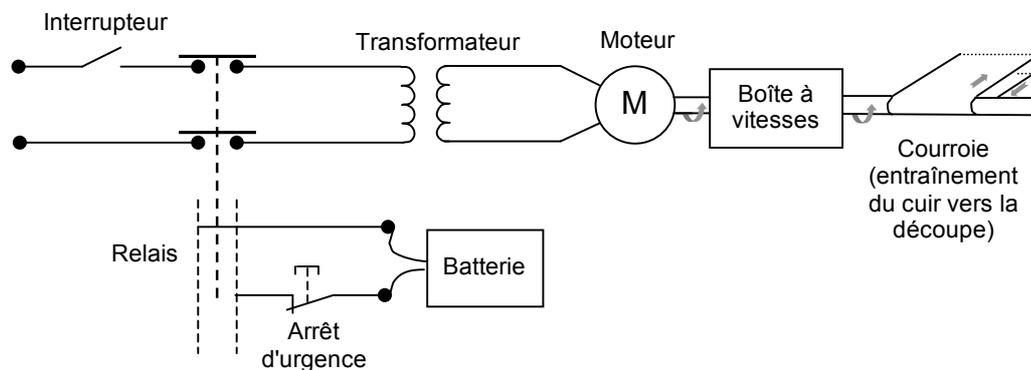




M1Info/BIA/TP1/2007
TP5/6 PROLOG
(PROgrammation LOGique)
Compte-Rendu à rendre avant le 21 décembre.

Avertissement : Ce TP est noté. Vous devez rendre un CR correspondant au plan proposé avant le 21 décembre (le tout sous format électronique : PDF pour le CR et texte pour les sources commentés).

Atelier de découpe de cuir



Objectif du TP :

Modéliser un système de diagnostic pour la résolution des problèmes d'une ligne de découpe de cuir.

Déroulement actuel de la résolution de problème :

A l'atelier de découpe de cuir, on appelle l'expert quand un problème survient (l'ouvrier veut mettre en route son installation mais cela ne fonctionne pas : la courroie ne tourne pas). Il faut réparer le plus vite possible.

- Chaque composant de l'installation a un rôle bien défini : la courroie, entraînée par le moteur, lui-même relayé par la boîte à vitesses interviennent dans la fonction de découpe du cuir. L'interrupteur, le secteur et le transformateur servent à l'alimentation en énergie. Le bouton d'arrêt d'urgence, le relais et la batterie assurent la sécurité.

- On peut faire un certain nombre d'observations pour savoir si un composant est en défaut ou non. Une fonction peut ne pas être assurée pour deux raisons : soit parce que l'un de ses composants est défaillant, soit parce qu'elle dépend d'une autre fonction qui est défaillante. Par exemple, on ne peut pas avoir de découpe s'il n'y a pas d'alimentation en énergie, et la sécurité peut être amenée à interrompre l'alimentation si nécessaire.

- Quelquefois, c'est tout simplement la batterie qui n'alimente pas le relais : soit elle doit être changée (la tension aux bornes dans ce cas est nulle), soit elle est déchargée (la tension n'est pas nulle mais ne fait pas les 24 Volts nécessaires) et il suffit de la recharger. Quand le relais n'est pas alimenté, alors le transformateur ne l'est pas et le moteur non plus évidemment.

- Dans le cas où le secteur fonctionne et qu'on n'a pas procédé à un arrêt d'urgence, si la batterie est bonne et qu'il n'y a pas de courant au primaire du transformateur, c'est que le relais est défectueux.

- Il se peut aussi que le moteur soit grillé. Là, ce n'est pas difficile à savoir : le moteur ne tourne pas et il y a de la tension au secondaire du transformateur. En revanche, s'il n'y en a pas au secondaire du transformateur mais qu'il y en a au primaire, c'est que le transformateur est défectueux. Il faut le changer. Si le moteur tourne, et que la courroie ne tourne pas, c'est que la vitesse n'est pas bien passée. Il faut ouvrir le carter de la boîte et débloquer la tringlerie à la main.

Les dirigeants de la société de découpe de cuir, désirant alléger le travail de leurs experts, et surtout leur éviter des déplacements trop fréquents en atelier pour des incidents classiques, ont décidé de mettre en place un système d'aide au diagnostic et à la résolution des problèmes de l'atelier afin de permettre à l'ouvrier de se dépanner sans faire intervenir l'expert. L'ouvrier rencontrant un problème n'aura alors qu'à soumettre le symptôme détecté pour que le système identifie le problème et lui propose une solution adaptée, ainsi que la catégorie du problème (découpe, énergie, sécurité). Dans le cas où le symptôme fourni par l'ouvrier n'est pas suffisant pour que le système formule un diagnostic, le système interroge l'ouvrier de manière à compléter les observations et pouvoir fournir une réponse.

