Université d'Aix-Marseille
Faculté des sciences

Master 2, MAAP Informatique et Mathématiques Discrètes (IMD)
Mémoire de stage

Amplification de la Distance Contextuelle dans les Langages de programmation probabilistes d'ordre supérieur

---

Réalisé par
Houssein Mansour

Sous la direction de
Raphaëlle Crubillé

Laboratoire d'Informatique et des Systèmes LIS

4 mars 2024 – 3 juillet 2024

Remerciements

Je tiens à exprimer ma profonde gratitude à ma tutrice de stage, Raphaëlle Crubillé, pour son accueil chaleureux, ses conseils avisés, et son encadrement tout au long de cette période. Son expertise et sa disponibilité ont été d'une grande aide pour mener à bien mes missions. Elle m'a également permis d'assister à des séminaires et à des écoles de recherche, enrichissant considérablement mon expérience dans les domaines de mon stage. De plus, elle m'a encouragé à soumettre notre travail à un workshop international.

Je souhaite aussi remercier toute l'équipe de LSC pour leur soutien, leur bonne humeur, et leur collaboration, ainsi que l'équipe Ldp pour les séminaires communs entre ces deux équipes.

Mes remerciements s'étendent également à mes responsables de parcours en M2, Monsieur Laurent Regnier et Monsieur Kevin Perrot, pour leurs conseils précieux, leur suivi régulier, et leur soutien pédagogique. Je remercie aussi le responsable du M1 MAAP, Monsieur Dimitri Ara, ainsi que tous mes enseignants du M1 et du M2.

Je tiens encore à exprimer ma gratitude spéciale à Monsieur Lionel Vaux, toujours disponible pour répondre à mes demandes de conseils.

Enfin, j'adresse un grand merci à ma famille et à tous mes amis, je specifie mon ami Mounir, pour leur soutien moral constant et leurs encouragements tout au long de mes années de Master.

Houssein MANSOUR

Déroulement du stage

Au cours de ce stage, j'ai d'abord entrepris une étude bibliographique approfondie sur des sujets tels que le lambda-calcul probabiliste, la sémantique des langages de programmation probabilistes, l'équivalence et la distance contextuelle, ainsi que le phénomène de trivialisation... ([1, 2, 3]...) Cette revue de littérature a donné lieu à ma contribution originale : le développement d'une nouvelle preuve de la trivialisation. Nous avons effectué une soumission des résultats obtenus à HOPE 2024, un workshop international pour la comité de langages de programmation d'ordre supérieur avec effets.

J'ai réalisé ce stage sous la supervision de Raphaëlle Crubillé au LIS, au sein de l'équipe LSC (Logique, Sémantique et Catégories), et j'ai intégré les activités scientifiques de cette équipe ainsi que de l'équipe LDP (Logique de la programmation) de l'I2M, à travers des séminaires communs entre ces deux équipes, séminaires logique et interactions. En plus, les séminaires des doctorants de l'I2M, du CPT et du LIS m'ont permis d'avoir des interactions avec des doctorants de ces laboratoires.

Parallèlement à ces activités, j'ai assisté à divers séminaires et écoles de recherche, enrichissant ainsi mon expérience et élargissant mes perspectives académiques. Cela inclut les rencontres mensuelles CHoCoLa à Lyon, qui explore la correspondance Curry-Howard, entre le calcul et la logique. J'ai également assisté au workshop DiλLL2024 au CIRM, célébrant 20 ans de recherches sur le λ-calcul différentiel et la logique linéaire différentielle, et je compte assister la semaine du 17 juin 2024 à l'École des Jeunes Chercheurs et Chercheuses en Informatique Mathématique à Nantes EJCIM 2024, organisée par le Groupe de Recherche Mathématiques Informatique du CNRS.

Table des matières

1 Introduction

Un problème central dans l'étude des langages de programmation d'ordre supérieur est la question de l'équivalence des programmes : quand peut-on dire que deux programmes, bien que syntaxiquement différents, sont néanmoins équivalents ? À titre d'illustration, si l'on considère les transformations de programmes effectuées à des fins d'optimisation, on aimerait pouvoir affirmer et prouver que le programme résultant est, en quelque sorte, équivalent au programme initial. Une définition largement acceptée est l'équivalence contextuelle, qui a été introduite pour la première fois par Morris [10] : deux programmes sont équivalents lorsqu'ils se comportent exactement de la même manière, peu importe dans quel **contexte** on les place, et cette définition est rendue autonome en représantant les contextes comme des programmes écrits dans le même langage. Plus formellement, une définition générique de l'équivalence contextuelle pour un langage d'ordre supérieur pourrait être de la forme :

$$M \sim N \text{ lorsque } \forall C : \text{contexte}, \ \mathrm{Obs}(C[M]) = \mathrm{Obs}(C[N]),$$

où Obs est une notion d'**observation** associée au langage — par exemple, dans un langage probabiliste avec les booléens comme type de base, on peut dire que le type de sortie des contextes est Bool, et que $\mathrm{Obs}(C[M])$ devient la probabilité que $C[M]$ retourne 0.

Cependant, lorsque le langage qu'on considère est probabiliste, on aimerait également pouvoir dire qu'une transformation de programme renvoie un programme qui n'est pas toujours équivalent au programme d'entrée, mais qui se comporte de la même manière avec au moins une certaine probabilité donnée : cette idée a conduit au concept de **distances entre programmes**, pour les langages de programmation aléatoires [3, 4, 7, 8, 9]. Une généralisation naturelle de l'équivalence contextuelle, lorsque la notion d'observation est quantitative — dans le cas qui nous intéresse, une probabilité dans l'intervalle $[0, 1]$— pourrait être [3] :

$$d^{ctx}(M, N) = \sup_{C \text{ contexte}} |\mathrm{Obs}(C[M]) - \mathrm{Obs}(C[N])|$$

Il s'agit d'un **raffinement** de l'équivalence contextuelle, dans le sens où l'on peut récupérer les informations sur l'équivalence des programmes à partir de la pseudo-métrique d^{ctx} : il est facile de voir que deux programmes M, N sont contextuellement équivalents si et seulement si $d^{ctx}(M, N) = 0$. La définition ci-dessus, cependant, ne se comporte pas comme prévu dès que les contextes ont des **capacités de copie** : en particulier, si l'on commence avec un λ-calcul Λ avec copie, et avec un système de types assurant la terminaison, un **phénomène de trivialisation** se produit [3].

Théorème 1. *[3] Soit $\mathcal{L}$ un langage probabiliste où tous les programmes terminent, et où les contextes possèdent au moins les capacités suivantes : copier leur argument (avant et après l'évaluation), et calculer toutes les fonctions sur les nombres naturels construites en utilisant la récursion primitive. Alors pour toute paire de programmes comparables M, N, soit ils sont contextuellement équivalents, soit $d^{ctx}(M, N) = 1$.*

Cet résultat, démontré par Crubillé et Dal Lago, met en avant que, lorsque les contextes sont suffisament expressifs, alors pour n'importe quelle paire de programmes comparables M, N, ou bien ils sont contextuellement équivalents, ou bien

la distance contextuelle $d^{\text{ctx}}(M, N)$ est égale à 1. Ce résultat a été prouvé [3] en utilisant des techniques issues de l'analyse réelle : l'outil principal était le théorème de Bernstein, qui permet d'approximer toute fonction continue à valeurs réelles bornées par une séquence de fonctions polynomiales.

Dans ce stage , notre objectif est d'effectuer une **nouvelle preuve** qui utilise un résultat assez élémentaire en **théorie des probabilités**, à savoir la **loi des grands nombres** — prouvée par Bernoulli [11] dès 1713 — qui stipule que la moyenne empirique d'une distribution de probabilité **converge** — dans un sens précis — vers son **espérance**. Notre preuve donne une interprétation probabiliste du phénomène de trivialisation, qui apparaît dès lors comme une conséquence directe du fait que les contextes sont capables de calculer la moyenne empirique du terme fourni en argument.

2 $\mathbb{T}_\oplus$: Variante Probabiliste du Système $\mathbb{T}$

Dans ce rapport, tous les développements techniques sont effectués dans le langage $\mathbb{T}_\oplus$, qui est une variante probabiliste du système $\mathbb{T}$ de Gödel qui conserve la propriété d'arrêt. La sémantique de ce langage a été étudiée par Breuvart, Dal Lago et Herrou [2].

2.1 Syntaxe et Règles de Typage pour $\mathbb{T}_\oplus$

Dans cette section, on présente la syntaxe et la sémantique de $\mathbb{T}_\oplus$ telles que définies dans [2]. Le système $\mathbb{T}$ est un lambda calcul typé avec les nombres entiers comme type de base, et un opérateur pour la récursion primitive. La grammaire pour les types dans $\mathbb{T}_\oplus$ est définie comme suit : le seul type de base est celui des nombres naturels, mais nous pouvons à partir de là construire des types produits et des types fonctionnels.

$$\sigma, \tau \in \mathcal{T}_\oplus ::= \text{Nat} \mid \sigma \to \tau \mid \sigma \times \tau.$$

La syntaxe de $\mathbb{T}_\oplus$ est une extension de celle de $\mathbb{T}$, comprenant un opérateur de choix binaire probabiliste, dénoté par $\oplus$. L'objectif de cet opérateur est de modeler le comportement du programme $M \oplus N$ de sorte qu'il exécute M ou N avec une probabilité respective de $\frac{1}{2}$. De façon formelle, on suppose qu'on dispose d'un ensemble dénombrable de variables $\mathbf{V}$. Les termes de $\mathbb{T}_\oplus$ sont générés selon la grammaire suivante :

$$M ::= x \mid \underline{0} \mid succ(M) \mid \langle M, N\rangle \mid \pi_1 \mid \pi_2 \mid rec \mid M \oplus N \mid \lambda x.M \mid MN, \quad \text{où } x \in \mathbf{V}.$$

On note pour tout $n \in \mathbb{N}$, $\underline{n} = \underbrace{succ(\dots(succ(\underline{0})))}_{n \text{ fois}}$.

Si un terme est de la forme $\lambda x.M$, on dit que la variable x est liée dans M. Si M est un terme et x est une variable qui apparaît dans une position non liée, on dit que x est libre dans M. On notera $FV(M)$ l'ensemble des variables libres dans M. On

dit qu'un terme est fermé (clos), lorsqu'il ne contient pas de variables libres. Ce sont uniquement ces termes fermés qui ont pour nous une signification computationnelle, dans le sens où ils peuvent être exécutés. Pour cette raison, on les appellera des programmes et on notera $P_{\mathbb{T}_\oplus}$ l'ensemble des programmes de $\mathbb{T}_\oplus$.

