**Expérience de développement**

Après avoir regardé rapidement les tests unitaires, nous avons déterminé que nous allions suivre l’ordre des tests pour construire l’évaluateur Ouf! La première étape était donc les entrées *define* pour stocker des variables et des fonctions dans l’environnement. Pour la traduction du langage Ouf! vers une forme évaluable, nous avons ajouté trois cas dans la fonction specialForm2Exp. Si *define* n’est pas suivi d’argument, la fonction renvoie une erreur. Si la fonction reçoie un nom de variable et un nombre, on transforme la variable en symbole et le nombre en expression évaluable. Enfin, si la fonction reçoit en paramètre une autre fonction, tel qu’une fonction lambda par exemple, un appel récursif se passe jusqu’à ce que la dernière fonction soit traduite pour l’évaluation. Pour la partie d’évaluation des fonctions *define*, nous utilisons tout simplement la fonction insertVar pour ajouter la variable à l’environnement.

La deuxième étape de l’évaluateur consiste à lire des entrées de primitives (+, -, \*) en préfixe et de les évaluer. Pour la traduction, nous utilisons la fonction sexp2Exp qui vérifie que le nombre d’arguments est suffisant.