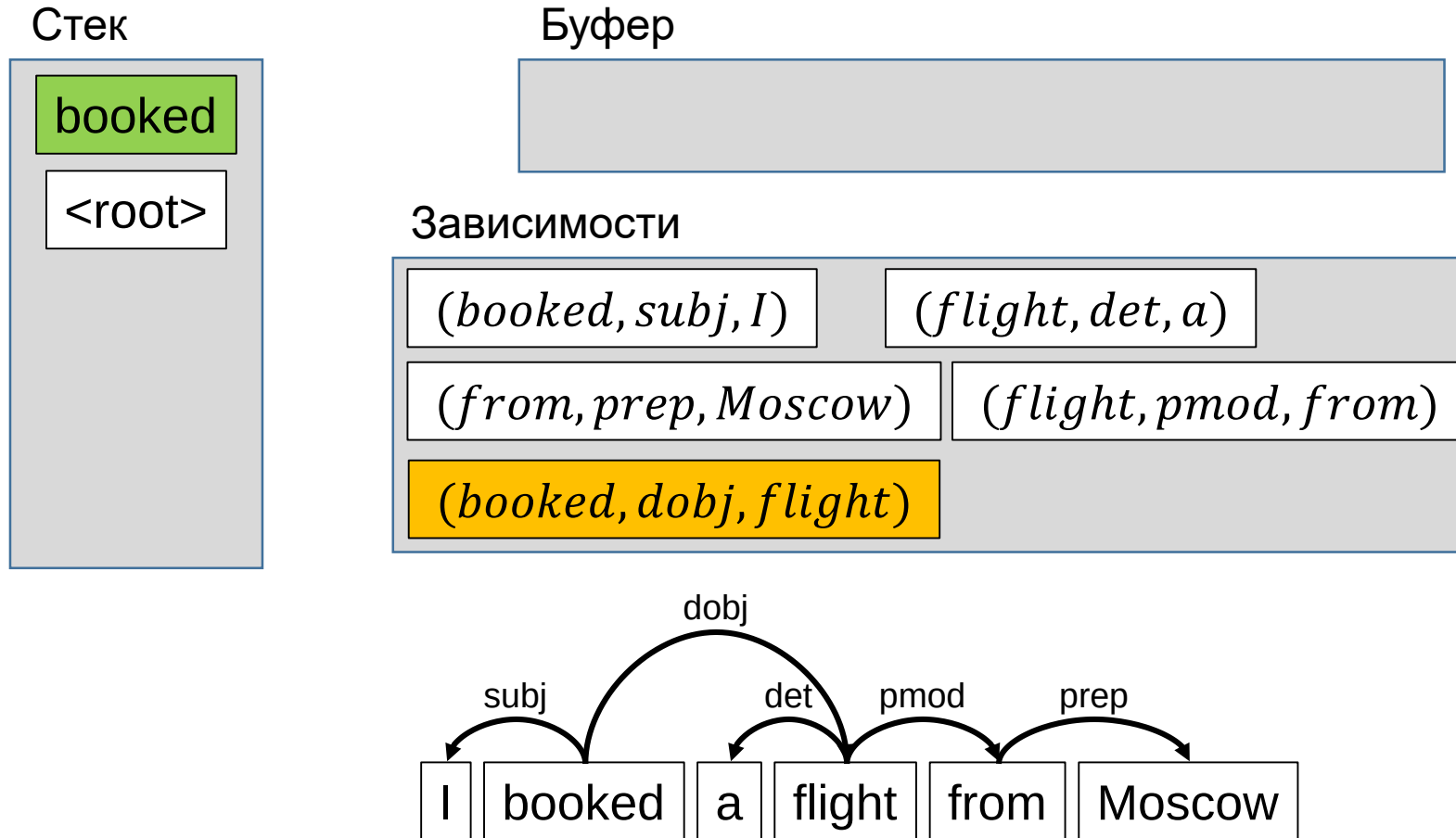
RightArc_l

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



RightArc_{dobj}

Arc-standard



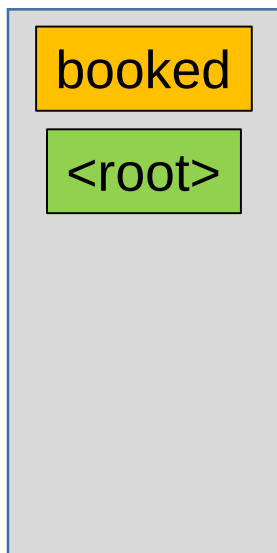
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

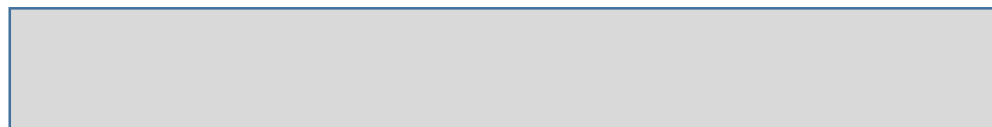
RightArc_l
 $([\sigma|i|j], B, A) \rightarrow ([\sigma|i], B, A \cup \{(i, l, j)\})$

Arc-standard

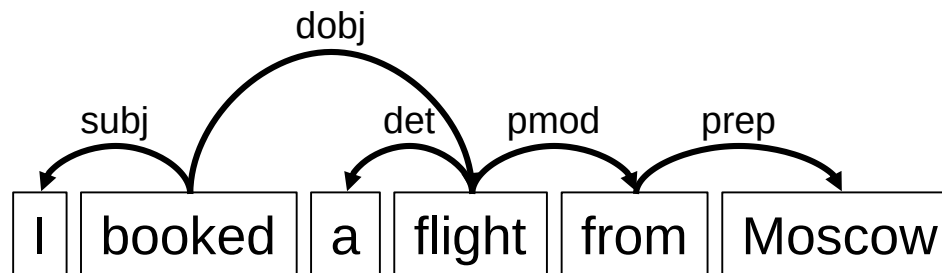
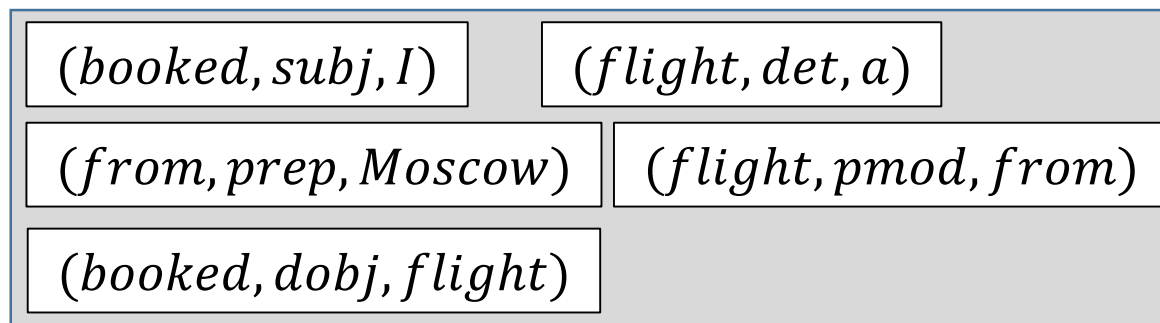
Стек



Буфер



Зависимости



Shift

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

LeftArc_l

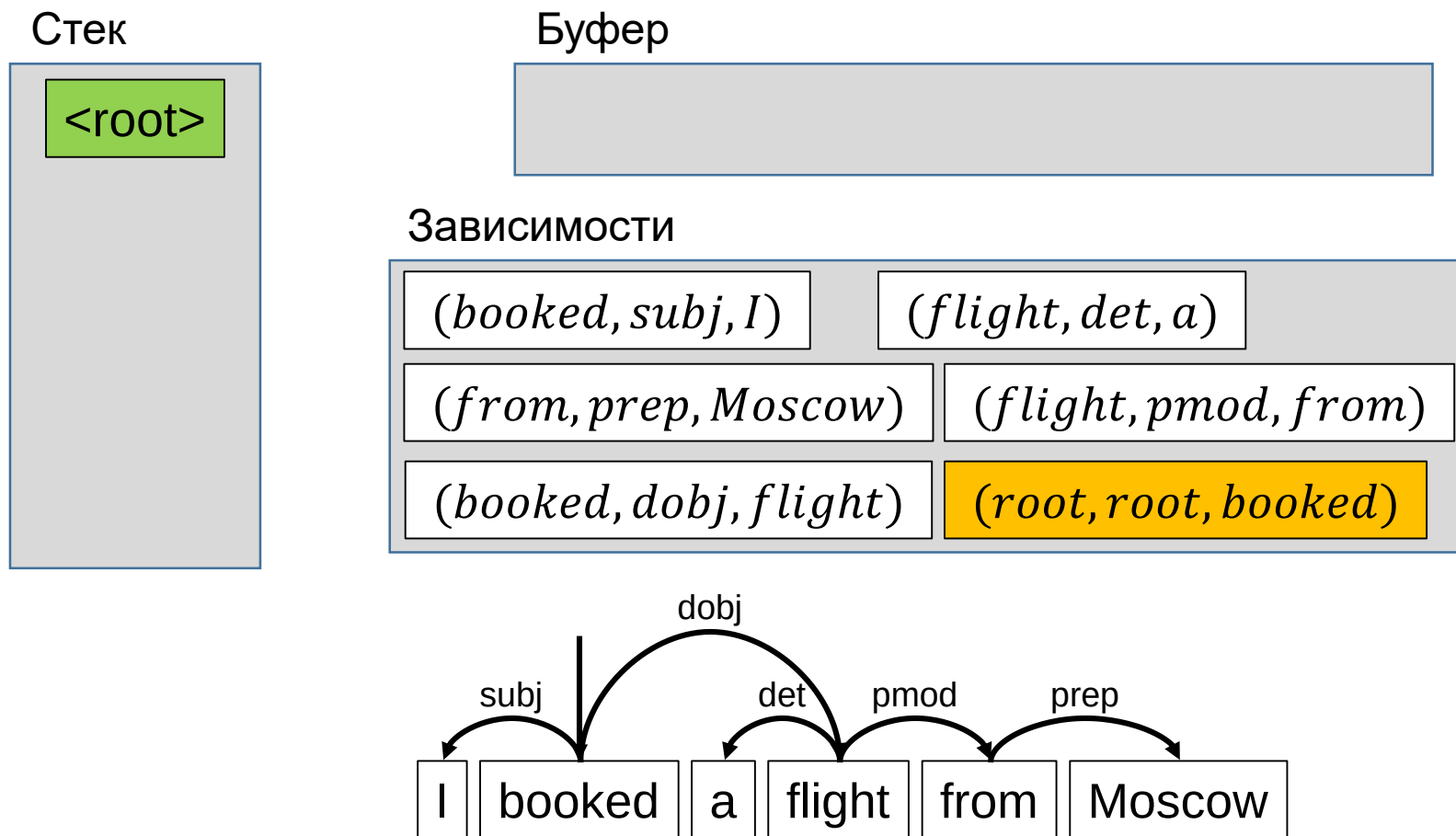
$$([\sigma|i|j], B, A) \rightarrow ([\sigma|j], B, A \cup \{(j, l, i)\})$$

RightArc_l

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

RightArc_{root}

Arc-standard



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

Arc-standard

Стек

<root>

Буфер

ЗАВИСИМОСТИ

(booked, subj, I)

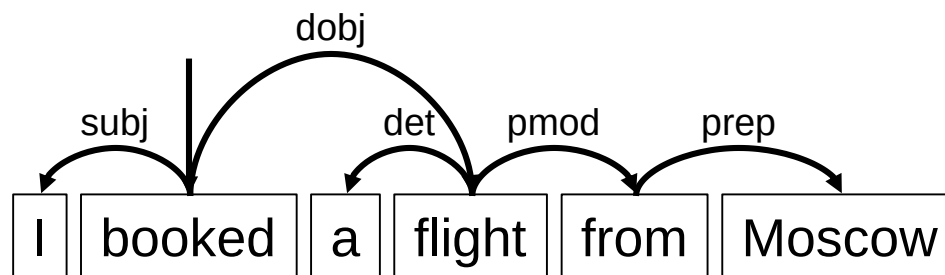
(flight, det, a)

(from, prep, Moscow)

(flight, pmod, from)

(booked, dobj, flight)

(root, root, booked)



DONE!!!

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

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

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

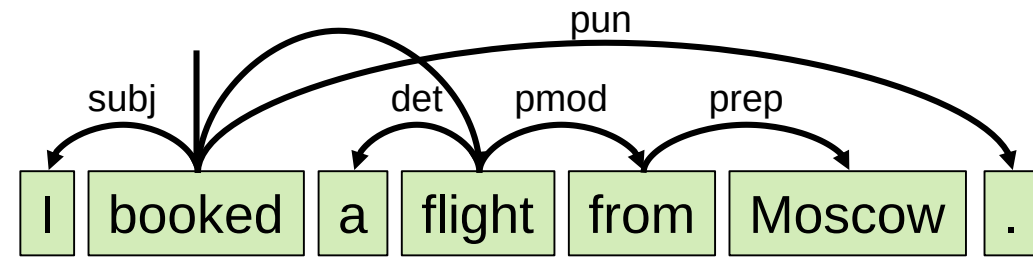
- $(Shift) \quad (\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

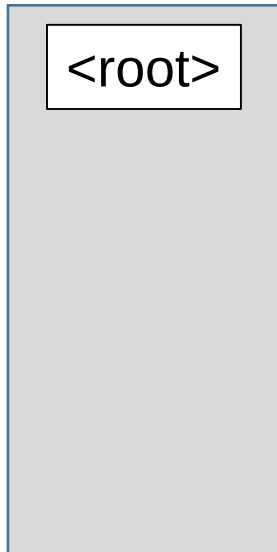
- $(RightArc_l) \quad ([\sigma|i], [j|\beta], A) \rightarrow ([\sigma|i|j], \beta, A \cup \{(i, l, j)\})$

- $(Reduce) \quad ([\sigma|i], B, A) \rightarrow (\sigma, B, A)$
если $\exists k: (k, *, i) \in A$

Arc-eager



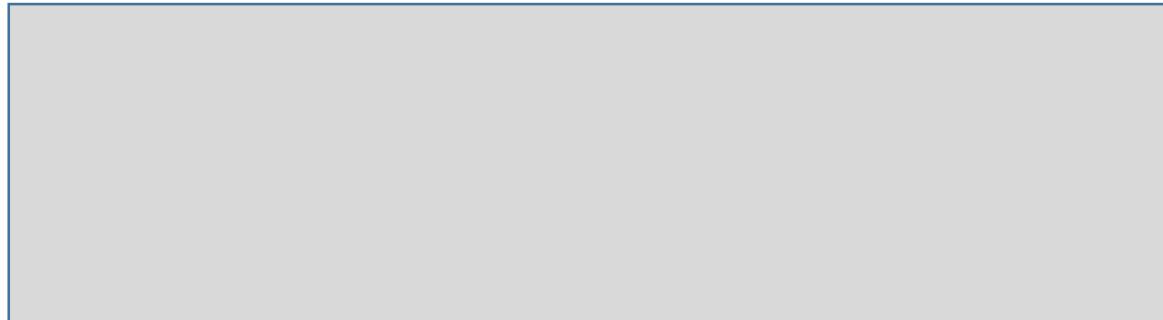
Стек



Буфер



Зависимости



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

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

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

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

Arc-eager

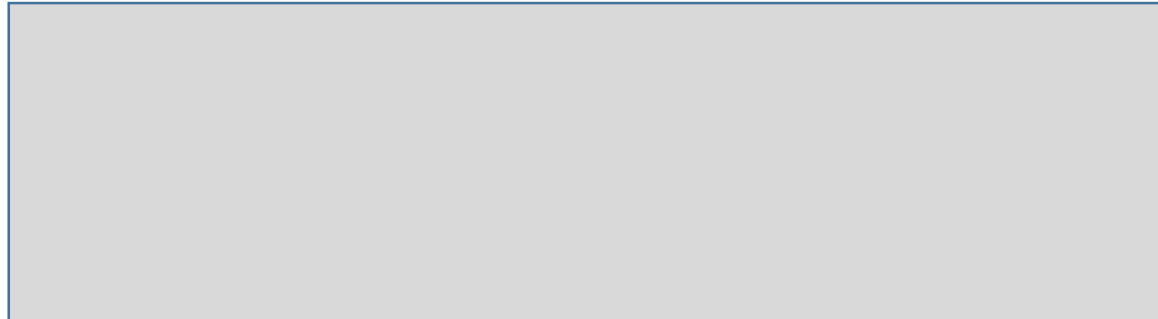
Стек

<root>

Буфер

I booked a flight from Moscow .

Зависимости



Shift

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

LeftArc_l

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

RightArc_l

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

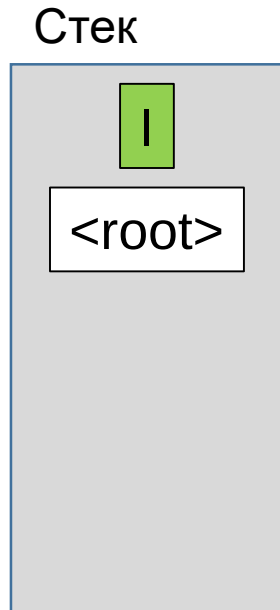
Reduce

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

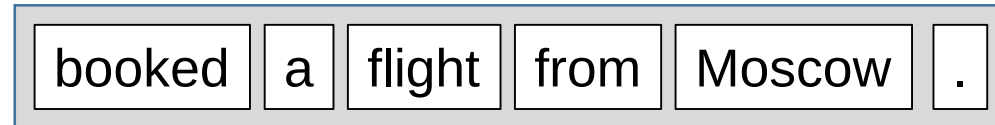
Shift

I booked a flight from Moscow .

