Introducing

PLASSE

Laboratory

& its Current Research

Kyung-Goo Doh

Programming
Languages
Application to
Software
Security &
Engineering

Current Research Members

- professor : 1
- research professor : 1
- Ph.D. students : 4 (-3) (+1)
- M.S. students : 7 (-2) (+?)
- on-campus collaborators
- domestic collaborators
- international collaborators

Research Directions

- Theoretical research
  - ✓ develop theory and methodology
- Industrial applications
  - ✓ implement analysis engines and tools
  - ✓ transfer technology to software industry

Research Theme

- Semantics engineering
  - ✓ simple, user-friendly semantics meta-language
- String analysis
  - ✓ syntax and semantics analysis for dynamically generated strings
- Software security
  - ✓ static/dynamic detection of security vulnerabilities from source code
- Software maintenance
  - ✓ extraction of software properties and metrics from source code

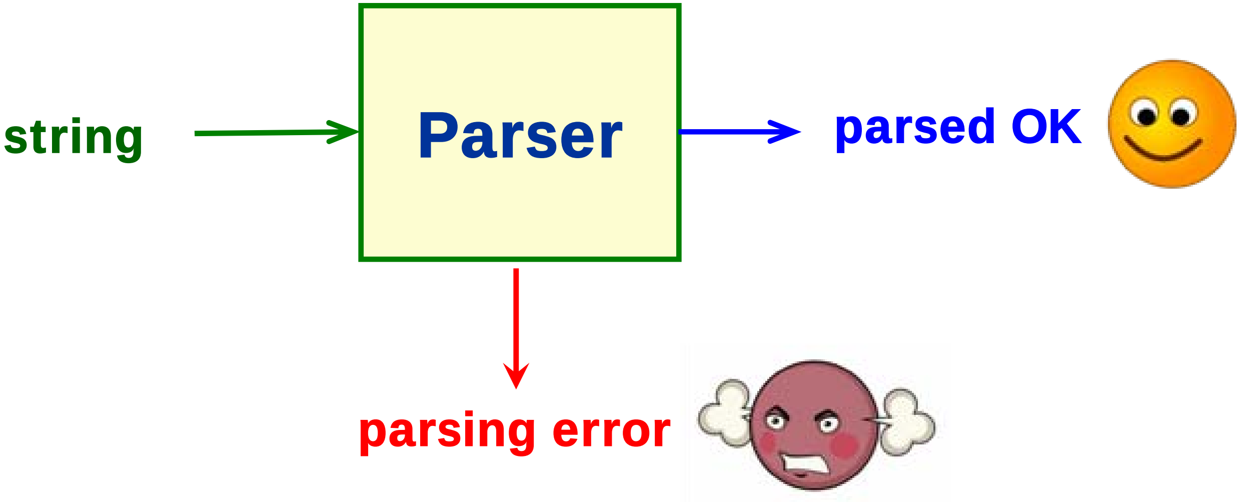
Abstract Parsing

Joint work with **Hyunha Kim** & **David A. Schmidt**

Kyung-Goo Doh

“Classic” String Analysis

- on statically generated string
- (scanning +) parsing



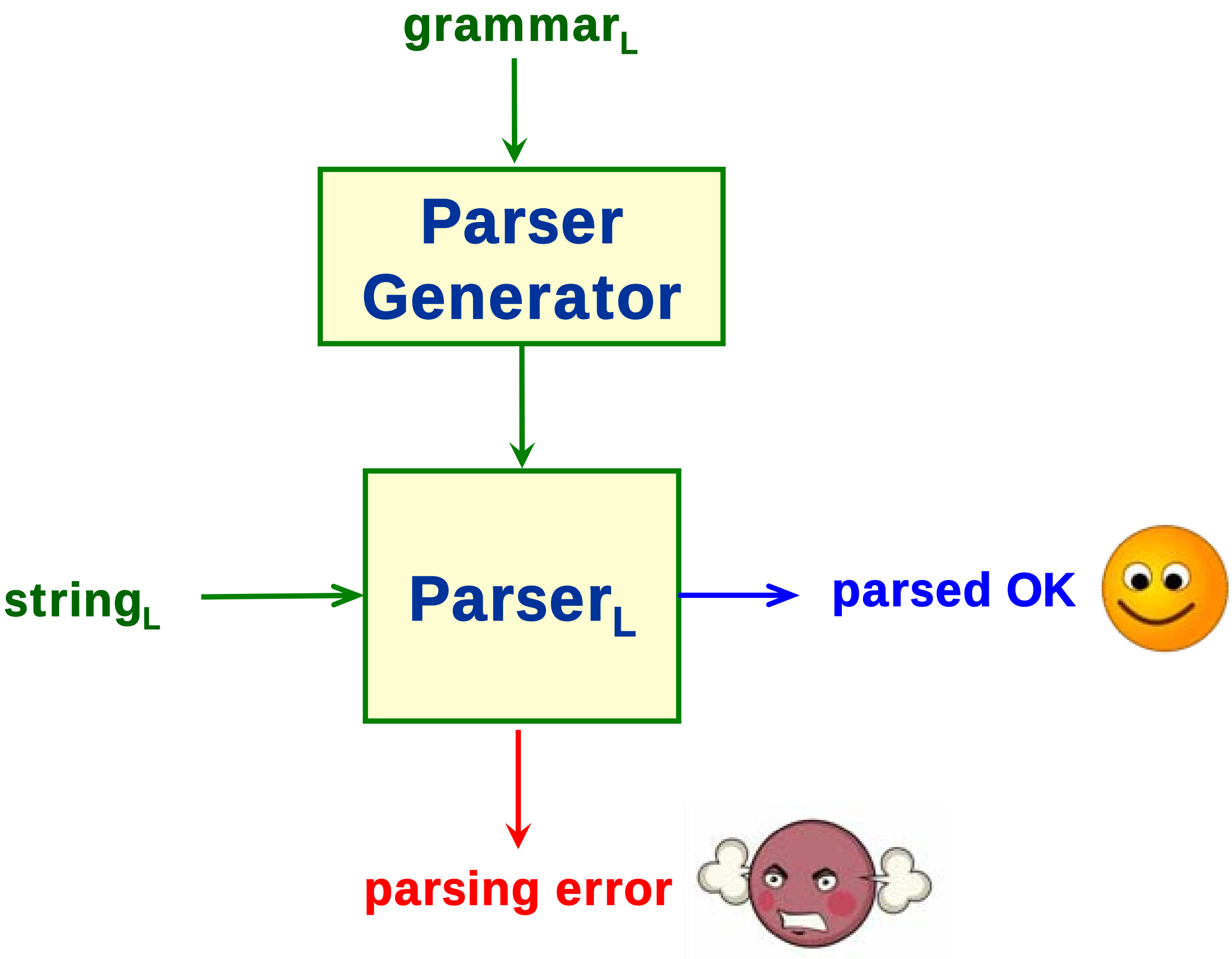
string

Parser

parsed OK

parsing error

Parsing



grammar$_L$

Parser Generator

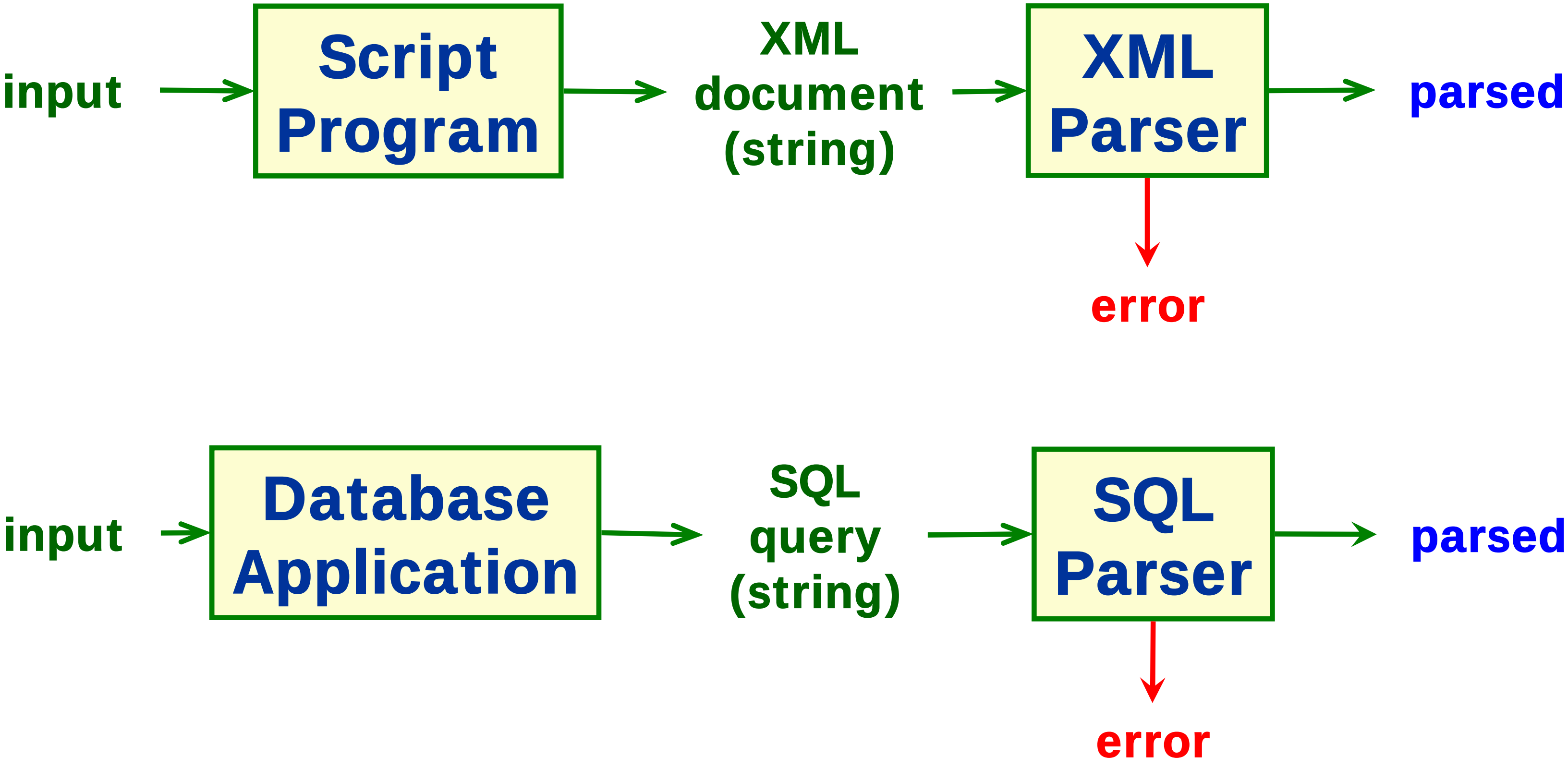
string$_L$

Parser$_L$

parsed OK

parsing error

Dynamically Generated String



Example: Database Application

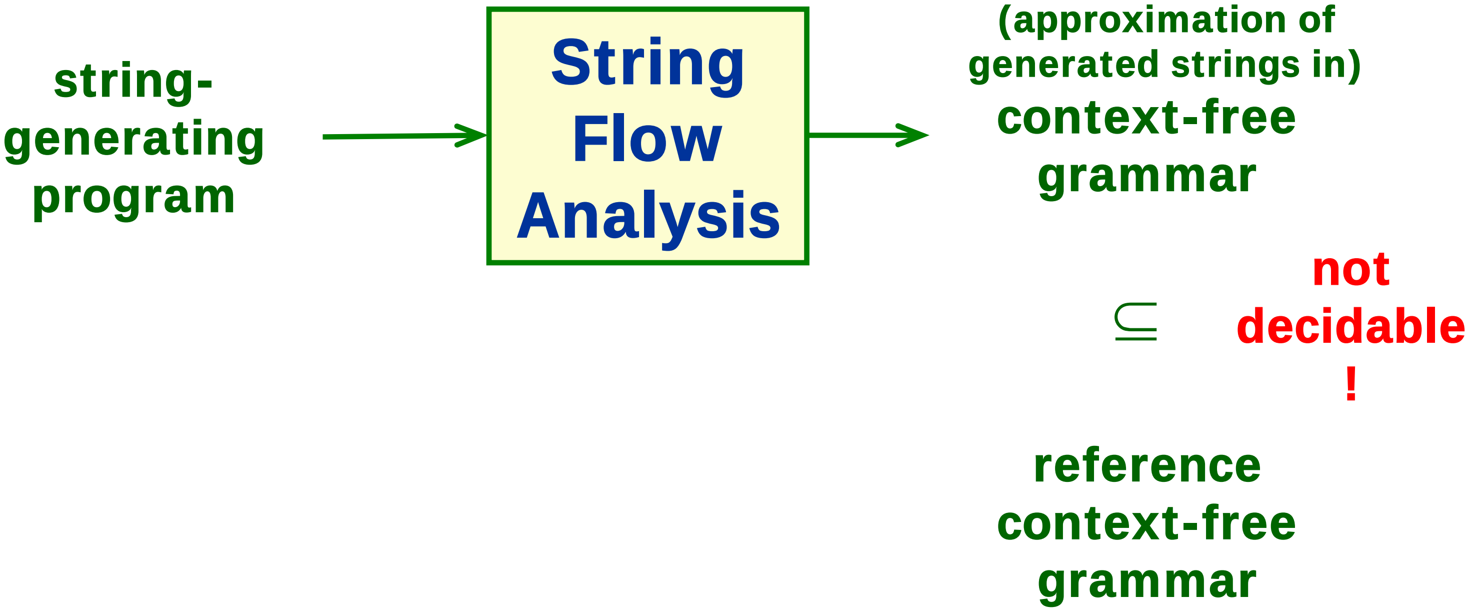
