Correction du TD 11

Julien Reichert

Induction

Exercice 1

Soit $(E, \leq)$ un ensemble bien ordonné, c'est-à-dire qui ne contient pas de suite infinie strictement décroissante pour l'ordre ici noté $\leq$ (avec $<$ pour la version stricte).

Soit P une partie non vide de E. Si P est finie, alors P admet un élément minimal, car on ne peut pas construire une suite de plus d'éléments que le cardinal que P d'éléments l'un strictement inférieur au suivant sans répéter un élément, ce qui voudrait dire que par transitivité il existe x dans E tel que $x < x$, absurde.
Si P est infinie [le raisonnement reste valable pour P finie non vide], comme il n'existe pas de suite infinie strictement décroissante, on considère (u_n) qui est une suite d'éléments de P telle que u_0 soit quelconque et $u_{n+1} < u_n$ pour tout n. Cette suite ne pouvant pas être infinie, il existe un $M \in \mathbb{N}$ tel que u_{M+1} n'ait pas pu être trouvé, car aucun élément de P n'est strictement inférieur à u_M. Dans ce cas, u_M est un élément minimal de P.

Cette preuve est plus facile à rédiger en raisonnant par contraposée, par ailleurs.

Soit $(E, \leq)$ un ensemble ordonné dont toute partie non vide admet un élément minimal. Considérons une suite infinie d'éléments de E. L'ensemble des valeurs prises est une partie non vide de E. Elle admet donc un élément minimal. Une fois celui-ci atteint, l'élément suivant ne peut pas être strictement inférieur, et la suite elle-même ne peut pas être strictement décroissante.

Exercice 2

Montrons que l'ordre produit de deux ordres bien fondés est bien fondé.

Supposons par l'absurde qu'il existe une suite infinie strictement décroissante pour l'ordre produit. De par la construction de l'ordre produit, on en déduit deux suites décroissantes pour l'ordre respectif avec à chaque étape au moins une des décroissances qui est stricte. Sur une infinité d'étapes, au moins une des deux suites a sa décroissance qui est stricte une infinité de fois (là aussi par l'absurde s'il s'agit de détailler), et donc un des deux ordres ne pouvait pas être bien fondé.

Montrons que l'ordre lexicographique sur un produit cartésien de deux ensembles bien fondé est bien fondé.

Supposons par l'absurde qu'il exisrte une suite infinie strictement décroissante pour l'ordre lexicographique. Comme la preuve se complique, des notations seront nécessaires : soit $(w_n) = (u_n, v_n)$ la suite en question. Par définition, on a (u_n) décroissante (pas forcément strictement) et (v_n) strictement décroissante au moins sur tous les intervalles d'indices où (u_n) ne décroît pas strictement. Alors chacun des intervalles d'indices où (u_n) ne décroît pas strictement doit être fini (sinon une suite extraite de v_n serait infinie et strictement décroissante), et l'indice de début de chacun de ces intervalles, le nombre de ces intervalles étant infini (le retour de la preuve par l'absurde qu'on ne détaillera pas), permet de définir une suite extraite de (u_n) qui sera infinie et strictement décroissante, ce qui est absurde là encore.

Pour illustrer la preuve, on imagine l'ordre lexicographique sur $\mathbb{N}^2$. Si $w_0 = (u_0, v_0)$, le nombre de fois où la deuxième composante peut décroître strictement en laissant u_0 constant est au plus v_0. Ensuite, il faudra avoir un (u_k, v_k) avec $u_k < u_0$ et v_k quelconque, mais bien fixé et « définitif », ce qui ne laisse que v_k occasions pour maintenir u_k avant une nouvelle décroissance stricte, et il ne peut plus y avoir plus de u_k décroissances strictes de la première composante par ailleurs.

Une preuve alternative utilise l'exercice 1...

Exercice 3

Remarque : Le produit cartésien d'aucun ensemble est un ensemble de taille 1 n'ayant aucune composante. En quelque sorte c'est un zéro-uplet. Dans ce cas tout ordre sur cet ensemble est bien fondé...

