

Annexe de l'article : Filtrer sans s'appauvrir : inférer les paramètres constants des modèles réactifs probabilistes

Dans cette annexe, nous présentons tout d'abord les éléments de la sémantique de ProbZelus qui nous permettent ensuite de démontrer le Théorème 1 page 23.

A Sémantique de ProbZelus

La sémantique de ProbZelus est tirée de [2, Appendix A] avec les adaptations suivantes.

Variables globales. La définition d'une variable globale `let $x = e$` associe à la variable x la valeur obtenue en exécutant la fonction de transition de l'expression sur l'état initial. L'état d'arrivée n'a pas d'importance, seule la première valeur est retenue.

Expressions dans les équations `init`. La sémantique d'un ensemble d'équations mutuellement récursives est proche de la sémantique originale. La première partie de l'état stocke des mémoires qui contiennent la valeur calculée à l'instant précédent pour toutes les variables déclarées avec un `init`. Le reste de l'état correspond aux états de toutes les sous-expressions. À la différence de la sémantique originale, ici l'état des mémoires est représenté par un type option *None/Some*. La fonction de transition commence par récupérer la valeur des mémoires. Si c'est *None*, les mémoires ne sont pas encore initialisées. Comme pour les variables globales, on exécute alors un pas de chaque fonction de transition pour calculer les premières valeurs c_1, \dots, c_k . Sinon, on récupère simplement c_1, \dots, c_k . Comme dans la sémantique originale, on peut ensuite peupler l'environnement courant avec (1) des variables spéciales x_i_last qui permettent d'accéder à la valeur des x_i à l'instant précédent via l'opérateur `last`, et (2) la définition des y_i en exécutant chacune des sous-expressions. On peut enfin calculer l'expression principale dans l'environnement obtenu.

Symétriquement dans la sémantique probabiliste, les définitions locales sont interprétées en intégrant successivement les mesures des sous-expressions pour explorer toutes les exécutions possibles. Comme dans le cas déterministe, on commence par récupérer la valeur des mémoires avant de peupler l'environnement avec les valeurs de x_i_last et y_i pour enfin évaluer l'expression principale. On note $\int \mu(dv, ds) f(v, s)$ l'intégrale de f le long de la mesure μ , et $\delta(v)$ la mesure de Dirac ($\delta(v)(U) = 1$ si $v \in U$ et 0 sinon).

$$\begin{aligned}
\llbracket x \rrbracket_{\gamma}^{\text{init}} &= () \\
\llbracket x \rrbracket_{\gamma}^{\text{step}} &= \lambda s. (\gamma(x), s) \\
\llbracket \text{last } x \rrbracket_{\gamma}^{\text{init}} &= () \\
\llbracket \text{last } x \rrbracket_{\gamma}^{\text{step}} &= \lambda s. (\gamma(x_{\text{last}}), s) \\
\llbracket (e_1, e_2) \rrbracket_{\gamma}^{\text{init}} &= (\llbracket e_1 \rrbracket_{\gamma}^{\text{init}}, \llbracket e_2 \rrbracket_{\gamma}^{\text{init}}) \\
\llbracket (e_1, e_2) \rrbracket_{\gamma}^{\text{step}} &= \lambda(s_1, s_2). \text{let } v_1, s'_1 = \llbracket e_1 \rrbracket_{\gamma}^{\text{step}}(s_1) \text{ in} \\
&\quad \text{let } v_2, s'_2 = \llbracket e_2 \rrbracket_{\gamma}^{\text{step}}(s_2) \text{ in } ((v_1, v_2), (s'_1, s'_2)) \\
\llbracket e \text{ where rec init } x_1 = e_1 \dots \text{ and init } x_k = e_k \rrbracket_{\gamma}^{\text{init}} &= (None, (\llbracket e'_1 \rrbracket_{\gamma}^{\text{init}}, \dots, \llbracket e'_n \rrbracket_{\gamma}^{\text{init}}), \llbracket e \rrbracket_{\gamma}^{\text{init}}) \\
\llbracket e \text{ where rec init } x_1 = e_1 \dots \text{ and init } x_k = e_k \rrbracket_{\gamma}^{\text{step}} &= \left[\begin{aligned} &= \lambda(m, (s_1, \dots, s_n), s). \\ &\quad \text{let } c_1, \dots, c_k = \\ &\quad \quad \text{match } m \text{ with} \\ &\quad \quad | \text{Some}(c_1, \dots, c_k) \rightarrow (c_1, \dots, c_k) \\ &\quad \quad | \text{None} \rightarrow \text{let } s_1 = \llbracket e_1 \rrbracket_{\gamma}^i \text{ in let } v_1, _ = \llbracket e_1 \rrbracket_{\gamma}^{\text{step}}(s_1) \text{ in } \dots \\ &\quad \quad \quad \text{let } s_k = \llbracket e_k \rrbracket_{\gamma}^i \text{ in let } v_k, _ = \llbracket e_k \rrbracket_{\gamma}^{\text{step}}(s_k) \text{ in} \\ &\quad \quad \quad (v_1, \dots, v_k) \end{aligned} \right] \text{ évaluer les inits} \\
&\quad \text{in} \\
&\quad \text{let } \gamma_1 = \gamma[x_{1_last} \leftarrow c_1] \text{ in } \dots \\
&\quad \text{let } \gamma_k = \gamma_{k-1}[x_{k_last} \leftarrow c_k] \text{ in} \\
&\quad \text{let } v_1, s'_1 = \llbracket e'_1 \rrbracket_{\gamma_k}^{\text{step}}(s_1) \text{ in let } \gamma'_1 = \gamma_k[y_1 \leftarrow v_1] \text{ in } \dots \\
&\quad \text{let } v_n, s'_n = \llbracket e'_n \rrbracket_{\gamma'_{n-1}}^{\text{step}}(s_n) \text{ in let } \gamma'_n = \gamma'_{n-1}[y_n \leftarrow v_n] \text{ in} \\
&\quad \text{let } v, s' = \llbracket e \rrbracket_{\gamma'_n}^{\text{step}}(s) \text{ in} \\
&\quad (v, (\text{Some}(\gamma'_n(x_1), \dots, \gamma'_n(x_k)), (s'_1, \dots, s'_n), s')) \end{aligned} \right] \text{ peupler l'environnement} \\
&\quad \text{résultat}
\end{aligned}$$

