Lecture 22 - Dec. 1

Syntactic Analysis

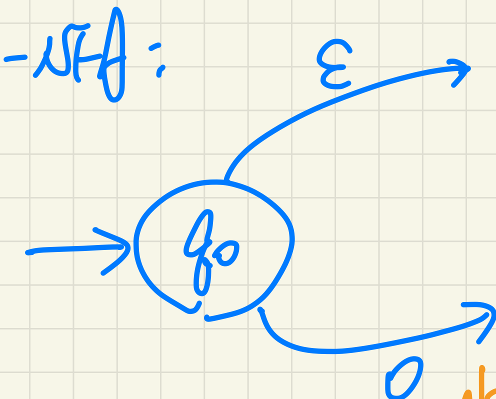
Canonical Collection vs. Subset States
Algorithms: closure, goto

Announcements

- **Project** final submission guideline to be released on Friday

- **Review session** on Thursday, December 8?

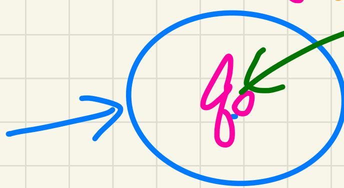
Input ε-NFA:



closure

output DFA:

$ECLOSE(q_0) = \{ \ldots \}$



subset state

CC Construction: closure

```
ALGORITHM: closure
  INPUT: CFG G = (V, Σ, R, S), a set s of LR(1) items
  OUTPUT: a set of LR(1) items
PROCEDURE:
  lastS := ∅
  while (lastS ≠ s):
    lastS := s
    for [A → ··· • C δ, a] ∈ s:
      for C → γ ∈ R:
        for b ∈ FIRST(δa):
          s := s ∪ { [ C → •γ, b] }
  return s
```

keep growing the output set s until nothing new can be added.

is the "follow" of C

1. What has been recognized? ...
2. What's expected to be recognized next? C

all alternatives returning to C.

$b \in FIRST(\delta a)$

Q. Why not $b \in FIRST(\delta)$?

$\varepsilon \in FIRST(\delta)$

∵ δ might be nullable

[C → • γ, b]

new LR(1) item to be added to the closure.

Analogy: ε-NFA to DFA

Subset construction (with ***lazy evaluation*** and *epsilon closures*) produces a ***DFA*** transition table.

	$d \in 0..9$	$s \in \{+,-\}$	.
$\{q_0, q_1\}$	$\{q_1, q_4\}$	$\{q_1\}$	$\{q_2\}$
$\{q_1, q_4\}$	$\{q_1, q_4\}$	$\varnothing$	$\{q_2, q_3, q_5\}$
$\{q_1\}$	$\{q_1, q_4\}$	$\varnothing$	$\{q_2\}$
$\{q_2\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$
$\{q_2, q_3, q_5\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$
$\{q_3, q_5\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$

starting set

For example, $\delta(\{q_0, q_1\}, d)$ is calculated as follows: $[d \in 0..9]$

$$\bigcup\{\text{ECLOSE}(q) \mid q \in \delta(q_0, d) \cup \delta(q_1, d)\}$$

set of subset states → a set of LR(1) items

CC Construction: CC_0

Calculate CC_0 of the following grammar.

Hint: Closure of the singleton set containing the parser's initial state.

1	*Goal*	→	*List*
2	*List*	→	*List Pair*
3		|	*Pair*
4	*Pair*	→	(*Pair*)
5		|	( )

```
ALGORITHM: closure
  INPUT: CFG G = (V, Σ, R, S), a set s of LR(1) items
  OUTPUT: a set of LR(1) items
PROCEDURE:
  lastS := ∅
  while (lastS ≠ s) :
    lastS := s
    for [A → ··· • C δ, a] ∈ s:
      for C → γ ∈ R:
        for b ∈ FIRST(δa):
          s := s ∪ { [C → •γ, b] }
  return s
```

Initial subset state of the parser

ε-NFA q_0 → DFA ECLOSE(q_0)

→ CC_0

parser's initial state:

$\{[Goal \to \bullet List,\ eof]\}$ = input to closure

CC Construction: CC_0 Step 1

( )   (( ))   (( ))( )

1	$Goal$	$\rightarrow List$
2	$List$	$\rightarrow List\ Pair$
3		$\mid\ Pair$
4	$Pair$	$\rightarrow \underline{(}\ Pair\ \underline{)}$
5		$\mid\ \underline{(}\ \underline{)}$

(0) [Goal → • List, eof]   initial parser state

Hint 1. How is $[A \rightarrow \beta \bullet C \delta, a]$ instantiated?

