Corrigé DS4 - type Mines

Déduction naturelle

① on remarque alors que :

$\mu(\bot) = \phi$

$\mu(A \wedge B) = \mu(A) \cap \mu(B) = \phi$ ssi $\mu(A) = \phi$ ou $\mu(B) = \phi$

$\mu(A \vee B) = \mu(A) \cup \mu(B) = \phi$ ssi $\mu(A) = \phi$ et $\mu(B) = \phi$

$\mu(A \to B) = \overline{\mu(A)^c \cup \mu(B)} = \phi$ ssi $\mu(B) = \phi$ et $\mu(A) = \mathbb{R}$

Ainsi en posant $\mu(A) = \phi$ ssi $v(A) = $ faux

et $\mu(A) = \mathbb{R}$ ssi $v(A) = $ vrai

on retrouve exactement la sémantique booléenne standart.

②

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash (A \vee B) \wedge \neg A}\, r}{\Gamma \vdash A \vee B}\, e\wedge \quad ; \quad \dfrac{\dfrac{\dfrac{}{\Gamma, A \vdash (A \vee B) \wedge \neg A}\, r}{\Gamma, A \vdash A} \quad ; \quad \dfrac{}{\Gamma, A \vdash \neg A}\, r}{\Gamma, A \vdash B}\, e\wedge \quad ; \quad \dfrac{}{\Gamma, B \vdash B}\, r}{\Gamma = (A \vee B) \wedge \neg A \vdash B}\, e\vee$$

$$\dfrac{\Gamma = (A \vee B) \wedge \neg A \vdash B}{\vdash ((A \vee B) \wedge \neg A) \to B}\, i\to$$

③ Comme $\mu(\phi) = \mathbb{R}$ il faut montrer que

$$\mu\left(((A \vee B) \wedge \neg A) \rightarrow B\right) = \mathbb{R}$$

on a: $\mu(A \vee B) = \mu(A) \cup \mu(B)$

et $\mu\left((A \vee B) \wedge \neg A\right) = \left(\mu(A) \cup \mu(B)\right) \cap \left(\overset{\circ}{\overline{\mu(A)^c}}\right)$

or $\overset{\circ}{\overline{\mu(A)^c}} \subset \mu(A)^c$ donc:

$$\mu\left((A \vee B) \wedge \neg A\right) = \phi \cup \mu(B) \cap \overset{\circ}{\overline{\mu(A)^c}}$$
$$= \mu(B) \cap \overset{\circ}{\overline{\mu(A)^c}}$$

et ainsi,

$$\mu\left(((A \vee B) \wedge \neg A) \rightarrow B\right) = \overset{\circ}{\overline{\left(\mu(B) \cap \overset{\circ}{\overline{\mu(A)^c}}\right)^c \cup \mu(B)}}$$

or $\left(\mu(B) \cap \overset{\circ}{\overline{\mu(A)^c}}\right)^c = \mu(B)^c \cup \left(\overset{\circ}{\overline{\mu(A)^c}}\right)^c$

$$\supset \mu(B)^c$$

donc

$$\mu\left(((A \vee B) \wedge \neg A) \rightarrow B\right) \supset \mu(B) \cup \mu(B)^c = \mathbb{R}$$

(4)

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A\to B,A\vdash A}\,r \quad \dfrac{}{A\to B,A\vdash A\to B}\,r}{A\to B,A\vdash B}\,MP}{A\to B,A\vdash \neg A\vee B}\,iv \quad ; \quad \dfrac{\dfrac{}{A\to B,\neg A\vdash \neg A}\,r}{A\to B,\neg A\vdash \neg A\vee B}\,vi}{A\to B\vdash \neg A\vee B}\,ev \quad ; \quad \dfrac{}{A\to B\vdash A\vee \neg A}\,TE$$

$$\dfrac{A\to B\vdash \neg A\vee B}{\vdash (A\to B)\to(\neg A\vee B)}\,i\to$$

(5) on pose $\mu(A)=\mu(B)=\mathbb{R}^*$

$$\mu(A\to B)=\overline{\mu(A)^c \cup \mu(B)}^{\circ} = \mathring{\mathbb{R}} = \mathbb{R}$$

$$\mu(\neg A\vee B)=\overline{\mu(A)^c}^{\circ}\cup\mu(B)=\emptyset\cup\mathbb{R}^*=\mathbb{R}^*$$

donc:

$$\mu\big((A\to B)\to(\neg A\vee B)\big)=\overline{\emptyset\cup\mathbb{R}^*}^{\circ}=\mathbb{R}^*\neq\mathbb{R}$$

Ce séquent n'est donc pas valide pour la sémantique de Heyting.

