Script generated by TTT

Title: Petter: Compiler Construction (28.05.2020) -22: LR(0)-Parser

Date: Mon May 11 14:33:26 CEST 2020

Duration: 27:23 min

Pages: 18

LR(0)-Parser

Idea for a parser:

- The parser manages a viable prefix $\alpha = X_1 \dots X_m$ on the pushdown and uses $LR(G)$ to identify reduction spots.
- It can reduce with $A \to \gamma$, if $[A \to \gamma \bullet]$ is admissible for α

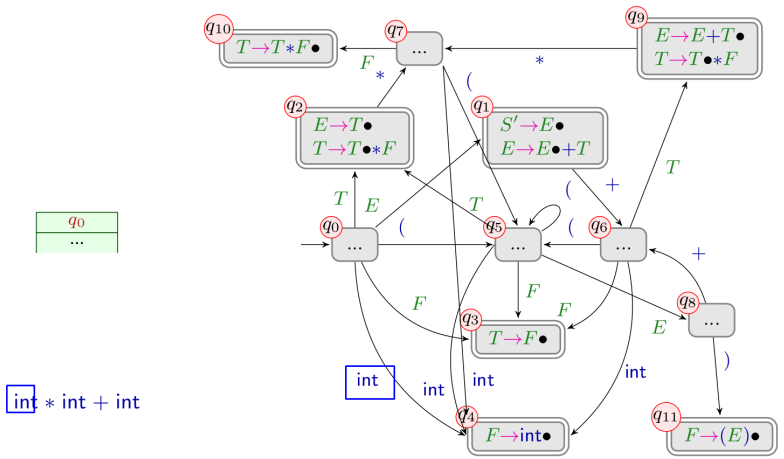
Optimization:

We push the states instead of the X_i in order not to process the pushdown's content with the automaton anew all the time.
Reduction with $A \to \gamma$ leads to popping the uppermost $|\gamma|$ states and continue with the state on top of the stack and input A.

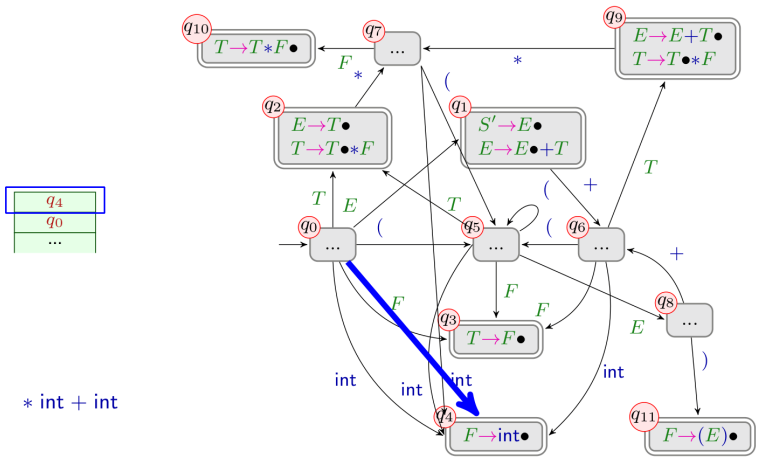
Attention:

This parser is only deterministic, if each final state of the canonical $LR(0)$-automaton is conflict free.

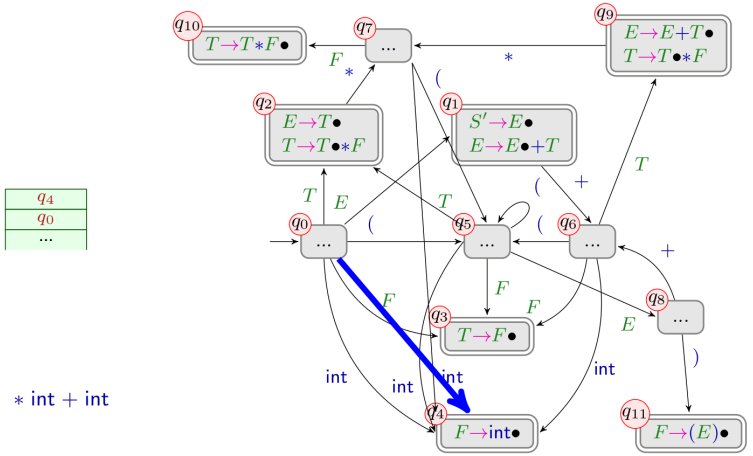
LR(0)-Parser – Example:



