

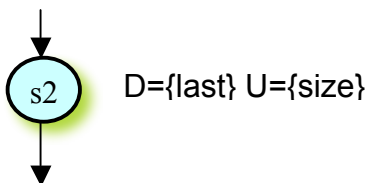
Examen Final

LOG3430

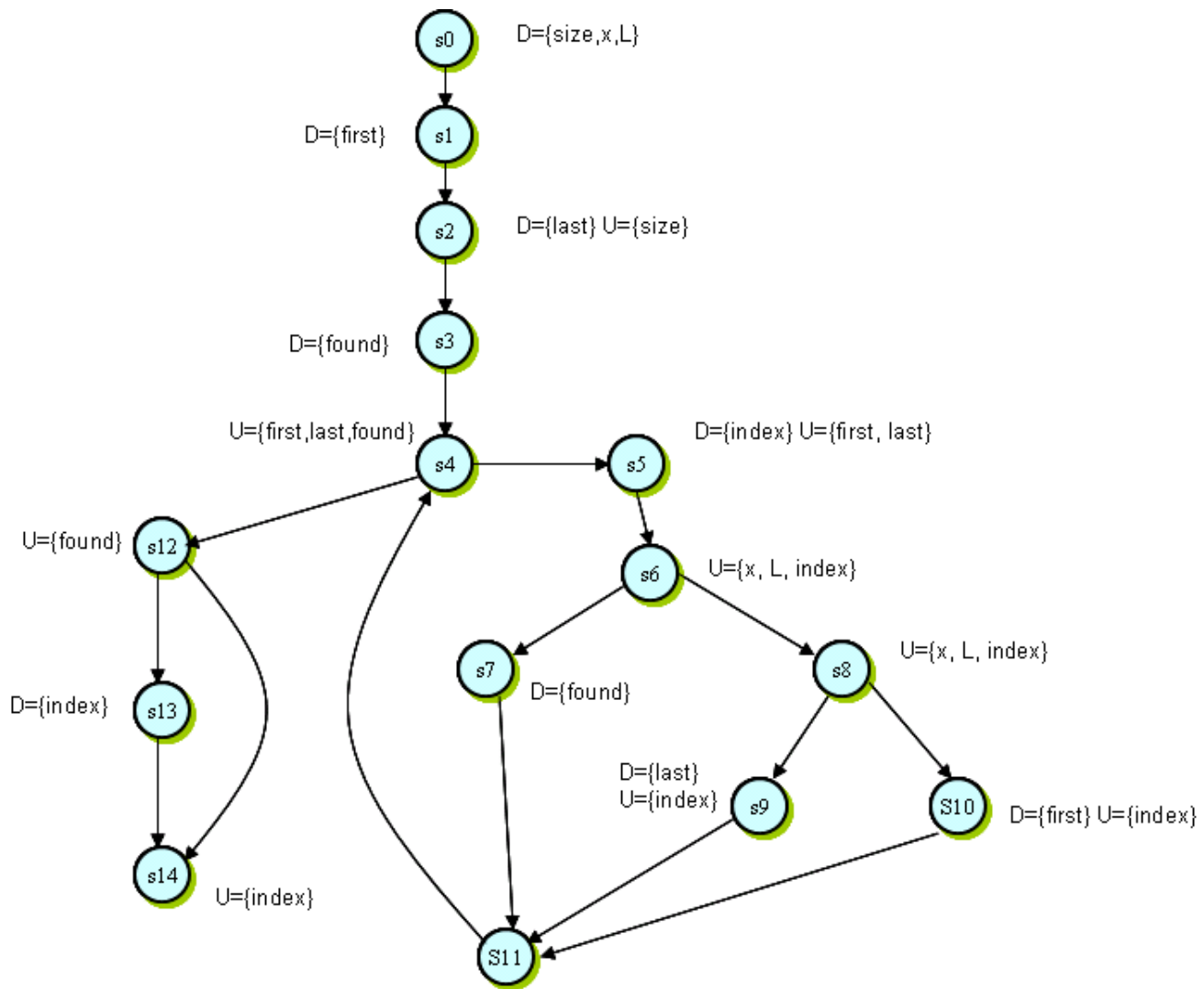
Problème 1 : Pour le programme suivant:

```
0. int binary(int size, int x, int L[ ]){  
    Int first, last, found, index;  
1. first = 0;  
2. last = size -1;  
3. found = 0;  
4. while (first <= last && !found) {  
5.     index = (first+last)/2;  
6.     if( x==L[index] )  
7.         found = 1;  
8.     else if (x < L[index])  
9.         last = index-1;  
10.    else first = index +1;  
11.    }  
12. if (!found)  
13.     index = -1;  
14. printf("%d\n",index);  
}
```

