Règles de typage pour miniJava

1 Notations

Dans ce qui suit, P désignera un programme miniJava et C une classe. On supposera que
- classe(C,P) est vrai si la classe C est définie dans le programme P et faux sinon.
- sous_classe(C,D,P) est vrai si C hérite de D dans P et faux sinon.
- sous_classe_immediate(C,D,P) est vrai si C hérite directement de D dans P et faux sinon.
- assignable(e) e vrai si l'expression e dénote un emplacement mémoire et faux sinon.

Les types de miniJava considérés seront :
- void
- boolean
- int
- Inst(C) (le type correspondant à une classe C)

2 La relation de sous-classe

Le fait que la classe C est une sous-classe (au sens large) de la classe D dans le programme P s'écrira

$$\mathrm{P} \vdash \mathrm{C} \prec \mathrm{D}$$

3 La relation de sous-type

Le fait que le type t_1 est un sous-type du type t_2 dans le programme P s'écrira

$$\mathrm{P} \vdash t_1 \leq t_2$$

Cette relation est définie par les règles suivantes :

$$\frac{}{\mathrm{P} \vdash \mathrm{t} \leq \mathrm{t}}$$

$$\frac{\mathrm{P} \vdash \mathrm{C} \prec \mathrm{D}}{\mathrm{P} \vdash \mathrm{Inst\ C} \leq \mathrm{Inst\ D}}$$

4 Typage des expressions miniJava

On supposera que FEnv(C,P) désigne l'environnement de types des champs définis dans la classe C du programme P. Ces champs peuvent être définis dans la classe C elle-même ou bien dans l'une de ses superclasses. FEnv(C,P) est un environnement semblable à ceux qui ont été utilisés pour le typage de C et qui seront utilisés à nouveau ici pour les variables locales (paramètres des méthodes ou variables déclarées dans un bloc). Toutefois, l'environnement FEnv(C,P) fournira deux informations : le type du champ et aussi son statut (champ d'instance ou champ statique). Ceci permettra de vérifier qu'un champ d'instance n'est pas utilisé dans une méthode statique. Par contre, on n'utilisera pas ici la protection. Tous les champs seront traités ici comme des champs publics.

De façon similaire, on désignera par MEnv(C,P) un environnement associant à chaque nom de méthode m utilisable dans la classe C, une liste de triplets comportant une **signature**, un type résultat et un statut pour cette méthode. La signature d'une méthode est la liste des types de ses arguments.

Le calcul de MEnv(C,P), qui dans un vérificateur de type pour miniJava sera fait une seule fois pour chaque classe, sera l'occasion de vérifier que les définitions de méthodes rencontrées satisfont bien les contraintes imposés par miniJava, c'est-à-dire par notamment :
- une méthode d'instance ne doit pas masquer une méthode statique ayant les mêmes types d'arguments
- une méthode statique ne doit pas masquer une méthode d'instance ayant les mêmes types d'arguments
- Lorsqu'une méthode en masque une autre, le type du résultat doit être le même dans les deux méthodes.

Le typage d'une expression sera exprimé par une formule

$$\mathrm{P,C,E,s} \vdash \mathrm{e} : \ \mathrm{t}$$

où
- P est le programme considéré
- C est la classe considérée dans le programme P
- E est l'environnement donnant le types des variables locales susceptibles d'être utilisées dans l'expression e
- s est le statut de la méthode dans laquelle apparaît l'expression e. Cette information est nécessaire pour savoir si on a le droit d'utiliser this et super et quel sens on doit leur attribuer ainsi que pour vérifier la correction des accès aux champs.

$$\frac{}{\mathrm{P,C,E,inst} \vdash \mathrm{this} : \ \mathrm{Inst\ C}} \ (\mathrm{This})$$

$$\frac{\text{C extends D}}{\text{P,C,E,nonstatic} \vdash \text{super} : \text{Inst D}} \ (\text{Super})$$

$$\frac{\text{E(x)=t}}{\text{P,C,E,s} \vdash \text{x} : \text{t}} \ (\text{VarLocale})$$

$$\frac{\text{E(x) non défini} \qquad \text{t,nonstatic=FEnv(P,C)(x)}}{\text{P,C,E,nonstatic} \vdash \text{x} : \text{t}} \ (\text{Champ1})$$

$$\frac{\text{E(x) non défini} \qquad \text{t,static= FEnv(P,C)(x)}}{\text{P,C,E,s} \vdash \text{x} : \text{t}} \ (\text{Champ2})$$

$$\frac{}{\text{P,C,E,s} \vdash \text{new D()} : \text{Inst D}} \ (\text{New})$$

$$\frac{\text{P,C,E,s} \vdash \text{e} : \text{Inst C'} \qquad \text{t,\_=FEnv(P,C')(x)}}{\text{P,C,E,s} \vdash \text{e.x} : \text{t}} \ (\text{AccesChamp})$$

$$\frac{\begin{array}{c}\text{P,C,E,s} \vdash e_1 : t_1 \qquad \text{P,C,E,s} \vdash e_2 : t_2 \\ \text{assignable(e}_1\text{)} \qquad \text{P} \vdash t_2 \leq t_1\end{array}}{\text{P,C,E,s} \vdash e_1 = e_2 : t_1} \ (\text{Affectation})$$

