

Rapport Projet - Bicloo

Vérifications et preuves formelles
2023 / 2024

Rapport rédigé par :

ARISTOR Christella

BARRY Saikou yaya

Tables des matières

I. Introduction.....	2
II. Modélisation formelle.....	3
III. Bilan sur Atelier B.....	6
IV. Conclusion.....	9



I. Introduction

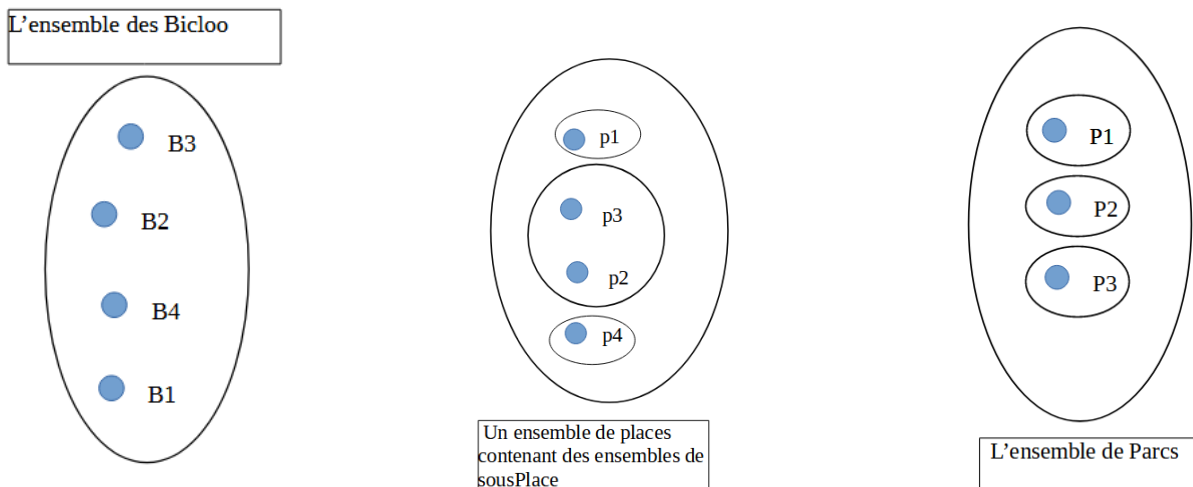
Ce rapport de projet présente l'étude de cas portant sur la modélisation formelle d'un système de gestion Bicloo. Les Bicloo sont des objets (vélos) abstraits et requièrent un parc doté de places pour les ranger. Chaque place ne peut accueillir qu'un Bicloo à la fois, et réciproquement, chaque Bicloo occupe une place unique.

Le rapport est structuré en deux grandes parties : la modélisation formelle illustrée avec des schémas avec les ensembles, relations, fonctions, ... et le bilan de l'expérimentation de Atelier B. La conclusion synthétise les résultats et souligne les points forts et les limites du modèle proposé.

II. Modélisation formelle

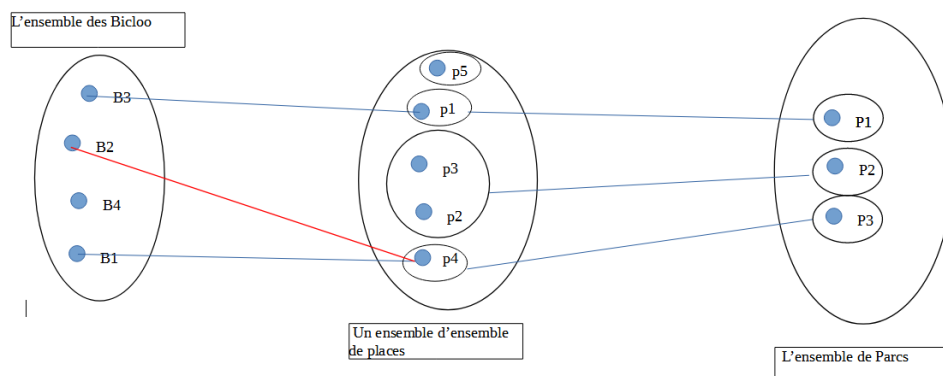
Pour formaliser le système de gestion de Bicloo, nous avons défini trois ensembles principaux et 3 sous-ensembles :