Règles de typage

Les règles de typage du système $\mathbb{T}_\oplus$ sont les suivantes :

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma} \qquad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

$$\frac{}{\Gamma \vdash 0 : \mathrm{Nat}} \qquad \frac{\Gamma \vdash M : \mathrm{Nat}}{\Gamma \vdash \mathrm{succ}(M) : \mathrm{Nat}} \qquad \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M \oplus N : \sigma}$$

$$\frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau}{\Gamma \vdash \langle M, N \rangle : \sigma \times \tau} \qquad \frac{}{\Gamma \vdash \pi_1 : (\sigma \times \tau) \to \sigma} \qquad \frac{}{\Gamma \vdash \pi_2 : (\sigma \times \tau) \to \tau}$$

$$\frac{}{\Gamma \vdash \mathrm{rec} : (\sigma \times (\mathrm{Nat} \to \sigma \to \sigma) \times \mathrm{Nat}) \to \sigma}$$

Il est désormais nécessaire de fournir à $\mathbb{T}_\oplus$ une sémantique opérationnelle qui spécifie l'exécution de ces programmes. Dans notre travail, la réduction, comme dans [1], se fait en appel par valeur (CBV). Dans ce qui suit on va définir la stratégie d'évaluation CBV et expliciter la sémantique opérationnelle.

2.2 Sémantique opérationnelle

Dans le contexte déterministe, la sémantique opérationnelle d'un programme M est caractérisée par la relation $M \downarrow V$, où V est la valeur résultante de l'exécution de M. En revanche, dans un environnement probabiliste, l'exécution d'un programme peut aboutir à plusieurs valeurs potentielles, chacune associée à une probabilité spécifique. Ainsi, la sémantique opérationnelle d'un programme probabiliste de $\mathbb{T}_\oplus$ est une distribution de probabilité sur l'ensemble des valeurs, indiquant la probabilité associée à chaque valeur potentielle issue de l'exécution du programme.

Il devient donc nécessaire de considérer une relation de réduction probabiliste pour la relation de réduction en une seule étape.

Définition 1. *[Distribution de probabilité] On définit une distribution de probabilité sur un ensemble dénombrable S comme une fonction $\mathcal{D} : S \to [0, 1]$, avec la condition que $\sum_{s \in S} \mathcal{D}(s) = 1$.*

Notation 1. *Soit $\mathcal{D}$ une distribution sur S, on note* $\mathrm{supp}(\mathcal{D})$ *le support de $\mathcal{D}$, c'est-à-dire l'ensemble des éléments $s \in S$ tels que $\mathcal{D}(s) > 0$.*

Notation 2. *Soit S un ensemble dénombrable, on note $\mathcal{D}(S)$ l'ensemble des distributions sur S.*

Il est important de noter qu'une distribution qui assigne la probabilité 1 à un élément $s \in S$ est appelée une distribution de Dirac, dénotée par $\{s\}^1$, où $\{s\}^1(s') = 1$ si $s' = s$ et $\{s\}^1(s') = 0$ sinon.

Remarque 1. *Soit S un ensemble dénombrable, pour $\mathcal{D} \in \mathcal{D}(S)$, on peut toujours écrire :*

$$\mathcal{D} = \sum_{s \in \text{supp}(\mathcal{D})} \mathcal{D}(s) \cdot \{s\}^1,$$

où on utilise le fait que l'ensemble des distributions est stable par somme convexe.

2.2.1 Les règles de réduction pour les redexes

On note par $M\{V/x\}$ le terme obtenu en remplaçant dans M chaque occurrence libre de la variable x par V. Puisque nous avons choisi dans ce travail le paradigme de l'appel par valeur, nous devons d'abord définir formellement quand un programme est une valeur : on définit $\mathcal{V}_{\mathbb{T}_\oplus}$, l'ensemble des valeurs de $\mathbb{T}_\oplus$, comme l'ensemble des termes clos générés par la grammaire suivante :

$$V, W \in \mathcal{V}_{\mathbb{T}_\oplus} ::= \lambda x.M \mid \langle V, W\rangle \mid \pi_1 \mid \pi_2 \mid \text{rec} \mid \underline{0} \mid \text{succ}(V) \quad \text{où } x \in \mathbf{V}$$

Les redexes de $T_\oplus$ sont générés par la grammaire suivante :

$$\begin{aligned} M, N \in \mathrm{R}_{\mathbb{T}_\oplus} ::= & M \oplus N \mid (\lambda x.M)V \mid \pi_1\langle V, W\rangle \mid \pi_2\langle V, W\rangle \mid rec\langle\langle V, W\rangle, \underline{0}\rangle \mid \\ & rec\langle\langle V, W\rangle, succ(S)\rangle \quad \text{où } x \in \mathbf{V} \text{ , et } V, W \text{ et } S \in \mathcal{V}_{\mathbb{T}_\oplus}. \end{aligned}$$

Un terme est toujours évalué avant d'être passé en argument à une fonction, et un choix probabiliste est toujours exécuté avant que sa branche gauche ou sa branche droite ne soit réduite.

Définition 2. *On définit les règles de réduction CBV sur les redexes comme la relation $\to_R \subseteq R_{\mathbb{T}_\oplus} \times \mathcal{D}(P_{\mathbb{T}_\oplus})$ avec :*

- $M \oplus N \to_R \frac{1}{2}\{M\}^1 \oplus \{N\}^1$ *quand* $M, N \in P_{\mathbb{T}_\oplus}$;
- $(\lambda x.M)V \to_R \{M\{V/x\}\}^1$ *quand* $V \in \mathcal{V}_{\mathbb{T}_\oplus}$;
- $\pi_1\langle V, W\rangle \to_R \{V\}^1$ *quand* $V, W \in \mathcal{V}_{\mathbb{T}_\oplus}$;
- $\pi_2\langle V, W\rangle \to_R \{W\}^1$ *quand* $V, W \in \mathcal{V}_{\mathbb{T}_\oplus}$;
- $rec\langle\langle V, W\rangle, \underline{0}\rangle \to_R \{V\}^1$; $\quad rec\langle\langle V, W\rangle, succ(S)\rangle \to_R \{W\, S\, rec\langle\langle V, W\rangle, S\rangle\}^1$ *quand* V, W *et* $S \in \mathcal{V}_{\mathbb{T}_\oplus}$.

2.2.2 Relation de réduction en un pas pour $\mathbb{T}_\oplus$

Les contextes d'évaluation pour $\mathbb{T}_\oplus$ sont définis pour la stratégie CBV comme suit :

$$\mathcal{E} \in E_{\mathbb{T}_\oplus} ::= [\cdot] \mid \mathcal{E}M \mid V\mathcal{E} \mid \langle \mathcal{E}, M\rangle \mid \langle V, \mathcal{E}\rangle \mid succ(\mathcal{E}), \quad M \in P_{\mathbb{T}_\oplus}, \quad V \in \mathcal{V}_{\mathbb{T}_\oplus}.$$

Observez que dans un contexte d'évaluation $\mathcal{E}$, il y a exactement une occurrence du trou $[\cdot]$. Nous notons par $\mathcal{E}[M]$ le terme obtenu lorsque nous remplaçons le trou $[\cdot]$ par M dans $\mathcal{E}$.
Il est important de noter qu'il existe pour chaque programme M, au plus une seule décomposition $M = \mathcal{E}[R]$ avec $\mathcal{E}[\cdot] \in E_{\mathbb{T}_\oplus}$ et $R \in R_{\mathbb{T}_\oplus}$.

Définition 3. *La relation de réduction sur $\mathbb{T}_\oplus$ par rapport à CBV, $\to \subseteq P_{\mathbb{T}_\oplus} \times \mathcal{D}(P_{\mathbb{T}_\oplus})$, se définit comme suit :*

$$\frac{R \to_R \mathcal{D} \quad \mathcal{E} \in E_{\mathbb{T}_\oplus} \quad R \in R_{\mathbb{T}_\oplus}}{\mathcal{E}[R] \to \sum_{N \in supp(\mathcal{D})} \mathcal{D}(N) \cdot \{\mathcal{E}[N]\}^1}.$$

Cette relation est déterministe, dans le sens où pour chaque programme il existe au plus une réduction probabiliste possible.
D'après [2], le système $\mathbb{T}_\oplus$ préserve la propriété de réduction de sujet pour la relation de réduction $\to$, ce qui signifie que si $\vdash M : \sigma$ et $M \to \mathcal{D}$, alors $\forall N_i \in supp(\mathcal{D}), \vdash N_i : \sigma$.

2.2.3 Sémantique Opérationnelle

Dans cette sous-section, on transforme la relation de réduction sur les programmes définie dans la sous-section précédente en une relation de réduction déterministe sur les distributions de probabilité.

Définition 4. *On définit une rélation pour $\mathbb{T}_\oplus$ par rapport à $\to$ comme étant la plus petite relation $\to_{\mathcal{D}(P_{\mathbb{T}_\oplus})} \subseteq \mathcal{D}(P_{\mathbb{T}_\oplus}) \times \mathcal{D}(P_{\mathbb{T}_\oplus})$ qui satisfait la règle suivante :*

$$\frac{\mathcal{D} = \sum_i \alpha_i \cdot \{M_i\}^1 + \sum_j \beta_j \cdot \{V_j\}^1 \quad M_i \to \mathscr{E}_i \quad V_i \in \mathcal{V}_{\mathbb{T}_\oplus}}{\mathcal{D} \to_{\mathcal{D}(P_{\mathbb{T}_\oplus})} \sum_i \alpha_i \cdot \mathscr{E}_i + \sum_j \beta_j \cdot \{V_j\}^1}.$$

Notation 3. *Par abus de notation, on note également $\to$ pour la relation $\to_{\mathcal{D}(P_{\mathbb{T}_\oplus})}$.*

Définition 5. *On définit $\to^*$ comme la clôture réflexive et transitive de $\to_{\mathcal{D}(P_{\mathbb{T}_\oplus})}$. Cela signifie que $\to^*$ inclut toutes les séquences d'étapes de réduction qui peuvent être formées en utilisant $\to_{\mathcal{D}(P_{\mathbb{T}_\oplus})}$ une ou plusieurs fois.*

Lemme 1. *Dans le système $\mathbb{T}_\oplus$, pour tout programme M, il existe $k \in \mathbb{N}$ tel que toutes les branches d'execution probabilistes de ce programme ont une longueur $\leq k$ d'après [2].*

Plus généralement, il existe des propriétés plus faibles de terminaison des programmes, comme la Terminaison Presque Sûre (TPS), qui signifie que l'ensemble des exécutions de calcul infinies a une probabilité nulle. Nos résultats seraient encore valables pour des langages de programmation probabilistes vérifiant cette propriété plus faible de terminaison.

Définition 6. *[sémantique opérationnelle] Soit M un programme dans $\mathbb{T}_\oplus$. On définit la sémantique opérationnelle de M comme suit :*

$$[\![M]\!] = \{\textit{l'unique } D, \textit{ distribution sur des valeurs dans } \mathcal{V}_{\mathbb{T}_\oplus}, \textit{ tel que } M \to^* D\}.$$

Nous allons maintenant montrer à travers un exemple comment décrire la réduction de programmes dans $\mathbb{T}_\oplus$.