$$\begin{aligned}
\llbracket \text{infer}(e) \rrbracket_{\gamma}^{\text{init}} &= \delta(\llbracket e \rrbracket_{\gamma}^{\text{init}}) \\
\llbracket \text{infer}(e) \rrbracket_{\gamma}^{\text{step}} &= \lambda \sigma. \\
&\quad \text{let } \mu = \lambda U. \int \sigma(ds) \llbracket e \rrbracket_{\gamma}^{\text{step}}(s)(U) \text{ in} \\
&\quad \text{let } \nu = \lambda U. \mu(U) / \mu(\top) \text{ in} \\
&\quad (\pi_{1*}(\nu), \pi_{2*}(\nu)) \\
\llbracket \text{let } x = e \rrbracket_{\gamma} &= \gamma[(x, _) \leftarrow \llbracket e \rrbracket_{\gamma}^{\text{step}}(\llbracket e \rrbracket_{\gamma}^{\text{init}})] \\
\llbracket \text{let node } f \text{ } x = e \rrbracket_{\gamma} &= \gamma[f_init \leftarrow \llbracket e \rrbracket_{\gamma}^{\text{init}}] \left[f_step \leftarrow \lambda v. \lambda s. \llbracket e \rrbracket_{\gamma[x \leftarrow v]}^{\text{step}} \right] \\
\llbracket \text{let proba } f \text{ } x = e \rrbracket_{\gamma} &= \gamma[f_init \leftarrow \llbracket e \rrbracket_{\gamma}^{\text{init}}] \left[f_step \leftarrow \lambda v. \lambda s. \llbracket e \rrbracket_{\gamma[x \leftarrow v]}^{\text{step}} \right] \\
\llbracket d_1 \ d_2 \rrbracket_{\gamma} &= \text{let } \gamma_1 = \llbracket d_1 \rrbracket_{\gamma} \text{ in } \llbracket d_2 \rrbracket_{\gamma_1}
\end{aligned}$$

FIGURE 9 – Définitions de la sémantique des constructions déterministes de ProbZelus et des déclarations utilisées dans la preuve.

$$\begin{aligned}
\llbracket e \rrbracket_{\gamma}^{\text{init}} &= \llbracket e \rrbracket_{\gamma}^{\text{init}} \quad \text{if } \text{kindOf}(e) = \mathsf{D} \\
\llbracket e \rrbracket_{\gamma}^{\text{step}} &= \lambda s. \lambda U. \delta(\llbracket e \rrbracket_{\gamma}^{\text{step}}(s))(U) \quad \text{if } \text{kindOf}(e) = \mathsf{D} \\
&= \lambda s. \lambda U. \begin{cases} 1 & \text{if } \llbracket e \rrbracket_{\gamma}^{\text{step}}(s) \in U \\ 0 & \text{otherwise} \end{cases} \\
\llbracket f(e) \rrbracket_{\gamma}^{\text{init}} &= (\llbracket e \rrbracket_{\gamma}^{\text{init}}, \gamma(f_{\text{init}})) \\
\llbracket f(e) \rrbracket_{\gamma}^{\text{step}} &= \lambda(s_1, s_2). \lambda U. \\
&\quad \text{let } v_1, s'_1 = \llbracket e \rrbracket_{\gamma}^{\text{step}}(s_1) \text{ in} \\
&\quad \text{let } \mu_2 = \gamma(f_{\text{step}})(v_1)(s_2) \text{ in} \\
&\quad \int \mu_2(dv_2, ds'_2) \delta(v_2, (s'_1, s'_2))(U) \\
\llbracket \text{sample}(e) \rrbracket_{\gamma}^{\text{init}} &= \llbracket e \rrbracket_{\gamma}^{\text{init}} \\
\llbracket \text{sample}(e) \rrbracket_{\gamma}^{\text{step}} &= \lambda s. \lambda U. \text{let } \mu, s' = \llbracket e \rrbracket_{\gamma}^{\text{step}}(s) \text{ in } \int \mu(dv) \delta(v, s')(U)
\end{aligned}$$