Goal ε List ε eof

Hint 2. What are $C \rightarrow \gamma \in R$?

→ List → List Pair   List → Pair

Hint 3. $\textbf{FIRST}(\delta a)$ = FIRST(ε eof) = FIRST(eof) = { eof }

Two new LR(1) items:

1. [List → • List Pair, eof]
2. [List → • Pair, eof]

How should s be extended?

```
for [A → ··· • C δ, a] ∈ s:
  for C → γ ∈ R:
    for b ∈ FIRST(δa):
      s := s ∪ { [C → •γ, b] }
```

$$CC_0 = \left\{ \begin{array}{ccc} [Goal \rightarrow \bullet\, List, \texttt{eof}] & [List \rightarrow \bullet\, List\ Pair, \texttt{eof}] & [List \rightarrow \bullet\, List\ Pair, \underline{(}] \\ [List \rightarrow \bullet\, Pair, \texttt{eof}] & [List \rightarrow \bullet\, Pair, \underline{(}] & [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)}, \texttt{eof}] \\ [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)}, \underline{(}] & [Pair \rightarrow \bullet\, \underline{(}\ \underline{)}, \texttt{eof}] & [Pair \rightarrow \bullet\, \underline{(}\ \underline{)}, \underline{(}] \end{array} \right\}$$

CC Construction: CC_0 — Step 2

1	*Goal*	$\rightarrow$	*List*
2	*List*	$\rightarrow$	*List Pair*
3		$\mid$	*Pair*
4	*Pair*	$\rightarrow$	(*Pair*)
5		$\mid$	( )

(0) [Goal $\rightarrow \bullet$ List, eof]

(1) [List $\rightarrow \bullet$ List Pair, eof]

(2) [List $\rightarrow \bullet$ Pair, eof]

Hint 1. How is $[\underbrace{A}_{\text{List}} \rightarrow \underbrace{\beta}_{\varepsilon} \bullet \underbrace{C}_{\text{Pair}}\ \underbrace{\delta}_{\varepsilon},\ \underbrace{a}_{\text{eof}}]$ instantiated?

Hint 2. What are $C \rightarrow \gamma \in R$? Pair → (Pair) Pair → ( )

Hint 3. $\text{FIRST}(\delta a)$ = FIRST(ε eof) = FIRST(eof) = { eof }

How should s be extended?

[Pair → • (Pair) , eof]

[Pair → • ( ) , eof]

```
for [A → ··· • C δ, a] ∈ s:
  for C → γ ∈ R:
    for b ∈ FIRST(δa):  ✓
      s := s ∪ { [C → •γ, b] }
```

$$CC_0 = \left\{ \begin{array}{lll} [Goal \rightarrow \bullet\, List,\ \texttt{eof}] & [List \rightarrow \bullet\, List\ Pair,\ \texttt{eof}] & [List \rightarrow \bullet\, List\ Pair,\ (] \\ [List \rightarrow \bullet\, Pair,\ \texttt{eof}] & [List \rightarrow \bullet\, Pair,\ (] & [Pair \rightarrow \bullet\, (\ Pair\ ),\ \texttt{eof}] \\ [Pair \rightarrow \bullet\, (\ Pair\ ),(] & [Pair \rightarrow \bullet\, (\ ),\ \texttt{eof}] & [Pair \rightarrow \bullet\, (\ ),(] \end{array} \right\}$$

CC Construction: CC_0 Step 3

(0) [Goal → • List, eof]

(1) [List → • List Pair, eof]

(2) [List → • Pair, eof]

(3) [Pair → • (Pair), eof]

(4) [Pair → • ( ), eof]

1	*Goal* → *List*
2	*List* → *List Pair*
3	| *Pair*
4	*Pair* → (*Pair*)
5	| ( )

List ε List Pair

Hint 1. How is $[A \rightarrow \beta \bullet C\ \delta, a]$ instantiated?

eof $\varepsilon \notin$ FIRST(Pair).