QP1.1) Utilisez les numéros des lignes pour identifier les noeuds et donnez le *Graphe de flot de contrôle*; indiquez les ensembles des définitions et usages pour chaque sommet e.g.,



SOLUTION



QP1.2) Déterminez les définitions-usages des données; donnez les résultats en replissant le tableau suivant :

Dans chaque cellule écrivez les lignes ou la variable (prime cellule) est usage, voir l'exemple pour x qui est défini a la ligne 0 et usage a le lignes 6 et 8

| Defin. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|--------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|
| size | 2 | | | | | | | | | | | | | |
| x | | | | | | | | | | | | | | |
| L | | | | | | | | | | | | | | |
| first | | | | | | | | | | | | | | |
| last | | | | | | | | | | | | | | |
| found | | | | | | | | | | | | | | |
| index | | | | | | | | | | | | | | |

SOLUTION

| | | | | | | | | | | | | | | |
|--------|-----|-----|-----|------|---|-------------|---|------|---|-----|-----|----|----|----|
| Defin. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| size | 2 | | | | | | | | | | | | | |
| x | 6,8 | | | | | | | | | | | | | |
| L | 6,8 | | | | | | | | | | | | | |
| first | | 4,5 | | | | | | | | | 4,5 | | | |
| last | | | 4,5 | | | | | | | 4,5 | | | | |
| found | | | | 4,12 | | | | 4,12 | | | | | | |
| index | | | | | | 6,8,9,10,14 | | | | | | | | 14 |

QP1.3) Pour un vecteur L de trois cellules ou mineur (i.e., size ≤ 3) donnez les valeurs du x, size et L[] pour couvrir le critère all-uses. Pour chaque valeur de x, size et L[] précisez la/les définition-use couverte(s). Les données doivent être spécifiées dans le suivant tableaux :

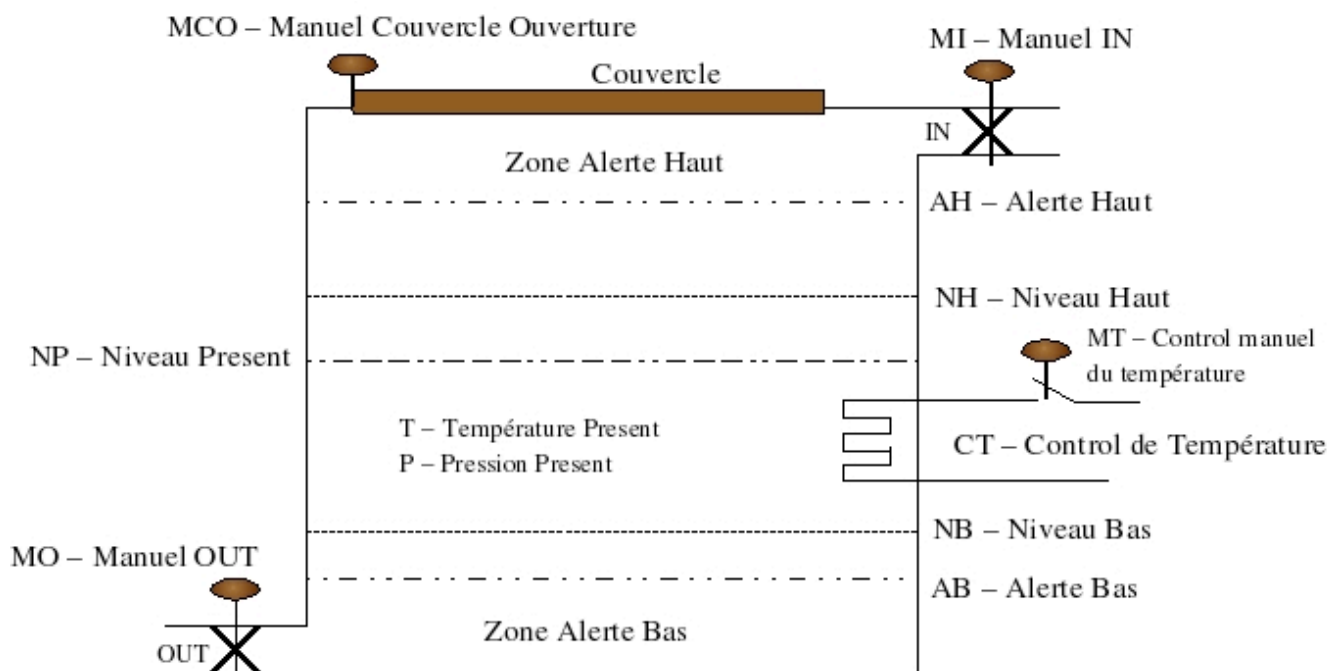
| Cas du test | x | size | L[0] | L[1] | L[2] | index | DU couverte | | | | | | |
|-------------|---|------|------|------|------|-------|-------------|---|---|-------|------|-------|-------|
| | | | | | | | size | x | L | first | last | found | index |
| T1 | | 3 | 0 | 1 | 3 | | | | | | | | |
| T2 | | | | | | | | | | | | | |
| T3 | | | | | | | | | | | | | |
| T4 | | | | | | | | | | | | | |
| T5 | | | | | | | | | | | | | |
| T6 | | | | | | | | | | | | | |

