Maîtrise Informatique
Génie Logiciel II

Université d'Évry
2004/2005

Travaux dirigés numéro 1

Ceci est la fiche-TD[1] sur CCS et les algèbres de procesus. Tout au long des exercices, vous en profiterez pour découvrir `CWB-NC` (commandes et validation de vos réponses lorsque cela est possible). L'utilisation de cet outil fait partie intégrante des TD.

Exercice 1.1 (Découverte de `CWB-NC`)
Quand vous utilisez `CWB-NC`, pensez à avoir toujours le manuel sous la main.

- Lancez `CWB-NC` avec comme langage CCS (essayez avec et sans interface graphique)
- Au prompt, que se passe-t-il avec les commandes `quit` et `help` ? Et avec `CTRL^C` ?
- Créez un procesus simple `Psimple` dans un fichier nommé `Psimple.css` et chargez le fichier dans `CWB-NC`.
- Simulez votre processus. Calculez son LTS.
- Recherchez les blocages dans `Psimple` (notez que vous devez d'abord sortir de la simulation).

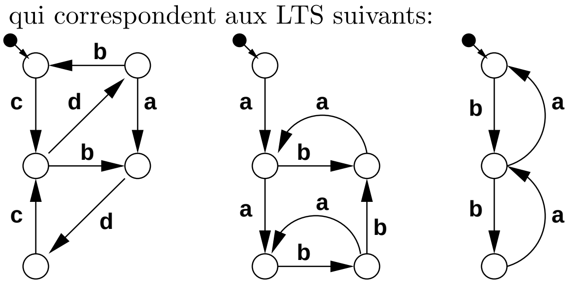
Exercice 1.2 (Syntaxe de CCS)

- Dites si les processus suivants sont syntaxiquement correct ou non (les processus incorrects correspondent dans la plupart des cas à des erreurs fréquemment faites par les utilisateurs inexpérimentés). S'il sont incorrects, corrigez les.

- $a.b.0|\overline{a}.\overline{b}0$
- $(a.(b.0|c.0))$
- $Q = a.Q + a.d.0 + a.d.Q$
- $(a.b.0|\overline{a}.\overline{b}.0)\backslash a, b$
- $B[a\backslash b]$
- $P = a.(b.0 + b.P)$
- $(a.0 + b.0).c.0$
- $(a.0 + \tau.0)\backslash\tau$
- $R = a(b.0 + c.R)$
- $a.(b.0|c.)$

Exercice 1.3 (LTS et processus)

- Donnez des processus CCS qui correspondent aux LTS suivants:



- Expliquez comment vérifier vos réponses avec `CWB-NC`.

Exercice 1.4 (Traces et processus)

[1] C'est-à-dire une fiche qui donne lieu à plusieurs TD.

Donnez, lorsque cela est possible, des exemples de procesus P correspondant aux traces suivantes:

- $\{\epsilon, a, aa, aaa, \ldots\}$
- $\{\epsilon, a, ab, ac, abb, abc, abbb, abbc, \ldots\}$
- $\{abcd, acbd, acdb, cabd, cadb, cdab\} \subset L(P)$
- $\{\epsilon, a, ab, ac\}$
- $\{\epsilon, a, ab, cd\}$
- $\{\epsilon, l, lt, ltt, lttb, lttbt, lttbtt, lttbttb, ...\}$

- Expliquez comment vérifier vos réponses avec `CWB-NC`.

Exercice 1.5 (SOS)
Donnez les LTS correspondant à l'application des règles de SOS de CCS pour les processus suivants :

- $P = 0$
- $P = a.0$
- $P = a.0 + b.0$
- $P = a.0$, $Q = \overline{a}.0$, $R = P|Q$, $S = R\backslash\{a\}$
- $P = a.b.\overline{c}.0$, $Q = P[in/a, out/c]$, $R = Q\backslash\{b\}$, $S = Q\backslash\{in, out\}$
- $P = a.0 + b.P$
- $((a.0 + b.0)|(c.0 + d.0))$
- $((a.E + b.0)|\overline{a}.F)\backslash a$

Exercice 1.6 (Preuves et SOS)
Prouvez :

- $0 \not\rightarrow$
- $a.b.0|a.\overline{b}.0 \xrightarrow{*}\xrightarrow{\tau} 0|0$
- avec $A = a.A'$, $A' = \overline{c}.A$, $B = c.B'$ et $B' = \overline{b}.B$, prouvez :
  - $(A' + B')[b/c] \xrightarrow{\overline{b}} A[b/c]$
  - $(A|B)\backslash\{c\}[c/a] \xrightarrow{c} (A'|B)\backslash\{c\}[c/a]$
  - $b.a.0 + (A'|B)\backslash\{c\} \xrightarrow{\tau} (A|B')\backslash\{c\}$

Exercice 1.7 (Simulation et bissimulation)
Soient les deux processus $P = a.b.0 + a.0$ et $Q = a.b.0$.

