

Logique du premier ordre

On considère un langage avec des variables et les quantificateurs « pour tout » et « il existe ». On enrichit ce langage avec des prédicats et des fonctions.

1. Motivation

1.1. Quantificateurs

La logique Vrai/Faux des chapitres précédents est au final assez limitée et ne permet pas d'exprimer les phrases naturelles d'un cours de mathématiques. Dans ce chapitre, on considère un langage plus puissant : la logique du premier ordre. Elle permet d'écrire les phrases suivantes :

- « $\forall n \in \mathbb{Z} \quad n^2 \geq 0$ » : « pour tout entier relatif, son carré est positif ».
- « $\exists z \in \mathbb{C} \quad z^2 = -1$ » : « il existe un nombre complexe dont le carré vaut -1 ».
- « $\forall g \in G \quad \exists g' \in G \quad g \circ g' = e$ » : « tout élément de G admet un inverse », pour un groupe $(G, \circ)$ d'élément neutre e .

Avec ce langage, on peut définir des notions compliquées avec des nuances subtiles. Par exemple, soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction.

- La continuité sur $\mathbb{R}$ se définit par :

$$\forall x \in \mathbb{R} \quad \forall \varepsilon > 0 \quad \exists \delta > 0 \quad \forall y \in \mathbb{R} \quad |x - y| < \delta \rightarrow |f(x) - f(y)| < \varepsilon$$

- Alors que la continuité uniforme se définit par :

$$\forall \varepsilon > 0 \quad \exists \delta > 0 \quad \forall x \in \mathbb{R} \quad \forall y \in \mathbb{R} \quad |x - y| < \delta \rightarrow |f(x) - f(y)| < \varepsilon$$

1.2. Nouveautés

La principale nouveauté, c'est bien sûr l'apparition des quantificateurs « $\forall$ » et « $\exists$ ». En une phrase « $\forall n \in \mathbb{N} \quad n^2 \geq 0$ », on exprime une infinité de vérités. Avec la logique Vrai/Faux on ne saurait pas prouver une telle affirmation, car on devrait considérer le cas $n = 0$, puis $n = 1$, etc., donc avec un nombre infini d'étapes. Ici on considère la phrase « $\forall n \in \mathbb{N} \quad n^2 \geq 0$ » comme une seule affirmation à prouver.

Le second intérêt de la logique du premier ordre est qu'elle s'applique à des situations variées. En logique propositionnelle, on considèrerait des formules et la satisfaction consistait à remplacer les variables par des valeurs V/F. Ici on souhaite toujours décider si une formule est vraie ou fausse, mais les variables correspondent à des objets plus généraux. Par exemple considérons la formule :

$$(\forall x \quad P(x)) \vee (\exists x \quad \neg P(x))$$

Elle s'applique avec des x dans un ensemble quelconque, mais aussi avec n'importe quel prédicat $P(x)$. En français, elle signifie « une propriété est vraie pour tous les éléments, ou bien il existe un élément pour lequel elle n'est pas vraie ». Voici ce qui donne cette formule dans trois structures différentes :

- $(\forall x \in \mathbb{R} \quad x^2 \neq 0) \vee (\exists x \in \mathbb{R} \quad x = 0)$

- $(\forall n \in \mathbb{N} \quad n + 1 > n) \vee (\exists n \in \mathbb{N} \quad n + 1 \leq n)$
- $(\forall g \in G \quad g \circ g = e) \vee (\exists g \in G \quad \neg(g \circ g = e))$, dans un groupe G .

Les langages du premier ordre s'adaptent à une grande variété de structures (ensembles, groupes, corps, espaces vectoriels...) et aux démonstrations qui vont avec.

