Ré-identification sans coordination dans les types de données répliquées sans conflits

Matthieu Nicolas (matthieu.nicolas@loria.fr)

20 décembre 2022

Rapporteurs :	Hanifa Boucheneb	Professeure, Polytechnique Montréal
	Davide Frey	Chargé de recherche, HdR, Inria Rennes Bretagne-Atlantique
Examinateurs :	Hala Skaf-Molli	Maîtresse de conférences, HdR, Nantes Université, LS2N
	Stephan Merz	Directeur de Recherche, Inria Nancy - Grand Est
Encadrants :	Olivier Perrin	Professeur des Universités, Université de Lorraine, LORIA
	Gérald Oster	Maître de conférences, Université de Lorraine, LORIA

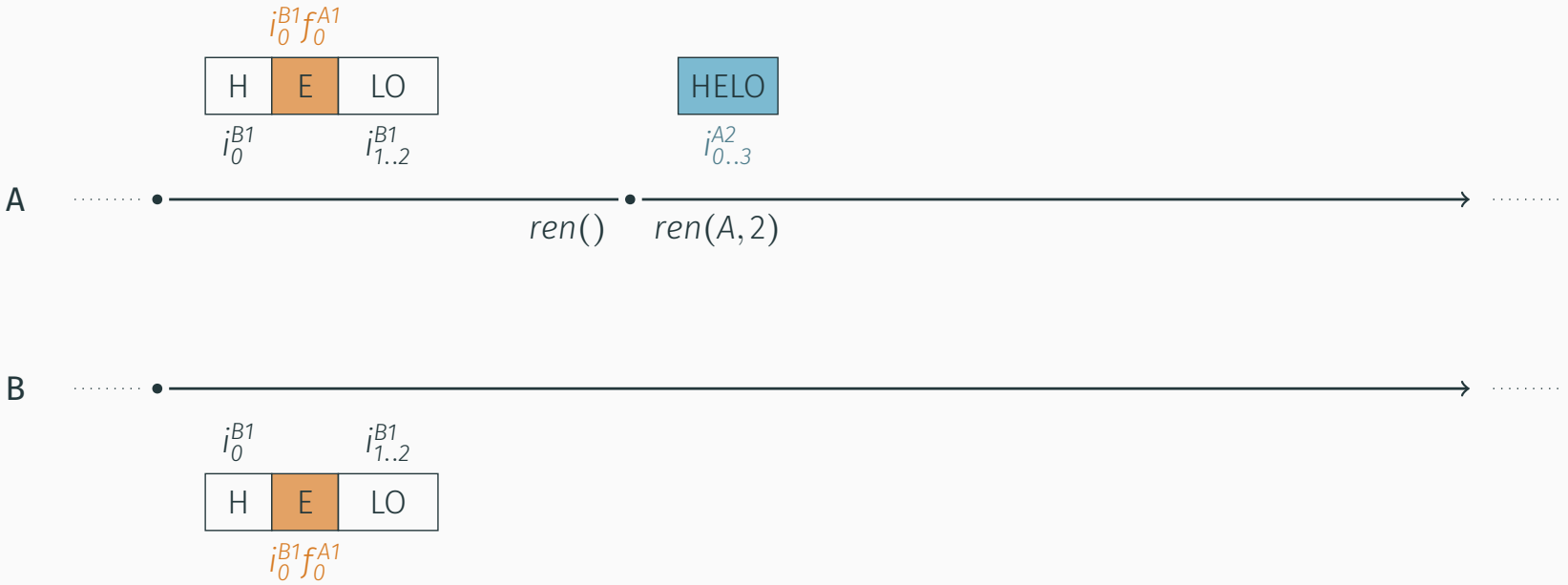
Loria
Laboratoire lorrain de recherche en informatique et ses applications

Inria

cnrs
dépasser les frontières

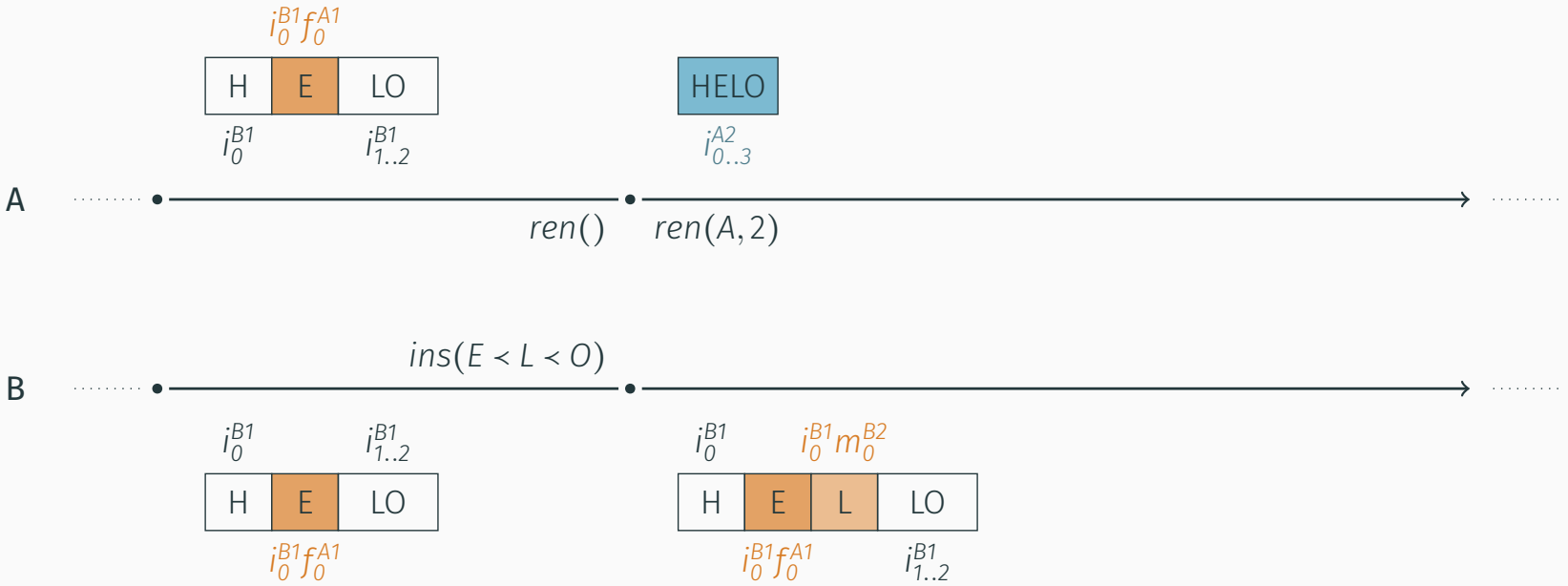
RenamableLogootSplit

Intégration des opérations *insert* et *remove* concurrentes



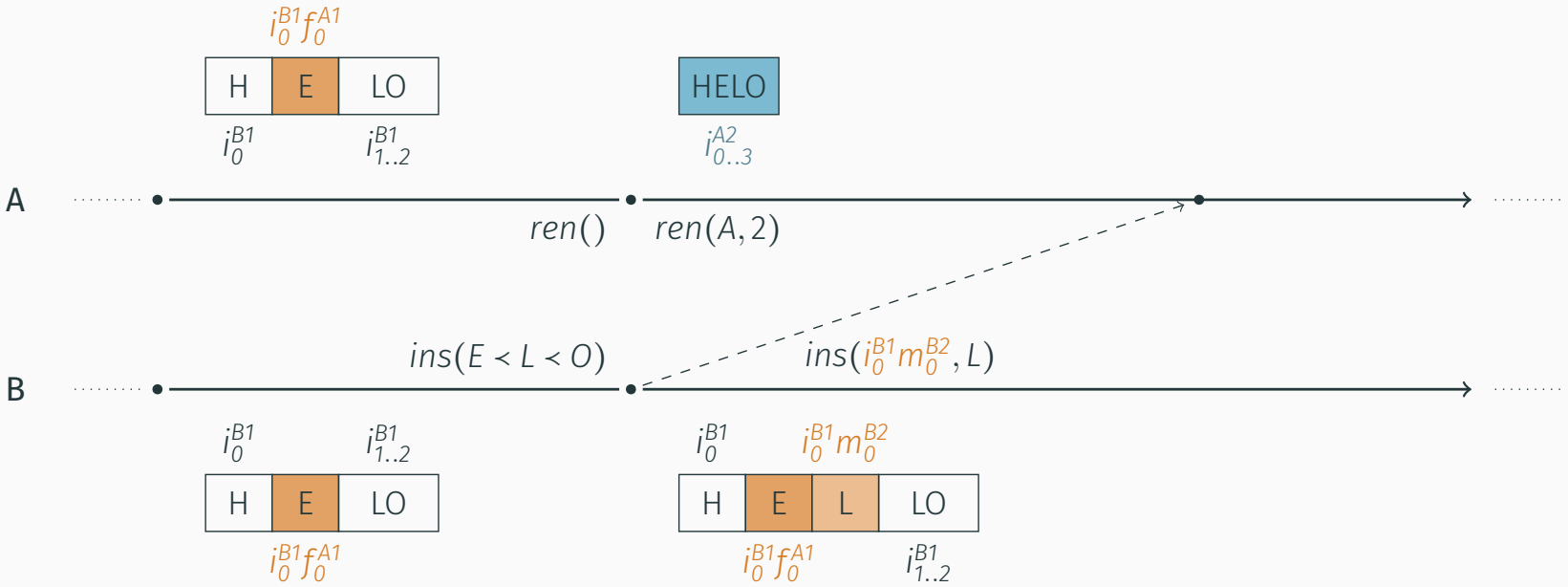
- Peuvent générer opérations concurrentes aux opérations *rename*

Intégration des opérations *insert* et *remove* concurrentes



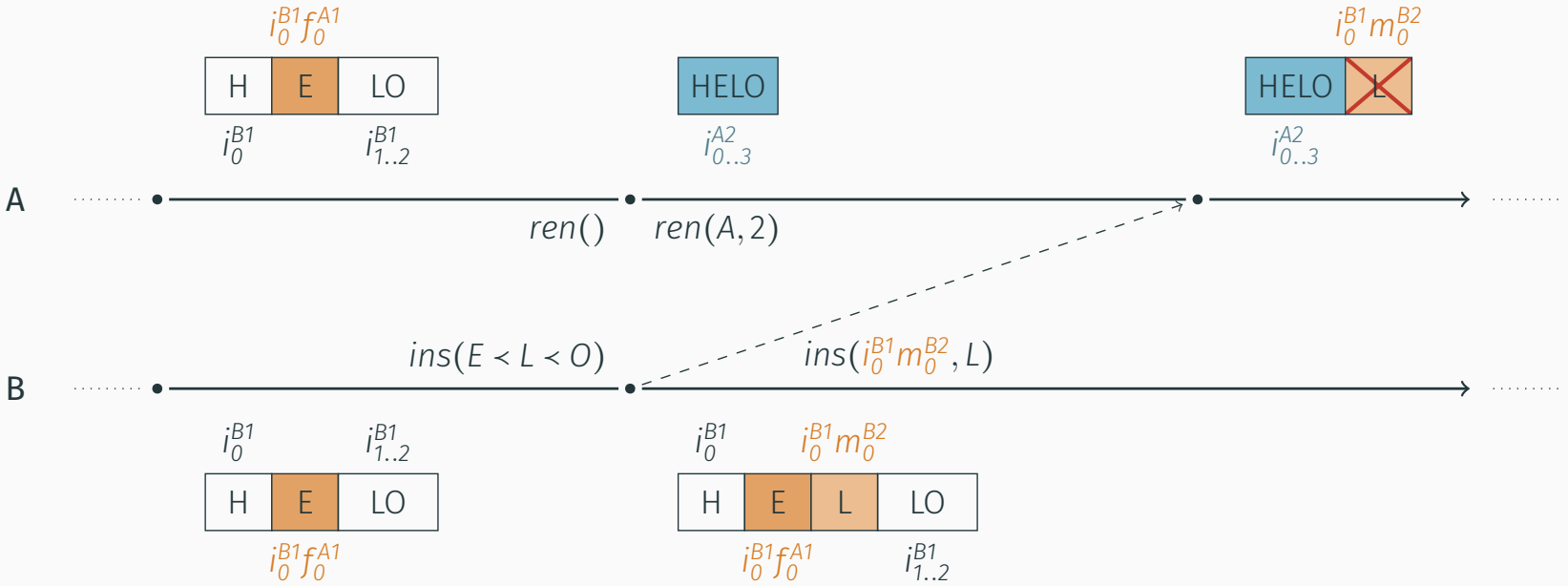
- Peuvent générer opérations concurrentes aux opérations *rename*

