1.5 Théorèmes d'équivalence

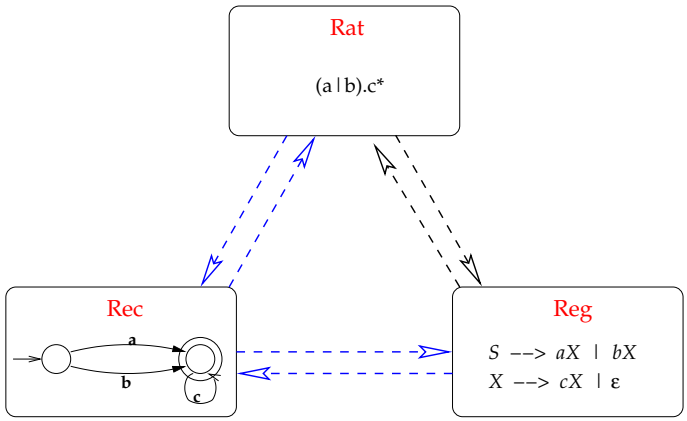
1.5.1 Le théorème triangulaire

On a établi (Kleene y a contribué) une ensemble de résultats d'équivalence que l'on peut résumer de la façon suivante[2]

$$\mathcal{L}_{\text{Rec}} = \mathcal{L}_{\text{Rat}} = \mathcal{L}_{\text{Reg}}$$

où
– $\mathcal{L}_{\text{Rec}}$ est la classe des langages **rec**onnaissables par un automate à nombre fini d'états ;
– $\mathcal{L}_{\text{Rat}}$ est la classe des langages que l'on peut décrire avec une expression **rat**ionnelle ;
– $\mathcal{L}_{\text{Reg}}$ est la classe des langages engendrés par une grammaire **rég**ulière.
On symbolise en général ce résultat sous la forme du triangle représenté à la figure 1.2.

FIG. 1.2 – Théorème d'équivalence



La démonstration du thèorème peut se faire de manière *constructive* : par exemple, pour montrer que tout langage rationnel est reconnaissable, il suffit d'exhiber un algorithme qui prenant une expression rationnelle quelconque en entrée, produit en sortie un automate qui reconnaît le même langage. Outre la difficulté de définir l'algorithme, il faut pour que la démonstration soit valide, d'une part garantir que l'algorithme fournit une réponse pour toute entrée possible, et d'autre part démontrer que l'automate fourni reconnaît bien le même langage.

Nous ne verrons pas ici ces deux derniers aspects de la démonstration (qui sont assez techniques), nous nous contenterons de donner (sous forme d'exemples) les algorithmes pour certaines des flèches pointillées de la figure (en bleu). Les algorithmes manquants existent, mais ils sont théoriquement inutiles si les deux autres équivalences sont établies.

Plus précisément, nous définirons les algorithmes permettant de démontrer :

$\mathcal{L}_{\text{Rec}} \subset \mathcal{L}_{\text{Reg}}$ Algorithme construisant une grammaire régulière à partir d'un automate, en identifiant les non-terminaux de la grammaire et les états de l'automate. (§ 1.5.2.2)

$\mathcal{L}_{\text{Reg}} \subset \mathcal{L}_{\text{Rec}}$ Algorithme très proche du précédent, toujours basé sur l'identité entre symbole non-terminal et état. (§ 1.5.2.3)

$\mathcal{L}_{\text{Rat}} \subset \mathcal{L}_{\text{Rec}}$ Algorithme basé sur la décomposition syntaxique de l'expression rationnelle, et la composition d'automates correspondants. (§ 1.5.3.1)

$\mathcal{L}_{\text{Rec}} \subset \mathcal{L}_{\text{Rat}}$ Algorithme de Mac Naughton et Yamada, qui construit itérativement, en partant de l'automate initial, un *automate fini généralisé* qui finit par ne contenir qu'une transition étiquetée par une expression rationnelle équivalente. (§ 1.5.3.2)

[2] Le théorème de Kleene correspond à l'équation : $\mathcal{L}_{\text{Rec}} = \mathcal{L}_{\text{Rat}}$.