SOLUTION

| Cas du test | x | size | L[0] | L[1] | L[2] | index | DU couverte | | | | | | | |
|-------------|---|------|------|------|------|-------|-------------|---------|---------|---------|---------|-------|---------|--|
| | | | | | | | size | x | L | first | last | found | index | |
| T1 | 4 | 3 | 0 | 1 | 2 | -1 | 0.2 | 0.6 0.8 | 0.6 0.8 | 1.4 1.5 | 2.4 2.5 | 3.4 | 5.6 5.8 | |

| | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|-----|---------|---------|---------|--------------------|--------------------|----------------|
| | | | | | | | 0,2 | 0,6 | 0,6 | 1,4 1,5 | 2,4 2,5 | 3,4 7,4 7,12 | 5,6 5,14 |
| T2 | 1 | 3 | 0 | 1 | 2 | 1 | | | | | | | |
| | | | | | | | 0,2 | 0,6 0,8 | 0,6 0,8 | 1,4 1,5 | 2,4 2,5 9,4 9,5 | 3,4 7,4 7,12 | 5,6 5,8 5,9 |
| T3 | 0 | 3 | 0 | 1 | 2 | 0 | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

Problème 2 : La figure suivante donne une idée du principe d'une cuve industrielle pour la stérilisation du lait. Température, pression, niveau du lait, ouverture du couvercle, opérations du chauffage, du remplissage et de vidage sont contrôlées par un logiciel.



Les contraintes suivantes sont à respecter :

- $NP > AB$ et $NP < AH$, NP est le niveau courant du lait
- Si $NP < AB$ ou $NP > AH$ il faut générer une alerte
- Si $AB < NP < NB$ il faut avoir $OUT = F$
- Si $NH < NP < AH$ il faut avoir $IN = F$
- MO et MI ne peuvent pas désactiver le signal d'alerte
- En mode automatique il faut forcer $IN = T$ si $NP < NB$ et $T < T_{max}$ et $P < P_{max}$
- En mode automatique il faut forcer $OUT = T$ si $NP > AH$ et $T < T_{max}$ et $P < P_{max}$
- En mode automatique il faut forcer $CT = T$ si $NB < NP < NH$ et $T < T_{max}$ et $P < P_{max}$

- Les modes manuels peuvent forcer IN=T ou OUT=T ou CT=T

Des transducteurs transforment T, P, NP, MCT,

- TB (A) est vrai si $T < T_{max}$
- PB (B) est vrai si $P < P_{max}$
- COB (C) est vrai si le couvercle est ouvert
- NPB (D) est vrai si $NB < NP < NH$
- MCT (E) est vrai si la commande manuelle force le control de température
- CTB (Z) doit être vrai si on doit réchauffer le lait