1.3. Limitations

La logique du premier ordre est puissante, cependant il y a certaines choses qu'il n'est pas possible d'exprimer avec ces langages. Le plus flagrant est qu'il n'est pas possible de trouver une formule pour décider si un ensemble est fini ou non (ce qui entraîne par exemple qu'on ne peut pas définir la notion de dimension d'un espace vectoriel). Travailler en logique du premier ordre signifie que l'on considère des éléments dans un ensemble E , noté « $x \in E$ », mais on ne peut pas travailler avec l'ensemble des parties $A \subset E$. Ainsi, la phrase « toute partie non vide admet un plus petit élément » (qui est la définition d'un ensemble *bien ordonné*) est une phrase exprimée dans la *logique du second ordre* car elle porte sur les sous-ensembles et pas sur les éléments. En conclusion, la logique du premier ordre décrit bien les propriétés locales des objets, mais moins bien les propriétés globales liées à l'infini.

2. Langage

Il faut comprendre un langage de la logique du premier ordre comme un cadre général commun qu'on spécialisera ensuite à chaque situation (les entiers, les réels, les groupes...).

2.1. Éléments de langage

Les formules de la logique du premier ordre sont construites à partir de langages composés des éléments suivants.

Symboles logiques

- **variables** : $x, y, z, x_1, x_2, \dots$
- parenthèses : (et)
- connecteurs : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- **quantificateurs** : $\forall$ (pour tout), $\exists$ (il existe)

Symboles non logiques

- **constantes** : $a, b, c, c_1, c_2, \dots$
- **prédicats** : $P(x), P(x, y), \dots, P(x_1, \dots, x_n)$; dont l'égalité $x = y$
- **fonctions** : $f(x), f(x, y), \dots, f(x_1, \dots, x_n)$

Nous revenons en détail ci-dessous sur les nouvelles notions. Dans un premier temps on va considérer des langages sans fonctions ; on les ajoutera à la fin de ce chapitre.

2.2. Symboles logiques

Le langage est beaucoup plus riche que celui de la logique Vrai/Faux. On retrouve bien sûr les connecteurs logiques (et les parenthèses) qui joueront ici le même rôle qu'en logique propositionnelle. Détaillons les nouveautés.

- **Quantificateur universel** $\forall$. C'est un A inversé, introduit par Gerhard Gentzen en 1935, comme initiale de *all* ou *alle*. Il se lit « pour tout » ou « quel que soit ».
- **Quantificateur existentiel** $\exists$. C'est un E inversé, introduit par Giuseppe Peano à la fin du XIX^e siècle. Il se lit « il existe » ou « il y a au moins un ».
- **Variables**. Elles servent à parler des objets manipulés sans les nommer individuellement. Ainsi « $\forall x P(x)$ » ou « $\exists x P(x)$ » va s'appliquer à tous les x (ou à un x), mais il n'y a pas vraiment un

objet nommé x . On parle de **variable liée** (ou *variable muette*). Ainsi « $\forall y P(y)$ » aurait la même signification.

Note : pour les variables, on réserve les lettres de la fin de l'alphabet x, y, z , avec des indices si besoin.

2.3. Symboles non logiques

C'est aussi une nouveauté de la logique du premier ordre. On veut des constructions et des théories qui s'appliquent à de nombreuses situations. Les constantes, les prédicats et les fonctions sont des éléments comme des coquilles vides, qu'il faudra préciser à chaque application.

