

# TP n° 11

Paradigmes et interprétation  
Licence Informatique  
Université Côte d’Azur

Pour ce TP, utilisez le fichier `typed-record.rkt`. C’est une extension (à compléter) de l’interpréteur `type.rkt`.

## Typage des enregistrements

Le fichier `typed-record.rkt` contient les définitions pour les enregistrements (expressions, valeur et type). L’analyseur syntaxique et l’interpréteur ont déjà été mis à jour (de manière similaire à ce qui a été vu au Cours n° 4).

Complétez la fonction `typecheck` pour pouvoir vérifier les types des trois expressions liées aux enregistrements. Pour cet exercice, vous ne devez pas prendre en compte le sous-typage.

## Sous-typage

Ajoutez la fonction `is-subtype?` dans l’interpréteur.

```
(define (is-subtype? [t1 : Type] [t2 : Type]) : Boolean
  (equal? t1 t2))
```

Vous allez progressivement modifier la fonction `is-subtype?` pour prendre en compte le sous-typage. L’appel `(is-subtype? t1 t2)` renverra `#t` si `t1` est un sous-type de `t2`; `#f` sinon.

1. Modifiez les règles de typage de l’appel de fonction et de la mise à jour des champs pour prendre en compte le sous-typage. Supposez pour cela que `is-subtype?` est correctement définie.
2. Modifiez la fonction `is-subtype?` pour implémenter :
  - le sous-typage naturel des enregistrements (champs dans un ordre différent et/ou champs supplémentaires),
  - le sous-typage des enregistrements avec sous-typage récursif des champs,
  - le sous-typage des fonctions en prenant en compte la covariance et la contravariance.

À chaque étape, de plus en plus de tests (inclus dans le fichier) réussiront.