

Construction, exploitation et reconstruction d'un référentiel de compétences pour l'enseignement de la programmation récursive en Licence 1

Nathalie Guin, Marie Lefevre



Contexte de production / exploitation du référentiel

- Projet ANR COMPER
- Un des terrains d'expérimentation : Licence 1 Math-Info à L'Université Lyon 1
- Une UE d'enseignement de la programmation récursive
 - langage Scheme
 - Entre 400 et 450 étudiants par an
- Dispositif d'enseignement hybride :
 - Une UE de 3 ects : CM/TD/TP sur 13 semaines d'enseignement
 - ASKER : exercices en ligne d'auto-évaluation
 - entraînement sur les notions vues en CM avant les TD/TP

Processus de construction du référentiel

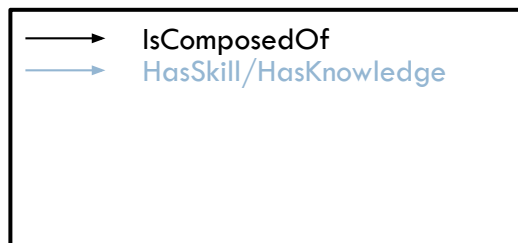
- Matériaux exploités : enseignement existant (CM/TD/TP + évaluations)
 - Description détaillée de tout le contenu de l'enseignement
 - “Je veux expliciter pour les étudiants tout ce qu'ils devraient savoir faire”
- Fondements épistémologiques : aucun
- Domaine de couverture : l'UE

→ Production d'un premier référentiel par Nathalie, soumis pour correction et validation à Marie

- Difficulté à respecter le méta-modèle COMPER
 - Les relations de pré-requis peuvent seulement être définies entre éléments de même type

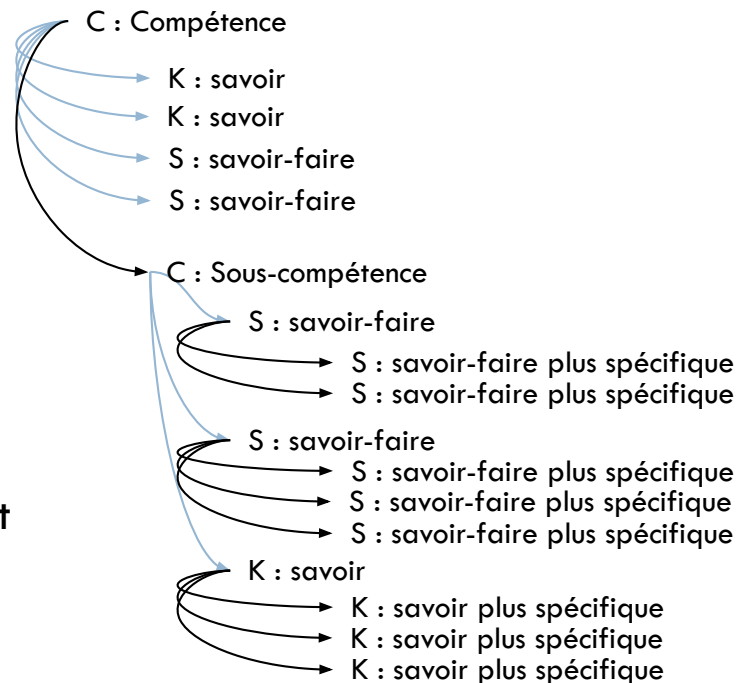
Éléments structurants du référentiel : le méta-modèle COMPER