Pour $n = 1$, le produit cartésien d'un seul ensemble est l'ensemble lui-même, l'ordre produit est l'ordre sur l'ensemble et l'ordre lexicographique aussi. On a directement l'initialisation puisque cela fait partie de l'hypothèse. Pour $n = 2$, c'est l'exercice précédent, par ailleurs.

Pour passer de n à $n+1$, on considère le produit cartésien $E_1 \times \cdots \times E_{n+1}$, qu'on parenthèse (en profitant de l'associativité du produit cartésien) pour avoir $\tilde{E} \times E_{n+1}$ en notant $\tilde{E} = E_1 \times \cdots \times E_n$. Par hypothèse de récurrence, l'ordre produit et l'ordre lexicographique sur E sont bien fondés et d'après l'exercice précédent puisque c'est vrai pour E et pour E_{n+1} c'est vrai pour le produit cartésien, ce qui conclut la preuve.

Exercice 4

On a déjà vu dans le cours qu'il s'agissait d'un ordre. Soit une partie non vide P d'un ensemble inductif muni de l'ordre structurel. Elle contient au moins un élément x dont le nombre de constructeurs récursifs utilisés est noté c_x. Alors l'ensemble des nombres de constructeurs récursifs utilisés sur tous les éléments de P est une partie non vide de $\mathbb{N}$ (contenant c_x) et elle admet pour cette raison un minimum. Soit y l'élément de P utilisant ce nombre minimum de constructeurs. Alors s'il existait un élément z de P qui soit strictement inférieur à y selon l'ordre structurel, on pourrait construire y à partir de z en utilisant un constructeur récursif (c'est la définition), et z nécessiterait strictement moins de constructeurs récursifs que y, ce qui est absurde.

Exercice 5

Posons le contexte : Soit E un ensemble inductif défini par E_0 et un ensemble de règles C_1, ..., C_n. Soit une propriété $\mathcal{P}$ vraie sur E_0 et demeurant vraie sur les éléments produits par une règle C_i à partir d'arguments dans E pour lesquels la propriété est supposée vraie.

Montrons que la propriété $\mathcal{P}$ est vraie sur tous les éléments de E.

Une preuve pratique mais peu intuitive est de dire que l'ensemble $\tilde{E}$ des éléments de E qui vérifient la propriété $\mathcal{P}$ contient E_0 et reste stable par l'application des règles, et puisque E est le plus petit ensemble à avoir cette propriété, alors $E \subseteq \tilde{E}$, avec l'inclusion réciproque qui est évidente.

Autrement, on peut toujours utiliser l'ordre structurel et considérer par l'absurde un élément minimal (l'ordre étant bien fondé et un sous-ensemble non-vide des éléments ne vérifiant pas $\mathcal{P}$ étant considéré ici) de E qui ne vérifie pas $\mathcal{P}$. Cet élément contredit les hypothèses sur $\mathcal{P}$.

Exercice 6

Exercice laissé au lecteur. Pour le cinquième, on notera que reconnaître les nombres de Catalan n'est pas nécessaire ici, tant qu'on identifie la bonne relation de récurrence faisant intervenir tous les découpages de $n-1$ comme somme de deux entiers naturels.

Arbres

Exercice 7

Soit un arbre binaire dont la différence de hauteur de deux frères (définie comme la hauteur du sous-arbre dont ils sont la racine respective) est d'au plus un à tout endroit. Montrons que cet arbre est équilibré dans la mesure où tout arbre non vide vérifiant cette propriété est de hauteur inférieure à une constante fois le logarithme de la taille.

Il s'agit de considérer la taille minimale d'un tel arbre à hauteur fixée. Pour une hauteur de -1, la taille est 0 et pour une hauteur de 0, la taille est 1. Pour une hauteur supérieure, la taille minimale d'un arbre de hauteur n vérifiant cette propriété est au moins un de plus (pour la racine) que la taille minimale d'un arbre de hauteur $n-1$ vérifiant cette propriété (le fils le plus haut de la racine) plus la taille minimale d'un arbre de hauteur $n-1$ ou $n-2$ (donc $n-2$ car on a clairement intérêt à minimiser la hauteur pour minimiser la taille) vérifiant cette propriété.

