Optimalised grammars in Chomsky Normal Form

Jan Horáček

VUT v Brně
Fakulta informačních technologií

November 9, 2009

Review

Context free grammar

Grammar $G = (N, T, P, S)$ is CFG if every $p \in P$ is a finite relation from N to $(N \cup T)^*$.

Chomsky Normal Form

Type-2 grammar $G = (N, T, P, S)$ is in CNF if every rule $p \in P$ has one of these forms:

1. $A \rightarrow BC$
2. $A \rightarrow a$
3. $S \rightarrow \varepsilon$

Where $A, B, C, S \in N$ and $a \in T$

Often problem

1. We have original context-free grammar and we want to translate it into Chomsky Normal Form.
2. We have already translated grammar in CNF and we want to reduce number of rules or nonterminals.
   (harder??? - may be, but...)

Clasic conversion

1. For each terminal symbol $a \in T$:
   - Add new A' into N.
   - Add new rule $A' \rightarrow a$ into P.
   - Find all rules, where a occurs and replace terminal symbol a to nonterminal A' (skip $A' \rightarrow a$ rules).
2. Convert $A \rightarrow B_1 \dots B_n \in P$ rules as:

$$\begin{array}{rcl} A & \rightarrow & B_1 \langle B_2 \dots B_n \rangle \\ \langle B_2 \dots B_n \rangle & \rightarrow & B_2 \langle B_3 \dots B_n \rangle \\ & \vdots & \\ \langle B_{n-2} \dots B_n \rangle & \rightarrow & B_{n-2} \langle B_{n-1} B_n \rangle \\ \langle B_{n-1} B_n \rangle & \rightarrow & B_{n-1} B_n \end{array}$$

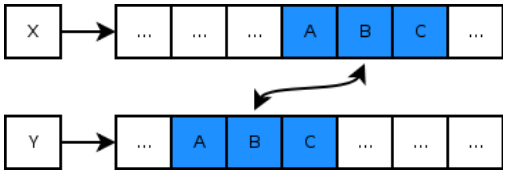
And add newly created nonterminals into N.

Characteristics

- As we showed last time, if we use this algorithm, we can get same or less rules then if we use tree algorithms. (good algorithm)
- Used very often. (We have allmost all grammars reduced in this way)
- New nonterminals are lineary dependent from left to right. (realy realy usefull)

Matching parts from different rules

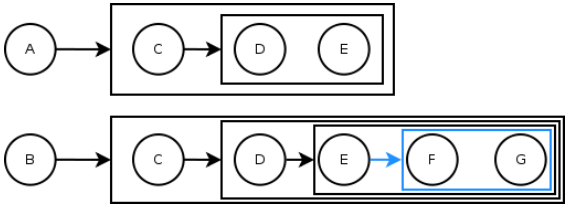
- How can we reduce number of rules - matching parts:



- Lets consider rules:

  $A \rightarrow \mathbf{CDE}$

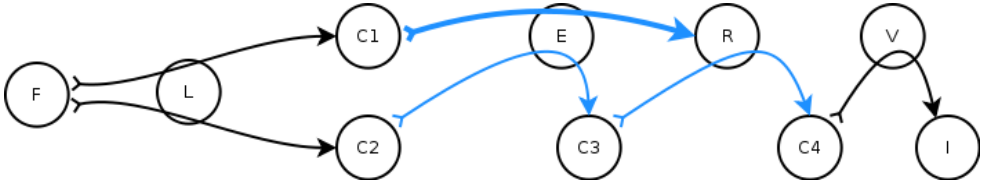
  $B \rightarrow \mathbf{CDE}FG$



- Can't we make another nonterminal for **CDE** sequence?