$$\begin{aligned}
&\left\{ e \text{ where } \text{rec init } x_1 = e_1 \dots \text{ and init } x_k = e_k \right\}_{\gamma}^{\text{init}} = (\text{None}, (\llbracket e'_1 \rrbracket_{\gamma}^{\text{init}}, \dots, \llbracket e'_n \rrbracket_{\gamma}^{\text{init}}), \llbracket e \rrbracket_{\gamma}^{\text{init}}) \\
&\left\{ e \text{ where } \text{rec init } x_1 = e_1 \dots \text{ and init } x_k = e_k \right\}_{\gamma}^{\text{step}} \\
&= \lambda(m, (s_1, \dots, s_n), s). \lambda U. \\
&\quad \text{let } \mu_c = \\
&\quad \quad \text{match } m \text{ with} \\
&\quad \quad | \text{Some}(c_1, \dots, c_k) \rightarrow \delta(c_1, \dots, c_k) \\
&\quad \quad | \text{None} \rightarrow \lambda U. \\
&\quad \quad \quad \text{let } \mu_1 = \llbracket e_1 \rrbracket_{\gamma}^{\text{step}}(\llbracket e_1 \rrbracket_{\gamma}^{\text{init}}) \text{ in} \\
&\quad \quad \quad \int \mu_1(dv_1, ds_1) \dots \\
&\quad \quad \quad \text{let } \mu_k = \llbracket e_k \rrbracket_{\gamma}^{\text{step}}(\llbracket e_k \rrbracket_{\gamma}^{\text{init}}) \text{ in} \\
&\quad \quad \quad \int \mu_k(dv_k, ds_k) \delta(v_1, \dots, v_k)(U) \\
&\quad \text{in} \\
&\quad \int \mu_c(dc_1, \dots, dc_k) \\
&\quad \text{let } \gamma_1 = \gamma[x_1_{\text{last}} \leftarrow c_1] \text{ in } \dots \\
&\quad \text{let } \gamma_k = \gamma_{k-1}[x_k_{\text{last}} \leftarrow c_k] \text{ in} \\
&\quad \text{let } \mu_1 = \llbracket e'_1 \rrbracket_{\gamma_k}^{\text{step}}(s_1) \text{ in} \\
&\quad \int \mu_1(dv_1, ds'_1) \text{let } \gamma'_1 = \gamma_k[y_1 \leftarrow v_1] \text{ in } \dots \\
&\quad \text{let } \mu_n = \llbracket e'_n \rrbracket_{\gamma'_{n-1}}^{\text{step}}(s_n) \text{ in} \\
&\quad \int \mu_n(dv_n, ds'_n) \text{let } \gamma'_n = \gamma'_{n-1}[y_n \leftarrow v_n] \text{ in} \\
&\quad \text{let } \mu = \llbracket e \rrbracket_{\gamma'_n}^{\text{step}}(s) \text{ in} \\
&\quad \int \mu(dv, ds') \\
&\quad \quad \delta(v, (\text{Some}(\gamma'_n(x_1), \dots, \gamma'_n(x_k)), (s'_1, \dots, s'_n), s'))(U)
\end{aligned}$$

évaluer les inits

peupler l'environnement

résultat

FIGURE 10 – Définitions de la sémantique des constructions probabilistes de ProbZelus utilisées dans la preuve.

B Détails de la preuve du Théorème 1

Théorème 1. *Pour tout programme $prog$ tel que $\emptyset, \emptyset \vdash prog : H, C$, pour tout modèle probabiliste m de $prog$ sous la forme `proba m $x = e$ where rec init $\theta = \text{sample}(d)$ and $\theta = \text{last } \theta$` , avec $H(m) = \{\theta \leftarrow d\}$, et e ne contient pas d'appel de nœud, alors $\llbracket \text{infer}(m(obs)) \rrbracket \equiv \llbracket \text{APF.infer}(m_prior, m_model, obs) \rrbracket$.*

Démonstration.

Hypothèses et notations.

Soit un programme $prog$ tel que $\emptyset, \emptyset \vdash prog : H, C$, et m un modèle probabiliste de $prog$ sous la forme `proba m $x = e$ where rec init $\theta = \text{sample}(d)$ and $\theta = \text{last } \theta$` , avec $H(m) = \{\theta \leftarrow d\}$, et e sans d'appel de nœud. On suppose également que e n'utilise pas `last θ` (ce n'est pas restrictif car θ est constant donc `last $\theta = \theta$`).

On pose :

$$\begin{aligned} e_m &= e \text{ where rec init } \theta = \text{sample}(d) \text{ and } \theta = \text{last } \theta \\ e_{apf} &= m_model(\alpha, obs) \text{ where rec init } \alpha = \text{sample}(m_prior) \text{ and } \alpha = \text{last } \alpha \end{aligned}$$

Par hypothèse, la distribution *a priori* d est constante et $C \vdash e : \emptyset$ car le seul paramètre constant, θ , est défini à l'extérieur de e et donc $\mathcal{C}_\emptyset(e) = e$.