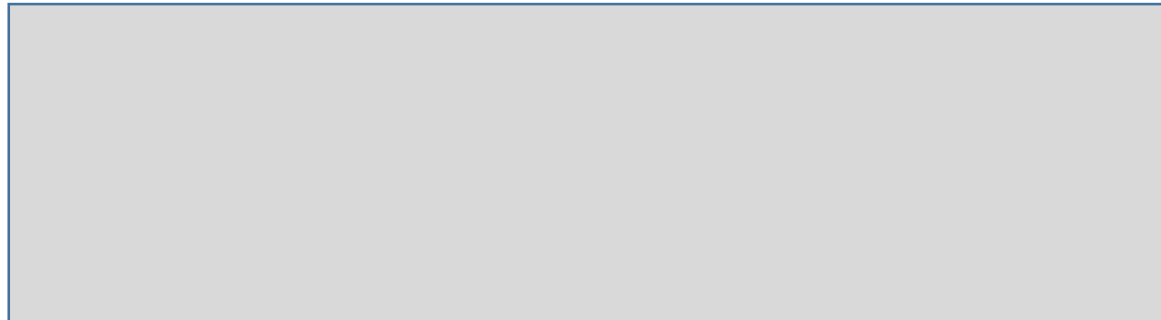
Arc-eager



Буфер



Зависимости



$$\textit{Shift}$$

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

$$\textit{LeftArc}_l$$

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

$$\textit{RightArc}_l$$

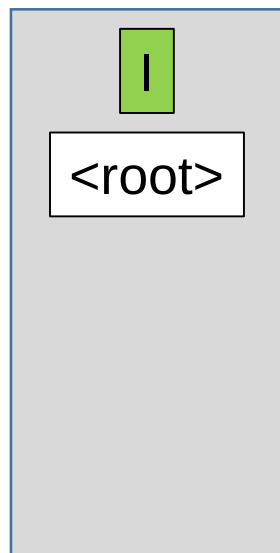
$$([\sigma|i], [j|\beta], A) \rightarrow ([\sigma|i|j], \beta, A \cup \{(i, l, j)\})$$

$$\textit{Reduce}$$

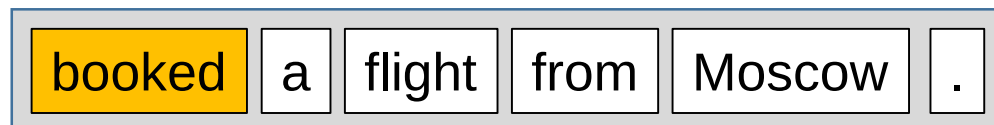
$$([\sigma|i], B, A) \rightarrow (\sigma, B, A)$$

Arc-eager

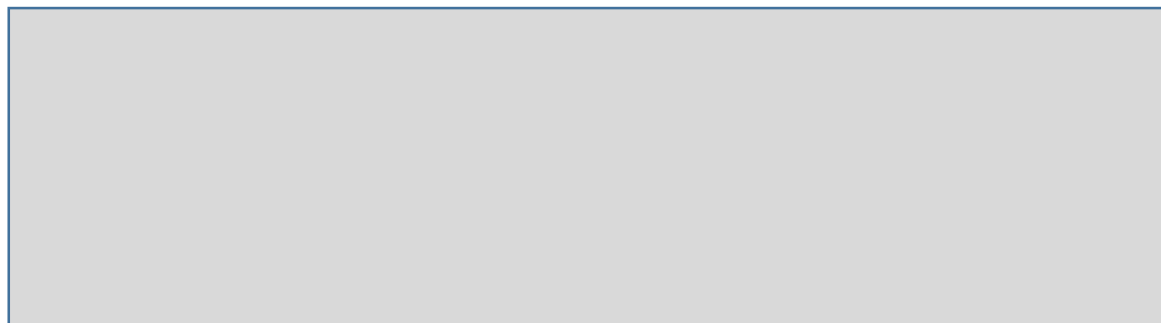
Стек



Буфер



Зависимости



$$\text{Shift} \\ (\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$$

$$\text{LeftArc}_l \\ ([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$$

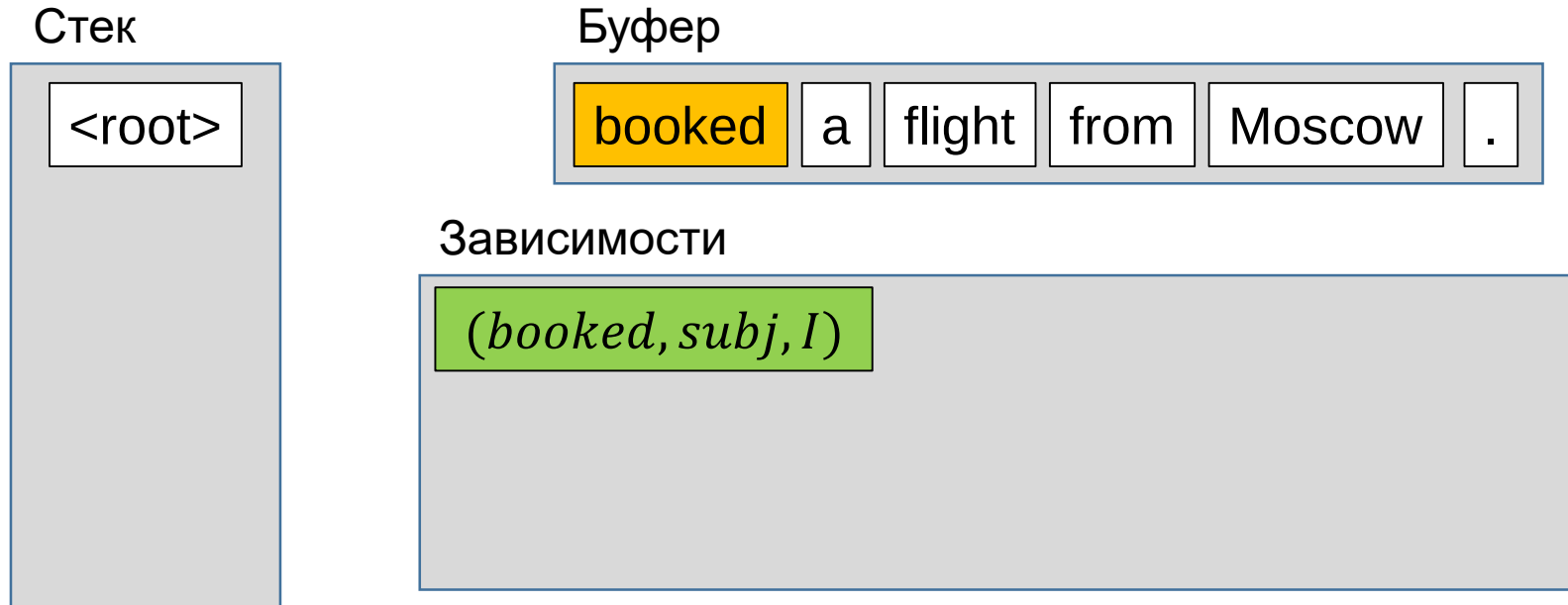
$$\text{RightArc}_l \\ ([\sigma|i], [j|\beta], A) \rightarrow ([\sigma|i|j], \beta, A \cup \{(i, l, j)\})$$

$$\text{Reduce} \\ ([\sigma|i], B, A) \rightarrow (\sigma, B, A)$$



LeftArc_{subj}

Arc-eager



$$\text{Shift}$$

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

$$\text{LeftArc}_l$$

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

$$\text{RightArc}_l$$

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

$$\text{Reduce}$$

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

Arc-eager

Стек

<root>

Буфер

booked a flight from Moscow .

Зависимости

(booked, subj, I)

Shift

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

LeftArc_l

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

RightArc_l

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

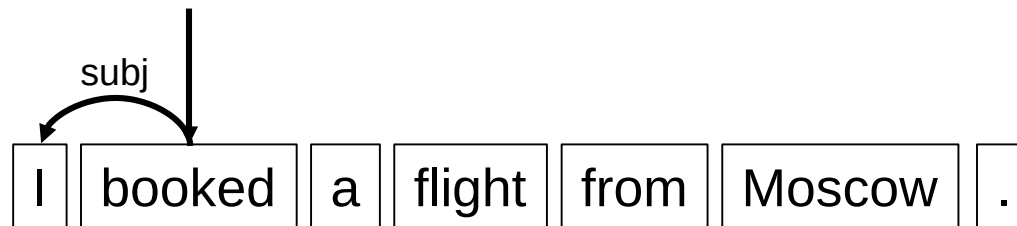
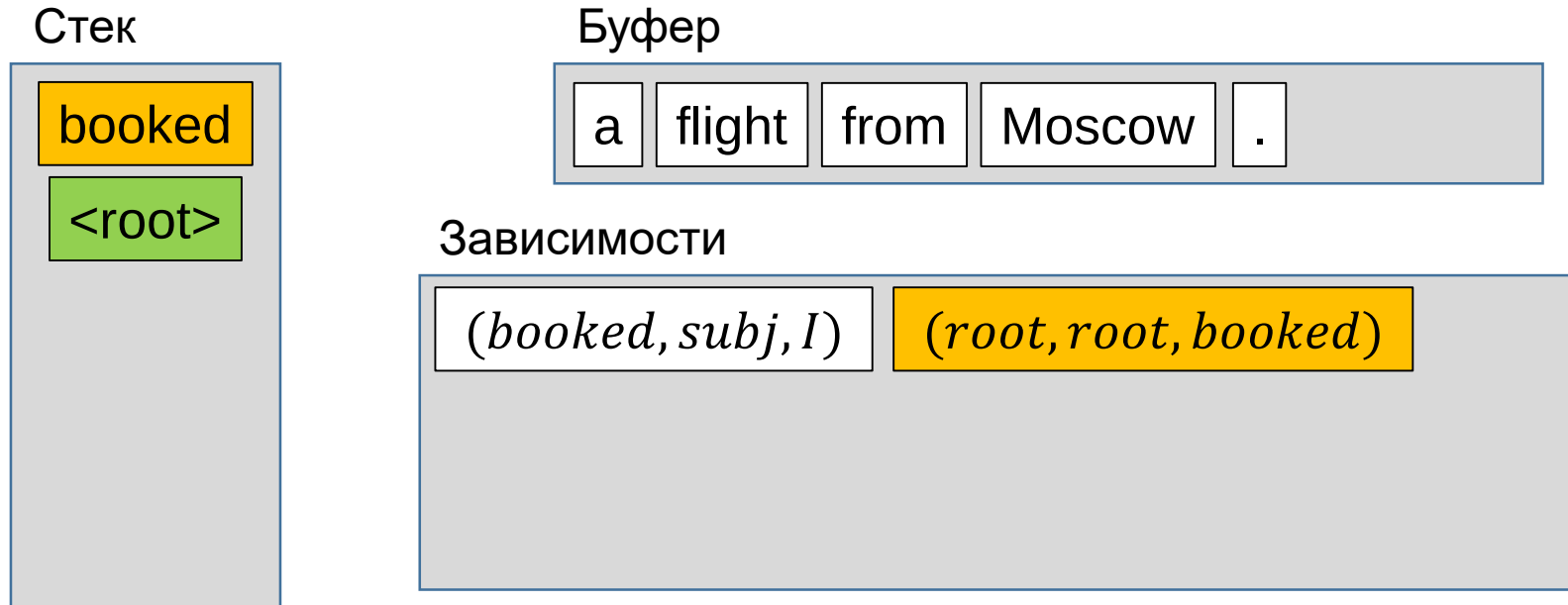
Reduce

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

RightArc_{root}



Arc-eager



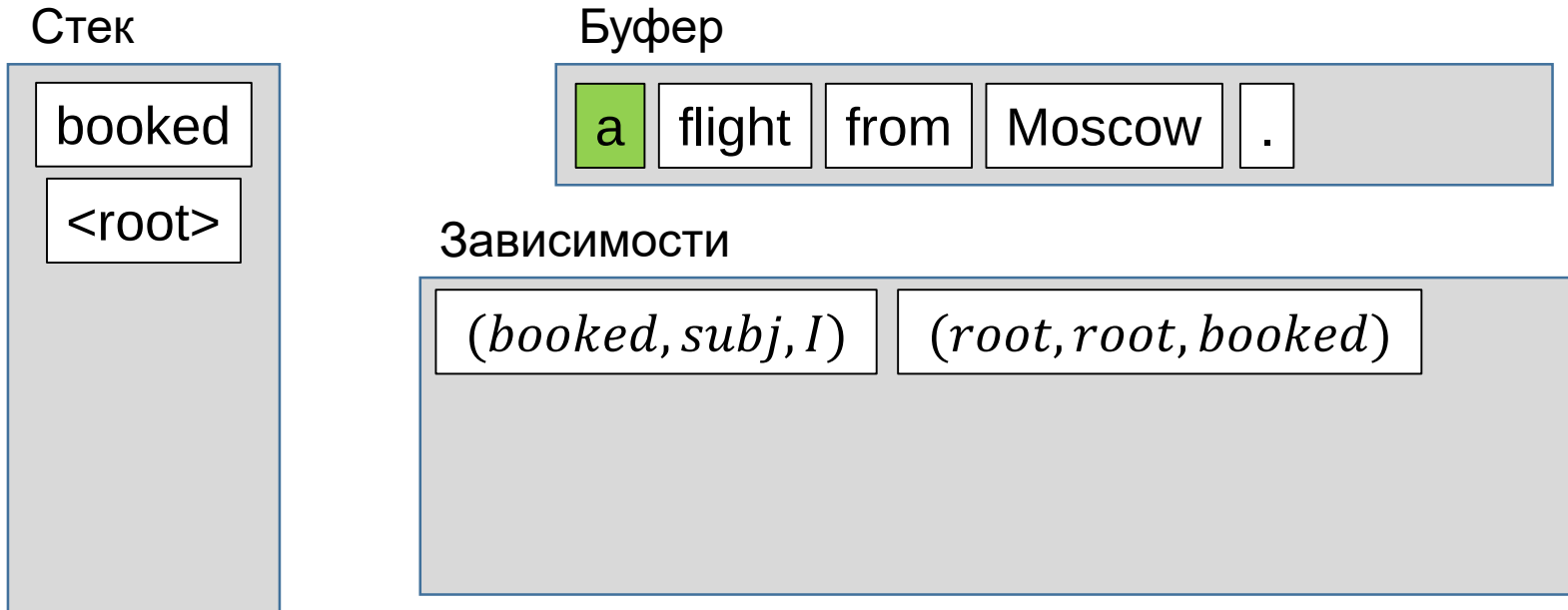
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

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

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

Arc-eager

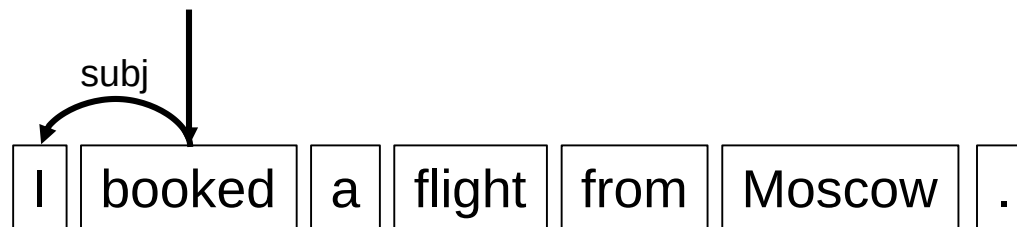
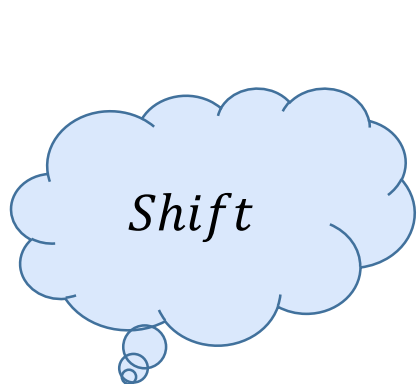


Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

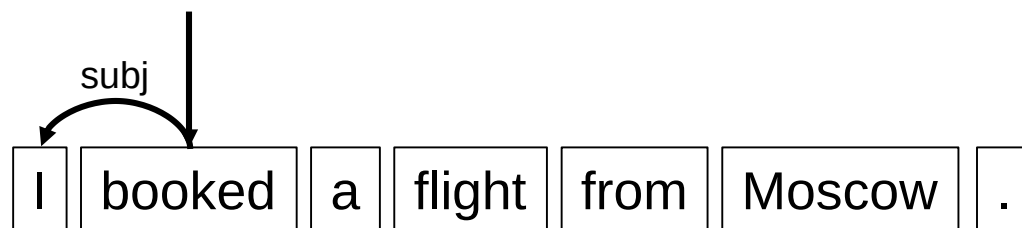
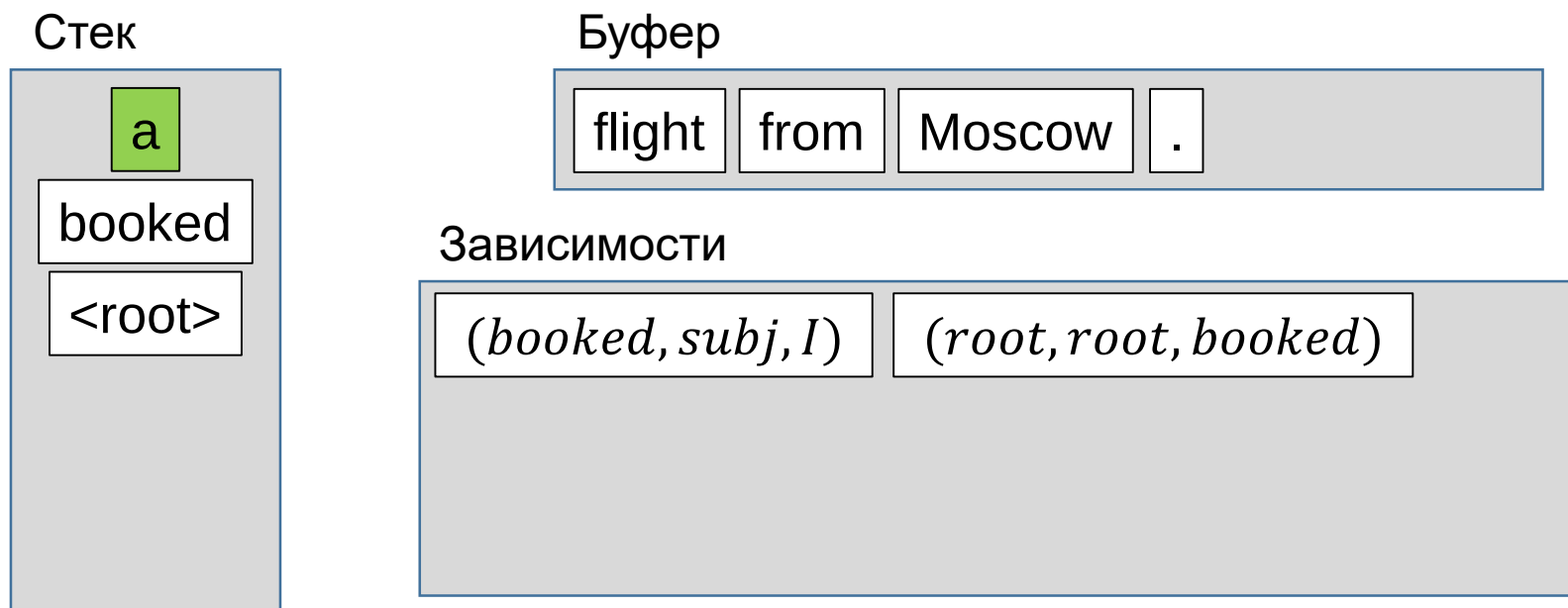
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

