#### LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammar

# LR Parsing

Kilian Evang

Seminar für Sprachwissenschaft Universität Tübingen

2007-01-24

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

## Outline

Introduction

Building an LR Parser

Using an LR Parser

Determinism

LR(k) grammars

LR Parsing

Kilian Evang

ntroduction Building Jsing

Determinism

LR(k) grammars

Summary

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

### bottom-up

- directional (consumes input from left to right)
- related to Earley parsing
- handles left recursion
- handles *ɛ*-rules in its more sophisticated form
- fast parses in linear time
- deterministic

#### LR Parsing

Kilian Evang

Introduction

Building

Jsing

Determinism

\_R(k) grammars

Summary

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 のへで

### bottom-up

### directional (consumes input from left to right)

- related to Earley parsing
- handles left recursion
- handles \varepsilon-rules in its more sophisticated form
- fast parses in linear time
- deterministic

#### LR Parsing

Kilian Evang

Introduction

Building

Jsing

Determinism

\_R(k) grammars

Summary

### bottom-up

- directional (consumes input from left to right)
- related to Earley parsing
- handles left recursion
- handles  $\varepsilon$ -rules in its more sophisticated form
- fast parses in linear time
- deterministic

#### LR Parsing

Kilian Evang

Introduction

Building

Jsing

Determinism

\_R(k) grammars

Summary

▲□▶▲□▶▲□▶▲□▶ □ のQ@

- bottom-up
- directional (consumes input from left to right)
- related to Earley parsing
- handles left recursion
- handles *ɛ*-rules in its more sophisticated form
- fast parses in linear time
- deterministic

#### LR Parsing

Kilian Evang

Introduction

Building

Jsing

Determinism

\_R(k) grammars

Summary

- bottom-up
- directional (consumes input from left to right)
- related to Earley parsing
- handles left recursion
- handles *\varepsilon*-rules in its more sophisticated form
- fast parses in linear time
- deterministic

#### LR Parsing

Kilian Evang

Introduction

Building

Jsing

Determinism

LR(k) grammars

Summary

▲□▶▲□▶▲□▶▲□▶ □ のQ@

- bottom-up
- directional (consumes input from left to right)
- related to Earley parsing
- handles left recursion
- handles *\varepsilon*-rules in its more sophisticated form
- fast parses in linear time
- deterministic

#### LR Parsing

Kilian Evang

Introduction

Sunaing

Jsing

Determinism

\_R(k) grammars

Summary

▲□▶▲□▶▲□▶▲□▶ □ のQ@

- bottom-up
- directional (consumes input from left to right)
- related to Earley parsing
- handles left recursion
- handles *\varepsilon*-rules in its more sophisticated form
- fast parses in linear time
- deterministic

#### LR Parsing

Kilian Evang

Introduction

. .

Determinism

LR(k) grammars

Summary

### Kilian Evang

▲□▶▲□▶▲□▶▲□▶ □ のQ@

Introduction Building Using Determinism LR(k) grammars Summary

LR Parsing

### The central task of the parser is to find handles in the input.

- Peter goes on a trip with his wife
- Peter goes PP with his wife
- ►  $4 + 3 \times 5$
- ▶ 4+15
- In parsing, a lot depends on finding the right handles as early as possible.

- The central task of the parser is to find handles in the input.
- Peter goes on a trip with his wife
- Peter goes PP with his wife
- ► 4+3×5
- ► 4+15
- In parsing, a lot depends on finding the right handles as early as possible.

LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars Summary

・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・
・

- The central task of the parser is to find handles in the input.
- Peter goes on a trip with his wife
- Peter goes PP with his wife
- ►  $4 + 3 \times 5$
- ► 4+15
- In parsing, a lot depends on finding the right handles as early as possible.

▲□▶▲□▶▲□▶▲□▶ □ のQ@

LR Parsing

Kilian Evang

- The central task of the parser is to find handles in the input.
- Peter goes on a trip with his wife
- Peter goes PP with his wife
- $\blacktriangleright 4 + 3 \times 5$
- ► 4+15
- In parsing, a lot depends on finding the right handles as early as possible.

▲□▶▲□▶▲□▶▲□▶ □ のQ@

LR Parsing

Kilian Evang

- The central task of the parser is to find handles in the input.
- Peter goes on a trip with his wife
- Peter goes PP with his wife
- $\blacktriangleright 4 + 3 \times 5$
- ► 4+15
- In parsing, a lot depends on finding the right handles as early as possible.

▲□▶▲□▶▲□▶▲□▶ □ のQ@

LR Parsing

Kilian Evang

- The central task of the parser is to find handles in the input.
- Peter goes on a trip with his wife
- Peter goes PP with his wife
- $\blacktriangleright 4 + 3 \times 5$
- ► 4+15
- In parsing, a lot depends on finding the right handles as early as possible.

▲□▶▲□▶▲□▶▲□▶ □ のQ@

LR Parsing

Kilian Evang

- A sequence of tokens can be a handle only if it could have been derived from some non-terminal in that position, which in turn could have been derived...and so on, going all the way back up to the start symbol.
- To take that into consideration, LR parsers use a top down component like the Earley parser.