Intégration des opérations *insert* et *remove* concurrentes



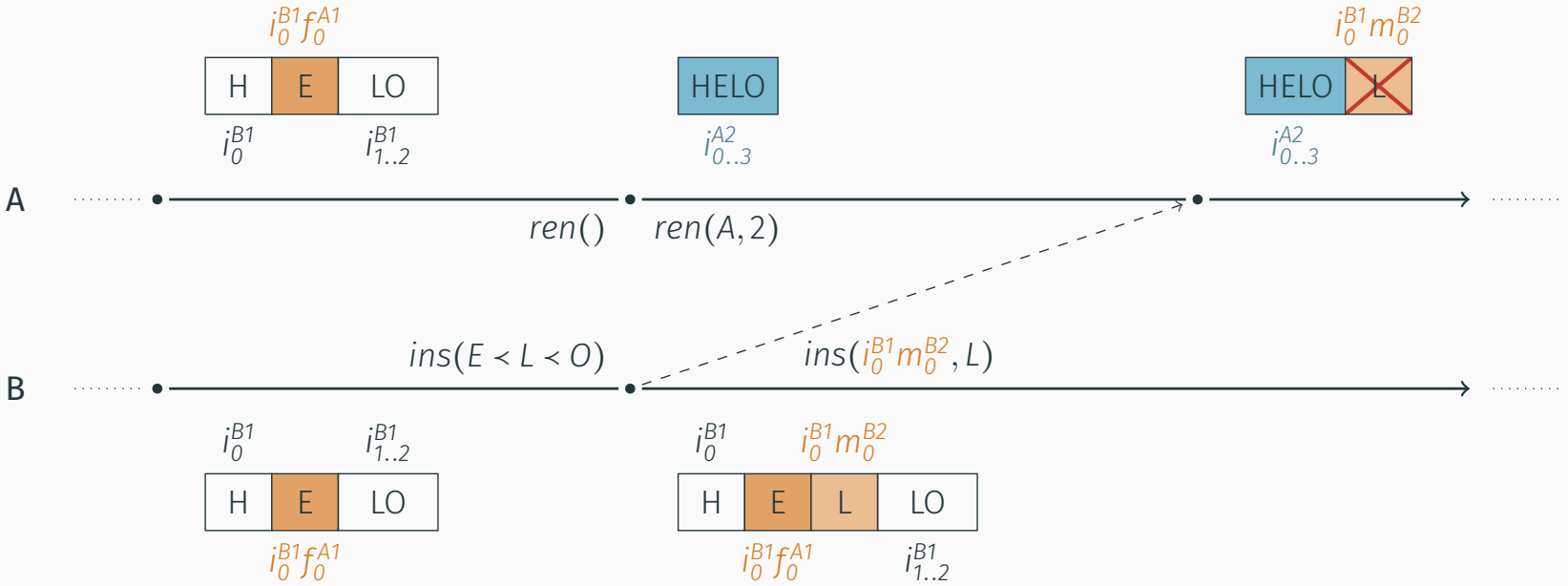
- Peuvent générer opérations concurrentes aux opérations *rename*

Intégration des opérations *insert* et *remove* concurrentes



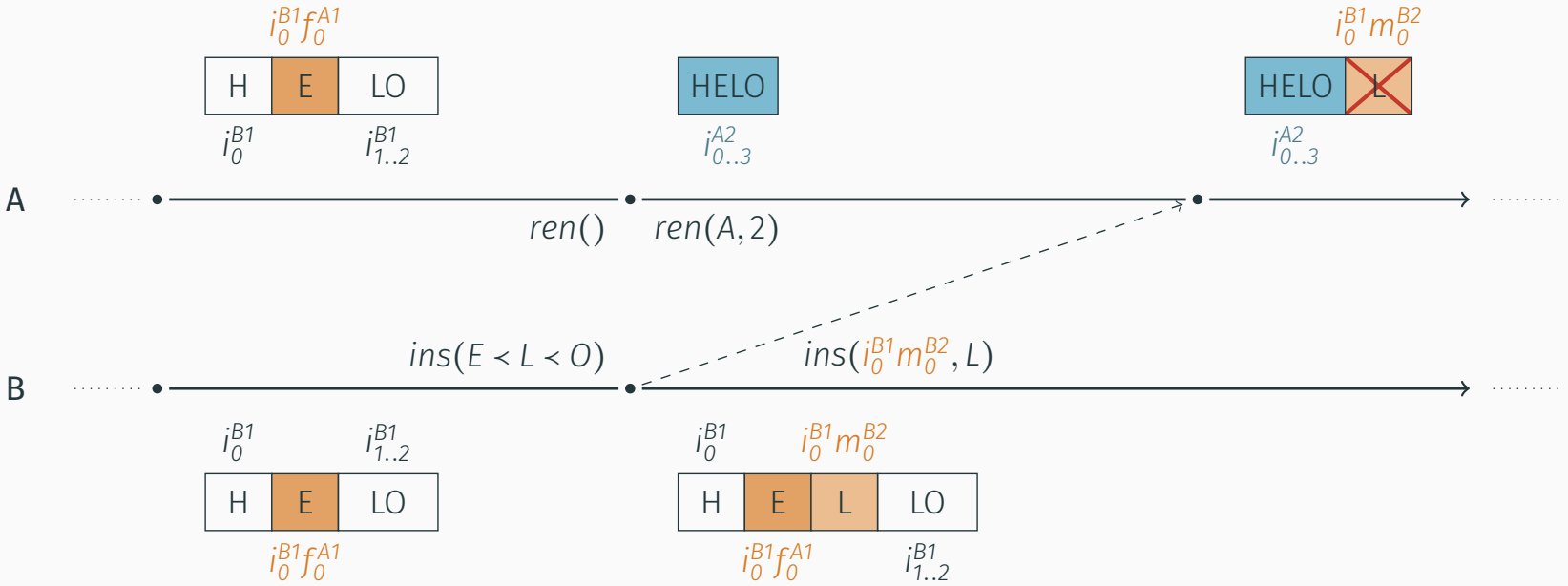
- Peuvent générer opérations concurrentes aux opérations *rename*

Intégration des opérations *insert* et *remove* concurrentes



- Peuvent générer opérations concurrentes aux opérations *rename*
- Produisent anomalies si intégrées naïvement

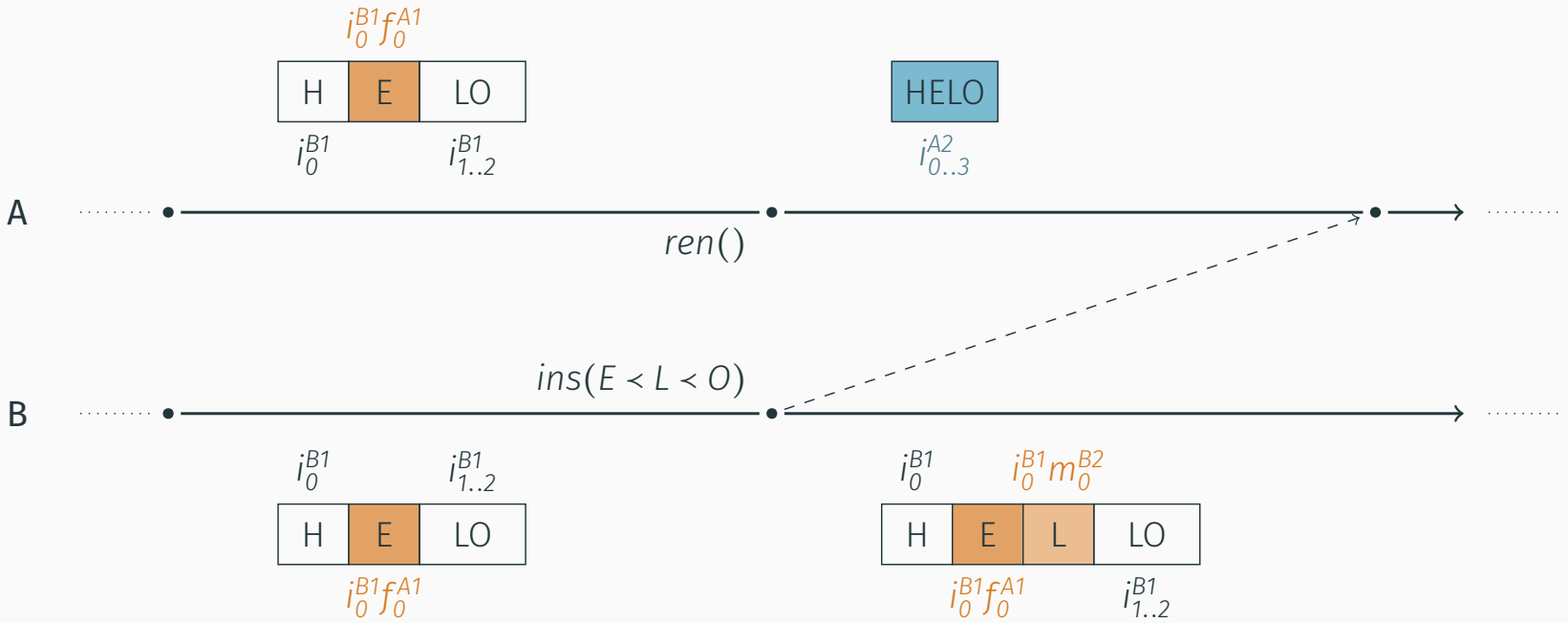
Intégration des opérations *insert* et *remove* concurrentes



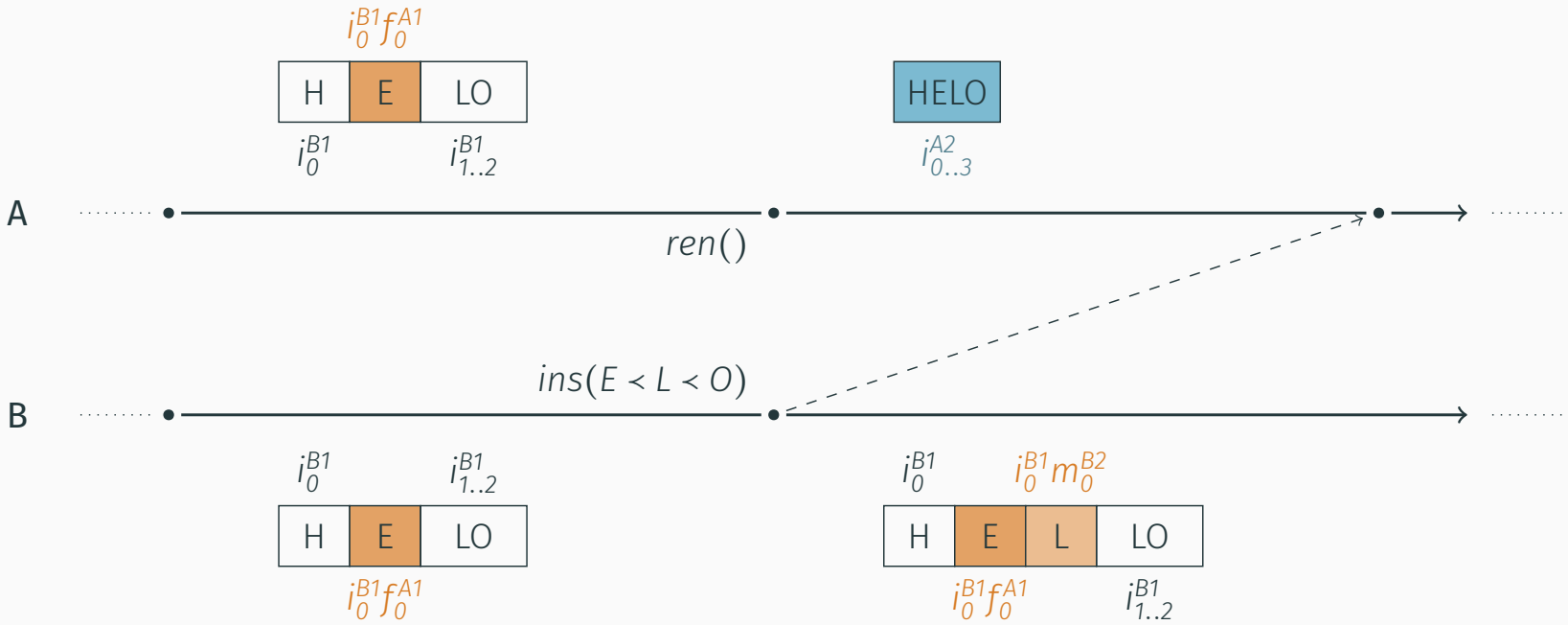
- Peuvent générer opérations concurrentes aux opérations *rename*
- Produisent anomalies si intégrées naïvement