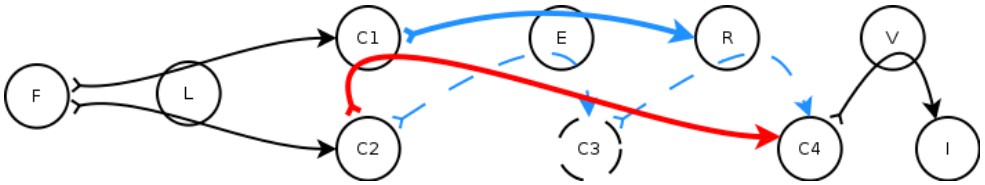
Simple grammar - synchronising

- Rules:
  $F \rightarrow LER$
  $F \rightarrow LERVI$
- Converted into CNF:
  $F \rightarrow LC_1$
  $F \rightarrow LC_2$
  $C_1 \rightarrow ER$
  $C_2 \rightarrow EC_3$
  $C_3 \rightarrow RC_4$
  $C_4 \rightarrow VI$



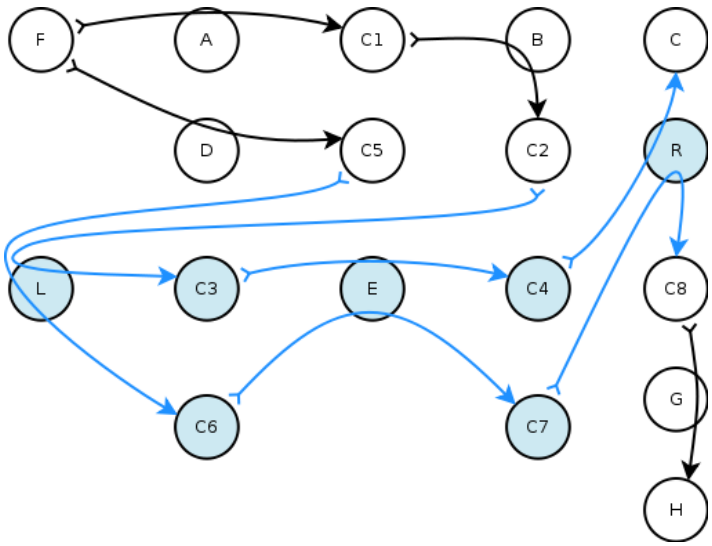
Optimalised grammar - synchronising

- Rules:
  $F \rightarrow LER$
  $F \rightarrow LERVI$
- Optimalised CNF:
  $F \rightarrow LC_1$
  $F \rightarrow LC_2$
  $C_1 \rightarrow ER$
  $C_2 \rightarrow C_1C_4$
  $C_4 \rightarrow VI$



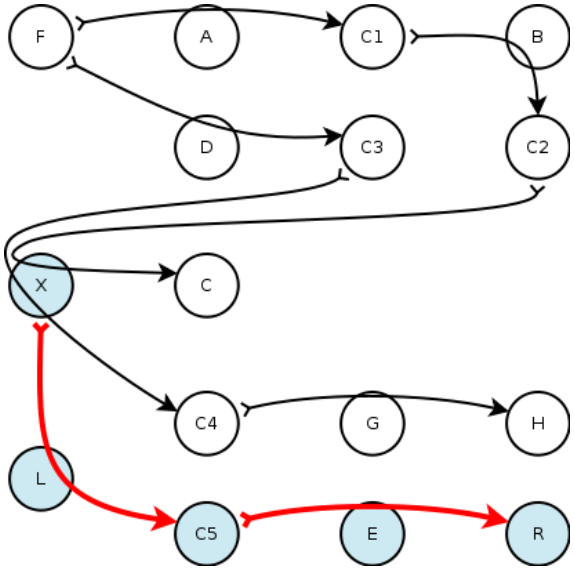
Simple grammar 2 - packing

- Rules:
  $F \rightarrow ABLERC$
  $F \rightarrow DLERGH$
- Converted into CNF (graphics form):



Optimalised grammar 2 - packing

- Rules:
  $F \rightarrow ABLERC$
  $F \rightarrow DLERGH$
- We can reduce it as:
  $F \rightarrow ABXC$
  $F \rightarrow DXGH$
  $X \rightarrow LER$



Real example

Grammar with rules

$I \rightarrow a|b|Ia|Ib$
$F \rightarrow I|(E)|(E)\,\hat{}\,I$
$T \rightarrow F|T * F$
$E \rightarrow T|E + T$