```
public void printAddresses(String id) throws SQLException {
  Connection con = DriverManager.getConnection(“students.db”);
  String q = “SELECT * FROM address”;
  if (id != 0) q = q + “WHERE studentid=“ + id;
  ResultSet rs = con.createStatement().executeQuery(q);
  while (rs.next()) { System.out.println.getString(“addr”)); }
```

taken from Christensen/Moeller/Schwartzbach’s SAS2003 paper
“Precise analysis of string expression” with minor modification

- This Java program parses and compiles OK.
- You can check if the dynamically generated SQL query parses OK at run-time.
- Question: Can we statically check if all the SQL queries generated by this program parse OK?

Previous Approach

- Analysis-then-parse



string-generating program

String Flow Analysis

(approximation of generated strings in) context-free grammar

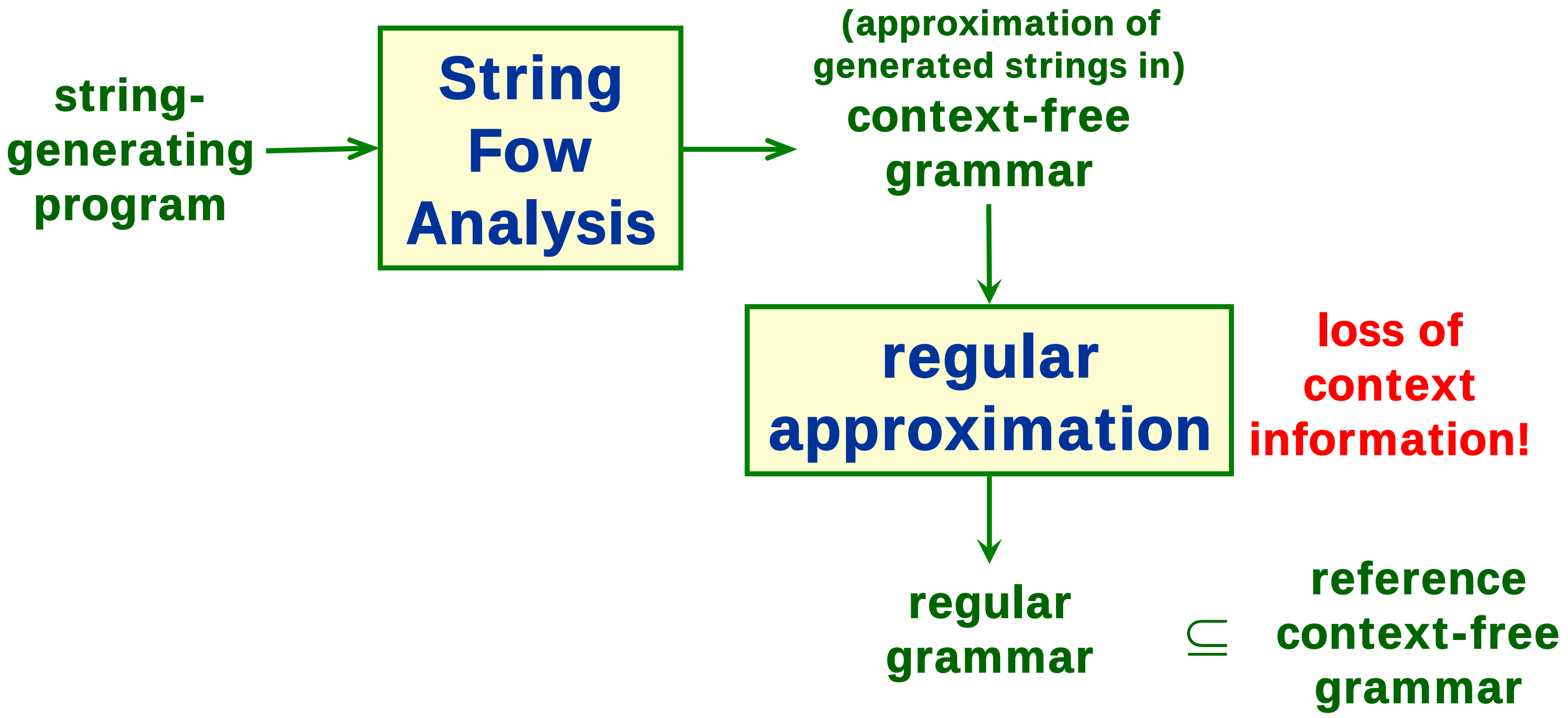
$\subseteq$

not decidable !

reference context-free grammar

Previous Approach

- Analysis-then-regular_approximation-then-parse



string-generating program

String Fow Analysis

(approximation of generated strings in) context-free grammar

regular approximation

loss of context information!

regular grammar ⊆ reference context-free grammar

Example

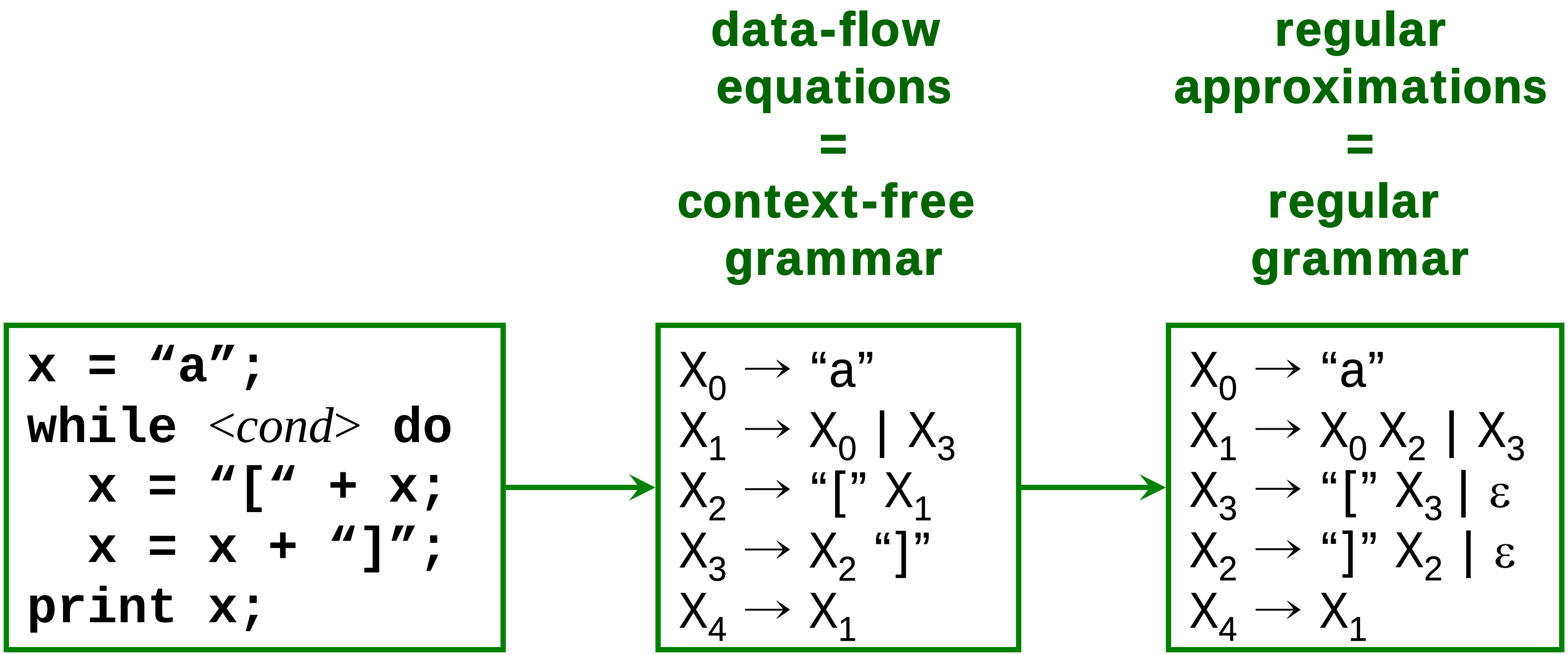
```
x = “a”;
while <cond> do
  x = “[” + x;
  x = x + “]”;
print x;
```

hot spot

- Is the string generated at hot spot above conformed to the following reference grammar?

