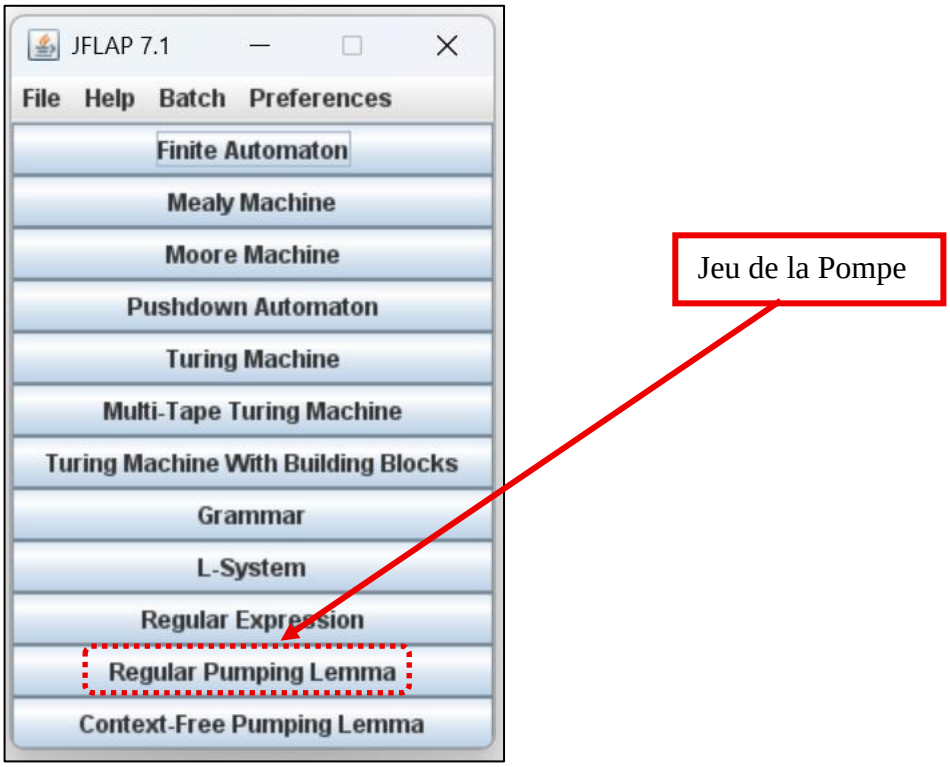
Les automates - TP 4
Lemme de la Pompe – Intersection de langages

Le but de ce TP est d'appliquer les méthodes vues en cours et de vérifier concrètement avec JFLAP que les automates obtenus répondent réellement au problème posé.

Exercice 1 : Lemme de la Pompe sous JFLAP

JFLAP propose un « jeu de la pompe ».

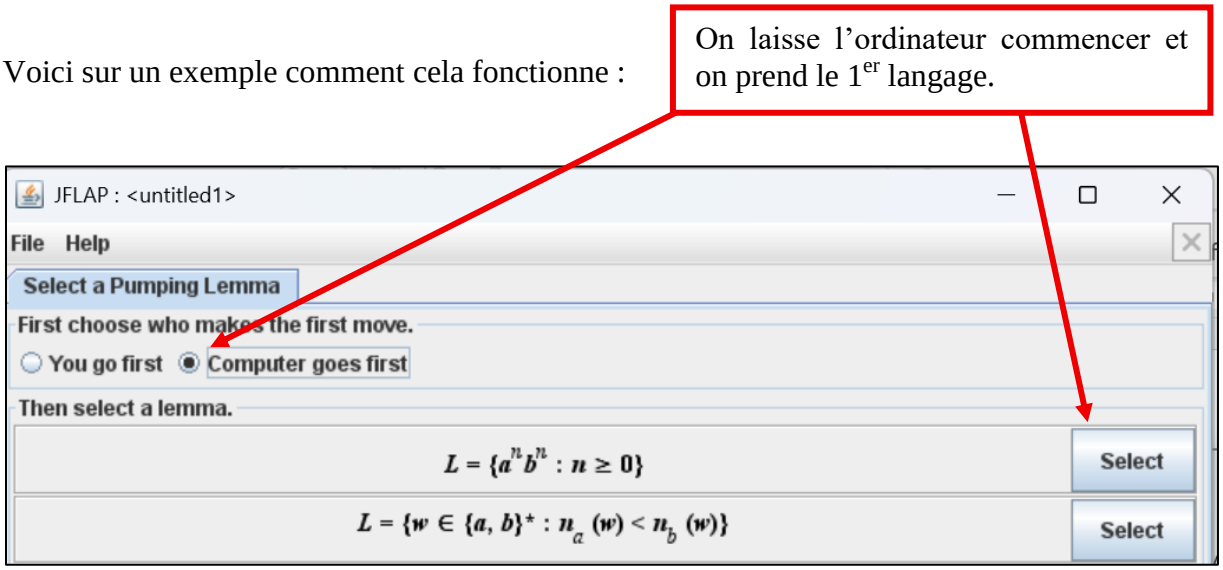


Principe du jeu :