- Dites (et prouvez ce que vous avancez) si P simule[2] Q, si Q simule P, si P et Q sont bissimilaires.

Exercice 1.8 (Bissimulation)
Avec les définitions suivantes: $R = b.R + b.a.0$, $S = b.T$, $T = b.T + a.0$ et $U = b.U + b.T$, prouvez pour chaque paire si elle correspond à deux processus bissimilaires:

- (R, S)
- (S, U)
- $(b.(U + T), b.T + b.U)$

Vérifiez vos réponses avec `eq -S bisim`.

Exercice 1.9 (BissimulationS)

[2] La définition de la simulation sera vue en TD.

Dessinez lorsque cela est possible les systèmes de transitions pour les processus suivants et dites s'ils sont bissimilaires, observationnellement équivalents ou non.

- $P = a.b.P + a.c.P$, $Q = a.(b.Q + c.Q)$
- $P = a.P + \tau.P$, $Q = a.Q$
- $P = a.P$, $Q = a.(Q|a.Q)$
- $P = a.0 + a.P$, $Q = a.(\tau.0 + \tau.Q)$
- $P = (a.b.0|\overline{b}.c.0)\backslash\{b\}$, $Q = a.c.0$

Vérifiez vos réponses avec `eq -S bisim` et `eq -S obseq`.

Exercice 1.10 (HML)

Avec $[K]\phi$ ($K \in \mathcal{A}$) défini comme $P \models [K]\phi$ si $\forall P' \in \mathcal{P}\ \forall a \in K, P \xrightarrow{a} P' \Rightarrow P' \models \phi$, donnez les définitions formelles et le sens en français de :

- $\langle K\rangle\phi$ ($\equiv \neg[K]\neg\phi$)
- $[-K]\phi$ ($\equiv [\mathcal{A} - K]\phi$)
- $[-]\phi$ ($\equiv [\mathcal{A}]\phi$)
- $\langle -\rangle\phi$
- Donnez la forme générale des formules HML permettant :
  - possibilité de faire une action
  - impossibilité de faire une action
  - blocage
  - nécessité de faire une action

Soit le processus $A = b.(c.A + a.b.0) + c.a.A + b.d.0$.

- Traduisez en français puis prouvez que $A \overset{?}{\models} \langle b\rangle[d, b]ff$
- Traduisez en français puis prouvez que $A \overset{?}{\models} [b, c]\langle a, d\rangle tt$
- Vérifiez vos réponses avec `chk Proc ‘‘Formule’’`.

Exercice 1.11 (Synthèse : vérification avec les algèbres de processus)

- Donnez différents types de propriétés pouvant être vérifiées dans le cadre d'un modèle décrit avec CCS
- Donnez un exemple de chaque type
- Donnez différentes techniques de vérifications possibles avec CCS
- Expliquez comment vérifier vos exemples

Exercice 1.12 (Synthèse : comparaison des équivalences)

Il existe de nombreuses équivalences dans le contexte des algèbres de processus.

- Donnez des exemples d'équivalences
- Décrivez ce qui les différencie. Donnez des exemples.

Exercice 1.13 (Modélisation : digicode)

Un digicode a dix boutons correspondant aux nombres de 0 à 1. Lorsque le code correct est entré, le digicode envoit un signal d'ouverture (à la porte). Lorsque le code est faux, un signal sonore est émis. Dans la suite, vous supposerez que le code est `1234`.