Une compétence est un ensemble de savoirs et savoir-faire



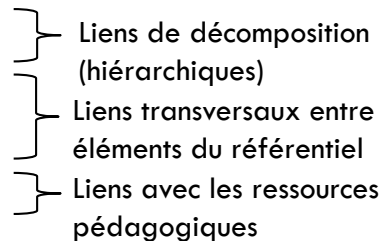
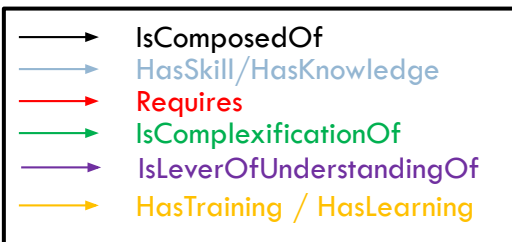
Liens de décomposition
(hiérarchiques)

- La relation **IsComposedOf** sert à décomposer un élément en des éléments de même type
- Les relations **HasSkill** et **HasKnowledge** servent à définir les éléments S et K constituant une compétence

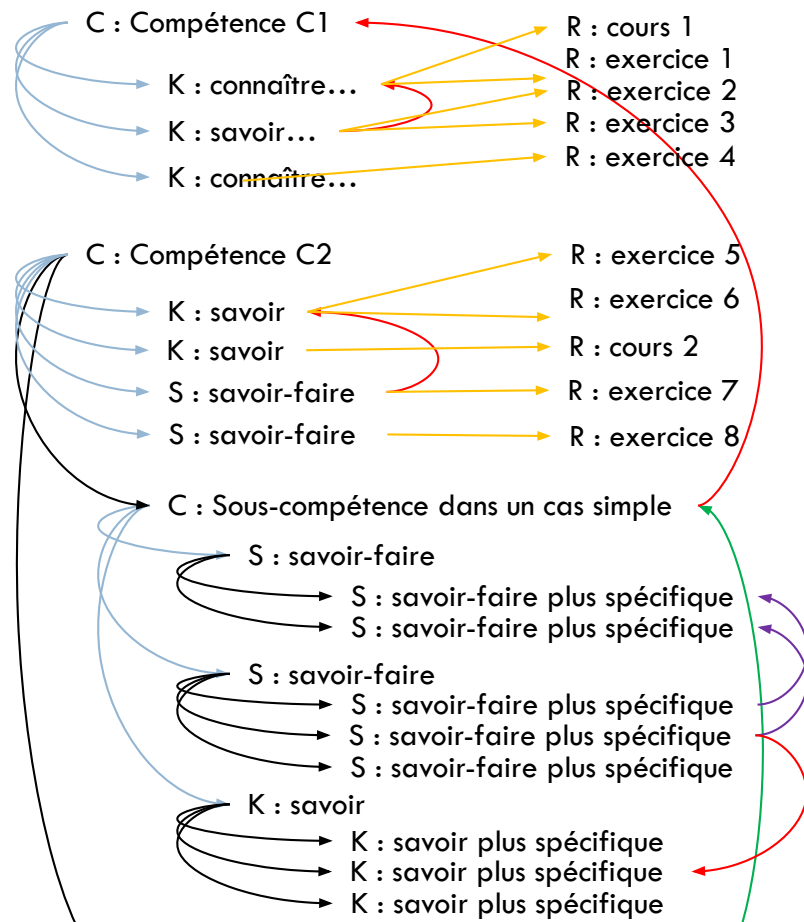


Éléments structurants du référentiel :

