

Norme

L'objet central de GreenTea est la norme.

Une norme est:

- Une fonction d'un type A vers un type B ,
- qui exploite un canal de communication entre un couple d'agents : $i \mid j$,
- qui exporte un modèle catégoriel de A vers B ,
- qui requiert (en entrée) un contexte logique/un état, γ , dépendant de A et du canal
- qui garantit (en sortie) un contexte logique/un état, λ , dépendant de B et du canal

La norme a été formalisée pour être à l'intersection de

- la théorie des catégories : les normes forment une catégorie monoidale, elles sont assemblables avec les opérateurs «andThen», «and», «or» et «xor», - cf https://stringfixer.com/fr/Sieve/category_theory/),
- des systèmes multi-agents ? : elles représentent un message associé chaque couple d'agents, en fonction d'un contexte et d'une entrée fournis - autrement dit un protocole à état,
- de la sémantique de Kripke : elles représentent la transition entre deux mondes chacun caractérisé par l'ensemble de propriétés qu'il vérifie, respectivement, γ et λ .

Interpreteur

L'implémentation a été conçue pour maximiser la malléabilité et la réexploitabilité de la norme.

Pour cela chaque «aspect» de la norme a été «habillée» par un type la généralisant tout en la soumettant à un «contrat» vis à vis de son utilisation. Tous ces types abstraits sont réunis au sein d'un unique objet, *l'interpreteur*, qui doit être fourni à la norme pour que celle-ci soit exécutée. Il s'agit de:

- ρ : le type des données. Ainsi une norme n'est pas une fonction de $A \Rightarrow B$ mais plutôt de $\rho[A] \Rightarrow \rho[B]$. ρ peut matérialiser le fait que A peut résulter d'un calcul et par là être une exception, peut être une liste d'objet, une distribution statistique, un accès distant, un objet crypter homomorphiquement, ...,
- κ , le pendant de ρ pour les données logiques (γ, λ),
- φ : l'environnement de calcul. Immédiat ou paresseux, sur une plateforme tel que Jade ou Akka ou sur une infrastructure tel que Cuda ou JNI,
- ψ : l'extracteur qui permet d'obtenir une valeur depuis l'environnement de calcul,
- \mid : le canal de communication,
- η, Ω , les *holons* faisant de cette norme un contrat. Ils généralisent les fonctions définies entre tous couples d'agents en une fonction définie pour la société grâce à la spécification d'un protocole de consensus (e.g. calcul parallèle, vote, contract-net protocol, etc)
- ε : l'environnement partagé par les agents dans le contrat résultant de l'utilisation de la norme. Il doit être accessible en lecture et en écriture, et cela de façon individualisée à chaque couple d'agent,
- Δ traduit *à posteriori* une norme en un modèle catégoriel permettant soit le prototypage (e.g. une norme devient une fabrique de normes) soit l'interprétation (e.g. des métriques d'analyse, de certifications, etc.)

Les interpréteurs sont définis en tant que librairie et chargés pour instancier une norme à partir d'une prénorme (cf prochain tuto). Tout interpréteur n'est cependant pas compatible avec toute prénorme.

Modalité

Le type de retour d'une norme est aussi encapsulé au sein d'un type que nous appelons «modalité». En effet, il permet d'implémenter la logique modale. Ce type ne dépend pas de l'interpréteur mais est défini par l'utilisateur.

En logique aléthique, base de la sémantique de Kripke, on considère 4 modalités du vrai:

- La nécessité : une proposition est toujours vraie. Ceci correspond au type "Id" c'est-à-dire pas d'encapsulation : l'objet est retourné tel quel
- La possibilité : une proposition est parfois vrai. Nous représentons cela avec le type "Maybe" qui correspond au retour d'un calcul qui peut échouer en une exception
- La contingence : une proposition peut être vraie ou pas. Nous représentons ceci avec le type "Option"
- L'impossibilité : une proposition n'est jamais vraie. Nous représentons cela conformément à l'approche agent (cf FIPA-ACL) en considérant la cause de cette impossibilité. Soit un échec du calcul, une impossibilité de le réaliser ou un refus de l'agent.

La logique aléthique, à travers la sémantique de Kripke, couvre la logique métier de nombreux domaines (cf https://fr.wikipedia.org/wiki/S%C3%A9mantique_de_Kripke). L'intégration de la récursivité nous permet de plus de modéliser la logique temporelle et de raisonner sur les exécutions possibles. De plus, on peut graphiquement assembler des «plans de secours» (valeur par défaut en cas de contingence, ou résolution d'exception en cas de possibilité) pour rendre une norme nécessairement vraie. Enfin, il est naturel d'utiliser d'autres modalités que celle décrites dans la logique aléthique si le besoin se fait.