

Corrigé type examen OF 2024

Questions de cours

1. Systèmes critiques

2. Validation précoce, précision, complétude, documentation stable, vérification plus exhaustive, coût réduit.

3. Exemple :

var E : Edge .

vars P Q R S : Path .

cmb E ; P : Path if target(E) = source(P) .

ceq (P ; Q) ; R = P ; (Q ; R)

if target(P) = source(Q) \wedge target(Q) = source(R) .

4. Voir le cours

Exercice 1

fmod nombre is

protecting NAT .

protecting BOOL .

sort List .

subsort Nat < List .

op nil : -> List [ctor] .

op _:_ : List List -> List [ctor assoc id: nil] .

op len : List -> Nat .

op chiffre : Nat -> List .

op inverser : List -> List .

op suphalf : List Nat Nat -> List .

op number : List -> Nat .

vars N M I : Nat .

var L : List .

eq len(nil) = 0 .

eq len(N, L) = 1 + len(L) .

ceq chiffre(N) = (N rem 10), chiffre(N quo 10) if N \neq 0 .

```

ceq chiffre(N) = 0 if N == 0 .
ceq inverser(L, N) = N, inverser(L) if L /= nil .
ceq inverser(L, N) = N if L == nil .
ceq suphalf((N, L), I, M) = N, suphalf(L, I + 1, M) if I /= M and L /= nil .
ceq suphalf((N, L), I, M) = suphalf(L, I + 1, M) if I == M and L /= nil .
ceq suphalf((N, L), I, M) = N if L == nil .
eq number(N, L) = N * (10 ^ len(L)) + number(L) .
ceq number(L) = 0 if L = nil .

endfm

reduce number(suphalf(inverser(chiffre(5590155)), 1, (len(inverser(chiffre(5590155))) quo 2) + 1 )) .

```

Exercice 2

```

(omod Lavage is
protecting INT .

sort etatLav .

ops debut lavage sechage attente lustrage fin1 fin2 indef : -> etatLav .

class systeme | etat : etatLav, H : etatLav, horloge : Int .

msgs reprise urgence1 urgence2 : Oid -> Msg .

var S : etatLav .

var V : Oid .

var T : Int .

rl [debut] : < V : systeme | etat : debut, H : S, horloge : T > => < V : systeme | etat : lavage, H : S,
horloge : T > .

crl [lavage] : < V : systeme | etat : lavage, horloge : T > => < V : systeme | etat : lavage, horloge : T - 1
> if T > 0 .

crl [lavage-sechage] : < V : systeme | etat : lavage, horloge : T > => < V : systeme | etat : sechage,
horloge : 2 > if T == 0 .

crl [sechage] : < V : systeme | etat : sechage, horloge : T > => < V : systeme | etat : sechage, horloge :
T - 1 > if T > 0 .

crl [sechage-lustrage] : < V : systeme | etat : sechage, horloge : T > => < V : systeme | etat : lustrage,
horloge : 2 > if T == 0 .

crl [lustrage] : < V : systeme | etat : lustrage, horloge : T > => < V : systeme | etat : lustrage, horloge :
T - 1 > if T > 0 .

crl [lustrage-fin2] : < V : systeme | etat : lustrage, horloge : T > => < V : systeme | etat : fin2, horloge :
2 > if T == 0 .

```

rl [arret-fin2] : urgence2(V) < V : systeme | etat : lustrage > => < V : systeme | etat : fin2 > .

rl [lavage-attente] : urgence1(V) < V : systeme | etat : lavage > => < V : systeme | etat : attente, H : lavage > .

rl [lavage-attente] : urgence1(V) < V : systeme | etat : sechage > => < V : systeme | etat : attente, H : sechage > .

crl [attente] : < V : systeme | etat : attente, horloge : T > => < V : systeme | etat : attente, horloge : T - 1 > if T > 0 .

crl [attente-fin1] : < V : systeme | etat : attente, horloge : T > => < V : systeme | etat : fin1, horloge : 2 > if T == 0 .

crl [attente-H] : reprise(V) < V : systeme | etat : attente, H : S > => < V : systeme | etat : lavage, horloge : 4 > if S == lavage .

crl [attente-H] : reprise(V) < V : systeme | etat : attente, H : S > => < V : systeme | etat : sechage, horloge : 2 > if S == sechage .

endom)

(rew urgence1(V) reprise(V) < V : systeme | etat : debut, H : indef, horloge : 4 > .)