Nécessité d'un mécanisme dédié

Détection des opérations concurrentes à opération *rename*

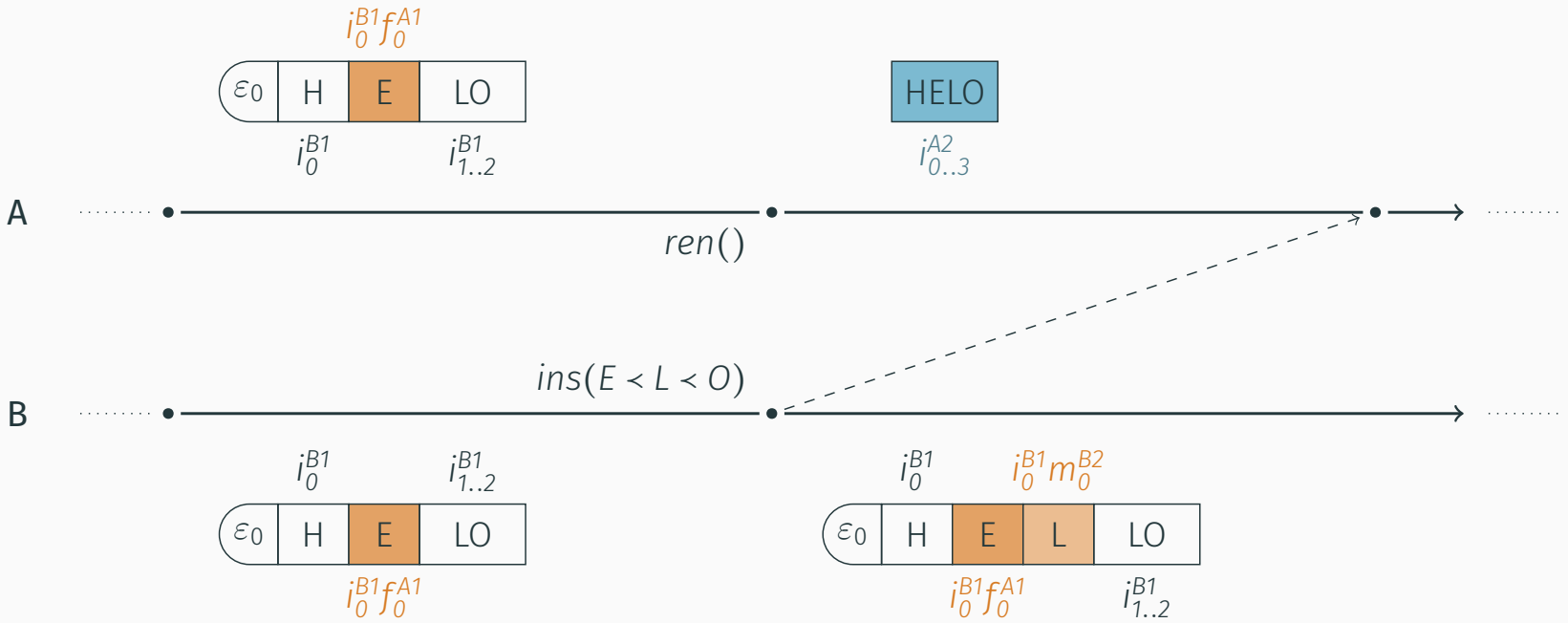


Détection des opérations concurrentes à opération *rename*



Ajout mécanisme d'époques

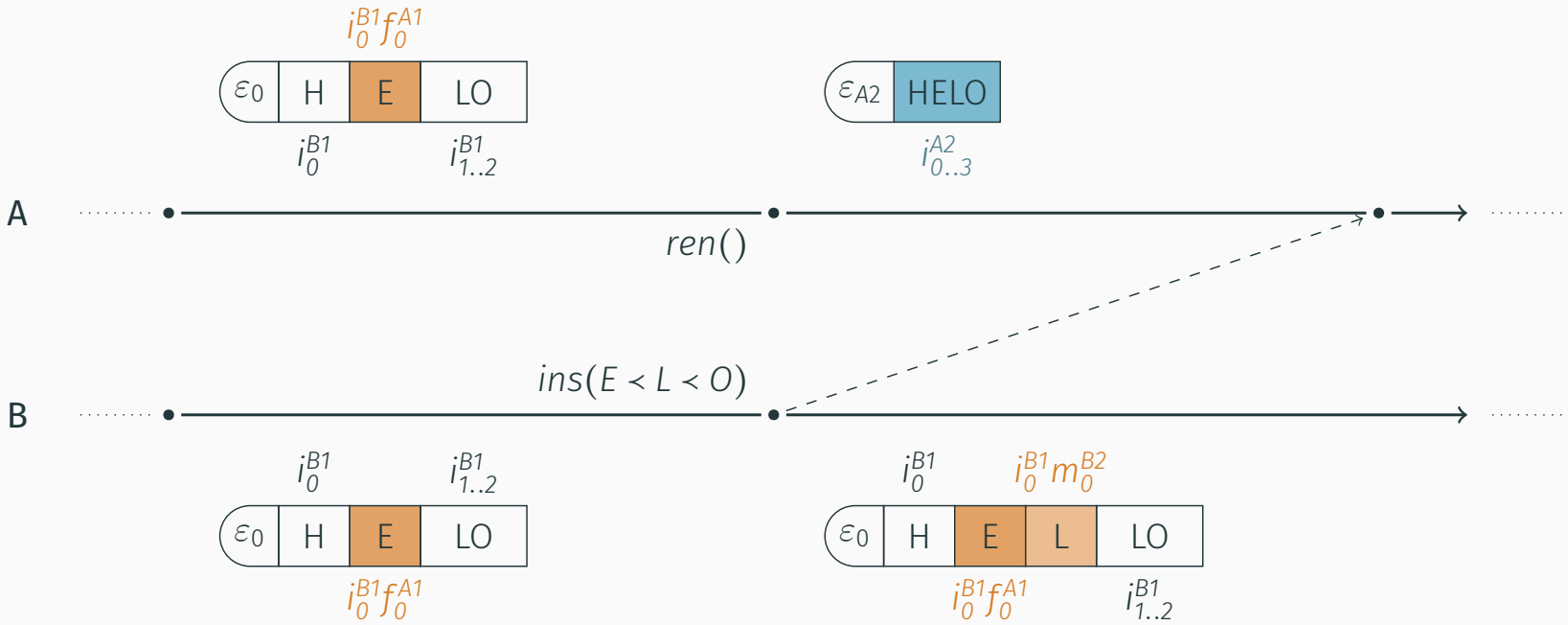
Détection des opérations concurrentes à opération *rename*



Ajout mécanisme d'époques

- Séquence commence à époque d'origine, notée ε_0

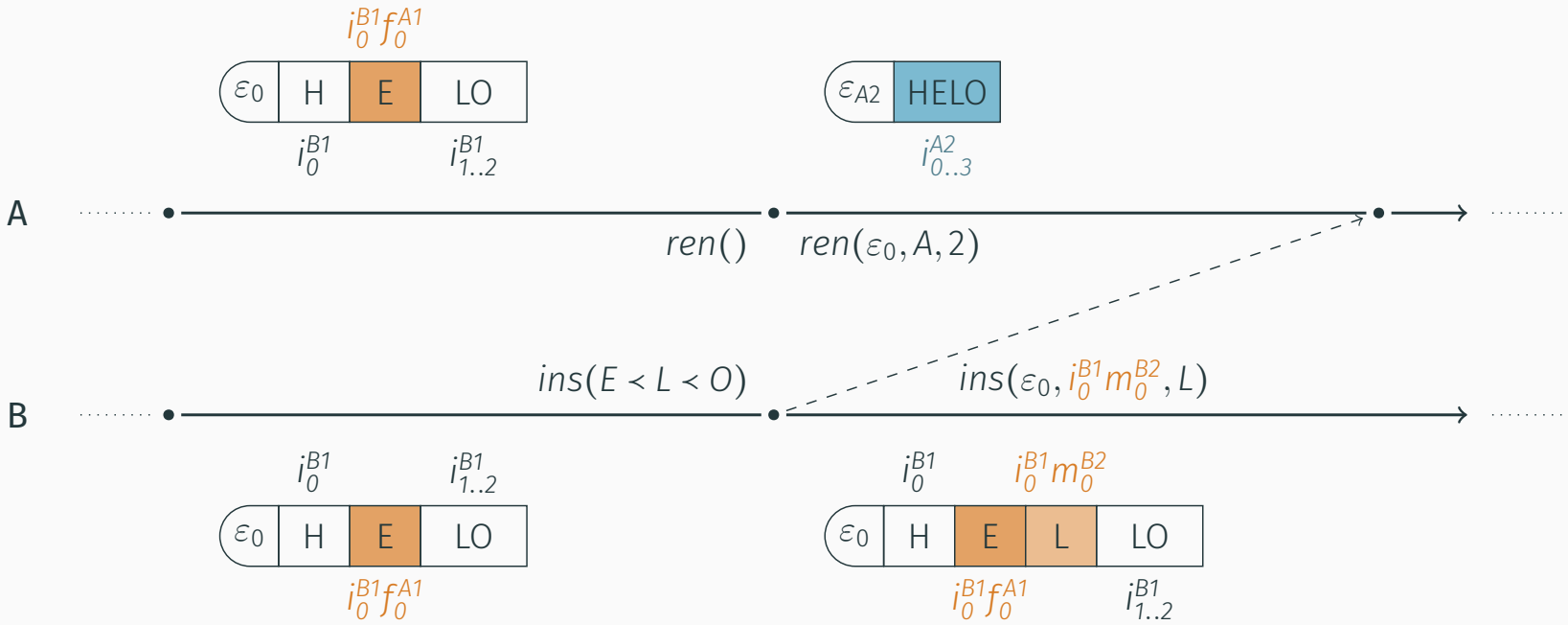
Détection des opérations concurrentes à opération *rename*



Ajout mécanisme d'époques

- Séquence commence à époque d'origine, notée ε_0
- *rename* font progresser à nouvelle époque, $\varepsilon_{nodeId\ nodeSeq}$