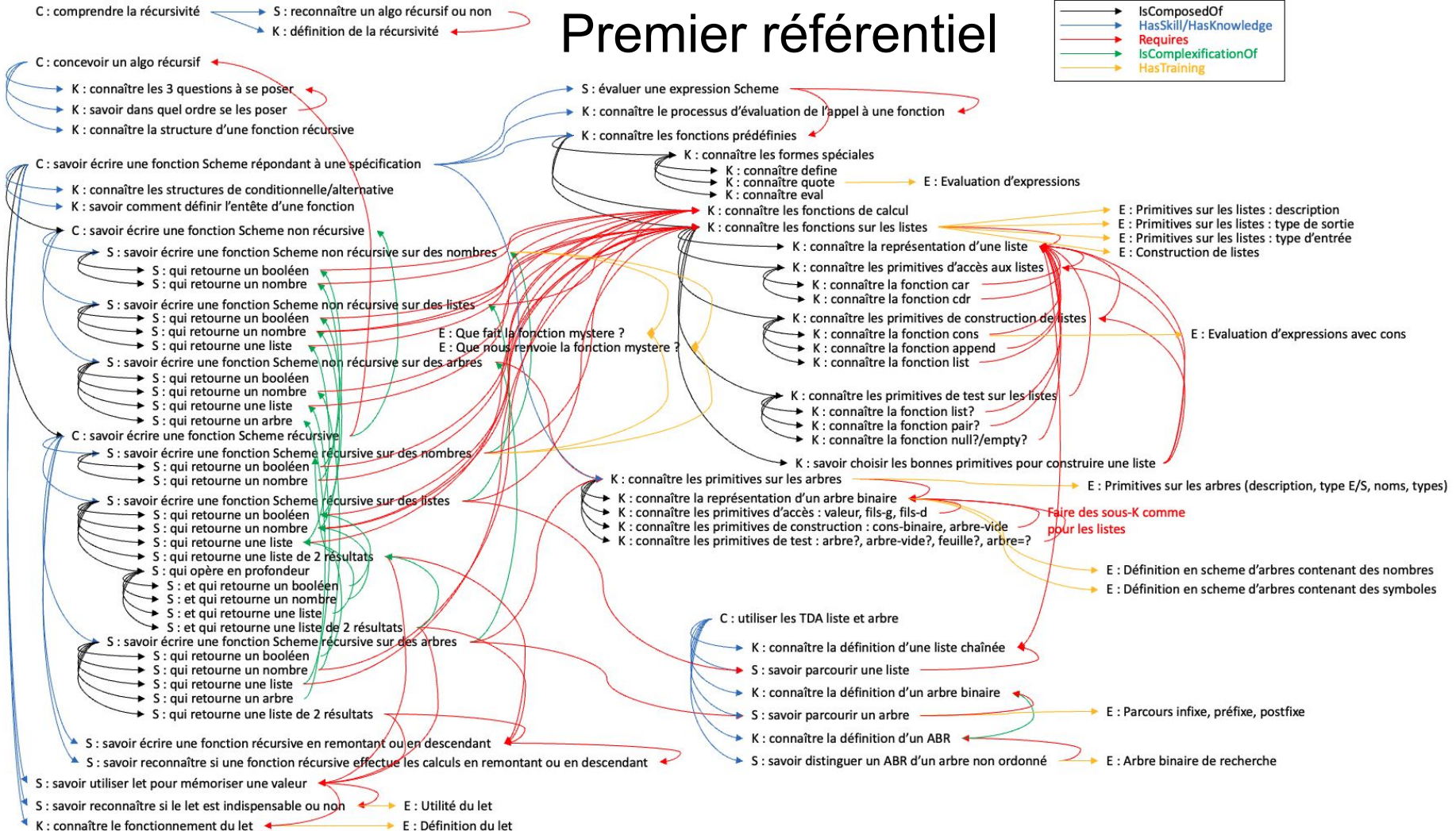
le méta-modèle COMPER



- Les relations **Requires**, **IsComplexificationOf**, **IsLeverOfUnderstandingOf** peuvent relier des éléments de même type, et aussi relier des Knowledge et Skill



Premier référentiel



Représentation / formalisation du référentiel

- Quels choix de représentation / formalisation du référentiel
 - Ontologie
- Utilisation de l'éditeur de référentiels du projet COMPER :
 - Competency Framework Editor
- Puis rattachement des exercices ASKER de l'UE au référentiel
 - Platform2ref

Exploitations du référentiel

- Exploitation du référentiel uniquement interne au projet mais référentiel accessible publiquement
- Exploitations effectivement réalisées :
 - Calcul des profils de compétences des étudiants à partir des traces ASKER
 - Recommandations personnalisées d'exercices ASKER en fonction :
 - i. d'un objectif d'apprentissage fixé par l'enseignant
 - ii. du profil de compétence de l'étudiant
 - iii. d'une stratégie de recommandation