Systeme

1. Définir la base de règles sous forme de listes composées de listes de prémisses et de listes de conclusions. (4 points)
2. Définir un prédicat permettant à l'utilisateur d'initialiser la base de faits. On utilisera le prédicat `asserta` pour ajouter `vrai(Fait)` pour les faits positifs et `faux(Fait)` pour les faits négatifs. (2 points)
3. Définir un prédicat `resoudre` qui sature la base de règles et produit une trace de son fonctionnement. L'algorithme utilisé pour ce moteur sera le suivant (8 points) :

```
Changement <-- Vrai
Tant que Changement est Vrai
  Changement <-- Faux
  Boucle sur les règles : soit R une règle de BaseRègles
    Si R n'est pas marquée et si les prémisses de R
      appartiennent à BaseFaits
      Alors ajouter les conclusions de R à BaseFaits
      changement <-- Vrai
      marquer R
  FinSi
FinBoucle
FinTantQue
```

Voir un exemple d'exécution souhaitée page suivante.

Exemple d'exécution souhaitée :

```
1 ?- faits([non(relais_non_alimente),
non(primaire_transformateur_non_alimente),
secondaire_transformateur_non_alimente]).

Yes

2 ?- go.
r10
non(relais_non_alimente) non(primaire_transformateur_non_alimente)
secondaire_transformateur_non_alimente transformateur_mort
r11
non(relais_non_alimente) non(primaire_transformateur_non_alimente)
secondaire_transformateur_non_alimente transformateur_mort moteur_non_alimente
r12
non(relais_non_alimente) non(primaire_transformateur_non_alimente)
secondaire_transformateur_non_alimente transformateur_mort moteur_non_alimente
moteur_arrête
r14
non(relais_non_alimente) non(primaire_transformateur_non_alimente)
secondaire_transformateur_non_alimente transformateur_mort moteur_non_alimente
moteur_arrête courroie_arretee

Yes
```

4. On distingue deux catégories de faits :

- les faits "terminaux", qui ne figurent jamais en partie gauche d'une règle,
- les faits "observables", qui ne figurent jamais en partie droite d'une règle.

a) (2 points) Définir le prédicat `terminal(F)` (respectivement `observable(F)`) vrai si le fait `F` est terminal (resp. observable).

b) (4 points) Si les symptômes fournis par l'ouvrier ne permettent pas de conclure sur la nature de la panne (pas de fait terminal), on veut lui poser des questions afin de préciser la situation et de pouvoir conclure. Pour cela, on procède en deux temps :

- On cherche une règle "presque déclenchable", c'est-à-dire dont toutes les prémisses sont vraies sauf une, qui est inconnue et qui porte sur un fait observable.
- On essaie de déclencher cette règle, en posant une question à l'utilisateur sur la prémisses inconnue. L'utilisateur peut répondre "oui", "non", ou "je ne sais pas". Dès qu'on a une réponse "oui" ou "non" à une de nos questions, on peut ajouter un fait à la base de faits et relancer le moteur. Si l'on n'arrive toujours pas à conclure sur un fait terminal, il faut réitérer le processus. Attention à ne pas poser deux fois la même question à l'ouvrier.

Exemple d'exécution :

```
1 ?- faits([non(relais_non_alimente), secondaire_transformateur_non_alimente]).

Yes

2 ?- go3.
r11
non(relais_non_alimente) secondaire_transformateur_non_alimente moteur_non_alimente

Est-ce que tension_sous_24 ? (o/n/i)
|: i.

Est-ce que tension_nulle ? (o/n/i)
```

|: i.

Est-ce que primaire_transformateur_non_alimente ? (o/n/i)

|: n.

r10

non(relais_non_alimente) non(primaire_transformateur_non_alimente) secondaire_transformateur_non_alimente
moteur_non_alimente transformateur_mort

r12

non(relais_non_alimente) non(primaire_transformateur_non_alimente) secondaire_transformateur_non_alimente
moteur_non_alimente transformateur_mort moteur_arrete

r14

non(relais_non_alimente) non(primaire_transformateur_non_alimente) secondaire_transformateur_non_alimente
moteur_non_alimente transformateur_mort moteur_arrete courroie_arretee

c5

Changer le transformateur

Yes

Question subsidiaire (+ 5 points) :

Dans le cas où le système ne peut résoudre le problème, même après avoir demandé à l'ouvrier de préciser les symptômes, ce dernier fait alors appel à l'expert, qui une fois le problème résolu, rajoute dans le système la nouvelle règle permettant de le résoudre. Vous proposerez donc à la fin du système d'ajouter une règle à celui-ci.

Remerciements :

Ce sujet est très fortement inspiré du cours de Guy Caplat (Département informatique – INSA de Lyon)