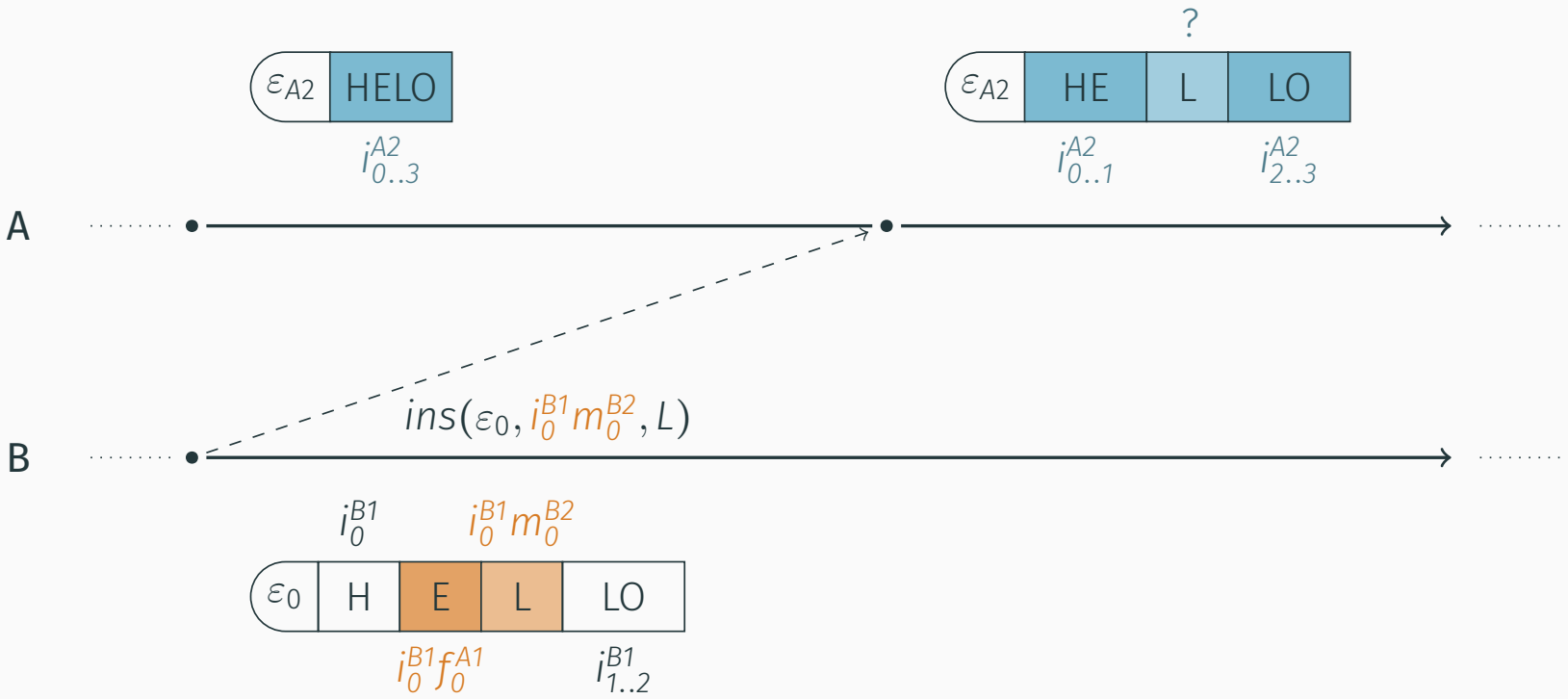
Détection des opérations concurrentes à opération *rename*



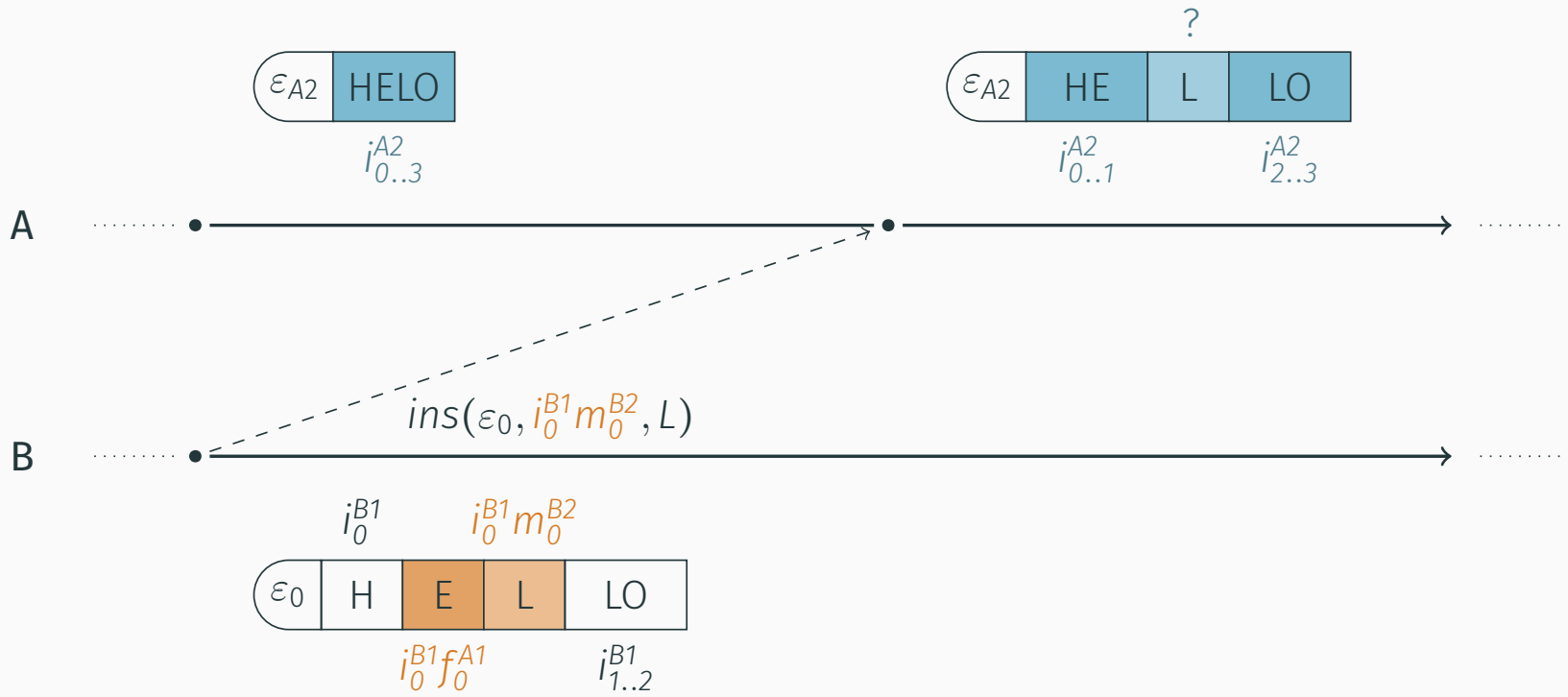
Ajout mécanisme d'époques

- Séquence commence à époque d'origine, notée ε_0
- *rename* font progresser à nouvelle époque, $\varepsilon_{nodeId\ nodeSeq}$
- Opérations labellisées avec époque de génération

Gestion des opérations *insert* et *remove* concurrentes

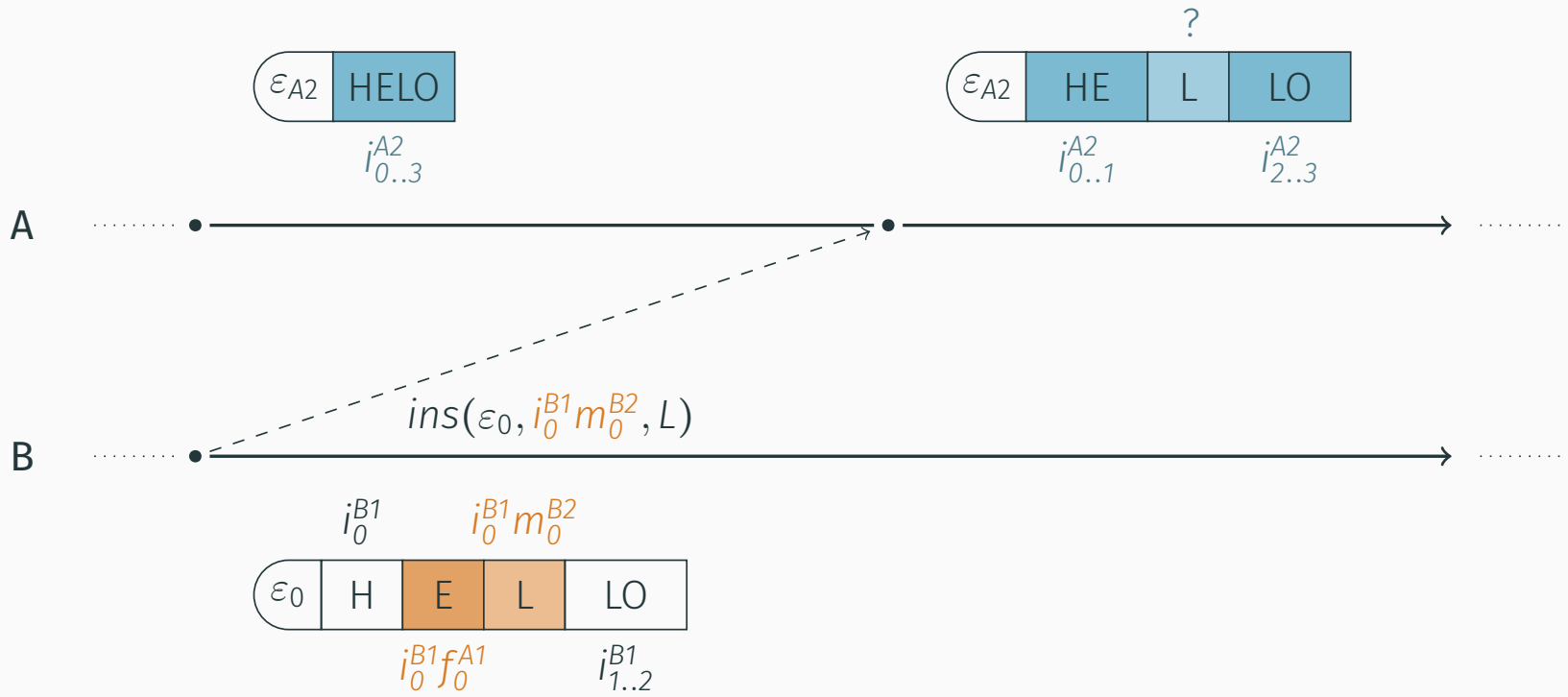


Gestion des opérations *insert* et *remove* concurrentes



Ajout d'un mécanisme de transformation des opérations *insert* et *remove* concurrentes

