

Calcul de PREMIER et SUIVANT

Ref: Wilhelm, Les compilateurs, théorie, construction, génération.

- Dans le cas général, on a le système récursif d'égalité suivant pour def PREMIER_N:

$$\text{PREMIER}_N(X) = \bigcup_{\{p \mid p[0] = X\}} F_p(\text{PREMIER}_N(p[1]), \dots, \text{PREMIER}_N(p[n_p])) \quad \forall X \in V_N$$

$$\text{où } F_p(d_1, \dots, d_{n_p}) = \{u_0 \oplus_k d_1 \oplus_k \dots \oplus_k d_{n_p} \oplus_k u_{n_p}\}$$

$$\text{si } p = (X_0 \rightarrow u_0 X_1 u_1 X_2 \dots X_{n_p} u_{n_p})$$

$$\text{où } L \oplus_k L' = \{x \oplus_k y \mid x \in L, y \in L'\} = \{k \circ xy, x \in L, y \in L'\}$$

- Dans le cas $k=1$, on a le résultat suivant:

Lemme: Étant donné deux langages $L_1, L_2 \subseteq V^{\leq 1}$, on a:

$$(a) \quad L_1 \oplus_1 L_2 = \begin{cases} \emptyset & \text{si } L_2 = \emptyset \\ L_1 & \text{si } L_2 \neq \emptyset \text{ et } \epsilon \notin L_1 \\ (L_1 - \{\epsilon\}) \cup L_2 & \text{si } L_2 \neq \emptyset \text{ et } \epsilon \in L_1 \end{cases}$$

$$(b) \quad L_1 \oplus_1 L_2 \subseteq L_1 \cup L_2$$

On peut supposer avoir enlevé les non-terminal improductif, de sorte que $\text{PREMIER}(X) \neq \emptyset \quad \forall X \in V_N$, et donc on ne tombe jamais dans le cas 1.

- On voit donc que c'est ϵ qui joue un rôle important; on définit donc la fonction $\text{eps}(X) = T$ si X peut produire ϵ .

Cette fonction est définie par le système d'équations récursif suivant:

$$\text{eps}(X) = \bigvee_{\{p \mid p[0] = X\}} \bigwedge_{i=1}^{n_p} \text{eps}(X_i) \quad \text{si } p = X_0 \rightarrow X_1 \dots X_{n_p}$$

On peut donc calculer eps de manière brutale: on fait du tableau eps ~~à 1~~ pour tout X et on itère les équations ci-dessus tant qu'on trouve une case change. Le nombre d'itération est donc borné par $|V_N|$ et chaque itération se fait en temps $O(|R|)$.

- Maintenant qu'on a calculé eps, on peut redéfinir PREMIER à laquelle on ajoute les ϵ via des systèmes plus simples :

$$\epsilon\text{-PREMIER}(X) = \bigcup \{ \epsilon\text{-PREMIER}(Y) \mid X \rightarrow \alpha Y \beta \in R \text{ et } \alpha \xrightarrow{*} \epsilon \}$$

- On ramène donc le problème au calcul de fonction défini par des relations de la forme :

$$(*) \quad f(a) = g(a) \cup \{ f(b) \mid a R b \} \quad \text{où } R \text{ est une relation sur } A \times A \text{ et } g \text{ une fonction connue de } A \rightarrow \mathcal{P}(E)$$

* Pour PREMIER : $X R_{PR} Y \Leftrightarrow (\exists (X \rightarrow \alpha Y \beta) \in P : \epsilon \xrightarrow{*} \alpha)$

$$g_{PR}(X) = \{ a \mid (X \rightarrow \alpha a \beta) \in P \text{ et } \epsilon \xrightarrow{*} \alpha, a \in V \}$$

* Pour SUivant : $Y R_{SU} X \Leftrightarrow (\exists (X \rightarrow \alpha Y \beta) \in P : \beta \xrightarrow{*} \epsilon)$

$$g_{SU}(Y) = \bigcup \{ \epsilon\text{-PREMIER}(X) \mid X \rightarrow \alpha Y \beta \in P \text{ et } \beta \xrightarrow{*} \epsilon \}$$

- Regardons maintenant comment résoudre les problèmes de type (*):

* Soit $G_R = (A, E)$ le graphe induit par R , i.e. $a \rightarrow b$ est un arc de E ssi $a R b$. Alors un parcours bien choisi de G_R permet de calculer f , si G_R est sans cycle.

* Que se passe-t-il si G_R contient un cycle ? Considérons $a_1, \dots, a_n$ une suite telle que $a_i R a_{i+1}$ et $a_n R a_1$, ainsi la relation est donnée :

$$f(a_1) \supseteq f(a_2) \supseteq \dots \supseteq f(a_n) \supseteq f(a_1)$$

i.e. on a $f(a_1) = f(a_2) = \dots = f(a_n)$

Ainsi, f est constant sur les composantes fortement connexes

* L'algorithme commence donc par calculer les CFC du graphe G_R et remplace chaque composante C par un seul sommet s_C tq $g(s_C) = \bigcup_{s \in C} g(s)$

Puis on calcule un ordre topologique du graphe G_R' obtenu et en un passage on calcule f sur tous les sommets