- **Constantes.** Notées $a, b, c, c_1, c_2, \dots$ ce sont des symboles du langage qui vont servir à distinguer certains éléments. On peut par exemple définir un langage adapté à la notion de groupe avec une constante c , qui jouera le rôle de l'élément neutre e , $0, 1, \dots$ selon le groupe considéré. Lors de la définition du langage, on décide du nom et du nombre des constantes.
- **Prédicats.** Un prédicat est un peu comme une super-proposition de la logique Vrai/Faux, cela renvoie une valeur V/F, mais en fonction de la valeur d'un ou plusieurs paramètres. Leur symbole dans le langage sera souvent P, Q ou R .
 - Prédicat unaire P , noté $P(x)$: « est-ce que x a la propriété P ? » (cela peut être vrai ou faux selon la valeur affectée à x).
Par exemple un tel prédicat pourra désigner « $x^2 \geq x$ » ou « $x^2 = 1$ » ou « $x^5 + x^3 + 1 = 0$ » ... Cela peut être vrai ou faux en fonction de x .
 - Prédicat binaire P , noté $P(x, y)$: « est-ce que (x, y) a la propriété P ? », mais qui peut aussi être compris comme « est-ce que x est en relation P avec y ? ».
Par exemple un prédicat binaire peut désigner « $x < y$ » ou « $x^2 + y^2 = 1$ » ...
 - L'**égalité** « $x = y$ » est un prédicat binaire spécial. Sauf mention contraire (et rare) on supposera toujours que l'égalité est présente dans le langage.
 - Prédicat n -aire $P(x_1, \dots, x_n)$ (on parle aussi de prédicat à n places).
Encore une fois, ce qu'est concrètement chaque prédicat devra être précisé dans chaque structure. Au niveau du langage c'est juste un symbole P , mais on se permettra de le noter $P(x)$ ou $P(x_1, \dots, x_n)$ afin de clarifier le nombre de places.
- **Fonctions.** À la fin du chapitre on ajoutera des fonctions au langage. Par exemple $f(x)$ (qui peut représenter $f(x) = 2x + 3$ ou $f(x) = 1/x, \dots$), $f(x, y)$ (pour $f(x, y) = x + y$ ou $f(x, y) = x \times y, \dots$). Les fonctions ajoutent une certaine technicité à la définition de la notion de formule, donc pour l'instant nos langages n'auront pas de fonctions.

3. Formules

3.1. Définitions

Comme le langage est riche, il y a un vocabulaire spécifique qu'il faut bien maîtriser.

Soit $\mathcal{L}$ un langage sans fonctions.

Définition.

On appelle **terme** d'un langage les variables ou les constantes.

Remarque : on modifiera cette définition pour les langages avec fonctions (voir en fin de chapitre).

Définition.

On appelle **formule atomique** un mot de la forme $P(t_1, \dots, t_n)$ où P est un prédicat n -aire et $t_1, \dots, t_n$ sont des termes.

Définition.

L'ensemble des **formules** $\text{Form}(\mathcal{L})$ du langage est construit selon les règles suivantes :

- les formules atomiques sont des formules,
- si $\mathcal{P}$ et $\mathcal{Q}$ sont des formules alors $\neg\mathcal{P}$, $(\mathcal{P} \wedge \mathcal{Q})$, $(\mathcal{P} \vee \mathcal{Q})$, $(\mathcal{P} \rightarrow \mathcal{Q})$, $(\mathcal{P} \leftrightarrow \mathcal{Q})$ sont aussi des formules,
- si x est une variable et $\mathcal{P}$ est une formule alors

$$\forall x \mathcal{P} \quad \text{et} \quad \exists x \mathcal{P}$$

sont aussi des formules,

- aucun autre mot n'est une formule.

3.2. Remarques

- L'écriture « $\forall x \mathcal{P}$ » est un mot du langage, et dans ce langage x est une variable abstraite, pas un élément d'un ensemble. Lorsque l'on interprétera ultérieurement la formule, on définira un domaine E , par exemple $\mathbb{N}$, $\mathbb{R}$, G , ... et alors la formule deviendra « $\forall x \in \mathbb{N} \mathcal{P}$ » ou « $\forall x \in \mathbb{R} \mathcal{P}$ », etc.
- Le cas intéressant est bien sûr celui où la formule $\mathcal{P}$ dépend effectivement de la variable x . On se permettra souvent de noter une formule $\mathcal{P}(x)$ pour montrer sa dépendance en la variable x . Ainsi on écrira plutôt « $\forall x \mathcal{P}(x)$ » ou « $\forall x \exists y \mathcal{P}(x, y)$ »...
- Comme d'habitude on s'autorisera à ajouter et supprimer des parenthèses pour clarifier les formules. Par exemple on écrira : « $(\forall x \mathcal{P}) \rightarrow (\exists y \mathcal{Q})$ » au lieu de la formule officielle « $(\forall x \mathcal{P} \rightarrow \exists y \mathcal{Q})$ »

La définition de formule donnée ci-dessus était la définition inductive (« par le haut ») de $\text{Form}(\mathcal{L})$. On peut aussi en donner une définition récursive (« par le bas »).