Soient $\gamma_1 \equiv \gamma_2$, d'après la Définition 1, pour toute variable $x \in \text{dom}(\gamma_1) \cap \text{dom}(\gamma_2)$, $\gamma_1(x) = \gamma_2(x)$ et si on note $\xi = \{\theta \leftarrow d\}$:

$$\left\{ \begin{array}{l} \gamma_1(m_init) = \llbracket e_m \rrbracket_{\gamma_1}^{\text{init}} \\ \gamma_1(m_step) = \lambda v. \llbracket e_m \rrbracket_{\gamma_1[x \leftarrow v]}^{\text{step}} \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} \gamma_2(m_model_init) = \llbracket \mathcal{C}_\xi(e_m) \rrbracket_{\gamma_2}^{\text{init}} \\ \gamma_2(m_model_step) = \lambda(\theta, v). \llbracket \mathcal{C}_\xi(e_m) \rrbracket_{\gamma_2[\text{dom}(\xi) \leftarrow \theta][x \leftarrow v]}^{\text{step}} \\ \gamma_2(m_prior) = \text{img}(\xi) \end{array} \right.$$

Enfin, on a :

$$\begin{aligned} \gamma_2(m_model_init) &= \llbracket \mathcal{C}_\xi(e_m) \rrbracket_{\gamma_2}^{\text{init}} = \llbracket e \rrbracket_{\gamma_1}^{\text{init}} \text{ car } e \text{ ne contient pas d'appel de nœud,} \\ \gamma_2(m_model_step) &= \lambda(\theta, v). \llbracket \mathcal{C}_\xi(e_m) \rrbracket_{\gamma_2[\text{dom}(\xi) \leftarrow \theta][x \leftarrow v]}^{\text{step}} = \lambda\theta, v. \llbracket e \rrbracket_{\gamma_1[x \leftarrow v][\text{dom}(\xi) \leftarrow \theta]}^{\text{step}}, \text{ et} \\ \text{dom}(\xi) &= \theta. \end{aligned}$$

Bisimulation.

Il faut maintenant montrer qu'il existe une bisimulation \mathcal{P} telle que :

$$\llbracket m(obs) \rrbracket \equiv_{\mathcal{P}} \llbracket m_model(\alpha, obs) \text{ where rec init } \alpha = \text{sample}(m_prior) \text{ and } \alpha = \text{last } \alpha \rrbracket.$$

On pose :

$$(\sigma_1, \sigma_2) \in \mathcal{P} \iff \exists \sigma \begin{cases} \sigma_1 = \lambda U. \int \sigma(dm, ds) \delta(((), (m, (), s))(U) \\ \sigma_2 = \lambda U. \int \sigma(dm, ds) \delta(m, (), (((), ()), s))(U). \end{cases}$$

État initiaux.

On vérifie d'abord que \mathcal{P} relie les états initiaux.

$$\begin{aligned}
\llbracket m(obs) \rrbracket_{\gamma_1}^{\text{init}} &= (\llbracket obs \rrbracket_{\gamma_1}^{\text{init}}, \gamma_1(m_init)) \\
&= ((), \llbracket e \text{ where rec init } \theta = \text{sample}(d) \text{ and } \theta = \text{last } \theta \rrbracket_{\gamma_1}^{\text{init}}) \\
&= ((), (None, \llbracket \text{last } \theta \rrbracket_{\gamma_1}^{\text{init}}, \llbracket e \rrbracket_{\gamma_1}^{\text{init}})) \\
&= ((), (None, (), \llbracket e \rrbracket_{\gamma_1}^{\text{init}}))
\end{aligned}$$

$$\begin{aligned}
\llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{init}} &= \llbracket m_model(\alpha, obs) \text{ where rec init } \alpha = \text{sample}(m_prior) \text{ and } \alpha = \text{last } \alpha \rrbracket_{\gamma_2}^{\text{init}} \\
&= (None, \llbracket \text{last } \alpha \rrbracket_{\gamma_2}^{\text{init}}, \llbracket m_model(\alpha, obs) \rrbracket_{\gamma_2}^{\text{init}}) \\
&= (None, (), (\llbracket (\alpha, obs) \rrbracket_{\gamma_2}^{\text{init}}, \gamma_2(m_model_init))) \\
&= (None, (), (((), ()), \llbracket \mathcal{C}_\xi(e_m) \rrbracket_{\gamma_2}^{\text{init}})) \\
&= (None, (), (((), ()), \llbracket e \rrbracket_{\gamma_1}^{\text{init}}))
\end{aligned}$$

On montre $(\delta(\llbracket m(obs) \rrbracket_{\gamma_1}^{\text{init}}), \delta(\llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{init}})) \in \mathcal{P}$ avec $\sigma = \delta(None, \llbracket e \rrbracket_{\gamma_1}^{\text{init}})$, pour tout U

$$\begin{aligned}
\int \delta(None, \llbracket e \rrbracket_{\gamma_1}^{\text{init}})(dm, ds) \delta((), (m, (), s))(U) &= \delta((), (None, (), \llbracket e \rrbracket_{\gamma_1}^{\text{init}}))(U) \\
&= \delta(\llbracket m(obs) \rrbracket_{\gamma_1}^{\text{init}})(U) \\
\int \delta(None, \llbracket e \rrbracket_{\gamma_1}^{\text{init}})(dm, ds) \delta(m, (), (((), ()), s))(U) &= \delta(None, (), (((), ()), \llbracket e \rrbracket_{\gamma_1}^{\text{init}}))(U) \\
&= \delta(\llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{init}})(U)
\end{aligned}$$

Transition.