Safe reduction:

- Remove ε-production ($\Theta(2^n)$)
- Remove unit productions ($\Theta(n^2)$)
- Remove useless symbols ($\Theta(n)$)
- CNF ($\Theta(n)$)

Reduction

- Remove ε-production - done.
- Remove unit productions:

  $I \rightarrow a|b|Ia|Ib$

  $F \rightarrow a|b|Ia|Ib|(E)|(E)\hat{}\,I$

  $T \rightarrow a|b|Ia|Ib|(E)|(E)\hat{}\,I|T*F$

  $E \rightarrow a|b|Ia|Ib|(E)|(E)\hat{}\,I|T*F|E+T$
- Remove useless symbols - done.
- Terminals into nonterminals:

  $I \rightarrow a|b|IA|IB$

  $F \rightarrow a|b|IA|IB|LER|LERVI$

  $T \rightarrow a|b|IA|IB|LER|LERVI|TMF$

  $E \rightarrow a|b|IA|IB|LER|LERVI|TMF|EPT$

  $A \rightarrow a$ $\qquad$ $P \rightarrow +$

  $B \rightarrow b$ $\qquad$ $L \rightarrow ($

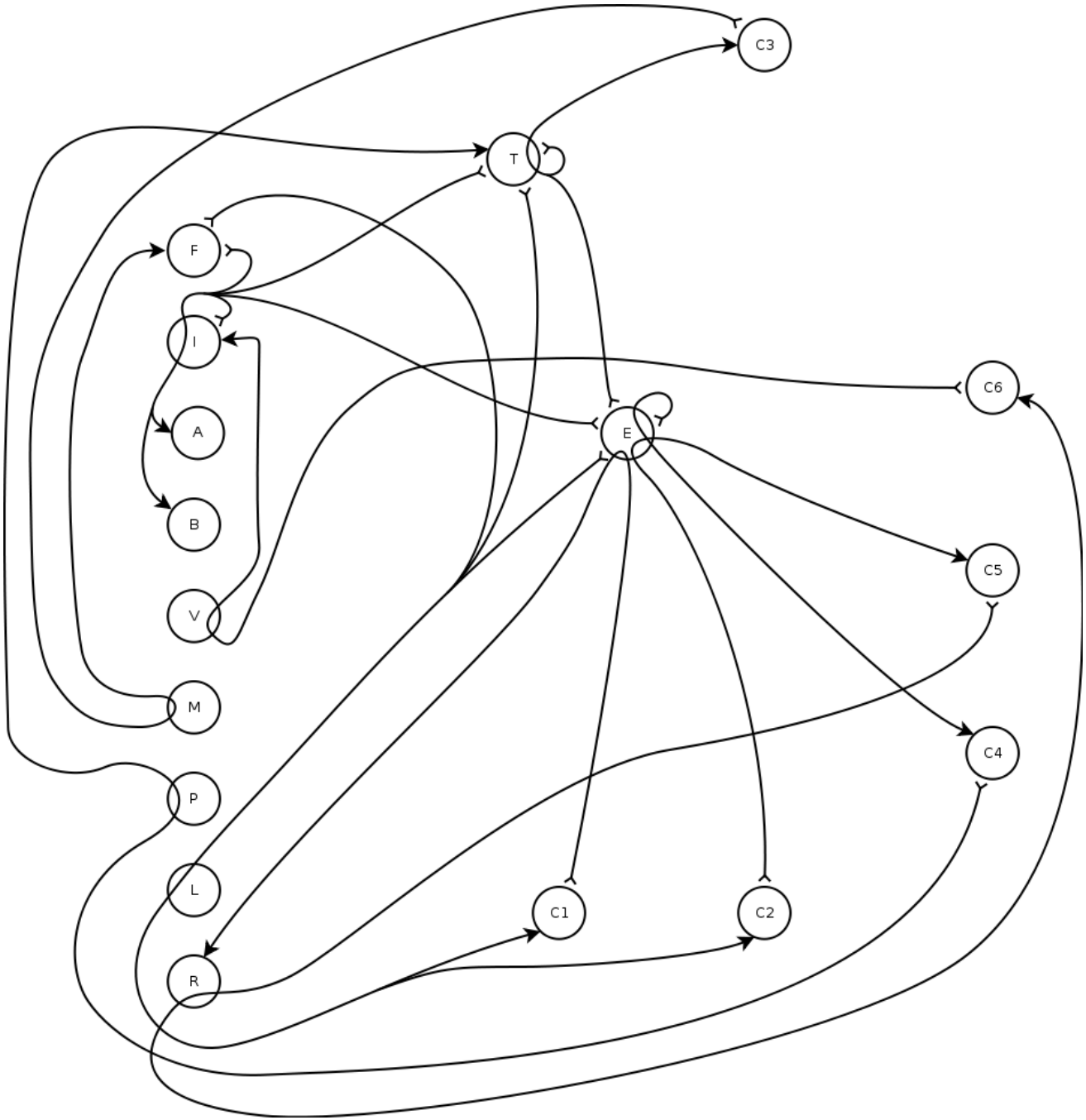
  $V \rightarrow \hat{}$ $\qquad$ $R \rightarrow)$