$$\frac{\begin{array}{c}\text{P,C,E,s} \vdash \text{e} : \text{Inst D} \qquad (t_1, ..., t_n)\text{,t,\_=MEnv(P,D)(m)} \\ \forall i \ \ \text{P,C,E,s} \vdash e_i : t'_i \qquad \forall i \ \ \text{P} \vdash t'_i \leq t_i\end{array}}{\text{P,C,E,s} \vdash \text{e.m}(e_1, ..., e_n) \ : \ \text{t}} \ (\text{AppelMethode1})$$

$$\frac{\begin{array}{c}(t_1, ..., t_n)\text{,t,static=MEnv(P,D)(m)} \\ \text{P,C,E,s} \vdash e_i : t'_i \qquad \forall i \ \ \text{P} \vdash t'_i \leq t_i\end{array}}{\text{P,C,E,s} \vdash \text{D.m}(e_1, ..., e_n) \ : \ \text{t}} \ (\text{AppelMethode2})$$

5 Typage des instructions miniJava

$$\frac{\text{P,C,E,s} \vdash \text{decls : E'} \qquad \text{P,C,E',s} \vdash \text{stats}}{\text{P,C,E,s} \vdash \{\text{decls stats }\}} \text{ (Bloc)}$$

$$\frac{\text{P,C,E,s} \vdash \text{e : t}}{\text{P,C,E,s} \vdash \text{(e ;)}} \text{ (Comput)}$$

$$\frac{\text{P,C,E,s} \vdash \text{e : bool} \qquad \text{P,C,E,s} \vdash \text{stat}_1 \qquad \text{P,C,E,s} \vdash \text{stat}_2}{\text{P,C,E,s} \vdash \text{if (e) stat}_1 \text{ else stat}_2} \text{ (If)}$$

$$\frac{\text{P,C,E,s} \vdash \text{e}_1 : \text{t}_1 \qquad \text{P,C,E,s} \vdash \text{e}_2 : \text{boolean} \qquad \text{P,C,E,s} \vdash \text{e}_3 : \text{t}_3 \qquad \text{P,C,E,s} \vdash \text{stat}}{\text{P,C,E,s} \vdash \text{for (e}_1\text{ ;e}_2\text{ ;e}_3\text{) stat}} \text{ (For)}$$

$$\frac{\text{E(return) = void}}{\text{P,C,E,s} \vdash \text{return}} \text{ (Return0)}$$

$$\frac{\text{E(return) = t} \qquad \text{P,C,E,inst} \vdash \text{e : t'} \qquad \text{P} \vdash \text{t'} \leq \text{t}}{\text{P,C,E,s} \vdash \text{return(e)}} \text{ (Return)}$$

$$\frac{}{\text{P,C,E,s} \vdash \text{Nop}} \text{ (Nop)}$$

$$\frac{\text{P,C,E,s} \vdash \text{stat} \qquad \text{P,C,E,s} \vdash \text{stats}}{\text{P,C,E,s} \vdash \text{stat stats}} \text{ (Seq)} \qquad \frac{}{\text{P,C,E,s} \vdash \text{/* vide */}} \text{ (Seq0)}$$

6 Typage des déclarations de variables locales

Le typage des déclarations de variables locales augmente l'environnement de type des variables locales.

$$\frac{}{\text{P,C,E,s} \vdash \text{t x : [x :t]} \oplus \text{E}} \text{ (SimpleDeclVar)}$$

$$\frac{\text{P,C,E,s} \vdash e:\ t_1 \qquad \text{P} \vdash t_1 \leq t}{\text{P,C,E,s} \vdash \text{t x = e} \ :\ \text{[x :t]} \oplus \text{E}} \text{(InitDeclvar)}$$

$$\frac{\text{P,C,E,s} \vdash \text{decl}:\ \text{E'} \qquad \text{P,C,E',s} \vdash \text{decls}:\ \text{E''}}{\text{P,C,E',s} \vdash \text{decl decls}:\ \text{E''}} \text{(SeqDecl)}$$

$$\frac{}{\text{P,C,E,s} \vdash \text{/*vide*/}:\ \text{E}} \text{(Decl0)}$$

7 Typage des déclarations de champs et de méthodes

Le typage des déclarations de champs obéit aux mêmes règles que celui des déclarations de variables locales à ceci près qu'elles n'enrichissent pas l'environnement local mais l'environnement des champs fournit par FEnv. On peut noter cependant que le résultat de FEnv ne dépend pas du typage des initialisations. On peut donc implémenter FEnv indépendamment de la vérification de types puis utiliser ensuite la fonction de typage des expressions pour vérifier que les initialisations sont bien correctes. Il en sera de même pour MEnv et les déclarations de méthodes.
Par conséquent, le typage des déclarations de méthodes se présente comme une simple vérification de correction et sera exprimé par des formules de la forme

$$\text{P,C} \vdash \text{mdecl}$$

Voici les règles définissant la correction des déclarations de méthodes :

$$\frac{\text{P,C,}[x_1 : t_1; ...x_n : t_n; return : t]\text{,inst} \vdash \text{bloc}}{\text{P,C} \vdash \text{t m}(t_1 x_1, ..., t_n x_n) \text{ bloc}} \text{(MethDecl1)}$$

$$\frac{\text{P,C,}[x_1 : t_1; ...x_n : t_n; return : t]\text{,static} \vdash \text{bloc}}{\text{P,C} \vdash \text{static t m}(t_1 x_1, ..., t_n x_n) \text{ bloc}} \text{(MethDecl2)}$$