On définit $\text{Form}(\mathcal{L}) = \bigcup_{n \geq 0} \text{Form}_n(\mathcal{L})$, avec :

$\text{Form}_0(\mathcal{L})$ est l'ensemble des formules atomiques

puis, pour $n \geq 0$:

$$\begin{aligned} \text{Form}_{n+1}(\mathcal{L}) = \text{Form}_n(\mathcal{L}) & \\ & \cup \{ \neg \mathcal{P} \mid \mathcal{P} \in \text{Form}_n(\mathcal{L}) \} \\ & \cup \{ (\mathcal{P} \wedge \mathcal{Q}) \mid \mathcal{P}, \mathcal{Q} \in \text{Form}_n(\mathcal{L}) \} \\ & \cup \dots \\ & \cup \{ \forall x_k \mathcal{P} \mid x_k \text{ variable}, \mathcal{P} \in \text{Form}_n(\mathcal{L}) \} \\ & \cup \{ \exists x_k \mathcal{P} \mid x_k \text{ variable}, \mathcal{P} \in \text{Form}_n(\mathcal{L}) \} \end{aligned}$$

où l'union porte bien sûr aussi sur les connecteurs $\vee$, $\rightarrow$, $\leftrightarrow$ et où on a noté $x_1, x_2, \dots$ l'ensemble des variables de $\mathcal{L}$.

4. Exemples

Prenons un peu d'avance sur le prochain chapitre avec des exemples de structures afin de mieux comprendre les notions précédentes.

4.1. Avec des entiers

On souhaite modéliser les entiers naturels avec un zéro et une relation d'ordre. On définit un langage $\mathcal{L} = (c, P)$ avec une constante c et un prédicat binaire P , noté aussi $P(x, y)$ (il y a aussi par défaut le prédicat d'égalité $x = y$). En toute rigueur on devrait aussi préciser le nom des variables, mais on utilisera la convention de nommage $x, y, \dots$ partout.

Voici une formule du langage :

$$\forall x \exists y \ P(x, y)$$

Pour pouvoir utiliser ce langage, on précise un domaine, par exemple ici $E = \mathbb{N}$. Ensuite il faut définir tous les symboles non logiques :

- la constante : $c = 0$,
- le prédicat : $P(x, y) : \ll x < y \gg$.

Les symboles logiques, eux, ne doivent pas être spécifiés : la variable x reste une variable, le connecteur $\wedge$ reste le ET, etc. On a ainsi défini une **structure**, notée $\mathcal{S}$. Dans cette structure, la formule précédente s'interprète en :

$$\forall x \in \mathbb{N} \exists y \in \mathbb{N} \ x < y,$$

c'est-à-dire « pour tout entier, il existe un entier encore plus grand ».

Pour le même langage $\mathcal{L}$, on peut faire d'autres choix de structure et la même formule s'interprète différemment.

Voici quelques exemples pour illustrer le vocabulaire lié aux formules avec à chaque fois un mot du langage et son interprétation dans la structure.

Commençons par les termes : il n'y a que la constante c et les variables $x, y, \dots$. Dans la structure on remplace c par 0.

Terme dans $\mathcal{L}$	Interprétation dans $\mathcal{S}$
c	0
x	x

Les formules atomiques sont formées par un prédicat. Ici, seul le prédicat P est défini et est interprété par $x < y$ dans la structure. On a en plus toujours le prédicat d'égalité « $x = y$ ».

Formule atomique dans $\mathcal{L}$	Interprétation dans $\mathcal{S}$
$P(c, x)$	$0 < x$
$P(c, c)$	$0 < 0$
$P(x, y)$	$x < y$
$P(y, x)$	$y < x$
$x = c$	$x = 0$

Pour les formules, on autorise les connecteurs et les quantificateurs. Une formule « $\forall x \ P$ » s'interprète dans la structure en « $\forall x \in \mathbb{N} \ P$ ».