Les valeurs suivantes ont été utilisées pour le système :

| PARAM. | VAL. |
|----------|-------------|
| AB | 15 |
| NB | 20 |
| NH | 100 |
| AH | 110 |
| Pmax | 2 |
| Tmax | 150 |
| MI,MO,MT | TRUE, FALSE |

Une compagnie a développé un logiciel qui implémente la fonction logique:

$$CTB = TB \text{ PB } (MCB + NPB \overline{COB}) = A \text{ B } (E + D \overline{C})$$

Une compagnie concurrente a donné une fonction :

CTLOUT(NP,T,P,MO)

La fonction CTLOUT prend trois paramètres numériques entiers (NP, T et P) et une variable (chaîne de caractères) de trois lettres pour générer la variable de control OUT qui est retournée par la fonction CTLOUT.

QP2.1 générez un jeux de test pour la fonction $Z = A \text{ B } (E + D \overline{C})$ avec les critères de couverture des clauses (clause coverage) et ACC. Organisez les résultats dans les tableaux (Vrai=TRUE=1, Faux=FALSE=0) :

Clause Coverage

| TC | A | B | C | D | E | Z |
|----|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Active Clause Coverage(ACC) - Résultats Intermediaire

| Clause | A | B | C | D | E | Z | A | B | C | D | E | Z |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | | | | | | | | | | | |
| B | | | | | | | | | | | | |
| C | | | | | | | | | | | | |
| D | | | | | | | | | | | | |
| E | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |

Active Clause Coverage(ACC) - Jeux de test

| Cas du Test | A | B | C | D | E | Z |
|-------------|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

SOLUTION

Clause Coverage

| TC | A | B | C | D | E | Z |
|-----|---|---|---|---|---|---|
| Tc1 | 1 | 1 | 0 | 1 | 0 | 1 |
| Tc2 | 0 | 0 | 1 | 0 | 1 | 0 |

Les combinassions où chaque clause est évaluée à vrai et faux.

Active Clause Coverage(ACC) - Résultats Intermediaire

| Clause | A | B | C | D | E | Z | A | B | C | D | E | Z |
|--------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| A | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| B | <i>1</i> | <i>1</i> | <i>0</i> | <i>1</i> | <i>1</i> | <i>1</i> | 1 | 0 | 0 | 1 | 1 | 0 |
| C | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| D | <i>1</i> | <i>1</i> | <i>0</i> | <i>1</i> | <i>0</i> | <i>1</i> | 1 | 1 | 0 | 0 | 0 | 0 |
| E | 1 | 1 | 0 | 0 | 1 | 1 | 1 | <i>1</i> | <i>0</i> | <i>0</i> | <i>0</i> | <i>0</i> |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |

Active Clause Coverage(ACC) - Jeux de test

| Cas du Test | A | B | C | D | E | Z |
|-------------|---|---|---|---|---|---|
| Tc1 | 1 | 1 | 0 | 1 | 1 | 1 |
| Tc2 | 0 | 1 | 0 | 1 | 1 | 0 |
| Tc3 | 1 | 0 | 0 | 1 | 1 | 0 |
| Tc4 | 1 | 1 | 0 | 1 | 0 | 1 |
| Tc5 | 1 | 1 | 1 | 1 | 0 | 0 |
| Tc6 | 1 | 1 | 0 | 0 | 0 | 0 |
| Tc7 | 1 | 1 | 0 | 0 | 1 | 1 |

Les combinaisons où chaque « clause majeure » est vraie et fausse et détermine Z vraie/fausse

QP2.3 Il est requis de définir un test frame avec la méthode category partition pour la fonction CTLOUT selon le critérium *each choice* et *base choice*. Pour le dernière n'est pas requis de détailler tous les combinaisons mais il est suffisant d'exprimer avec expression régulières les éléments du test frame

SOLUTION

La fonction CTLOUT prend trois paramètres numériques entiers NP, T et P :

- NP numérique entier categories
 - Zone critique bas : $NP < AB, AB \leq NP < NB$
 - Zone OK : $NB \leq NP < NH$
 - Zone critique haut : $NH \leq NP < AH, AH < NP$
 - NON VALID
 - NON ENTIER
 - NON NUMÉRIQUE
- T numérique entier categories
 - Zone OK $T < T_{max}$
 - Zone critique haut $T \geq T_{max}$
 - NON VALID
 - NON ENTIER
 - NON NUMÉRIQUE
- P numérique entier categories
 - Zone OK $P < P_{max}$
 - Zone critique haut $P \geq P_{max}$
 - NON VALID
 - NON ENTIER
 - NON NUMÉRIQUE
- MO 3 caractères categories
 - TRUE
 - FALSE
 - OTHER