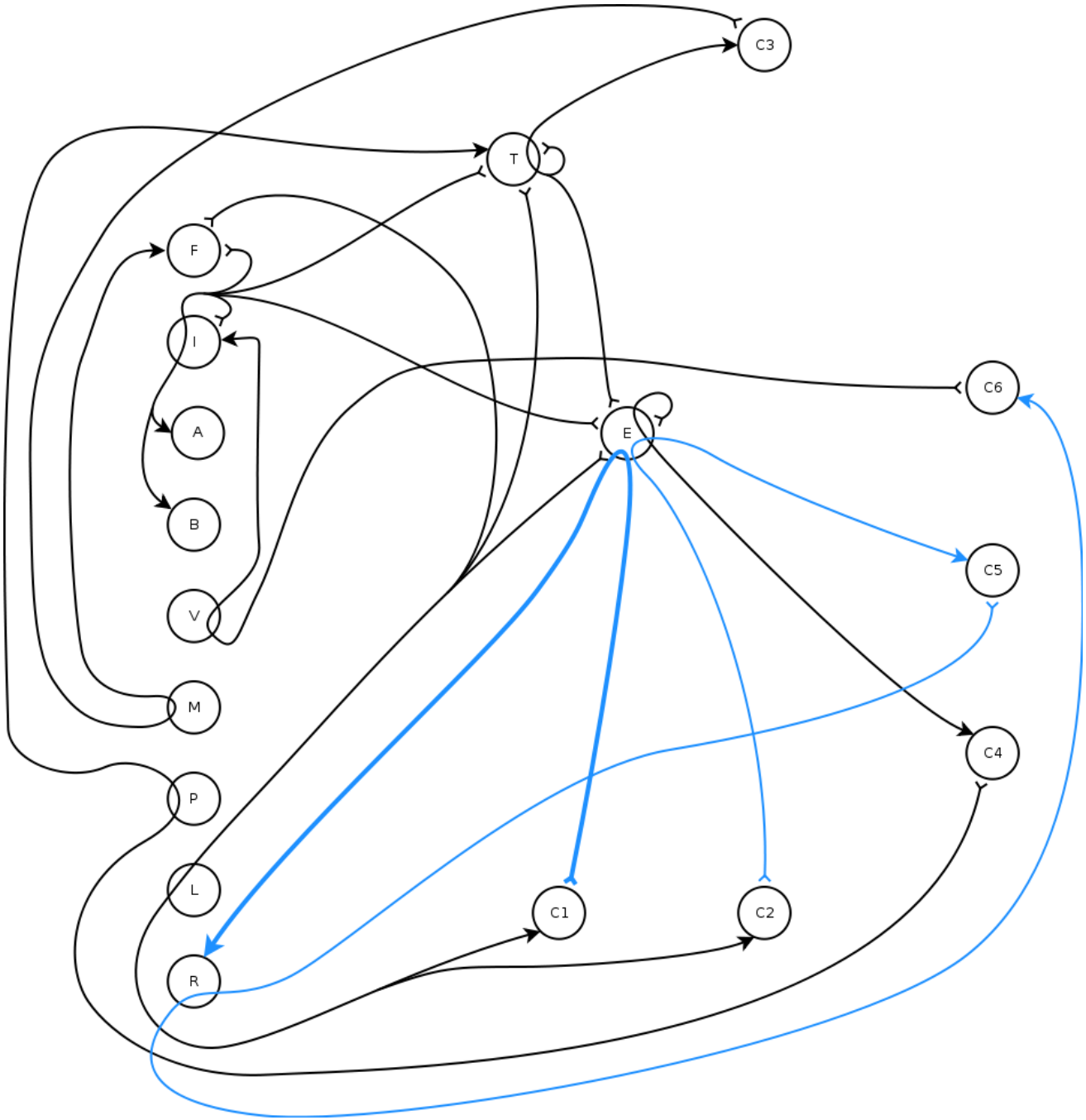
  $M \rightarrow *$

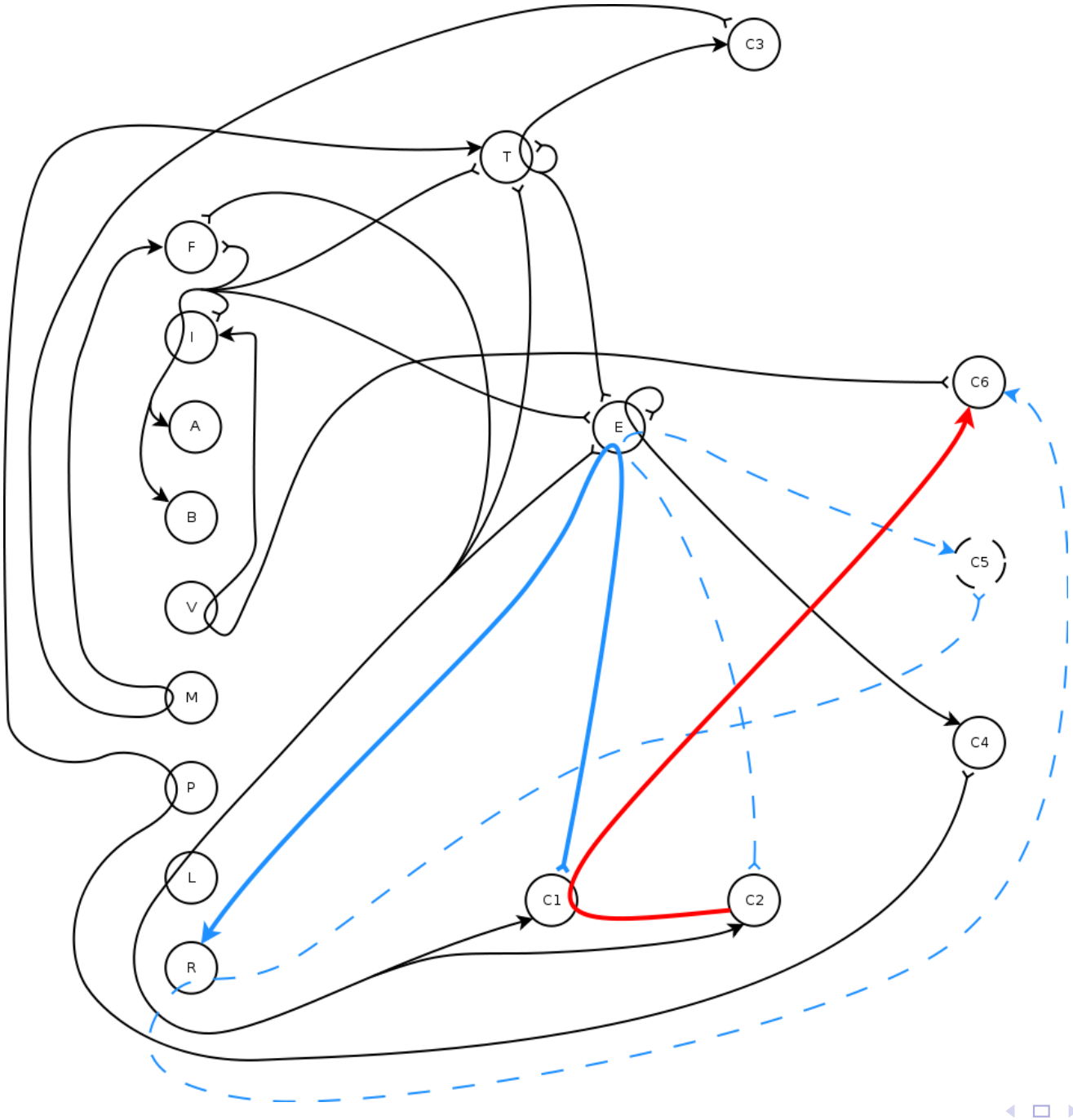
Grammar in CNF

$I \rightarrow a|b|IA|IB$
$F \rightarrow a|b|IA|IB|LC_1|LC_2$
$T \rightarrow a|b|IA|IB|LC_1|LC_2|TC_3$
$E \rightarrow a|b|IA|IB|LC_1|LC_2|TC_3|EC_4$
$A \rightarrow a$
$B \rightarrow b$
$V \rightarrow \hat{}$
$M \rightarrow *$
$P \rightarrow +$
$L \rightarrow ($
$R \rightarrow)$

$C_1 \rightarrow ER$
$C_2 \rightarrow EC_5$
$C_3 \rightarrow MF$
$C_4 \rightarrow PT$
$C_5 \rightarrow RC_6$
$C_6 \rightarrow VI$