⑥

* $\dfrac{\Gamma \vdash A \;;\; \Gamma \vdash A \to B}{\Gamma \vdash B} \to e$

on suppose que $\mu(\Gamma) \subset \mu(A)$ et $\mu(\Gamma) \subset \mu(A \to B) = \overset{\circ}{\overline{\mu(A)^c \cup \mu(B)}}$

Ainsi :

$$\mu(\Gamma) \subset \mu(A) \cap \overset{\circ}{\overline{(\mu(A)^c \cup \mu(B))}} = \overset{\circ}{\overline{\mu(A) \cap (\mu(A)^c \cup \mu(B))}} = \overset{\circ}{\overline{\mu(A) \cap \mu(B)}} \subset \mu(B)$$

car $\mu(A) = \overset{\circ}{\mu}(A)$ (ouvert)
et $\overset{\circ}{U} \cap \overset{\circ}{V} = \overset{\circ}{\overline{U \cap V}}$

*

⑦ Induction sur l'arbre de preuve

⑧ On pose
$\mu(A) = \mathbb{R}^*$ alors
$\mu(A \vee \neg A) = \mu(A) \cup \overset{\circ}{\overline{\mu(A)^c}} = \mathbb{R}^* \neq \mathbb{R}$.

Ceci nous donne une formule qui n'est pas valide pour la sémantique de Heyting. Le système intuitionniste est correct pour cette sémantique d'après la question 3 donc cette formule ne peut pas être prouvée en logique intuitionniste car cette correction garantirait alors sa validité - absurde.

⑨

```
int sl_recherche(slliste* l, char* clef) {
    slNoeud* curr = l -> sentinelle;
    slNoeud* save = l -> sentinelle;
    int niveau = l -> niveau_actuel;
    while (niveau >= 0) {
        while (curr != NULL && strcmp(curr -> clef, clef) < 0)
        {
            save = curr;
            curr = curr -> suivant_par_niveau[niveau];
        }
        if (curr == NULL) {
            niveau--;
            curr = save;
        }
        else if (strcmp(curr -> clef, clef) == 0) {
            return (curr -> val);
        }
        else { niveau--;
            curr = save;
        }
    }
    return (-1);
}
```

(10) Si k = MAX_NIVEAU, la probabilité que la taille soit k vaut $\frac{1}{2^k}$ et si $k \in [\![0; MAX\_NIVEAU - 1]\!]$, c'est $\frac{1}{2^{k+1}}$

(11)

```
for (int n = MAX_NIVEAU; n>=0; n--) {
    slNoeud* save = courant;
    while (courant != NULL &&
           strcmp (courant->clef, clef) < 0) {
        save = courant;
        courant = courant->suivant_par_niveau[n];
    }
    courant = save;
    noeuds_a_mettre_a_jour[n] = save;
}
```

(12) on ajoute `liste->niveau_actuel = niveau;`

(13)

```
for(int n=MAX_NIVEAU; n>=0; n--){
    slNoeud* no = noeuds_a_mettre_ajour[n];
    slNoeud* next = no->suivant_par_niveau[n];
    /*no ≠ NULL*/
    no->suivant_par_niveau[n] = nouveau_noeuds;
    nouveau_noeuds->suivant_par_niveau[n] = next;
}
```

(14) Las Vegas.

(15) Cela revient à faire une dichotomie : à chaque niveau on fait une comparaison qui découpe l'ensemble des valeurs possibles en 2 et donc la complexité est en $O(\log_2(n))$.

(16) Soit N_k la va qui représente le nb d'élts au niveau k et n le nb d'élts ds la skip-list.

$$E(N_0) = n \text{ et}$$

$$\forall k \geq 0, \quad E(N_{k+1}) = \frac{1}{2} E(N_k)$$

Ainsi, $\forall k \geq 0$, $E(N_k) = \frac{n}{2^k}$ (rec sur k)

(17) Soit M_n le niveau maximum atteint.