Choices

- NP
 - NP1) NP<AB
 - NP2) AB<=NP<NB
 - NP3) NB<=NP<NH
 - NP4) NH<=NP<AH
 - NP5) AH<NP
 - NON VALID
 - NP6) NON ENTIER [error]
 - NP7) NON NUMÉRIQUE [error]
- T
 - T1) T<Tmax
 - T2) T>=Tmax
 - NON VALID
 - T3) NON ENTIER [error]
 - T4) NON NUMÉRIQUE [error]
- P
 - P1) P<Pmax
 - P2) P>=Pmax
 - NON VALID
 - P3) NON ENTIER [error]
 - P4) NON NUMÉRIQUE [error]
- MO
 - M1) TRUE
 - M2) FALSE
 - OTHER
 - M3) 3 caractères [error]
 - M4) 0-2 caractères [error]
 - M5) >3 caractères [error]
 - M6) Numerique [error]

Each Choice

One value from each choice for each categorie must be used at least in one test case

NP1 T1 P1 M2
 NP2 T2 P2 M1
 NP3 T3 P3 M3
 NP4 T4 P4 M5
 NP5 T1 P1 M6
 NP6 T1 P1 M6
 NP7 T1 P1 M6

Base Choice

Base : NP1, T1, P1, M2 le test frame est :
 NP1 x T1 P1 M2

NP1 T1 x M2
NP1 T1 P1 x

Avec la condition d'erreur n'est pas nécessaire de faire tous les combinassions.

Problème 3 : Le tableau suivant (TAB 1) contient les défaillances registrées (en sec.) pour un système réel:

| | | | | |
|------|------|------|-------|-------|
| 311 | 3085 | 5447 | 8625 | 11486 |
| 366 | 3089 | 5644 | 8982 | 12708 |
| 608 | 3089 | 5837 | 9175 | 13251 |
| 676 | 3565 | 5843 | 9411 | 13261 |
| 1098 | 3623 | 5922 | 9442 | 13277 |
| 1278 | 4080 | 6738 | 9811 | 13806 |
| 1288 | 4380 | 8089 | 10559 | 14185 |
| 2434 | 4477 | 8237 | 10559 | 14229 |
| 3034 | 4740 | 8258 | 10791 | 14358 |
| 3049 | 5192 | 8491 | 11121 | 15168 |

Q3.1) Pour le model de Musa base :

$$\mu(t) = \beta_0 [1 - e^{-\beta_1 t}] \text{ with } \beta_{0,1} > 0$$

$$\lambda(t) = \beta_0 \beta_1 e^{-\beta_1 t}$$

utilisez les équations:

$$\hat{\beta}_0 = \frac{n}{1 - \exp[-\hat{\beta}_1(t_n + x)]}$$

$$\frac{n}{\hat{\beta}_1} - \frac{n(t_n + x)}{\exp[\hat{\beta}_1(t_n + x)] - 1} - \sum_{i=1}^n t_i = 0$$

Pour obtenir les valeurs des paramètres bêtas pour les donnes du TAB 1 (la somme des temps de défaillance est 356173); la fonction exponentielle peut être approximée pour série de Tailor ordre 2 :

$$e^x \approx 1 + x + \frac{x^2}{2}$$

SOLUTION

$$\frac{50}{\hat{\beta}_1} - \frac{50(15168)}{\exp[\hat{\beta}_1 15168] - 1} - 356173 = 0$$

$$\frac{50}{\hat{\beta}_1} - \frac{50(15168)}{[\hat{\beta}_1 15168/2 + 1]15168\hat{\beta}_1} - 356173 = 0 \quad \frac{1}{\hat{\beta}_1} - \frac{1}{[\hat{\beta}_1 7584 + 1]\hat{\beta}_1} - 7123.46 = 0$$

$$7584 - 7123.46 \left[\hat{\beta}_1 7584 + 1 \right] = 0 \quad \hat{\beta}_1 \neq 0 \quad \hat{\beta}_1 = 460.72 / 7584 * 7123.46 = 8.528 \cdot 10^{-6}$$

$$\hat{\beta}_0 = \frac{50}{1 - \exp \left[-\hat{\beta}_1 (15168) \right]} = 412$$