Mon profil

Mettre à jour mon profil

Cacher les ressources

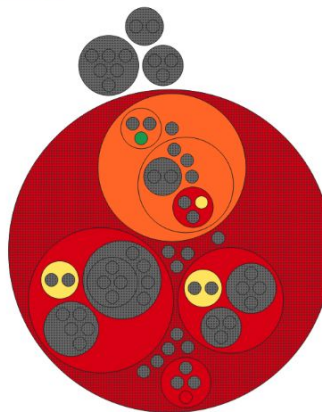
Arborescent



- LIFAP2_2021
 - > Comprendre_la_récurtivité
 - V Savoir_écrire_une_fonction_Scheme_répondant_à_une_spécification
 - connaître_le_processus_d'évaluation_de_l'appel_à_une_fonction
 - > connaître_les_primitives_sur_les_arbres
 - connaître_les_structures_de_conditionnelle_alternative
 - V connaître_les_fonctions_prédéfinies
 - V connaître_les_fonctions_sur_les_listes
 - Exercice** : Construction de listes
 - Exercice** : Primitives sur les listes : type d'entrée
 - Exercice** : Primitives sur les listes : type de sortie
 - Exercice** : Primitives sur les listes : description
 - Savoir_choisir_les_bonnes_primitives_pour_construire_une_liste
 - connaître_la_représentation_d_une_liste
 - V connaître_les_primitives_de_construction_de_listes
 - connaître_la_fonction_list
 - > connaître_la_fonction_cons
 - connaître_la_fonction_append
 - connaître_les_primitives_de_test_sur_les_listes
 - > connaître_les_primitives_d'accès_aux_listes
 - connaître_les_fonctions_de_calcul
 - V connaître_les_formes_spéciales
 - V connaître_quote
 - Exercice** : Evaluation d'expressions

Mon profil

Mettre à jour mon profil

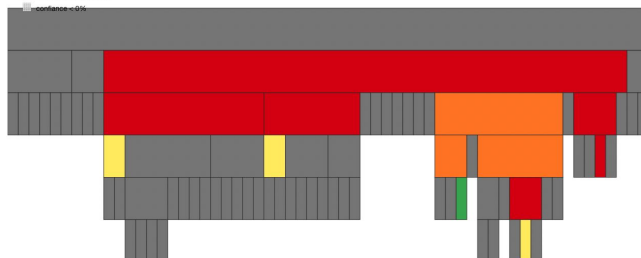


Mon profil

Mettre à jour mon profil

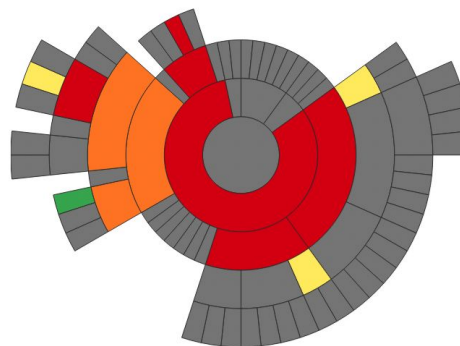


- Maitrise non évaluée
- maitrise ∈ [0%,25%]
- maitrise ∈ [25%,50%]
- maitrise ∈ [50%,75%]
- maitrise ∈ [75%,100%]



Mon profil

Mettre à jour mon profil



- Maitrise non évaluée
- taux de maîtrise ∈ [0%,25%]
- taux de maîtrise ∈ [25%,50%]
- taux de maîtrise ∈ [50%,75%]
- taux de maîtrise ∈ [75%,100%]

Exercices

Compétences et recommandations

Mes objectifs

Choix des objectifs

Récupérer les objectifs de l'enseignant

Récupérer mes objectifs

Compétence : connaître les primitives sur les arbres**Intention pédagogique** : Revision