Exemple 1. *Considérons le terme fermé de $\mathbb{T}_\oplus$ suivant :*

$$M = rec\langle\langle \underline{m}, \lambda y.\lambda x.succ(x)\rangle, \underline{0} \oplus succ(\underline{0})\rangle,$$

Trouvons la sémantique opérationnelle de M :

$$\begin{aligned}
M &= rec\langle\langle \underline{m}, \lambda y.\lambda x.succ(x)\rangle, \underline{0} \oplus succ(\underline{0})\rangle \\
&\to \frac{1}{2}\{rec\langle\langle \underline{m}, \lambda y.\lambda x.succ(x)\rangle, \underline{0}\rangle\}^1 + \frac{1}{2}\{rec\langle\langle \underline{m}, \lambda y.\lambda x.succ(x)\rangle, succ(\underline{0})\rangle\}^1 \\
&\to \frac{1}{2}\{\underline{m}\}^1 + \frac{1}{2}\{(\lambda y.\lambda x.succ(x))\,\underline{0}\,rec\langle\langle \underline{m}, \lambda y.\lambda x.succ(x)\rangle, \underline{0}\rangle\}^1 \\
&\to \frac{1}{2}\{\underline{m}\}^1 + \frac{1}{2}\{(\lambda x.succ(x))\,rec\langle\langle \underline{m}, \lambda y.\lambda x.succ(x)\rangle, \underline{m}\rangle\}^1 \\
&\to \frac{1}{2}\{\underline{m}\}^1 + \frac{1}{2}\{(\lambda x.succ(x))\,\underline{m}\}^1 \\
&\to \frac{1}{2}\{\underline{m}\}^1 + \frac{1}{2}\{succ(\underline{m})\}^1.
\end{aligned}$$

Parsuite, $[\![M]\!] = \frac{1}{2}\{\underline{m}\}^1 + \frac{1}{2}\{succ(\underline{m})\}^1$.

3 Équivalence Contextuelle

L'objectif de cette section est de présenter formellement les définitions de l'équivalence contextuelle et la distance contextuelle dans le système $\mathbb{T}_\oplus$, ainsi que le phénomène de trivialisation. Pour parler d'équivalence contextuelle, on a besoin d'une notion de contextes, et une notion d'observables, qui désignent ce que l'on décide d'observer sur les résultats de l'exécution d'un programme.

3.1 Contextes et Observables

Les contextes de $\mathbb{T}_\oplus$, $C_{\mathbb{T}_\oplus}$, sont générés par la grammaire suivante :

$$C \in C_{\mathbb{T}_\oplus} ::= [\cdot] \mid \lambda x.C \mid C\,M \mid M\,C \mid \langle C, M\rangle \mid \langle M, C\rangle \mid C \oplus M \mid M \oplus C, \quad M \in P_{\mathbb{T}_\oplus}.$$

Un contexte C contient exactement une occurrence du trou $[\cdot]$. On désigne par $C[M]$ le terme obtenu en remplaçant le trou $[\cdot]$ par M. Observons ici que, parce que l'on travaille avec des termes M clos dans $P_{\mathbb{T}_\oplus}$, $C[M]$ est toujours clos dans $P_{\mathbb{T}_\oplus}$.

Notation 4. *On note $[\cdot] : \alpha \vdash C : \tau$ lorsque ce jugement de typage est valide par les règles de typage de la section 2.1. En considérant $[\cdot]$ comme une variable, on peut voir que si $\vdash M : \alpha$, alors $\vdash C[M] : \tau$.*

Puisque $\mathbb{T}_\oplus$ est un langage typé, un choix raisonnable d'observation est d'observer des programmes de type de base : si M est un programme de type **Nat**, on prend $\mathrm{Obs}_{\mathrm{Nat}}(M) = [\![M]\!](\underline{0})$, c'est-à-dire la probabilité que le résultat de l'exécution de M soit $\underline{0}$.

3.2 Distance contextuelle et phénomène de trivialisation dans $\mathbb{T}_\oplus$

Maintenant, on est capables de définir l'équivalence et la distance contextuelle dans $\mathbb{T}_\oplus$.

Définition 7. *[Équivalence contextuelle] Soient M_1 et $M_2 \in \mathbb{T}_\oplus$ de type α. On dit que M_1 et M_2 sont contextuellement équivalents et on note $M_1 \equiv_\alpha M_2$ si on a :*

$$\forall C \ \ t.q. \ \ [\cdot] : \alpha \vdash \mathcal{C} : Nat, \ [\![\mathcal{C}[M_1]]\!](\underline{0}) = [\![\mathcal{C}[M_2]]\!](\underline{0}).$$

Définition 8. *[Distance contextuelle] Soient M_1 et $M_2 \in \mathbb{T}_\oplus$ de type α. On définit la distance contextuelle entre M_1 et M_2 comme suit :*

$$(\delta^{ctx}_{\mathbb{T}_\oplus})_\alpha(M_1, M_2) = \sup_{\mathcal{C} \ \ t.q. \ \ [\cdot]:\alpha\vdash\mathcal{C}:Nat} \{|[\![\mathcal{C}[M_1]]\!](\underline{0}) - [\![\mathcal{C}[M_2]]\!](\underline{0})|\}\,.$$

Théorème 2. *[3] La distance contextuelle $(\delta^{ctx}_{\mathbb{T}_\oplus})_\alpha$ trivialise. Dans le sens où, pour toutes paires de programmes M_1 et M_2 de même type α, on a soit $M_1 \equiv_\alpha M_2$, soit $(\delta^{ctx}_{\mathbb{T}_\oplus})_\sigma(M_1, M_2) = 0$.*

Rappelons que la preuve de Crubillé et Dal Lago [3] de cet résultat de trivialisation dans $\mathbb{T}_\oplus$ est basée sur des techniques d'analyse réelle en utilisant les polynômes de Bernstein. Notre objectif dans ce qui suit est de montrer le résultat de trivialisation dans $\mathbb{T}^\oplus$ en utilisant des techniques de théorie des probabilités.

4 Trivialisation dans $\mathbb{T}_\oplus$

Pour notre démonstration du trivialisation, nous allons nous servir de la loi faible des grands nombres, qui est facilement énonçable en termes de variables aléatoires. Pour cette raison, nous résumons ci-dessous ce formalisme mathématique des variables aléatoires.

4.1 Variables aléatoires et loi faible de grands nombres

Définition 9. *[Espace mesurable et espace probabilisé] Un espace mesurable est un couple $(E, \mathcal{E})$ où :*

- *E est un ensemble.*
- *$\mathcal{E}$ est une σ-algèbre sur E, c'est-à-dire une collection de sous-ensembles de E qui satisfait les propriétés suivantes :*
  1. *$E \in \mathcal{E}$,*
  2. *Si $A \in \mathcal{E}$, alors $E \setminus A \in \mathcal{E}$,*
  3. *Si $A_n \in \mathcal{E}$ pour $n \in \mathbb{N}$, alors $\bigcup_{n=1}^{\infty} A_n \in \mathcal{E}$.*

Un espace probabilisé est un triplet $(\Omega, \mathcal{F}, \mu)$ où :

- *Ω est l'ensemble des événements possibles (l'espace échantillon).*
- *$\mathcal{F}$ est une σ-algèbre sur Ω.*

— *μ est une mesure de probabilité sur $(\Omega, \mathcal{F})$, c'est-à-dire une fonction $\mu : \mathcal{F} \to [0,1]$ qui satisfait les propriétés suivantes :*
  1. *$\mu(\Omega) = 1$,*
  2. *Pour toute suite d'ensembles disjoints $A_n \in \mathcal{F}$, on a $\mu\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mu(A_n)$.*

Définition 10. *[Fonction mesurable] Soient $(\Omega, \mathcal{F})$ et $(E, \mathcal{E})$ deux espaces mesurables. Une fonction $X : \Omega \to E$ est dite mesurable si, pour tout ensemble $B \in \mathcal{E}$, l'ensemble $X^{-1}(B) = \{\omega \in \Omega \mid X(\omega) \in B\} \in \mathcal{F}$.*

Définition 11. *[Variable aléatoire] Une variable aléatoire est une fonction mesurable $X : \Omega \to E$ définie sur un espace probabilisé (Ω, μ) et prenant ses valeurs dans un espace mesurable E.*

Si E est un ensemble dénombrable alors on dit que X est une variable aléatoire discrète.
Dans la suite E sera souvent un ensemble dénombrable et Ω également. Dans cette cas, une mesure de probabilité est donc simplement une distribution de probabilité. Ainsi, un espace probabilisé est donc juste (Ω, μ), où μ est une distribution de probabilité sur Ω.

Définition 12. *[Mesure image (push-forward)] Soient $(\Omega, \mathcal{F})$ et $(E, \mathcal{E})$ deux espaces mesurables, μ une mesure sur $(\Omega, \mathcal{F})$, et $f : \Omega \to E$ une fonction mesurable. La mesure image (ou push-forward) de μ par f, notée $f_*\mu$, est la mesure sur $(E, \mathcal{E})$ définie par :*

$$(f_*\mu)(B) = \mu(f^{-1}(B)) \quad \textit{pour tout } B \in \mathcal{E}.$$

Définition 13. *[Loi de probabilité d'une variable aléatoire] Soient (Ω, μ) un espace probabilisé et $(E, \mathcal{E})$ un espace mesurable, la loi de probabilité d'une variable aléatoire $X : \Omega \to E$ est la mesure image $X_*\mu$ sur $(E, \mathcal{E})$, définie par :*

$$(X_*\mu)(B) = \mu(X^{-1}(B)) \quad \textit{pour tout } B \in \mathcal{E}.$$

Notation 5. *Soient (Ω, μ) un espace probabilisé et E un espace mesurable, on peut écrire simplement $loi(X)$ pour $(X_*\mu)$.*

Définition 14. *[Loi jointe de variables aléatoires] Soient (Ω, μ) un espace probabilisé et $(E, \mathcal{E})$ un espace mesurable. Soit $k \in \mathbb{N}$, et soient $X_i : \Omega \to E$ pour $1 \leq i \leq k$ des variables aléatoires. La fonction $(X_1, \cdots, X_k) : \Omega \to \mathcal{E}^k$, qui à chaque ω associe $(X_1(\omega), \cdots, X_k(\omega))$, est une variable aléatoire. On appelle la loi de cette variable aléatoire la loi jointe des variables aléatoires X_i, et elle est définie comme suit :*

$$loi(X_1, \ldots, X_k)(B) = \mu\left(\{\omega \in \Omega \mid (X_1(\omega), \ldots, X_k(\omega)) \in B\}\right) \quad \textit{pour tout } B \in \mathcal{E}^k.$$

Définition 15. *[Variables aléatoires indépendantes dans le cas discrèt] Soient (Ω, μ) un espace probabilisé et $(E, \mathcal{E})$ un espace mesurable. Soit $k \in \mathbb{N}$, et soient $X_i : \Omega \to E$ pour $1 \leq i \leq k$ des variables aléatoires discrètes. On dit que ces variables aléatoires sont indépendantes si la loi jointe de $(X_1, \ldots, X_k)$ est égale au produit de leurs lois individuelles, c'est-à-dire :*

$$loi(X_1 \times \cdots \times X_k)(B) = \prod_{i=1}^{k} loi(X_i)(B_i) \quad \textit{pour tout } B = B_1 \times \cdots \times B_k \in \mathcal{E}^k.$$

Définition 16. *[Espérance d'une variable aléatoire discrète] Soit X une variable aléatoire discrète définie sur un espace probabilisé (Ω, μ) dans un espace mésurable E, l'espérance mathématique de X, notée $\mathbb{E}(X)$, est définie par :*

$$\mathbb{E}(X) = \sum_{x \in E} x \cdot Proba(X = x),$$

Remarque 2. *L'espérance d'une variable aléatoire continue est définie en utilisant l'intégrale de Lebesgue. Cependant, nous n'allons pas travailler avec des espérances de variables aléatoires continues dans ce travail.*