S → “a” | “[” S “]”

Example: Previous String Analyzer



data-flow equations = context-free grammar

regular approximations = regular grammar

```
x = “a”;
while <cond> do
   x = “[“ + x;
   x = x + “]”;
print x;
```

$$
\begin{aligned}
X_0 &\rightarrow \text{“a”} \\
X_1 &\rightarrow X_0 \mid X_3 \\
X_2 &\rightarrow \text{“[”}\ X_1 \\
X_3 &\rightarrow X_2\ \text{“]”} \\
X_4 &\rightarrow X_1
\end{aligned}
$$

$$
\begin{aligned}
X_0 &\rightarrow \text{“a”} \\
X_1 &\rightarrow X_0\ X_2 \mid X_3 \\
X_3 &\rightarrow \text{“[”}\ X_3 \mid \varepsilon \\
X_2 &\rightarrow \text{“]”}\ X_2 \mid \varepsilon \\
X_4 &\rightarrow X_1
\end{aligned}
$$

[* a]*

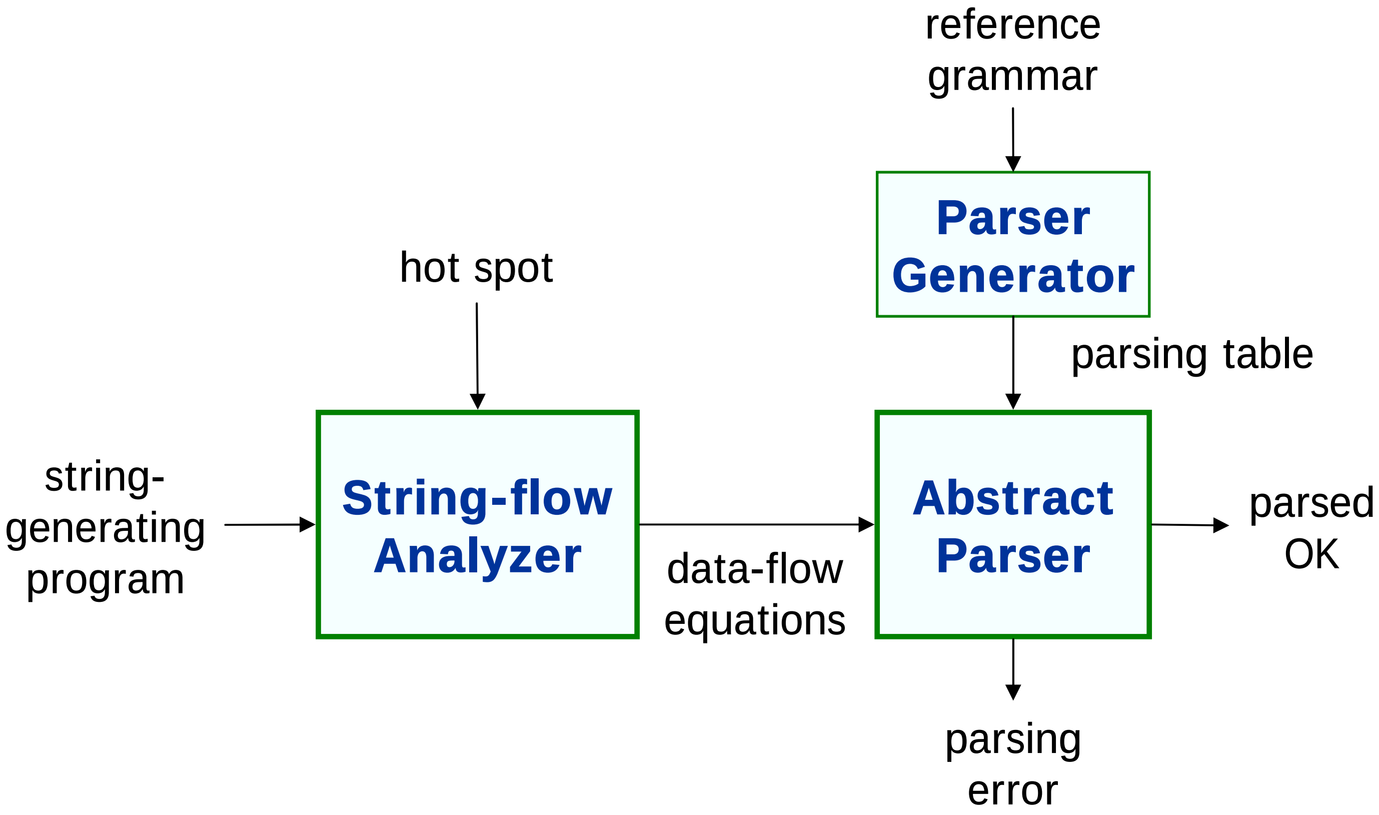
$\supseteq$

$S \rightarrow \text{“a”} \mid \text{“[”}\ S\ \text{“]”}$

Abstract Parsing

- Simultaneous analysis-and-parsing
  - ✓ statically analyze a program that dynamically generates strings, and, at the same time, parse the generated strings with the LR(k) reference grammar
- Abstract parse stacks as abstract string values
  - ✓ encode a generated string’s context-free structure

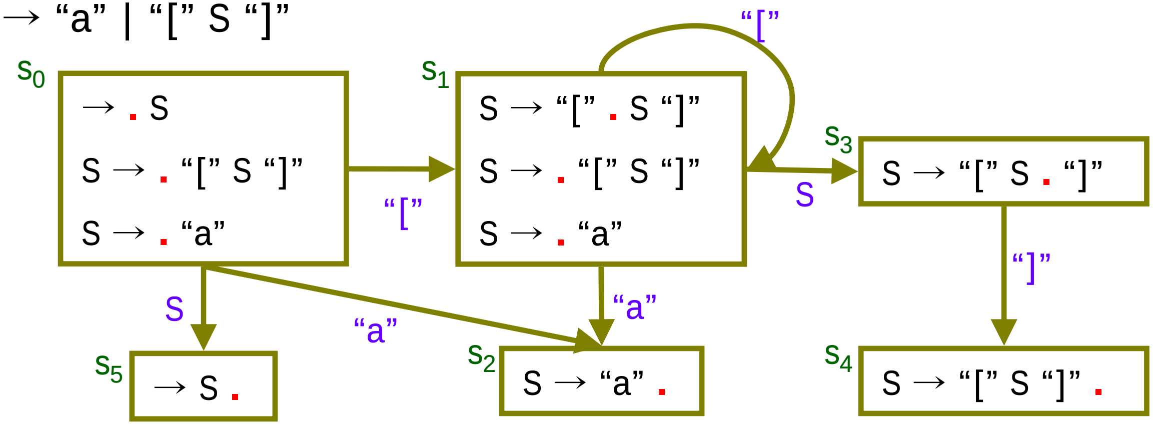
Architecture of Abstract Parsing



Our abstract parser solves data-flow equations in the domain of abstract stack.

LR(0) Parsing

Goto Controller for parser built from LR(0)-items
for the reference grammar, S → “a” | “[” S “]”



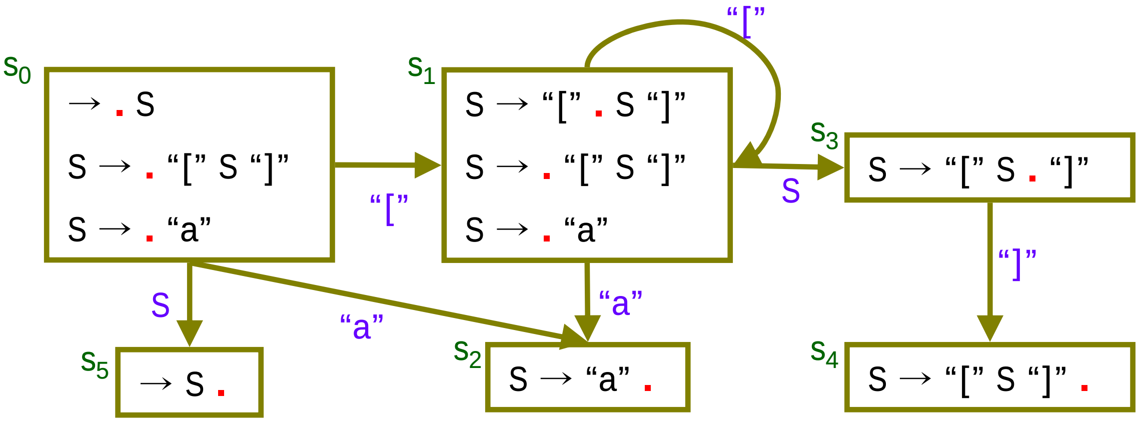
Parse of input sequence,
[[a]]

parse stack (top at right)	**input sequence** (front at left)	
s_0	[[a]]	shift
$s_0\ s_1$	[a]]	shift
$s_0\ s_1\ s_1$	a]]	shift
$s_0\ s_1\ s_1\ s_2$	]]	reduce: S → “a”
$s_0\ s_1\ s_1$	S]]	shift
$s_0\ s_1\ s_1\ s_3$	]]	shift
$s_0\ s_1\ s_1\ s_3\ s_4$	]	reduce: S → “[” S “]”
$s_0\ s_1$	S]	shift
$s_0\ s_1\ s_3$	]	shift
$s_0\ s_1\ s_3\ s_4$		reduce: S → “[” S “]”
s_0	S	shift
$s_0\ s_5$		(done)

Abstract Parsing: Example 1

```
if <cond>
then x = “a“;
else x = “[” + “a”
print x;
```

$$\begin{aligned} X_0 &= \text{“a”} \\ X_1 &= \text{“[”} \cdot \text{“a”} \\ X_2 &= X_0 \sqcup X_1 \end{aligned}$$



$X_2(s_0) = ?$

$$X_2(s_0) = X_0(s_0) \sqcup X_1(s_0) =$$

$$\begin{aligned} X_0(s_0) &= \text{goto}(s_0, \text{“a”}) \\ &= s_2 \quad (\text{reduce: } S \rightarrow \text{“a”}) \\ &= \text{goto}(s_0, S) \\ &= s_5 \end{aligned}$$

$$\begin{aligned} X_1(s_0) &= (\text{“[”} \cdot \text{“a”})(s_0) \\ &= \text{goto}(s_0, \text{“[”}) * \text{“a”} \\ &= s_1 * \text{“a”} \\ &= s_1 \cdot \text{goto}(s_1, \text{“a”}) \\ &= s_1 \cdot s_2 \quad (\text{reduce: } S \rightarrow \text{“a”}) \\ &= s_1 \cdot \text{goto}(s_1, S) \\ &= s_1 \cdot s_3 \end{aligned}$$

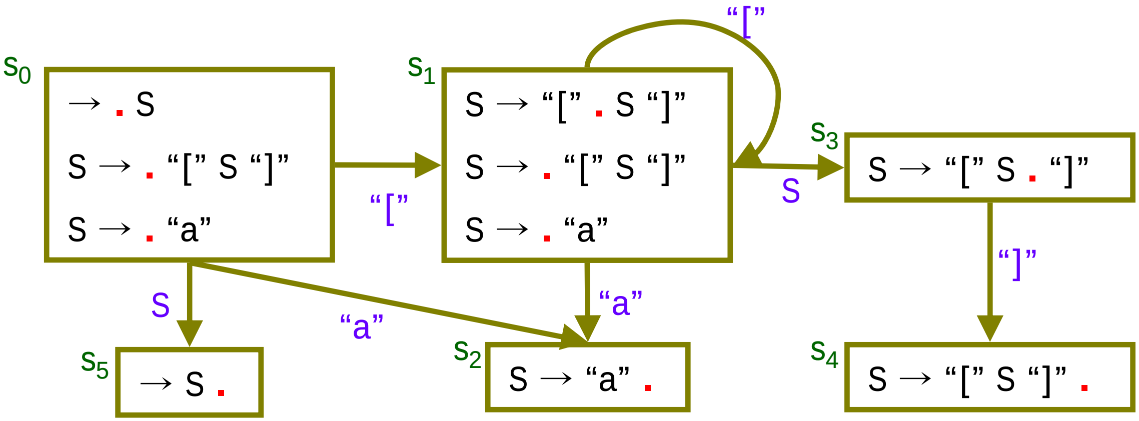
Abstract Parsing: Example 1

```
if <cond>
then x = “a“;
else x = “[” + “a”
print x;
```

X_0 = “a”

X_1 = “0” · “a”

$X_2 = X_0 \sqcup X_1$



$X_2(s_0)$ = ?