Pour le model Musa base la relation entra défailances trouves et intensité de défailances est :

$$\lambda = \lambda_0 \left(1 - \frac{\mu}{\nu_0} \right)$$

Pour k=5 déterminez :

Q3.2) les couples (m,r) et les valeurs de λ_0, ν_0 pour les donnes du TAB 1;

SOLUTION

| | | | | |
|-------------|-------------|-------------|--------------|--------------|
| 311 | 3085 | 5447 | 8625 | 11486 |
| 366 | 3089 | 5644 | 8982 | 12708 |
| 608 | 3089 | 5837 | 9175 | 13251 |
| 676 | 3565 | 5843 | 9411 | 13261 |
| 1098 | 3623 | 5922 | 9442 | 13277 |
| 1278 | 4080 | 6738 | 9811 | 13806 |
| 1288 | 4380 | 8089 | 10559 | 14185 |
| 2434 | 4477 | 8237 | 10559 | 14229 |
| 3034 | 4740 | 8258 | 10791 | 14358 |
| 3049 | 5192 | 8491 | 11121 | 15168 |

Tableau du temps, défailainces, etc :

| Init | End | Delta | r | m |
|-------|-------|-------|----------|----|
| 0 | 1098 | 1098 | 0.004579 | 0 |
| 1098 | 3049 | 1951 | 0.002563 | 5 |
| 3049 | 3623 | 574 | 0.008710 | 10 |
| 3623 | 5192 | 1569 | 0.003187 | 15 |
| 5192 | 5922 | 730 | 0.006849 | 20 |
| 5922 | 8491 | 2567 | 0.001948 | 25 |
| 8491 | 9442 | 951 | 0.005257 | 30 |
| 9442 | 11121 | 1679 | 0.002978 | 35 |
| 11121 | 13277 | 2156 | 0.002319 | 40 |
| 12277 | 15168 | 2891 | 0.001729 | 45 |

Si $y=a+bx$ et $x=m$, $y=r$, b est :
$$\hat{b} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - \left(\sum x_i \right)^2}$$

$$\hat{b} = \frac{\lambda_0}{\nu_0} = \frac{10 * 0.7659 - 225 * 0.04011}{10 * 7125 - 50625} = -6.6314e^{-5} \quad a = \bar{r} + \bar{m}b = 0.004 + 6.63e^{-5} * 22.5 = 0.0055$$

$$\lambda_0 = 0.0055, \nu_0 = 82.99 \approx 83$$

Q3.4) combien de défauts sont encore dans le logiciel ?

SOLUTION

On a trouve 50 défaillances et on estime que encore 33 fautes sont dans le logiciel.

Q3.5) Si le logiciel est de 25000 LOC quelles sont les pourcentages deux défauts initiales et finales ?

SOLUTION

Les pourcentages initial et final de fautes sont 83/25000 et 33/25000 i.e. 3.3 et 1.3 pour KLOC.

Q3.6) Si la valeur de λ n'est pas acceptable et pas plus d'une défaillance par jour est permise, déterminez des défaillances à corriger et le temps CPU des activités du test requis pour assurer la valeur de défaillance requise. Si pour une heure de temps CPU 20 heures de travail du personnel sont requises, combien d'heures de main d'œuvre sont requises pour obtenir la valeur désirée?

SOLUTION

Avec $\lambda_0 = 0.0055, \nu_0 = 82.99 \approx 83$ on a un taux de défaillance observé de 0.00173 CPU/sec, c'est-à-dire $60*60*12*0.00173=75$ défaillances pour 12 heures mais on veut un taux de défaillance de $1/43200=0.000231$ défaillances par sec. Les défaillances à corriger sont :

$$\Delta\mu = \frac{\nu_0}{\lambda_0} (\lambda_p - \lambda_f) = 15079.79(0.00173 - 0.000231) = 22.6 \approx 23$$

Le temps requis est :