$$E(M_n) = \sum_{k=0}^{+\infty} k \, P_R(M_n = k) = \sum_{k=1}^{+\infty} P_R(M_n \geq k)$$

avec $P_R(M_n \geq k) = 1 - P_R(M_n < k)$

$$= 1 - \left(1 - \frac{1}{2^k}\right)^n$$

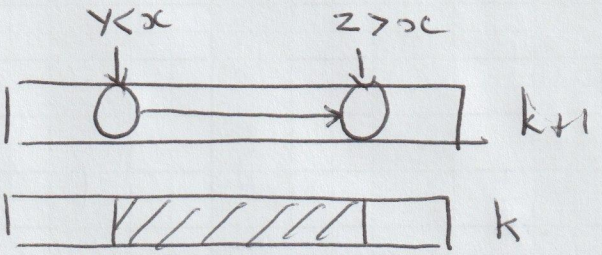
en effet : P_R (un noeud soit de hauteur au Θk ~~exact~~) $= \frac{1}{2^k}$

On obtient donc :

$$E(M_n) = \sum_{k=1}^{+\infty} \left(1 - \left(1 - \frac{1}{2^k}\right)^n\right)$$

(18) Considérons une recherche dans le niveau k. Elle s'arrêtera au pire dès lors qu'on rencontrera un noeud qui est aussi présent au niveau k+1 car l'on parcourt les sommets du niveau k compris entre les deux sommets du niveau k+1 qui nous ont fait descendre de niveau.

Recherche de x :



Chaque sommet du niveau k a une chance sur 2 d'être au niveau k+1 donc le nombre de noeuds parcouru au niveau cas suit une loi géométrique de paramètre $\frac{1}{2}$ dont l'espérance est 2.

On parcourt donc 2 nœuds horizontalement en moyenne

(19) On a une hauteur moyenne de la liste qui est logarithmique par rapport à n et un nombre moyen de comparaisons par étage qui vaut 2

Ainsi, l'espérance de la recherche est en $O(\log(n))$

(20)