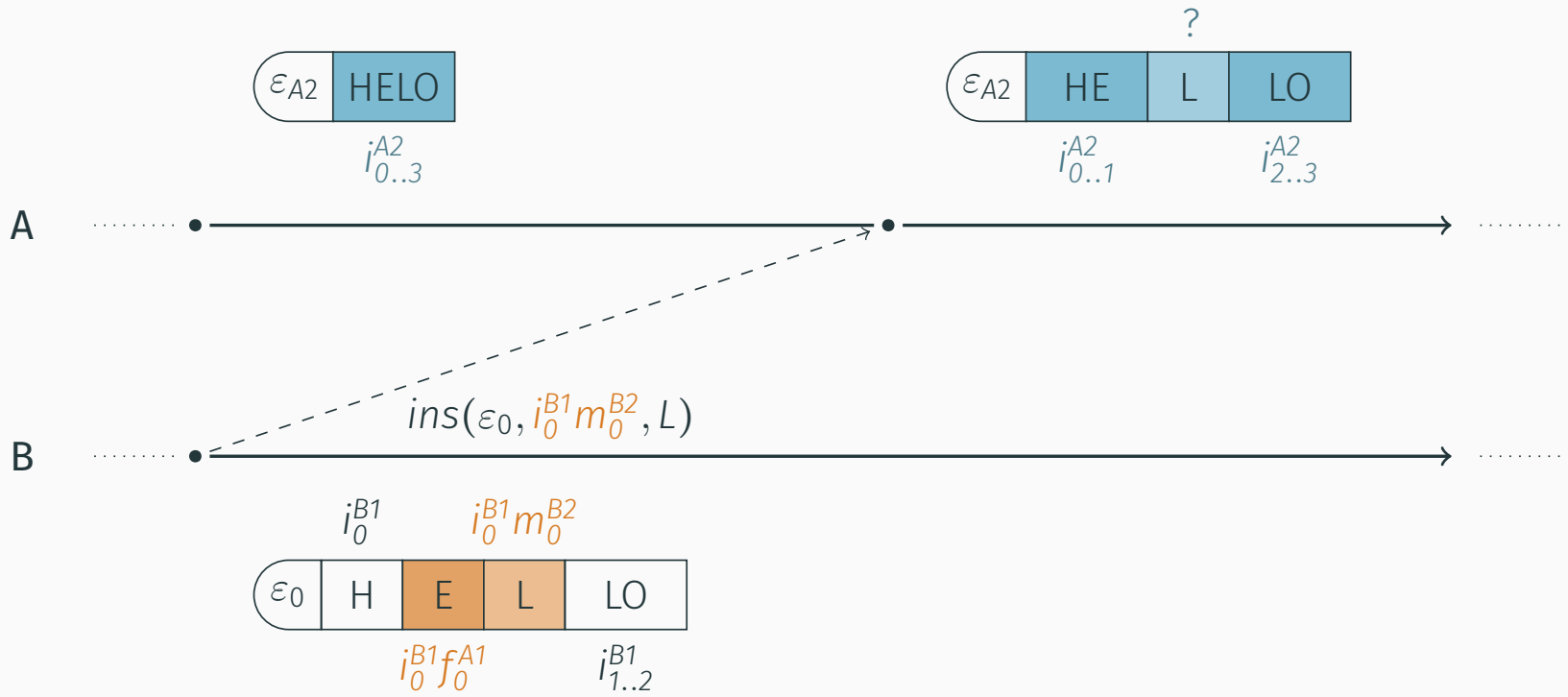
Gestion des opérations *insert* et *remove* concurrentes



Ajout d'un mécanisme de transformation des opérations *insert* et *remove* concurrentes

- Prend la forme de l'algorithme `renameId`

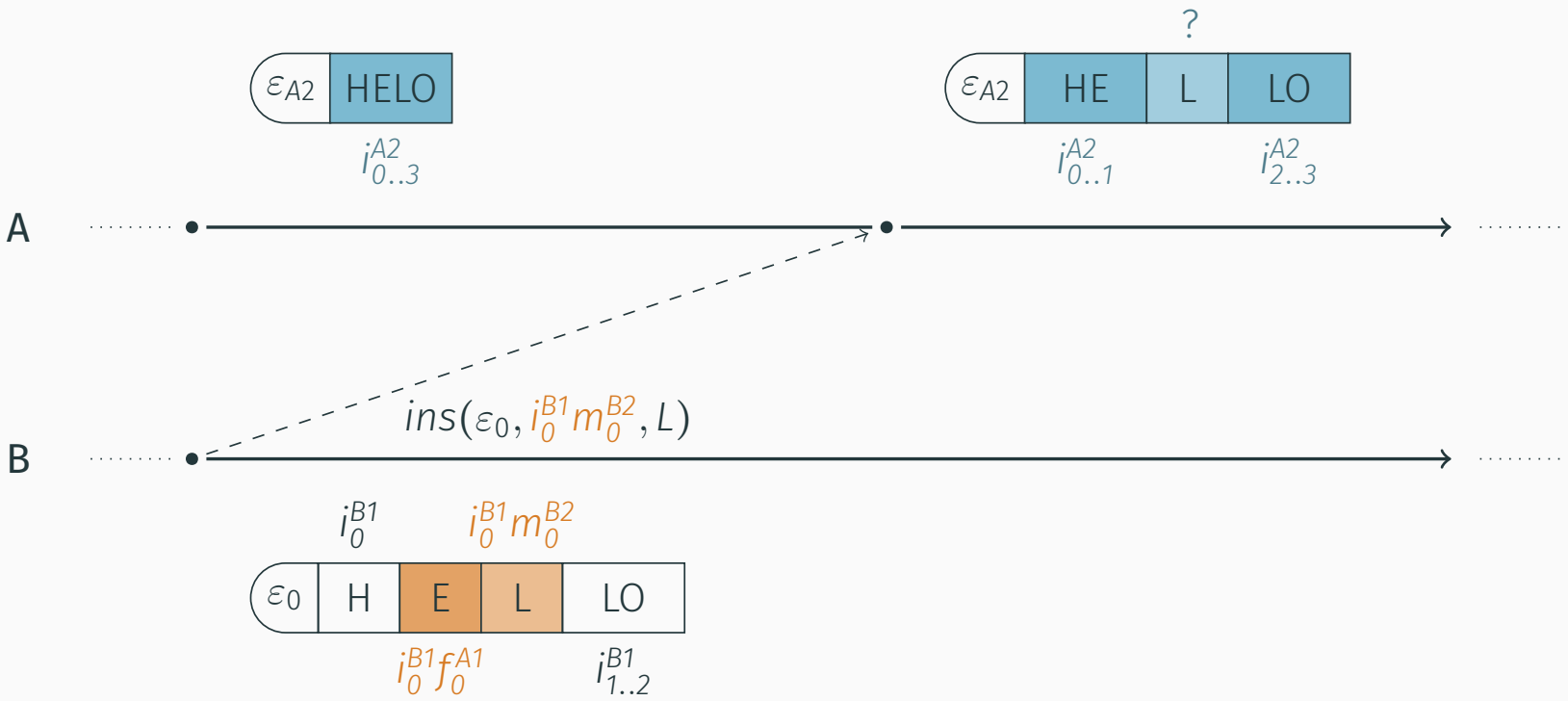
Gestion des opérations *insert* et *remove* concurrentes



Ajout d'un mécanisme de transformation des opérations *insert* et *remove* concurrentes

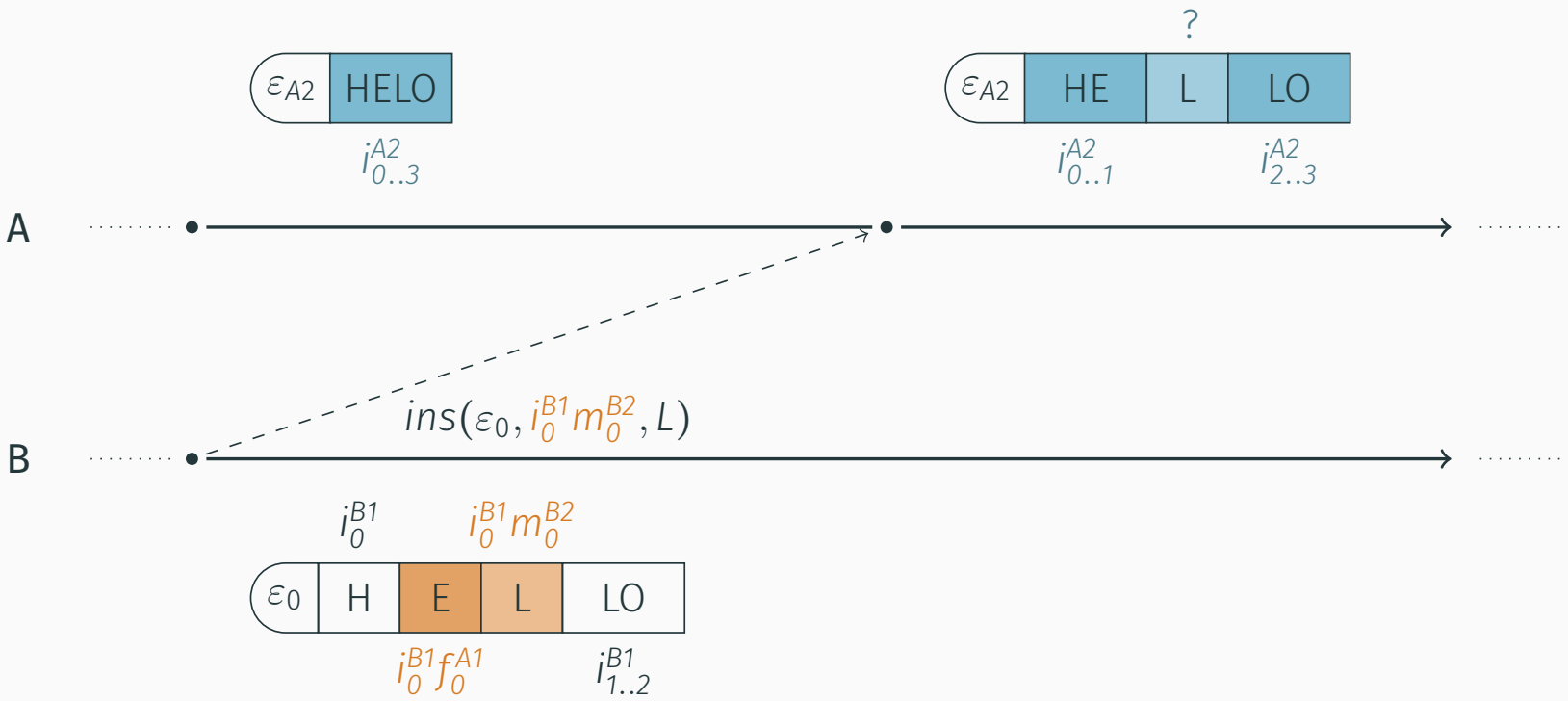
- Prend la forme de l'algorithme `renameId`
- Inclure l'effet de l'opération *rename* dans l'opération transformée