▲□▶▲□▶▲□▶▲□▶ □ のQ@

#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

R(k) grammars

Summary



#### LR Parsing

Kilian Evang





#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

・ロト・国ト・モート ヨー うへの





▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへで

#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary



#### LR Parsing

Kilian Evang

Introduction

Building

Jsing

Determinism

\_R(k) grammars

Summary



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ ─臣 ─のへで



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□ → ◆□ → ◆ 三 → ◆ 三 → ● へ ○



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

R(k) grammars

Summary

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ ─臣 ─のへで



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ ─臣 ─のへで



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

◆□ ▶ ◆□ ▶ ◆ 三 ▶ ◆ 三 ● ○ ○ ○ ○



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

くりょう 小田 マイビット 日 うくの



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

・ロト・日本・日本・日本・日本・日本



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

・ロト・西ト・モート ヨー うへの



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

・ロト・日本・日本・日本・日本・日本



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ ─臣 ─のへで



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□▶ ◆□▶ ◆三▶ ◆三▶ ◆□▶ ◆□▶



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ = ・ の < ()



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

・ロト・国ト・モート ヨー うへの



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

くりょう 小田 マイビット 日 うくの



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□ → ◆□ → ◆三 → ◆三 → ● ◆ ● ◆ ●



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ → のへぐ


#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ → のへぐ



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ → のへぐ



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□▶ ◆□▶ ◆三▶ ◆三▶ ◆□▶ ◆□▶



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

R(k) grammars

Summary

◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ = ・ の < ()



#### LR Parsing

#### Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary

◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ → の < ⊙



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

LR(k) grammars

Summary

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○三 のへで



#### LR Parsing

Kilian Evang

Introduction

Building

Using

Determinism

\_R(k) grammars

Summary



LR Parsing

Kilian Evang

Building Using

Determinism LR(k) grammars

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = ∽ Q Q @



LR Parsing

Kilian Evang

Determinism

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = ∽ Q Q ?



LR Parsing

Kilian Evang

ntroduction Building

Jsing

Determinism

R(k) grammars

Summary



LR Parsing

Kilian Evang

Determinism

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = ∽ Q Q ?



LR Parsing

Kilian Evang

Building

Determinism LR(k) grammars

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○三 ● ● ●



LR Parsing

Kilian Evang

Determinism

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = ∽ Q Q ?



LR Parsing

Kilian Evang

ntroduction Building

Jsing

Determinism

R(k) grammars

Summary

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = ∽ Q Q ?



LR Parsing

Kilian Evang

Building Using Determinism

.R(k) grammars

Summary



LR Parsing

Kilian Evang

ntroduction Building

Jsing

Determinism

R(k) grammars

Summary



LR Parsing

Kilian Evang

Building Using Determinism

.R(k) grammars

Summary



LR Parsing

Kilian Evang

ntroduction Building

Jsing

Determinism

R(k) grammars

Summary



LR Parsing

Kilian Evang

Determinism

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = ∽ Q Q ?



LR Parsing

Kilian Evang

ntroduction Building

Jsing

Determinism

R(k) grammars

Summary

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = ∽ Q Q ?



LR Parsing

Kilian Evang

Building

Determinism LR(k) grammars

・ロト・日本・日本・日本・日本・日本

### Conflicts caused by non-LR(0) grammars



▲□▶▲□▶▲□▶▲□▶ ▲□ ● ●

LR Parsing

Kilian Evang

### An inadequate handle recognizer



LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars

Summary

◆□ → ◆□ → ◆ 三 → ◆ 三 → ● へ ○

#### ▶ The 0 in *LR*(0) stands for 0 symbols lookahead.

- Grammars which are not LR(0) may still be LR(1), or LR(k) with some higher k.
- LR parsers can be built for such grammars by introducing lookahead.
- Lookahead makes reductions depend on what follows the handle.

LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars

▲□▶ ▲圖▶ ▲臣▶ ▲臣▶ 臣 のへぐ

- ► The 0 in *LR*(0) stands for 0 symbols lookahead.
- Grammars which are not LR(0) may still be LR(1), or LR(k) with some higher k.
- LR parsers can be built for such grammars by introducing lookahead.
- Lookahead makes reductions depend on what follows the handle.

LR Parsing Kilian Evang

Introduction Building Using Determinism LR(k) grammars

・ロト・西ト・モート ヨー うへの

- The 0 in LR(0) stands for 0 symbols lookahead.
- Grammars which are not LR(0) may still be LR(1), or LR(k) with some higher k.
- LR parsers can be built for such grammars by introducing lookahead.
- Lookahead makes reductions depend on what follows the handle.

LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars

Summary

- The 0 in LR(0) stands for 0 symbols lookahead.
- Grammars which are not LR(0) may still be LR(1), or LR(k) with some higher k.
- LR parsers can be built for such grammars by introducing lookahead.
- Lookahead makes reductions depend on what follows the handle.

ntroduction Building Jsing Determinism

LR Parsing

Kilian Evang

LR(k) grammars

Summary

# A non-deterministic handle recognizer with lookahead



LR Parsing

Kilian Evang

LR(k) grammars

### A deterministic handle recognizer with lookahead



#### LR Parsing

Kilian Evang

Introduction

Building

Jsing

Determinism

LR(k) grammars

Summary

◆□ ▶ ◆□ ▶ ◆ 三 ▶ ◆ 三 ● ● ● ●

### A table representation of our LR(1) parser

	n	-	(	)	#	s	Е	т
1	s5	е	s7	е	е	s2	s6	s4
2	e	е	е	е	s3			
3/acc								
4	e e	r4	e	e	r4			
5	e	r5	e	e	r5			
6	e	s8	e	е	r2			
7	s11	e	s14	e	е		s12	s10
8	s5	e	s7	e	е			s9
9	e	r3	e	e	r3			
10	e	r4	e	r4	е			
11	e	r5	e	r5	е			
12	e	s15	e	s13	е			
13	e	r6	e	е	r6			
14	s11	е	s14	e	е		s17	s10
15	s11	e	s14	e	e			s16
16	e	r3	e	r3	е			
17	e	s15	e	s18	e			
18	е	r6	е	r6	е			

LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars Summary

・ロト・西ト・モート ヨー うへの

# Limitations of LR(k) parsing

Although an LR(2) parser is more powerful than an LR(1) in that it can handle some grammars that the other cannot, the emphasis is on some. (Grune + Jacobs)

LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars Summary

▲□▶ ▲圖▶ ▲≣▶ ▲≣▶ ▲国 ● のへで

- Any LR(k) grammar with k > 1 can be transformed into an LR(k-1) grammar, thus to LR(1), but not always to LR(0).
- A language that has an LR(1) grammar is called deterministic.
- Many programming languages are LR(1). Well-known exceptions are C++, Perl and Python.

LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars

Summary

- Any LR(k) grammar with k > 1 can be transformed into an LR(k-1) grammar, thus to LR(1), but not always to LR(0).
- A language that has an LR(1) grammar is called deterministic.
- Many programming languages are LR(1). Well-known exceptions are C++, Perl and Python.

LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars

Summary

Any LR(k) grammar with k > 1 can be transformed into an LR(k-1) grammar, thus to LR(1), but not always to LR(0).

LR Parsing

Kilian Evang

LR(k) grammars

- A language that has an LR(1) grammar is called deterministic.
- Many programming languages are LR(1). Well-known exceptions are C++, Perl and Python.

▲□▶▲□▶▲□▶▲□▶ □ のQ@

Any LR(k) grammar with k > 1 can be transformed into an LR(k-1) grammar, thus to LR(1), but not always to LR(0).

LR Parsing

Kilian Evang

LR(k) grammars

- A language that has an LR(1) grammar is called deterministic.
- Many programming languages are LR(1). Well-known exceptions are C++, Perl and Python.

▲□▶▲□▶▲□▶▲□▶ □ のQ@

#### An LR(1) parser has the immediate error detection property: It will stop at the first incorrect token.

The parsing tables of LR parsers with lookahead quickly assume gargantuan size. A way around that is LALR parsing, a very widely used variation of LR.



Kilian Evang

ntroductior

Building

Jsing

Determinism

LR(k) grammars

Summary

▲□▶▲□▶▲□▶▲□▶ □ のQ@
# Some more properties of LR(k) parsing (cont.)

- An LR(1) parser has the immediate error detection property: It will stop at the first incorrect token.
- The parsing tables of LR parsers with lookahead quickly assume gargantuan size. A way around that is LALR parsing, a very widely used variation of LR.

Kilian Evang

ntroductior

Building

Jsing

Determinism

LR(k) grammars

Summary

▲□▶▲□▶▲□▶▲□▶ □ のQ@

## Summary

- You have seen how to construct and use LR(k) parsers, the strongest deterministic parsers possible and possibly the strongest linear-time parsers known.
- Key concepts to bear in mind: Handle recognition, items that call for specific reductions, use of nondeterministic and deterministic automata, lookahead

LR Parsing

Kilian Evang

Introduction Building Using Determinism LR(k) grammars

Summary

## Summary

- You have seen how to construct and use LR(k) parsers, the strongest deterministic parsers possible and possibly the strongest linear-time parsers known.
- Key concepts to bear in mind: Handle recognition, items that call for specific reductions, use of nondeterministic and deterministic automata, lookahead

Kilian Evang

Introduction Building Using Determinism LR(k) grammars Summary

## Summary

- You have seen how to construct and use LR(k) parsers, the strongest deterministic parsers possible and possibly the strongest linear-time parsers known.
- Key concepts to bear in mind: Handle recognition, items that call for specific reductions, use of nondeterministic and deterministic automata, lookahead

Kilian Evang

Introduction Building Using Determinism LR(k) grammars Summary

#### References

LR Parsing

Kilian Evang

Building

Using

Determinism

\_R(k) grammars

Summary

Dick Grune & Ceriel Jacobs (1990). *Parsing Techniques.* Amsterdam. pp. 200-213 Content, examples and all diagrams taken from there.

◆□▶ ◆□▶ ◆目▶ ◆目▶ 目 のへぐ