Exemple 2. — *Variable aléatoire de Bernoulli : Une variable aléatoire discrète X est une variable aléatoire de Bernoulli de paramètre p si elle peut prendre uniquement deux valeurs, typiquement 0 et 1, c'est-à-dire $X(\Omega) = \{0, 1\}$, avec des probabilités p pour 1 et $1 - p$ pour 0, où $0 \leq p \leq 1$.*

— *La loi de probabilité d'une variable aléatoire de Bernoulli X est une distribution de probabilitée sur {0,1} donnée par :*

$$loi(X)(x) = proba(X = x) = \begin{cases} p & si\ x = 1, \\ 1 - p & si\ x = 0. \end{cases}$$

— *Push-forward de $loi(X)$: soient $X : \Omega \to E$ une variable aléatoire discrète et $f : E \to F$, le push-forward de $loi(X)$ est définie par :*

$$f_*(loi(X)) = \sum_{x \in E} loi(X)(x) \cdot \{f(x)\}^1.$$

— *Loi jointe : Soient (Ω, μ) un espace probabilisé et $k \in \mathbb{N}$. Considérons des variables aléatoires indépendantes $X_1, \ldots, X_k$ qui suivent chacune une loi de Bernoulli avec des paramètres $p_1, \ldots, p_k$ respectivement, et telles que $X_i : \Omega \to \{0, 1\}$. La loi jointe de ces variables, notée $loi(X_1, \ldots, X_k)$, est une distribution de probabilité sur $\{0, 1\}^k$ telle que :*

$$\begin{aligned} loi(X_1, \ldots, X_k)(x_1, \ldots, x_k) &= proba(X_1 = x_1, \ldots, X_k = x_k) \\ &\overset{v.a.\ indépendantes}{=} proba(X_1 = x_1) \times \cdots \times proba(X_k = x_k) \\ &= \prod_{i=1}^{k} p_i^{x_i} (1 - p_i)^{1 - x_i}, \end{aligned}$$

où chaque x_i est soit 0 soit 1.

Lemme 2. *[6](loi faible de grands nombres) Soit (X_k) une suite de variables aléatoires réelles indépendantes et de même loi admettant une espérance e. La moyenne empirique $\overline{X}_k = \frac{1}{k} \sum_{i=1}^{k} X_i$ converge en probabilité vers cette espérance : pour tout $\epsilon > 0$, on a*

$$\lim_{k \to \infty} P(|\overline{X}_k - e| \leq \epsilon) = 1.$$

Lemme 3. *Soient (Ω, μ) un espace probabilisé, A et B deux espaces mesurables , $X : \Omega \to A$ une variable aléatoire et $f : A \to E$ une fonction mesurable. On a que :*

$$loi(f(X)) = f_*(loi(X)).$$

Démonstration. Soit Σ_2 la tribu associée à E et $B \in \Sigma_2$,

— $(f \circ X)_*(\mu)(B) = \mu((f \circ X)^{-1}(B)) = \mu((X^{-1} \circ f^{-1})(B))$.

— $(f_*(X_*(\mu)))(B) = X_*(\mu)(f^{-1}(B)) = \mu((X^{-1}(f^{-1}(B))) = \mu((X^{-1} \circ f^{-1})(B))$.

Donc $(f \circ X)_*(\mu) = (f_*(X_*(\mu)))$.

— $loi(f(X)) = (f(X))_*(\mu) = (f \circ X)_*(\mu) = (f_*(X_*(\mu))$.

— $f_*(loi(X)) = f_*(X_*(\mu))$.

□

4.2 Trivialisation dans $\mathbb{T}_\oplus$

Dans cette section, nous allons nous intéresser particulièrement aux programmes dont la sémantique opérationnelle peut être modélisée par une variable aléatoire de Bernoulli. Fondamentalement, ces variables aléatoires sont intéressantes pour nous parce que deux distributions de probabilité qui prennent des valeurs dans {0, 1} sont égales si et seulement si leurs espérances sont égales. C'est pour cela que nous introduisons une notation pour parler des programmes de type Nat qui prennent seulement 0 et 1 comme valeurs.

Notation 6. *Dans la suite de notre travaille on note $\vdash M :$ **Bool** si $\vdash M :$ Nat tel que supp$([\![M]\!]) = \{\underline{0}, \underline{1}\}$.*

Soit $M \in \mathbb{T}^\oplus$ tel que $\vdash M :$ Bool, alors il existe un entier $m \in \mathbb{N}$ tel que $M \to^* [\![M]\!]$ avec un nombre d'étapes $\leq m$. En utilisant ce fait, nous pouvons conclure que pour tout M de type Bool, il existe $e \in \mathbb{Q}_2$ tel que

$$[\![M]\!] = [\![\underline{0} \oplus^e \underline{1}]\!].$$

où $\mathbb{Q}_2$ est l'ensemble de nombres dyadiques qui est défini par

$$\mathbb{Q}_2 = \left\{ \frac{k}{2^p} : k \in \mathbb{N}, p \in \mathbb{N} \right\}.$$

Cela signifie que le programme M renvoie $\underline{1}$ avec une probabilité e et $\underline{0}$ avec une probabilité $1 - e$. Ainsi nous définirons l'espérance de M, $\mathbb{E}(M) = \mathbb{E}([\![M]\!]) = e$.

Dans ce qui suit, notre objectif est de montrer la trivialisation de la distance contextuelle sur $\mathbb{T}_\oplus$. Nous allons suivre le même chemin que [3]. Tout d'abord, nous cherchons à trouver, pour deux programmes non équivalents M_1 et M_2 dans $\mathbb{T}_\oplus$, tels que $\vdash M_1, M_2 :$ Bool, une famille de contextes qui amplifient la distance.

Définition 17. *[Contextes d'amplification] Une famille de contextes C_n est considérée comme une famille de contextes d'amplification si elle satisfait la condition suivante :*

$$\lim_{n\to\infty} |Obs(C_n[M_1]) - Obs(C_n[M_2])| = 1,$$

où M_1 et M_2 sont deux programmes de même type.

Notre objectif se fera en deux étapes qui seront la contribution originale de ce rapport, ensuite une troizième étape pour completer la preuve.

— Les deux étapes sont les deux sous-sections **Contextes d'amplification** et **Utilisation de la loi des grands nombres pour l'amplification**.

— Ensuite, dans la dernière sous-section **Trivialisation dans** $\mathbb{T}_\oplus$, nous utiliserons ces deux étapes pour montrer la trivialisation pour tous les programmes M_1 et M_2 de n'importe quel type α.

4.2.1 Contextes d'amplification

Pour trouver une telle famille de contextes, nous avons besoin de la définition ci-dessous.

Définition 18. *Soit $k \in \mathbb{N}$, e et $\epsilon \in Q_2$ nous définissons :*

$$f_k^{e,\epsilon} : \mathbb{N}^k \to \{0,1\}$$

où

$$f_k^{e,\epsilon}(n_1, n_2, \ldots, n_k) = \begin{cases} 0 & si \ \left| \frac{\sum_{i=1}^k n_i}{k} - e \right| \leq \epsilon, \\ 1 & sinon \end{cases}$$

Le système $\mathbb{T}_\oplus$ n'est pas Turing-complet, mais il est raisonnablement expressif, comme noté dans [2], où il est mentionné que l'on peut encoder toutes les fonctions de $\mathbb{N}$ dans $\mathbb{N}$ construites à l'aide de la récursion primitive .
Nous démontrons dans l'Annexe A que nous sommes capables d'encoder l'addition, la multiplication, la valeur absolue, l'exponentielle, le if-then-else, ainsi que la comparaison inférieur ou égal. Ensuite, nous prouvons que nous sommes capables d'encoder $f_k^{e,\epsilon}$ dans $\mathbb{T}_\oplus$, c'est à dire de trouver un terme $\underline{f_k^{e,\epsilon}}$ tel que :

$$\underline{f_k^{e,\epsilon}}(\underline{n_1}, \ldots, \underline{n_k}) \to^* \left\{ \underline{f_k^{\alpha,\epsilon}(n_1, \ldots, n_k)} \right\}^1.$$

Définition 19. *Soient $k \in \mathbb{N}, e$ et $\epsilon \in \mathbb{Q}_2$, nous définissons la famille de contextes :*

$$C_k^{e,\epsilon}[\cdot] = \lambda x.((\lambda y_1 \ldots \lambda y_k.\underline{f_k^{e,\epsilon}}(y_1, \ldots, y_k)) \underbrace{(x\,\underline{0}) \ldots (x\,\underline{0})}_{k fois}) \lambda x.[\cdot].$$

Pour comprendre le comportement de cette famille de contextes $C_k^{e,\epsilon}$, on peut decrire son comportement sur un programme M avec $\vdash M$: Bool. Dans ce cas, on peut représenter un contexte $C_k^{e,\epsilon}$ comme prenant en argument une distribution $\mathcal{D}$ sur les booléens : elle duplique cette distribution k fois sans évaluation initiale (parce que les étapes de réduction ne se font pas sous un λ, donc $\lambda x.\mathcal{D}$ empêche la distribution de s'évaluer avant d'être copiée), puis, après copie, elle fait évaluer chaque copie indépendamment, parce que la réduction se fait en appel par valeur. Ensuite, elle applique $\underline{f_k^{e,\epsilon}}$ aux résultats pour calculer la moyenne empirique de $\mathcal{D}$ au rang k. Finalement, elle teste si cette moyenne est égale à la valeur e , avec une précision ϵ.

Notre objectif maintenant est de calculer $[\![C_k^{e,\epsilon}[M]]\!]$ avec $\vdash M$: Bool.

Proposition 1. *Soit $M \in \mathbb{T}_{\oplus}$ avec $\vdash M : Bool$ Alors $\forall k \in \mathbb{N}, \forall e$ et $\epsilon \in \mathbb{Q}_2$, on a que :*

$$\begin{aligned}
[\![C_k^{e,\epsilon}[M]]\!] &= \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot \left\{\underline{f_k^{e,\epsilon}(n_1,\ldots,n_k)}\right\}^1 \\
&= \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k \ t.q. \ \left|\frac{\sum_{i=1}^k n_i}{k}-e\right|\leq\epsilon} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot \{\underline{0}\}^1 \\
&+ \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k \ t.q. \ \left|\frac{\sum_{i=1}^k n_i}{k}-e\right|>\epsilon} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot \{\underline{1}\}^1
\end{aligned}$$

Le reste de cette sous-section est consacré à la preuve de cette proposition.

Lemme 4. *Pour toute expression $M \in \mathbb{T}^{\oplus}$ et contexte d'évaluation $\mathcal{E}$ de M, avec $M \to^* [\![M]\!]$, on a que :*

$$[\![\mathcal{E}[M]]\!] = \sum_{v\in support([\![M]\!])} [\![M]\!](v) \cdot [\![\mathcal{E}[v]]\!].$$

Démonstration. Résultat direct de [2].