Exemple de `renameId`



Exemple avec $i^{B1}_0 m^{B2}_0$

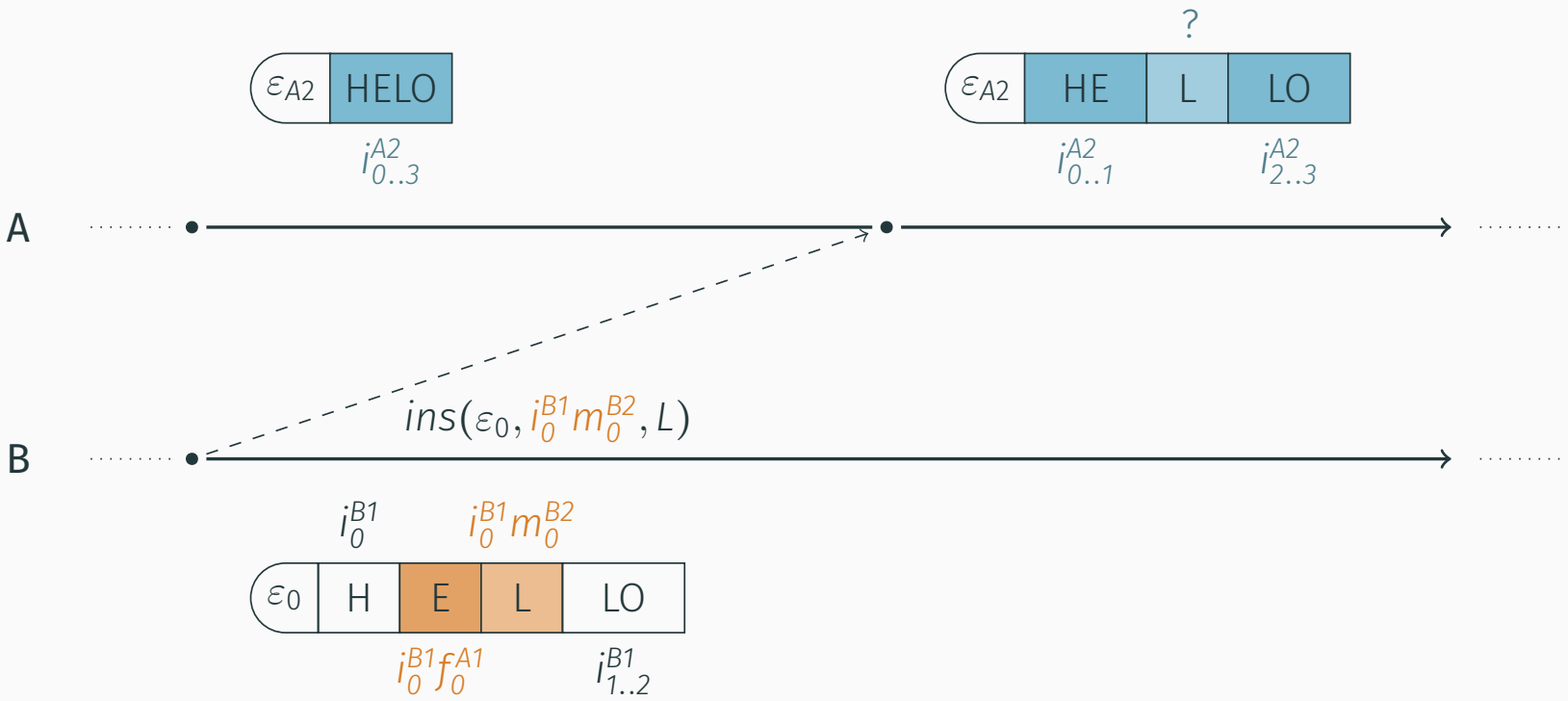
Exemple de `renameId`



Exemple avec $i^{B1}_0 m^{B2}_0$

- Trouver son prédecesseur à l'époque d'origine ε_0 : $i^{B1}_0 f^{A1}_0$

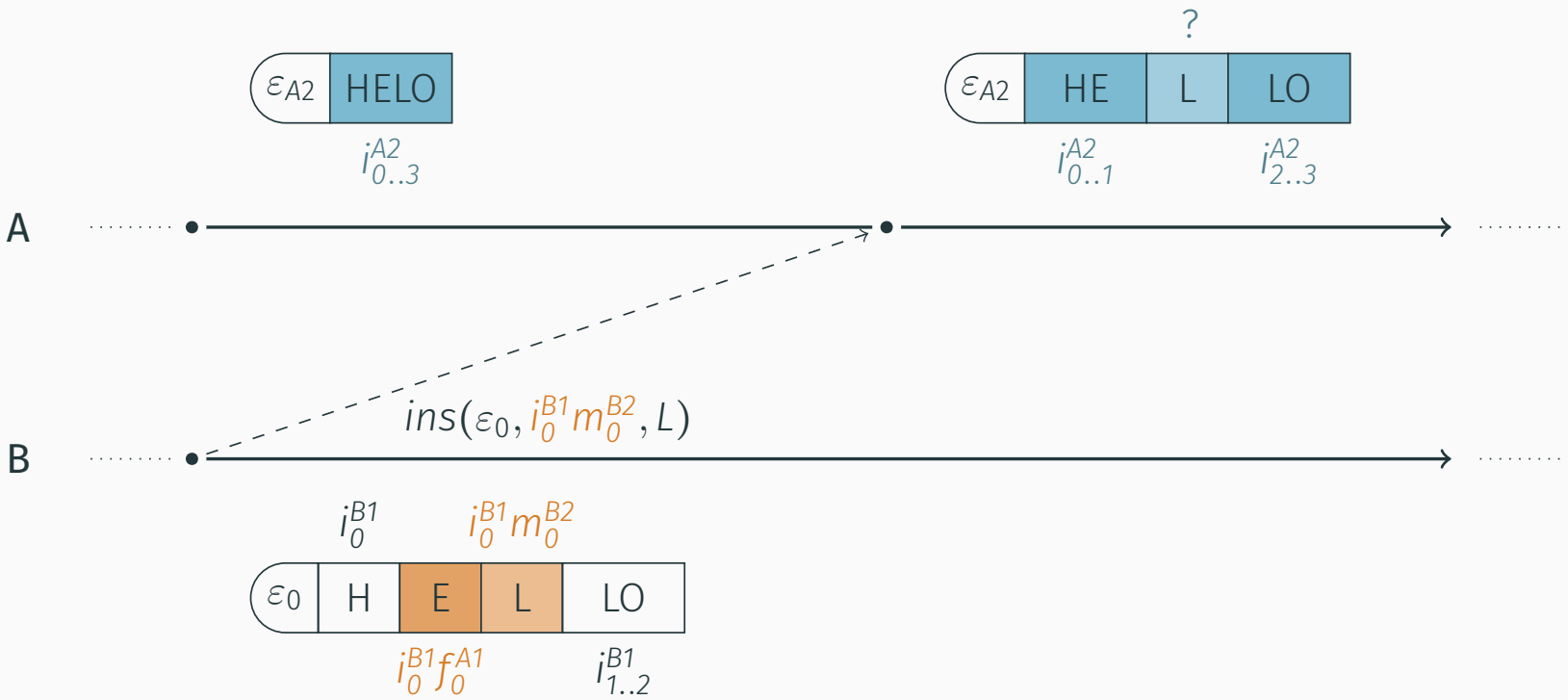
Exemple de `renameId`



Exemple avec $i^{B1}_0 m^{B2}_0$

- Trouver son prédecesseur à l'époque d'origine ε_0 : $i^{B1}_0 f^{A1}_0$
- Trouver son équivalent à l'époque cible ε_{A2} : i^{A2}_1

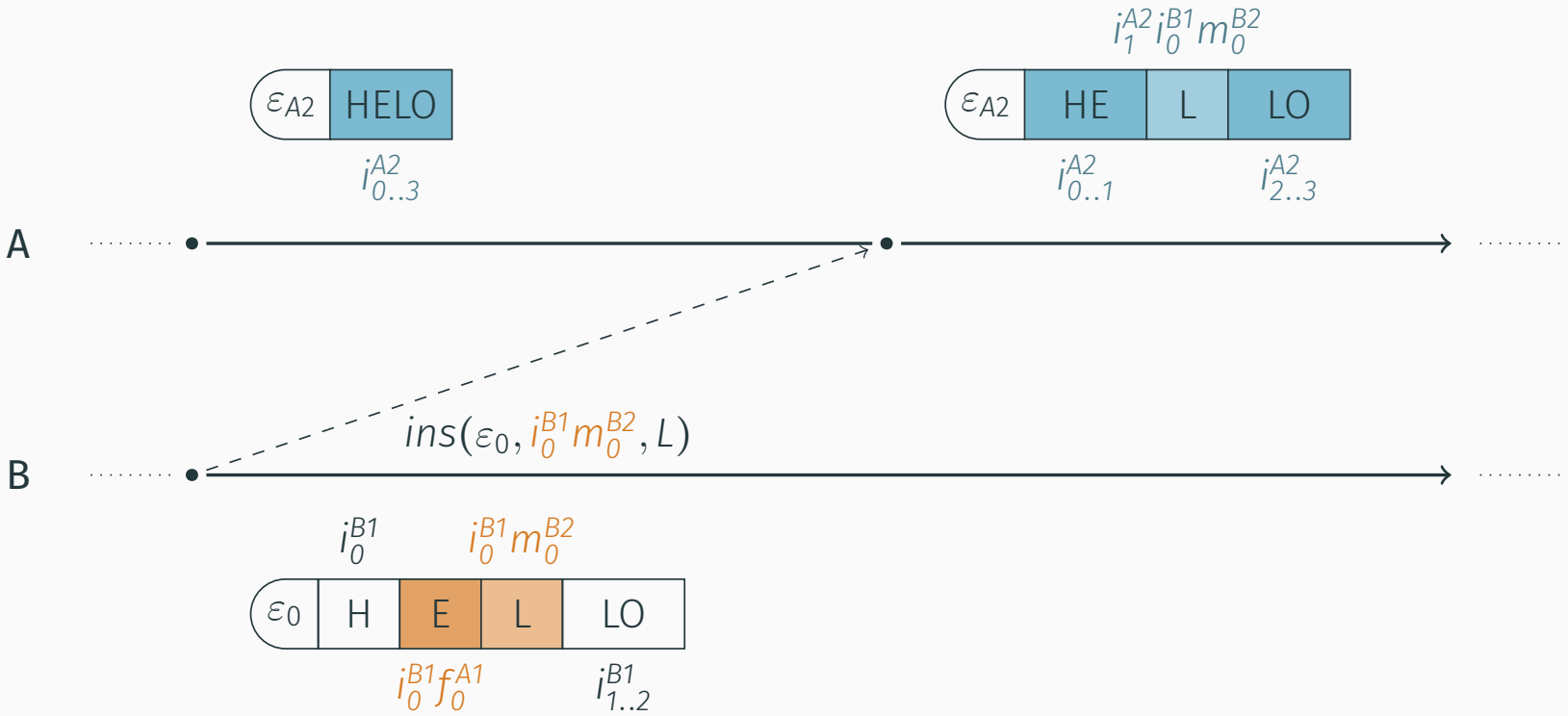
Exemple de `renameId`



Exemple avec $i^{B1}_0 m^{B2}_0$

- Trouver son prédecesseur à l'époque d'origine ε_0 : $i^{B1}_0 f^{A1}_0$
- Trouver son équivalent à l'époque cible ε_{A2} : i^{A2}_1
- Concaténer ce dernier à $i^{B1}_0 m^{B2}_0$ pour obtenir son équivalent à ε_{A2} : $i^{A2}_1 i^{B1}_0 m^{B2}_0$