Formule dans $\mathcal{L}$	Interprétation dans $\mathcal{S}$
$\forall x \ P(c, x)$	$\forall x \in \mathbb{N} \ 0 < x$
$\exists y \ x = y$	$\exists y \in \mathbb{N} \ x = y$
$\forall x \ P(c, x) \vee (x = c)$	$\forall x \in \mathbb{N} \ (0 < x) \vee (x = 0)$
$\forall x \forall y \ P(x, y) \rightarrow (\exists z \ P(x, z) \wedge P(z, y))$	$\forall x \in \mathbb{N} \forall y \in \mathbb{N} \ (x < y) \rightarrow (\exists z \in \mathbb{N} \ (x < z) \wedge (z < y))$

La dernière formule s'interprète avec les entiers sous la forme « entre deux entiers distincts, on peut trouver un entier strictement compris entre les deux ». C'est clairement faux (par exemple, il n'y a aucun entier strictement compris entre 7 et 8). Ce serait vrai pour une autre structure où le domaine est l'ensemble des réels $\mathbb{R}$ (entre deux réels distincts, il existe même une infinité de réels).

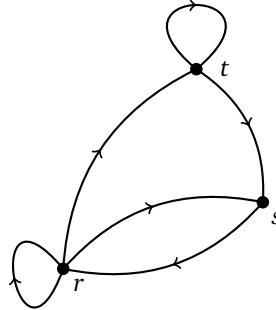
4.2. Avec des graphes

On veut modéliser des graphes (comme par exemple sur le dessin ci-dessous). On définit un langage $\mathcal{L}$ avec le prédicat binaire R (et aucune constante).

Dans ce langage, les seuls termes sont les variables, x et y par exemple. Les formules atomiques sont $R(x, x)$, $R(x, y)$, $R(y, x)$, $R(y, y)$ et aussi $x = x$, $x = y$, $y = x$, $y = y$. Voici un exemple de formule :

$$\forall x \exists y R(x, y)$$

Nous avons notre langage. Comment modéliser un graphe ? Voici le graphe orienté que l'on souhaite modéliser :



Il y a trois sommets, on va donc définir une structure dont le domaine est l'ensemble des trois éléments $E = \{r, s, t\}$. Dans la structure, c'est le prédicat qui décrit les arêtes : on interprète le prédicat R en disant que $R(x, y)$ est vrai s'il existe une arête allant de x vers y (on dit que x est en relation avec y). Ainsi pour notre exemple, on définit le prédicat dans la structure par :

$$R(r, s) = V, \quad R(s, t) = F, \quad R(t, s) = V, \dots$$

La formule précédente du langage $\mathcal{L}$ s'interprète dans cette structure comme :

$$\forall x \in \{r, s, t\} \exists y \in \{r, s, t\} R(x, y)$$

Autrement dit, « pour tout sommet x , il existe un sommet y pour lequel on a une arête de x vers y », en d'autres termes « de tout sommet part au moins une arête », ce qui est vrai pour le graphe considéré. Voici des phrases du langage. À vous de déterminer ce qu'elles signifient pour les graphes et si, pour l'exemple de graphe ci-dessus, elles sont vraies ou fausses.

- $\forall x \forall y R(x, y)$
- $\exists x \exists y R(x, y)$
- $\exists x \forall y R(x, y)$
- $\forall x \forall y R(x, y) \vee R(y, x)$
- $\exists x \exists y R(x, y) \wedge R(y, x)$

4.3. Il existe exactement un, il existe deux

- « $\exists x P(x)$ » demande l'existence d'au moins un x pour lequel $P(x)$ est vérifié. Il peut y en avoir un ou plusieurs.
- Voici comment exprimer qu'il existe exactement un élément qui vérifie $P(x)$:

$$(\exists x P(x)) \wedge (\forall x \forall y (P(x) \wedge P(y) \rightarrow x = y))$$

On exige d'une part l'existence d'un élément, et d'autre part s'il en existe un autre, c'est le même. On n'utilisera pas la notation « $\exists!$ ».

- Voici comment exprimer qu'il existe au moins deux éléments :

$$\exists x \exists y (x \neq y) \wedge P(x) \wedge P(y)$$

On note $\neg(x = y)$ par $x \neq y$.