Mes recommandations

Obtenir de nouvelles recommandations

connaître les primitives sur les arbres


 Definition en Schéma d'arbre contenant des symboles


 Definition en Schéma d'arbre contenant des nombres


 Primitives sur les arbres : leurs noms


 Primitives sur les arbres : les types de sortie


 Primitives sur les arbres : les types d'entrée

Mon profil

Mettre à jour mon profil

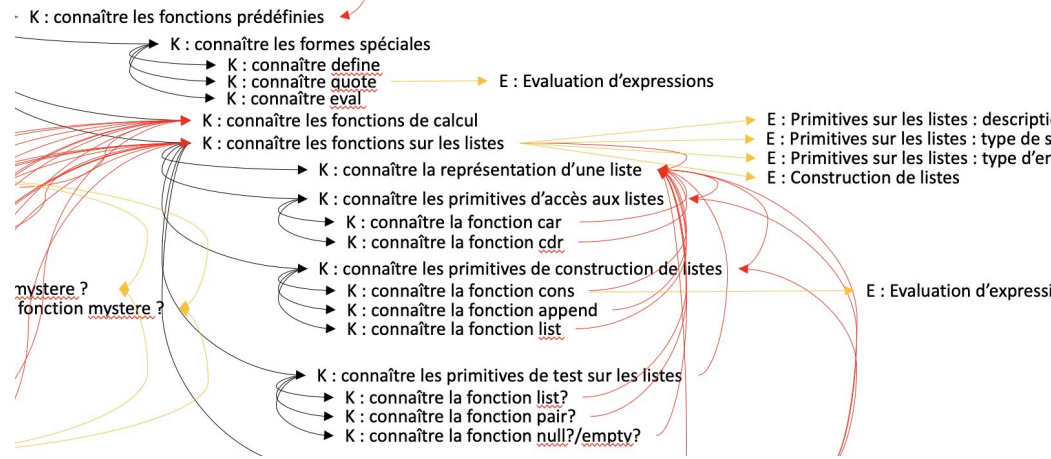
Cacher les ressources

Arborescent



	maîtrise	confiance
<input type="radio"/> LIFAP2_2021	-	-
<input type="radio"/> Comprendre la récursivité	0%	0%
<input type="radio"/> Connaître la définition de la récursivité	0%	0%
<input type="radio"/> Reconnaître un algorithme récursif ou non	0%	0%
<input type="radio"/> Savoir écrire une fonction Scheme répondant à une spécification	5%	25%
<input type="radio"/> connaître le processus d'évaluation de l'appel à une fonction	0%	0%
<input type="radio"/> connaître les primitives sur les arbres	3%	21%
Exercice : Primitives sur les arbres : leurs descriptions		
Exercice : Primitives sur les arbres : test, accès ou construction ?		
Exercice : Primitives sur les arbres : les types d'entrée		
Exercice : Primitives sur les arbres : les types de sortie		
Exercice : Primitives sur les arbres : leurs noms		
<input type="radio"/> connaître les primitives de test sur les arbres	0%	0%
<input checked="" type="radio"/> connaître la représentation d'un arbre binaire	31%	52%
Exercice : Definition en Schéma d'arbre contenant des symboles		
Exercice : Definition en Schéma d'arbre contenant des nombres		
<input type="radio"/> connaître les primitives de construction d'arbres	0%	0%
<input type="radio"/> connaître les primitives d'accès aux arbres	0%	0%
<input type="radio"/> connaître les structures de conditionnelle alternative	0%	0%
<input type="radio"/> connaître les fonctions prédéfinies	25%	38%
<input type="radio"/> connaître les fonctions sur les listes	43%	55%
Exercice : Construction de listes		
Exercice : Primitives sur les listes : type d'entrée		
Exercice : Primitives sur les listes : type de sortie		
Exercice : Primitives sur les listes : description		
<input type="radio"/> Savoir choisir les bonnes primitives pour construire une liste	0%	0%
<input type="radio"/> connaître la représentation d'une liste	0%	0%
<input checked="" type="radio"/> connaître les primitives de construction de listes	23%	54%
<input type="radio"/> connaître la fonction list	0%	0%
<input checked="" type="radio"/> connaître la fonction cons	68%	54%
Exercice : Evaluation d'expressions avec la fonction cons		
<input type="radio"/> connaître la fonction append	0%	0%
<input type="radio"/> connaître les primitives de test sur les listes	0%	0%