La taille minimale vérifie donc $c_{-1} = 0, c_0 = 1$, $c_1 = 2$ et $c_{n+2} = c_{n+1} + c_n + 1$, soit $(c_{n+2} + 1) = (c_{n+1} + 1) + (c_n + 1)$ avec $(c_0 + 1) = 2$ et $(c_1 + 1) = 3$, donc $(c_n + 1)$ est la suite de Fibonacci décalée de trois crans. Ainsi, la taille minimale est de l'ordre d'une constante fois le nombre d'or à la puissance de la hauteur, et la taille tout court est en particulier supérieure à cela. En bricolant l'inégalité ainsi obtenue, la hauteur est alors inférieure à une constante fois le logarithme (dont la base n'importe désormais plus) de la taille.

Exercice 8

Soit un arbre bicolore de taille notée n, de hauteur notée h et de hauteur noire notée k. Montrons que $h \leq 2k$ et $n \geq 2^k - 1$ par induction structurelle sur l'arbre.

Le cas de base de l'arbre vide est correct, car sa hauteur est -1, sa hauteur noire est nulle (on ne compte pas les vides) et sa taille aussi, on a bien $-1 \leq 0$ et $0 \geq 2^0 - 1$.

Le cas de l'arbre réduit à sa racine est également correct, avec la hauteur noire k qui est de 0 ou 1, forcément supérieure au double de la hauteur (0) et la taille de (1) reste supérieure ou égale à $2^k - 1$.

Soient g et d les sous-arbres gauche et droit de l'arbre. Si la racine de l'arbre est rouge, soit g et d sont vides (cas précédent) soit les deux sont non vides (un vide impose qu'aucun nœud de l'autre ne soit noir, et comme la racine de l'autre ne peut pas être rouge il y a une contradiction) et de racine noire. Dans ce dernier cas, on utilise d'une part les hypothèses d'induction sur chaque sous-arbre enraciné en les enfants de g et de d. Leur hauteur noire est un de moins que k, leur hauteur est au plus deux de moins que h d'où la première propriété. Pour la deuxième propriété, la taille de l'arbre est un de plus que la taille de g plus la taille de d, chacune supérieures à $2^k - 1$ puisque leur hauteur noire est la même, ce qui permet de conclure en simplifiant.

Si la racine de l'arbre est noire, on considère seulement g et d sans tenir compte de leurs enfants. La hauteur de l'arbre est un de plus que la hauteur de g et d, qui sont de hauteur noire $k - 1$ et qui vérifient l'hypothèse d'induction, d'où la première propriété pour l'arbre entier. Concernant la taille, on a un de plus que quelque chose qui est supérieur ou égal à $2^{k-1} - 1$ pour la gauche plus quelque chose qui est supérieur ou égal à $2^{k-1} - 1$ pour la droite, ce qui permet une fois de plus de conclure en simplifiant.

Exercice 9

Gérons d'abord la propriété d'ABR, sans rédiger d'induction.

La fonction d'insertion détermine quel arbre vide remplacer par la nouvelle valeur en partant à gauche ou à droite au niveau de chaque nœud en fonction de la comparaison avec son étiquette. La propriété d'ABR est maintenue par la création du nouveau nœud (à ce stade, on supppose que la correction de l'insertion dans un ABR a été prouvée, mais ce n'est pas la partie difficile que ce corrigé est amené à traiter). Par la suite, seules des rotations et des changements de couleur d'un nœud sont effectuées. Or, les rotations préservent la propriété d'ABR (là aussi, cela a déjà été signalé et on peut le prouver en reprenant le schéma des rotations), et les changements de couleur n'ont pas d'influence sur les positions, ce qui termine la preuve.

Au niveau de la fonction de suppression, on retrouve la localisation du nœud à supprimer conformément avec la propriété d'ABR, en particulier la non découverte de l'étiquette recherchée implique son absence de l'arbre. Après la suppression, si tant est qu'elle ait lieu, ce sont de nouveau des rotations et des changements de couleur qui sont effectués, d'où là aussi la préservation de la propriété d'ABR.

Concentrons-nous maintenant sur les deux règles principales sur la couleur (pas de père rouge d'un nœud rouge, et hauteur noire constante), en rappelant qu'à la fin des opérations on remet la racine en noir si nécessaire.