- Donnez un processus CCS correspondant au digicode
- Expliquez comment prouver que la porte ne s'ouvre que si le code a été correctement saisi

Exercice 1.14 (Modélisation : distributeurs de boissons)
Soient les deux distributeurs de boissons suivants:

1. il est possible de mettre un euro puis demander un thé ou bien remettre encore un euro et demander un café
2. il est soit possible de mettre un euro et demander un thé, soit mettre deux fois un euro et demander un café

- Donnez les expressions en CCS de ces processus
- Sont-ils trace-équivalents ? Sont-il bissimilaires ? Sont-ils observationnellement équivalents ? (si oui expliquez, si non donnez un contre exemple)
- modifiez le premier distributeur comme suit :
  - on peut mettre des pièces de un ou de deux euros
  - il est possible de mettre jusqu'à une somme totale de quatre euros avant de demander sa boisson
  - la machine rend la monnaie de façon non déterminée (mais non choisie par l'environnement extérieur)
- donnez une implantation "parallèle" (un changeur de monnaie + un distributeur) du distributeur et montrez qu'il est équivalent à une spécification plus abstraite

Exercice 1.15 (Modélisation : sémaphores)
On désire modéliser en CCS le fonctionnement d'un sémaphore. Un système à sémaphore est sensé fonctionner de la façon suivante :

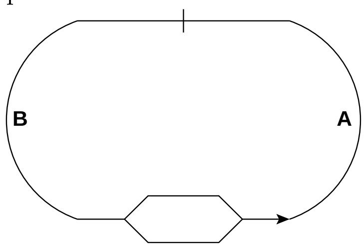
- un sémaphore assure la sécurité d'accès à une section critique
- le système est composé d'un ensemble de processus `P` mis en concurrence avec le sémaphore, `Sem`
- les processus communiquent avec le sémaphore en faisant les actions `p` pour demander à entrer en section critique et `v` pour en sortir

- Ecrivez en CCS le comportement du sémaphore et prouvez qu'il est correct
- Donnez un exemple (intelligent) de processus `P`
- Décrivez le système
- Vérifiez sur un système à deux processus `P` que le système assure la sécurité d'accès

Exercice 1.16 (Modélisation : sémaphores (2))
On désire généraliser les sémaphores à k accès possibles en même temps en section critique.

- Donnez un exemple d'utilisation d'un tel sémaphore
- En notant `Sem(k)` ce type de sémaphore, donnez une définition utilisant des sémaphores simples `Sem`
- Donnez une autre définition en utilisant la valeur de k
- Montrez que les deux définitions sont bissimilaires

Exercice 1.17 (Modélisation : train électrique)
On désire modéliser le train électrique suivant:



Le circuit est constitué d'une voie unique avec une station à double voie. La voie hors station est composée de deux sections, A et B. Les trains circulent dans une seule direction. Il est possible d'arrêter les trains grâce à des signaux au niveau de la station ou à la connection des sections.

- Donnez les propriétés de sécurité qu'il est indispensable d'assurer sur ce système
- Décriver la spécification du système en CCS (le plus simple et structuré possible) en utilisant les actions observables suivantes :
  - `a` : un train entre en section A
  - `b` : un train entre en section B
  - `s` : un train entre en station

  Appelez ce processus `SpecTrain`
- Décrivez une implantation possible de ce système avec N trains et en utilisant des éléments permettant de controller les trains (les signaux). Appelez ce processus `ImplTrainN`. Essayez avec differentes valeurs de N. Quelle(s) valeur(s) conduisent à un blocage ?
- Justifiez que votre implantation du système est correcte par rapport à sa spécification et vérifiez le.
- Les propriétés de sécurité sont-elles vérifiées ?

Exercice 1.18 (Modélisation : imagination)
Choisissez une activité/machine/scénario plus ou moins habituel(le) à modéliser en CCS. Vous pouvez choisir ce que vous voulez (c'est la chance pour vous d'être imaginatifs et pour moi d'être surpris), à condition de rentrer dans les contraintes présentées ci-dessous.
Il est évident que l'objectif n'est pas de faire un exemple vu en cours ou en TD.

- décrivez les actions observables (au moins 5)
- donnez une description textuelle du système (spécification et implantation)
- donnez une spécification et une implantation possible en CCS (au moins 7 états et un branchage intéressant)
- donnez des propriétés à vérifier et vérifiez les

Exercice 1.19 (+ sans plus)

- Donnez une définition du choix en n'utilisant pas l'opérateur $+$.

Concrètement, étant donnés deux processus P et Q, définissez un processus R, ne contenant pas le symbole $+$ et qui est fortement bissimilaire au processus $\tau.P + \tau.Q$.

Exercice 1.20 (Un morceau de preuve de congruence ...)

- Prouvez que la bissimulation forte est fermée par rapport à la composition parallèle, *i.e.* que $\forall R \in \mathcal{P}, P \sim Q \Leftrightarrow P|R \sim Q|R$