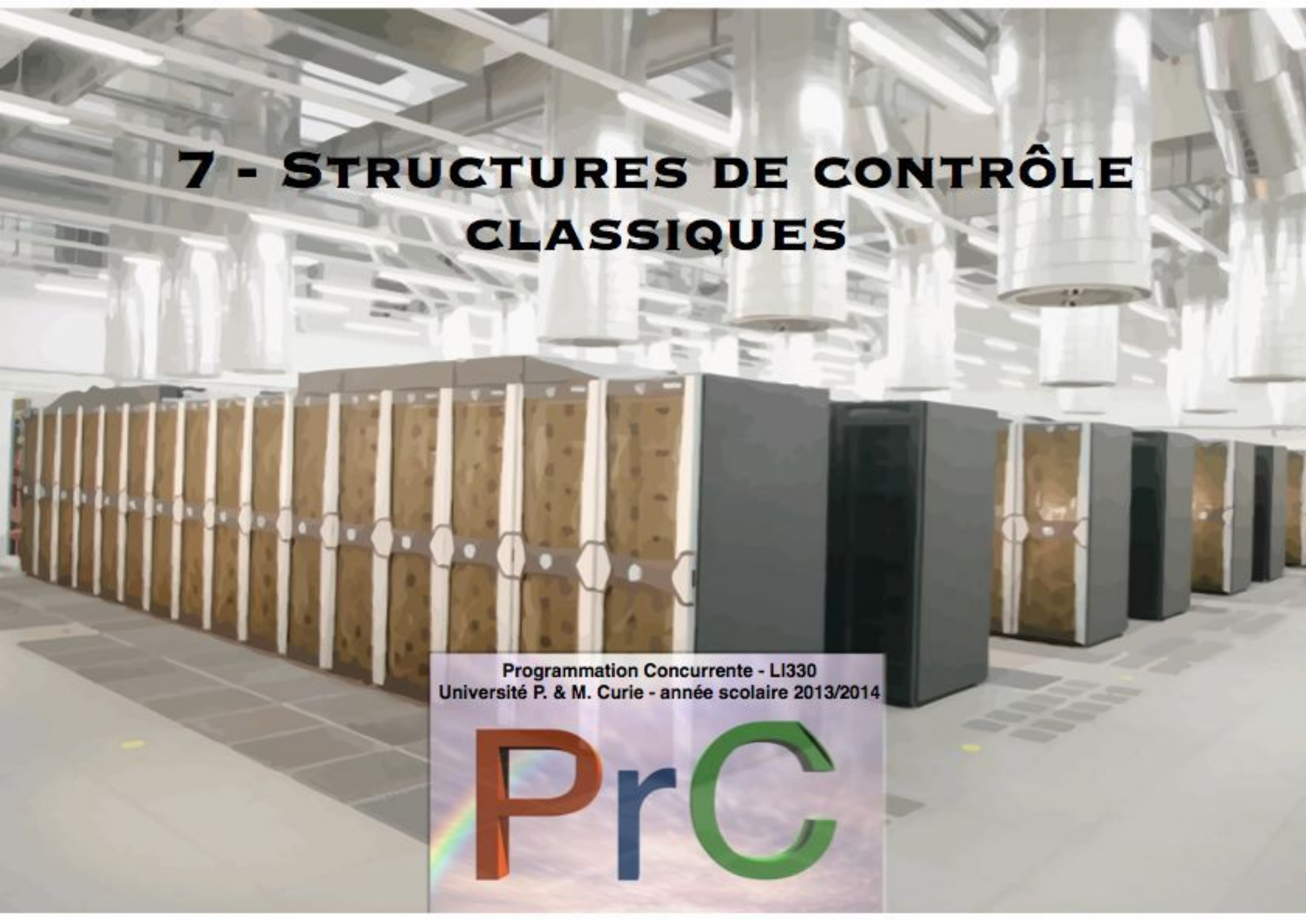
7 - Structures de contrôle classiques



Programmation Concurrente - LI330
Université P. & M. Curie - année scolaire 2013/2014

PrC

Conditions

- Sémantique: exécution d'une unique séquence parmi celles qui sont gardées

```
if N mod 4 /= 0 then
   Non_Bissextile;
elsif N mod 100 /= 0 then
   Bissextile;
elsif N mod 400 = 0 then
   Bissextile;
else
   Non_Bissextile;
end if;
```

Énumération de cas

Sémantique: Exécution d'une séquence sélectionnée en fonction de la valeur d'une expression

```
-- On suppose que S est de type «character»
case S is
   when 'G' | 'M' =>
      Est_Garcon;
   when 'F' =>
      Est_Fille;
   when others => -- car le type «character» possède d'autres valeurs
      Erreur;
end case;
```

La BNF du «case»

Backus-Naur Form

```
expr_case  case expression is
               when valeurs => liste_d'instructions
               {when valeurs => liste_d'instructions}
               [when others => liste_d'instructions]
           end case;

valeurs    littéral du type considéré (valeur immédiate)          |
           littéral1 .. littéral2     (intervalle discret)        |
           identificateur d'un sous-type                          |
           valeurs | valeurs
```

Permet de regrouper des intervalles de valeurs (union mathématique)

Sémantique: l'instruction de test est explicitement indiquée das le corps de la boucle

```
-- Initialisations nécessaires
Proposition := 0;
Essai       := 1;
-- boucler jusqu'à ce que l'on ait trouvé le juste prix
-- ou que le nombre maximum d'essais soit atteint
loop
   Put ("votre proposition ? ");
   Get (Proposition);
   Essai := Essai + 1;
   exit when Essai = Max_Essai or Proposition = Prix;
end loop;
```

BNF des boucles itératives

Backus-Naur Form

Possibilité de nommer la boucle

```
expr_loop: [identificateur:]
           loop
                 liste_d'instructions
                  [sortie_boucle
                 liste_d'instructions]
           end loop [identificateur];

sortie_boucle:   exit [identificateur] [ when expression booléenne] ;
```

Sortie dès que la condition est vérifiée

Boucles while

- Sémantique: tant que la condition est vérifiée, continuer à exécuter le corps de la boucle

```
Somme := 0; -- Initialisation de la somme
Ma_boucle:
while V <= Max loop
   Somme := Somme + V; -- Cumul
   V := V + 1; -- Incrémentation du compteur
end loop ma_boucle;
```

BNF des boucles while

Backus-Naur Form

```
expr_while:     [identificateur:]
          while expression_booléenne loop
             liste_d'instructions
          end loop [identificateur];
```

Exemple de boucle énumérative

Sémantique: pour toutes les valeurs de l'intervalle, appliquer le corps de la boucle avec une valeur affectée à l'indice

```
-- Boucle sur les lignes
for I in 1 .. 5 loop
   -- Boucle sur les colonnes
   for J in 1.. I loop
      Put ('*');
   end loop;
   New_Line;
end loop;
```

- Lors de la boucle sur les colonnes, la valeur de I est fixée
- Les variables I et J ont une portée limitée
- Les variables I et J ne sont pas à déclarer

BNF DES BOUCLE ÉNUMÉRATIVE

Backus-Naur Form

```
expr_for: [identificateur:]
          for variable in [reverse] intervalle loop
              liste_d'instructions
          end loop [identificateur];
```

La variable d'indice est déclarée implicitement comme étant du type défini par l'intervalle.
C'est une variable en «lecture seule»