Une fois la relation $\rhd$ donnée, la relation $\hookrightarrow$ se définit alors ainsi à partir de la fermeture réflexive-transitive $\rhd^*$ de la relation $\rhd$

$$\texttt{t} \hookrightarrow \texttt{s} \text{ si et seulement si } \texttt{t} \rhd^* \texttt{s} \text{ et } \texttt{s} \text{ est irréductible}$$

Le fait que le terme `s` soit irréductible signifie qu'il n'y a plus rien à calculer dans `s`, par exemple le terme `20` est irréductible, mais pas le terme `16 + 4`. Cela peut se définir par le fait qu'il n'existe pas de terme `s'` tel que `s` $\rhd$ `s'`.

1.3.4 La non-terminaison

L'exécution d'un programme peut donner un résultat, produire une erreur ou ne pas terminer. Les erreurs peuvent être considérées comme des résultats particuliers. Il y a, en revanche, plusieurs manières d'exprimer la sémantique des programmes qui ne terminent pas toujours. La première est de considérer que si le terme `t` ne termine pas, alors il n'y a pas de couple de la forme `(t,s)` dans la relation $\hookrightarrow$. La seconde est d'ajouter un élément particulier $\perp$ aux valeurs de sorties, et de décider que la relation $\hookrightarrow$ contient le couple `(t,`$\perp$`)` quand le terme `t` ne termine pas.

Cette alternative peut paraître superficielle : il est très facile, quand on a une relation, d'enlever tous les couples de la forme `(t,`$\perp$`)`, ou au contraire d'ajouter un tel couple quand il n'y a pas de couple de la forme `(t,s)` dans la relation. Ceux qui ont suivi un cours de calculabilité remarqueront cependant que, quand on ajoute les couples `(t,`$\perp$`)`, la relation $\hookrightarrow$ cesse d'être récursivement énumérable.