Parallel Parsing

The Implementation of a Parallel LL Parser Generator

William Henrich Due

30rd June 2023

KØBENHAVNS UNIVERSITET

LL parsing

Here is a LL(1) grammar.

$$1)\ T \to R \qquad 2)\ T \to aTc \qquad 3)\ R \to \varepsilon \qquad 4)\ R \to bR$$

LL parsing

Here is a LL(1) grammar.

$$1)\ T \to R \qquad 2)\ T \to aTc \qquad 3)\ R \to \varepsilon \qquad 4)\ R \to bR$$

$$(abc, T, ())$$

LL parsing

Here is a LL(1) grammar.

$$1)\ T \to R \qquad 2)\ T \to aTc \qquad 3)\ R \to \varepsilon \qquad 4)\ R \to bR$$

$$(abc, T, ()) \vdash (abc, aTc, 2)$$

LL parsing

Here is a LL(1) grammar.

$$1)\ T \to R \qquad 2)\ T \to aTc \qquad 3)\ R \to \varepsilon \qquad 4)\ R \to bR$$

$$(abc, T, ()) \vdash (abc, aTc, 2) \vdash (bc, Tc, 2)$$

LL parsing

Here is a LL(1) grammar.

$$1)\ T \rightarrow R \qquad 2)\ T \rightarrow aTc \qquad 3)\ R \rightarrow \varepsilon \qquad 4)\ R \rightarrow bR$$

$$(abc, T, ()) \vdash (abc, aTc, 2) \vdash (bc, Tc, 2) \vdash (bc, Rc, (2, 1))$$

LL parsing

Here is a LL(1) grammar.

$$1)\ T \to R \qquad 2)\ T \to aTc \qquad 3)\ R \to \varepsilon \qquad 4)\ R \to bR$$

$$\begin{aligned}(abc, T, ()) &\vdash (abc, aTc, 2) \vdash (bc, Tc, 2) \vdash (bc, Rc, (2,1)) \\ &\vdash (bc, bRc, (2,1,4))\end{aligned}$$

LL parsing

Here is a LL(1) grammar.

$$1)\;\; T \to R \qquad 2)\;\; T \to aTc \qquad 3)\;\; R \to \varepsilon \qquad 4)\;\; R \to bR$$

$$\begin{aligned}(abc, T, ()) \vdash (abc, aTc, 2) \vdash (bc, Tc, 2) \vdash (bc, Rc, (2, 1))\\ \vdash (bc, bRc, (2, 1, 4)) \vdash (c, Rc, (2, 1, 4))\end{aligned}$$

LL parsing

Here is a LL(1) grammar.

$$1)\ T \to R \qquad 2)\ T \to aTc \qquad 3)\ R \to \varepsilon \qquad 4)\ R \to bR$$

$$\begin{aligned}(abc, T, ()) &\vdash (abc, aTc, 2) \vdash (bc, Tc, 2) \vdash (bc, Rc, (2,1)) \\ &\vdash (bc, bRc, (2,1,4)) \vdash (c, Rc, (2,1,4)) \\ &\vdash (c, c, (2,1,4,3))\end{aligned}$$

LL parsing

Here is a LL(1) grammar.

$$1)\ T \to R \qquad 2)\ T \to aTc \qquad 3)\ R \to \varepsilon \qquad 4)\ R \to bR$$

$$\begin{aligned}(abc, T, ()) &\vdash (abc, aTc, 2) \vdash (bc, Tc, 2) \vdash (bc, Rc, (2,1)) \\ &\vdash (bc, bRc, (2,1,4)) \vdash (c, Rc, (2,1,4)) \\ &\vdash (c, c, (2,1,4,3)) \vdash (\varepsilon, \varepsilon, (2,1,4,3))\end{aligned}$$

Accepted! the string "*abc*" can be parsed.

LLP parsing

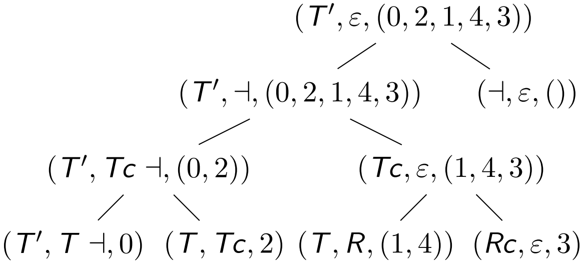
Augment the grammar.

$$0)\ \ T' \rightarrow\ \vdash T \dashv$$

Now we parse the string "$\vdash abc \dashv$" instead.

$(\varepsilon, \vdash)$	$(\vdash, a)$	(a, b)	(b, c)	$(c, \dashv)$
$(T', T \dashv, 0)$	$(T, Tc, 2)$	$(T, R, (1, 4))$	$(Rc, \varepsilon, 3)$	$(\dashv, \varepsilon, ())$

LLP parsing



$(T', \varepsilon, (0, 2, 1, 4, 3))$

$(T', \dashv, (0, 2, 1, 4, 3))$ $(\dashv, \varepsilon, ())$

$(T', Tc \dashv, (0, 2))$ $(Tc, \varepsilon, (1, 4, 3))$

$(T', T \dashv, 0)$ $(T, Tc, 2)$ $(T, R, (1, 4))$ $(Rc, \varepsilon, 3)$

Liste med punkter

- Punkt 1
- Punkt 2
    - Punkt 2.1
    - Punkt 2.2
- Punkt 3

Slide med 2 kolonner

- Punkt 1
- Punkt 2

- Punkt 3
- Punkt 4
- Punkt 5

Slide med en kolonne og et billede

- Punkt 1
- Punkt 2

Slide med stort billede

Et lille mærke med en vilkårlig tekst

Dette er et stort mærke. Her kan man f.eks. have en liste

- Punkt 1
- Punkt 2
- Punkt 3
- Punkt 4

> “
>
> Prediction is very difficult, especially if it’s about the future.

Niels Bohr

Research	Forskning
Education	Uddannelse
Exchange of knowdledge	Forskningsformidling