$C' \rightarrow CL \mid L$

(21) $V \rightarrow MV' \mid N$ et $V' \rightarrow NV' \mid \varepsilon$

(22) $T \rightarrow \{S\}$

$\rightarrow \{K, S\} \rightarrow \{K, K, S\} \rightarrow \{K, K, K\}$

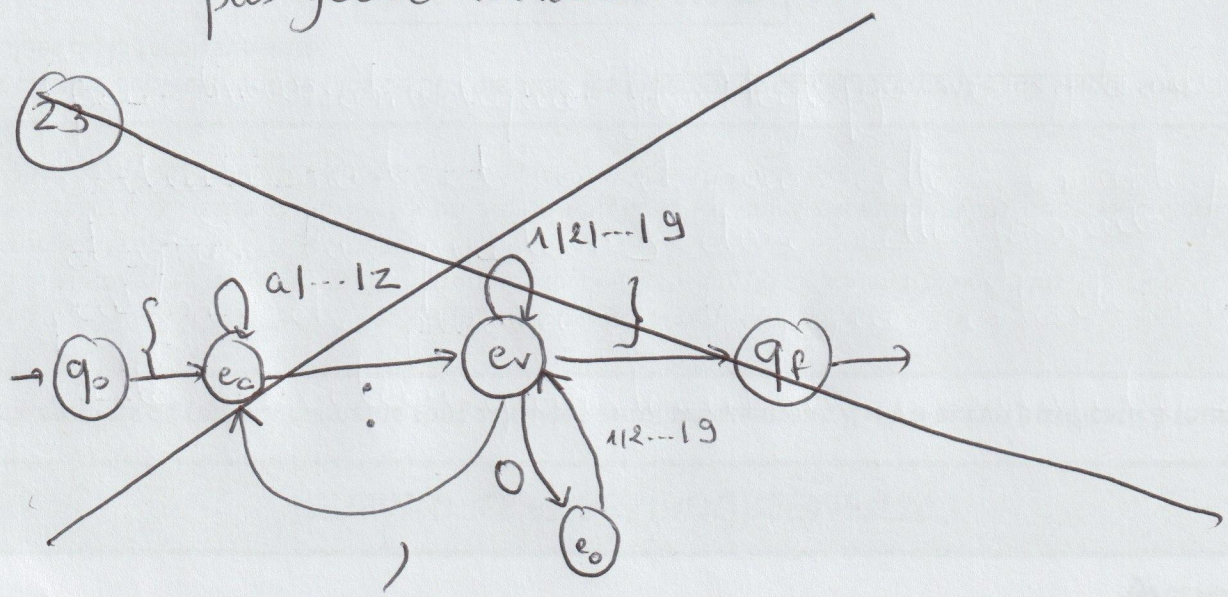
$\rightarrow \{C:V, K, K\}$

$\rightarrow \{C:V, C:V, K\}$

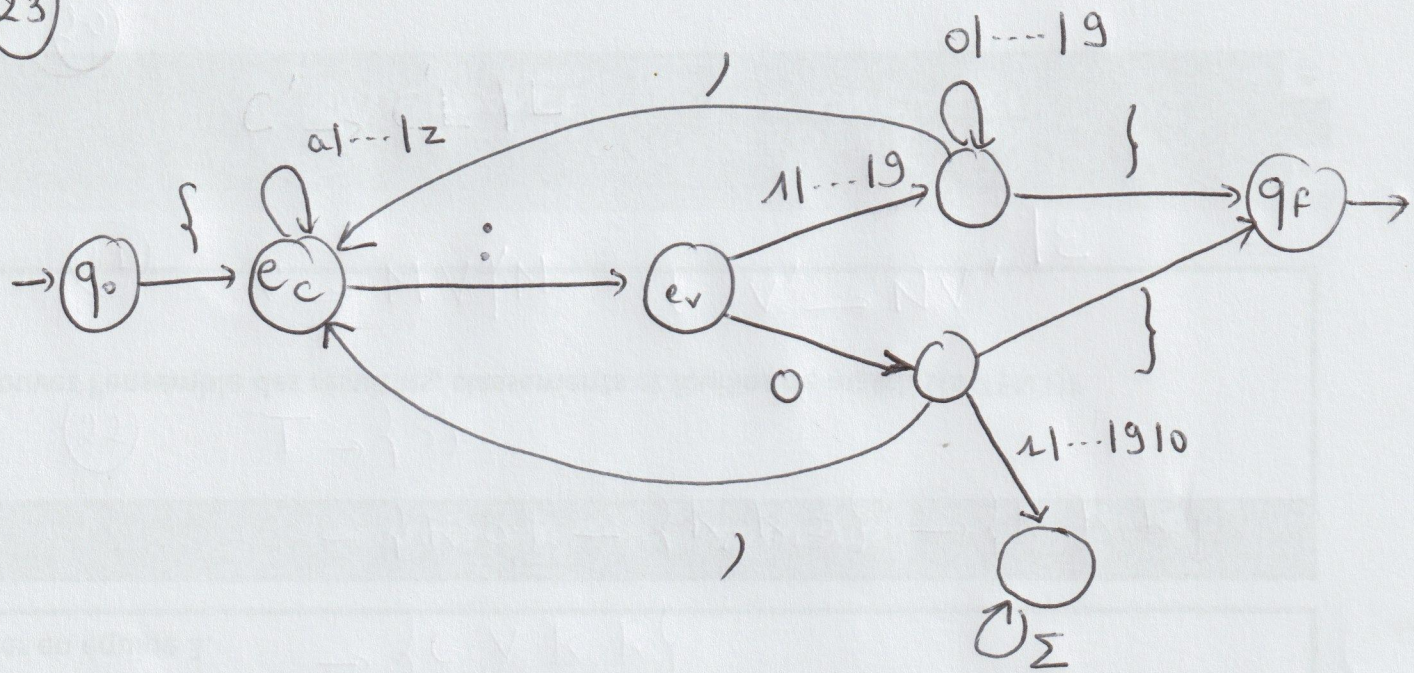
$\rightarrow \{C:V, C:V, C:V\}$

et à partir de C et de V on engendrera id, valeur, ok, 5, 12 et 0 donc le mot est bien généré par la grammaire

Le deuxième mot ne l'est pas car on ne saura pas généré "la valeur" "007"



(23)



(24) ok

(25) $\binom{bba}{bb}\binom{ab}{aa}\binom{bba}{bb}\binom{a}{baa}$

3 2 3 1 fournit une solut° ; c'est bien une instance positive du pb.

(26) La combinaison commence forcément par

$\binom{abc}{ab}$ car c'est le seul domino dont les lettres en haut et en bas coïncident.

1

Aucun domino n'a de motif du bas qui commence par c, on est donc bloqué et l'instance est négative.

(27) Si on a une solut° $(i_1, \dots, i_k)$ alors pour tout entier c, on a $\underbrace{(i_1, \dots, i_k, i_1, \dots, i_k \dots)}_{\text{répété c fois}}$

qui est une autre solution et donc on peut construire une infinité de solut° à partir de l'une d'entre elles.

(28) cours : Si on a A_2 un algo qui décide f_2 alors si $g : E_1 \to E_2$ est la réduction supposée par l'énoncé, on aurait

A_1 sur entrée $x \in E_1$

renvoyer $A_2(g(x))$

qui décide f_1 ce qui est absurde car supposé impossible

(29) $S \to u_i S a_i$ pour chaque $i \in [\![1, n]\!]$.