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



Arc-eager



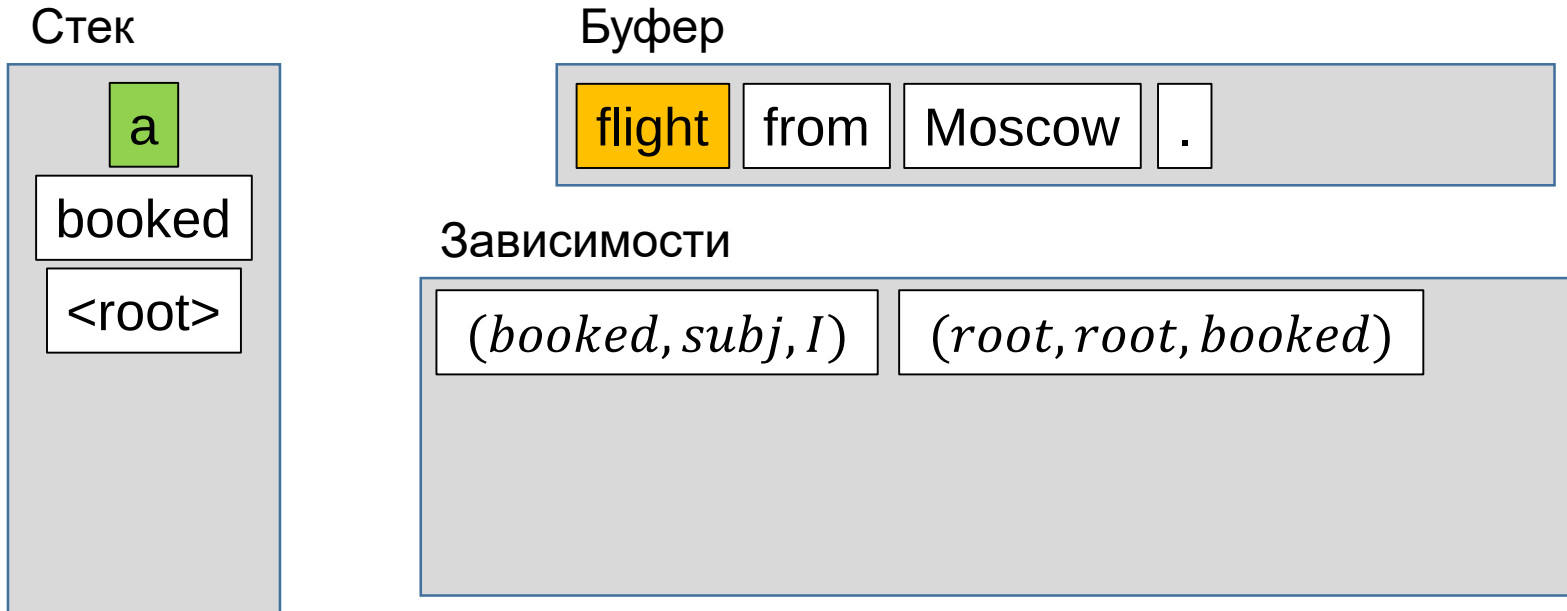
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

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

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

Arc-eager

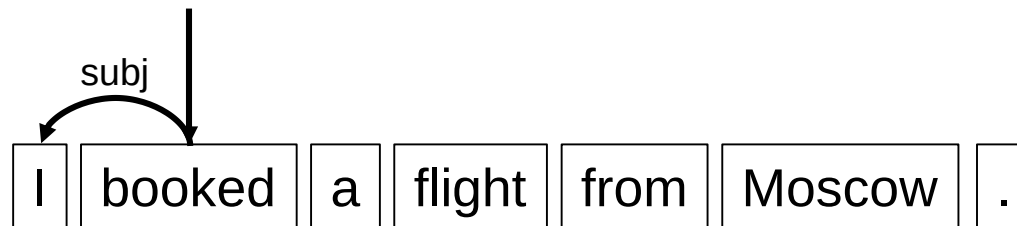


Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

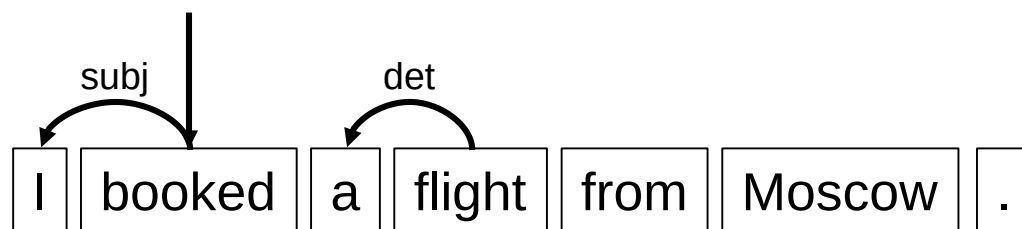
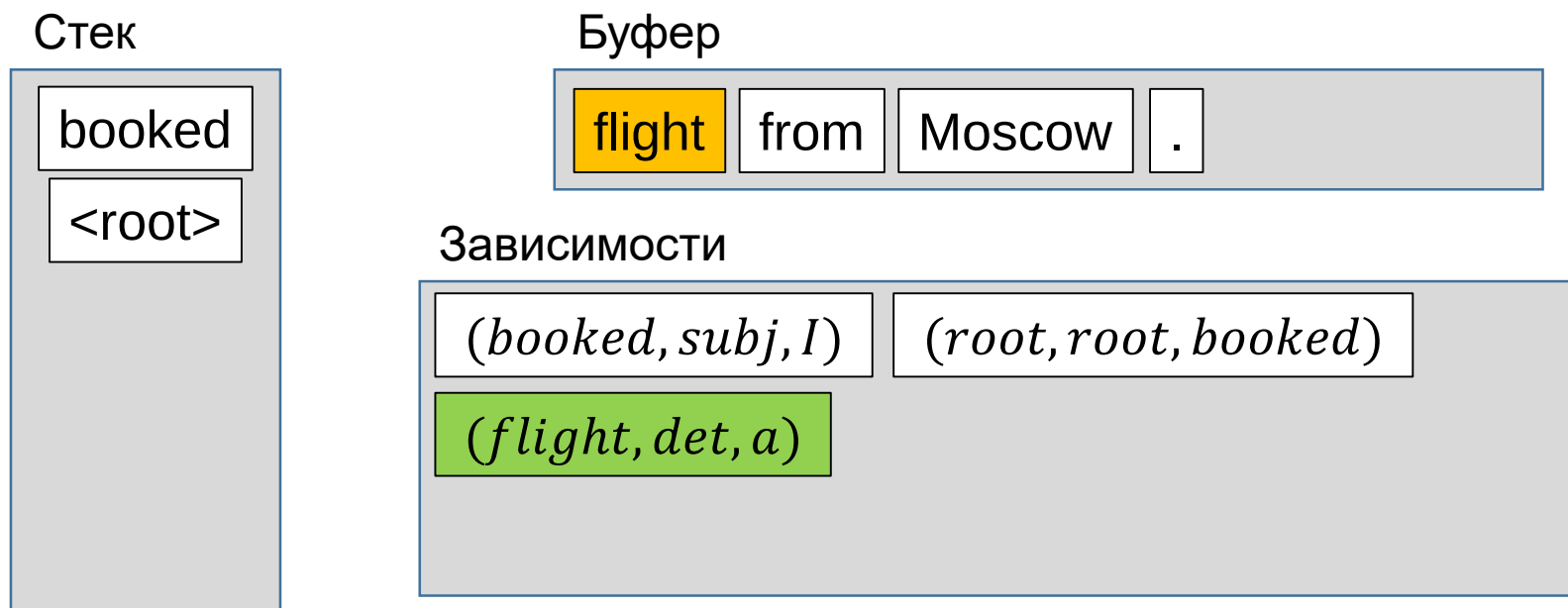
RightArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow ([\sigma|i]j, \beta, A \cup \{(i, l, j)\})$

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



LeftArc_{det}

Arc-eager



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

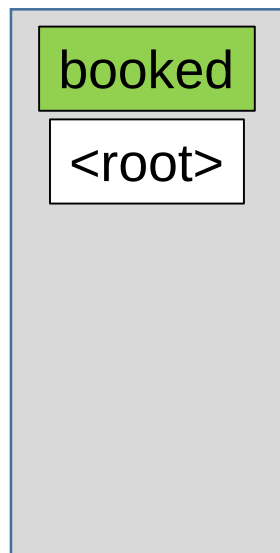
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

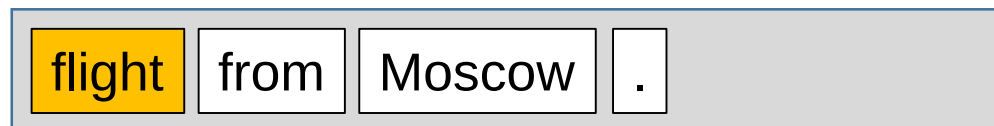
Reduce
 $([\sigma|i], B, A) \rightarrow (\sigma, B, A)$

Arc-eager

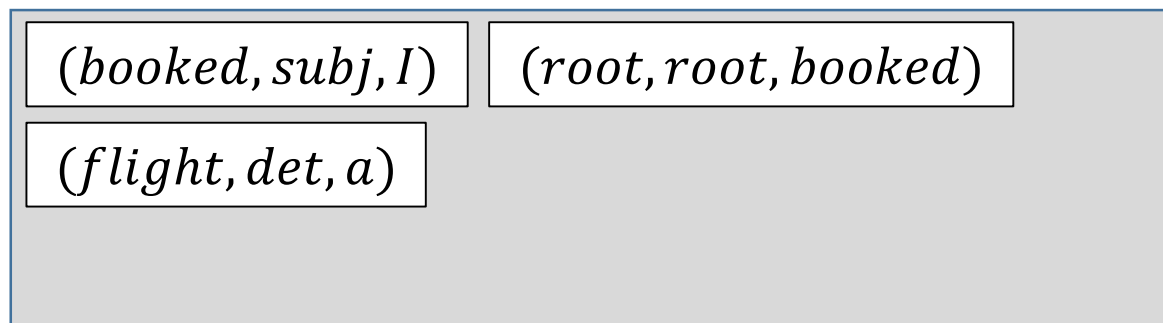
Стек



Буфер



Зависимости



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

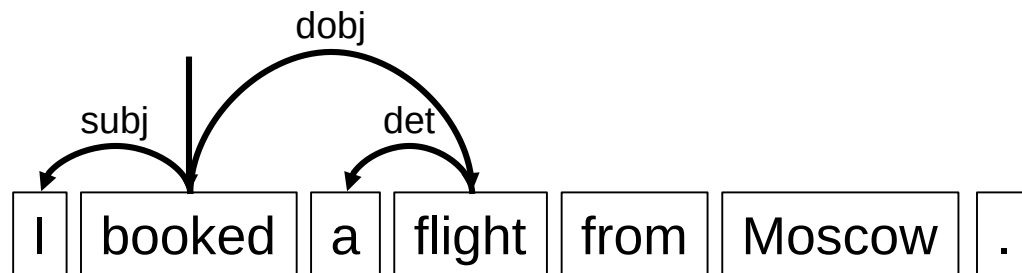
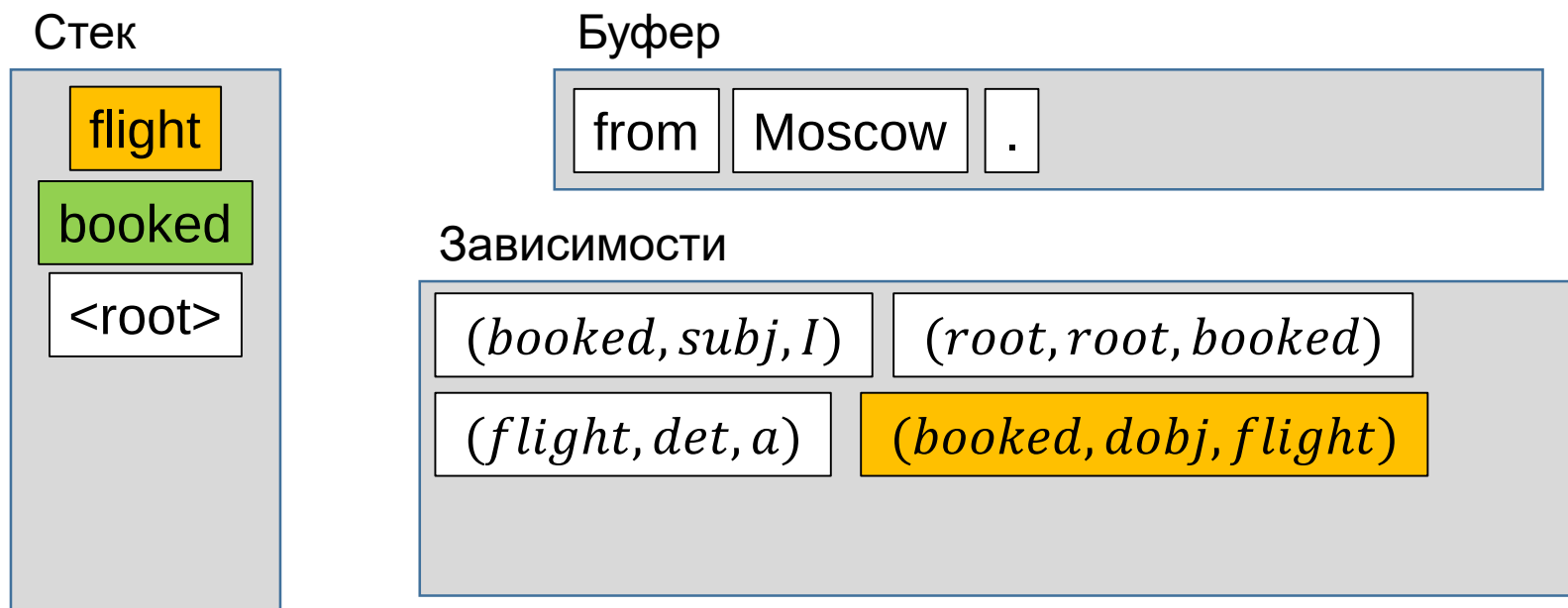
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

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

RightArc_{dobj}

Arc-eager



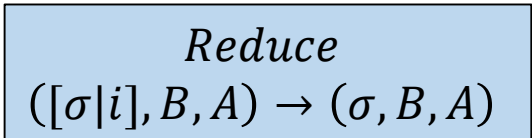
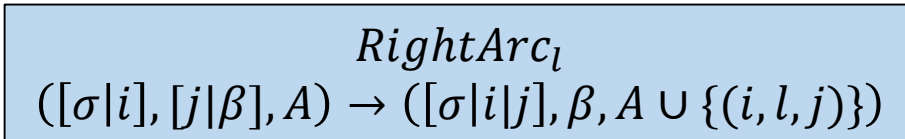
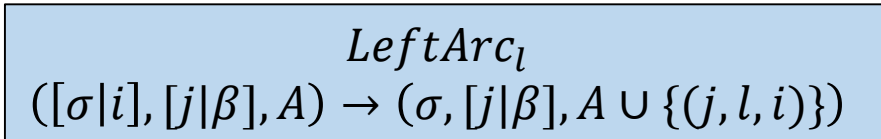
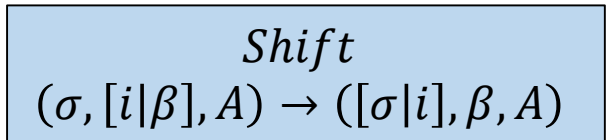
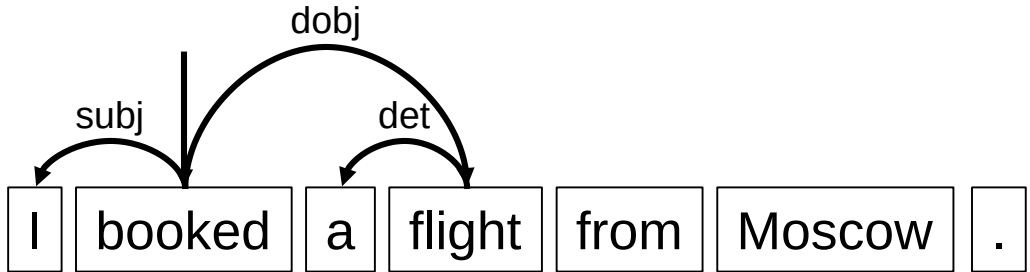
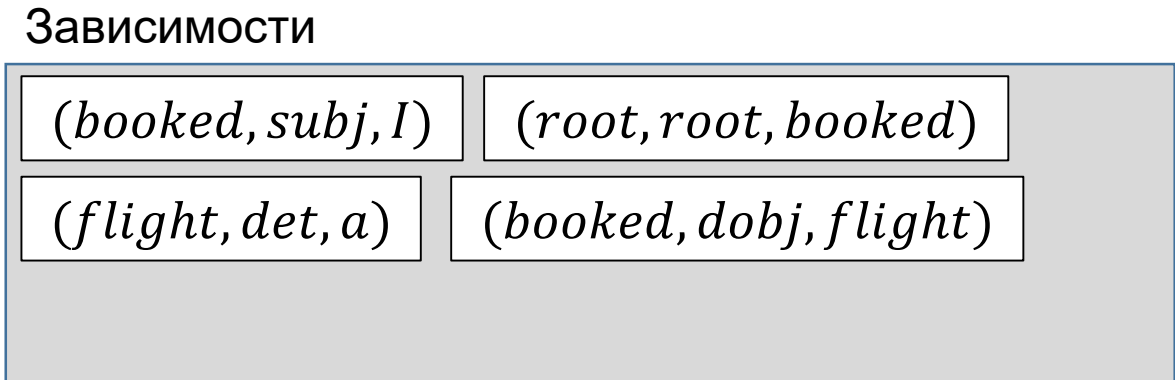
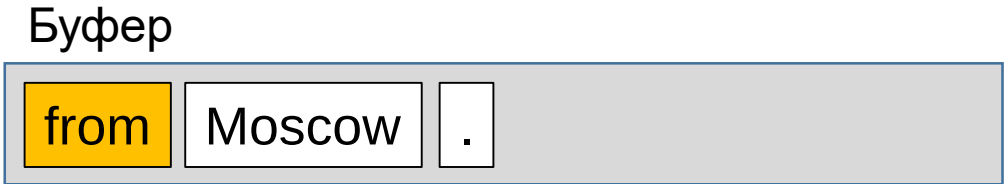
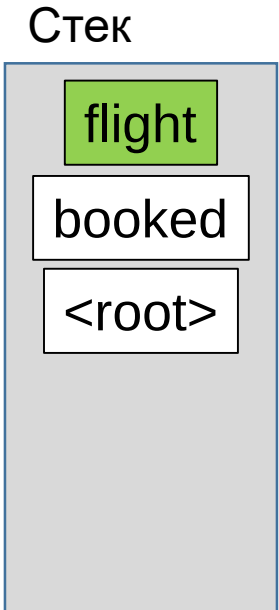
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