□

Définition 20. *Soit $M \in \mathbb{T}_{\oplus}$ avec $\vdash M : Bool$. $\forall k \in \mathbb{N}$, $\forall e$ et $\epsilon \in \mathbb{Q}_2$, pour tout $i \in \{0,\ldots,k-1\}$, on définit $P_{n_1,\ldots,n_i}$ comme suit :*

$$P^{k,e,\epsilon}_{n_1,\ldots,n_i} = (\lambda y_{i+1} \ldots \lambda y_k.\, \underline{f_k^{e,\epsilon}}(\underline{n_1},\ldots,\underline{n_i},y_{i+1},\ldots,y_k))\, \underbrace{((\lambda x.M)\,\underline{0}) \ldots ((\lambda x.M)\,\underline{0})}_{\textit{(k-i) fois}}.$$

Lemme 5. *Soit $M \in \mathbb{T}_{\oplus}$ avec $\vdash M : Bool$. Alors $\forall k \in \mathbb{N}$, $\forall e$ et $\epsilon \in \mathbb{Q}_2$, pour tout $i \in \{0,\ldots,k-1\}$, on a que :*

$$[\![P^{k,e,\epsilon}_{n_1,\ldots,n_i}]\!] = [\![M]\!](\underline{0}) \cdot [\![P^{k,e,\epsilon}_{n_1,\ldots,n_i,0}]\!] + [\![M]\!](\underline{1}) \cdot [\![P^{k,e,\epsilon}_{n_1,\ldots,n_i,1}]\!].$$

Démonstration. Soient $k \in \mathbb{N}, \epsilon \in \mathbb{Q}_2'$ et $i \in \{0,\ldots,k\}$, on a que :

$$\begin{aligned}
P^{k,e,\epsilon}_{n_1,\ldots,n_i} &= (\lambda y_{i+1} \ldots \lambda y_k.\underline{f_k^{e,\epsilon}}(\underline{n_1},\ldots,\underline{n_i},y_{i+1},\ldots,y_k))\,(\lambda x.M\,\underline{0})\,\underbrace{(\lambda x.M\,\underline{0}) \ldots (\lambda x.M\,\underline{0})}_{(k-(i+1))\text{ fois}} \\
&\to (\lambda y_{i+1} \ldots \lambda y_k.\underline{f_k^{e,\epsilon}}(\underline{n_1},\ldots,\underline{n_i},y_{i+1},\ldots,y_k))\,M\,\underbrace{(\lambda x.M\,\underline{0}) \ldots (\lambda x.M\,\underline{0})}_{(k-(i+1))\text{ fois}}.
\end{aligned}$$

alors d'après lemme 4 ,

$$\left[\!\left[P^{k,e,\epsilon}_{n_1,\ldots,n_i}\right]\!\right] = [\![M]\!]\,(\underline{0}) \cdot \left[\!\left[(\lambda y_{i+1}\ldots\lambda y_k.\underline{f^{e,\epsilon}_k}(\underline{n_1},\ldots,\underline{n_i},y_{i+1},\ldots,y_k))\,\underline{0}\,\underbrace{(\lambda x.M\,\underline{0})\ldots(\lambda x.M\,\underline{0})}_{(k-(i+1))\text{ fois}}\right]\!\right]$$
$$+\,[\![M]\!](\underline{1}) \cdot \left[\!\left[(\lambda y_{i+1}\ldots\lambda y_k.\underline{f^{e,\epsilon}_k}(\underline{n_1},\ldots,\underline{n_i},y_{i+1},\ldots,y_k))\,\underline{1}\,\underbrace{(\lambda x.M\,\underline{0})\ldots(\lambda x.M\,\underline{0})}_{(k-(i+1))\text{ fois}}\right]\!\right].$$

En effectuant les étapes de réduction suivantes :

$$(\lambda y_{i+1}\ldots\lambda y_k.\underline{f^{e,\epsilon}_k}(\underline{n_1},\ldots,\underline{n_i},y_{i+1},\ldots,y_k))\,\underline{0}\,\underbrace{(\lambda x.M\,\underline{0})\ldots(\lambda x.M\,\underline{0})}_{(k-(i+1))\text{ fois}}$$
$$\to (\lambda y_{i+2}\ldots\lambda y_k.\underline{f^{e,\epsilon}_k}(\underline{n_1},\ldots,\underline{n_i},\underline{0},y_{i+2},\ldots,y_k))\,\underbrace{(\lambda x.M\,\underline{0})\ldots(\lambda x.M\,\underline{0})}_{(k-(i+1))\text{ fois}}$$

$$(\lambda y_{i+1}\ldots\lambda y_k.\underline{f^{e,\epsilon}_k}(\underline{n_1},\ldots,\underline{n_i},y_{i+1},\ldots,y_k))\,\underline{1}\,\underbrace{(\lambda x.M\,\underline{0})\ldots(\lambda x.M\,\underline{0})}_{(k-(i+1))\text{ fois}}$$
$$\to (\lambda y_{i+2}\ldots\lambda y_k.\underline{f^{e,\epsilon}_k}(\underline{n_1},\ldots,\underline{n_i},\underline{1},y_{i+2},\ldots,y_k))\,\underbrace{(\lambda x.M\,\underline{0})\ldots(\lambda x.M\,\underline{0})}_{(k-(i+1))\text{ fois}}.$$

On obtient :

$$[\![P^{k,e,\epsilon}_{n_1,\ldots,n_i}]\!] = [\![M]\!](\underline{0}) \cdot [\![P^{k,e,\epsilon}_{n_1,\ldots,n_i,0}]\!] + [\![M]\!](\underline{1}) \cdot [\![P^{k,e,\epsilon}_{n_1,\ldots,n_i,1}]\!].$$

□

Revenons maintenant à démontrer la **Proposition.1** Rappelons l'ennoncé de cette proposition :
Soit $M \in \mathbb{T}_\oplus$ avec $\vdash M$: Bool . Alors $\forall k \in \mathbb{N}, \forall e$ et $\epsilon \in \mathbb{Q}_2$, on a que :

$$[\![C^{e,\epsilon}_k[M]]\!] = \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot \left\{\underline{f^{e,\epsilon}_k(n_1,\ldots,n_k)}\right\}^1.$$

Démonstration. D'abord, nous démontrons par récurrence sur k que :

$$[\![C^{e,\epsilon}_k[M]]\!] = \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot \left[\!\left[P^{k,e,\epsilon}_{n_1,\ldots,n_k}\right]\!\right].$$

Cas de base $(k = 1)$: On a :

$$[\![\mathcal{C}^{e,\epsilon}_k[M]]\!] = [\![P^{k,e,\epsilon}_{\{\}}]\!],$$

et d'après Lemme 4 on a :

$$[\![P^{k,e,\epsilon}_{\{\}}]\!] = [\![M]\!](\underline{0}) \cdot [\![P^{k,e,\epsilon}_{0}]\!] + [\![M]\!](\underline{1}) \cdot [\![P^{k,e,\epsilon}_{1}]\!].$$

Pas inductif : Supposons que pour $k \in \mathbb{N}$,

$$[\![C^{e,\epsilon}_{k}[M]]\!] = \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot [\![P^{k,e,\epsilon}_{n_1,\ldots,n_k}]\!].$$

Pour $k+1$:

$$\begin{aligned}
[\![C^{e,\epsilon}_{k}[M]]\!] &= \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot [\![P^{k+1,e,\epsilon}_{\underline{n_1},\ldots,\underline{n_k}}]\!] \\
&\overset{\text{Lemme 2}}{=} \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot \left(\underbrace{[\![M]\!](\underline{0}) \cdot [\![P^{k+1,e,\epsilon}_{n_1,\ldots,n_i,0}]\!] + [\![M]\!](\underline{1}) \cdot [\![P^{k+1,e,\epsilon}_{n_1,\ldots,n_i,1}]\!]}_{} \right) \\
&= \sum_{(n_1,\ldots,n_{k+1})\in\{0,1\}^{k+1}} \prod_{i=1}^{k+1} [\![M]\!](\underline{n_i}) \cdot [\![P^{k+1,e,\epsilon}_{n_1,\ldots,n_k,n_{k+1}}]\!].
\end{aligned}$$

Ensuite, on a que $P^{k,e,\epsilon}_{n_1,\ldots,n_k} = \underline{f^{e,\epsilon}_{k}}(\underline{n_1},\ldots,\underline{n_k})$. Alors,

$$P^{k,e,\epsilon}_{n_1,\ldots,n_k} \to^* \left\{ \underline{f^{e,\epsilon}_{k}(n_1,\ldots,n_k)} \right\}^1,$$

et donc

$$[\![P^{k,e,\epsilon}_{n_1,\ldots,n_k}]\!] = \left\{ \underline{f^{e,\epsilon}_{k}(n_1,\ldots,n_k)} \right\}^1.$$

Finallement, nous obtenons :

$$\begin{aligned}
[\![C^{e,\epsilon}_{k}[M]]\!] &= \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot [\![P^{k,e,\epsilon}_{n_1,\ldots,n_k}]\!] \\
&= \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot \left\{ \underline{f^{e,\epsilon}_{k}(n_1,\ldots,n_k)} \right\}^1.
\end{aligned}$$

□

4.2.2 Utilisation de la loi des grands nombres pour l'amplification

Dans cette sous-section, nous cherchons à adopter un formalisme de variable aléatoire qui nous permettre d'appliquer la loi des grands nombres et, par la suite, démontrer que la famille de contextes que nous avons trouvée est effectivement une famille qui amplifie la distance. Le théorème suivant est une conséquence assez directe de la proposition démontrée dans la sous-section précédente.

Théorème 3. *Soit $M \in \mathbb{T}_{\oplus}$ avec $\vdash M : Bool$.Alors $\forall k \in \mathbb{N}, \forall e$ et $\epsilon \in \mathbb{Q}_2$, on a :*

$$[\![\mathcal{C}_k^{e,\epsilon}[M]]\!] = \underline{loi(f_k^{e,\epsilon}(X_1, \ldots, X_k))}.$$

où $X_1, \ldots, X_k$ sont des variables aléatoires identiques et indépendantes suivant la loi de Bernoulli avec paramètre $\mathbb{E}(M) = e'$.

Signalons que $\forall i \in \{1, \ldots, k\}$ l'espérance de X_i, notée $\mathbb{E}(X_i)$, est calculée par l'expression : $0 \cdot \text{Proba}(X_i = 0) + 1 \cdot \text{Proba}(X_i = 1)$. Ainsi, $\mathbb{E}(X_i) = \text{Proba}(X_i = 1) = \mathbb{E}(M) = e'$.