Exemple de `renameId`

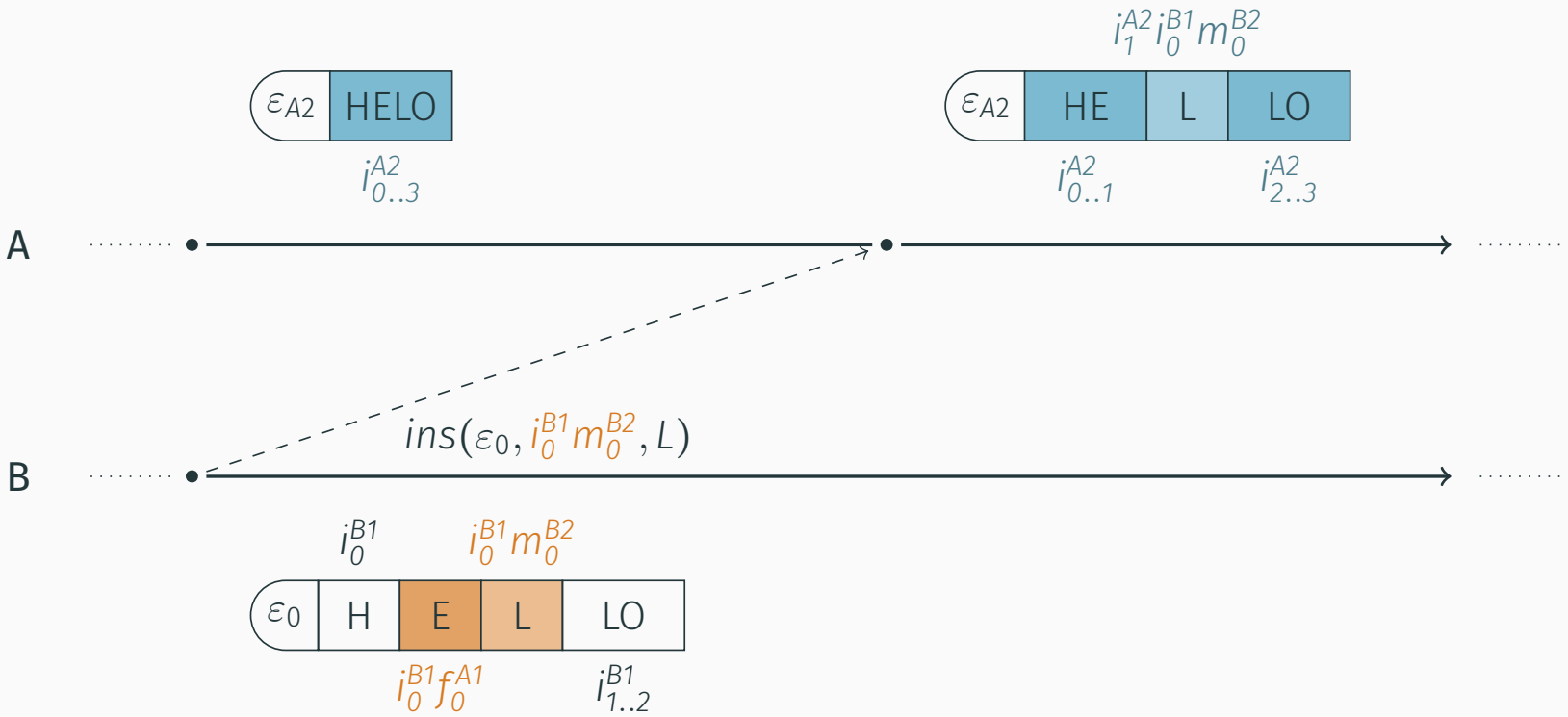


Exemple avec $i_0^{B1} m_0^{B2}$

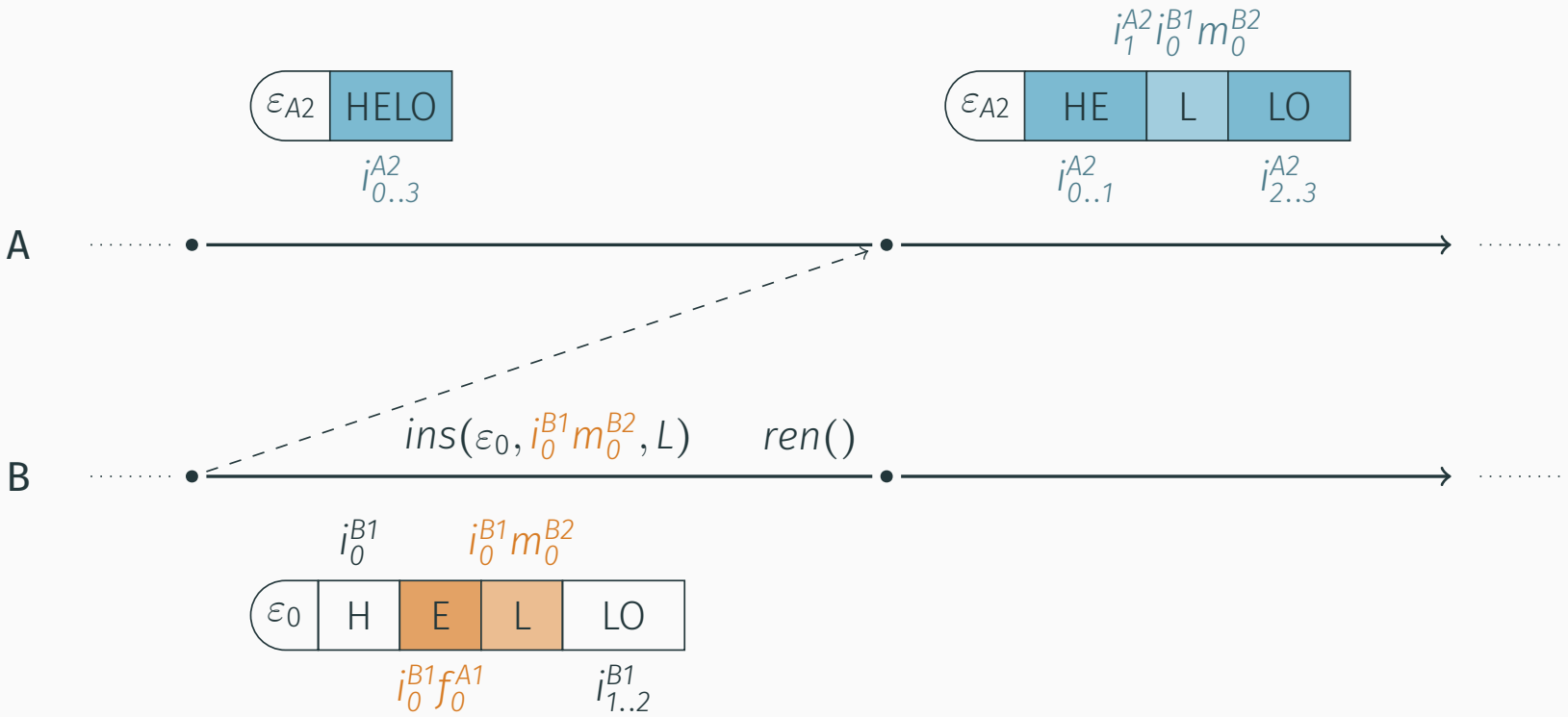
- Trouver son prédecesseur à l'époque d'origine ε_0 : $i_0^{B1} f_0^{A1}$
- Trouver son équivalent à l'époque cible ε_{A2} : i_1^{A2}
- Concaténer ce dernier à $i_0^{B1} m_0^{B2}$ pour obtenir son équivalent à ε_{A2} : $i_1^{A2} i_0^{B1} m_0^{B2}$

Et en cas d’opérations *rename* concurrentes ?