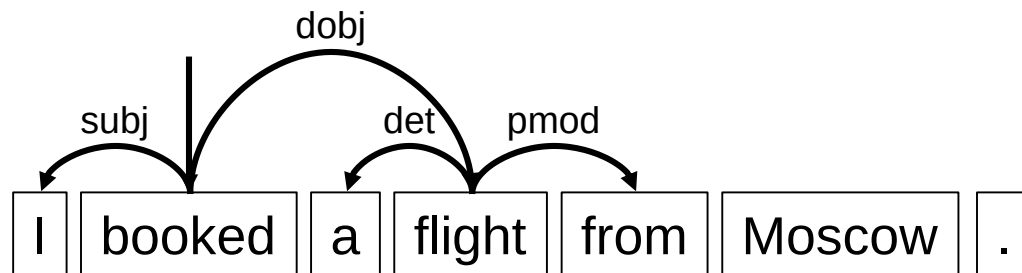
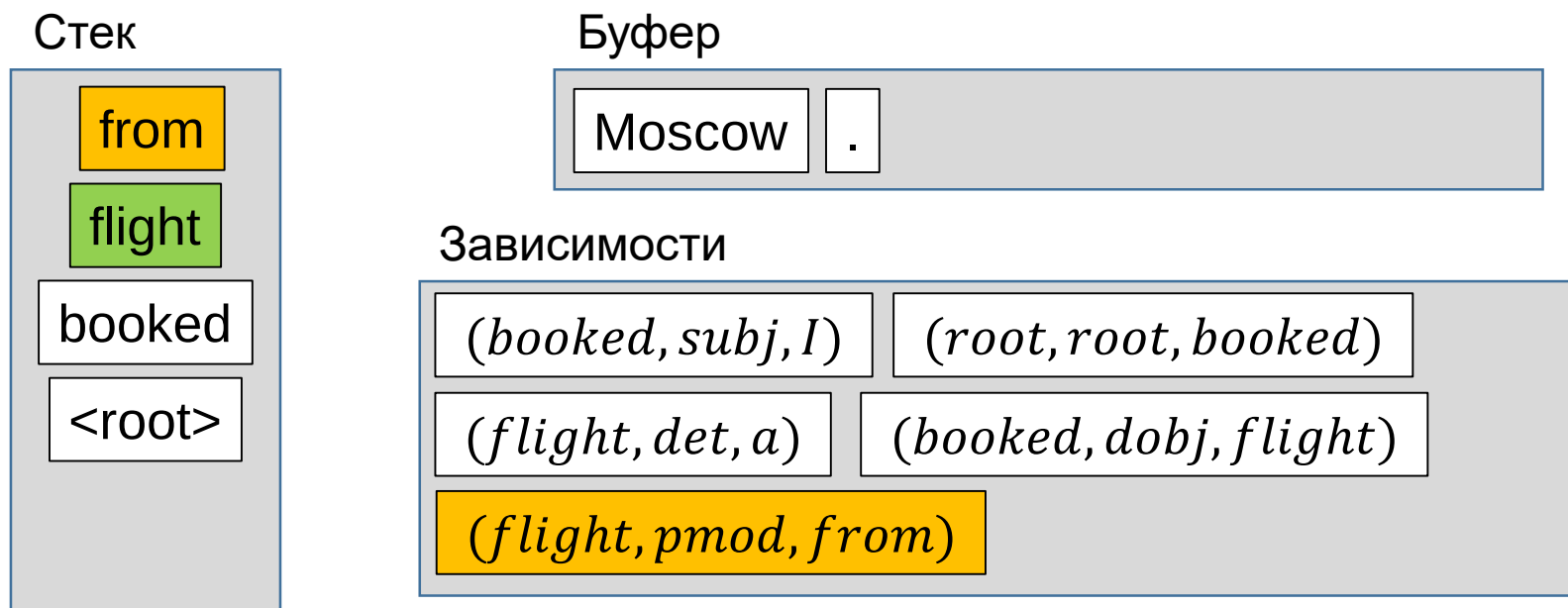
RightArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow ([\sigma|i]j, \beta, A \cup \{(i, l, j)\})$

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

Arc-eager



Arc-eager



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

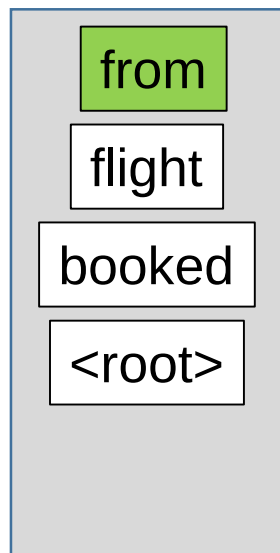
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

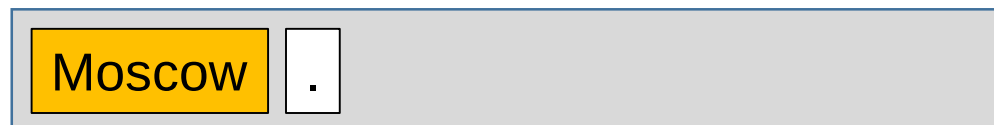
Reduce
 $([\sigma|i], B, A) \rightarrow (\sigma, B, A)$

Arc-eager

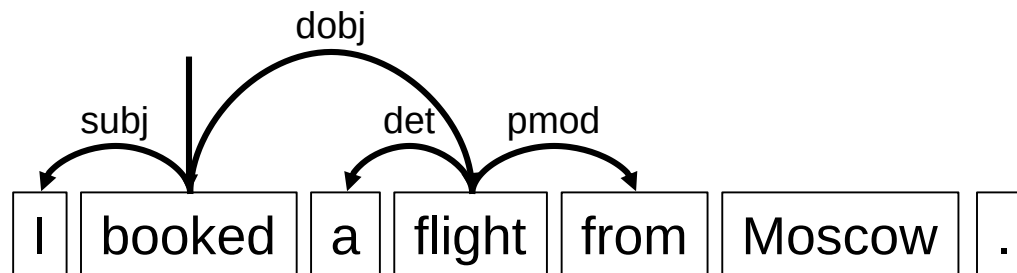
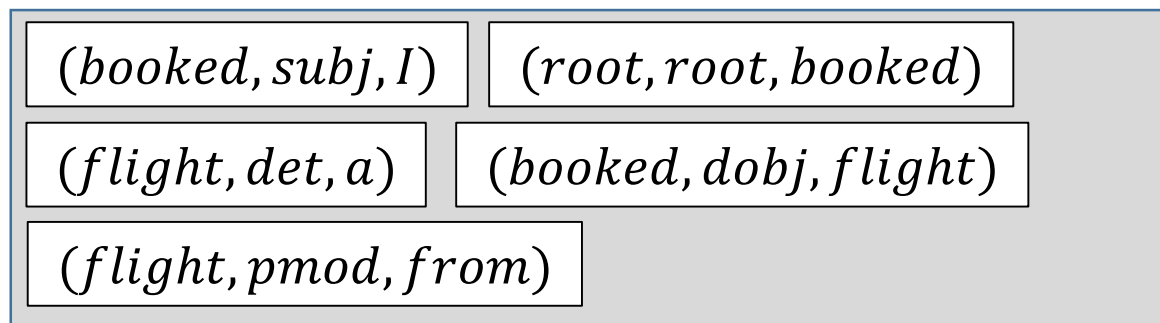
Стек



Буфер



Зависимости



RightArc_{prep}

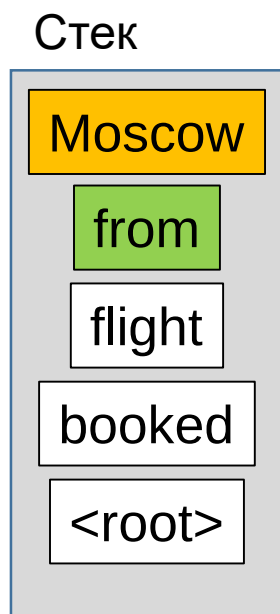
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

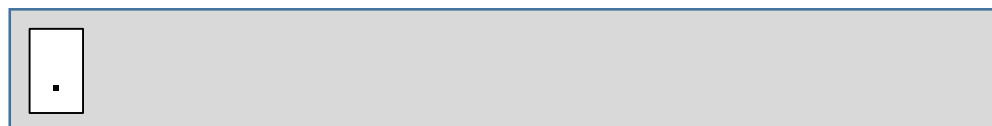
RightArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow ([\sigma|i]j, \beta, A \cup \{(i, l, j)\})$

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

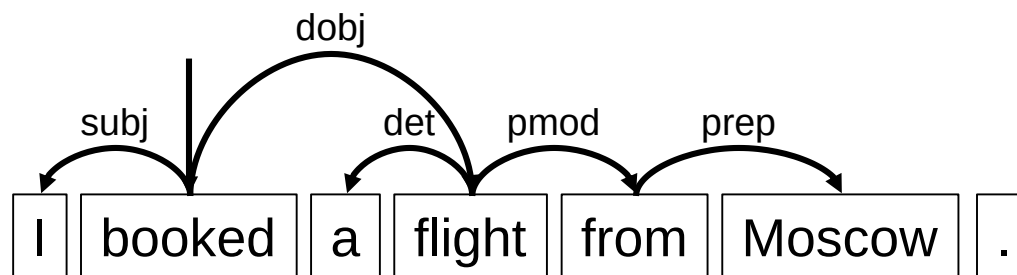
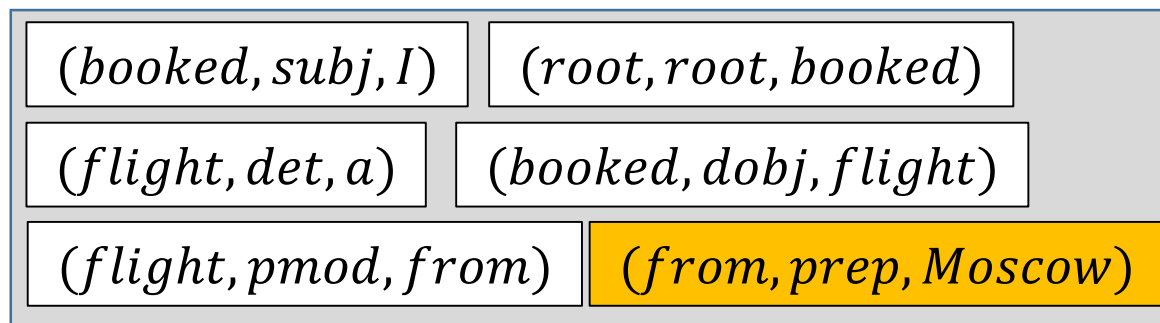
Arc-eager



Буфер



ЗАВИСИМОСТИ



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

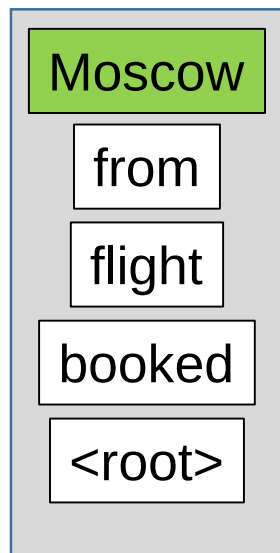
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

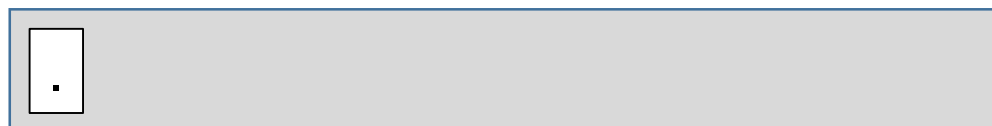
Reduce
 $([\sigma|i], B, A) \rightarrow (\sigma, B, A)$

Arc-eager

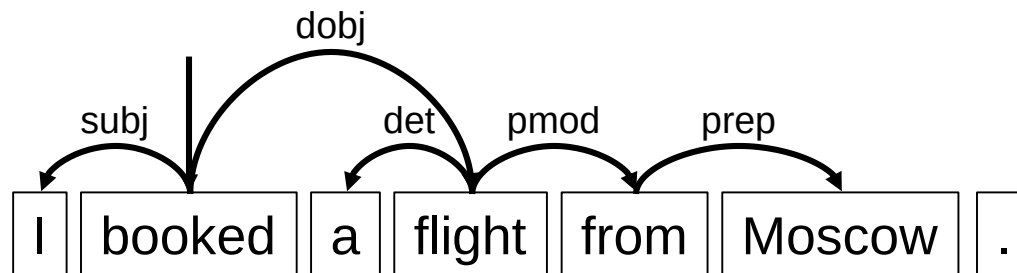
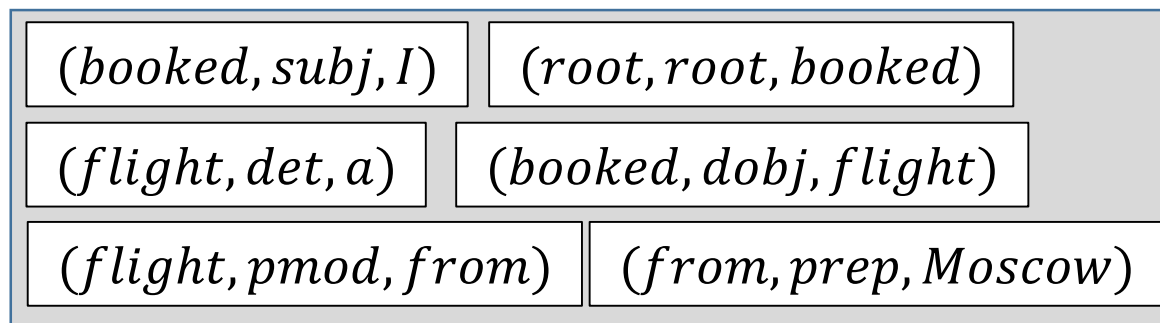
Стек



Буфер



ЗАВИСИМОСТИ



Reduce

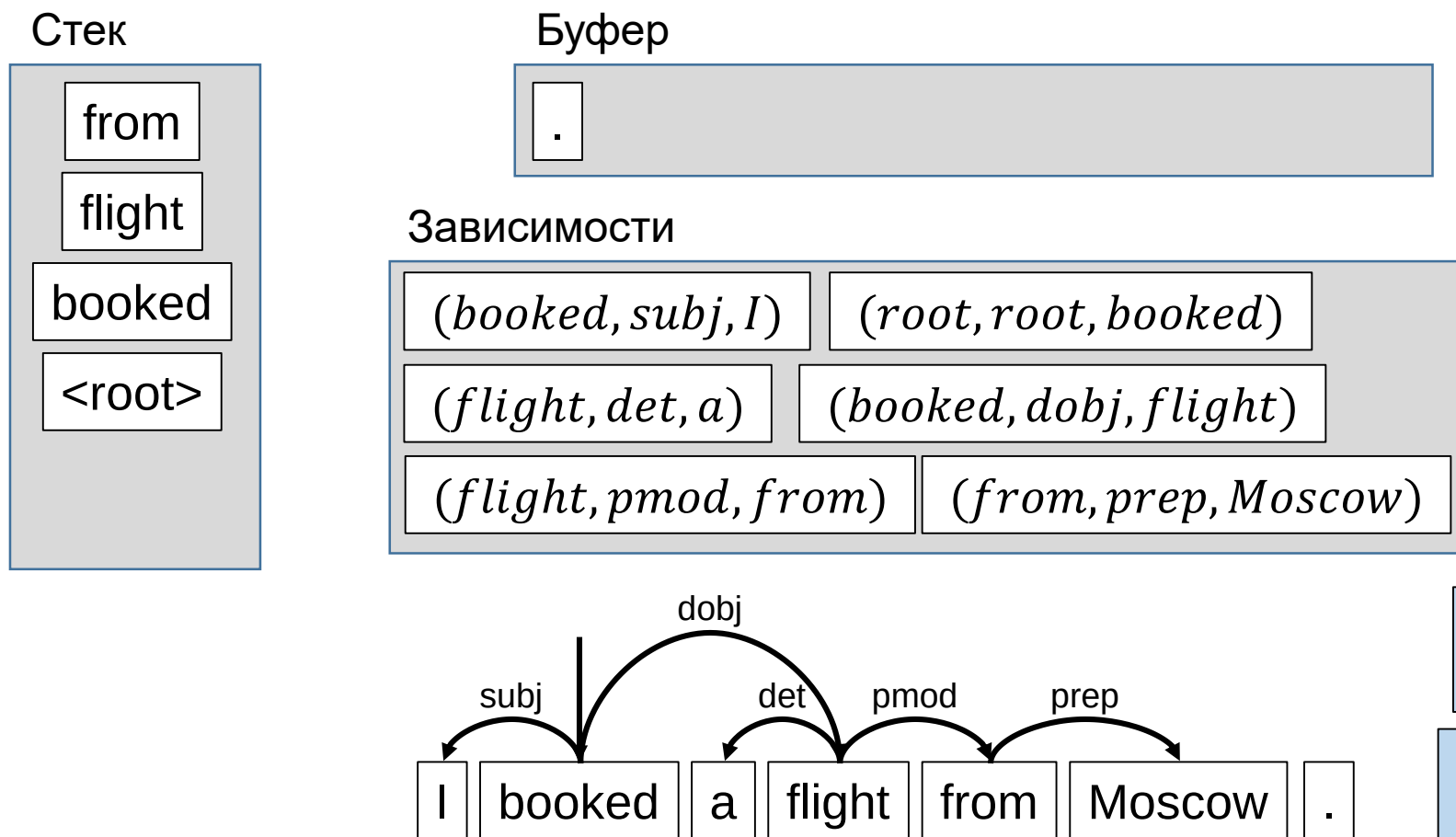
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