Appelons « problème 1 » le fait qu'un nœud rouge ait un fils rouge (éventuellement précisé en « problème 1 à gauche » ou « problème 1 à droite ») et « problème 2 » le fait qu'un nœud soit la racine d'un sous-arbre de hauteur noire inférieure d'une unité à la hauteur noire attendue.

Cas de l'insertion

On met un nœud rouge à la place d'un arbre vide. Notons n ce nœud et p son père (s'il existe). Si p est noir ou n'existe pas, tout est en ordre. Si p est rouge, on a un problème 1 en p et la « boucle de correction » (qui est une fonction lancée à chaque appel récursif de l'insertion dans la correction du TP 10) s'exécute sur p.

On rappelle que puisque p est rouge, il ne s'agit pas de la racine, et on peut noter g son père, forcément noir.

Reprenons la distinction de cas effectuée par la correction :

— *« Si g a deux fils tous deux rouges, on met g en rouge et ses deux fils en noir, puis on considère g en vue de l'éventuel tour suivant de la boucle de correction. Sinon, on passe aux points suivants. »* Ici, le nombre de nœuds noirs de chaque branche est resté le même, et le problème 1 au niveau de p a été résolu. Si g a un père rouge, cela crée un problème 1 au niveau de celui-ci, et l'appel à la correction à son niveau le résoudra (ceci peut constituer une hypothèse de récurrence sur la profondeur du nœud où l'appel récursif a lieu dans le programme écrit).

— *« Si n est le fils gauche de p alors que p est le fils droit de g, on commence par une rotation droite sur le sous-arbre enraciné en p. »* Ici, le problème 1 à gauche de p devient le problème 1 à droite de n, qui devient fils droit de g et père de p si on considère les étiquettes. La boucle de correction peut donc changer de côté un problème 1 car on a la version symétrique en échangeant toutes les occurrences de « droite » et « gauche » dans la citation en italique (point qui a été retiré pr la suite).

— *« Par la suite, si p (ou n après rotation) est le fils gauche de g, on fait une rotation droite sur le sous-arbre enraciné en g, sinon une rotation gauche, et dans tous les cas on met p (ou n après rotation) en noir et g en rouge et la boucle de correction est terminée. »* Ici, on a cité le TP 10 tel quel, mais sans perte de généralité on suppose que p est fils gauche de g et a un problème 1 à gauche. Après la rotation droite, g est remplacé par p (mis en noir) et devient son fils droit (mis en rouge), n (resté rouge) devient fils gauche de p, le problème 1 de p est résolu, au vu des cas précédents n et g n'ont pas de problème 1 malgré leur couleur, et la hauteur noire n'est pas modifiée par l'opération, ce qui confirme que tous les problèmes sont résolus, de sorte que les appels à la fonction de correction qui seront effectués plus haut n'aient pas de travail à faire.

Cas de la suppression

Plusieurs scénarios ici :

- Le nœud qu'on retire est rouge sans enfants : aucun problème.
- Le nœud qu'on retire est noir sans enfants : problème 2 sur ce nœud.
- Le nœud qu'on retire n'a qu'un enfant (forcément rouge dans ce cas, donc le nœud retiré est noir) : le nœud est remplacé par son fils qui est mis en noir, aucun problème.
- Le nœud qu'on retire a deux enfants : on remplace ce nœud, sans changer sa couleur, par le nœud d'étiquette minimale à droite obtenu en allant à gauche jusqu'à ce que cela ne soit plus possible (c'est ce qui est fait dans la correction du TP 10, mais on aurait pu choisir le maximum à gauche), nœud qui est alors retiré : on étudie alors le cas de ce nœud, qui est parmi les trois premiers car il n'a pas d'enfant à gauche.

Reprenons la distinction de cas de l'énoncé du TP 10 pour la correction du problème 2.

Ici, le nœud noté n ne sera pas toujours le nœud retiré mais le nœud qui est racine du sous-arbre où se situe le problème 2. Bien entendu, la première fois qu'on traitera le problème 2, n sera effectivement le nœud retiré.