Hint 2. What are $C \rightarrow \gamma \in R$?

Hint 3. FIRST(δa) = FIRST(Pair eof) = { (}

[List → • List Pair, ( ]

[List → • Pair, ( ]

How should s be extended?

```
for [A → ··· • C δ, a] ∈ s:
  for C → γ ∈ R:
    for b ∈ FIRST(δa):
      s := s ∪ { [C → •γ, b] }
```

$$
CC_0 = \left\{
\begin{array}{lll}
[Goal \rightarrow \bullet\, List,\ \texttt{eof}] & [List \rightarrow \bullet\, List\ Pair,\ \texttt{eof}] & [List \rightarrow \bullet\, List\ Pair,\ \underline{(}] \\
[List \rightarrow \bullet\, Pair,\ \texttt{eof}] & [List \rightarrow \bullet\, Pair,\ \underline{(}] & [Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)},\ \texttt{eof}] \\
[Pair \rightarrow \bullet\, \underline{(}\ Pair\ \underline{)},\underline{(}] & [Pair \rightarrow \bullet\, \underline{(}\ \underline{)},\ \texttt{eof}] & [Pair \rightarrow \bullet\, \underline{(}\ \underline{)},\underline{(}]
\end{array}
\right\}
$$

CC Construction: CC_0 Step 4

(0) [Goal → • List, eof]
(1) [List → • List Pair, eof]
(2) [List → • Pair, eof]
(3) [Pair → • (Pair), eof]
(4) [Pair → • ( ), eof]
(5) [List → • List Pair, ( ]
(6) [List → • Pair, ( ]

1	$Goal \rightarrow List$
2	$List \rightarrow List\ Pair$
3	$\mid\ Pair$
4	$Pair \rightarrow \underline{(}\ Pair\ \underline{)}$
5	$\mid\ \underline{(}\ \underline{)}$

List ε Pair ε (

- **Hint 1**. How is $[A \rightarrow \beta \bullet C\ \delta, a]$ instantiated?
- **Hint 2**. What are $C \rightarrow \gamma \in R$?
- **Hint 3**. FIRST(δa) = FIRST(ε ( ) = { (}

How should s be extended?

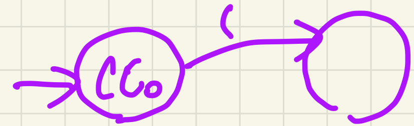
```
for [A → ··· • C δ, a] ∈ s:
  for C → γ ∈ R:
    for b ∈ FIRST(δa):
      s := s ∪ { [C → •γ, b] }
```

Two additional LR(1) items:

1. [Pair → • (Pair), (]
2. [Pair → • ( ), (]

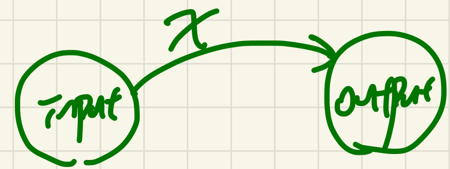
$$CC_0 = \left\{ \begin{array}{ccc} [Goal \rightarrow \bullet List, \texttt{eof}] & [List \rightarrow \bullet List\ Pair, \texttt{eof}] & [List \rightarrow \bullet List\ Pair, \underline{(}] \\ [List \rightarrow \bullet Pair, \texttt{eof}] & [List \rightarrow \bullet Pair, \underline{(}] & [Pair \rightarrow \bullet \underline{(}\ Pair\ \underline{)}, \texttt{eof}] \\ [Pair \rightarrow \bullet \underline{(}\ Pair\ \underline{)}, \underline{(}] & [Pair \rightarrow \bullet \underline{(}\ \underline{)}, \texttt{eof}] & [Pair \rightarrow \bullet \underline{(}\ \underline{)}, \underline{(}] \end{array} \right\}$$