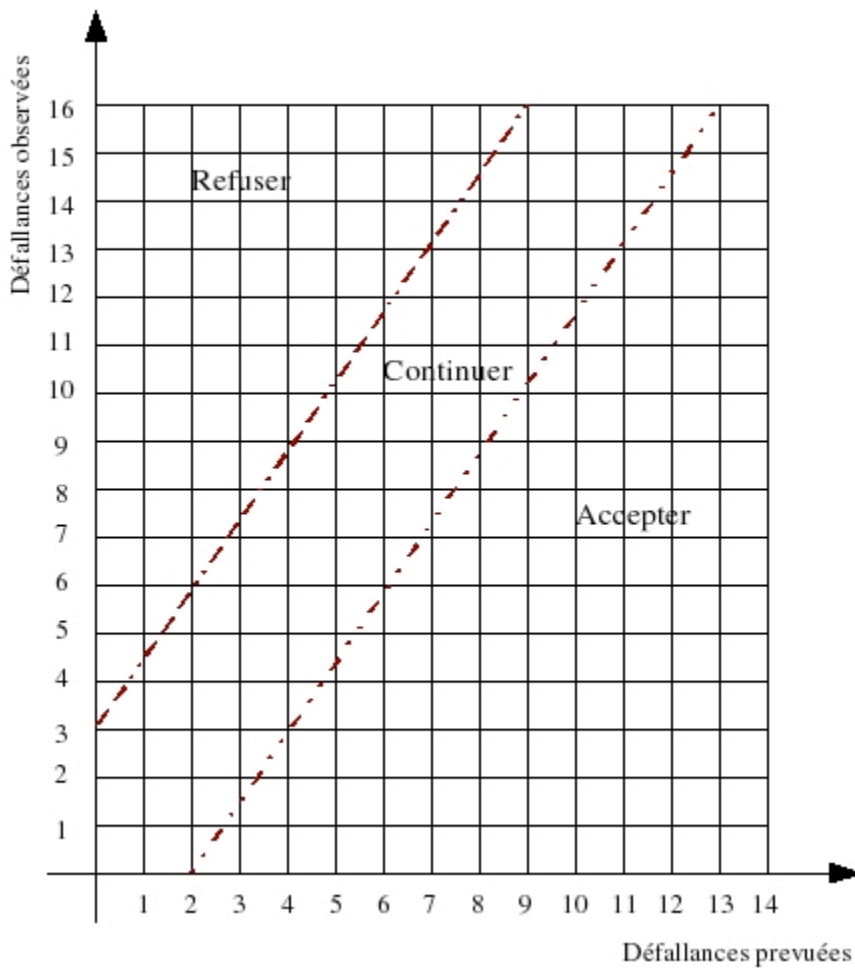
$$\Delta t = \frac{\nu_0}{\lambda_0} \lg(\lambda_p / \lambda_f) = 15079.79 \log(0.00173 / 0.000231) = 30362.5 \approx 30363 \text{ sec}$$

30362 sec sont 8.43 heures de temps CPU donc entre 8 et 9 heures. Pour chaque heure 20 heures de travail sont requises donc $8*20=160$ heures qui équivaut à 1 mois (1 mois et une semaine).

Problème 4 : On a effectué des activités de test sur une base des données, logiciel du 25 KLOC ; on a observé les défaillances :

| Défaillance | Million d'opérations |
|-------------|----------------------|
| 1 | 0.25 |
| 2 | 0.85 |
| 3 | 1.4 |
| 4 | 2.1 |

Q4.1) Si on ne permet pas plus de 2 défaillances par 1 million d'opération, utilisez le graphe de décision suivant pour décider si vous accepter, refuser le logiciel ou continuer de tester.



SOLUTION

| Défaillance | Million d'opérations | Def Prevues | decision |
|-------------|----------------------|-------------|----------|
| 1 | 0.25 | 0.5 | Cont |
| 2 | 0.85 | 1.7 | Cont |
| 3 | 1.4 | 2.8 | Cont |
| 4 | 2.1 | 4.3 | Cont |

Q4.2) Si la prochaine défaillance est observée après 3.15 million d'opération quelle serait votre décision ?

SOLUTION

| Défaillance | Million d'opérations | Def Prevues | decision |
|-------------|----------------------|-------------|------------|
| 1 | 0.25 | 0.5 | Cont |
| 2 | 0.85 | 1.7 | Cont |
| 3 | 1.4 | 2.8 | Cont |
| 4 | 2.1 | 4.3 | Cont |
| 5 | 3.15 | 6.3 | Accepter ! |