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

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

Arc-eager



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

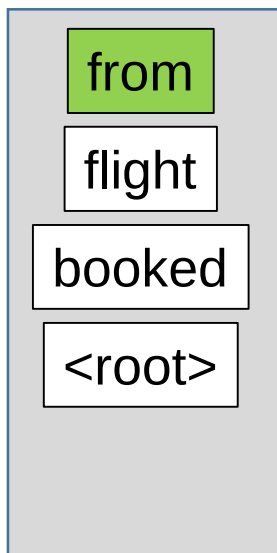
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

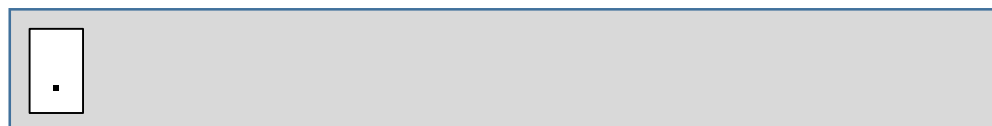
Reduce
 $([\sigma|i], B, A) \rightarrow (\sigma, B, A)$

Arc-eager

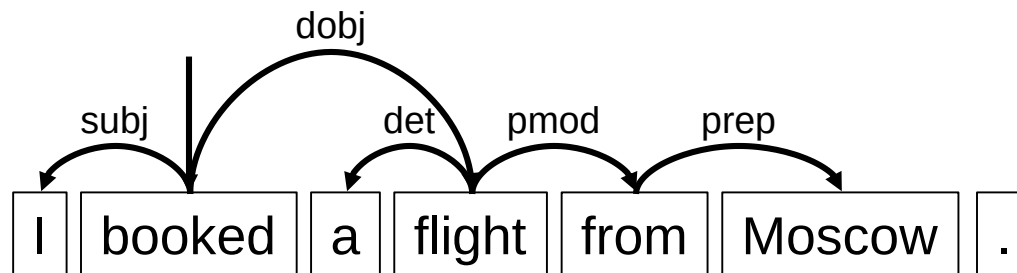
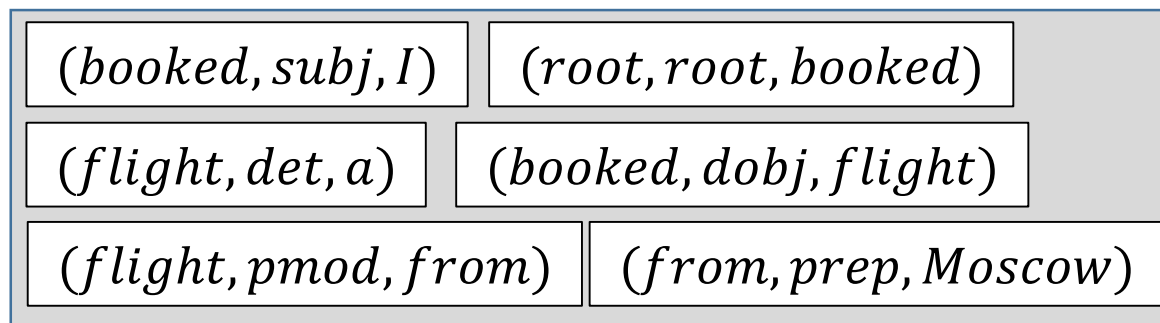
Стек



Буфер



ЗАВИСИМОСТИ



Reduce

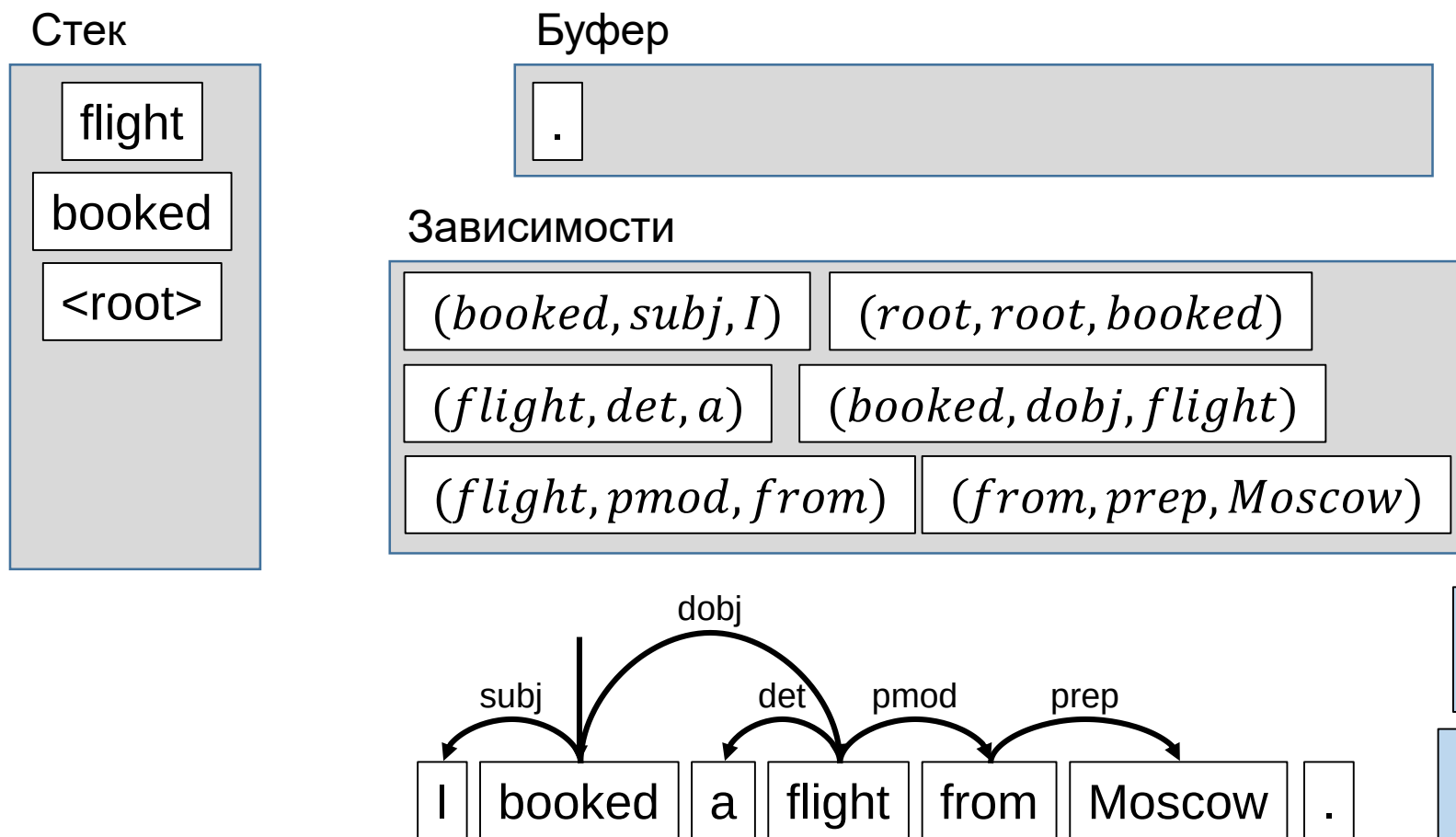
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

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

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

Arc-eager



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

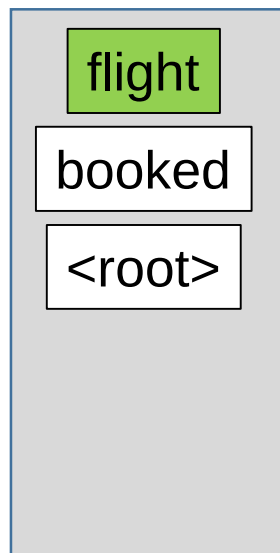
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

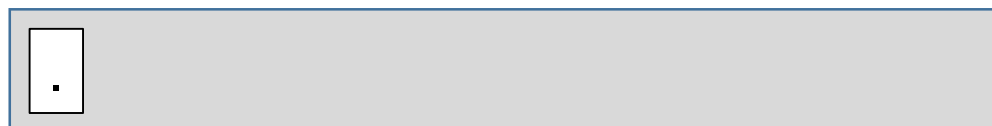
Reduce
 $([\sigma|i], B, A) \rightarrow (\sigma, B, A)$

Arc-eager

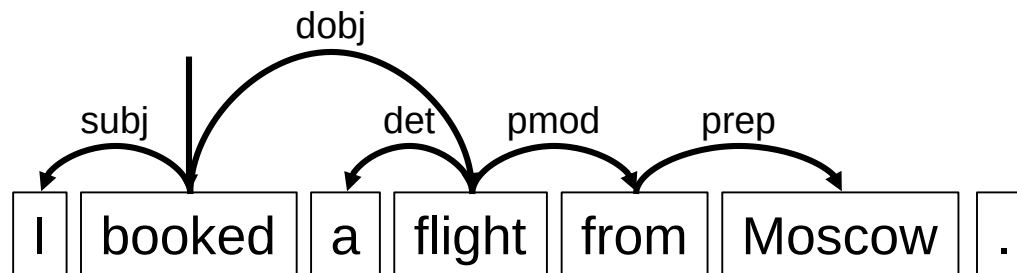
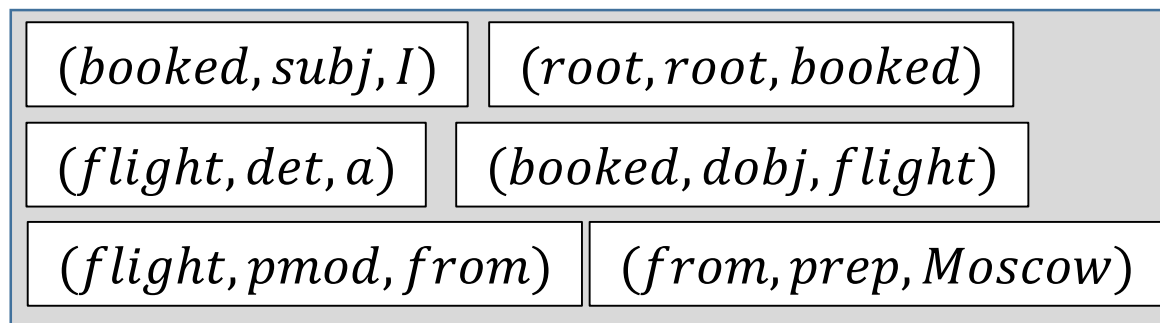
Стек



Буфер



Зависимости



Reduce

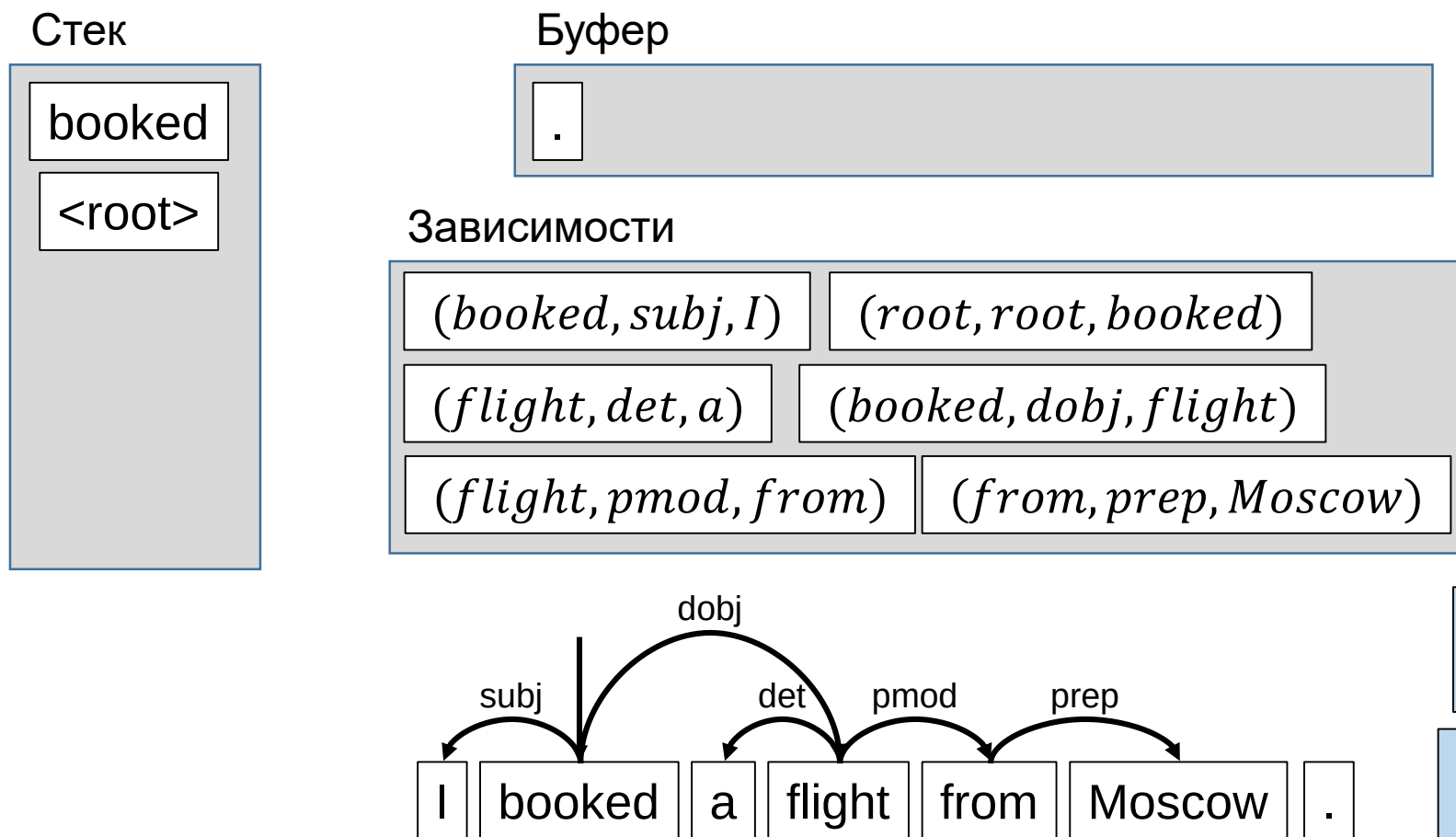
Shift
 $(\sigma, [i|\beta], A) \rightarrow ([\sigma|i], \beta, A)$

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

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

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

Arc-eager



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

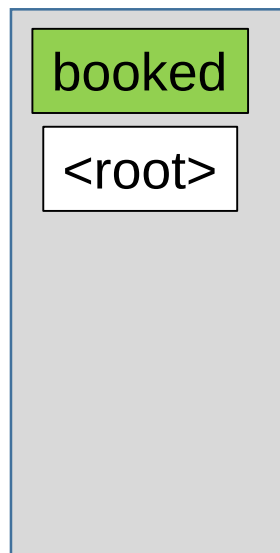
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

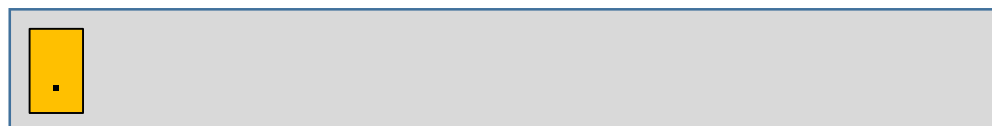
Reduce
 $([\sigma|i], B, A) \rightarrow (\sigma, B, A)$

Arc-eager

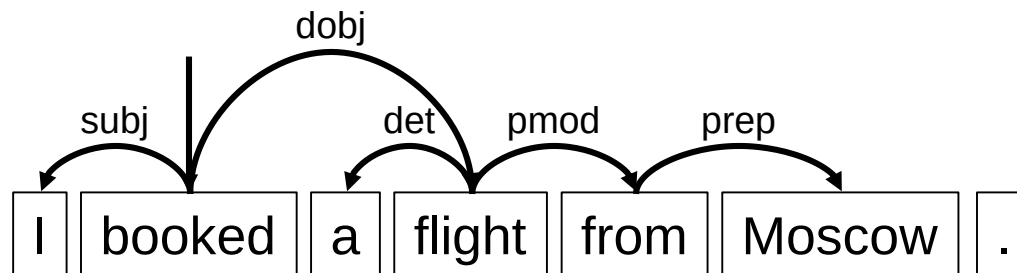
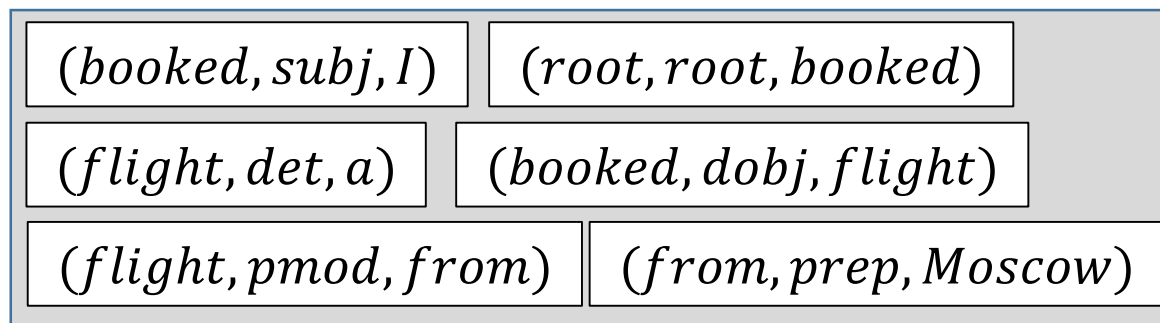
Стек