5. Langage avec fonctions

5.1. Fonctions

Une **fonction** f prend en entrée un ou plusieurs objets et renvoie en sortie un objet. Il ne faut pas confondre les fonctions avec les prédicats : les prédicats prennent aussi des objets en entrée mais renvoient une valeur V/F.

Voici des exemples d'interprétation de fonctions :

- $f(x) = x^2$, fonction à une variable (ici pour les réels),
- $f(x, y) = x \% y$, reste de la division euclidienne de x par y (ici pour les entiers).

Par contre $P(x) : \langle x = 0 \rangle$ et $Q(x, y) : \langle x < y \rangle$ sont des prédicats, pas des fonctions.

Dorénavant, on s'autorise l'ajout de fonctions n -aires (ou à n places) comme symboles non logiques du langage. Comme pour les prédicats, on notera souvent abusivement les fonctions par $f(x)$, $f(x, y)$, $f(x_1, \dots, x_n)$, alors que le symbole du langage est bien seulement f .

5.2. Formules

Avec l'introduction des fonctions, il faut adapter notre définition de la notion de formule.

Soit $\mathcal{L}$ un langage (avec ou sans fonctions).

Définition.

On appelle **terme** d'un langage :

- les variables et les constantes,
- et si $t_1, \dots, t_n$ sont des termes et f un symbole de fonction n -aire, alors $f(t_1, \dots, t_n)$ est aussi un terme.

Noter que la définition est récursive.

La définition de **formule atomique** est inchangée : c'est une expression de la forme $P(t_1, \dots, t_n)$ où P est un prédicat n -aire et $t_1, \dots, t_n$ sont des termes (avec notre nouvelle définition).

La définition de **formule** est inchangée, c'est la même construction à l'aide des formules atomiques, des connecteurs, des quantificateurs (sauf que bien sûr les termes et les formules atomiques peuvent maintenant contenir des fonctions).

5.3. Exemples

Soit $\mathcal{L}$ le langage formé par :

- des constantes a, b ,
- un prédicat binaire P , noté (abusivement) $x < y$,
- une fonction unaire f , une fonction binaire g .
- Exemples de termes :

$$a, \quad y, \quad f(x), \quad g(a, y), \quad g(f(x), b), \quad f(g(y, x)), \quad \dots$$

- Exemples de formules atomiques :

$$x < a, \quad f(x) < f(y), \quad g(a, f(b)) < x, \quad \dots$$

- Exemples de formules :

$$\neg(x < y), \quad (f(x) < a) \wedge (f(y) < b), \quad \forall x \, f(x) < b, \quad \exists x \, g(x, y) = a,$$

$$\forall x \, (f(x) < a) \rightarrow (\exists y \, g(x, y) < b)$$

Comme auparavant, on anticipe sur la notion de structure afin de rendre concrètes ces fonctions. On peut par exemple se placer sur l'ensemble des réels $\mathbb{R}$ et considérer l'interprétation suivante des symboles non logiques :

- $a = 0, b = 1,$
- $P(x, y)$ désigne l'inégalité stricte usuelle $x < y,$
- $f(x) = x^2$ et $g(x, y) = x + y.$
- Exemples de termes dans cette interprétation :

$$x + 1, \quad y^2, \quad x^2 + y, \quad (x + y)^2, \quad x + (y^2 + 1)^2, \quad \dots$$

- Exemples de formules atomiques dans cette interprétation :

$$x < 0, \quad x + y < x^2, \quad x^2 + y^2 = 1, \quad \dots$$

- Exemples de formules dans cette interprétation :

$$(x^2 + y^2 < 1) \rightarrow ((x < 1) \wedge (y < 1)), \quad \forall x \quad x^2 < x + y, \quad (y < 1) \rightarrow (\exists x \quad x^2 = y), \quad \dots$$

Exercices

Langage, Formules, Exemples

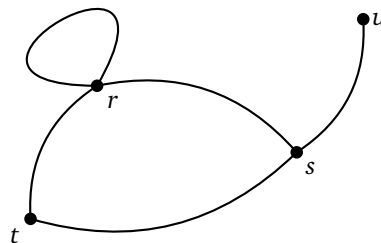
Exercice 1. On considère le langage $\mathcal{L}$ formé des constantes a et b , d'un prédicat unaire P et d'un prédicat binaire Q (et en plus, les variables x, y , les connecteurs, les quantificateurs). Dire si les mots suivants sont des termes, des formules atomiques, des formules ?