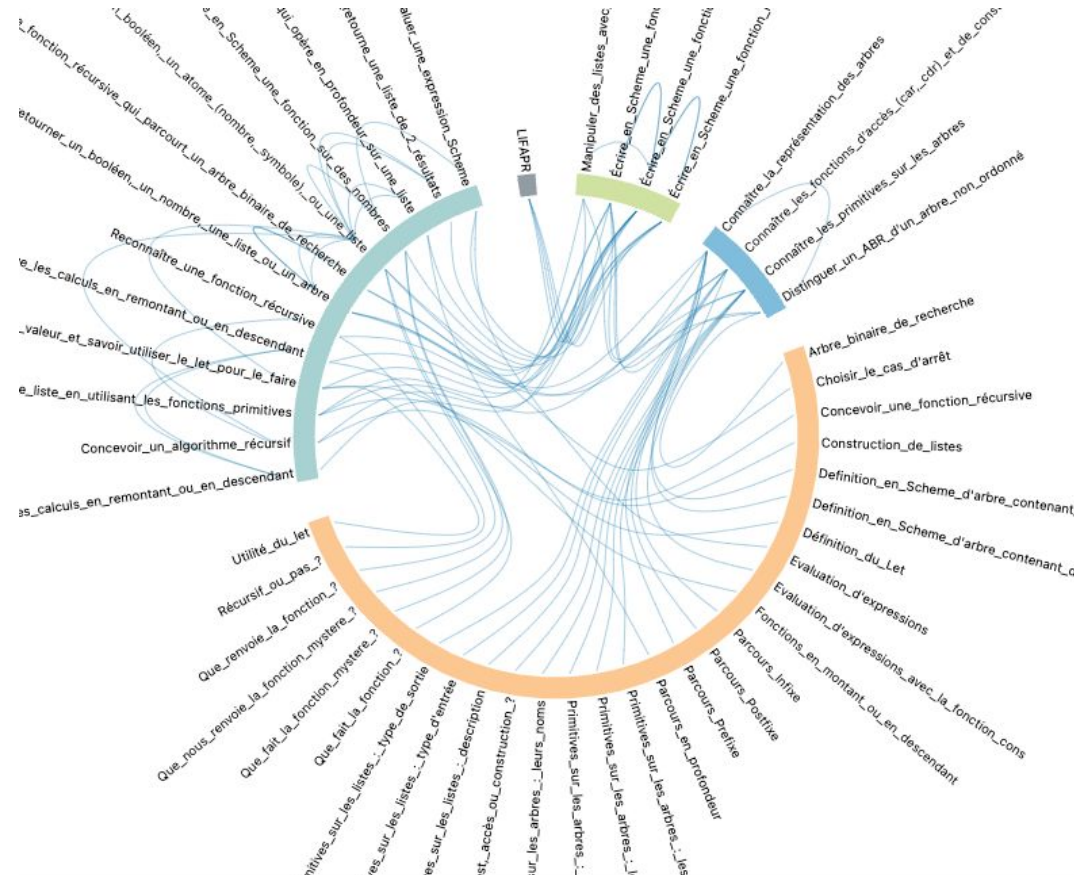
Retours d'expérience



- Construction du référentiel indépendamment
 - des exercices ASKER disponibles
 - de l'avancement chronologique de l'enseignement
- Certains exercices qui sont moins spécifiques que les notions du référentiels
- Des notions sans exercices ASKER
 - pas de possibilité pour les étudiants de compléter leur profil de compétences
- Retour des étudiants :
 - le profil de compétences permet une vue d'ensemble (parfois trop exhaustive) sur le cours
 - besoin d'une structuration plus forme pour l'organisation du travail en autonomie