$X_2(s_0) = X_0(s_0) \sqcup X_1(s_0) = \{s_5, s_1 \cdot s_3\}$

$$\begin{aligned} X_0(s_0) &= \text{goto}(s_0, \text{“a”}) \\ &= s_2 \quad (\text{reduce: } S \rightarrow \text{“a”}) \\ &= \text{goto}(s_0, S) \\ &= s_5 \end{aligned}$$

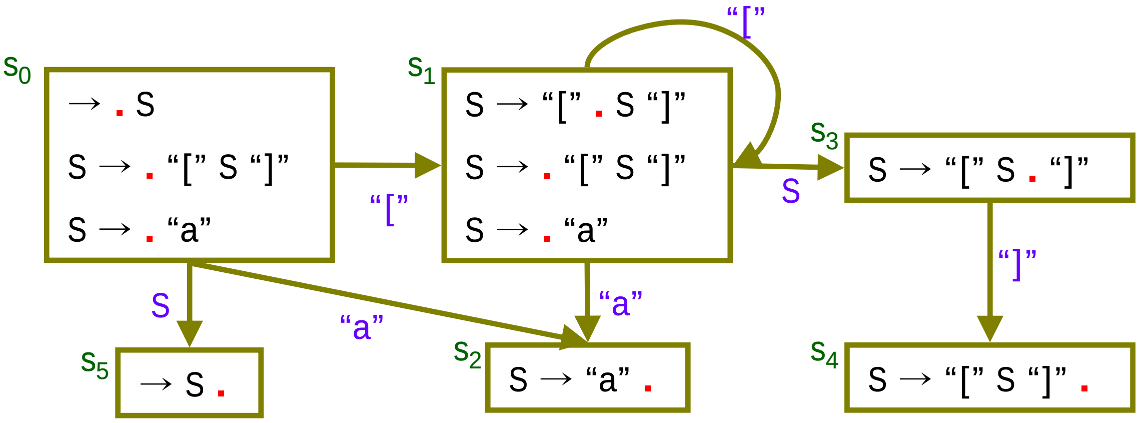
$$\begin{aligned} X_1(s_0) &= (\text{“[”} \cdot \text{“a”})(s_0) \\ &= \text{goto}(s_0, \text{“[”}) * \text{“a”} \\ &= s_1 * \text{“a”} \\ &= s_1 \cdot \text{goto}(s_1, \text{“a”}) \\ &= s_1 \cdot s_2 \quad (\text{reduce: } S \rightarrow \text{“a”}) \\ &= s_1 \cdot \text{goto}(s_1, S) \\ &= s_1 \cdot s_3 \end{aligned}$$

Abstract Parsing: Example 2

```
x = “”;
while <cond> do
  x = x + “[”
x = x + “a”;
print x;
```

$X_0 = \in$
$X_1 = X_0 \sqcup X_2$
$X_2 = X_1 \cdot “[”$
$X_3 = X_1 \cdot “a”$

$X_3(s_0) = ?$



$$\begin{aligned} X_3(s_0) &= (X_1 \cdot “a”)(s_0) \\ &= X_1(s_0) * “a” \\ &= \end{aligned}$$

$$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) =$$

$$X_0(s_0) = s_0$$

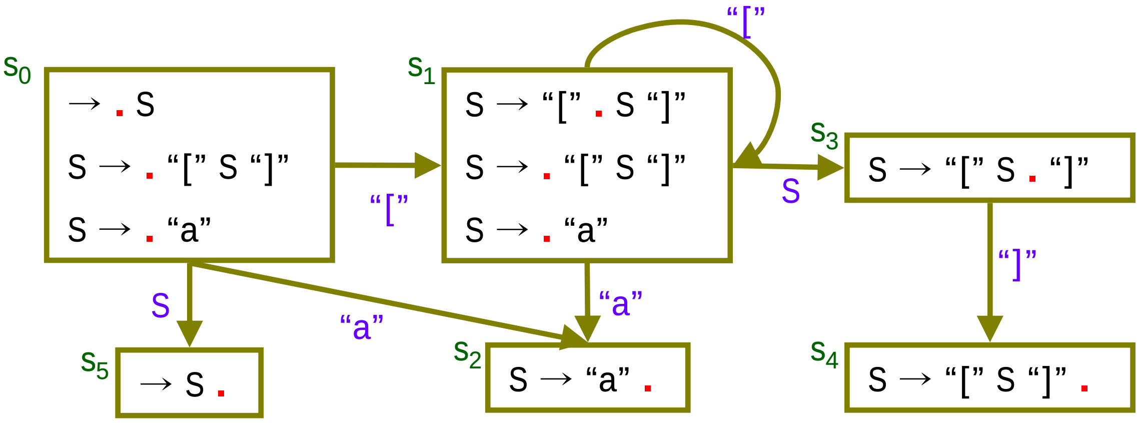
$$\begin{aligned} X_2(s_0) &= (X_1 \cdot “[”)(s_0) \\ &= X_1(s_0) * “[” \\ &= \perp \end{aligned}$$

Abstract Parsing: Example 2

```
x = "";
while <cond> do
  x = x + "["
x = x + "a";
print x;
```

$$X_0 = \in$$
$$X_1 = X_0 \sqcup X_2$$
$$X_2 = X_1 \cdot \text{“[”}$$
$$X_3 = X_1 \cdot \text{“a”}$$

$X_3(s_0) = ?$



$$\begin{aligned} X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\ &= X_1(s_0) * \text{“a”} \\ &= \end{aligned}$$

$$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) = \{s_0\}$$

$$X_0(s_0) = s_0$$

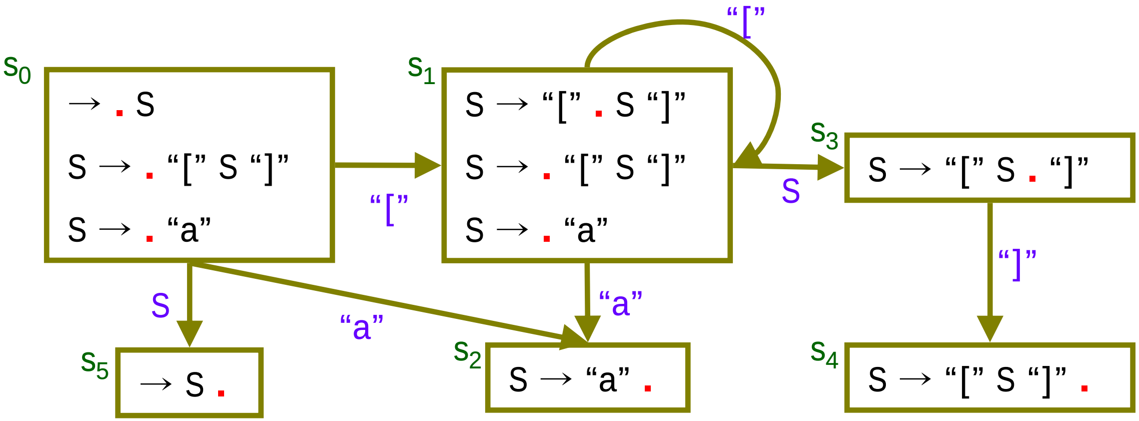
$$\begin{aligned} X_2(s_0) &= (X_1 \cdot \text{“[”})(s_0) \\ &= X_1(s_0) * \text{“[”} \\ &= \perp \end{aligned}$$

Abstract Parsing: Example 2

```
x = “”;
while <cond> do
  x = x + “[”
x = x + “a”;
print x;
```

$$X_0 = \in$$
$$X_1 = X_0 \sqcup X_2$$
$$X_2 = X_1 \cdot \text{“[”}$$
$$X_3 = X_1 \cdot \text{“a”}$$

$X_3(s_0) = ?$



s_0

→ . S

S → . “[” S “]”

S → . “a”

s_1

S → “[” . S “]”

S → . “[” S “]”

S → . “a”

s_2

S → “a” .

s_3

S → “[” S . “]”

s_4

S → “[” S “]” .

s_5

→ S .

$$\begin{aligned} X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\ &= X_1(s_0) * \text{“a”} \\ &= \end{aligned}$$

$$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) = \{s_0\}$$

$$X_0(s_0) = s_0$$

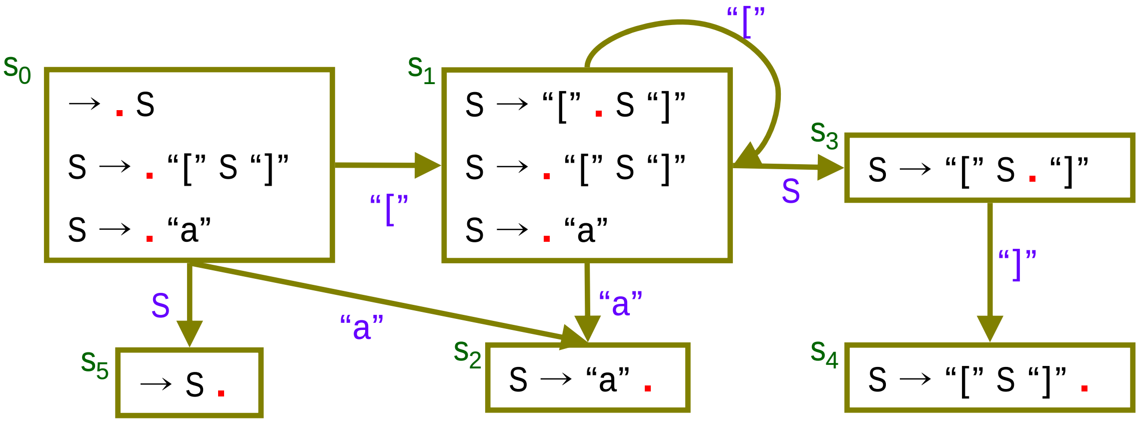
$$\begin{aligned} X_2(s_0) &= (X_1 \cdot \text{“[”})(s_0) \\ &= X_1(s_0) * \text{“[”} \\ &= s_0 * \text{“[”} \\ &= s_0 \cdot \mathrm{goto}(s_0, \text{“[”}) \\ &= s_0 \cdot s_1 \end{aligned}$$