On vérifie maintenant que \mathcal{P} se propage à travers les fonctions de transition. On veut montrer que pour tout $\gamma_1 \equiv \gamma_2$, si $(\sigma_1, \sigma_2) \in \mathcal{P}$ et :

$$\begin{aligned}
\mu_1 &= \lambda U. \int \sigma_1(ds) \llbracket m(obs) \rrbracket_{\gamma_1}^{\text{step}}(s)(U) \\
\nu_1 &= \lambda U. \mu_1(U) / \mu_1(\top) && \text{(normalisation)} \\
o_1, \sigma'_1 &= \pi_{1*}(\nu_1), \pi_{2*}(\nu_1) && \text{(marginalisation)} \\
\mu_2 &= \lambda U. \int \sigma_2(ds) \llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{step}}(s)(U) \\
\nu_2 &= \lambda U. \mu_2(U) / \mu_2(\top) && \text{(normalisation)} \\
o_2, \sigma'_2 &= \pi_{1*}(\nu_2), \pi_{2*}(\nu_2) && \text{(marginalisation)}
\end{aligned}$$

Alors $o_1 = o_2$ (égalité des distributions de sorties), et $(\sigma'_1, \sigma'_2) \in \mathcal{P}$ (bisimulation des états suivants).

Calcul de μ_1 .

Pour calculer $\mu_1 = \lambda U. \int \sigma_1(ds) \llbracket m(obs) \rrbracket_{\gamma_1}^{\text{step}}(s)(U)$:

1. On commence par calculer le corps de m : $\llbracket e_m \rrbracket^{\text{step}}$.
2. On peut ensuite calculer la sémantique de l'appel de nœud : $\llbracket m(obs) \rrbracket_{\gamma_1}^{\text{step}}$.
3. On intègre enfin le long de σ_1 pour obtenir μ_1 .

1. Calcul de $\llbracket e_m \rrbracket_\gamma^{\text{step}}$.

Par typage, on sait que l'état de e_m est de la forme $(m, (), s)$. On a pour tout environnement γ et ensemble mesurable U :

$$\begin{aligned}
\llbracket e_m \rrbracket_\gamma^{\text{step}}(m, (), s)(U) = & \text{let } \mu_c = \text{match } m \text{ with} \\
& | \text{Some}(c) \rightarrow \delta(c) \\
& | \text{None} \rightarrow \text{let } \nu_d = \lambda U. \\
& \quad \text{let } s_d = \llbracket \text{sample}(d) \rrbracket_\gamma^{\text{init}} \text{ in} \\
& \quad \text{let } \mu_d = \llbracket \text{sample}(d) \rrbracket_\gamma^{\text{step}}(s_d) \text{ in} \\
& \quad \int \mu_d(dv_d, ds'_d) \delta(v_d)(U) \\
& \quad \text{in} \\
& \quad \nu_d \\
& \quad \text{in} \\
& \int \mu_c(dc) \text{ let } \gamma' = \gamma [\theta \leftarrow c] [\theta_last \leftarrow c] \text{ in} \\
& \quad \text{let } \mu = \llbracket e \rrbracket_{\gamma'}^{\text{step}}(s) \text{ in} \\
& \quad \int \mu(dv, ds') \delta(v, (\text{Some}(c), (), s'))(U)
\end{aligned}$$

On calcule :

$$\begin{aligned}
\nu_d(U) &= \text{let } s_d = \llbracket \text{sample}(d) \rrbracket_\gamma^{\text{init}} \text{ in} \\
& \quad \text{let } \mu_d = \llbracket \text{sample}(d) \rrbracket_\gamma^{\text{step}}(s_d) \text{ in} \\
& \quad \int \mu_d(dv_d, ds'_d) \delta(v_d)(U) \\
(\text{définition de } \text{sample}) &= \text{let } s_d = \llbracket d \rrbracket_\gamma^{\text{init}} \text{ in} \\
& \quad \text{let } \mu_d = \lambda U. \\
& \quad \quad \text{let } \mu, s' = \llbracket d \rrbracket_\gamma^{\text{step}}(s_d) \text{ in} \\
& \quad \quad \int \mu(dv) \delta(v, s')(U) \\
& \quad \text{in} \\
& \quad \int \mu_d(dv_d, ds'_d) \delta(v_d)(U) \\
(\text{simplification}) &= \text{let } s_d = \llbracket d \rrbracket_\gamma^{\text{init}} \text{ in} \\
& \quad \int (\lambda U. \text{let } \mu, s' = \llbracket d \rrbracket_\gamma^{\text{step}}(s_d) \text{ in} \\
& \quad \quad \int \mu(dv) \delta(v, s')(U)) (dv_d, ds'_d) \delta(v_d)(U) \\
&= \text{let } s_d = \llbracket d \rrbracket_\gamma^{\text{init}} \text{ in} \\
& \quad \text{let } \mu, s' = \llbracket d \rrbracket_\gamma^{\text{step}}(s_d) \text{ in } \int \mu(dv) \delta(v)(U) \\
&= \text{let } s_d = \llbracket d \rrbracket_\gamma^{\text{init}} \text{ in} \\
& \quad \text{let } \mu, s' = \llbracket d \rrbracket_\gamma^{\text{step}}(s_d) \text{ in } \mu(U) \\
&= d(U)
\end{aligned}$$