$$\neg a, \quad xy, \quad (x \wedge a), \quad P(a), \quad \neg Q(x) \\ \forall x \quad Q(x, y), \quad \exists y \quad P(x), \quad \forall a \quad \exists Q(a, b) \\ \forall x \exists y \quad P(x \rightarrow y), \quad \exists x \left((\forall y \quad Q(x, y)) \vee (\forall y \quad \neg Q(x, y)) \right)$$

Exercice 2. On considère le langage $\mathcal{L}$ muni d'une constante c et d'un prédicat binaire $P(x, y)$. Dire si les mots suivants sont des termes, des formules atomiques, des formules ? Si oui, donner l'interprétation dans la structure de domaine $\mathbb{N}$, avec $c = 0$ et $P(x, y)$ défini par $x < y$. Faire ensuite l'interprétation dans la structure de domaine $\mathbb{R}$, avec $c = 1$ et $P(x, y)$ défini par $x \leq y$.

$$P, \quad c^2, \quad P(c, c), \quad \neg P(c, 0), \quad P(x, P(y, c)), \quad P(x = c) \rightarrow x = c \\ P(x, c), \quad \exists x, \quad x > c \rightarrow \forall y \quad P(y, c), \quad \forall y \quad (P(x, c) \rightarrow y = x) \\ \exists x \vee y \quad P(x, c) \vee P(y, c), \quad \forall x \quad ((x = c) \leftrightarrow (\exists y \quad P(y, c)))$$

Exercice 3. On considère un langage avec un prédicat binaire $R(x, y)$. On considère la structure de domaine $E = \{r, s, t, u\}$ et on interprète $R(x, y)$ comme vrai s'il existe une arête (non orientée) entre les sommets x et y d'un graphe G (par exemple comme ci-dessous).



Traduire dans le langage $\mathcal{L}$ les phrases suivantes :

1. Il existe un sommet relié à lui-même.
2. Il existe un sommet relié à tous les autres.
3. Tout sommet est relié à au moins un autre.

4. Il existe un sommet relié à aucun autre.
5. Tous les sommets ne sont pas reliés entre eux.
6. Il existe un seul sommet relié à lui même.

Exercice 4. On considère un langage $\mathcal{L}$ muni d'un prédicat $P(x)$ et de l'égalité $x = y$. Traduire par une formule de $\mathcal{L}$ les phrases suivantes :

1. Aucun élément ne vérifie P .
2. Il existe au plus un élément qui vérifie P .
3. Il existe exactement deux éléments qui vérifient P .
4. Il existe au moins trois éléments qui vérifient P .
5. Tous les éléments sauf un vérifient P .

Langage avec fonctions

Exercice 5. On considère un langage $\mathcal{L}$ ayant les constantes a, b , un prédicat $P(x)$ et deux fonctions $f(x)$ et $g(x, y)$. Dire si les mots suivants sont des termes, des formules atomiques, des formules de $\mathcal{L}$? Si oui, donner l'interprétation dans la structure de domaine $\mathbb{R}$, avec $a = 0$, $b = 1$, $P(x)$ défini par $x \geq 0$, $f(x) = \cos(x)$, $g(x, y) = x \times y$.

$$\begin{aligned}
 & f(x), \quad g(a, b), \quad f(P(a)), \quad \neg P(f(a)), \quad P(g(y, x)), \quad P(g(f(x), f(f(b)))) \\
 & P(x) \wedge P(y) \rightarrow g(x, y), \quad \exists x \, f(x) = b, \quad \forall y \, P(y) \leftrightarrow P(f(y)) \\
 & \forall x \forall b \, P(g(x, b)) \rightarrow (P(x) \wedge P(b)), \quad \exists x \, (P(x) \leftrightarrow (\forall y \, (P(y) \rightarrow P(g(x, y))))))
 \end{aligned}$$