- Bicloo : représentant l'ensemble de tous les Bicloo
- Parc : pour l'ensemble de tous les parcs
- Place : pour l'ensemble des places dans tous les parcs
- bicloos : pour accéder à chaque bicloo
- parcs : pour accéder à chaque parc
- sousPlaces : pour un ensemble de place appartenant à un parc

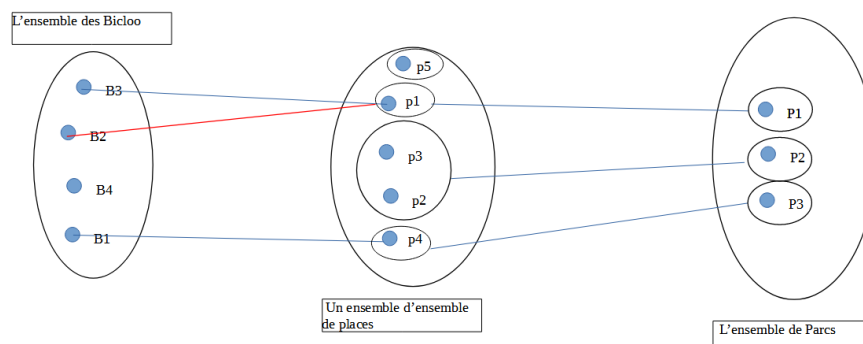


Le respect des propriétés énoncées a été intégré à la modélisation les schémas suivants illustrent les contraintes suivantes (les éléments en rouge représente la propriété) :

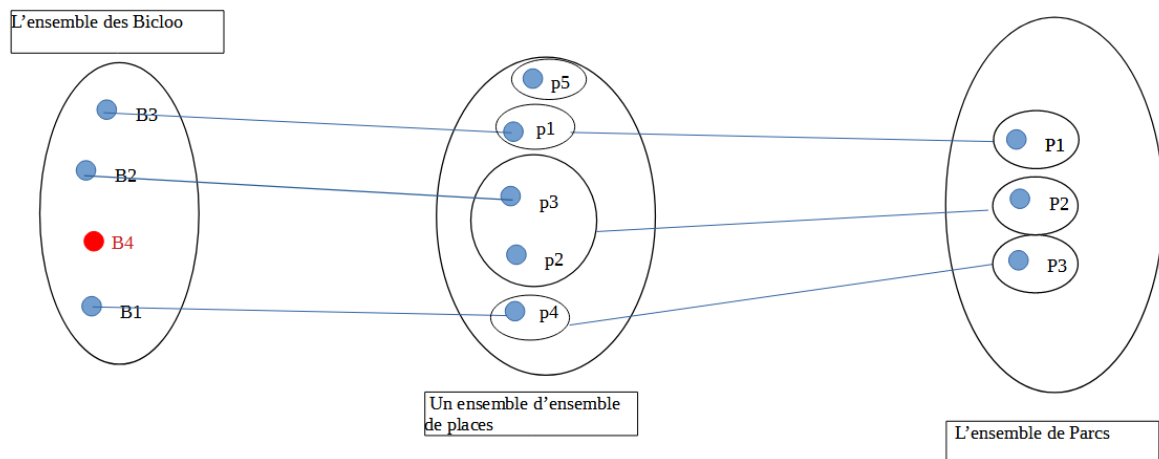
1. Aucun parc ne peut contenir plus de Bicloo que le nombre initial.