Démonstration.

$$\begin{aligned}
[\![\mathcal{C}_k^{e,\epsilon}[M]]\!] &= \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} \prod_{i=1}^{k} [\![M]\!](\underline{n_i}) \cdot [\![P^{k,e,\epsilon}_{n_1,\ldots,n_k}]\!] \\
&= \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} e'^{\text{ nb de } i/n_i=1} \times (1-e')^{\text{nb de } i/n_i=0} \times \left\{\underline{f_k^{e,\epsilon}(n_1, \ldots, n_k)}\right\}^1.
\end{aligned}$$

Ainsi,

$$[\![\mathcal{C}_k^{e,\epsilon}[M]]\!] = \underline{{f_k^{e,\epsilon}}_*(\varepsilon)}, \quad (\text{ push forward de } \varepsilon \text{ suivant } f_k^{e,\epsilon}),$$

avec

$$\varepsilon(n_1, \ldots, n_k) = e'^{\text{ nb de } i/n_i=1} \times (1-e')^{\text{nb de } i/n_i=0}.$$

Soient $X_1, \ldots, X_k$ des variables aléatoires identiques et indépendantes suivant la loi de Bernoulli de paramètre $\mathbb{E}(M) = e'$. (on peut construire une telle suites de variables aléatoire [5]).
Alors

$$e'^{\text{ nb de } i/n_i=1} \times (1-e')^{\text{nb de } i/n_i=0} = loi(X_1, \ldots, X_k)(n_1, \ldots, n_k),$$

Enfin,

$$\begin{aligned}
[\![\mathcal{C}_k^{e,\epsilon}[M]]\!] &= \underline{{f_k^{e,\epsilon}}_*(\varepsilon)} \\
&= \underline{{f_k^{e,\epsilon}}_*(loi(X_1, \ldots, X_k))} \\
&\overset{Lemme3}{=} \underline{loi(f_k^{e,\epsilon}(X_1, \ldots, X_k))}.
\end{aligned}$$

□

Ce formalisme de variable aléatoire va maintenant nous permettre de prouver facilement que la famille de contextes $C_k^{e,\epsilon}$ — pour e et ϵ bien choisis — est effectivement une famille de contextes d'amplification pour la paire M_1, M_2. La première étape est de montrer que si nous prenons pour e l'espérance de $[\![M_1]\!]$, quelle que soit la précision ϵ que nous fixons, le comportement de $C_k^{e,\epsilon}[M_1]$ tendra vers le programme qui renvoie toujours 1. Nous formalisons cela dans la proposition ci-dessous :

Proposition 2. *Soit $M \in \mathbb{T}_\oplus$ avec $\vdash M : Bool$ et $\mathbb{E}(M) = e, \forall \epsilon \in \mathbb{Q}_2$,*

$$\lim_{k\to\infty} [\![\mathcal{C}_k^{e,\epsilon}[M]]\!](\underline{0}) = 1.$$

Démonstration. On a que $[\![\mathcal{C}_k^{e,\epsilon}[M]]\!](\underline{0}) = \underline{loi(f_k^{e,\epsilon}(X_1, \ldots, X_k))(\underline{0})}$, où X_i sont des variables aléatoires identiques et indépendantes suivant la loi de Bernoulli avec paramètre e. Ainsi, c'est la probabilité que $f_k^{e,\epsilon}(X_1, \ldots, X_k) = 0$.

On conclut que

$$[\![\mathcal{C}_k^{e,\epsilon}[M]]\!](\underline{0}) = \text{proba}\left(\left|\frac{X_1 + \cdots + X_k}{k} - e\right| \leq \epsilon\right).$$

La loi faible des grands nombres affirme que, parce que $e = \mathbb{E}(X_i)$, pour tout $\gamma > 0$,

$$\lim_{k\to\infty} \text{proba}\left(\left|\frac{X_1 + \cdots + X_k}{k} - e\right| \leq \gamma\right) = 1.$$

Par conséquent, pour tout $\epsilon \in \mathbb{Q}_2$,

$$\lim_{k\to\infty} \text{proba}\left(\left|\frac{X_1 + \cdots + X_k}{k} - e\right| \leq \epsilon\right) = 1.$$

Finalement, on a par conséquence que

$$\lim_{k\to\infty} [\![\mathcal{C}_k^{e,\epsilon}[M]]\!](\underline{0}) = \lim_{k\to\infty} \text{proba}\left(\left|\frac{X_1 + \cdots + X_k}{k} - e\right| \leq \epsilon\right) = 1.$$

□

Nous pouvons maintenant, pour une paire de programmes M_1 et M_2, prendre e comme l'espérance de $[\![M_1]\!]$. Cependant, nous allons maintenant choisir la précision ϵ en fonction de M_1 et M_2. Ensuite, nous montrerons que, pour ϵ bien choisi, on obtient une famille de contextes d'amplification pour M_1 et M_2. La preuve de la proposition 3 nécessite une nouvelle application de la loi des grands nombres.

Proposition 3. *Soit M_1 et $M_2 \in \mathbb{T}_\oplus$ tels que $\vdash M_1, M_2 : Bool$ avec $\mathbb{E}(M_1) = e_1$ et $\mathbb{E}(M_2) = e_2$, tels que M_1 et M_2 ne sont pas équivalents. Alors $\forall \epsilon \in \mathbb{Q}_2$, tel que $\epsilon < \frac{|e_1 - e_2|}{3}$, on a :*

$$\lim_{k\to\infty} [\![\mathcal{C}_k^{e_1,\epsilon}[M_1]]\!](\underline{0}) = 1,$$

et

$$\lim_{k\to\infty} [\![\mathcal{C}_k^{e_1,\epsilon}[M_2]]\!](\underline{0}) = 0.$$

Démonstration. D'après Proposition 2. , on a :

$$\lim_{k\to\infty} [\![\mathcal{C}_k^{e_1,\epsilon}[M_1]]\!](\underline{0}) = 1,$$

et

$$\lim_{k\to\infty} [\![\mathcal{C}_k^{e_1,\epsilon}[M_2]]\!](\underline{0}) = \lim_{k\to\infty} \text{proba}\left(\left|\frac{X'_1 + \cdots + X'_k}{k} - e_1\right| \leq \epsilon\right),$$

avec les X'_i sont des variables aléatoires identiques et indépendantes suivant la loi de Bernoulli avec paramètre e_2. En utilisant la loi faible des grands nombres, et sachant que l'espérance $\mathbb{E}(X'_i) = e_2$, on conclut que

$$\lim_{k\to\infty} \text{proba}\left(\left|\frac{X'_1 + \cdots + X'_k}{k} - e_2\right| \leq \epsilon\right) = 1,$$

On a que les événements

$$\underbrace{\left\{\left|\frac{X'_1 + \cdots + X'_k}{k} - e_1\right| \leq \epsilon\right\}}_{A} \cap \underbrace{\left\{\left|\frac{X'_1 + \cdots + X'_k}{k} - e_2\right| \leq \epsilon\right\}}_{B} = \emptyset.$$

Puisque A et B sont deux événements disjoints, on a $\text{proba}(A \cap B) = 0$, et donc $A \subseteq B^c$. Ainsi, $\text{proba}(A) \leq \text{proba}(B^c) = 1 - \text{proba}(B)$,

$$\text{proba}\left(\left|\frac{X'_1 + \cdots + X'_k}{k} - e_1\right| \leq \epsilon\right) \leq 1 - \text{proba}\left(\left|\frac{X'_1 + \cdots + X'_k}{k} - e_2\right| \leq \epsilon\right).$$

En faisant k tendre vers l'infini, on trouve que

$$\lim_{k\to\infty} \text{proba}\left(\left|\frac{X'_1 + \cdots + X'_k}{k} - e_1\right| \leq \epsilon\right) = 0.$$

Donc,

$$\lim_{k\to\infty} [\![\mathcal{C}_k^{e_1,\epsilon}[M_2]]\!](\underline{0}) = 0.$$

□

Corollaire 1. *Soit M_1 et $M_2 \in \mathbb{T}_\oplus$ avec $\vdash M_1, M_2 : Bool$ avec $\mathbb{E}(M_1) = e_1$ et $\mathbb{E}(M_2) = e_2$, tels que M_1 et M_2 ne sont pas équivalents. Alors $\forall \epsilon \in \mathbb{Q}_2$, tel que $\epsilon < \frac{|e_1 - e_2|}{3}$, on a :*

$$(\delta^{ctx}_{\mathbb{T}_\oplus})_\alpha(M_1, M_2) = 1.$$

Démonstration. Soit $\epsilon \in \mathbb{Q}_2$, tel que $\epsilon < \frac{|e_1-e_2|}{3}$

$$\begin{aligned}(\delta^{ctx}_{\mathbb{T}^\oplus})_\alpha(M_1, M_2) &= \sup_{\mathcal{C}\,\text{s.t. }[.]:\alpha\vdash\mathcal{C}:\,\text{Nat}} |[\![\mathcal{C}[M_1]]\!](\underline{0}) - [\![\mathcal{C}[M_2]]\!](\underline{0})| \\ &\geq \lim_{k\to\infty} |[\![\mathcal{C}_k^{e_1,\epsilon}[M_1]]\!](\underline{0}) - [\![\mathcal{C}_k^{e_1,\epsilon}[M_2]]\!](\underline{0})| \\ &\overset{\text{proposition 2,3}}{=} 1.\end{aligned}$$

□

4.2.3 Trivialisation dans $\mathbb{T}_\oplus$

Dans cette sous-section, similairement au [3], on va se servir de la famille de contextes d'amplification pour rendre notre preuve complète et démontrer la trivialisation.

On utilise maintenant le résultat du Corollaire 1. pour démontrer la trivialisation pour le langage $\mathbb{T}_\oplus$. L'ingrédient clé ici est que, s'il existe un contexte qui sépare M_1 et M_2, alors on peut construire un contexte qui sépare M_1 et M_2 et qui, de plus, renvoie soit 0 soit 1, c'est-à-dire essentiellement un contexte qui renvoie un booléen. On peut ensuite composer ce contexte avec la famille de contextes d'amplification que nous avons construite.

Lemme 6. *Soit M_1 et $M_2 \in \mathbb{T}_\oplus$ avec $\vdash M_1, M_2 : \alpha$, tels que M_1 et M_2 ne sont pas équivalents , alors :*

$$(\delta^{ctx}_{\mathbb{T}_\oplus})_\alpha(M_1, M_2) = 1.$$

Démonstration. On suppose $M_1 \not\equiv_\alpha M_2$. On sait déjà que $(\delta^{ctx}_{\mathbb{T}^\oplus})_\alpha(M_1, M_2) > 0$.
En conséquence, selon la definition de $(\delta^{ctx}_{\mathbb{T}^\oplus})_\alpha(M_1, M_2)$, $\exists \delta > 0$ et $\exists \mathcal{D}$ s.t. $[.] : \sigma \vdash \mathcal{D} : \text{Nat}$ tel que

$$|[\![\mathcal{D}[M_1]]\!](\underline{0}) - [\![\mathcal{D}[M_2]]\!](\underline{0})| = \delta > 0.$$

On considère le contexte suivant :

$$\mathcal{C}[\cdot] = \text{if}\, \mathcal{D}[\cdot]\, \underline{0}\, \text{succ}(\underline{0}).$$

Ce contexte prend en argument un programme M et renvoie $\underline{0}$ si $\mathcal{D}[M]$ renvoie $\underline{0}$, et renvoie $\underline{1}$ si $\mathcal{D}[M]$ renvoie un entier différent de $\underline{0}$. (Pour plus d'informations sur la construction de if, voir A).

