Réponse à la question 2

1) Invariants de l'algorithme

On montre que pour tous agents p, q pendant toute l'exécution :

a) mSent[p][q] ≤ ds[p][q] ≤ p.aSent[q]

- mSent[p][q] ne contient que des messages déjà reçues par le moniteur, donc des valeurs passées alors, mSent[p][q] ≤ ds[p][q]
- Si un message ('t', s, r) est en ransit, alors ds[p][q] = s[q], qui correspond à l'état local de p au moment de l'émission, donc s[q] ≤ p.aSent[q]
- Sinon ds = mSent[p][q]

  Donc mSent[p][q] ≤ ds[p][q] ≤ p.aSent[q]

b) mRcvd[p][q] ≤ dr[p][q] ≤ p.aRcvd[q]

- mRcvd[p][q] ne contient que des messages déjà reçues par le moniteur, donc des valeurs passées alors, mSent[p][q] ≤ ds[p][q]
- r[q] reflète l'état local de p lors de l'envoie du t
- r[q] ≤ p.aRcvd[q] car les messages reçus après émission du 't' n'y figurent pas

  Donc mRcvd[p][q] ≤ dr[p][q] ≤ p.aRcvd[q]

c) p.aSent[q] = q.aRcvd[p] + (messages 'm' en transit entre p et q)

- Chaque fois que p envoie un message à q, p.aSent[q] est incrémenté
- Si q le reçoit, q.aRcvd[p] est incrémenté
- La différence compte exactement les messages en transit

  Donc c'est un invariant naturel du modèle FIFO

d) Ordre des messages ‘t’

Si deux messages (‘t’, s, r) et (‘t’, s’, r’) transitent dans le canal entre p et le moniteur avec t avant t’, alors : s[p][q] ≤ s’[p][q] & r[p][q] ≤ r’[p][q]

Car l’agent ne décrémente jamais ses compteurs, et que les canaux sont FIFO

2) Propriété Coherent(A) & Menace(A)

- On montre :

A ⊆ eseen, si Coherent(A) et Menace(A) sont vraies alors

∃p ∈ A, q ∉ A tel que dr[p][q] ≠ p.aRcvd[q]

- Idée intuitive :

  - Coherent(A) garantit qu’à l’intérieur de A, il n’y a pas de différence de compteurs entre ce que les uns ont envoyé et ce que les autres ont reçu

  - Menace(A) garantit qu’au moins un agent de A n’est pas stabilisé

  - Pour que cette instabilité subsiste, il faut qu’un des flux de messages s’échappe vers l’extérieur, donc vers q ∉ A

- Preuve :
  - p.active = vrai

  Un agent actif peut encore envoyer un message vers un agent extérieur, donc message possible vers q ∉ A

  - ds[p][q] ≠ p.aSent[q]

Cela signifie qu’un message (‘t’, s, r) de p est en transit

Comme A ⊆ eseen, cela signifie que le moniteur n’a pas encore stabilisé l’état de p

Ce message peut concerner un q ∉ A

  - dr[p][q] ≠ p.aRcvd[q]

Alors le message m reçu p vient nécessairement de q ∉ A, sinon Coherent(A) serait violé.

Dans les trois cas, on en déduit :

$\exists p \in A,\ q \notin A$ tel que $dr[p][q] \neq p.aRcvd[q]$

3) Conclusion, correction de l'algorithme

Supposons que le moniteur déclare la terminaison.

Alors par condition de Kumar :

1. De tous les agents eseen = ensemble de tous les agents
2. Coherent(eseen) est vraie
3. Menace(eseen) est fausse

Par contraposée du point (2.), si Menace était vraie, un message transiterait vers l'extérieur, ce qui est impossible puisque eseen contient tout le monde.

Donc :

- aucun agent n'est actif
- aucun message n'est en transit
- tout message envoyé a été reçu

Ce qui est la définition de la terminaison globale

Donc l'algorithme de Kumar est correct