On en déduit que :

$$\begin{aligned}
\llbracket e_m \rrbracket_\gamma^{\text{step}}(m, (), s)(U) = & \text{let } \mu_c = \text{match } m \text{ with} \\
& | \text{Some}(c) \rightarrow \delta(c) \\
& | \text{None} \rightarrow d \\
& \quad \text{in} \\
& \int \mu_c(dc) \text{ let } \gamma' = \gamma [\theta \leftarrow c] [\theta_last \leftarrow c] \text{ in} \\
& \quad \text{let } \mu = \llbracket e \rrbracket_{\gamma'}^{\text{step}}(s) \text{ in} \\
& \quad \int \mu(dv, ds') \delta(v, (\text{Some}(c), (), s'))(U)
\end{aligned}$$

2. Calcul de l'appel de nœud $\llbracket m(obs) \rrbracket_{\gamma_1}^{\text{step}}$.

Par typage, on sait que l'état de $m(obs)$ est de la forme $(((), (m, (), s)))$. On a pour tout ensemble mesurable U :

$$\begin{aligned}
\llbracket m(obs) \rrbracket_{\gamma_1}^{\text{step}}(((), (m, (), s)))(U) &= \int \llbracket e_m \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)]}^{\text{step}}(m, (), s)(dv, ds') \delta(v, (((), s')))(U) \\
&= \int (\lambda U. \\
&\quad \text{let } \mu_c = \text{match } m \text{ with} \\
&\quad \quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad \quad | \text{None} \rightarrow d \\
&\quad \text{in} \\
&\quad \int \mu_c(dc) \\
&\quad \quad \text{let } \gamma'_1 = \gamma_1[x \leftarrow \gamma_1(obs)] [\theta \leftarrow c] [\theta_last \leftarrow c] \text{ in} \\
&\quad \quad \text{let } \mu = \llbracket e \rrbracket_{\gamma'_1}^{\text{step}}(s) \text{ in} \\
&\quad \quad \int \mu(dv_e, ds'_e) \delta(v_e, (\text{Some}(c), (), s'_e))(U))(dv, ds') \\
&\quad \delta(v, (((), s')))(U) \\
(\text{simplification}) &= \text{let } \mu_c = \text{match } m \text{ with} \\
&\quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad | \text{None} \rightarrow d \\
&\quad \text{in} \\
&\quad \int \mu_c(dc) \\
&\quad \quad \text{let } \gamma'_1 = \gamma_1[x \leftarrow \gamma_1(obs)] [\theta \leftarrow c] [\theta_last \leftarrow c] \text{ in} \\
&\quad \quad \text{let } \mu = \llbracket e \rrbracket_{\gamma'_1}^{\text{step}}(s) \text{ in} \\
&\quad \quad \int \mu(dv, ds') \delta(v, (((), (\text{Some}(c), (), s'))))(U) \\
(\text{last } \theta \text{ n'apparaît pas dans } e) &= \text{let } \mu_c = \text{match } m \text{ with} \\
&\quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad | \text{None} \rightarrow d \\
&\quad \text{in} \\
&\quad \int \mu_c(dc) \\
&\quad \quad \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(dv, ds') \\
&\quad \quad \delta(v, (((), (\text{Some}(c), (), s'))))(U)
\end{aligned}$$

3. Intégration pour obtenir μ_1 .

On a pour tout ensemble mesurable U :

$$\begin{aligned}
\mu_1(U) &= \int \sigma_1(((), (dm, (), ds))) \llbracket m(obs) \rrbracket_{\gamma_1}^{\text{step}}(((), (m, (), s)))(U) \\
&= \int \sigma_1(((), (dm, (), ds))) \\
&\quad \text{let } \mu_c = \text{match } m \text{ with} \\
&\quad \quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad \quad | \text{None} \rightarrow d \\
&\quad \text{in} \\
&\quad \int \mu_c(dc) \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(dv, ds') \\
&\quad \quad \delta(v, (((), (\text{Some}(c), (), s'))))(U)
\end{aligned}$$

Calcul de μ_2 .

Le calcul de $\mu_2 = \lambda U. \int \sigma_2(ds) \llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{step}}(s)(U)$ suit la même structure que pour μ_1 .

1. On commence par calculer $\llbracket m_model(\alpha, obs) \rrbracket_{\gamma}^{\text{step}}$.
2. On peut ensuite calculer la sémantique du corps de `APF.infer` : $\llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{step}}$.
3. On intègre enfin le long de σ_2 pour obtenir μ_2 .

1. Calcul de $\llbracket m_model(\alpha, obs) \rrbracket_{\gamma}^{\text{step}}$.

Par typage, on sait que l'état de $m_model(\alpha, obs)$ est de la forme $(((), ()), s)$. On a pour tout environnement γ et ensemble mesurable U :

