http://binaire.blog.lemonde.fr/2015/03/09/et-un-et-deux-et-trois-femmes-prix-turing/

09 mars 2015

Et un, et deux, et trois femmes Prix Turing !

Après Ada Lovelace et Grace Hopper, et à l'occasion de la Journée internationale de la femme, Anne-Marie Kermarrec nous parle de plusieurs grandes informaticiennes et scientifiques, toutes Prix Turing. Elle achève ainsi sa démonstration – s'il était possible de douter – l'informatique est aussi pour les filles ! Serge Abiteboul.

En 1966, l'ACM crée le prix Turing, l'équivalent du Nobel pour l'informatique, qui récompense les plus grands scientifiques du domaine. Il faudra attendre quarante ans pour voir entrer une femme au palmarès. Depuis, deux autres femmes ont été récompensées par le prestigieux trophée. Ce n'est pas si mal, quand la médaille Fields a récompensé une femme pour la première fois en 2014 !!

Frances Allen Source Wikipedia

2006 : Frances Allen, née en 1932, après voir été la première femme à recevoir le titre d'IBM fellow, est la première à se voir récompenser par le prix Turing pour ses contributions pionnières tant pratiques que théoriques dans l'optimisation des compilateurs. Lors de ses études de mathématiques à l'Université du Michigan, Ann Arbor, elle y prend aussi des cours d'informatique, parmi les premiers dispensés. Elle est engagée par IBM avec l'envie de revenir à ses premières amours et de revenir enseigner les mathématiques quand son prêt étudiant serait soldé. Elle restera 45 ans chez IBM. Sa passion pour la compilation lui vient de la lecture attentive du compilateur Fortran en 1957 quand d'autres lisent des romans ! En bref, un compilateur traduit un langage de programmation de haut niveau, comme le langage Cobol dont Grace Hopper est à l'origine rappelez-vous, un langage adapté à des humains, en instructions qu'un ordinateur peut exécuter. Un compilateur est donc par définition dépendant d'un langage de programmation et d'une architecture machine. Avec son équipe, elle conçoit le premier environnement de compilation multi-langages (Fortran, Autocoder qui est un langage proche de Cobol de Grace Hopper et Alpha). Les trois langages partageaient en particulier un socle

d’optimisation qui permettait de produire du code pour les deux architectures du supercalculateur Stretch et de son co-processeur Harvest. Elle travailla ensuite à la conception du premier ordinateur superscalaire (ACS) capable d’exécuter plusieurs instructions simultanément, y compris dans le « désordre ». Il va de soi qu’écrire des compilateurs associés à ce nouveau type d’architecture représentait un incroyable défi, qu’elle a su relever en représentant le code source comme un graphe plutôt que comme une séquence d’instructions. Cette représentation a permis en particulier de pouvoir détecter des relations entre différentes parties du code difficiles à détecter autrement. Son dernier projet a consisté à compiler des programmes séquentiels pour des architectures parallèles.

L’une des grandes vertus scientifiques de Frances Allen, a été à l’instar de Grace Hopper, non pas de réinventer des nouveaux paradigmes en langage de programmation mais de concevoir des mécanismes nouveaux d’analyse et d’optimisation permettant de traiter les langages tels qu’ils étaient utilisés en pratique.

Barbara Liskov
Source Wikipedia

2008 : Barbara Liskov reçoit le prix Turing pour ses travaux dans le domaine des langages de programmation et de la méthodologie polymorphe. Barbara Liskov, née en 1939, fait ses études à Berkeley, passe un doctorat à Stanford avant de rejoindre Mitre Corp où elle crée le système d’exploitation pour l’ordinateur Venus, un système d’exploitation qui permettait d’isoler, en utilisant la notion de machine virtuelle (ça vous rappelle quelque chose ?) pour isoler les actions, et donc potentiellement les erreurs, d’un utilisateur sur une machine partagée entre plusieurs utilisateurs : les débuts du temps partagé. Elle devient professeur au prestigieux MIT en 1971. Elle y conçoit un langage de programmation, appelé CLU, qui introduit les notions de modularité, d’abstractions de données et de polymorphisme (ce qui permet d’utiliser le même code pour des types d’objets différents), notions fondatrice des langages orienté-objet dont le plus connu est le plébiscité Java. Le langage Argus, sur lequel elle travaille plus tard, étend ces concepts pour faciliter la

programmation au dessus d'un réseau. C'est d'ailleurs dans le domaine des systèmes distribués, quand plusieurs machines connectées par un réseau exécutent ensemble une application, qu'elle continuera son illustre carrière. Elle est encore extrêmement active aujourd'hui et les travaux actuels du domaine reposent sur bien des concepts qu'elle a introduit en terme de réplication, tolérance aux défaillances, etc. Elles s'est en particulier attaquée à l'algorithmique Byzantine, qui consiste à tolérer la présence d'entités malicieuses (attaques ou fautes matérielles ou logicielles aléatoires) dans un système.