Abstract Parsing: Example 2

```
x = “”;
while <cond> do
  x = x + “[”
x = x + “a”;
print x;
```

$$
\begin{aligned}
X_0 &= \in \\
X_1 &= X_0 \sqcup X_2 \\
X_2 &= X_1 \cdot \text{“[”} \\
X_3 &= X_1 \cdot \text{“a”}
\end{aligned}
$$

$X_3(s_0) = ?$



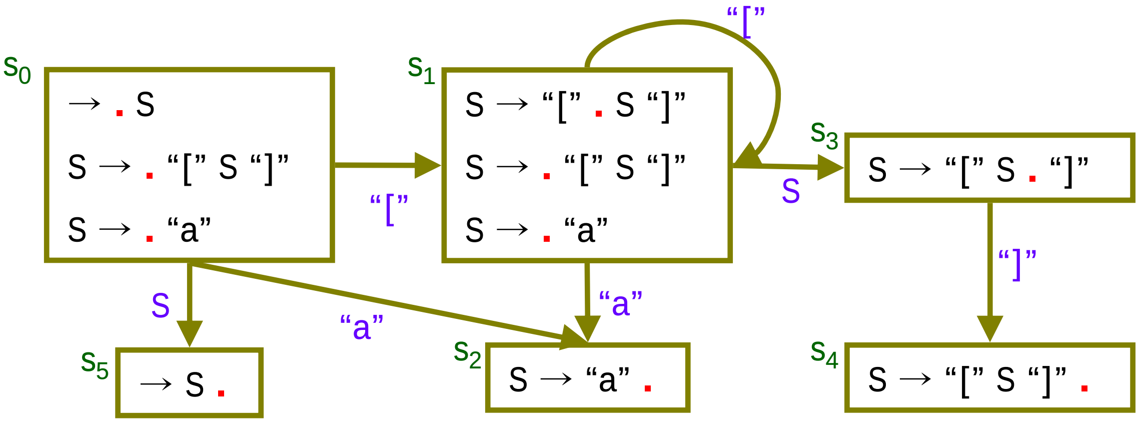
$$
\begin{aligned}
X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\
&= X_1(s_0) * \text{“a”} \\
&=
\end{aligned}
$$

$$
\begin{aligned}
X_1(s_0) &= X_0(s_0) \sqcup X_2(s_0) = \{s_0\} \cup \{s_0 \cdot s_1\} \\
& = \{s_0, s_0 \cdot s_1\} \\
X_0(s_0) &= s_0 \\
X_2(s_0) &= (X_1 \cdot \text{“[”})(s_0) \\
&= X_1(s_0) * \text{“[”} \\
&= s_0 * \text{“[”} \\
&= s_0 \cdot \text{goto}(s_0, \text{“[”}) \\
&= s_0 \cdot s_1
\end{aligned}
$$

Abstract Parsing: Example 2

```
x = "";
while <cond> do
  x = x + "["
x = x + "a";
print x;
```

$$
\begin{aligned}
X_0 &= \in \\
X_1 &= X_0 \sqcup X_2 \\
X_2 &= X_1 \cdot \text{“[”} \\
X_3 &= X_1 \cdot \text{“a”}
\end{aligned}
$$



$X_3(s_0) = ?$

$$
\begin{aligned}
X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\
&= X_1(s_0) * \text{“a”} \\
&=
\end{aligned}
$$

$$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) = \{s_0, s_0 \cdot s_1\}$$

$$
\begin{aligned}
X_0(s_0) &= s_0 \\
X_2(s_0) &= (X_1 \cdot \text{“[”})(s_0) \\
&= X_1(s_0) * \text{“[”} \\
&= s_0 \cdot s_1 * \text{“[”} \\
&= s_0 \cdot s_1 \cdot \text{goto}(s_1, \text{“[”}) \\
&= s_0 \cdot s_1 \cdot s_1
\end{aligned}
$$

Abstract Parsing: Example 2

```
x = “”;
while <cond> do
  x = x + “[”
x = x + “a”;
print x;
```

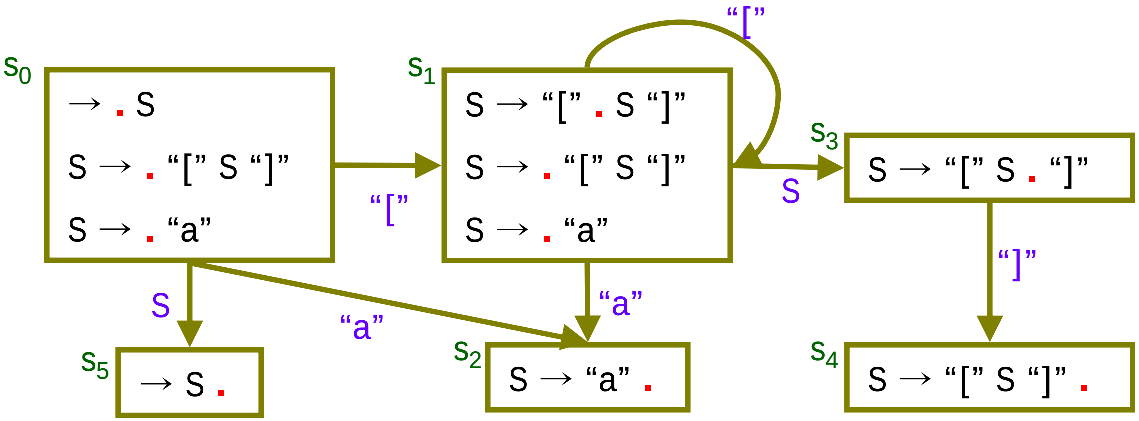
$X_0 = \in$

$X_1 = X_0 \bigsqcup X_2$

$X_2 = X_1 \cdot \text{“[”}$

$X_3 = X_1 \cdot \text{“a”}$

$X_3(s_0) = ?$



$$\begin{aligned} X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\ &= X_1(s_0) * \text{“a”} \\ &= \end{aligned}$$

$$X_1(s_0) = X_0(s_0) \bigsqcup X_2(s_0) = \{s_0, s_0 \cdot s_1, s_0 \cdot s_1 \cdot s_1\}$$

$$X_0(s_0) = s_0$$

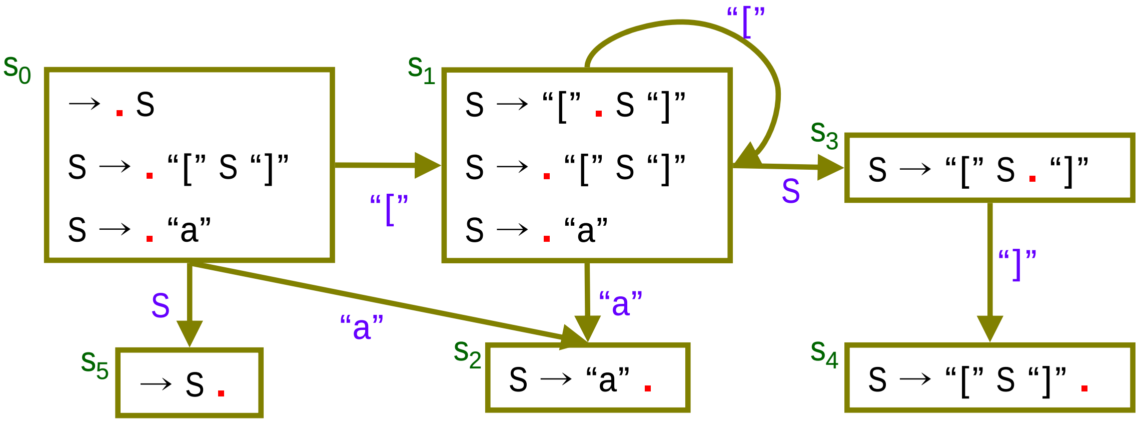
$$\begin{aligned} X_2(s_0) &= (X_1 \cdot \text{“[”})(s_0) \\ &= X_1(s_0) * \text{“[”} \\ &= s_0 \cdot s_1 * \text{“[”} \\ &= s_0 \cdot s_1 \cdot \text{goto}(s_1, \text{“[”}) \\ &= s_0 \cdot s_1 \cdot s_1 \end{aligned}$$

Abstract Parsing: Example 2

```
x = “”;
while <cond> do
  x = x + “[”
x = x + “a”;
print x;
```

$$\begin{aligned} X_0 &= \in \\ X_1 &= X_0 \sqcup X_2 \\ X_2 &= X_1 \cdot \text{“[”} \\ X_3 &= X_1 \cdot \text{“a”} \end{aligned}$$

$X_3(s_0) = ?$



$$\begin{aligned} X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\ &= X_1(s_0) * \text{“a”} \\ &= \end{aligned}$$

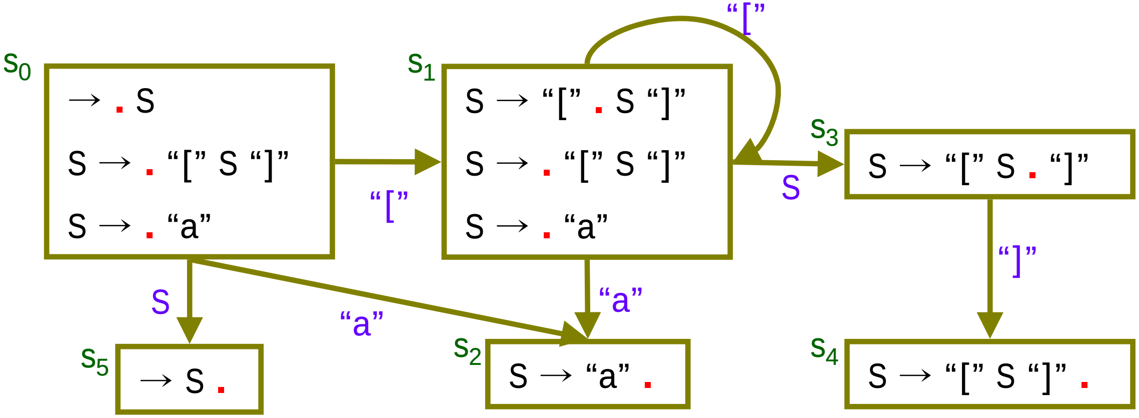
$$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) = \{s_0,\ s_0 \cdot s_1,\ s_0 \cdot s_1 \cdot s_1,\ \ldots\}$$

$$\begin{aligned} X_0(s_0) &= s_0 \\ X_2(s_0) &= (X_1 \cdot \text{“[”})(s_0) \\ &= X_1(s_0) * \text{“[”} \\ &= s_0 \cdot s_1 * \text{“[”} \\ &= s_0 \cdot s_1 \cdot \text{goto}(s_1, \text{“[”}) \\ &= s_0 \cdot s_1 \cdot s_1 \end{aligned}$$

Abstract Parsing: Example 2

```
x = "";
while <cond> do
  x = x + "["
x = x + "a";
print x;
```

$$X_0 = \epsilon$$
$$X_1 = X_0 \sqcup X_2$$
$$X_2 = X_1 \cdot \text{“[”}$$
$$X_3 = X_1 \cdot \text{“a”}$$



s_0: $\rightarrow$. S; S $\rightarrow$. “[” S “]”; S $\rightarrow$. “a”

s_1: S $\rightarrow$ “[” . S “]”; S $\rightarrow$. “[” S “]”; S $\rightarrow$. “a”

s_2: S $\rightarrow$ “a” .

s_3: S $\rightarrow$ “[” S . “]”

s_4: S $\rightarrow$ “[” S “]” .

s_5: $\rightarrow$ S .

