

PROJET LANCER DE RAYONS : LES OPERATIONS BOOLEENNES

Ce répertoire contient un lancer de rayon calculant les opérations booléennes. Dans ARCHI, il y a une petite animation réalisée avec ce lancer de rayon. ROMAN contient une animation de la surface de Roman.

Il vous faut écrire le fichier bool1D.py, à partir du fichier bool1D.py.initial. Il faut fournir les fonctions :

inter1D(a, b)

union1D(a, b)

differ1D(a, b)

La solution tient en 80 lignes de Python.

inter(a, b) est fournie et nécessite juste un post-traitement de simplification, éliminant les intervalles de longueur nulle et fusionnant les intervalles contigus quand c'est possible : l'intervalle [1, 3.5] suivi de l'intervalle [3.5, 6] devient l'intervalle [1, 6].

vtol convertit un tableau en liste, et ltov convertit une liste en tableau. Ils sont définis dans util.py

Exemples sous l'interprête Python :

```
a=vtol( [(1, 4), (6, 20), (24, 36)])
```

```
b=vtol( [(2, 7), (10, 14), (16, 22), (23, 40), (50, 100)])
```

```
union1D(a, b)
((1, 22), ((23, 40), ((50, 100), None)))
```

```
a inter1D(a, b)
((2, 4), ((6, 7), ((10, 14), ((16, 20), ((24, 36), None)))))
```

```
differ1D( a, b)
((1, 2), ((7, 10), ((14, 16), None)))
```

Testez sur des exemples encore plus simples !

Une fois que les opérations booléennes marchent sur des intervalles numériques, lancer python3 archi.py pour visualiser des opérations booléennes. Utilisez la fonction tof(contact) qui rend contact quand contact est un nombre, et contact.t quand contact est un contact.

REMARQUE : les changements par rapport à la version précédente

une fonction inter_polca(epsilon, polca) a été ajoutée au fichier polynom.py.
inter_polca rend une liste des intervalles, dans [1, infini], où le polynôme polca est négatif. Par exemple, pour le polynôme $t \rightarrow (t-3)(t-5)$, la fonction
`inter_polca(1e-10, ((Var('t') - Nb(3)) * (Var('t') - Nb(5))).topolent)`
rend la liste : ((3,5), None) --modulo l'imprécision numérique.

Dans ray.py, la méthode intersection a été modifiée. Son résultat est maintenant une liste d'intervalles de contacts.

Dans ray.py, des classes Union, Differ, Inter ont été ajoutées. Une méthode intersection avec un rayon a été définie pour chacune de ces classes.