Algorithme DPLL

Fonction DPLL(ϕ) = `retourne VRAI ssi` ϕ `est satisfaisable`

1. si $\phi = \top$ alors retourner VRAI (la formule est satisfaisable)
2. sinon si $\phi = \bot$ alors retourner FAUX (la formule n'est pas satisfaisable)
3. sinon si ϕ contient une clause unitaire x, alors retourner DPLL($\phi[x/1]$)
4. sinon si ϕ contient une clause unitaire $\neg x$, alors retourner DPLL($\phi[x/0]$)
5. sinon si ϕ contient une proposition x de polarité toujours positive, alors retourner DPLL($\phi[x/1]$)
6. sinon si ϕ contient une proposition x de polarité toujours négative, alors retourner DPLL($\phi[x/0]$)
7. sinon dans tous les autres cas, choisir une proposition x au hasard et retourner (DPLL($\phi[x/0]$) ou DPLL($\phi[x/1]$))

Exercice 1 On considére l'algorithme DPLL ci-dessus avec évaluation de la disjonction à l'étape 7 en court-circuit. Toujours à l'étape 7, si vous devez énumérer, faites-le en choisissant les variables dans l'ordre alphabétique x puis y puis z puis t.
Tracez précisément l'exécution de DPLL sur la formules suivantes et donnez la première solution trouvée par l'algorithme :

1. $(\neg x \vee y) \wedge x \wedge (\neg y \vee z)$
2. $(x \vee y) \wedge \neg y \wedge (\neg x \vee y \vee \neg z)$
3. $x \wedge \neg x$
4. $(x \vee y) \wedge (y \vee z) \wedge (\neg z \vee \neg t)$
5. $(\neg x \vee y) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (\neg y \vee z)$
6. $(\neg y \vee x \vee z) \wedge (\neg x \vee z) \wedge (\neg x \vee \neg z) \wedge (y \vee \neg x \vee \neg z)$
7. $(\neg x \vee \neg y) \wedge (x \vee y) \wedge (\neg x \vee \neg z) \wedge (x \vee z) \wedge (\neg y \vee \neg z) \wedge (y \vee z)$

Exercice 2 Montrez que l'algorithme DPLL ci-dessus :

1. termine pour toute entrée ;
2. est partiellement correct relativement à sa spécification.