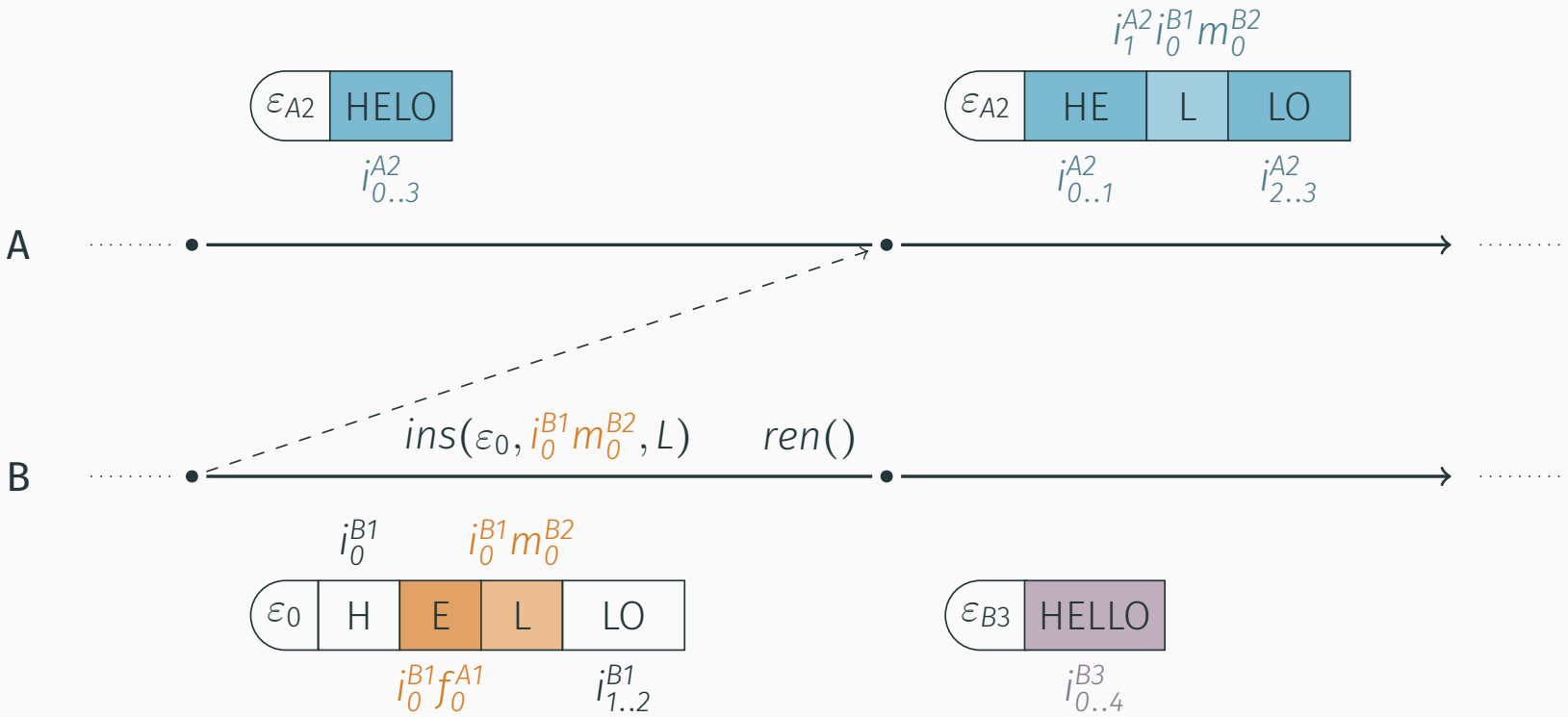
Opérations *rename* concurrentes



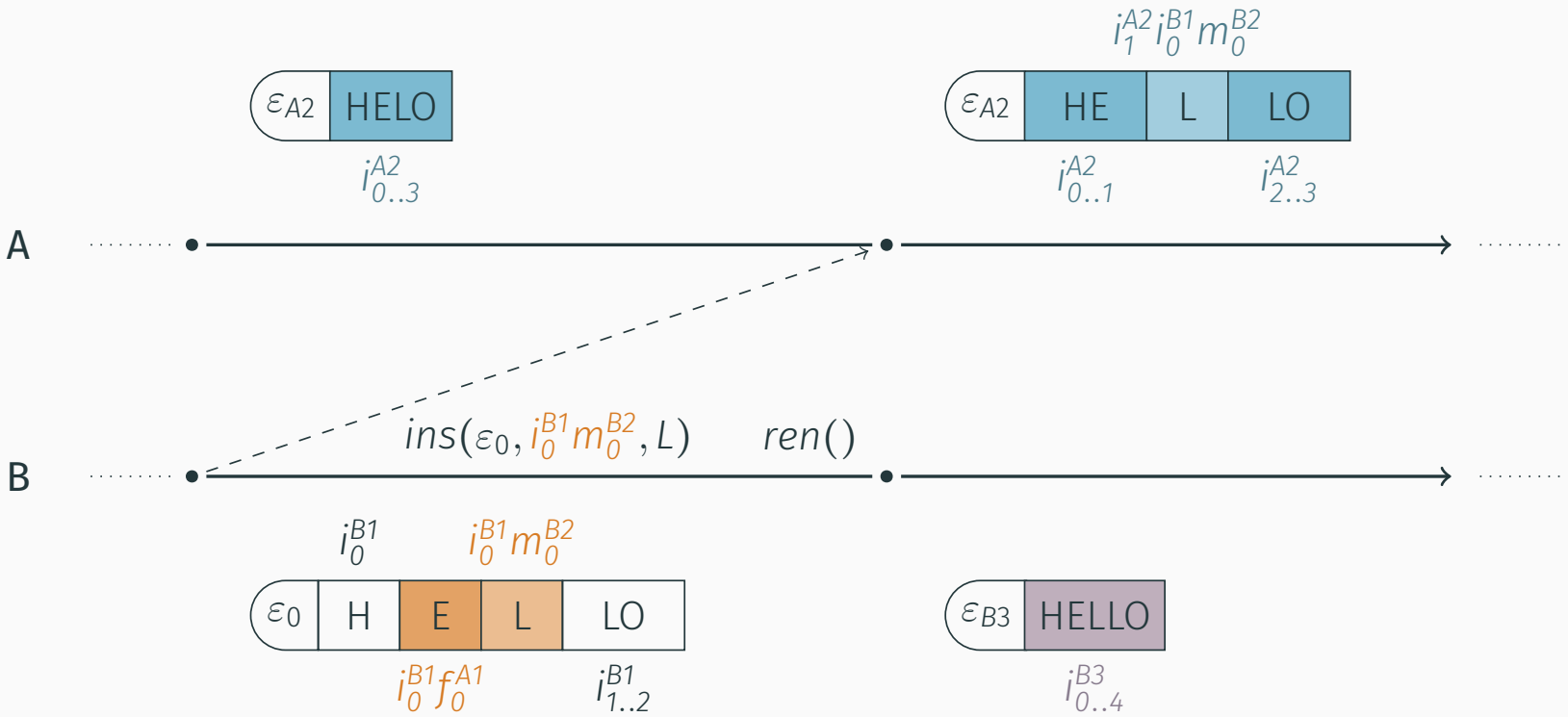
Opérations *rename* concurrentes



Opérations *rename* concurrentes

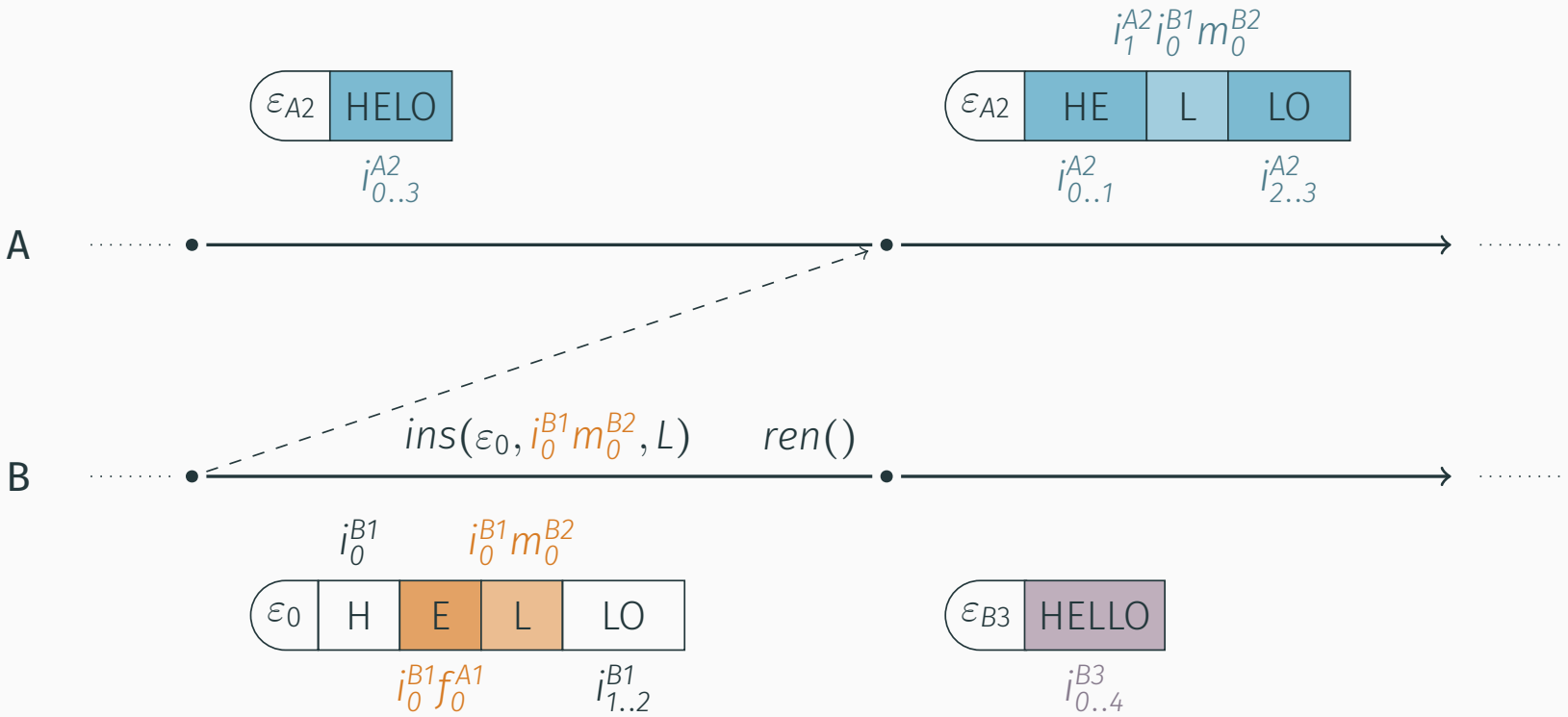


Opérations *rename* concurrentes



Comment faire converger les noeuds ?

Opérations *rename* concurrentes



Comment faire converger les noeuds ?

Besoin d'un mécanisme de résolution de conflits pour faire converger les noeuds

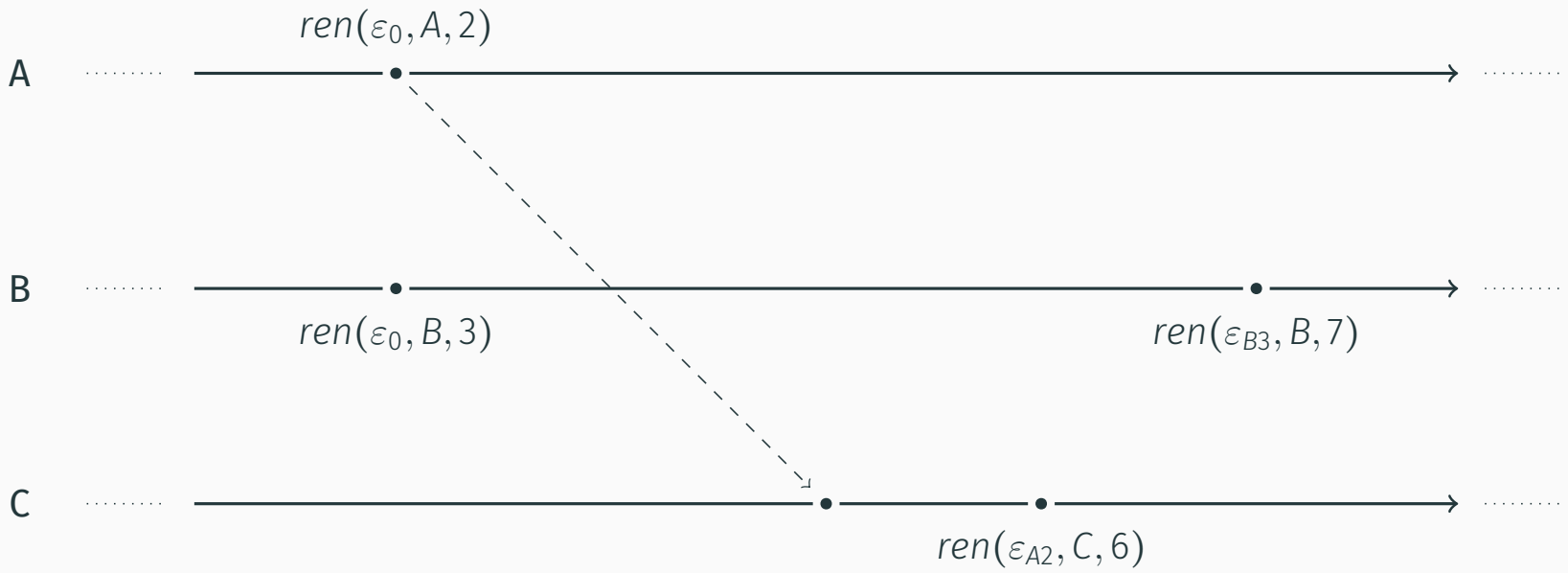
Résolution de conflits entre opérations *rename* concurrentes

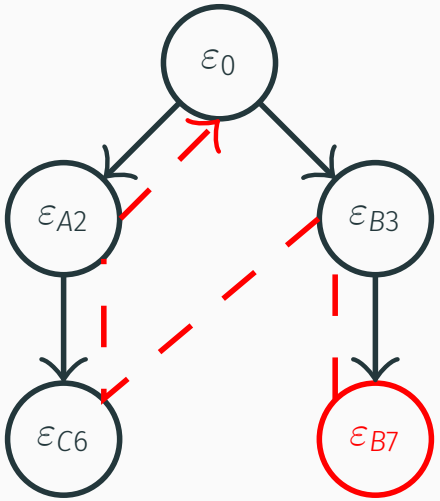
- Opérations *rename* sont des opérations systèmes, c.-à-d. pas d'intention utilisateur
- Peut ne pas appliquer les effets de certaines d'entre elles

Intuition

- Choisir une époque comme époque cible
- Calculer chemin entre époque courante et époque cible, et notamment leur Plus Proche Ancêtre Commun (PPAC)
- Annuler l'effet des opérations *rename* de l'époque courante au PPAC
- Appliquer l'effet des opérations *rename* du PPAC à l'époque cible

Choisir une époque comme époque cible





- Doit définir relation *priority*, notée $<_\varepsilon$, ordre strict total sur les époques
- Utilise ordre lexicographique sur chemins des époques dans l'arbre

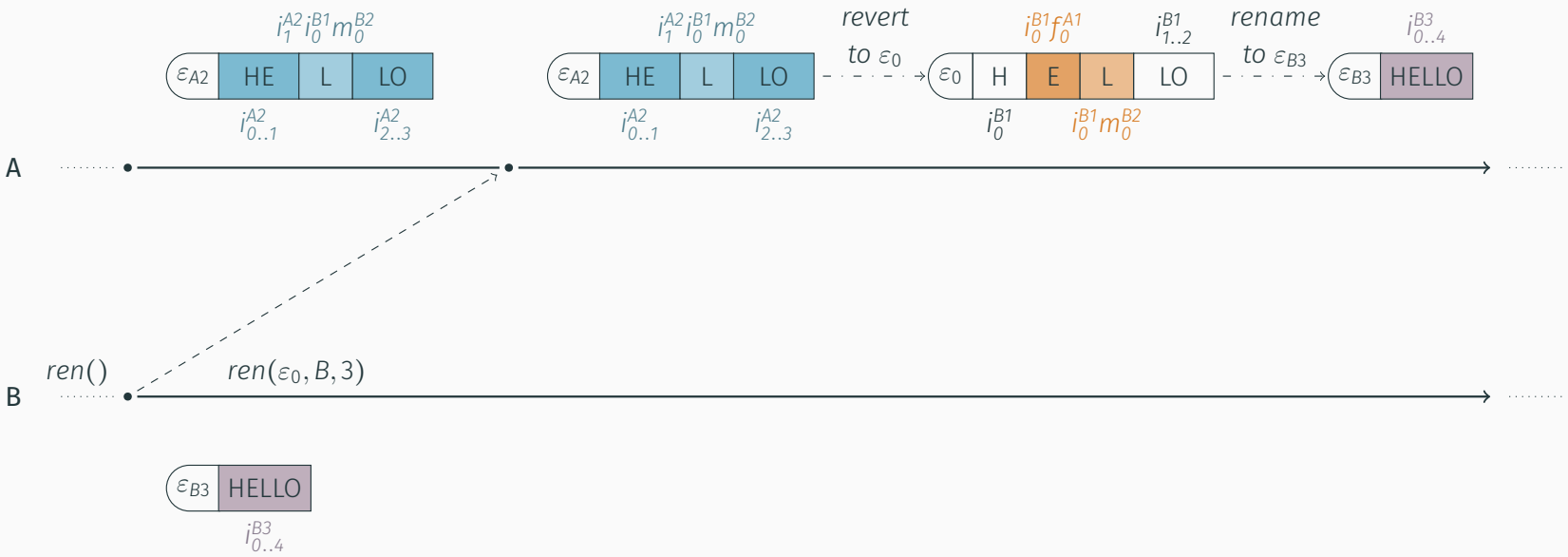
Annuler l'effet d'une opération *rename*

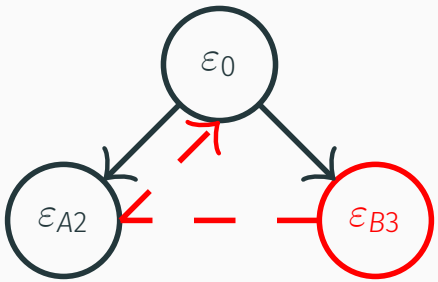
- Propose un nouvel algorithme, *revertRenameId*
- Permet d'obtenir l'identifiant correspondant à l'époque parente

Intuition

1. *id* fait partie des identifiants renommés : doit retourner son ancienne valeur
2. *id* a potentiellement été inséré en concurrence : doit restaurer son ancienne valeur
3. *id* a été inséré après le renommage : doit retourner une valeur qui préserve l'ordre

Opérations *rename* concurrentes





- TODO : Illustrer choix de l'époque cible, cas 1 et cas 2