Création d'un nouveau référentiel pour 2024

- Introduction de l'APC dans les CM/TD/TP et les évaluations
 - 8 compétences à acquérir
 - Quelles compétences évaluées dans chaque épreuve de CC
 - Un feedback sur les compétences évaluées dans chaque épreuve
- Un nouveau référentiel créé avec Competency Framework Editor :
 - 4 compétences
 - 2 niveaux : compétence et skill/knowledge
 - des exercices ASKER créés pour chaque skill/knowledge



Encore un problème d'alignement pédagogique...

- C1. Connaître les fonctions d'accès (car, cdr) et de construction (cons, list, append) de listes et savoir les utiliser à bon escient
- C2. Savoir écrire une fonction qui parcourt récursivement une liste pour retourner un booléen, un atome (nombre, symbole), ou une liste
- C3. Reconnaître quand il est nécessaire de mémoriser une valeur et savoir utiliser le let pour le faire
- C4. Savoir écrire une fonction qui retourne une liste de 2 résultats
- C5. Savoir écrire et utiliser un jeu de tests
- C6. Savoir reconnaître si une fonction effectue les calculs en remontant ou en descendant. Être capable de compléter une fonction écrite selon l'une ou l'autre des méthodes
- C7. Savoir écrire une fonction qui opère en profondeur sur une liste
- C8. Savoir écrire une fonction qui parcourt récursivement un arbre binaire, pour retourner un booléen, un nombre, une liste ou un arbre

- LIFAPR
- v Écrire en Scheme une fonction récursive qui parcourt une liste
 - v Écrire une fonction qui opère en profondeur sur une liste
 - v Reconnaître quand il est nécessaire de mémoriser une valeur et savoir utiliser le let pour le faire
 - Écrire une fonction qui retourne une liste de 2 résultats
 - Compléter une fonction qui effectue les calculs en remontant ou en descendant
 - v Écrire en Scheme une fonction simple qui parcourt récursivement une liste pour retourner un atome
 - v Reconnaître si une fonction effectue les calculs en remontant ou en descendant
- v Écrire en Scheme une fonction récursive qui parcourt un arbre binaire
 - v Distinguer un ABR d'un arbre non ordonné
 - v Connaître la représentation des arbres
 - v Connaître les primitives sur les arbres
 - Écrire en Scheme une fonction récursive qui parcourt un arbre binaire de recherche
 - Écrire en Scheme une fonction récursive qui parcourt un arbre binaire, pour retourner un atome
- v Manipuler des listes avec Scheme
 - v Connaître les fonctions d'accès (car, cdr) et de construction (cons, list, append) de listes
 - v Construire une liste en utilisant les fonctions primitives
- v Écrire en Scheme une fonction récursive sur des nombres
 - v Concevoir un algorithme récursif
 - v Évaluer une expression Scheme
 - v Reconnaître une fonction récursive
 - v Écrire en Scheme une fonction sur des nombres

Partage / accessibilité du travail et perspectives

Quelles modalités de partage du travail réalisé ?

- Référentiels en accès libre
- Outils COMPER disponibles pour tous
- Possibilité de brancher les services COMPER sur une plateforme d'enseignement

Perspectives associées à ce travail :

- Poursuite du travail pour LIFAPR : nouveaux exercices, alignement des référentiels
- Travaux en cours sur les explications
 - du calcul du profil et du service de recommandation
 - pour les apprenants et les enseignants
- Projet ANR Tea-time sur l'appropriation des outils COMPER par les enseignants