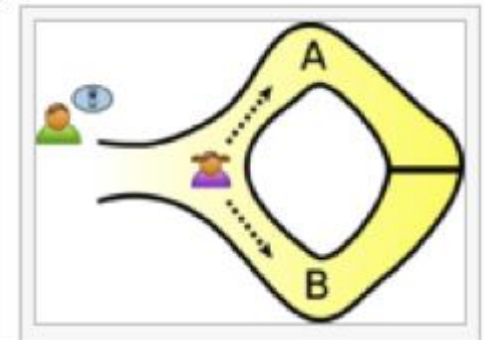
L'avantage de mettre autant de temps à récompenser les femmes dans cette discipline jeune est qu'elles sont toujours actives ! J'ai eu la chance de rencontrer Barbara Liskov, une grande dame de ma discipline, que nous admirons tous beaucoup et qui est en particulier une fervente défenseure de la cause féminine. Elle a beaucoup contribué à renforcer la présence des femmes professeurs au MIT, et met beaucoup d'énergie pour animer la communauté des femmes en système en particulier.

Shafi Goldwasser
Source Wikipedia

2012 : Shafi Goldwasser reçoit, avec Silvio Micali, le prix Turing pour ses travaux dans le domaine de la cryptographie et de la preuve informatique. C'est un peu comme si Babbage avait partagé son prix de la Royal Academy of Astronomy avec Ada… Shafi Goldwasser est née seulement en 1958 et son nom est déjà célèbre dans le domaine de la cryptographie. Cette volontaire et énergique Professeure au MIT est connue en particulier pour ses contributions pionnières dans le domaine de la cryptographie et des « preuves interactives connaissance-zéro ».

Durant ses études à Carnegie Mellon University, Shafi effectue un stage à RAND Corporation qui lui fait découvrir la Californie et surtout Berkeley où elle commence un doctorat sous l'égide du très connu Dave Patterson. Elle rencontre son brillant collaborateur Silvio Micali et commence à s'intéresser à la cryptographie. La cryptographie est un cauchemar à expliquer. Pour simplifier disons que l'une des contributions majeures de Shafi a été cette « preuve interactive connaissance-zéro », qui désigne une méthode dans laquelle une entité prouve à une autre entité

qu'une proposition est vraie mais ne donne aucun autre élément que la véracité de la proposition. La dernière fois que l'on m'a expliqué ce concept, c'était justement Shafi Goldwasser, qui nous avait fait le plaisir d'honorer de sa présence un évènement scientifique que nous organisons pour les étudiants. Une célébrité très accessible.



Preuve interactive connaissance-zéro, Wikipedia

Le principe de la « preuve interactive connaissance-zéro » est souvent expliquée de la manière suivante (source wikipedia). Imaginons Peggy (en rose sur l'image, une fois n'est pas coutume) et Victor (en vert), Victor souhaite savoir si Peggy connaît le code d'un passage secret entre une allée A et une allée B d'une cave. L'objectif de Peggy est de lui montrer qu'elle connaît le code sans le divulguer. Peggy entre dans la cave sans que Victor ne sache par quelle allée elle est entrée. Victor lui demande de sortir par l'une des allées, A ou B. Si Peggy connaît le code et que Victor lui demande de sortir par l'allée A, peu importe l'allée par laquelle elle est entrée, elle sortira par A (en ouvrant le passage secret si elle est entrée par B). Sinon elle a seulement une chance sur deux de sortir par l'allée demandée. En répétant cette opération (interactive) plusieurs fois, la probabilité que Peggy sorte par l'allée demandée devient très petite si elle ne connaît pas le code. Ainsi ceci fournit un moyen de vérifier que Peggy connaît le code (preuve) sans que Peggy ait à divulguer d'information (connaissance-zéro). Expliquer cet exemple est déjà un défi, quand à le prouver, cela vaut bien un Turing Award !

À quand la super production Hollywoodienne qui nous portera tout ça à l'écran ?

Anne-Marie Kermarrec, Inria Bretagne