▷ **Exercice 3 (9 points) :**

Le but de l'exercice est de générer des automates associés aux expressions régulières *non vides* sur le vocabulaire $V = \{a, b, c, d\}$. On considère donc une structure d'AST (arbres abstraits) pour ces expressions dont le typage et la sémantique sont donnés par le système d'attributs d'axiome $\mathbf{E}{\uparrow}\mathcal{P}(V^*)$ ci-dessous, où le terminal "$\mathtt{r}{\uparrow}x$" représente un caractère x de V *reconnu* par l'analyseur lexical.

$$\begin{array}{llll}
\mathbf{E}{\uparrow}\ell & ::= & \mathtt{e} & \ell := \{\epsilon\} \\
& | & \mathtt{r}{\uparrow}x & \ell := \{x\} \\
& | & \mathtt{.}\ \mathbf{E}{\uparrow}\ell_1\ \mathbf{E}{\uparrow}\ell_2 & \ell := \ell_1.\ell_2 \\
& | & \mathtt{+}\ \mathbf{E}{\uparrow}\ell_1\ \mathbf{E}{\uparrow}\ell_2 & \ell := \ell_1 \cup \ell_2 \\
& | & \mathtt{*}\ \mathbf{E}{\uparrow}\ell_1 & \ell := \ell_1^*
\end{array}$$

1) Donner un AST t_1 (ou un arbre d'analyse) qui engendre le langage correspondant à l'expression régulière "$(a.b + (c.(d + \epsilon))^*).a$".

2) On recherche un système d'attributs de profil $\mathbf{E}{\downarrow}\mathbb{N}{\downarrow}\mathbb{N}{\downarrow}\mathbb{N}{\uparrow}\mathbb{N}{\uparrow}\mathcal{P}(\mathbb{N}\times(V \cup \{\epsilon\})\times\mathbb{N})$ tel que pour tout arbre $t{\downarrow}i{\downarrow}f{\downarrow}n{\uparrow}n'{\uparrow}\delta$, si $\max(i, f) < n$ alors en posant $Q \stackrel{def}{=} \{i, f\} \cup [n, n'-1]$ les propriétés suivantes soient vérifiées :

- $n \leq n'$ et $\delta \subseteq (\{i\} \cup [n, n'-1])\times(V \cup \{\epsilon\})\times(\{f\} \cup [n, n'-1])$
- soit A l'automate fini non-déterministe, ayant Q comme ensemble d'états, i comme état initial, f comme unique état final, et δ comme ensemble de transitions ; soit ℓ le langage tel que $t{\uparrow}\ell$ pour la sémantique ci-dessus ; si $i \neq f$ alors A reconnaît exactement le langage ℓ, sinon A reconnaît exactement le langage ℓ^*.

Ce système d'attributs est basé sur les deux principes suivants. **Principe 1 :** comme toutes les expressions régulières représentées engendrent un langage non vide, les automates correspondants ont nécessairement au moins un état final ; quitte à ajouter des epsilon-transitions, on peut imposer à ces automates de n'avoir qu'un seul état final (ce qui simplifie les compositions d'automates). **Principe 2 :** les noms de l'état initial et de l'état final étant imposés par les paramètres i et f, les paramètres n et n' permettent de contrôler le nommage des autres états, pour éviter certains renommages lors des compositions d'automates.

Pour commencer, on considère le sous-système réduit aux 2 règles ci-dessous

$$\begin{array}{lll}
\mathbf{E}{\downarrow}i{\downarrow}f{\downarrow}n{\uparrow}n'{\uparrow}\delta & ::= & \mathtt{r}{\uparrow}x \\
& & \quad n' := n;\ \delta := \{(i, x, f)\} \\
& | & \mathtt{.}\ \mathbf{E}{\downarrow}i{\downarrow}n{\downarrow}n{+}1{\uparrow}n_1{\uparrow}\delta_1\ \mathbf{E}{\downarrow}n{\downarrow}f{\downarrow}n_1{\uparrow}n'{\uparrow}\delta_2 \\
& & \quad \delta := \delta_1 \cup \delta_2
\end{array}$$

Soit t_2 l'AST correspondant à "$\mathtt{.}\ \mathtt{r}{\uparrow}a\ \mathtt{.}\ \mathtt{r}{\uparrow}b\,\mathtt{r}{\uparrow}c$". Dessiner la propagation d'attributs correspondant à $t_2{\downarrow}0{\downarrow}2{\downarrow}5{\uparrow}n{\uparrow}\delta$. Que valent n, δ en sortie ? Dessiner l'automate résultant en nommant bien les états avec les numéros alloués par le système d'attributs.

3) Dessiner l'automate produit par le calcul $t_2{\downarrow}2{\downarrow}2{\downarrow}5{\uparrow}n{\uparrow}\delta$ en nommant bien les états avec les numéros alloués par le système d'attributs.

4) Étendre le système ci-dessus aux trois autres constructeurs de la structure d'AST.

5) Soit t_1 l'AST donné à l'item 1), dessiner l'automate produit par $t_1{\downarrow}0{\downarrow}1{\downarrow}2{\uparrow}n{\uparrow}\delta$ en nommant bien les états avec les numéros alloués par le système d'attributs.