Буфер



Зависимости



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

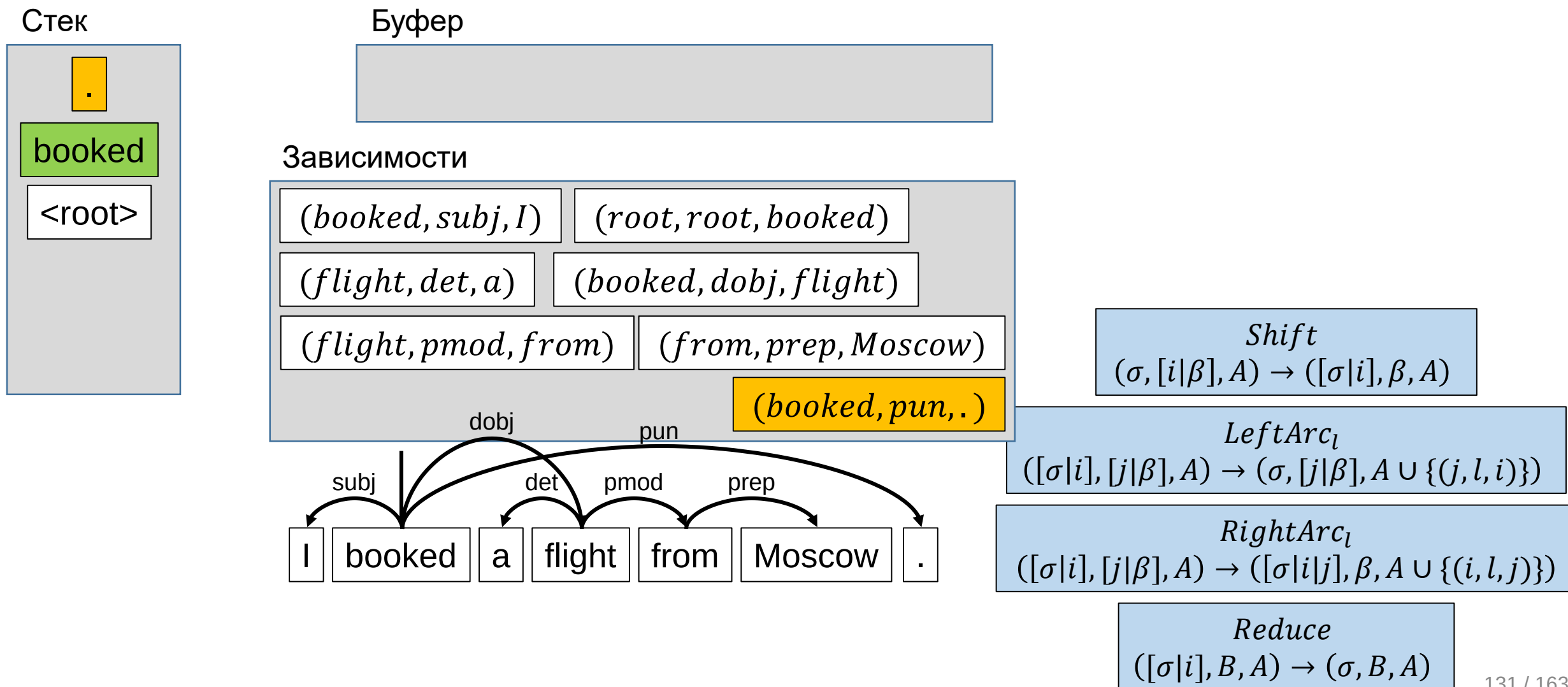
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

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

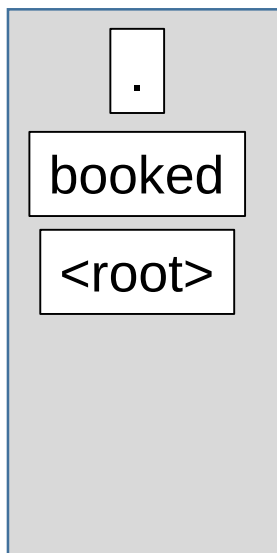
RightArc_{pun}

Arc-eager



Arc-eager

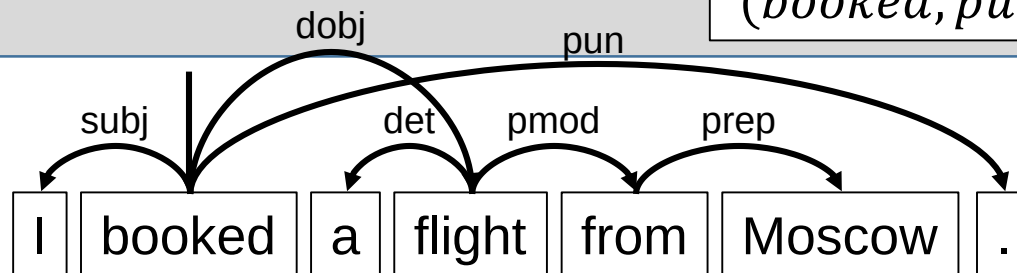
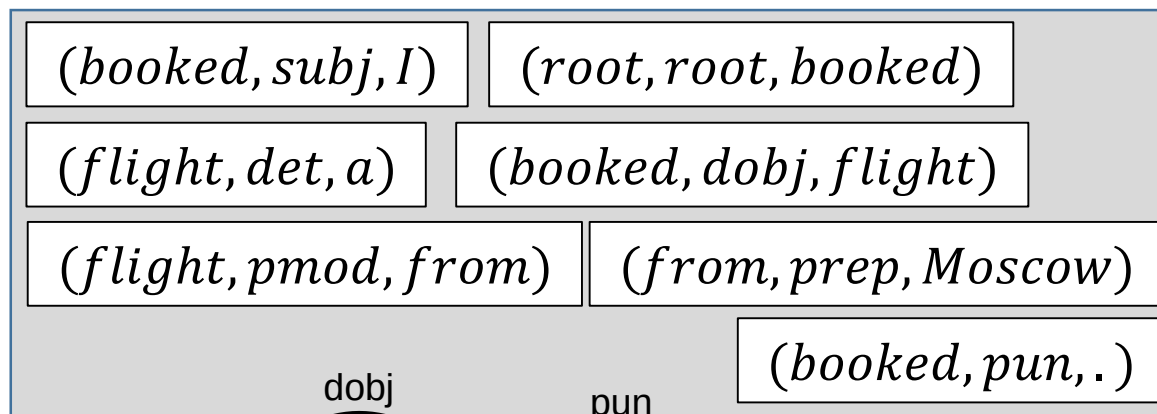
Стек



Буфер



Зависимости



DONE!!!

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

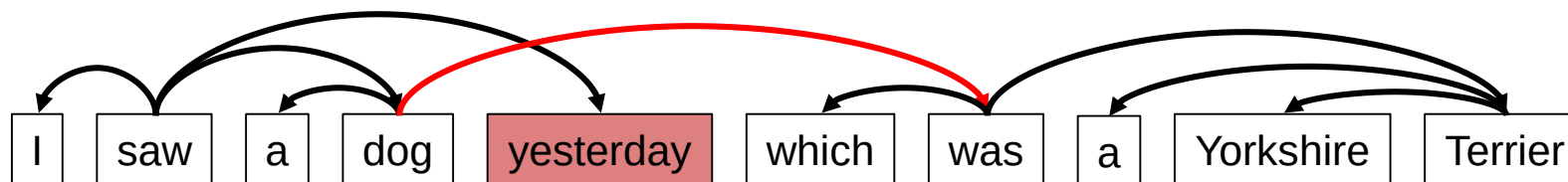
LeftArc_l
 $([\sigma|i], [j|\beta], A) \rightarrow (\sigma, [j|\beta], A \cup \{(j, l, i)\})$

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

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

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

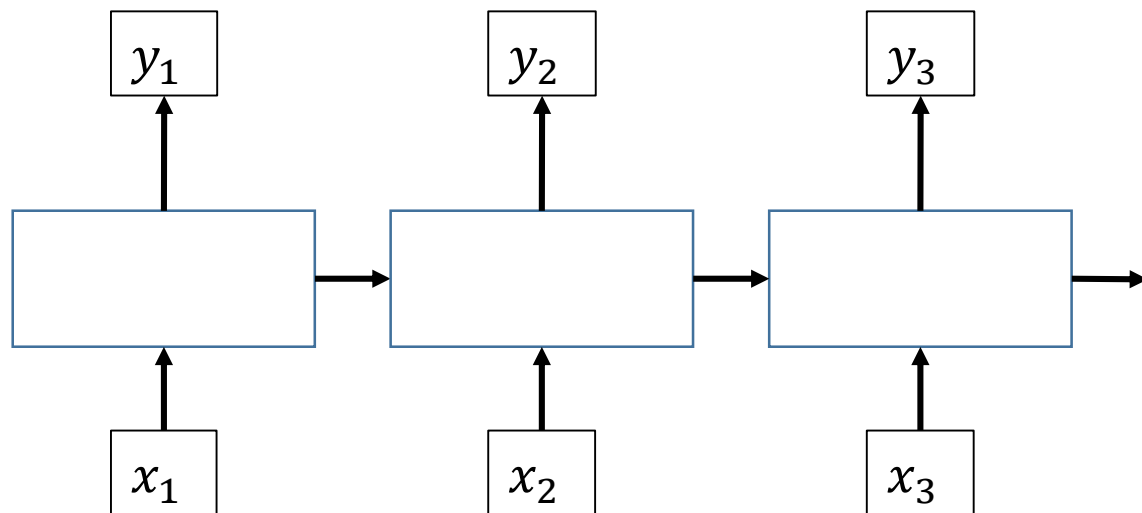
- Задача многоклассовой классификации
 - Объекты классификации: конфигурации парсера
 - Классы: правила перехода
- Методы:
 - Multiclass SVM
 - Naïve Bayes
 - Decision trees
 - Нейронные сети
 - ...

Признаки классификации

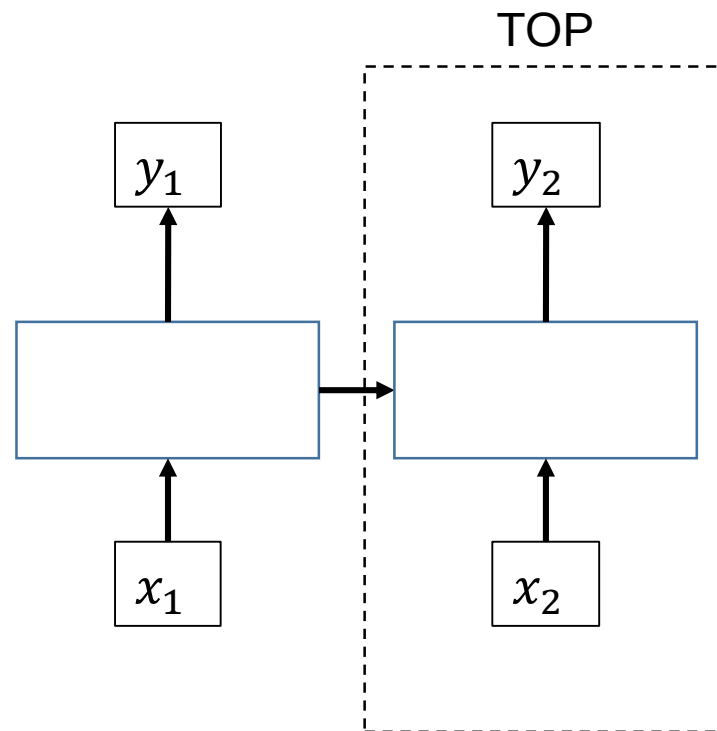
- Признаки для элемента
 - Словоформа
 - Лемма
 - Часть речи
 - Значения грамматических категорий
 - Тип отношения
- Элементы:
 - i -й элемент стека
 - i -й элемент буфера
 - Левое отношение элемента стека
 - Правое отношение элемента стека

Long Short-Term Memory

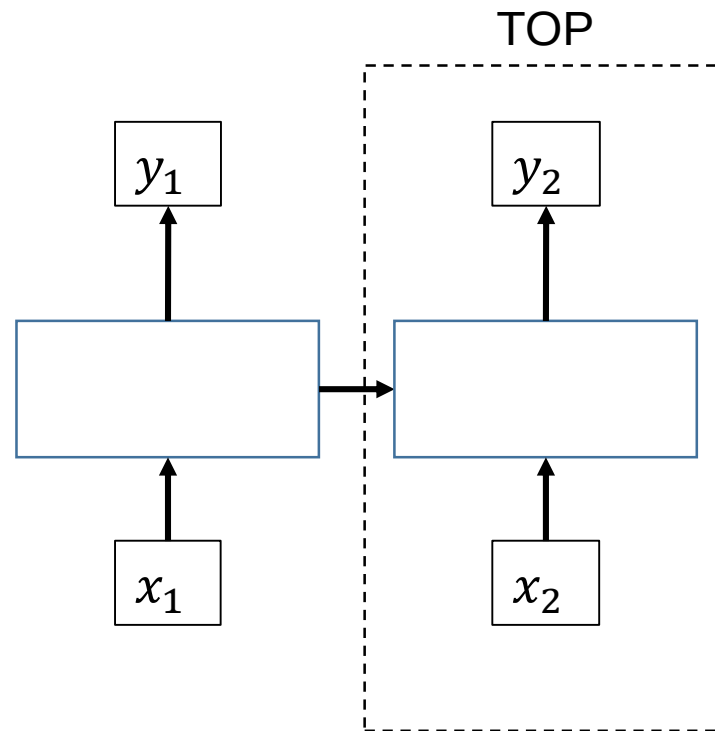
- $i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i)$
- $f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f)$
- $c_t = f_t \odot c_{t-1} + \tanh \odot (W_{cx}x_t + W_{ch}h_{t-1} + b_c)$
- $o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o)$
- $h_t = o_t \odot \tanh(c_t)$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