2. Aucun parc ne peut accueillir plus de Bicloo que de places disponibles.

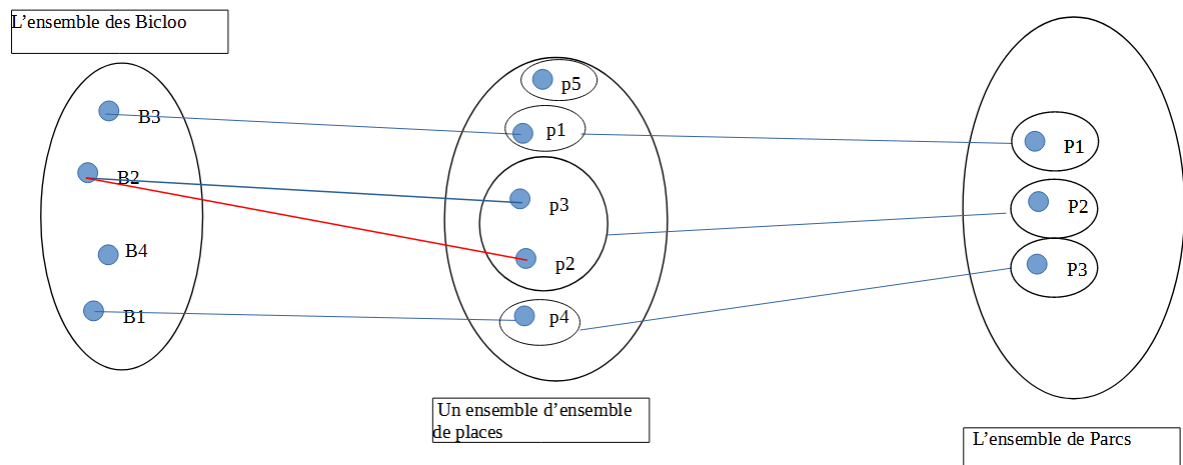


3. À tout moment, un Bicloo peut ne pas être situé dans l'un des parcs.

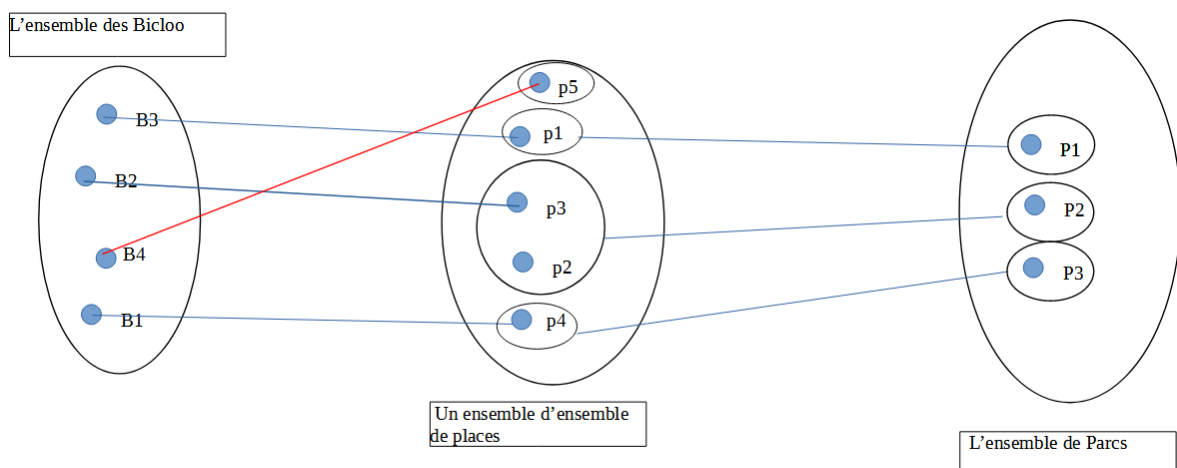


Nous avons aussi défini nos propres propriétés pour compléter le système de Bicloo :