- Etape 1 : Joueur1 propose un entier m.
- Etape 2 : Joueur2 propose un mot w de taille supérieure ou égale à m. $|w| \geq m$
- Etape 3 : Joueur1 propose une décomposition de w de la forme xyz avec $|y| > 0$ et $|xy| \leq m$.
- Etape 4 : Joueur2 propose une valeur de l'entier i telle que xy^iz n'appartient pas au langage.

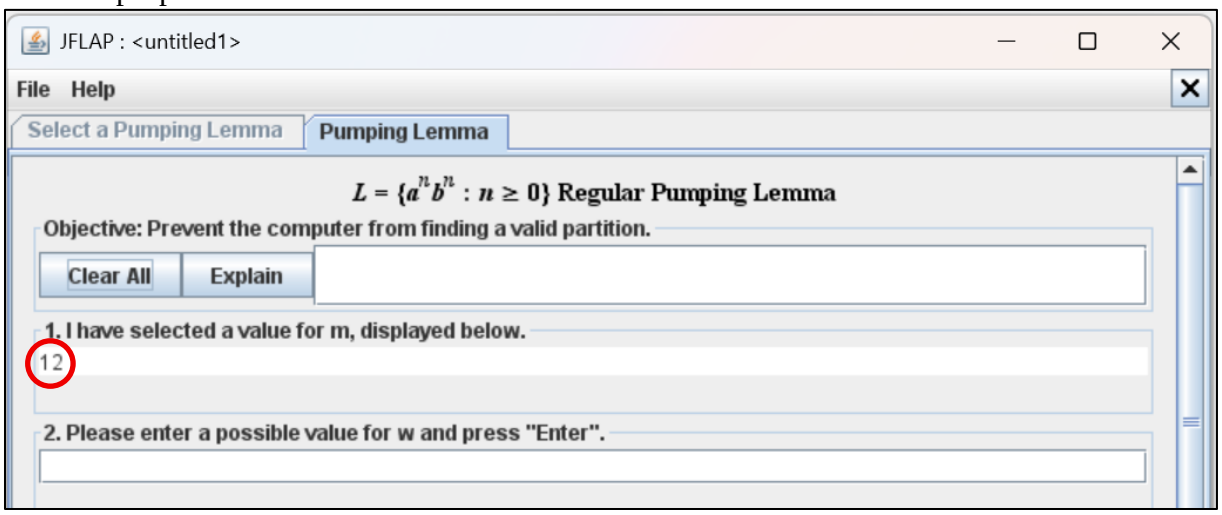
Si le Joueur2 arrive à faire l'étape 4, alors il a gagné. Sinon c'est le Joueur1 qui gagne.

Voici sur un exemple comment cela fonctionne :



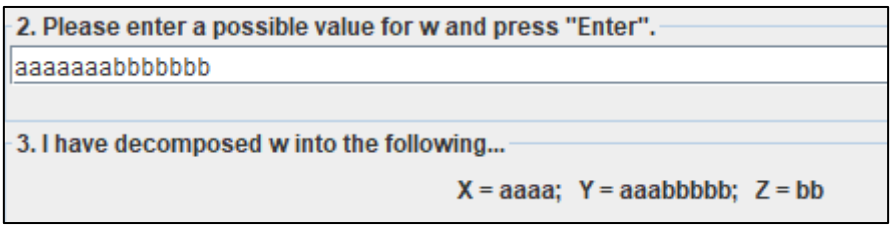
A partir de là, l'ordinateur va proposer une valeur pour *m*.

Ici il me propose 12 :

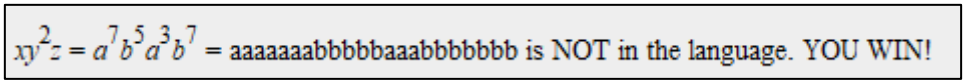


Je vais lui donner un mot du langage de taille au moins 12 et observer sa décomposition.
Je lui propose le mot : aaaaaaabbbbbbb (c'est-à-dire : $w = a^7b^7$ et $|w| = 14 \geq 12$)

Voici sa réponse :



Pour obtenir un mot XY^iZ qui n'est pas du langage il faut juste que je prenne une valeur de *i* différente de 1. Je peux prendre 0 ou 2. Voici ce qui se passe si je prends 2 :



1.a
En gardant le premier langage proposé dans le jeu, jouez plusieurs fois contre la machine en la laissant commencer et essayez de gagner à chaque fois. (*c'est possible*…)

1.b
Prenez maintenant le langage $L = \{a^n, n\ is\ even\}$
Vous savez que ce langage et régulier (vous savez même construire un automate qui le reconnait). Etant régulier, ce langage respecte le Lemme de la Pompe.
Ainsi à votre avis qui devrait gagner ? Celui qui commence ? L'autre ?
Faites des tests pour confirmer votre idée...