C3
T
F
I
A
B
V
M
P
L
R
E
C6
C5
C4
C1
C2

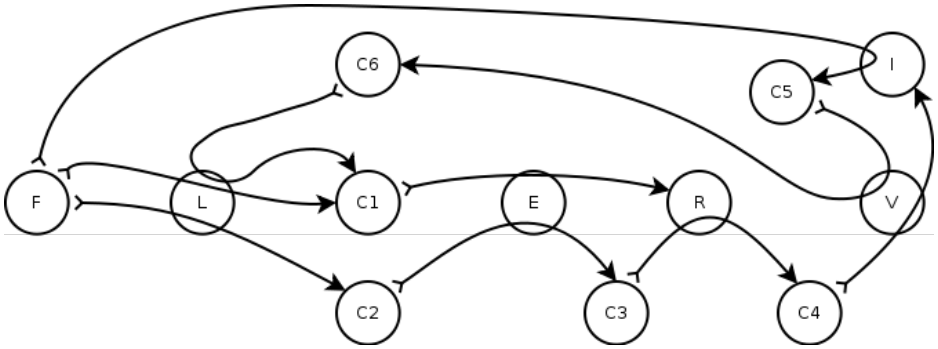

C3
T
F
I
A
B
V
M
P
L
R
E
C6
C5
C4
C1
C2

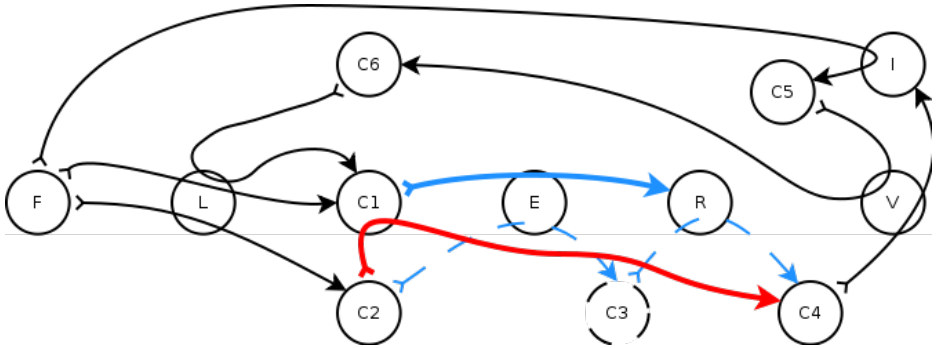

C3
T
F
I
A
B
V
M
P
L
R
E
C6
C5
C4
C1
C2

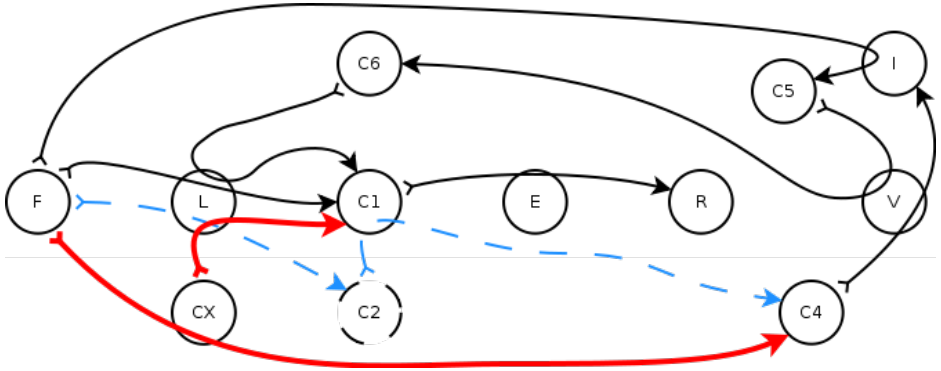
Real example 2

Grammar with rules

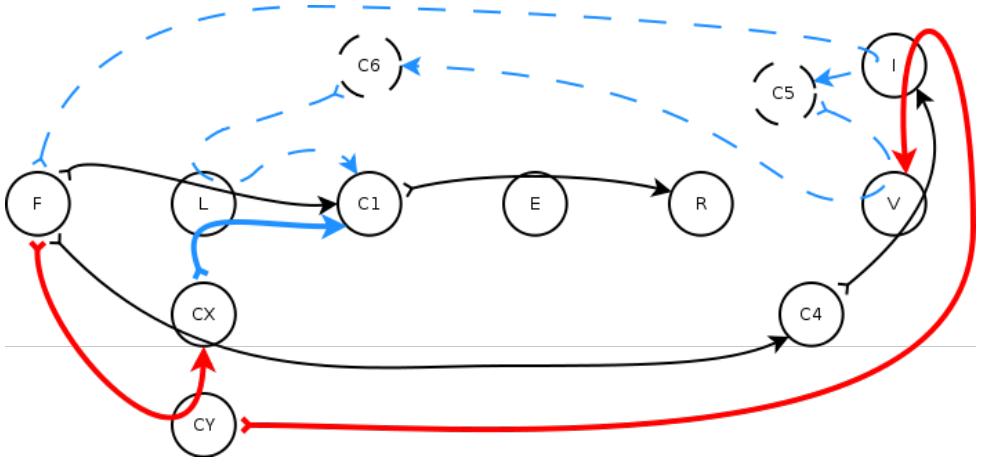
$F \rightarrow (E)$
$F \rightarrow (E) \hat{} I$
$F \rightarrow I \hat{} (E)$
We don't care about other nonterminals now (Same as in previous example)




C6
C5
I
F
L
C1
E
R
V
C2
C3
C4


C6
C5
I
F
L
C1
E
R
V
CX
C2
C4

Reduced grammar - our goal



Conclusion

Does it work?

- At the start we had 12 nonterminals and 9 rules.
- Reduced grammar after "synchronising" has 11 nonterminals and 8 rules.
- Reduced grammar after "packing" has only 10 nonterminals and 7 rules.

We proved that it is not bad if we have grammar in CNF and we want to optimalise it. In fact it is usefull (this point of view can solve dependency from parrents nonterminals).

Literature

- Hopcroft J., Motwani R., Ullman J.: *Introduction to Automata Theory, Languages, and Computation*; 3rd Edition; Pearson Education, Inc.; 2007; ISBN 0-321-47617-4