Stack Long Short-Term Memory

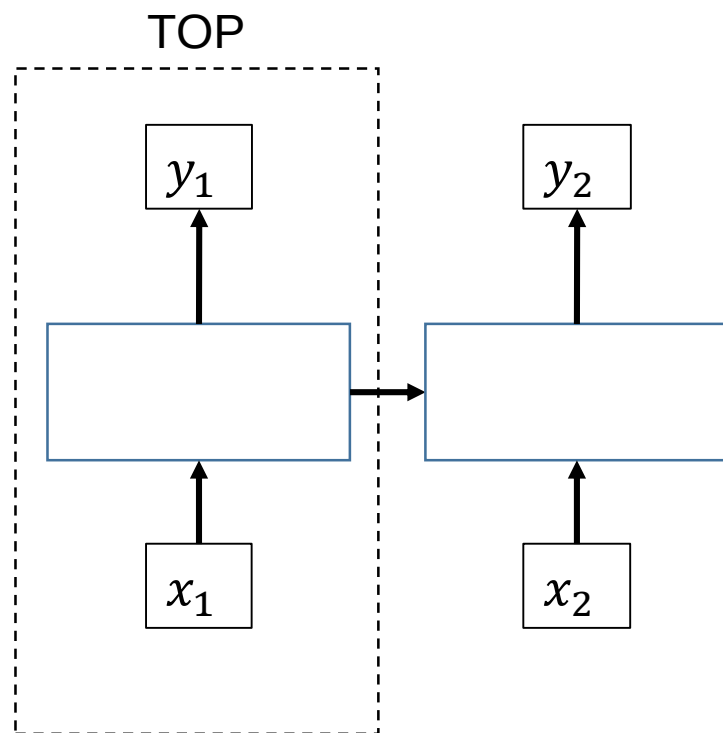


Stack Long Short-Term Memory

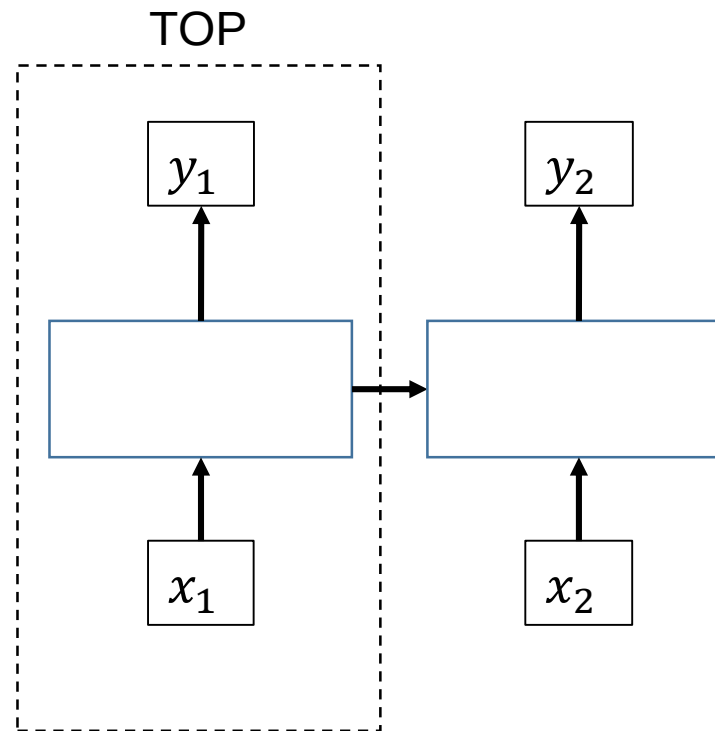


POP

Stack Long Short-Term Memory

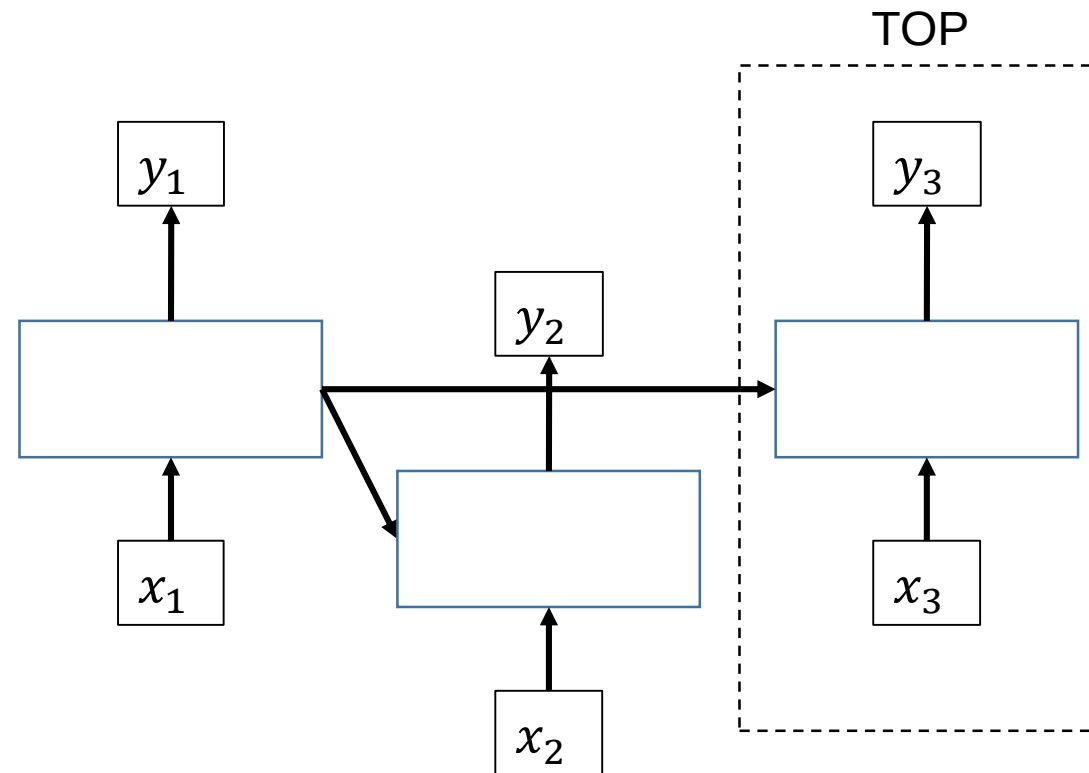


Stack Long Short-Term Memory

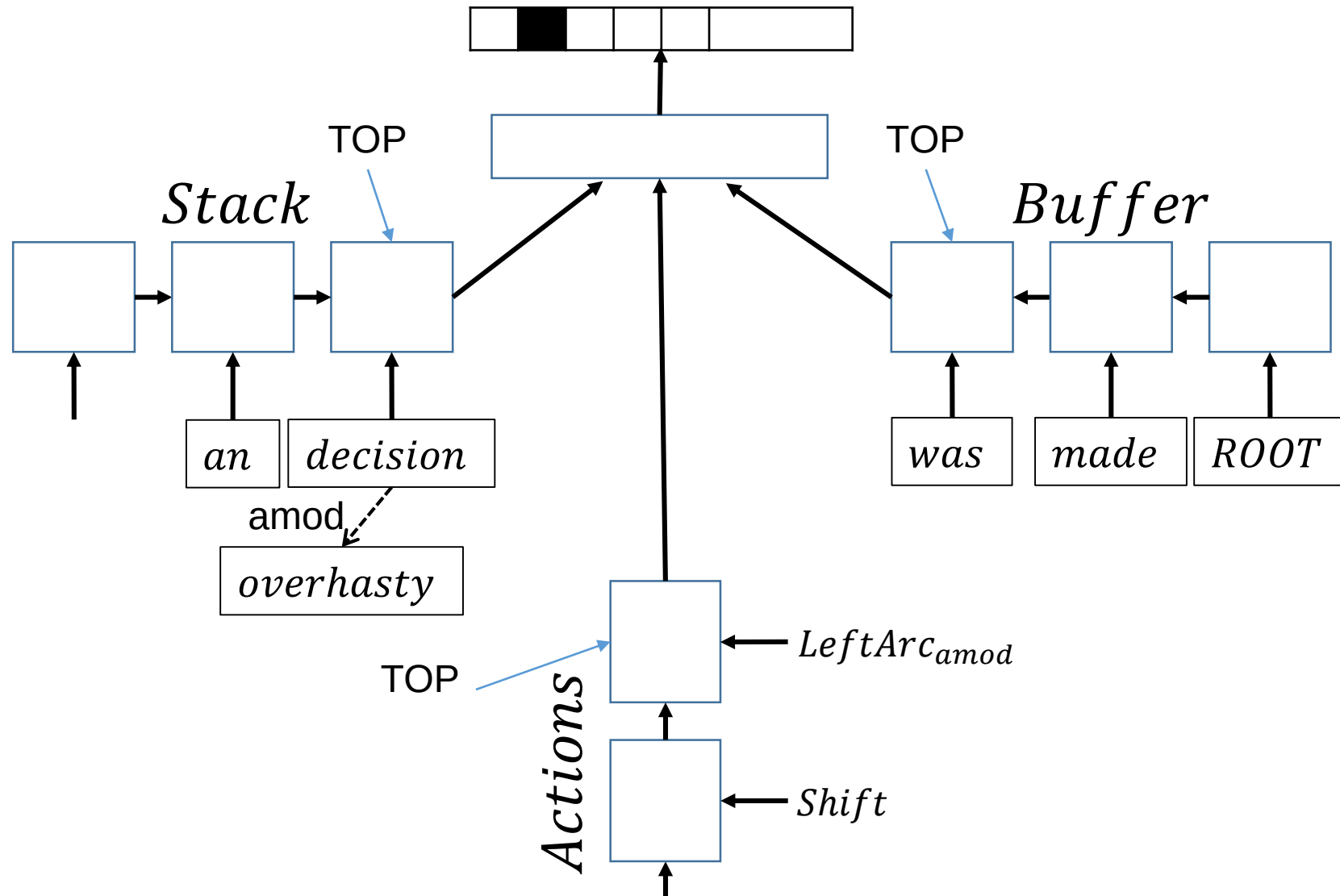


PUSH x_3

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

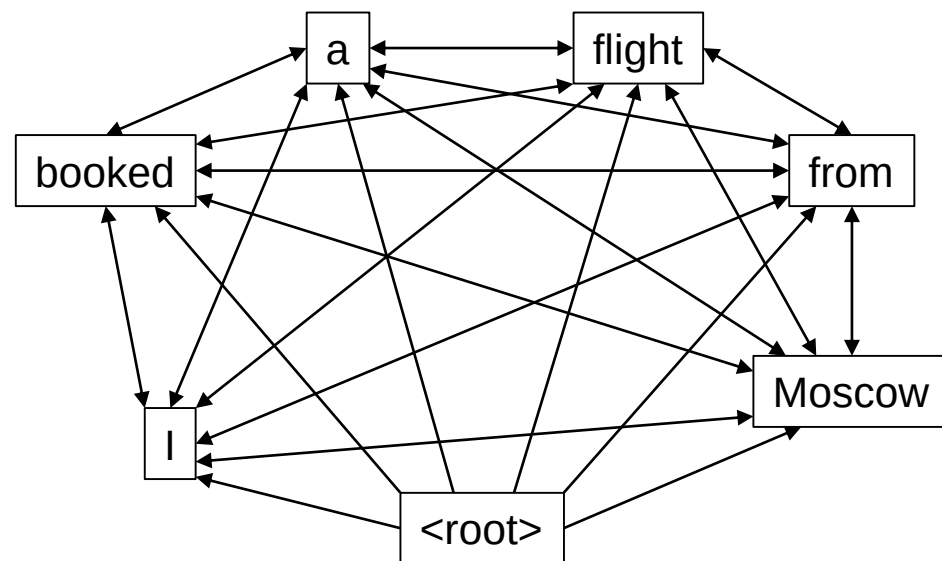
- Labeled attachment score (LAS):
 - Отношение количества правильно построенных дуг к общему количеству дуг
- Labeled exact match (LEM):
 - Отношение количества правильно построенных деревьев к общему количеству деревьев
- Unlabeled attachment score/exact match (UAS/UEM):
 - Те же метрики, но дуги рассматриваются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

Оценка качества

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

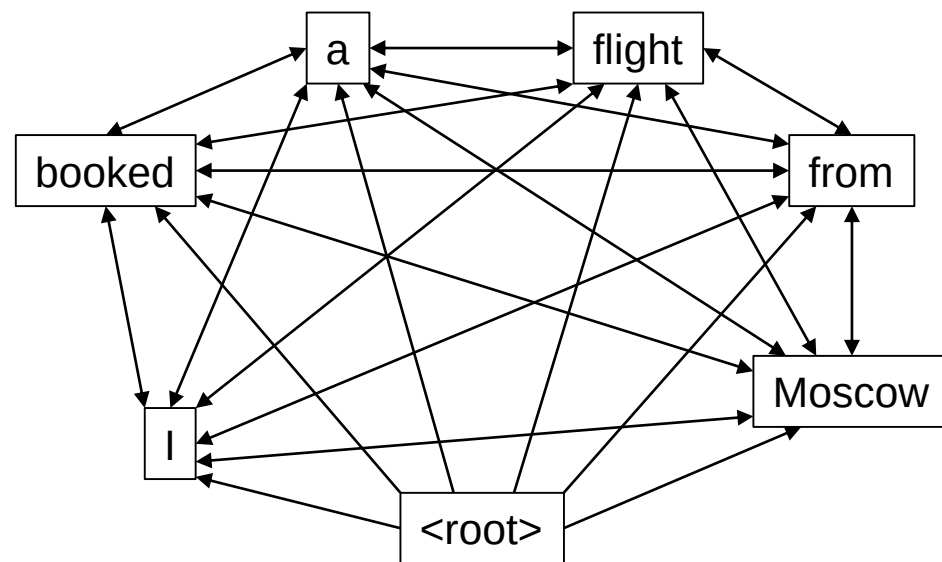
Графовые методы

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



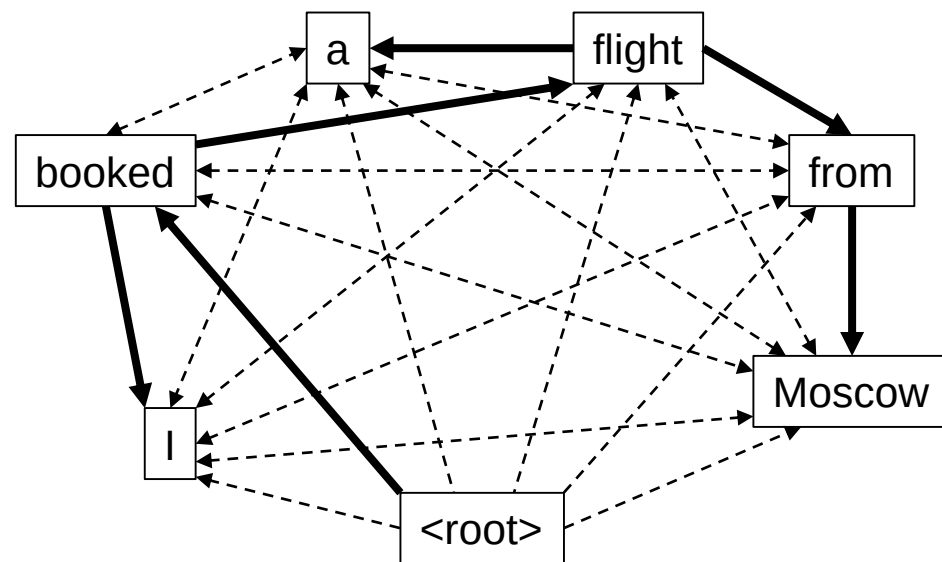
Графовые методы

- 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 остовное дерево (MST)

- Задача синтаксического анализа формулируется как задача нахождения максимального остовного дерева (MST) в связанном взвешенном графе
 - Остовное дерево графа – подграф, который:
 - Включает все вершины графа
 - Ациклический
 - Связный



Алгоритм Эдмондса

- На входе:
 - Ориентированный граф $D = \langle V, E \rangle$
 - Вершина $r \in V$ – корень
 - вещественные значения весов $w(e), \forall e \in E$
- На выходе:
 - Остовное ориентированное дерево A минимального веса $w(A) = \sum_{e \in A} w(e)$ с корнем r

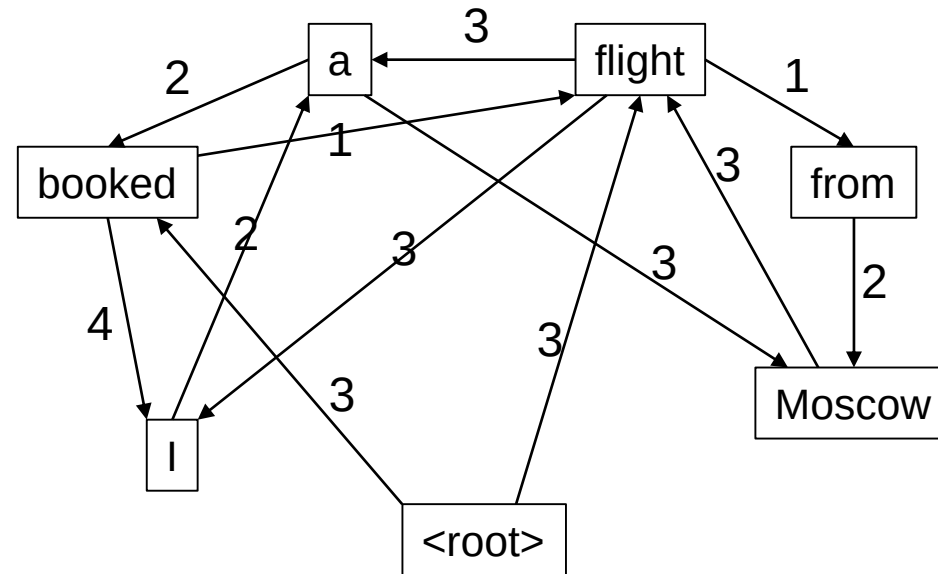
Алгоритм Эдмондса

- Пусть $f(\langle V, E \rangle, r, w)$ возвращает ориентированное дерево минимального веса с корнем в r
 1. Удалить все $(u, r) \in E$
 2. $\pi(v) = \arg \min_{u \in V} w((u, v)) \quad \forall v \in V$
 3. $P = \{(\pi(v), v) | v \in V \setminus \{r\}\}$. Если в P нет циклов, то P – искомое дерево
 4. C – произвольный цикл в P . Построим новый граф $D' = \langle V', E' \rangle$. $V' = \{v | v \in V, v \notin C\} \cup \{v_C\}$. E' построим так:
 - $(u, v) \in E, u \notin C, v \in C \Rightarrow$ добавляем в E' ребро $e = (u, v_C)$,
 $w'(e) = w(u, v) - w(\pi(v), v)$
 - $(u, v) \in E, u \in C, v \notin C \Rightarrow$ добавляем в E' ребро $e = (v_C, v)$,
 $w'(e) = w(u, v)$
 - $(u, v) \in E, u \notin C, v \notin C \Rightarrow$ добавляем в E' ребро $e = (u, v)$,
 $w'(e) = w(u, v)$
 5. $A' = f(\langle V', E' \rangle, r, w')$

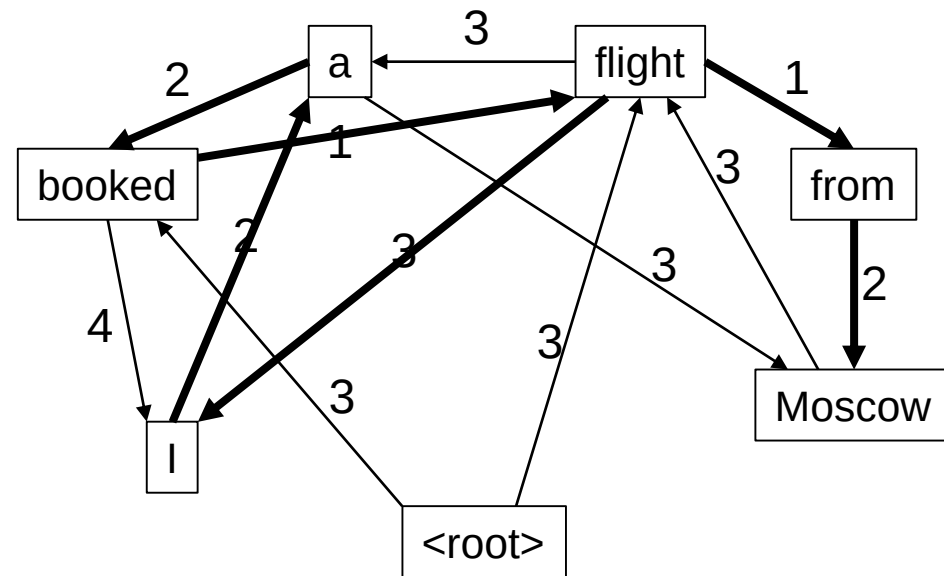
Алгоритм Эдмондса

- A' – ориентированное дерево, поэтому каждая $v \in V', v \neq r$ имеет одно входящее ребро (u, v)
- $(u, v_C) \in A'$ соответствует ребру $(u, v) \in E, v \in C$
- Удалим ребро $(\pi(v), v)$ из C
- Построим P' - искомое дерево:
 - Добавим в P' ребра из C
 - Добавим в P' все ребра, соответствующие ребрам из A'

