

0/5) Non-existent 1/5) Insufficient 2/5) Less than adequate 3/5) Adequate 4/5) Good 5/5) Excellent 6/5) Exceeds expectations		Total score on 20:	19.7
Partie	Weight	Score	Feedback (most important pointers)
Abstractions	3	4	<p>Petite remarque : Pourquoi les Mots et les Chemins sont covariant en B? I. e., quel est l'utilité de savoir faire le suivant :</p> <pre>scala> val x : Mot[Any] = List[Int](1,2,3,4) val x: List[Any] = List(1, 2, 3, 4)</pre> <p>Petite remarque 2 : Pourquoi des classes et non des case classes pour le AFD ? Le premier permet plus facilement de comparer l'équivalence des AFD.</p> <p>Non-pénalisé : Pourquoi certaines fonctions sont privées ? Je ne retrouve pas des justifications dans le rapport. J'ai explicitement dit que je n'avais pas évalué l'utilisation de l'accessibilité des membres (privée, package, publique, etc.) pour ce cours. Vous les avez parfois déclarés sans justification.</p> <p>Résumé : Remise en question de la covariance et manque de quelques justifications.</p> <p>Vous avez des files de tuples d'arité 4. Une synonyme de type pour ce tuple aurait rendu vos spécifications encore plus simples à lire !</p> <p>p.s. : il m'a fallu un peu de temps de réaliser que "une file de propriété" était "une file de priorité" :-)</p>
La fonction accept	3	5	Utilisation intelligente de l'option
La fonction solve	6	5	<p>Vous avez simulé des <i>priority queues</i> avec des listes, ce qui est parfait pour cet exercice, bien que les files d'attente prioritaires « réelles » s'appuient généralement sur d'autres structures de données (par exemple, des arbres) pour être plus efficaces.</p> <p>Il y a des indentations étranges dans votre code (non pénalisées car elles n'étaient pas trop "offensives" dans votre code).</p>
L'évaluation non stricte	4	5	Utilisation intelligente de LazyList et vous avez même utilisé certaines techniques en dehors de lazySolve.
Chaînes binaires impaires	2	5	
Problème du loup, le mouton et du chou	4	4	Redondance dans les fonctions estInterdit (moins grave) et deltaLoupMoutonChou. Dans le dernier, lestInterdit((gauche, droite)) et potentiellement évalué deux fois pour chaque symbole.
Le taquin	4	4	<p>Je n'ai pas pénalisé cela dans l'abstraction, car cela fonctionnait pour des puzzles simples. Le problème est de créer un objet Taquin puis de passer plusieurs de ses membres au constructeur AFD. Ne pourrait-on pas considérer Taquin comme une spécialisation d'AFD ?</p> <pre>val taquin2 = Taquin(parseTaquin("[[2 1][3 _]]").getOrElse(List(List(Piece.Trou))) val ex2 = new AFD(taquin2.sigma, delta, taquin2, Set(taquin2.etatFinal))</pre> <p>Utilité du lazy val etatFinal car normalement nécessaire pour un puzzle ?</p> <p>Dans la fonction move --> vous obtenez un Option avec findTrou. Puis, vous testez si la valeur existe (isDefined) et, si oui, vous faites quelque chose. Si non, vous renvoyez None. Vous auriez pu faire: findTrue.flatMap((trouRow, trouCol) => ...).flatten (le dernier pour éviter un Some (None)). Petite faute, mais quand même important car vous simulez des flatMaps.</p>
Spécifications	2	3	<p>Il y a quelques fautes de frappe dans les spécifications, mais qui suis-je pour juger les fautes de frappe si les spécifications sont claires et lisibles ? :-)</p> <p>Quelques spécifications incomplètes (e.g., l'emplacement des accumulateurs des fonctions auxiliaires).</p>
Technical report	2	4	Plus que correcte, mais il me manquait ne justification pour ADF.
Bonus :	3	5	<p>Vos implémentations de vos heuristiques sont très élégantes, même si elles ne suivent pas la signature d'une heuristique : A => Double. Au lieu de cela, vous faites référence à H1 et H2 depuis l'intérieur d'un lambda.</p> <p>H1 et H2 sont déclarés à l'intérieur de la classe case car ils en sont membres. Une autre approche consistait à les déclarer comme des fonctions à l'intérieur d'un objet compagnon afin qu'ils ne soient pas aussi étroitement couplés. Ceci est déjà pénalisé car c'est une conséquence de votre implémentation (cf. not Taquin).</p>
			Vous avez créé des petits parsers, mais ce n'était pas ce qu'on attendait de vous. Cela dit, j'apprécie l'exercice et vos efforts pour gérer les effets secondaires (exceptions).