$X_3(s_0) = ?$

$$\begin{aligned} X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\ &= X_1(s_0) * \text{“a”} \\ &= \end{aligned}$$

$$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) = \{s_0,\ s_0 \cdot s_1{}^{+}\}$$

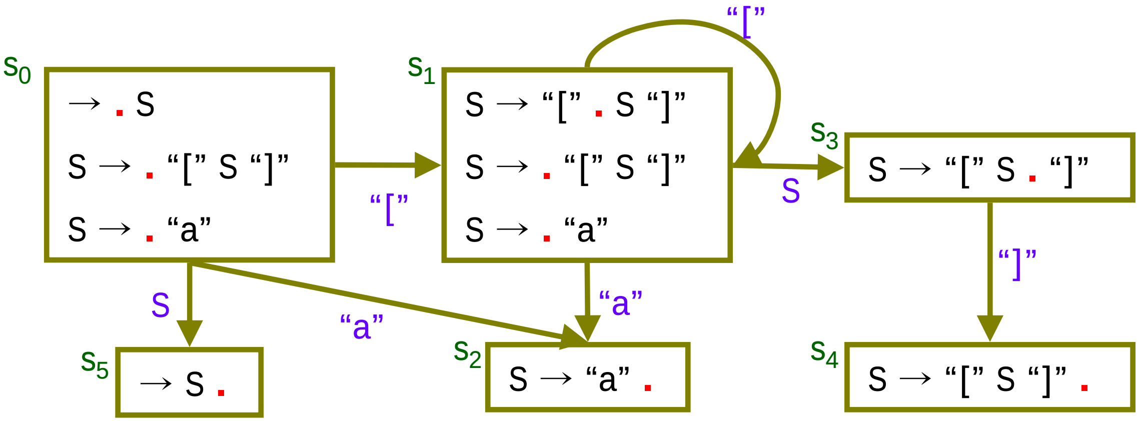
$$\begin{aligned} X_0(s_0) &= s_0 \\ X_2(s_0) &= (X_1 \cdot \text{“[”})(s_0) \\ &= X_1(s_0) * \text{“[”} \\ &= s_0 \cdot s_1 * \text{“[”} \\ &= s_0 \cdot s_1 \cdot \text{goto}(s_1, \text{“[”}) \\ &= s_0 \cdot s_1 \cdot s_1 \end{aligned}$$

Abstract Parsing: Example 2

```
x = "";
while <cond> do
  x = x + "["
x = x + "a";
print x;
```

$$X_0 = \in$$
$$X_1 = X_0 \sqcup X_2$$
$$X_2 = X_1 \cdot \text{“[”}$$
$$X_3 = X_1 \cdot \text{“a”}$$

$X_3(s_0) = ?$



$$\begin{aligned} X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\ &= X_1(s_0) * \text{“a”} \\ &= \{s_0,\ s_0 \cdot s_1{}^{+}\} * \text{“a”} \end{aligned}$$

$$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) = \{s_0,\ s_0 \cdot s_1{}^{+}\}$$

$$X_0(s_0) = s_0$$

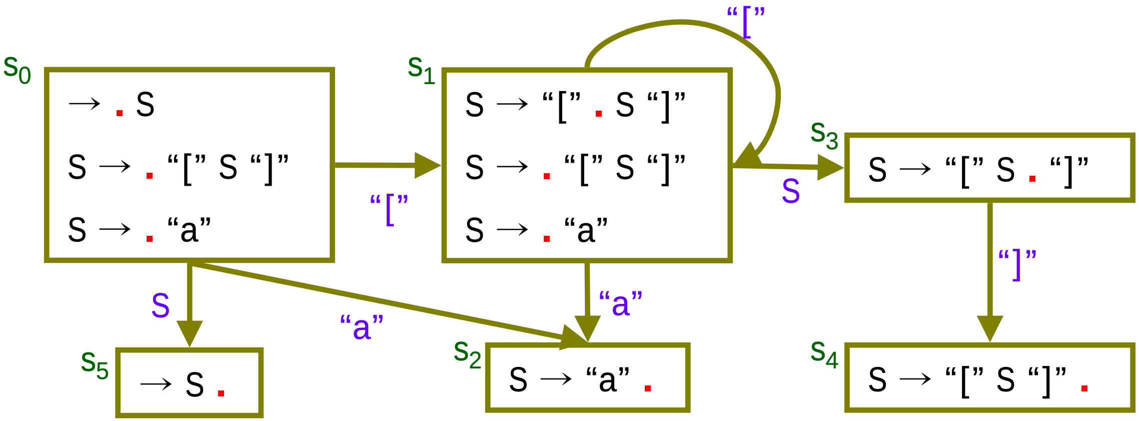
$$\begin{aligned} X_2(s_0) &= (X_1 \cdot \text{“[”})(s_0) \\ &= X_1(s_0) * \text{“[”} \\ &= s_0 \cdot s_1 * \text{“[”} \\ &= s_0 \cdot s_1 \cdot \text{goto}(s_1, \text{“[”}) \\ &= s_0 \cdot s_1 \cdot s_1 \end{aligned}$$

Abstract Parsing: Example 2

```
x = "";
while <cond> do
  x = x + "["
x = x + "a";
print x;
```

$$\begin{aligned} X_0 &= \in \\ X_1 &= X_0 \sqcup X_2 \\ X_2 &= X_1 \cdot \text{“[”} \\ X_3 &= X_1 \cdot \text{“a”} \end{aligned}$$

$X_3(s_0) = ?$



s_0: → . S; S → . "[" S "]"; S → . "a"

s_1: S → "[" . S "]"; S → . "[" S "]"; S → . "a"

s_2: S → "a" .

s_3: S → "[" S . "]"

s_4: S → "[" S "]" .

s_5: → S .

$$\begin{aligned} X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\ &= X_1(s_0) * \text{“a”} \\ &= \{s_0, s_0 \cdot s_1{}^+\} * \text{“a”} \\ &= \{s_0 * \text{“a”}, s_0 \cdot s_1{}^+ * \text{“a”}\} = \end{aligned}$$

$$\begin{aligned} & s_0 * \text{“a”} \\ &= s_0 \cdot \text{goto}(s_0, \text{“a”}) \\ &= s_0 \cdot s_2 \qquad \text{[reduce: S} \rightarrow \text{“a”]} \\ &= s_0 \cdot \text{goto}(s_0, \text{S}) \\ &= s_0 \cdot s_5 \end{aligned}$$

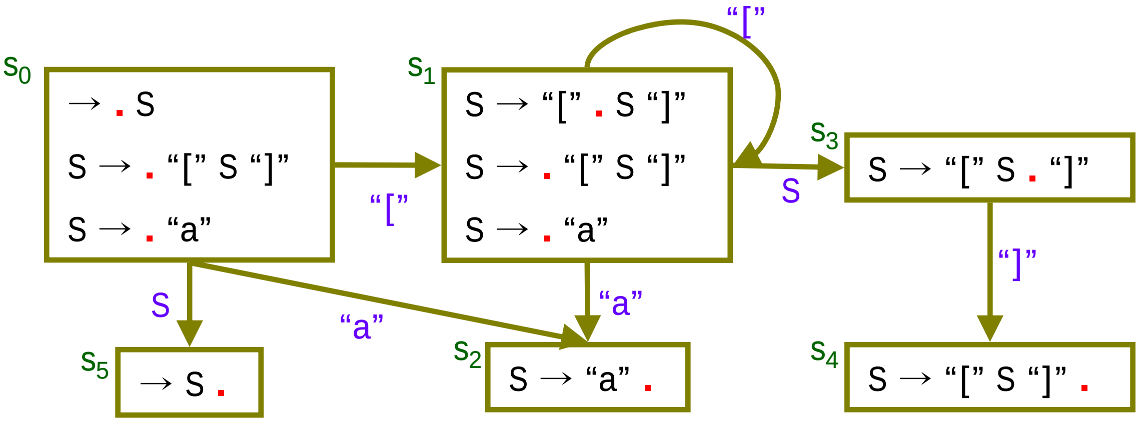
$$\begin{aligned} & s_0 \cdot s_1{}^+ * \text{“a”} \\ &= s_0 \cdot s_1{}^+ \cdot \text{goto}(s_1, \text{“a”}) \\ &= s_0 \cdot s_1{}^+ \cdot s_2 \qquad \text{[reduce: S} \rightarrow \text{“a”]} \\ &= s_0 \cdot s_1{}^+ \cdot \text{goto}(s_1, \text{S}) = s_0 \cdot s_1{}^+ \cdot s_3 \end{aligned}$$

Abstract Parsing: Example 2

```
x = "";
while <cond> do
  x = x + "["
x = x + "a";
print x;
```

$$X_0 = \in$$
$$X_1 = X_0 \sqcup X_2$$
$$X_2 = X_1 \cdot \text{“[”}$$
$$X_3 = X_1 \cdot \text{“a”}$$

$X_3(s_0) = ?$



s_0: → . S; S → . “[” S “]”; S → . “a”

s_1: S → “[” . S “]”; S → . “[” S “]”; S → . “a”

s_2: S → “a” .

s_3: S → “[” S . “]”

s_4: S → “[” S “]” .

s_5: → S .

$$\begin{aligned} X_3(s_0) &= (X_1 \cdot \text{“a”})(s_0) \\ &= X_1(s_0) * \text{“a”} \\ &= \{s_0,\ s_0 \cdot s_1^{+}\} * \text{“a”} \\ &= \{s_0 * \text{“a”},\ s_0 \cdot s_1^{+} * \text{“a”}\} = \{s_0 \cdot s_5,\ s_0 \cdot s_1^{+} \cdot s_3\} \end{aligned}$$

$$\begin{aligned} & s_0 * \text{“a”} \\ =\ & s_0 \cdot \text{goto}(s_0, \text{“a”}) \\ =\ & s_0 \cdot s_2 \qquad [\text{reduce: } S \to \text{“a”}] \\ =\ & s_0 \cdot \text{goto}(s_0, S) \\ =\ & s_0 \cdot s_5 \end{aligned}$$

$$\begin{aligned} & s_0 \cdot s_1^{+} * \text{“a”} \\ =\ & s_0 \cdot s_1^{+} \cdot \text{goto}(s_1, \text{“a”}) \\ =\ & s_0 \cdot s_1^{+} \cdot s_2 \qquad [\text{reduce: } S \to \text{“a”}] \\ =\ & s_0 \cdot s_1^{+} \cdot \text{goto}(s_1, S) = s_0 \cdot s_1^{+} \cdot s_3 \end{aligned}$$

