Algorithm 1 CNF

function ChomskyNormalForm($G :< NT, T, \rho >$) ▷

G : Grammaire, ρ : productions probabilis

$P' = \rho.copy()$; ▷ ρ est un dictionnaire de dictionnaire de fractions

function BINARISER($nt, p, proba$)

if |p| > 2 **then**

$nt^{\alpha} \leftarrow join(\downarrow, p[1:])$;

$NT = NT | nt^{\alpha}$;

$P'[nt][(p[0], nt^{\alpha})] \leftarrow proba$;

BINARISER($nt^{\alpha}, p[1:]$);

else

$P'[nt][p] \leftarrow proba$;

for $nt \in \rho$ **do**

for $p \in \rho[nt]$ **do**

if $|p| > 2$ **then**

BINARISER($nt, p, \rho[nt][p]$);

$cnf = cnf - cnf[nt][p]$;

for all $nt^A \rightarrow nt^B; \rho^1 \in P$ **do**

$nt^{\alpha} =$ JOIN($\uparrow, nt^A, nt^B$);

$NT \uplus nt^{\alpha}$;

for all $nt^C \rightarrow \alpha, nt^A, \gamma; \rho^2 \in P$ **do**

$cnf \uplus cnf[nt^C][(\alpha, nt^{\alpha}, \gamma)] = \rho^1 * \rho^2$;

$cnf[nt^C][(\alpha, nt^A, \gamma)] = (1 - \rho 1) * \rho^2$;

for all $nt^A \rightarrow \alpha; \rho^3 \in P$ **do**

$\rho^3 = \rho^3 / (1 - \rho^1)$;

for all $nt^B \rightarrow \alpha; \rho^4 \in P$ **do**

$cnf \uplus cnf[nt^{\alpha}][(\alpha)] = \rho^4$;

return $G' :< NT, T, P' >$