$$\begin{aligned}
\llbracket m_model(\alpha, obs) \rrbracket_{\gamma}^{\text{step}}((((), ()), s)(U) &= \text{let } v_a, s'_a = \llbracket (\alpha, obs) \rrbracket_{\gamma}^{\text{step}}(((), ())) \text{ in} \\
&\quad \int \gamma(m_model_step)(v_a)(s)(dv, ds') \\
&\quad \delta(v, (s'_a, s'))(U) \\
&= \text{let } v_{\alpha}, s'_{\alpha} = \llbracket \alpha \rrbracket_{\gamma}^{\text{step}}((())) \text{ in} \\
&\quad \text{let } v_{obs}, s'_{obs} = \llbracket obs \rrbracket_{\gamma}^{\text{step}}((())) \text{ in} \\
&\quad \text{let } v_a, s'_a = (v_{\alpha}, v_{obs}), (s'_{\alpha}, s'_{obs}) \text{ in} \\
&\quad \int \gamma(m_model_step)(v_a)(s)(dv, ds') \\
&\quad \delta(v, (s'_a, s'))(U) \\
&= \text{let } v_a, s'_a = (\gamma(\alpha), \gamma(obs)), (((), ()), s) \text{ in} \\
&\quad \int \gamma(m_model_step)(v_a)(s)(dv, ds') \\
&\quad \delta(v, (s'_a, s'))(U) \\
&= \int \gamma(m_model_step)(\gamma(\alpha), \gamma(obs))(s)(dv, ds') \\
&\quad \delta(v, (((), ()), s'))(U)
\end{aligned}$$

2. Calcul du corps de `APF.infer` $\llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{step}}$.

Rappelons que $e_{apf} = m_model(\alpha, obs)$ `where rec init $\alpha = \text{sample}(m_prior)$ and $\alpha = \text{last } \alpha$.`

On a $\llbracket \text{sample}(m_prior) \rrbracket_{\gamma_2}^{\text{init}} = \llbracket m_prior \rrbracket_{\gamma_2}^{\text{init}} = ()$ et

$$\begin{aligned}
\llbracket \text{sample}(m_prior) \rrbracket_{\gamma_2}^{\text{step}}((())) &= \lambda U. \text{let } \mu, s' = (\gamma_2(m_prior), ()) \text{ in } \int \mu(dv) \delta(v, s')(U) \\
&= \lambda U. \int d(v) \delta(v, (()))(U)
\end{aligned}$$

Par typage, on sait que l'état de e_{apf} est de la forme $(m, (), ((((), ()), s)))$. On a pour tout ensemble mesurable U :

$$\begin{aligned}
\llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{step}}(m, (), ((((), ()), s)))(U) &= \text{let } \mu_c = \\
&\quad \text{match } m \text{ with} \\
&\quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad | \text{None} \rightarrow \lambda U. \\
&\quad \quad \text{let } s_d = \llbracket \text{sample}(m_prior) \rrbracket_{\gamma_2}^{\text{init}} \text{ in} \\
&\quad \quad \text{let } \mu_d = \llbracket \text{sample}(m_prior) \rrbracket_{\gamma_2}^{\text{step}}(s_d) \text{ in} \\
&\quad \quad \int \mu_d(dv_d, ds'_d) \delta(v_d)(U) \\
&\text{in} \\
&\int \mu_c(dc) \text{let } \gamma'_2 = \gamma_2[\alpha \leftarrow c][\alpha_last \leftarrow c] \text{ in} \\
&\quad \text{let } \mu = \llbracket m_model(\alpha, obs) \rrbracket_{\gamma'_2}^{\text{step}}((((), ()), s) \text{ in} \\
&\quad \int \mu(dv, ds') \delta(v, (\text{Some}(c), ()), s'))(U)
\end{aligned}$$

$$\begin{aligned}
& \text{(cf. calcul de } \nu_d \text{ pour } \mu_1) & = & \text{let } \mu_c = \text{match } m \text{ with} \\
& & & \quad | \text{Some}(c) \rightarrow \delta(c) \\
& & & \quad | \text{None} \rightarrow d \\
& & & \text{in} \\
& & & \int \mu_c(dc) \text{ let } \gamma'_2 = \gamma_2 [\alpha \leftarrow c] [\alpha_last \leftarrow c] \text{ in} \\
& & & \quad \text{let } \mu = \llbracket m_model(\alpha, obs) \rrbracket_{\gamma'_2}^{\text{step}}((((), ()), s) \text{ in} \\
& & & \quad \int \mu(dv, ds') \delta(v, (\text{Some}(c), (), s'))(U) \\
& \text{(simplification)} & = & \text{let } \mu_c = \text{match } m \text{ with} \\
& & & \quad | \text{Some}(c) \rightarrow \delta(c) \\
& & & \quad | \text{None} \rightarrow d \\
& & & \text{in} \\
& & & \int \mu_c(dc) \\
& & & \quad \text{let } \gamma'_2 = \gamma_2 [\alpha \leftarrow c] [\alpha_last \leftarrow c] \text{ in} \\
& & & \quad \text{let } \mu = \lambda U. \\
& & & \quad \int \gamma'_2(m_model_step)(\gamma'_2(\alpha), \gamma'_2(obs))(s)(dv_e, ds'_e) \\
& & & \quad \delta(v_e, ((((), ()), s'_e))(U) \\
& & & \text{in} \\
& & & \int \mu(dv, ds') \delta(v, (\text{Some}(c), (), s'))(U) \\
& & = & \text{let } \mu_c = \text{match } m \text{ with} \\
& & & \quad | \text{Some}(c) \rightarrow \delta(c) \\
& & & \quad | \text{None} \rightarrow d \\
& & & \text{in} \\
& & & \int \mu_c(dc) \\
& & & \quad \text{let } \mu = \lambda U. \\
& & & \quad \int \gamma_2(m_model_step)(c, \gamma_2(obs))(s)(dv_e, ds'_e) \\
& & & \quad \delta(v_e, ((((), ()), s'_e))(U) \\
& & & \text{in} \\
& & & \int \mu(dv, ds') \delta(v, (\text{Some}(c), (), s'))(U) \\
& \text{(cf. hypothèse)} & = & \text{let } \mu_c = \text{match } m \text{ with} \\
& & & \quad | \text{Some}(c) \rightarrow \delta(c) \\
& & & \quad | \text{None} \rightarrow d \\
& & & \text{in} \\
& & & \int \mu_c(dc) \text{ let } \mu = \lambda U. \\
& & & \quad \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][dom(\xi) \leftarrow c]}^{\text{step}}(s)(dv_e, ds'_e) \\
& & & \quad \delta(v_e, ((((), ()), s'_e))(U) \\
& & & \text{in} \\
& & & \int \mu(dv, ds') \delta(v, (\text{Some}(c), (), s'))(U) \\
& \text{(simplification, } dom(\xi) = \theta) & = & \text{let } \mu_c = \text{match } m \text{ with} \\
& & & \quad | \text{Some}(c) \rightarrow \delta(c) \\
& & & \quad | \text{None} \rightarrow d \\
& & & \text{in} \\
& & & \int \mu_c(dc) \\
& & & \quad \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(dv, ds') \\
& & & \quad \delta(v, (\text{Some}(c), (), ((((), ()), s'))(U)
\end{aligned}$$