Abstract Parsing: Example 3

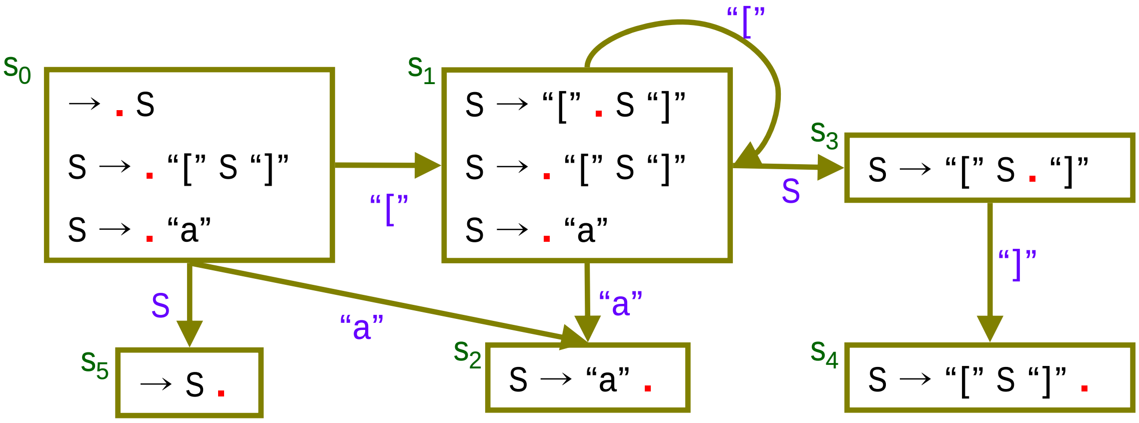
```
x = "a";
while <cond> do
  x = "[" + x + "]";
print x;
```

$X_0 = \text{“a”}$

$X_1 = X_0 \sqcup X_2$

$X_2 = \text{“[”} \cdot X_1 \cdot \text{“]”}$

$X_3 = X_1$



s_0: $\rightarrow . S$; $S \rightarrow . \text{“[” } S \text{ “]”}$; $S \rightarrow . \text{“a”}$

s_1: $S \rightarrow \text{“[” } . S \text{ “]”}$; $S \rightarrow . \text{“[” } S \text{ “]”}$; $S \rightarrow . \text{“a”}$

s_2: $S \rightarrow \text{“a” } .$

s_3: $S \rightarrow \text{“[” } S \, . \text{ “]”}$

s_4: $S \rightarrow \text{“[” } S \text{ “]” } .$

s_5: $\rightarrow S \, .$

$X_3(s_0) = ?$

$X_3(s_0) = X_1(s_0) =$

$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) =$

$$\begin{aligned} X_0(s_0) &= \mathrm{goto}(s_0, \text{“a”}) \\ &= s_2 \\ &\quad [\text{reduce: } S \rightarrow \text{“a”}] \\ &= \mathrm{goto}(s_0, S) \\ &= s_5 \end{aligned}$$

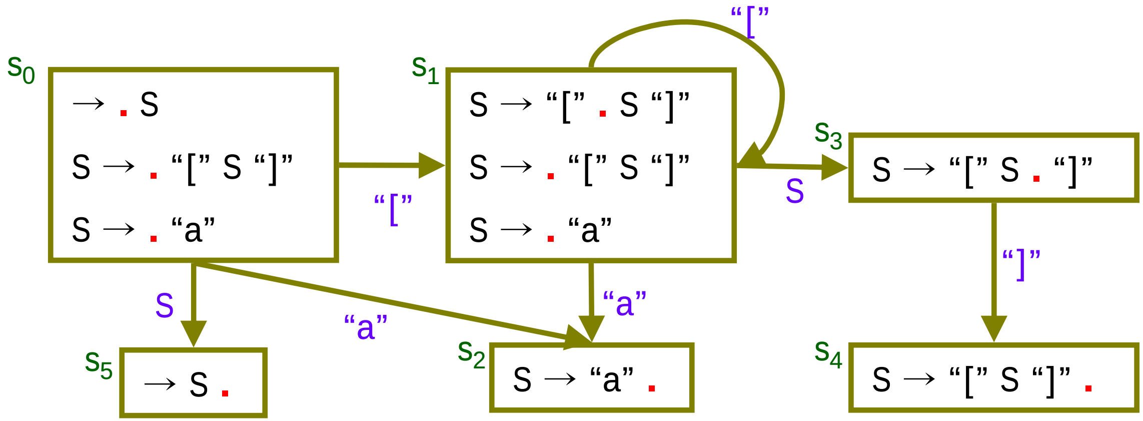
$$\begin{aligned} X_2(s_0) &= (\text{“[”} \cdot X_1 \cdot \text{“]”})(s_0) \\ &= \mathrm{goto}(s_0, \text{“[”}) * (X_1 \cdot \text{“]”}) \\ &= s_1 * (X_1 \cdot \text{“]”}) \\ &= s_1 \cdot X_1(s_1) * \text{“]”} \\ &= \end{aligned}$$

Abstract Parsing: Example 3

```
x = “a”;
while <cond> do
  x = “[” + x + “]”;
print x;
```

$X_0 = \text{“a”}$
$X_1 = X_0 \sqcup X_2$
$X_2 = \text{“[”} \cdot X_1 \cdot \text{“]”}$
$X_3 = X_1$

$X_3(s_0) = ?$


s_0
$\rightarrow . S$
$S \rightarrow . \text{“[”}\ S\ \text{“]”}$
$S \rightarrow . \text{“a”}$
s_1
$S \rightarrow \text{“[”}\ .\ S\ \text{“]”}$
$S \rightarrow . \text{“[”}\ S\ \text{“]”}$
$S \rightarrow . \text{“a”}$
“[”
“[”
S
s_3
$S \rightarrow \text{“[”}\ S\ .\ \text{“]”}$
“]”
S
“a”
“a”
s_5
$\rightarrow S .$
s_2
$S \rightarrow \text{“a”}\ .$
s_4
$S \rightarrow \text{“[”}\ S\ \text{“]”}\ .$

$X_1(s_1) = X_0(s_1) \sqcup X_2(s_1) =$

$X_0(s_1) = \text{goto}(s_1, \text{“a”})$
$= s_2$
[reduce: $S \rightarrow \text{“a”}$]
$= \text{goto}(s_1, S)$
$= s_3$

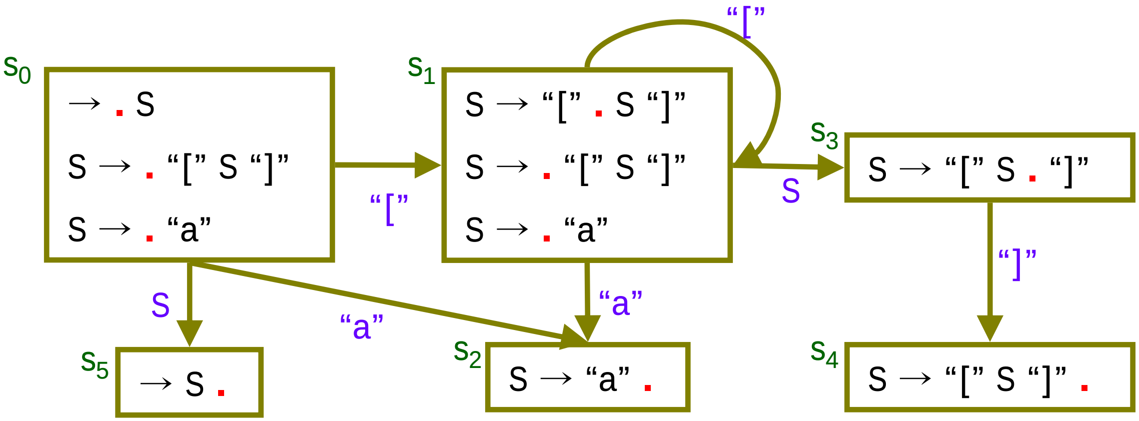
$X_2(s_1) = (\text{“[”} \cdot X_1 \cdot \text{“]”})(s_1)$
$= \text{goto}(s_1, \text{“[”}) * (X_1 \cdot \text{“]”})$
$= s_1 * (X_1 \cdot \text{“]”})$
$= s_1 \cdot X_1(s_1) * \text{“]”}$
$= \ldots$
$= s_1 \cdot s_3 \cdot s_4$

Abstract Parsing: Example 3

```
x = “a”;
while <cond> do
  x = “[” + x + “]”;
print x;
```

$X_0 = \text{“a”}$
$X_1 = X_0 \sqcup X_2$
$X_2 = \text{“[”} \cdot X_1 \cdot \text{“]”}$
$X_3 = X_1$

$X_3(s_0) = ?$



$X_1(s_1) = X_0(s_1) \sqcup X_2(s_1) = \{s_3,\ s_1 \cdot s_3 \cdot s_4\}$

$$\begin{aligned} X_0(s_1) &= \mathrm{goto}(s_1, \text{“a”}) \\ &= s_2 \\ &\quad [\text{reduce: } S \to \text{“a”}] \\ &= \mathrm{goto}(s_1, S) \\ &= s_3 \end{aligned}$$

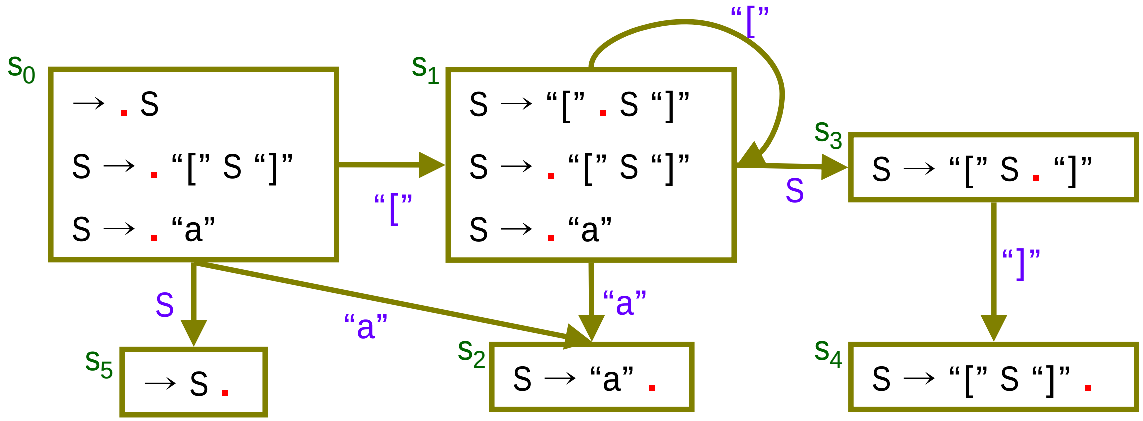
$$\begin{aligned} X_2(s_1) &= (\text{“[”} \cdot X_1 \cdot \text{“]”})(s_1) \\ &= \mathrm{goto}(s_1, \text{“[”}) * (X_1 \cdot \text{“]”}) \\ &= s_1 * (X_1 \cdot \text{“]”}) \\ &= s_1 \cdot X_1(s_1) * \text{“]”} \\ &= \ldots \\ &= s_1 \cdot s_3 \cdot s_4 \end{aligned}$$

Abstract Parsing: Example 3

```
x = "a";
while <cond> do
  x = "[" + x + "]";
print x;
```

X_0 = "a"
$X_1 = X_0 \sqcup X_2$
X_2 = "[" $\cdot X_1 \cdot$ "]"
$X_3 = X_1$

$X_3(s_0)$ = ?