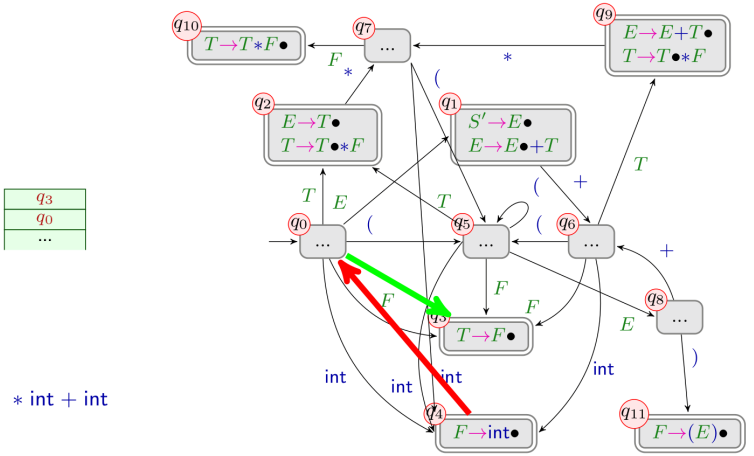
LR(0)-Parser – Example:



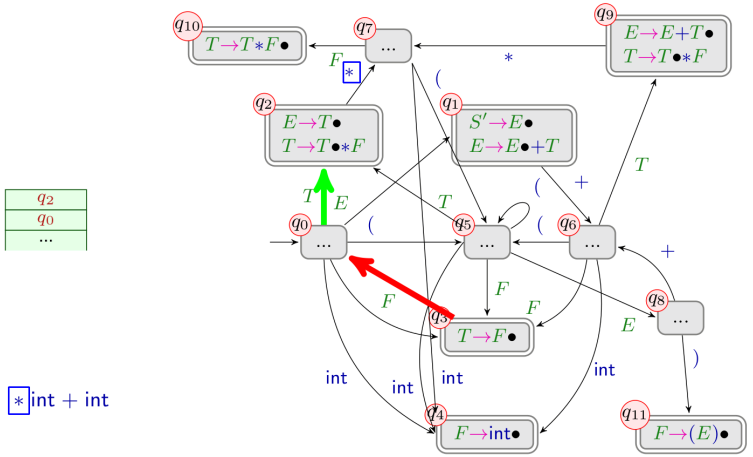
LR(0)-Parser – Example:



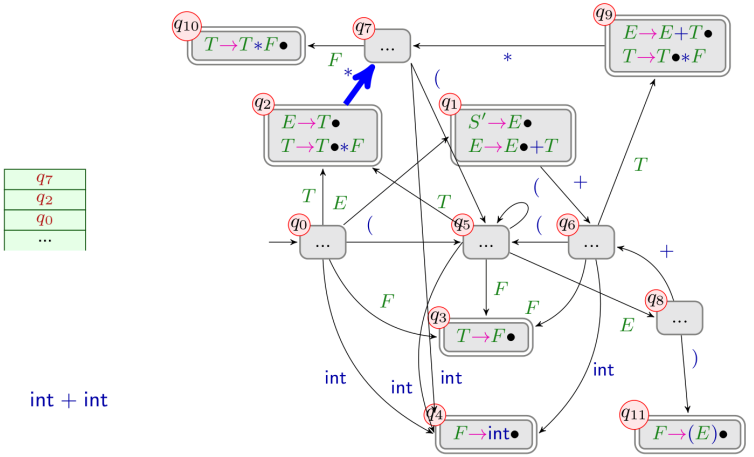
LR(0)-Parser – Example:



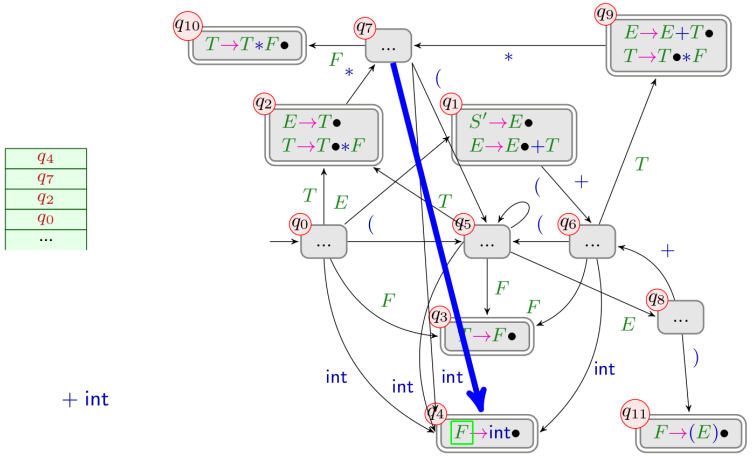
LR(0)-Parser – Example:



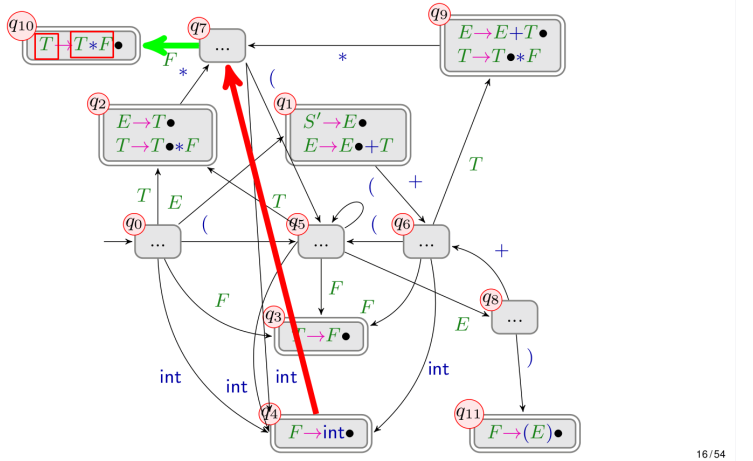
LR(0)-Parser – Example:



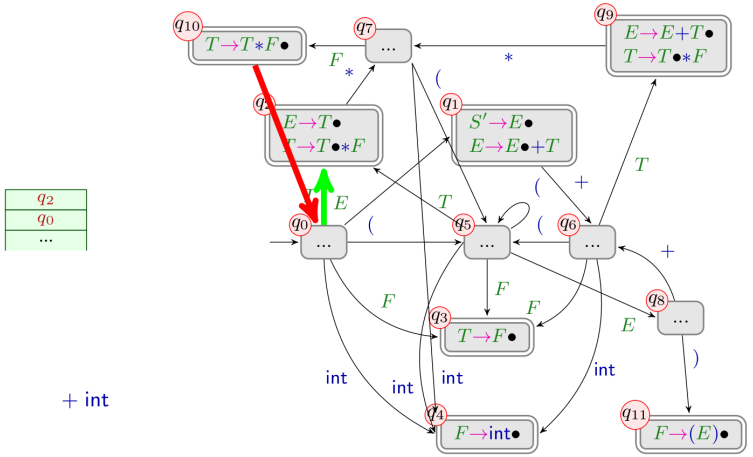
LR(0)-Parser – Example:



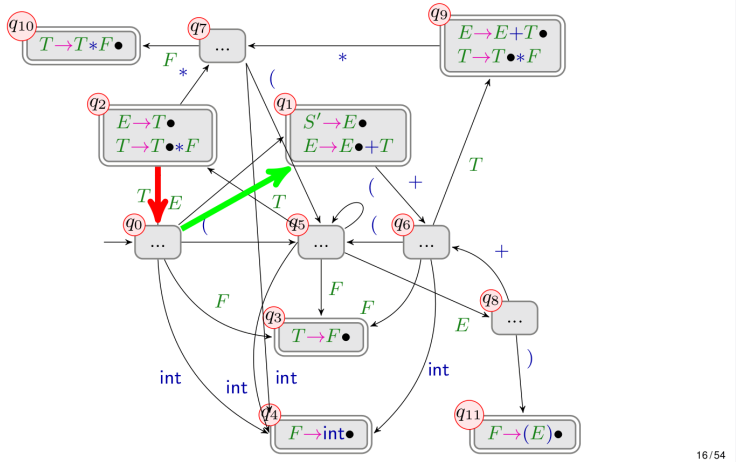
LR(0)-Parser – Example:



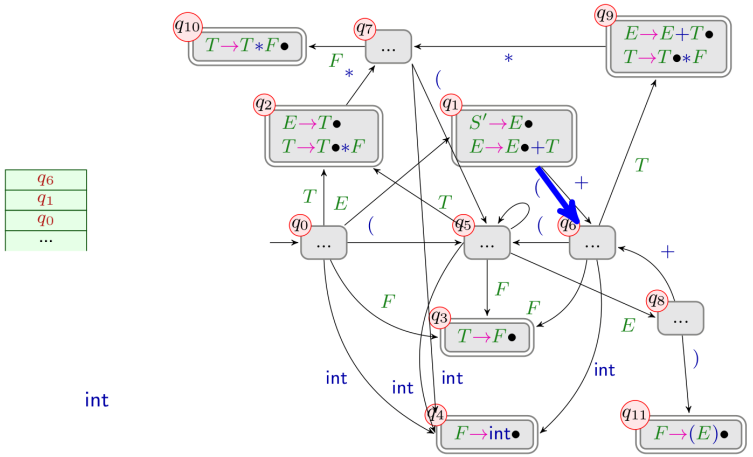
LR(0)-Parser – Example:



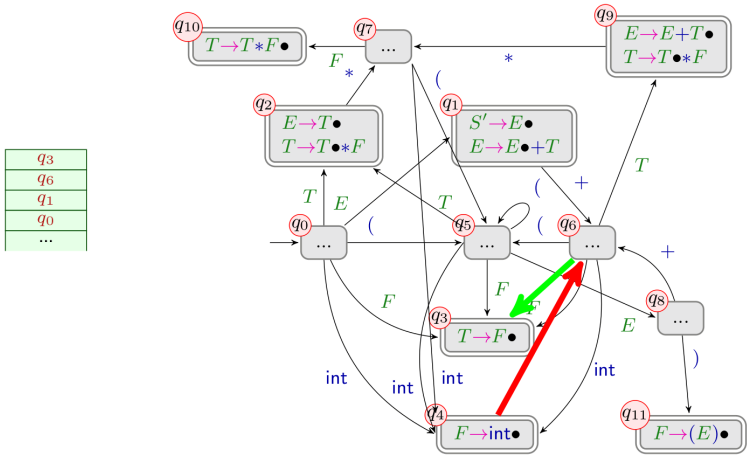
LR(0)-Parser – Example:



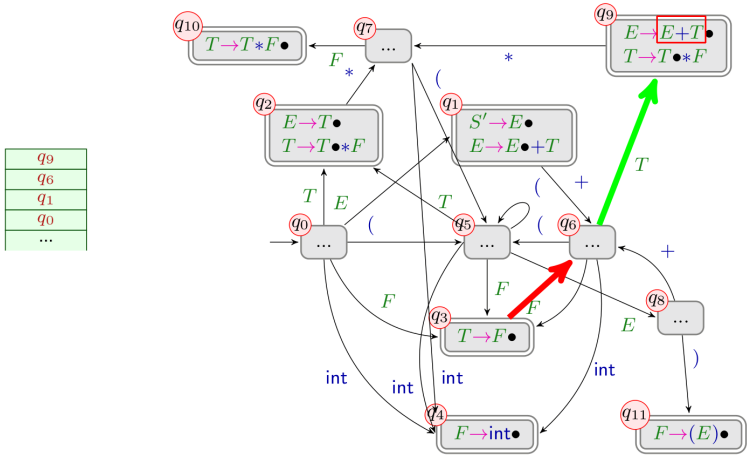
LR(0)-Parser – Example:



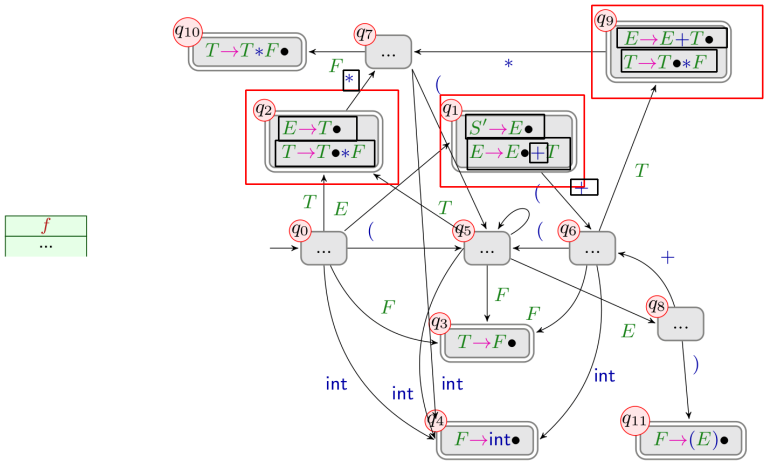
LR(0)-Parser – Example:



LR(0)-Parser – Example:

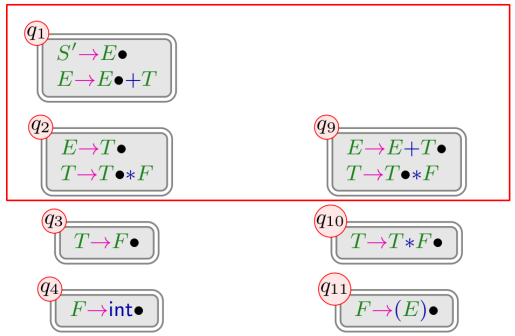


LR(0)-Parser – Example:



LR(0)-Parser

... we observe:



The final states q_1, q_2, q_9 contain more than one admissible item

$\Rightarrow$ non-deterministic!

LR(0)-Parser

The construction of the $LR(0)$-parser:

States: $Q \cup \{f\}$ (f fresh)

Start state: q_0

Final state: f

Transitions:

Shift: $(p\,a\,,p\,q)$ if $q = \delta(p, a) \neq \emptyset$

Reduce: $(p\,q_1 \ldots q_m, \epsilon, p\,q)$ if $[A \to X_1 \ldots X_m \bullet] \in q_m$, $q = \delta(p, A)$

Finish: $(q_0\,p, \epsilon, f)$ if $[S' \to S\bullet] \in p$

with the canonical automaton $LR(G) = (Q, T, \delta, q_0, F)$.

LR(0)-Parser

Correctness:

we show:

The accepting computations of an $LR(0)$-parser are one-to-one related to those of a shift-reduce parser M_G^R.

we conclude:

- The accepted language is exactly $\mathcal{L}(G)$
- The sequence of reductions of an accepting computation for a word $w \in T$ yields a reverse rightmost derivation of G for w