$S \to u_i a_i$

engendre L_u.

(30) Si PCP est positive alors $L_u \cap L_v \neq \emptyset$

Réciproquement, si $L_u \cap L_v \neq \emptyset$ alors m est engendrée à la fois par une grammaire pour L_u et une pour L_v et alors $m = u_{i_1} \dots u_{i_k} a_{i_1} \dots a_{i_k}$

$= \underbrace{v_{i_1} \dots v_{i_k}}_{\text{les } m \text{ indices}} \overleftrightarrow{a_{i_1} \dots a_{i_k}}$ les m lettres

$(i_1, \dots, i_k)$ fournit alors une solution au problème PCP.

(31) L'équivalence de la q°30 garantit que

$PCP \leq$ inter-vide

$\begin{pmatrix} u_1 \\ \vdots \\ v_1 \end{pmatrix} \dots \begin{pmatrix} u_n \\ \vdots \\ v_n \end{pmatrix} \longrightarrow L_u, L_v$

(32) Cette grammaire engendre $L_u \cup L_v$

(33) Réduct°:

$\begin{pmatrix} u_1 \\ \vdots \\ v_1 \end{pmatrix} \dots \begin{pmatrix} u_n \\ v_n \end{pmatrix} \longrightarrow$ grammaire de la q°32

Si $\binom{u_1}{v_1} \cdots \binom{u_n}{v_n}$ est une instance ≥ 0 de PCP alors la grammaire associée est ambigue car si $(i_1, \ldots, i_k)$ est une solution, on aura $u_{i_1} \cdots u_{i_k} a_{i_k} \cdots a_{i_1}$ et $v_{i_1} \cdots v_{i_k} a_{i_k} \cdots a_{i_1}$ qui sont égaux et qui correspondent à deux arbres $\neq$: l'un commençant par $S \to U$ et l'autre par $S \to V$

Réciproquement, si la grammaire est ambigue, en remarquant que les mots engendrés depuis U (resp V) le sont de manière non ambigüe alors l'ambiguité vient forcément d'un mot de $L_U \cap L_V$ qui est engendré d'une part depuis U et d'autre part depuis V et ainsi le mot $u_{i_1} \cdots u_{i_k} a_{i_k} \cdots a_{i_1}$ ainsi engendré fournit $(i_1, \ldots, i_k)$ comme solut° à l'instance de PCP

(34) C'est le complémentaire de L_u.

(35) $S \to D\beta D\alpha D$ pour chaque $\alpha \in \Sigma$ et $\beta \in \Delta$

$D \to \alpha D \mid \beta D \mid \varepsilon$

(36) ~~[illegible]~~

$S \to u_i S a_i \mid C$ pour chaque $i \in [\![1, n]\!]$

$C \to u_i A a_j$ pour chaque $(i,j) \in [\![1,n]\!]^2$ $i \neq j$

$A \to \varepsilon \mid \underbrace{\alpha A \beta}_{\text{inutile}} \mid \alpha A \mid A\beta$ avec $\alpha \in \Sigma$ et $\beta \in \Delta$

(37) On a vu que $\binom{u_1}{v_1} \cdots \binom{u_n}{v_n}$ est une instance positive de PCP ssi $L_u \cap L_v \neq \emptyset$ i.e ssi ${}^cL_u \cup {}^cL_v \neq \Sigma'^*$

Or on vient de voir comment construire une grammaire pour ${}^cL_u$ en faisant une grammaire qui engendre $L_u' \cup (\Sigma^*\Delta^*) \cup \{\varepsilon\}$ ce qui est faisable d'après les q° 35 et 36 et en utilisant la stabilité par union.
La stabilité par union permet aussi d'obtenir une grammaire pour ${}^cL_u \cup {}^cL_v$ qui sera universelle ssi l'instance de PCP est négative

(38) Soit $S \to \alpha S | \varepsilon \quad \forall \alpha \in \Sigma$

alors le langange engendré est Σ^*

Ainsi

$$Univ \leqslant Egal$$

$G \to (G, G_u)$ où G_u est la grammaire définie ci-dessus engendrant Σ^*.

On a bien que $G \in Univ$ ssi $(G, G_u) \in Egal$

Ainsi, $Univ \leq Egal$ et $Univ$ est indécidable car PCP l'est d'après la q°37 donc $Egal$ est aussi indécidable.