TD 4 — UE Spécifications Formelles
TD4 : Modélisation CCS

Exercice 1

CCS → actions + processus

a) $Serveur \triangleq accept.\overline{session}.Session$

barre, car pt de vue serveur, donne / quelque chose

$Session \triangleq upload.Session + \overline{download}.Session + end.Serveur$

le serveur reçoit des trucs, puis peut continuer la session

le serveur "donne" des choses, puis peut continuer la session

qd on fait end, session fermée, on revient à l'état initial du serveur. on peut réinitialiser une new session.

b) $\overline{session_1}$ $upload_1$ $download_1$ end_1 — $\overline{session_2}$ $upload_2$ $download_2$ end_2

Les 2 serveurs acceptent le même msg d'ouverture de session, c'est la seule chose qu'ils ont en commun.

$Serveur_1 \triangleq accept.\overline{session_1}.Session_1$

$Session_1 \triangleq upload.Session_1 + \overline{download_1}.Session_1 + end_1.Serveur_1$

$Serveur_2 \triangleq accept.\overline{session_2}.Session_2$

$Session_2 \triangleq upload_2.Session_2 + \overline{download_2}.Session_2 + end_2.Serveur_2$

$ServeurDouble \triangleq Serveur_1 \parallel Serveur_2$

Exercice 2

a) $accept\ \ session_1\ \ \overline{end_1}\ \ 0$

le client juste se connecte et finit la session (ds l'énoncé, = sans échange de données)

$clientFutif \triangleq \overline{accept}.(Session_1.\overline{end_1}.0 + \ldots Session_2.\overline{end_2}.0)$

↳ le client envoie la demande accept, si le serveur 1 lui envoie la confirmation de session ($session_1$), la session est créée, il va envoyer un message de fin de session $\overline{end_1}$ au serveur, la session est finie (0).

- raffinement (raffinage) de programmes séquentiels.
- raffinement par la relation d'héritage dans les langages objets.
- raffinement d'une spécification par une implantation : exemple des annotations JML JAVA qui constituent une spécification.
- raffinement de données : la structure de liste est un raffinement (implantation) de la notion d'ensemble.

ble de propriétés E.
de conservation de ces propriétés.
ffine un système S ssi S' conserve toutes
hées à S.

...ations formelles
...stèmes de transitions . (Bi)sim...

...isimulation

$\to s_0' \xrightarrow{a} s_2'$ —
$\xrightarrow{a} s_1'$ ε

...imulé par ℛ'
...ts initiaux
$\langle s_1, s_1' \rangle$ } est une rela°

Pour mq $\langle s_0, s_0' \rangle \in R$, il fa...

∃ un état s_1' tq $s_1' \xrightarrow{b}$
$s_1 \xrightarrow{b} s_1$.

...itiaux de S (s_0)
($\langle s_0, s_0' \rangle \in R$).

...at final, et on applique
...∈ S',
$s_1 \xrightarrow{a} s_2$ (∈R)
...me ∧ $s_1' \xrightarrow{a} s_2'$ (∈R...

...de S.

..., $\langle s_2', s_1 \rangle$, $\langle s_3$...

P = end . 0

⚠ **Remarque**

a $\overline{accept}.session_1.P_1 + \overline{accept}.session_2.P_2$

≠

b $\overline{accept}(session_1.P_1 + session_2 P_2)$

Si a, on fait soit accept, puis session, puis P_1, soit accept, puis session$_2$, puis P_2.
→ faux car on ne sait pas direct quel serveur va répondre.

Si b, on fait accept et là on décide sur quel serveur s'exécuter.
→ bien

b) le client envoie la demande de connec° (accept), il reçoit la valida° d'ouverture de se° de 1 ou 2 ($se°_1$, $se°_2$) puis il entre ds sa session.

$Client \triangleq \overline{accept}.(session_1.SessionClient_1 + session_2.SessionClient_2)$

$\forall i \in \{1,2\}$,
$SessionClient_i \triangleq \overline{upload_i}.SessionClient_i + download_i.SessionClient_i + \overline{end_i}.0$

↓ met le Processus 0 car la syntaxe est action. Processus. on aurait pas pu mettre juste $\overline{end_i}$ (Processus ≠ action. Processus).

SessionClient$_i$ m̄ chose que Session$_i$ mais avec les barres sur les actions "inversées".

Quant SessionClient$_i$ (= session sur le serveur i), il peut uploader des fichiers vers le serveurs (il "donne" → upload$_i$) et continuer la se°, ou il peut downloader des fich depuis le serveur (il = reçoit → download$_i$) et continuer la se°, ou il peut envoyer un message de termina° de session (il "envoie" → $\overline{end_i}$) puis le P est fini (0).

c) $NouveauxClient \triangleq \tau.(Client \parallel NouveauxClient) + \tau.0$

. Ajouter un nouveau client → prendre le client ou rien, si client on peut en ajouter un autre en ∥ q̄ est soit un new Client, soit le P terminal (0).
. τ est mis pour faire action. Processus
↓ neutre τ ↓ neutre 0

Si Q ∞ l'ac° b
→ νbQ veut dire que b est une ac° masquée

a) oui on peut masquer les canaux de communications avec les variables masquées (ou locales).

Exercice 3

a) [$\nu accept.\nu session_1.\nu session_2 ... \nu end_2$ ~~.(ServeurDouble~~
.(ServeurDouble ∥ NouveauxClient)] masqué ∧ ServeurDouble ∥ NouveauxClient (non masqué)

$\dfrac{P \xrightarrow{l} Q}{\nu a.P \xrightarrow{l} \nu a.Q}$ $l \neq a, \bar{a}$] rela° entre les versions masquée et non masquée
↳ Q ↓ νa.Q

b) le ClientFurtif ne peut pas avoir d'exécu° ∞, alors qu'avec le client normal (upload, download qd il veut), oui on en a.

divergent → exécu° ∞

Exercice 4

a)

$NouveauxClient' \triangleq \tau.\underline{do}.(Client' \parallel NouveauxClient') + \tau.0$
(do : en début de session est reçu par le client)

$SessionClient'_i \triangleq \overline{upload_i}.SessionClient'_i + download_i.SessionClient'_i + \overline{end_i}.\underline{done}.0$
(done : done est reçu en fin de session par le client (≃ log).)

le reste ne change pas.

b) $\nu accept.\nu session_1.\nu session_2 ... \nu end_2.(Client' \parallel ServeurDouble)$

c) ≈ done.0

b) Car ∃ vu au 3.b, les clients normaux peuvent avoir des exécu° ∞ (pas de 0 à un moment), contrairement aux clients furtifs (qui ont un 0, donc fini.

ENSEEIHT – 3ème année

1 Communicati...

On souhaite modéliser...
sentera 2 processus princip...
— le serveur qui peut ...
de session session...
— attendre la fin ...
— recevoir une inf...
— envoyer une inf...
— le client qui joue le...

Exercice 1 (Serveur)
a. Modéliser un serveur n...
b. On souhaite augmenter... on distinguer ces différ...

Exercice 2 (Client)
a. Modéliser un client furt... connections, sans échan...
b. Modéliser un client nor... deux connections.
c. Modéliser l'arrivée possi...

Exercice 3 (Système co...
a. Modéliser un système ... clients. Peut-on masquer... les versions masquée et n...
b. Le système proposé peut...

Exercice 4 (Propriétés)
...u système, de telle sorte qu... l'événement do (resp. don...
Modifier le système pour...
Quelle hypothèse sur le ... normaux terminent leur...
À quel processus le systè...

Exercice 5 Comment mod...