3. Intégration pour obtenir μ_2 .

On a pour tout ensemble mesurable U :

$$\begin{aligned}
\mu_2(U) &= \int \sigma_2(dm, (), (((), ()), ds)) \llbracket e_{apf} \rrbracket_{\gamma_2}^{\text{step}}(m, (), (((), ()), s))(U) \\
&= \int \sigma_2(dm, (), (((), ()), ds)) \\
&\quad \text{let } \mu_c = \text{match } m \text{ with} \\
&\quad \quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad \quad | \text{None} \rightarrow d \\
&\quad \text{in} \\
&\quad \int \mu_c(dc) \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(dv, ds') \delta(v, (\text{Some}(c), (), (((), ()), s')))(U)
\end{aligned}$$

Conclusion.

On peut maintenant vérifier que $\sigma_1 = \sigma_2$ et que la bisimulation s'applique bien aux nouveaux états $(\sigma'_1, \sigma'_2) \in \mathcal{P}$. Comme $(\sigma_1, \sigma_2) \in \mathcal{P}$, il existe σ tel que

$$\begin{aligned}
\sigma_1 &= \lambda U. \int \sigma(dm, ds) \delta((), (m, (), s))(U) \\
\sigma_2 &= \lambda U. \int \sigma(dm, ds) \delta(m, (), (((), ()), s))(U).
\end{aligned}$$

On a donc

$$\begin{aligned}
\mu_1(U) &= \int \sigma(dm, ds) \\
&\quad \text{let } \mu_c = \text{match } m \text{ with} \\
&\quad \quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad \quad | \text{None} \rightarrow d \\
&\quad \text{in} \\
&\quad \int \mu_c(dc) \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(dv, ds') \delta(v, ((), (\text{Some}(c), ()), s'))(U) \\
\mu_2(U) &= \int \sigma(dm, ds) \\
&\quad \text{let } \mu_c = \text{match } m \text{ with} \\
&\quad \quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad \quad | \text{None} \rightarrow d \\
&\quad \text{in} \\
&\quad \int \mu_c(dc) \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(dv, ds') \delta(v, (\text{Some}(c), (), (((), ()), s')))(U)
\end{aligned}$$

On peut alors calculer les constantes de normalisation :

$$\begin{aligned}
\mu_1(\top) = \mu_2(\top) &= \int \sigma(dm, ds) \text{ let } \mu_c = \text{match } m \text{ with} \\
&\quad \quad | \text{Some}(c) \rightarrow \delta(c) \\
&\quad \quad | \text{None} \rightarrow d \\
&\quad \text{in} \\
&\quad \int \mu_c(dc) \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(\top)
\end{aligned}$$

On peut maintenant montrer que les sorties sont égales :

$$\begin{aligned}
o_1(U) = o_2(U) &= 1/\mu_1(\top) \\
&\int \sigma(dm, ds) \text{ let } \mu_c = \text{match } m \text{ with} \\
&\quad \begin{array}{l} | \text{Some}(c) \rightarrow \delta(c) \\ | \text{None} \rightarrow d \end{array} \\
&\quad \text{in} \\
&\int \mu_c(dc) \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(dv, ds') \delta(v)(U)
\end{aligned}$$

Finalement on a bien $(\sigma'_1, \sigma'_2) \in \mathcal{P}$ avec :

$$\begin{aligned}
\sigma'(U) &= 1/\mu_1(\top) \\
&\int \sigma(dm, ds) \text{ let } \mu_c = \text{match } m \text{ with} \\
&\quad \begin{array}{l} | \text{Some}(c) \rightarrow \delta(c) \\ | \text{None} \rightarrow d \end{array} \\
&\quad \text{in} \\
&\int \mu_c(dc) \int \llbracket e \rrbracket_{\gamma_1[x \leftarrow \gamma_1(obs)][\theta \leftarrow c]}^{\text{step}}(s)(dv, ds') \delta(\text{Some}(c), s')(U)
\end{aligned}$$

□