On rappelle que si n est la racine, c'est que la suppression aura baissé la hauteur noire de l'arbre d'une unité (le sous-arbre ayant une hauteur noire trop faible est l'arbre entier, ce qui n'est plus un problème en soir).

D'un autre côté, si n est rouge, on le met en noir et le problème 2 est réglé. On suppose donc que n est déjà noir, et par ailleurs le problème 2 en n est le seul problème de l'arbre dans son intégralité (on observe que le problème est issu de la suppression, qu'il ne se pose qu'à un endroit et que les cas à suivre le déplacent sans en ajouter).

On note de nouveau p le père de n, et on note f l'autre fils de p, qui existe forcément puisque n était noir. Comme dans l'énoncé du TP 10, on suppose sans perte de généralité que n est le fils gauche de p, quitte à changer toutes les occurrences de « droit(e) » par « gauche » et vice-versa.

Il faut également garder à l'esprit que la suppression a déjà eu lieu, donc le nœud n initial peut être en pratique un arbre vide, sans préjudice au niveau de l'algorithme car les rotations ne nécessitent pas la non-vacuité du sous-arbre qui serait enraciné en n... si n était un nœud. On fera dans la suite l'assimilation de n à un nœud par abus, même si ce n'en est pas un.

— *« Si f est rouge, ce qui garantit que ses enfants éventuels sont tous noirs et que p l'est aussi, alors on fait une rotation gauche ou droite sur le sous-arbre enraciné en p suivant le fait que f soit le fils droit ou gauche de p, respectivement, et on échange les couleurs de f et p. On passe alors aux sous-cas suivants en adaptant les notations à la nouvelle situation. »* Ici, la profondeur de n a augmenté d'un mais par le nombre de nœuds noirs de la racine à n. En parallèle, le nouveau frère de n est noir pour passer à un cas suivant, et même si un des fils de f a vu sa profondeur diminuer d'un, la hauteur noire n'a pas été impactée.

— *« Si f est noir avec deux fils noirs, on met f en rouge et on déplace le problème à p dans un nouveau tour de boucle. »* Ici, le problème 2 est dupliqué entre n et f, donc en pratique il se retrouve au niveau de p. En raison de la couleur des enfants de f, on ne crée pas de problème 1, et en pratique si p était rouge, le passer en noir résoudrait immédiatement le problème 2 (sinon on ne créerait pas de problème 1 au niveau de p pour autant).

— *« Sinon, si f est noir avec son fils droit noir, on met f en rouge, son fils gauche en noir, on fait une rotation droite dans le sous-arbre enraciné en f et on passe à l'étape suivante avec les notations adaptées. »* Ici, suite aux deux opérations consécutives, la situation finale est qu'aucun problème 1 n'est créé (le fils gauche de f était rouge donc s'il avait des fils, ceux-ci étaient noirs, et le fils droit d'avant devient le fils gauche de f qui est entre temps devenu rouge, de plus l'ancien fils gauche de f est devenu noir), le problème 2 demeure en n uniquement (la hauteur noire reste la même au niveau de tous les sous-arbres concernés par la rotation) et le fils droit du nouveau frère de n est désormais rouge, justifiant que le cas suivant s'applique.

— *« Sinon, f est noir avec son fils droit rouge (peu importe son fils gauche), on fait alors une rotation gauche sur le sous-arbre enraciné en p. On met f à la couleur de p, on met p et le fils droit de f en noir, et le problème est résolu, ce qui permet là aussi de terminer la suppression. »* Ici, après la rotation et les changements de couleur, la profondeur de n a augmenté d'un, avec un père noir qui s'est inséré. Les deux autres sous-arbres concernés par la rotation (en plus de celui dont n est la racine) sont celui dont la racine était le fils gauche de f, et désormais le frère de n, et dont la hauteur noire est restée la même, et celui dont la racine était le fils droit de f, désormais toujours fils droit de f mais d'une part devenu noir et d'autre part de profondeur ayant diminué d'un, avec un ancêtre noir en moins, ce qui maintient la hauteur noire des branches passant par ce sous-arbre à la bonne valeur. Le problème 2 en n est donc résolu sans que d'autres ne se créent.