TD-TP 17 Testabilité (Polynôme)

Thème : on reprend le thème « type T_Polynome » mais il s'agit cette fois de tester un paquetage (implémentation inconnue) en utilisant les techniques vues au cours sur la **testabilité**. Les concepts vus dans les deux TD-TP 13A et 13B seront utiles.

Concepts (de cette dernière séance) **:** test fonctionnel (dit boîte noire), cas de test, environnement de test, driver de test, plan de validation, partition, limite, rapport de test. Fichier entrée, fichier sortie, fichier Oracle (pas de fichier d'environnement).

Introduction : on se propose de tester suivant une stratégie de **boîte noire** un paquetage inconnu sur les polynômes. Bien sûr il n'est pas intéressant de tester celui que vous avez mis au point (`P_Poly_V1_G.V2`) dans les séances précédentes car il est *forcément génial* donc juste ! Non, ici, la réalisation du paquetage `P_Poly_V1_G.V3` vous est inconnu (fichier `p_poly_v1_g-v3.adb`). Vous n'aurez pas le droit d'accéder aux sources (sinon ce serait du test **boîte claire**) le paquetage inconnu pourrait être disponible seulement compilé. Mais pour des raisons techniques cela ne sera pas possible aussi vous devrez faire l'effort de ne pas chercher à le regarder (contrat moral !).

Par contre les spécifications du paquetage `P_Poly_V1_G.V3` (fichier `p_poly_v1_g-v3.ads`) sont accessibles et visibles. En effet, c'est justement **ce contrat qu'il faut valider**.

Déroulement : le paquetage fils `P_Poly_V1_G.V3` est générique. Il comprend (comme `P_Poly_V1_G.V2`) 14 sous programmes :

"+"
"-" (unaire)
"-" (binaire)
"*"
"/"
"rem"
Derivee
Valeur
Get (sur fichier)
Put
Set
Get_Degre
Get_Coef
Set_Coef

Sous programmes qu'il faut (faudrait) tester (d'abord séparément).
Nous limiterons notre travail à quelques uns.

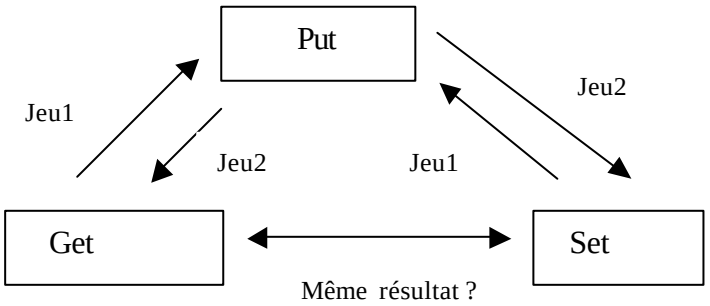
Pour **chaque sous programme** (revoyez le cours sur la testabilité) il faut préparer un **plan de test** (qui va tenter de couvrir le plus de cas possibles). Chaque plan induit un **fichier de donnée**, un **fichier de sortie** et un **fichier oracle** (résultat attendu). Pour piloter les deux premiers fichiers il faut écrire un **driver** pour ouvrir le fichier de donnée, exécuter le sous programme à tester sur les données et écrire le résultat sur le fichier de sortie, enfin il faut comparer le fichier « résultat » et le fichier oracle et établir un **rapport** de test.

En résumé : plan de test ⇒ fichier de donnée ⇒ écriture du driver ⇒ lancement du driver ⇒ comparaison ⇒ rapport

Détails : le premier travail doit commencer par le test des constructeurs (ici Get et Set). On comprend bien qu'il faudra ensuite partir de polynômes corrects syntaxiquement ! (C'est à dire correctement construits). Mais comment tester les constructeurs sans vérifier leur validité avec des sondes (c'est-à-dire en clair des opérations « accesseurs » permettant de savoir ce qu'il y a dans le polynôme) ici soit (Put ou à la rigueur Get_Degre et Get_Coef). On voit ainsi que **certains tests ne sont pas indépendants**. On peut encore tester Get et Set sur le presque même jeu de données et comparer les polynômes obtenus (puisque l'égalité existe sur ce type). Mais cela ne nous dispense pas de tester Put avec ces deux constructeurs. Ensuite on peut tester un à un les autres sous programmes.