1.c
Même travail avec le langage $L = \{a^n b^k, n\ is\ odd\ or\ k\ is\ even\}$.

Exercice 2 : Intersection de langages

On se place dans sur l'alphabet $\Sigma = \{0, 1\}$ et on considère les mots w de Σ^* comme des entiers nbre(w) en base 2 autorisant les 0 à gauche.

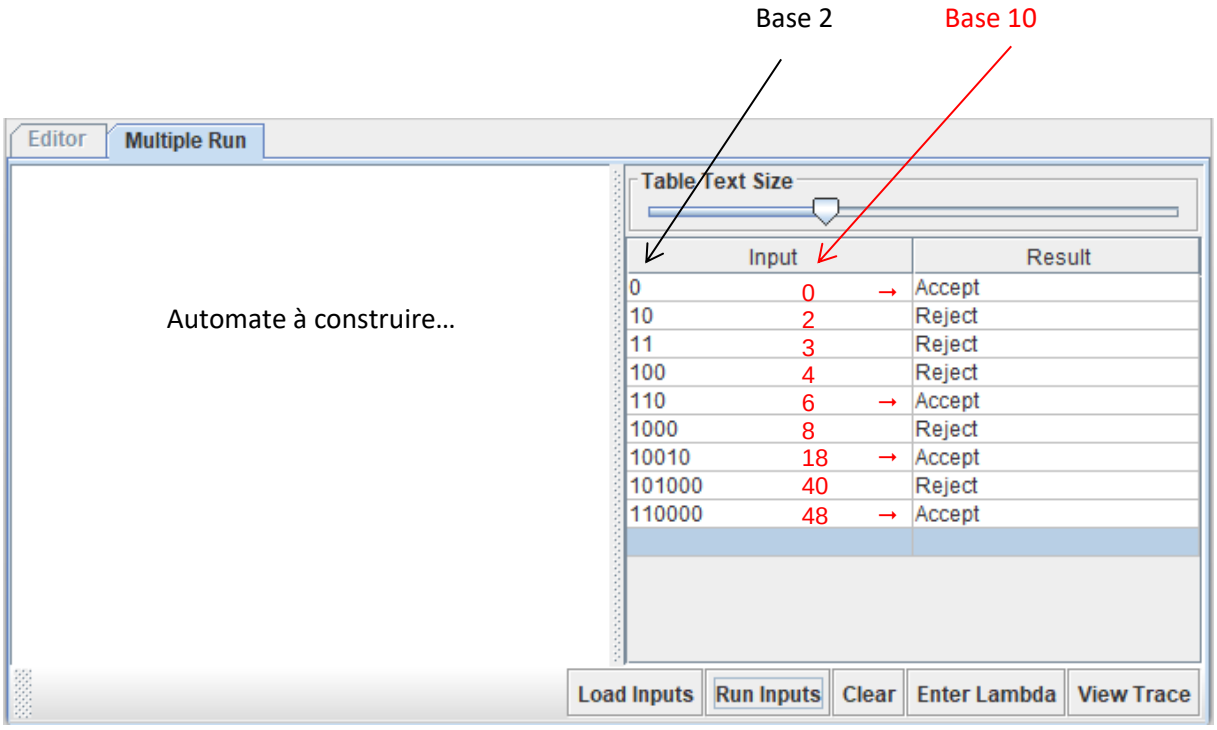
2.a
Soit $L_1 = \{ w \in \Sigma^* \mid \text{nbre}(w) \text{ est pair } \}$.
Tracer A_1 automate qui reconnait le langage L_1. (A_1 aura 2 états)

2.b
Soit $L_2 = \{ w \in \Sigma^* \mid \text{nbre}(w) \text{ est un multiple de 3 } \}$.
Tracer A_2 automate qui reconnait le langage L_2. (A_2 aura 3 états)

2.c
On considère le langage $L = L_1 \cap L_2$.

- Démontrer que L n'est pas vide.
- A l'aide de la méthode du cours, tracer la table de transitions de L.
- Construire alors l'automate qui reconnait L.

Pour finir, en utilisant le mode Multiple Run de JFLAP, vérifier que :



Remarque : L'automate qui reconnait L aurait pu être construit directement en considérant les restes de la division euclidienne d'un nombre par 6…