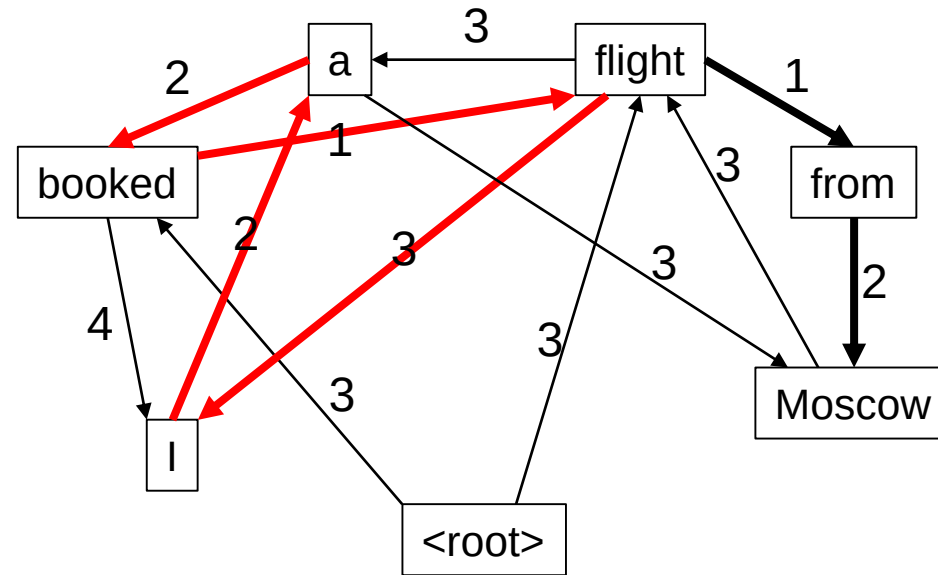
Алгоритм Эдмондса



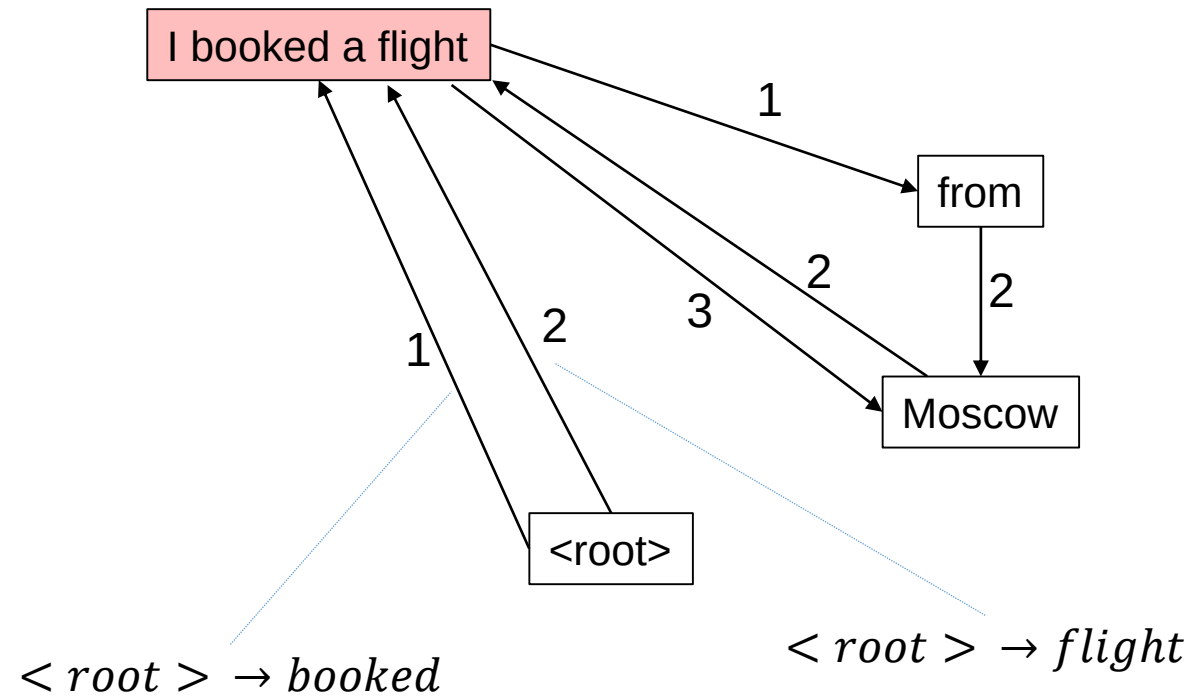
Алгоритм Эдмондса



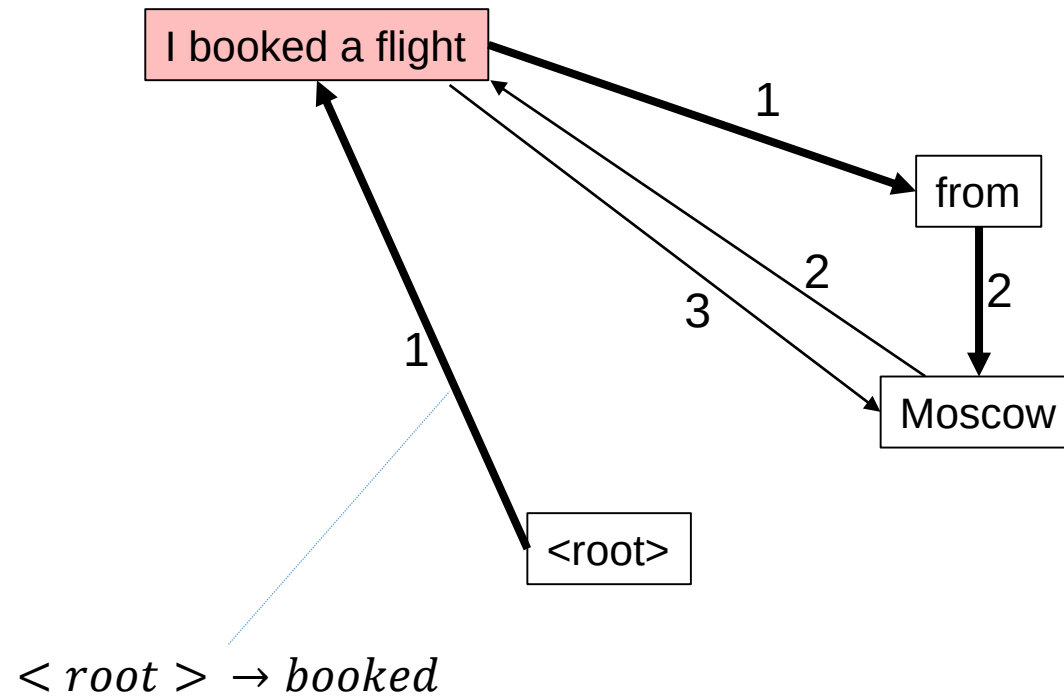
Алгоритм Эдмондса



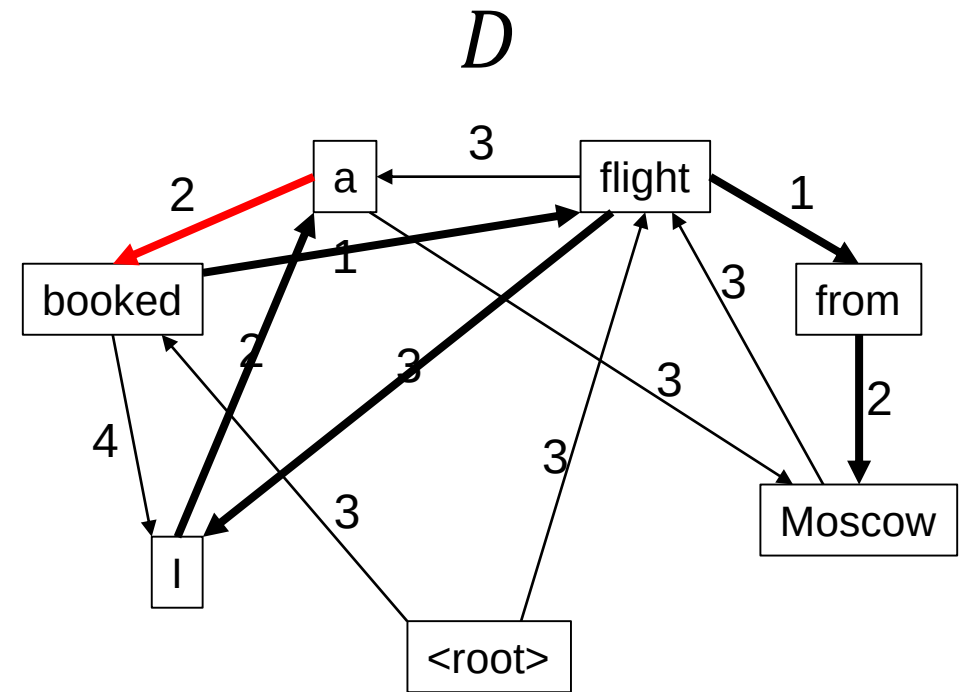
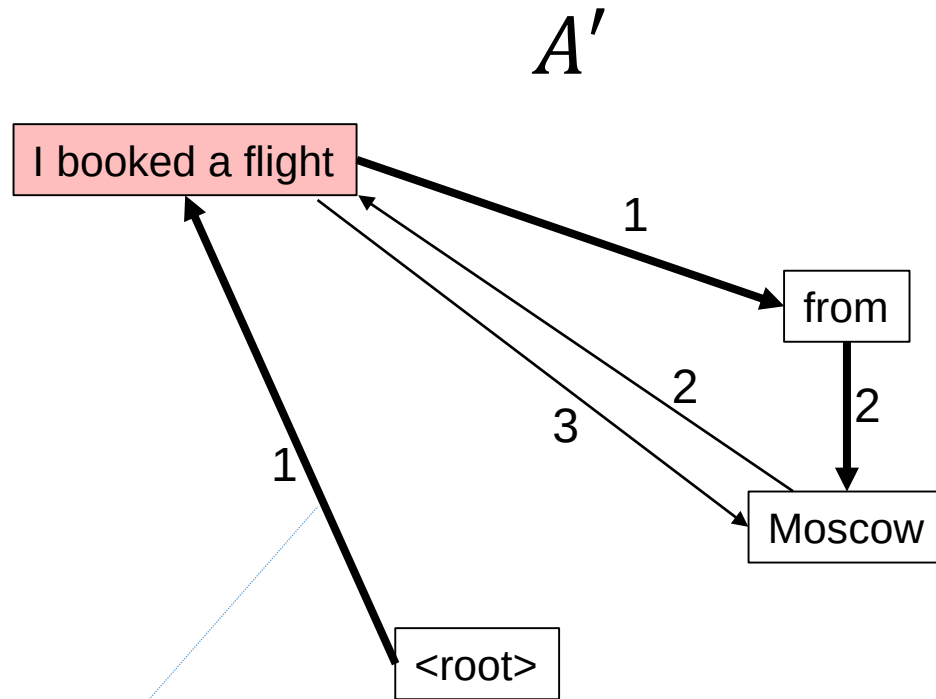
Алгоритм Эдмондса



Алгоритм Эдмондса

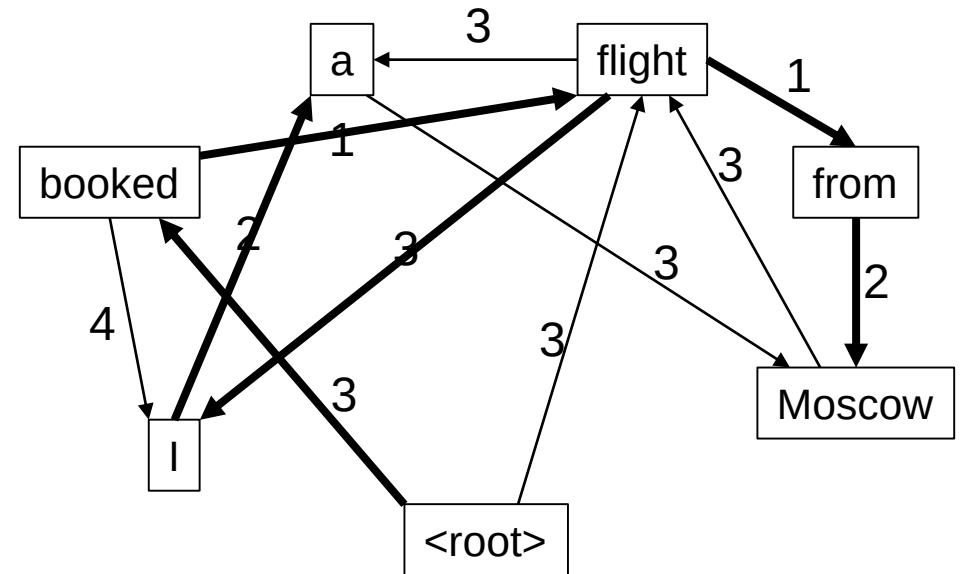
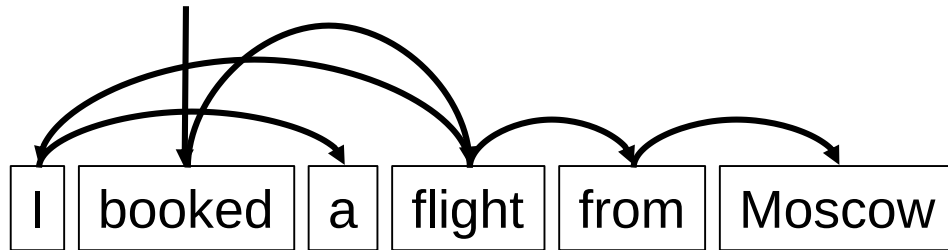


Алгоритм Эдмондса

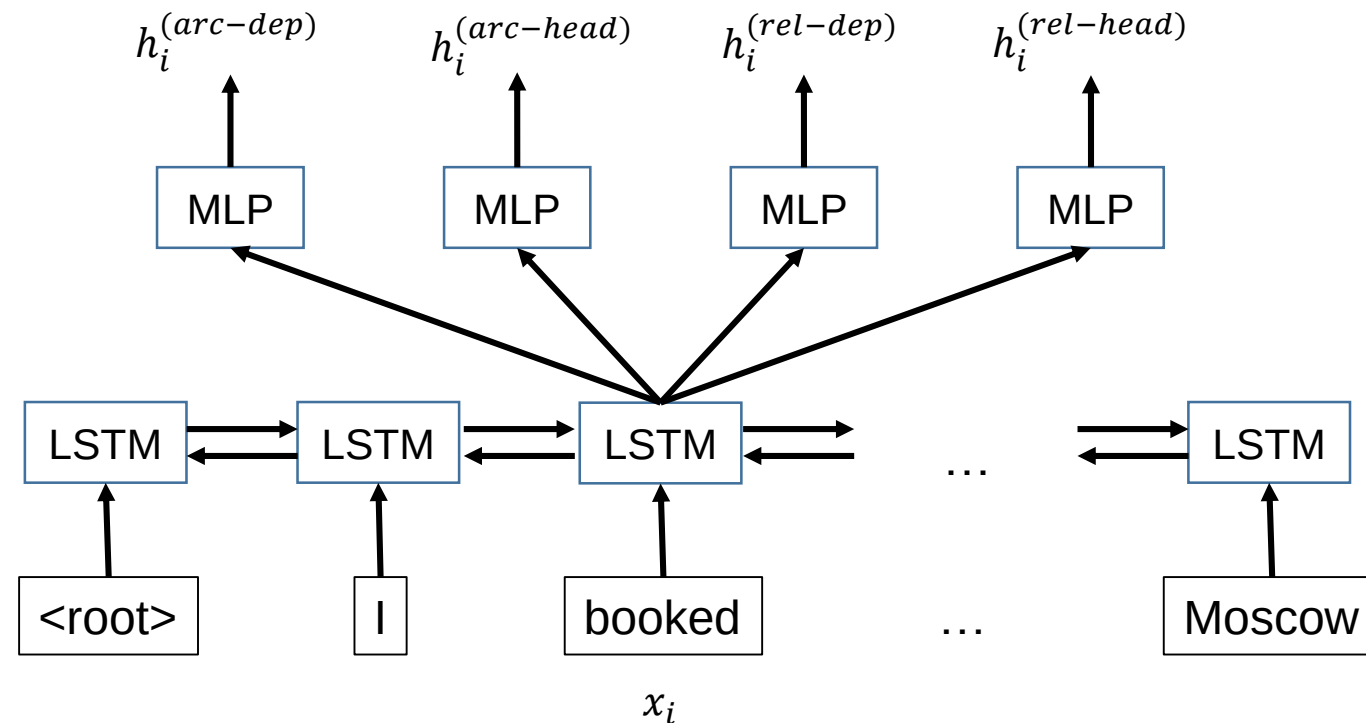


Удалим ребро $(\pi(v), v)$ из C

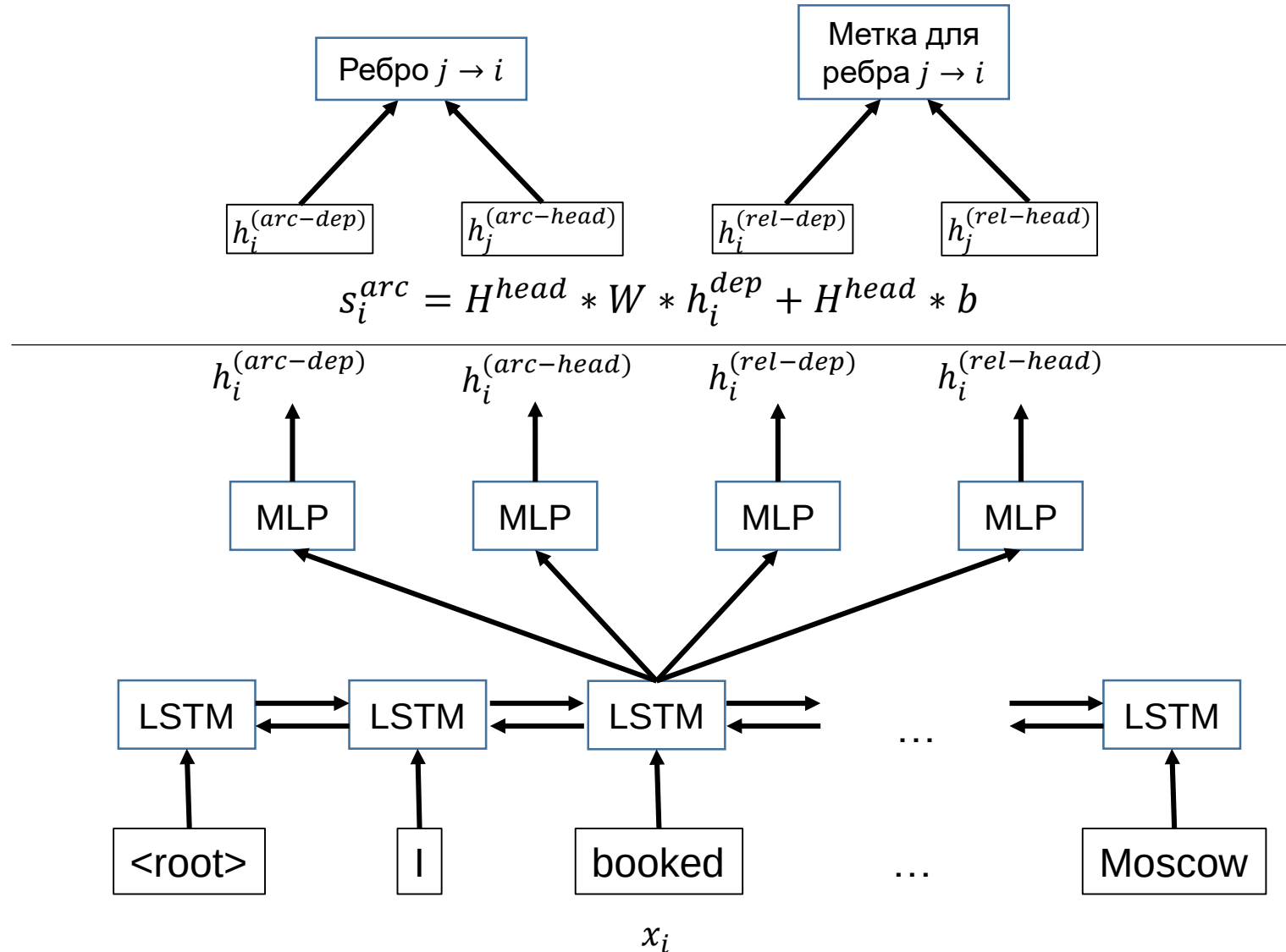
Алгоритм Эдмондса



Biaffine graph-based dependency parser



Biaffine graph-based dependency parser



Следующая лекция

Лексическая семантика