$X_3(s_0) = X_1(s_0)$ =

$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0)$ =

$X_0(s_0)$ = goto(s_0, "a")
= s_2
[reduce: S → "a"]
= goto(s_0, S)
= s_5

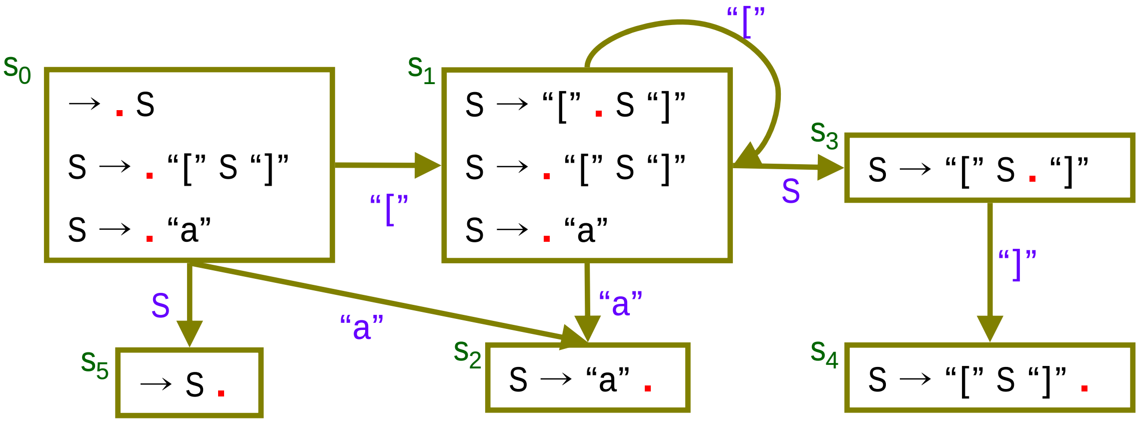
$X_2(s_0)$ = ("[" $\cdot X_1 \cdot$ "]")(s_0)
= goto(s_0, "[") * (X_1 · "]")
= s_1 * (X_1 · "]")
= $s_1 \cdot X_1(s_1)$ * "]"
= …
= $s_1 \cdot s_3$ * "]"
= $s_1 \cdot s_3$ · goto(s_3, "]")
= $s_1 \cdot s_3 \cdot s_4$ [reduce: S → "[" S "]"]
= goto(s0, S) = s_5

Abstract Parsing: Example 3

```
x = "a";
while <cond> do
  x = "[" + x + "]";
print x;
```

X_0 = "a"
$X_1 = X_0 \sqcup X_2$
X_2 = "[" $\cdot X_1 \cdot$ "]"
$X_3 = X_1$

$X_3(s_0)$ = ?



$X_3(s_0) = X_1(s_0) =$

$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) =$

$X_0(s_0)$ = goto(s_0, "a")
= s_2
[reduce: S → "a"]
= goto(s_0, S)
= s_5

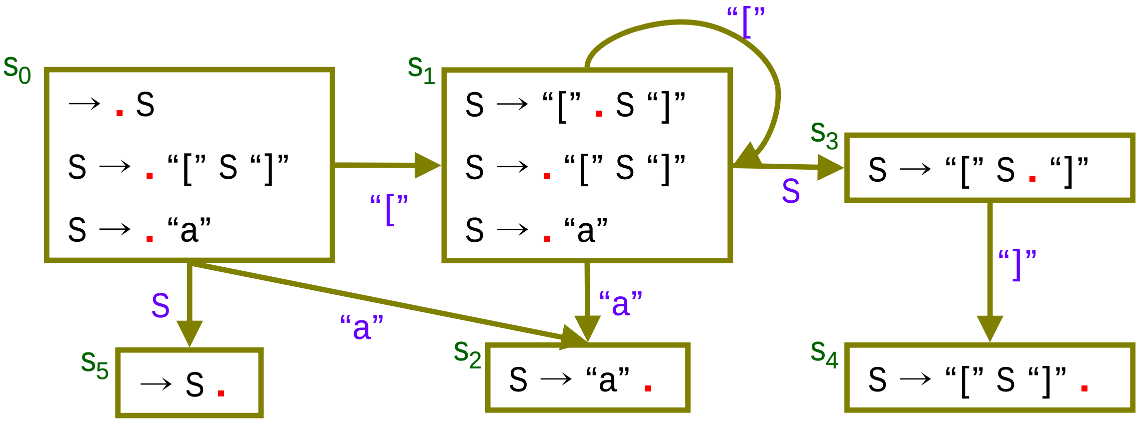
$X_2(s_0)$ = ("[" $\cdot X_1 \cdot$ "]")(s_0)
= goto(s_0, "[") * ($X_1 \cdot$ "]")
= $s_1 \cdot X_1(s_1)$ * "]"
= …
= $s_1 \cdot s_1 \cdot s_3 \cdot s_4$ * "]"
[reduce: S → "[" S "]"]
= s_1 · goto(s_1, S) * "]"
= $s_1 \cdot s_3$ * "]"
= $s_1 \cdot s_3 \cdot s_4$ = … = s_5

Abstract Parsing: Example 3

```
x = "a";
while <cond> do
  x = "[" + x + "]";
print x;
```

X_0 = "a"
$X_1 = X_0 \sqcup X_2$
X_2 = "[" · X_1 · "]"
$X_3 = X_1$

$X_3(s_0)$ = ?



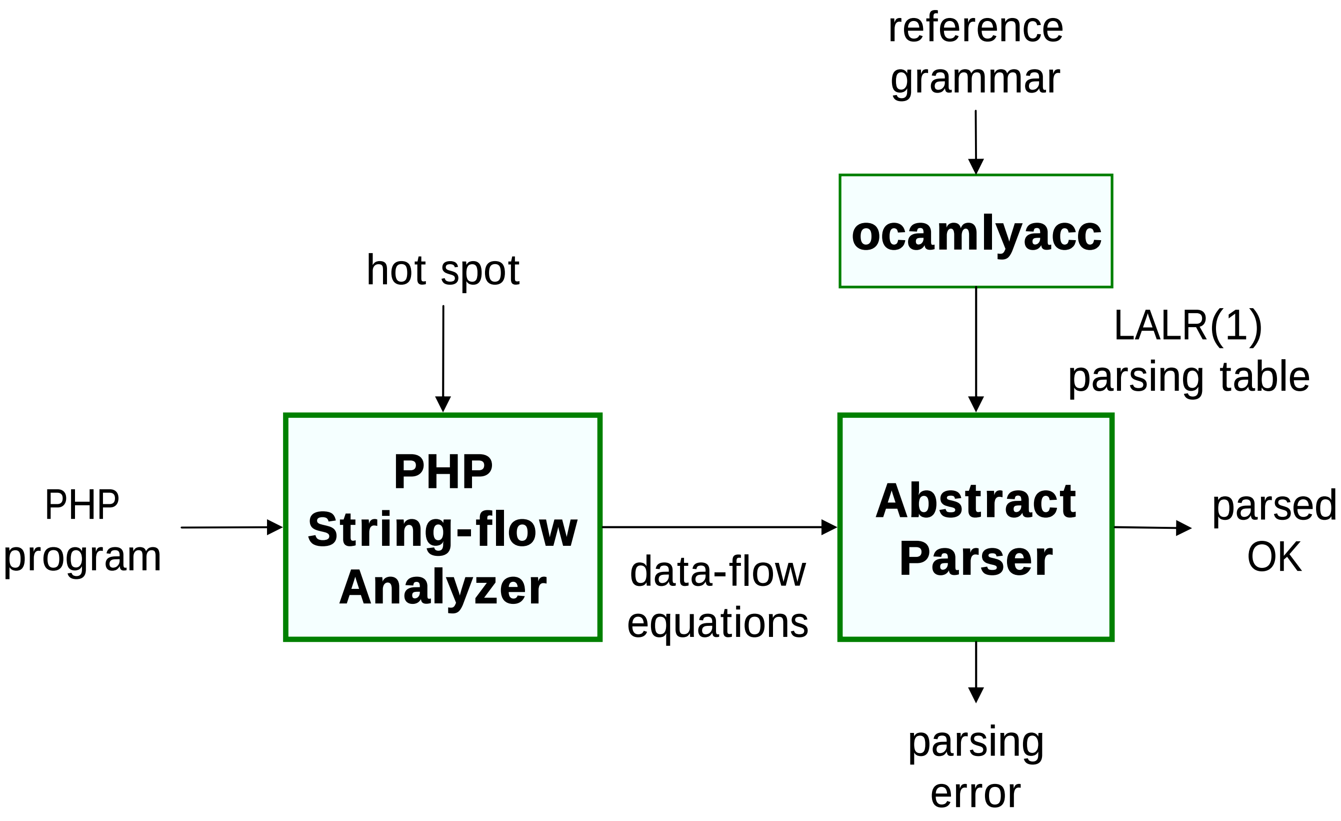
$X_3(s_0) = X_1(s_0) = s_5$

$X_1(s_0) = X_0(s_0) \sqcup X_2(s_0) = s_5$

$X_0(s_0)$ = goto(s_0, "a")
= s_2
[reduce: S → "a"]
= goto(s_0, S)
= s_5

$X_2(s_0)$ = ("[" · X_1 · "]")(s_0)
= goto(s_0, "[") * (X_1 · "]")
= s_1 · $X_1(s_1)$ * "]"
= …
= $s_1 \cdot s_1 \cdot s_3 \cdot s_4$ * "]"
[reduce: S → "[" S "]"]
= s_1 · goto(s_1, S) * "]"
= $s_1 \cdot s_3$ * "]"
= $s_1 \cdot s_3 \cdot s_4$ = … = s_5

Implementation



Conclusion

- Impact
  - ✓ improve the precision on the syntax analysis of dynamically generated strings
  - ✓ enhance the entire group of work based on previous string-analysis technique.
- Future work
  - ✓ Abstract semantic processing on parsed strings.
    - type checking on dynamically generated strings
    - static analysis on dynamically generated strings

Discussion

Kyung-Goo Doh