Cela implique que $\mathcal{C}[M_1]$ et $\mathcal{C}[M_2]$ sont tq. $\vdash M_1, M_2 :$ Bool avec $\mathbb{E}(\mathcal{C}[M_1]) = e_1 \in \mathbb{Q}_2$ et $\mathbb{E}(\mathcal{C}[M_2]) = e_2 \in \mathbb{Q}_2$. Alors $\forall \epsilon \in \mathbb{Q}_2$, tel que $\epsilon < \frac{|e_1 - e_2|}{3}$, on a d'après Corollaire 1. , que :

$$\lim_{k\to\infty} |[\![\mathcal{C}^{e_1,\epsilon}_k[\mathcal{C}[M_1]]]\!](\underline{0}) - [\![\mathcal{C}^{e_1,\epsilon}_k[\mathcal{C}[M_2]]]\!](\underline{0})| = 1.$$

Considérons la famille de contextes de M_1 et M_2, définie comme suit :

$$\begin{aligned}\mathcal{T}^{e_1,\epsilon}_k[\cdot] &= C^{e_1,\epsilon}_k[\mathcal{C}[\cdot]] \\ &= \lambda x.((\lambda y_1 \ldots \lambda y_k.\underline{f^{e_1,\epsilon}_k}(y_1, \ldots, y_k))\underbrace{(x\,\underline{0}) \ldots (x\,\underline{0})}_{k\,fois})\lambda x.(\mathcal{C}[\cdot]).\end{aligned}$$

Enfin, nous obtenons :

$$\begin{aligned}(\delta^{ctx}_{\mathbb{T}^\oplus})_\alpha(M_1, M_2) &= \sup_{\mathcal{T}\,\text{s.t.}\,[.]:\alpha\vdash\mathcal{T}:\text{Nat}} |[\![\mathcal{T}[M_1]]\!](\underline{0}) - [\![\mathcal{T}[M_2]]\!](\underline{0})| \\ &\geq \lim_{k\to\infty} |[\![\mathcal{T}^{e_1,\epsilon}_k[M_1]]\!](\underline{0}) - [\![\mathcal{T}^{e_1,\epsilon}_k[M_2]]\!](\underline{0})| \\ &= \lim_{k\to\infty} |[\![\mathcal{C}^{e_1,\epsilon}_k[\mathcal{C}[M_1]]]\!](\underline{0}) - [\![\mathcal{C}^{e_1,\epsilon}_k[\mathcal{C}[M_2]]]\!](\underline{0})| \\ &= 1.\end{aligned}$$

□

Théorème 4. *La distance $(\delta^{ctx}_{\mathbb{T}^\oplus})_\alpha$ trivialise.*

Démonstration. Soient M_1 et M_2 dans $\mathbb{T}^{\oplus}$ de type α. Si M_1 est équivalent à M_2, alors

$$(\delta^{ctx}_{\mathbb{T}^{\oplus}})_{\alpha}(M_1, M_2) = 0.$$

Si M_1 n'est pas équivalent à M_2, alors d'après Corollaire 1. et Lemme 6.,

$$(\delta^{ctx}_{\mathbb{T}^{\oplus}})_{\alpha}(M_1, M_2) = 1.$$

□

5 Conclusion

Notre preuve permet de mieux comprendre le théorème de trivialisation en termes de théorie des probabilités : dès que le langage que nous considérons est suffisamment expressif pour que les contextes puissent calculer la moyenne empirique de la distribution de probabilité qu'ils prennent en argument et ensuite tester sa proximité avec une valeur cible, alors ils sont capables de distinguer **avec une probabilité de 1** toute paire de distributions ayant des espérances différentes. C'est parce que la moyenne empirique tend vers l'espérance, d'après la loi faible de grands nombres, et tester la moyenne empirique est donc essentiellement la même chose que tester l'espérance.
Cela ouvre également des directions de recherche intéressantes concernant le taux de convergence de la distance contextuelle vers la métrique triviale lorsque nous autorisons les contextes à utiliser un nombre limité, mais croissant, de copies de leur argument. En effet, le taux de convergence de la loi faible des grands nombres est très bien étudié mathématiquement, et nous pourrions potentiellement extraire de là des informations sur le taux de convergence de la distance contextuelle.

A Annexe

A.1 Encodage dans le système $\mathbb{T}$

Soient $k \in \mathbb{N}, e$ et $\epsilon \in Q_2$. Le but de cette section est de trouver un encodage de la fonction $f_k^{e,\epsilon}$ définie de la manière suivante :

$$f_k^{e,\epsilon} : \mathbb{N}^k \to \{0,1\}$$

où

$$f_k^{e,\epsilon}(n_1, n_2, \ldots, n_k) = \begin{cases} 0 & \text{si } \left|\frac{\sum_{i=1}^k n_i}{k} - e\right| \leq \epsilon, \\ 1 & \text{sinon} \end{cases}$$

dans le système $\mathbb{T}$.
Ceci est équivalent à trouver l'encodage de :

$$f_k^{t,p,r,s}(n_1, n_2, \ldots, n_k) = \begin{cases} 0 & \text{si } 2^s \times \left|2^p \times \sum_{i=1}^k n_i - kt\right| \leq r \times 2^p \times k, \\ 1 & \text{sinon} \end{cases}$$

où t, p, r et $s \in \mathbb{N}$ tels que $e = \frac{t}{2^p}$ et $\epsilon = \frac{r}{2^s}$.

Pour atteindre cet objectif, nous avons besoin de l'encodage de l'addition, de la soustraction, de la multiplication, du if, de la valeur absolue, de l'opérateur $\leq$ et de EXP.

Définition 21. *On définit les expressions suivantes dans le système $\mathbb{T}$:*

$$add := \lambda m.\lambda n.\, rec\langle\langle m, \lambda k.\lambda x.\, succ(x)\rangle, n\rangle.$$

$$mult := \lambda m.\lambda n.\, rec\langle\langle 0, \lambda k.\lambda x.\, add\, m\, x\rangle, n\rangle.$$

$$pred := \lambda n.\, rec\langle\langle 0, \lambda y.\lambda x.succ(x)\rangle, n\rangle.$$

$$minus := \lambda m.\lambda n.\, rec\langle\langle m, \lambda k.\lambda x.\, pred\, x\rangle, n\rangle.$$

$$if := \lambda M.\lambda N.\lambda P.\, rec\langle\langle N, \lambda k.\lambda x.P\rangle, M\rangle.$$

$$leq := \lambda m.\lambda n.if\ (minus\, m\, n)\ suc(\underline{0})\ \underline{0}.$$

$$abs := \lambda m.\lambda n.if\ (minus\, m\, n)\ (minus\, n\, m)\ (minus\, m\, n)$$

Lemme 7. *$\forall m, n \in \mathbb{N}$, $add\, \underline{m}\, \underline{n} \to^* \underline{m+n}$, et parsuite add est l'encodage de l'addition.*

Démonstration. Nous démontrons ce lemme par récurrence sur n :
Cas de base $(n = 0)$:

$$add\, \underline{m}\, \underline{0} \to rec\langle\langle \underline{m}, \lambda k.\lambda x.\, succ(x)\rangle, \underline{0}\rangle \to \underline{m} = \underline{m+0}.$$

Pas inductif :
Supposons que pour $n \in \mathbb{N}$, $add\, \underline{m}\, \underline{n} \to^* \underline{m+n}$ soit vrai.

Pour $n+1$:

$$\begin{aligned} add\,\underline{m}\,\underline{n+1} &\to rec\langle\langle \underline{m}, \lambda k.\lambda x.\, succ(x)\rangle, succ(\underline{n})\rangle \\ &\to ((\lambda k.\lambda x.\, succ(x))\underline{n})(rec\langle\langle \underline{m}, \lambda k.\lambda x.\, succ(x)\rangle, \underline{n}\rangle) \\ &\to (\lambda x.\, succ(x))(rec\langle\langle \underline{m}, \lambda k.\lambda x.\, succ(x)\rangle, \underline{n}\rangle) \\ &\to^* (\lambda x.\, succ(x))\underline{m+n} \\ &\to succ(\underline{m+n}) \\ &= \underline{m+(n+1)}. \end{aligned}$$

□

Lemme 8. *$\forall m, n \in \mathbb{N}$, $mult\,\underline{m}\,\underline{n} \to^* \underline{m*n}$, et parsuite mult est l'encodage de la multiplication.*

Démonstration. Nous démontrons ce lemme par récurrence sur n :
Cas de base $(n = 0)$:

$$mult\,\underline{m}\,\underline{0} \to rec\langle\langle \underline{0}, \lambda k.\lambda x.\, add\,\underline{m}\,x\rangle, \underline{0}\rangle \to \underline{0} = \underline{m*0}.$$

Pas inductif :
Supposons que pour $n \in \mathbb{N}$, $mult\,\underline{m}\,\underline{n} \to^* \underline{m*n}$ soit vrai.
Pour $n+1$:

$$\begin{aligned} mult\,\underline{m}\,\underline{n+1} &\to rec\langle\langle \underline{m}, \lambda k.\lambda x.\, add\,\underline{m}\,x\rangle, succ(\underline{n})\rangle \\ &\to ((\lambda k.\lambda x.\, add\,\underline{m}\,x)\underline{n})(rec\langle\langle \underline{m}, \lambda k.\lambda x.\, add\,\underline{m}\,x\rangle, \underline{n}\rangle) \\ &\to (\lambda x.\, add\,\underline{m}\,x)(rec\langle\langle \underline{m}, \lambda k.\lambda x.\, add\,\underline{m}\,x\rangle, \underline{n}\rangle) \\ &\to^* (\lambda x.\, add\,\underline{m}\,x)\underline{m*n} \\ &\to add\,\underline{m}\,\underline{m*n} \\ &= \underline{m*(n+1)}. \end{aligned}$$

□

Définition 22. *Pour tout s appartenant à $\mathbb{N}$, soit EXP_s définie par la multiplication répétée de 2, s fois, c'est-à-dire :*

$$EXP_s = \underbrace{mult \ldots mult\,(mult\,\underline{1}\,\underline{2})\,\underline{2}\ldots\underline{2}}_{s\ fois}.$$

Lemme 9. *D'après (Lemme 8), $\forall s \in \mathbb{N}$ on a $EXP_s \to^* \underline{2^s}$, donc EXP_s est l'encodage de 2^s.*

Lemme 10. *$\forall n \in \mathbb{N}$, $pred\,\underline{n} \to^* \underline{predecessor(n)}$, et parsuite pred est l'encodage de predecessor, où la fonction predecessor est définie de cette manière : $predecessor : \mathbb{N} \to \mathbb{N}$, telle que :*

$$predecessor(n) = \begin{cases} 0 & si\ n = 0, \\ n-1 & sinon. \end{cases}$$

Démonstration. Nous démontrons ce lemme par récurrence sur n :
Cas de base $(n = 0)$:

$$pred\,\underline{0} \to rec\langle\langle\underline{0}, \lambda y.\lambda x.succ(x)\rangle, \underline{0}\rangle \to \underline{0} = \underline{predecessor(0)}.$$

Pas inductif :
Supposons que pour $n \in \mathbb{N}$, $pred\,\underline{n} \to^* \underline{predecessor(n)} = \underline{n-1}$ soit vrai.
Pour $n+1$:

$$\begin{aligned}
pred\,\underline{n+1} &\to rec\langle\langle\underline{0}, \lambda y.\lambda x.succ(x)\rangle, \underline{n+1}\rangle \\
&\to ((\lambda y.\lambda x.succ(x))\underline{n})(rec\langle\langle\underline{0}, \lambda y.\lambda x.succ(x)\rangle, \underline{n}\rangle) \\
&\to (\lambda x.succ(x))(rec\langle\langle\underline{0}, \lambda y.\lambda x.y\rangle, \underline{n}\rangle) \\
&\to^* (\lambda x.succ(x))\underline{predecessor(n)} \\
&\to succ(\underline{predecessor(n)}) \\
&\to succ(\underline{n-1}) \\
&\to \underline{n} \\
&\to \underline{predecessor(n+1)}.
\end{aligned}$$

