DS 2012 Parallélisme

Damien Crémilleux, Lauriane Holy, Boris Labbé

25 mars 2014

1 Problème 1 : Le salon de coiffure et les moniteurs de Hoare

1.1 Décrire les processus coiffeur et client

```
Class Client {
  SalonCoiffure.se_faire_coiffer();
}

Class Coiffeur {
  while(true) {
    SalonCoiffure.client_suivant();
    SalonCoiffure.terminer_coupe();
  }
}
```

1.2 Donner le texte du moniteur

```
Class SalonCoiffure implements HOARE_Monitor { //Non
   java - les methodes s'executent toutes en exclusion
   mutuelle

  private condition coiffeur_disponible; //declaration
     de la file (fauteuil) pour le coiffeur
  private condition fauteil_occupe; //declaration de la
      file (fauteuil) le client
  private condition porte_ouverte; //declaration de la
     file (ouverture de la porte) pour le client
  private condition client_sorti; //declaration de la
     file (attente que le client sorte) pour le client

  public se_faire_coiffer() {
    coiffeur_disponible.reprendre();
    fauteuil_occupe.attendre();
    porte_ouverte.attendre();
    client_sortie.attendre();
  }
```

```
    public client_suivant() {
      coiffeur_disponible.attendre();
    }

    public terminer_coupe() {
      //coiffure du client
      fauteuil_occupe.reprendre();
      porte_ouverte.reprendre();
      client_sortie.reprendre();
    }
  }
```

2 Problème 2 : Simulation de feux tricolores et CSP

2.1 Automate élémentaire

```
  *[true --> [etat = vert --> delay(d1); etat = orange; feu(orange)
              |
              etat = orange --> delay(d2); etat = rouge; feu(rouge)
              |
              etat = rouge --> delay(d3); etat = vert; feu(vert)]
   ]
```

2.2 Synchronisation 2 feux

```
  automate(i) : 0..1
  int go;
  [i = 0 --> etat = vert; feu(vert);
   |
   i = 1 --> etat = rouge; feu(rouge);
  ]
*[true --> [etat = vert --> delay(d1); etat = orange; feu(orange)
            |
            etat = orange --> delay(d2); etat = rouge; feu(rouge);
                automate((i+1)%2)!go
            |
            etat = rouge;automate((i+1)%2)?go--> delay(d3); etat =
                vert; feu(vert)]
  ]
```

2.3 Synchronisation des véhicules

```
  automate(i) : 0..1
  int go;
  int j;
  [i = 0 --> etat = vert; feu(vert);
   |
   i = 1 --> etat = rouge; feu(rouge);
```

```
  ]
*[true --> [etat = vert --> delay(d1); etat = orange; feu(orange)
            |
            etat = orange --> delay(d2); etat = rouge; feu(rouge);
                automate((i+1)%2)!go
            |
            etat = rouge;automate((i+1)%2)?go--> delay(d3); etat =
                vert; feu(vert); (j:1..nb_voiture)voiture(j)!i] //
                on envoie le numero du feu qui passe au vert a
                toutes les voitures
 ]

 borismobile(j) : 1..nb_voiture
  int feu_attente;
  int i;
  int k;
  *[(k:0..1)automate(k)?i  --> [i = feu_attente --> passe //on
      regarde que le message envoye provient du bon
                                |
                                i != feu_attente --> stop]
   ]
```