- Un Bicloo ne peut pas occuper 2 places



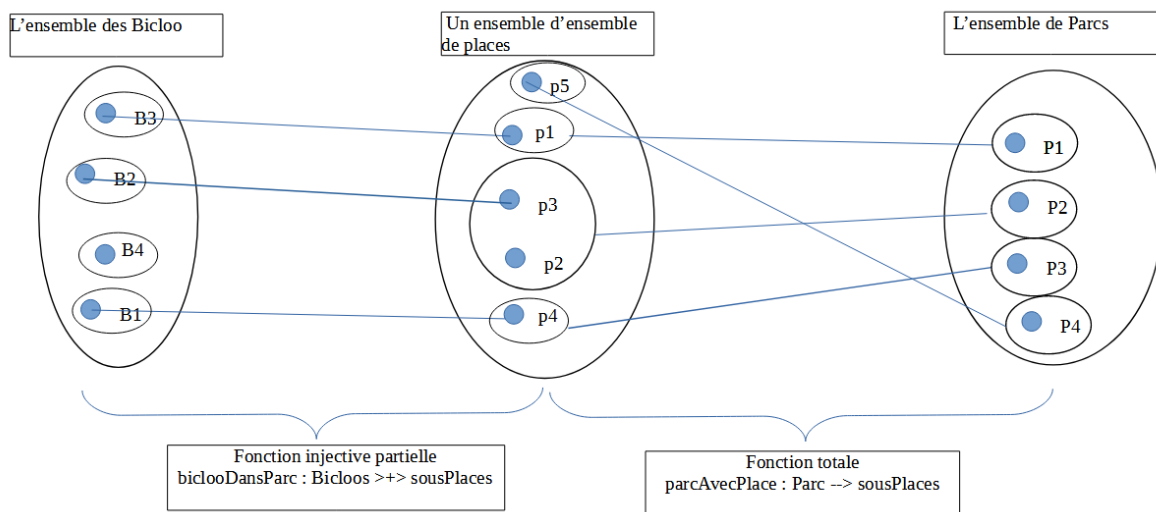
- Un Bicloo a une place seulement si la place est dans un Parc



Initialement, tous les ensembles et fonctions sont initialisés avec l'ensemble vide.

Trois fonctions ont été définies pour formaliser davantage notre système :

- `biclooDansParc`, définissant une injection partielle associant chaque Bicloo à un parc spécifique :
 - `biclooDansParc : Bicloos >+> Place`
- `parcAvecPlace`, définissant une fonction totale associant chaque parc à un sous-ensemble `sousPlace`
 - `parcAvecPlace : Parc --> sousPlaces`
- `sortiDeParc` est une fonction qui permet de définir si un bicloo est stationné sur une place ou non
 - `sortiDeParc : Bicloos --> BOOL`



•

III. Bilan sur Atelier B

Pour la modélisation en langage B, deux constantes ont été définies pour borner le nombre maximal de Bicloo et de places :

- maxBicloos, avec des propriétés définies de 1 à MAXINT
- maxPlaces, également avec des propriétés de 1 à MAXINT

Les invariants que nous avons définis sont les suivants :

1. bicloos <: Bicloos : L'ensemble 'bicloos' est un sous-ensemble de l'ensemble 'Bicloos'. Cela indique que tous les éléments présents dans 'bicloos' sont également des éléments de 'Bicloos'.

2. parcs <: Parc : L'ensemble 'parcs' est un sous-ensemble de l'ensemble 'Parc'. Cela implique que tous les éléments de 'parcs' font partie de l'ensemble 'Parc'.

3. sousPlaces <: Place : L'ensemble 'sousPlaces' est un sous-ensemble de l'ensemble 'Place'. Cela indique que tous les éléments de 'sousPlaces' sont également des éléments de 'Place'.

4. biclooDansParc : Bicloos >+> sousPlaces : Cette notation définit une relation entre les éléments de 'bicloos' et les éléments de 'sousPlaces'. Chaque élément dans 'bicloos' est associé ou non à un élément de 'sousPlaces'.

5. parcAvecPlace : Parc >+> sousPlaces : Cela établit une relation entre les éléments de 'Parc' et les éléments de 'sousPlaces'. Chaque Bicloo est lié à un sous-ensemble de place spécifique illustrant l'emplacement attribué au parc.