□

Lemme 11. *$\forall m, n \in \mathbb{N}$, $minus\,\underline{m}\,\underline{n} \to^* \underline{moins(m,n)}$, et parsuite minus est l'encodage de la fonction moins (la soustraction), où la fonction moins est définie de cette manière : $moins : \mathbb{N} \to \mathbb{N}$, telle que :*

$$moins(m,n) = \begin{cases} 0 & si\ m-n \leq 0, \\ m-n & sinon. \end{cases}$$

Démonstration. Nous démontrons ce lemme par récurrence sur n :
Cas de base $(n = 0)$:

$$minus\,\underline{m}\,\underline{0} \to rec\langle\langle\underline{m}, \lambda k.\lambda x.\,pred\,x\rangle, \underline{0}\rangle \to \underline{m} = \underline{moins(m,0)}.$$

Pas inductif :
Supposons que pour $n \in \mathbb{N}$, $minus\,\underline{m}\,\underline{n} \to^* \underline{moins(m,n)}$ soit vrai.
Pour $n+1$:

$$\begin{aligned}
minus\,\underline{m}\,\underline{n+1} &\to rec\langle\langle\underline{m}, \lambda k.\lambda x.\,pred\,x\rangle, succ(\underline{n})\rangle \\
&\to ((\lambda k.\lambda x.\,pred\,x)\underline{n})(rec\langle\langle\underline{m}, \lambda k.\lambda x.\,pred\,x)\rangle, \underline{n}\rangle) \\
&\to (\lambda x.\,pred\,x)(rec\langle\langle\underline{m}, \lambda k.\lambda x.\,pred\,x\rangle, \underline{n}\rangle) \\
&\to^* (\lambda x.\,pred\,x)\underline{moins(m,n)} \\
&\to pred\,\underline{moins(m,n)} \\
&= \begin{cases} pred\,\underline{0} \to \underline{0} = \underline{moins(m,n+1)} & \text{si } m-n \leq 0, \\ pred\,\underline{m-n} \to^* \underline{m-(n+1)} = \underline{moins(m,n+1)} & \text{sinon.} \end{cases}
\end{aligned}$$

□

Lemme 12. *Soit M, N et P dans $\mathbb{T}$ telle que M est de type Nat, alors :*

$$if M\,N\,P \to^* \begin{cases} N & si\ M \to^* \underline{0}, \\ P & si\ M \to^* succ(\underline{n})\ pour\ un\ certain\ n \in \mathbb{N}. \end{cases}$$

Démonstration. Considérons deux cas :

Cas 1 : Si $M \to^* 0$, alors :

$$\begin{aligned} \text{if } M\,N\,P &\to rec\langle\langle N, \lambda k.\lambda x.P\rangle, M\rangle \\ &\to^* rec\langle\langle N, \lambda k.\lambda x.P\rangle, \underline{0}\rangle \\ &\to N. \end{aligned}$$

Cas 2 : Si $M \to^* succ(\underline{n})$, nous procédons par récurrence sur $n \in \mathbb{N}$.

Cas de base $(n = 0)$:

$$\begin{aligned} if M\,N\,P &\to^* rec\langle\langle N, \lambda k.\lambda x.P\rangle, succ(\underline{0})\rangle \\ &\to ((\lambda k.\lambda x.P)(\underline{0}))rec\langle\langle N, \lambda k.\lambda x.P\rangle, \underline{0}\rangle \\ &\to (\lambda x.P)(\underline{0}) \\ &\to P. \end{aligned}$$

Pas inductif :

Supposons que pour $n \in \mathbb{N}$, $if MNP \to^* N$ soit vrai, si $M \to^* succ(\underline{n})$.

Pour $n+1$: $M \to^* succ(\underline{n+1})$.

$$\begin{aligned} if M\,N\,P &\to^* rec\langle\langle N, \lambda k.\lambda x.P\rangle, succ(\underline{n+1})\rangle \\ &\to ((\lambda k.\lambda x.P)(\underline{n+1}))rec\langle\langle N, \lambda k.\lambda x.P\rangle, \underline{n+1}\rangle \\ &= (\lambda x.P)rec\langle\langle N, \lambda k.\lambda x.P\rangle, suc(\underline{n})\rangle \\ &\to^* (\lambda x.P)P \\ &= P. \end{aligned}$$

□

Lemme 13. *L'expression leq est l'encodage de $\leq$, où $\leq (m, n)$, telle que pour tous $m, n \in \mathbb{N}$, est défini par :*

$$\leq (m, n) = \begin{cases} 1 & si\ moins(m, n) = 0, \\ 0 & sinon. \end{cases}$$

Démonstration. Soit $m, n \in \mathbb{N}$

$$\begin{aligned} leq\,\underline{m}\,\underline{n} &\to \text{if } (minus\,\underline{m}\,\underline{n})\ succ(\underline{0})\,\underline{0} \\ &\to^* \begin{cases} succ(\underline{0}) & \text{si } minus\,\underline{m}\,\underline{n} \to^* \underline{0}, \\ \underline{0} & \text{sinon} \end{cases} \\ &= \begin{cases} \underline{1} & \text{si } \underline{moins(m, n)} = \underline{0}, \\ \underline{0} & \text{sinon} \end{cases} \\ &= \underline{\leq (m, n)}. \end{aligned}$$

□

Lemme 14. *L'expression abs est l'encodage de la fonction $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, telle que $f(m,n)$ est défini par :*

$$f(m,n) = \begin{cases} moins(n,m) & si\ moins(m,n) = 0, \\ moins(m,n) & sinon. \end{cases}$$

Démonstration. Soit $m, n \in \mathbb{N}$ l'expression $abs\,\underline{m}\,\underline{n}$ se réduit de la manière suivante :

$$\begin{aligned} abs\,\underline{m}\,\underline{n} &\to \text{if } (minus\,\underline{m}\,\underline{n})\,(minus\,\underline{n}\,\underline{m})\,(minus\,\underline{m}\,\underline{n}) \\ &\to^* \begin{cases} minus\,\underline{n}\,\underline{m} & \text{si } (minus\,\underline{m}\,\underline{n}) \to^* \underline{0}, \\ minus\,\underline{m}\,\underline{n} & \text{sinon,} \end{cases} \\ &= \begin{cases} \underline{moins(n,m)} & \text{si } \underline{moins(m,n)} = \underline{0}, \\ \underline{moins(m,n)} & \text{sinon,} \end{cases} \\ &= \underline{f(m,n)}. \end{aligned}$$

□

Définition 23. *Étant donnés trois entiers naturels t, p,r et s, on définit les expressions suivantes :*

$$A_k^p := \lambda n_1 \ldots \lambda n_k.\, mult\, EXP_p\, \underbrace{(add(\ldots(add(add}_{n_k - 1\ fois\ add}\ n_1\, n_2)\, n_3) \ldots n_k.$$

$$B_k^t := mult\,\underline{k}\,\underline{t}.$$

La fonction $\underline{f_k^{t,p,r,s}}(\underline{n_1}, \ldots, \underline{n_k})$, telle que $(n_1, \ldots, n_k) \in \mathbb{N}^k$, est définie par :

$$\underline{f_k^{t,p,r,s}}(\underline{n_1}, \ldots, \underline{n_k}) :=$$
$$if\ \left(leq\, \underbrace{(mult\, EXP_s\ (abs\, A_k^p(\underline{n_1}.\ldots.\underline{n_k})\, B_k^t))}\ \underbrace{(mult\,\underline{r}\,(mult\,\underline{k}\, EXP_p))}\right)\,(succ(\underline{0}))\,(\underline{0})\,.$$

Théorème 5. *L'expression $\underline{f_k^{e,\epsilon}} = \underline{f_k^{t,p,r,s}}$ est l'encodage de la fonction $f_k^{e,\epsilon} = f_k^{t,p,r,s}$ définie au début de cette section.*

Démonstration. Nous avons comme résultat direct de tous les lemmes précédents de cette section que : $\underline{f_k^{e,\epsilon}}(\underline{n_1}, \ldots, \underline{n_k}) = \underline{f_k^{t,p,r,s}}(\underline{n_1}, \ldots, \underline{n_k}) \to^* \underline{f_k^{t,p,r,s}(n_1, \ldots, n_k)} = \underline{f_k^{e,\epsilon}(n_1, \ldots, n_k)}$, $\forall (n_1, \ldots, n_k) \in \mathbb{N}^k$.

□

A.2 Encodage dans le système $\mathbb{T}_\oplus$

Lemme 15. *Soit M un terme dans $\mathbb{T}$ tel que $M \to^* V$ où V est une forme normale. Alors, dans $\mathbb{T}_\oplus$, on a que $M \to^* \{V\}^1$ et par conséquent, on a que $[\![M]\!] = \{V\}^1$.*

Démonstration. D'après [2] .

□

Lemme 16. *Soient $k \in \mathbb{N}, e \in \mathbb{Q}_2$ et $\epsilon \in \mathbb{Q}_2$, on a que :*

$$\underline{f_k^{e,\epsilon}}(\underline{n_1}, \ldots, \underline{n_k}) \to \left\{\underline{f_k^{e,\epsilon}(n_1, \ldots, n_k)}\right\}^1.$$

Démonstration. Application directe du lemme 15.

□

Références

[1] U. Dal Lago and M. Zorzi. Probabilistic operational semantics for the lambda cal culus. RAIRO- Theor. Inf. and Applic., 46(3) :413-450, 2012.

[2] F. Breuvart, U. D. Lago, and A. Herrou. On higher-order probabilistic subrecursion. In Foundations of Software Science and Computation Structures- 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, pages 370-386, 2017.

[3] R. Crubillé and U. Dal Lago. Metric reasoning about lambda-terms : The affine case. In Proc. of LICS, pages 633–644, 2015.

[4] R. Crubille and U. Dal Lago. Metric reasoning about lambda-terms : The general case. In European Symposium on Programming, pages 341-367. Springer, 2017.

[5] Bogachev, Vladimir I., Measure Theory, Berlin : Springer Verlag, ISBN 9783540345138, 2007.

[6] Oscar Sheynin, On the Law of Large Numbers, page 5, traduction de Ars Conjectandi de Jacques Bernoulli.

[7] U. Dal Lago, N. Hoshino, and P. Pistone. On the lattice of program metrics. In M. Gaboardi and F. van Raamsdonk, editors, 8th International Conference on Formal Structures for Computation and Deduction (FSCD 2023), July 3-6, 2023, Rome, Italy, volume 260 of LIPIcs, pages 20 :1–20 :19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[8] T. Ehrhard. Differentials and distances in probabilistic coherence spaces. Log. Methods Comput. Sci., 18(3), 2022.

[9] D. Gebler, K. G. Larsen, and S. Tini. Compositional metric reasoning with probabilistic process calculi. In Proc. of FoSSaCS, pages 230–245, 2015.

[10] J. H. Morris Jr. Lambda-calculus models of programming languages. PhD thesis, Massachusetts Institute of Technology, 1969.

[11] J. Bernoulli. Ars coniectandi. Impensis Thurnisiorum, fratrum, 1713.