CC Construction: goto



ALGORITHM: $\boxed{goto}$ source subset state
  INPUT: $\boxed{\text{a set } s \text{ of } \mathbf{LR(1)} \text{ items,}}$ a symbol x
  OUTPUT: $\boxed{\text{a set of } \mathbf{LR(1)} \text{ items}}$ target subset state
PROCEDURE:
  moved $:= \varnothing$
  for *item* $\in s$:
    if *item* $= [\alpha \to \beta \bullet x \delta,\ \ a]$ **then** expecting to read x
      moved $:=$ *moved* $\cup \{ [\alpha \to \beta x \bullet \delta,\ \ a] \}$
      x already recognized.
    end
  return *closure*(*moved*)

Analogy: ε-NFA to DFA



Subset construction (with ***lazy evaluation*** and *epsilon closures*) produces a ***DFA*** transition table.

source	symb. $d \in 0..9$	$s \in \{+,-\}$	.
$\{q_0, q_1\}$	$\{q_1, q_4\}$	$\{q_1\}$	$\{q_2\}$
$\{q_1, q_4\}$ tar.	$\{q_1, q_4\}$	$\varnothing$	$\{q_2, q_3, q_5\}$
$\{q_1\}$	$\{q_1, q_4\}$	$\varnothing$	$\{q_2\}$
$\{q_2\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$
$\{q_2, q_3, q_5\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$
$\{q_3, q_5\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$

For example, $\delta(\{q_0, q_1\}, d)$ is calculated as follows: $[d \in 0..9]$

$$\bigcup\{\mathrm{ECLOSE}(q) \mid q \in \delta(q_0, d) \cup \delta(q_1, d)\}$$

$$\mathrm{ECLOSE}\left(\begin{array}{c}\delta(q_0, d)\\ \cup\\ \delta(q_1, d)\end{array}\right)$$

CC Construction: goto

Calculate goto(cc_0, ( )
i.e., "next subset state" from cc_0 taking (

1	*Goal*	→	*List*
2	*List*	→	*List Pair*
3		|	*Pair*
4	*Pair*	→	(*Pair*)
5		|	( )

$$CC_0 = \left\{ \begin{array}{lll} [Goal \to \bullet\, List, \texttt{eof}] & [List \to \bullet\, List\ Pair, \texttt{eof}] & [List \to \bullet\, List\ Pair, (] \\ [List \to \bullet\, Pair, \texttt{eof}] & [List \to \bullet\, Pair, (] & [Pair \to \bullet\, (\ Pair\ ), \texttt{eof}] \\ [Pair \to \bullet\, (\ Pair\ ), (] & [Pair \to \bullet\, (\ ), \texttt{eof}] & [Pair \to \bullet\, (\ ), (] \end{array} \right\}$$

[Pair → • (Pair), ( ]
[Pair → • ( ), eof]
[Pair → • (Pair), eof]
[Pair → • ( ), ( ]

will trigger additional items

[Pair → (• Pair), ( ]
[Pair → (•), eof]
[Pair → (• Pair), eof]
[Pair → (•), ( ]

closure

```
ALGORITHM: goto
  INPUT: a set S of LR(1) items, a symbol x
  OUTPUT: a set of LR(1) items
PROCEDURE:
  moved := ∅
  for item ∈ S:
    if item = [α → β • x δ, a] then
      moved := moved ∪ { [α → β x • δ, a] }
    end
  return closure(moved)
```

must be a terminal

$$CC_3 = \left\{ \begin{array}{lll} [Pair \to \bullet\, (\ Pair\ ),)] & [Pair \to (\ \bullet\, Pair\ ), \texttt{eof}] & [Pair \to (\ \bullet\, Pair\ ), (] \\ [Pair \to \bullet\, (\ ),)] & [Pair \to (\ \bullet\ ), \texttt{eof}] & [Pair \to (\ \bullet\ ), (] \end{array} \right\}$$

Exercise: why the highlighted items trigger the two additional items