6. sortiDeParc : Bicloos → BOOL : Cette expression établit une relation entre les éléments de 'Bicloos' et des valeurs booléennes (TRUE ou FALSE). Chaque élément dans 'Bicloos' est associé à une valeur booléenne, indiquant si le Bicloo correspondant est actuellement hors d'un parc (TRUE) ou non (FALSE). Cela permet de mémoriser l'état de sortie de chaque Bicloo par rapport aux parcs.

En résumé ces invariants sont utilisés pour garantir des propriétés importantes du système tout au long de son exécution, ils contribuent à maintenir la cohérence des données et à s'assurer que les opérations définies dans la machine abstraite respectent ces propriétés

Opérations :

- SortirBicloo
 - Paramètres d'entrée
 - Bicloo : le bicloo à sortir

- Place : la place du bicloo
- Description : Cette opération permet de signaler qu'un Bicloo donné est sorti d'un parc spécifique.
- Préconditions :
 - Le Bicloo doit être dans l'ensemble Bicloos.
 - La place spécifiée doit être dans l'ensemble Place.
 - Le Bicloo doit actuellement être dans le parc spécifié.
 - Le Bicloo ne doit pas avoir déjà été signalé comme sorti.
- Effet : La variable d'état sortiDeParc est mise à jour, signalant que le Bicloo est maintenant sorti du parc.

➤ RangerBicloo

- Paramètres d'entrée :
 - bicloo : le bicloo à ranger
 - place : la place qui lui a été attribué
- Description : Cette opération permet de ranger un Bicloo dans un parc spécifié.
- Préconditions :
 - Le Bicloo ne doit pas déjà être dans l'ensemble bicloos.
 - La place spécifiée doit être dans l'ensemble Place.
 - Le nombre total de Bicloo dans les parcs doit être inférieur à la limite maxBicloos.
- Effet :
 - Le Bicloo est ajouté à l'ensemble bicloos.
 - La place est ajoutée à l'ensemble sousPlaces.
 - La relation biclooDansParc est mise à jour pour indiquer le nouveau placement du Bicloo.
 - La variable d'état sortiDeParc est mise à jour pour indiquer que le Bicloo n'est pas sorti.

➤ AjouterParc

- Paramètre d'entrée : nouveauParc
- Description : Cette opération permet d'ajouter un nouveau parc à l'ensemble des parcs.
- Préconditions : Le nouveau parc ne doit pas déjà être dans l'ensemble des parcs.
- Effet : Le nouveau parc est ajouté à l'ensemble des parcs.

➤ SupprimerParc

- Paramètre d'entrée : parcASupprimer
- Description : Cette opération permet de supprimer un parc spécifié de l'ensemble des parcs.
- Préconditions : Le parc à supprimer doit être dans l'ensemble des parcs.
- Effet : Le parc spécifié est retiré de l'ensemble des parcs.

➤ DéplacerBicloo

- Paramètres d'entrée :
 - bicloo
 - nvPlace
- Description : Cette opération permet de déplacer un Bicloo vers une nouvelle place spécifiée.
- Préconditions :
 - Le Bicloo doit être dans l'ensemble Bicloos.
 - La nouvelle place doit être dans l'ensemble Place.
 - Le Bicloo ne doit pas être déjà dans la nouvelle place.
- Effet :
 - La place précédente du Bicloo est retirée de l'ensemble sousPlaces.
 - La nouvelle place est ajoutée à l'ensemble sousPlaces.
 - La relation biclooDansParc est mise à jour pour refléter le déplacement du Bicloo vers la nouvelle place.

IV. Conclusion

En conclusion, ce projet de modélisation formelle de Bicloo a permis de définir un système à l'aide d'Atelier B. Les invariants et les opérations ont été conçus pour garantir la validité du système. Cependant, des améliorations pourraient être envisagées pour optimiser la gestion du système Bicloo notamment l'ajout d'opérations.