Q4 2.1 What are Equational theories?
What are Inductive theories?

le but est de prouver une propriété à l'aide des axiomes et des règles

Contexte :
- 2 axioms
- prouver d'autres propriétés découlant de ces axiomes (comment procéder?)
-> Créer des preuve équationelle utilisant uniquement les 3 domaines de propriété :
- Propriété de variables : la substitution vrai pour x, vrai pour x=v (v une valeur)),
- Propriété de fonction : la substitutivité (x=y->f(x)=f(y)) pour les fonctions
- Propriété d'équation : la relation d'équivalence (=) : symétrie, transitivité, reflexivité
-> créer des théorèmes (équationels)

Définition inductive de la théorie équationelle :
Soit une spécification algébrique Spec=<Σ,X,AX>, Σ=<S,OP> et t,t',ti $\in (T_{\Sigma,X})_s$
La théorie Th(Spec) est alors :

Axiomes : t=t' ∈ Ax => t = t' ∈ Th(Spec)
Reflexivité : $\forall$t ∈ $(T_{\Sigma,X})_s$, t = t ∈ Th(Spec)
Symmetrie : t = t' ∈ Th(Spec) => t' = t ∈ Th(Spec)
Transitivité : t = t' ∈ Th(Spec) AND t' = t'' ∈ Th(Spec) => t = t'' ∈ Th(Spec)
Substitutivité : $\forall$f ∈ OP, t1 = t1'∈ Th(Spec) AND .. AND tn = tn'∈ Th(Spec)
=> f (t1, t2, ..., tn) = f (t1', t2', ..., tn') ∈ Th(Spec)
Substitution : x ∈ Xs , u ∈ $(T_{\Sigma,X})_s$, t = t' ∈ Th(Spec) => t[u/x] = t'[u/x] ∈ Th(Spec)
(x variable, remplace x par u)

a=>b :
Cut : Cond1 AND u = u' AND Cond2 => t = t' ∈ Th(Spec)
and Cond => u = u' ∈ Th(Spec)
then Cond1 ^ Cond ^ Cond2 => t = t' ∈ Th(Spec)

Exemple de preuve sur les naturels :
Axiomes : (1) 0+x=x et (2) Succ(x)+y=Succ(x+y)
À prouver : succ(0)+succ(succ(0))=succ(succ(succ(0)))
étape 1. démarrer de l'axiome 2 et substituer: x->0, y->succ(succ(0)) -> $S(0)+S(S(0)) = S(S(S(0)))$
->(3) Succ(0)+succ(succ(0))=Succ(0+succ(succ(0)))
étape 2. Axiome 1 et substitue x -> succ(succ(0)) -> $S(S(0))+0 = S(S(0))$
-> (4) 0+succ(succ(0))=succ(succ(0))
étape 3. Substitutivité de succ sur (4) -> on applique succ() aux 2 cotes de l equation
-> (5) succ(0+succ(succ(0)))=succ(succ(succ(0)))
étape 4. transitivité entre (3) et (5)
-> Succ(0)+succ(succ(0))=succ(succ(succ(0))) -> OK

Validité et complétude de la théorie equationelle. Si 2 termes sont égaux dans la théorie, alors pour tous modèle(sémantique), cette propriété est vraie:
Soit la spécification Spec=<Σ,X,AX>, $\forall$ t1,t2 ∈ $T_{\Sigma,X}$
Alors t1=t2 ∈ Th(Spec) <=> $\forall$ M ∈ Mod(Spec), M|-t1=t2
prouvé dans la théorie <-> prouvé dans le modèle, ->validité, <-complétude
Donc t1=t2 est valide pour toutes les implémentations I

on peut montrer que :0-0 = 0mais ce n est pas suffisant pour dire que x-x = 0du co

Théorie inductive :
Ajouter une nouvelle règle -> permet de trouver plus de théorèmes à partir des axiomes.
Nouvelle règle :
Soit G une formule avec x une variable. Si $\forall t : G[t/x] \in Th(Spec) \Rightarrow \forall t : G \in Th_{Ind}(Spec)$
Ex: $G(x)=(x-x=0)$: $(0-0=0) \in Th(Spec)$, $(Succ(0)-Succ(0)=0) \in Th(Spec)$, etc.
-> Preuve inductive : cas de base, hypothèse (preuve pour x) -> prouver pour succ(x) :
induction : $x \in (X_S \cap (Var(t) \cup Var(t')))$, $\bigwedge_{v_i \in (T_{\Sigma,X})_s} (t=t')[v_i/x] \in Th_{Ind}(Spec) \Rightarrow t=t' \in Th_{Ind}(Spec)$

Exemple : $(x-x=0) \in Th_{Ind}(Spec)$:
Axiomes (Specification):
(1) x-0=x
(2) Succ(x)-0=succ(x)
(3) Succ(x)-Succ(y)=x-y

Cas de base : 0-0=0 Substitution x=0 sur (1)
Hypothèse : (4) t-t=0 à prouver : succ(t)-succ(t)=0
substitution x=t, y=t sur (3) -> (5) succ(t)-succ(t)=t-t

$\rightarrow S(t)-S(t)=t-t=0$

hypothèse

transitivité entre (4) et (5) -> succ(t)-succ(t)=0 OK
->Vrai pour tout x -> $(x-x=0) \in ThInd(Spec)$

Validité et complétude de la théorie inductive.
Soit la spécification Spec=<Σ,X,AX>, $\forall$ t1,t2 $\in T_{\Sigma,X}$
Alors $t1=t2 \in Th_{Ind}(Spec) \Leftrightarrow \forall M \in Mod_{Gen}(Spec), M|\text{-}t1=t2$
prouvé dans la théorie <-> prouvé dans le modèle, ->validité, <-complétude
La théorie n'est parfois pas complète car il peut y avoir des valeurs non accessibles par le générateur (non prouvées, ex: Boolean en C->seulement 0 et 1, valeurs >1 considérées comme true=1)
Donc Tout ce qu'on peut calculer doit être accessible par le générateur, sinon le modèle n'est pas correct

éventuellement :
x+y=y+x :
cas de base pour x: 0+y=y+0
->induction sur y
cas de base pour y: 0+0=0+0
etc.