Soit par exemple la méthode suivante (commentée rapidement en TD) :

Tester le couple Get/Put avec un jeu de données noté JEU_1
Tester le couple Set/Put avec un jeu de données noté JEU_2
Tester le couple Get/Put avec un jeu de données noté JEU_2
Tester le couple Set/Put avec un jeu de données noté JEU_1
On réutilise les mêmes données ! C'est intéressant !
On peut aussi tester Set/Get avec les mêmes deux jeux de données.



Si tout est conforme les sous programmes Get, Set et Put sont réputés bons !

Ensuite pour chaque sous programme à tester on cherche un jeu de données (plan de test) significatif pour le sous programme. Voir si les jeux de données précédents ne peuvent (en totalité ou en partie) être utilisés et complétés.

Prenons par exemple le test de Get_Coef qui donne le coefficient d'un degré donné.
Quelques cas intéressants pour tenter de « couvrir » le plus de cas possibles :

- Le degré associé au coefficient est invalide (hors de 0..Max).
- Le degré associé au coefficient est invalide (hors de 0..Degre).
- Le coefficient du degré extrême Degre d'un polynôme (constant, de degré 1, de degré impair, de degré pair, de degré Max, etc.).
- Le coefficient de degré 0 d'un polynôme (constant, de degré 1, de degré impair, de degré pair, de degré Max, etc.).
- Le coefficient de degré intermédiaire (pair puis impair) donc entre 0..Degre (bornes exclues) d'un polynôme (de degré pair, de degré impair, de degré Max, etc.).
- Autres ?

On en déduit un plan de test (voir page 4 et aussi feuilles vierges distribuées en TP) à **remplir** ensemble : fichier entrée (une ligne = une liste de coefficients) et fichier oracle.
Puis on écrit le **driver de test** dont l'algorithme est schématiquement celui-ci :

Ouvrir les fichiers (entrée et résultat),
Lire une « donnée » , on initialise le polynôme avec Get (Get est supposé correct !), on met en œuvre le sous programme (ici Get_Coef), on écrit le résultat On répète jusqu'à fin de fichier. On ferme les fichiers. (voir cours Ada n°11 !).

On compare fichier résultat et oracle. On rédige le rapport de test : pas d'erreur découverte ou alors liste de problèmes rencontrés.

Travail à réaliser : on testera en priorité les 4 procédures Get_Degre, Valeur, Get_Coef et Derivee. Pour chacun des 4 tests il faut écrire un driver de test. Tous les drivers seront réalisés dans un paquetage appelé Dri_Poly. Vous rendrez **pour chaque test** : plan de test et rapport voir modèle page suivante (feuille distribuée à compléter) et listing du driver.

- Créer un répertoire tp17. Copiez les fichiers.
- Renommez (mv) p_poly_v1_g-v3.adb.squ (ne listez pas ce body sinon c'est du boîte claire).
- Renommez (mv) dri_poly.adb.squ en dri_poly.adb
- voir et lire p_poly_v1_g-v3.ads (spécifications connues ! guide pour boîte noire)
- Complétez au fur et à mesure les quatre drivers manquants (dans **dri_poly.adb**) Inspirez vous de ceux qui sont déjà réalisés. Compilez : **gnatgcc –c dri_poly.adb –I../tp13A –I../tp11**
- Préparez les fichiers de tests à partir de vos plans de tests manuscrits (cf. page suivante).
- Préparez un lanceur (un par driver à exécuter). Par exemple pour **Get/Put** le lanceur **ts_get_put.adb** vous est donné (consultez le).
- Faire l'édition de liens : **gnatmake ts_get_put (-I../ ? à voir !)**
- Exécuter le lanceur du driver **./ts_get_put > fichier.out**. Cet exemple (ainsi que ts_set_put) n'est pas à réaliser en TP c'est un modèle à reproduire pour les autres tests.
- Faites les éditions de liens pour chacun des 4 programmes. (fichiers **ts_get_coef.adb, ts_get_degre.adb**, **ts_valeur.adb** et **ts_derivee.adb**).
- Etablir et rendre les 4 feuilles (4 rapports de test) avec votre nom et votre groupe.

Plan, fichier et rapport de test

Nom du programmeur :

Nom du paquetage :`P_Poly_V1_G.V3`

Nom du sous-programme testé :

Plan de test :

N°	Cas de test	remarques	Oracle
1			
2			
3			
4			
5			
6			
7			
8			
9			

Fichier de test : (concrétisation des cas de test ci dessus)

Rapport de test :