Dans ce livre, nous avons exploré un certain nombre de liens entre les notions de démonstration et d'algorithme, à travers le théorème d'indécidabilité de la démontrabilité en logique des prédicats d'abord, puis à travers le résultat de décidabilité de la bonne formation d'une démonstration, qui a mené à un résultat de semi-décidabilité de la démontrabilité en logique des prédicats, à des algorithmes de vérification de démonstrations et de démonstration automatique et à travers des résultats de décidabilité pour des théories particulières. Enfin, la notion de constructivité a mis en évidence un autre lien entre les démonstrations et les algorithmes, qui a mené à une méthode, parmi d'autres, pour démontrer qu'un algorithme vérifie une spécification.

En chemin, nous avons découvert les quatre grandes notions autour desquelles la logique contemporaine est organisée : les notions de démonstration, d'algorithme, de modèle et d'ensemble. Ces quatre notions définissent les quatre branches de la logique : la théorie de la démonstration, la théorie de la calculabilité, la théorie des modèles et la théorie des ensembles. Cette classification, pour utile qu'elle soit, ne doit cependant pas occulter le fait que toutes ces notions sont utilisées, à des degrés divers, dans chacune de ces branches.

Jusqu'à la fin du XIX$^{\text{e}}$ siècle, il n'existait qu'une notion rudimentaire de démonstration, remontant à l'Antiquité, des notions informelles d'ensemble et d'algorithme et pas de notion de modèle. La logique s'est donc entièrement renouvelée quand ces quatre notions ont été dégagées entre les années soixante-dix du XIX$^{\text{e}}$ siècle et les années trente du XX$^{\text{e}}$ siècle.

Nous avons également abordé un certain nombre d'applications de la logique : en mathématiques d'abord, avec des résultats d'indépendance et de cohérence relative, mais aussi, de manière peut-être plus inattendue, avec des résultats d'algèbre, pour lesquels il n'était pas *a priori* évident que des outils logiques soient nécessaires.

Mais c'est surtout en informatique que la logique a trouvé un vaste champ d'applications : en théorie des langages de programmation, avec la conception de langages fonctionnels issus du lambda-calcul et de langages logiques issus d'algorithmes de démonstration automatique, en architecture où certains circuits sont représentés par des propositions de la logique propositionnelle, en théorie de la complexité, avec, par exemple, la notion de machine de Turing non déterministe, en théorie des bases de données, avec la notion de langage de requête issue de la théorie des modèles finis et en sûreté, avec la conception d'outils permettant de démontrer la correction de circuits et de programmes par rapport à leur spécification logique.

Le rôle central que joue la notion d'algorithme en logique pouvait certes laisser espérer certaines applications en informatique, mais sans doute pas au point que nous connaissons aujourd'hui. La logique semble, par certains aspects, être à l'informatique ce que le calcul différentiel est à la physique.

Et il n'est pas certain que nous ayons aujourd'hui complètement compris les raisons de cette déraisonnable efficacité de la logique en informatique.