COURS

Calculabilité, décidabilité et programme en tant que donnée

Tle

Programme de terminale générale, enseignement de spécialité — BO spécial n°8 du 25 juillet 2019

Relue le 14 novembre 2026

Un ordinateur peut-il tout calculer ? Peut-on écrire un programme qui détecte si n'importe quel autre programme va boucler à l'infini ? Alan Turing a montré en 1936 que la réponse est non : il existe des problèmes qu'aucun algorithme ne peut résoudre.

1 Un programme est une donnée

RÈGLE

Un programme est un texte, donc une donnée qu'un autre programme peut lire, analyser ou exécuter : c'est ce que font un **interpréteur** (Python), un **compilateur**, un antivirus ou un système d'exploitation qui lance une application.

2 Calculabilité

DÉFINITION

Fonction qu'un algorithme peut calculer en un nombre fini d'étapes, quelle que soit l'entrée.

RÈGLE

Ce qui est calculable ne dépend pas du langage de programmation : Python, C ou une machine de Turing calculent les mêmes fonctions (thèse de Church-Turing).

3 Le problème de l'arrêt

DÉFINITION

Déterminer, pour tout programme P et toute entrée e, si P s'arrête sur e. Turing a démontré en 1936 qu'aucun programme ne peut le résoudre : il est **indécidable**.

RÈGLE

Idée de la preuve : si un programme `arret(P, e)` existait, on construirait un programme `paradoxe(P)` qui boucle quand `arret(P, P)` répond vrai et s'arrête sinon ; `paradoxe(paradoxe)` mène alors à une contradiction.

EXEMPLE

Pourquoi un antivirus ne peut-il pas détecter à coup sûr tout comportement malveillant ?

1. Prévoir le comportement d'un programme quelconque revient à des problèmes comme celui de l'arrêt.
2. Ces problèmes sont indécidables en général.
3. L'antivirus utilise donc des signatures et des heuristiques, sans garantie absolue.

L'indécidabilité impose des limites théoriques aux outils qui analysent des programmes.

LE PIÈGE QUI COÛTE LE PLUS DE POINTS

Croire qu'indécidable signifie très long à calculer. — Un problème indécidable n'a aucun algorithme qui le résout dans tous les cas, même avec un temps illimité.

Le réflexe : Distinguer « difficile » et « impossible ».

RAISONNER PAR L'ABSURDE SUR L'ARRÊT

1. Supposer qu'un programme arret existe.
2. Construire paradoxe qui fait l'inverse de ce que prédit arret.
3. Appliquer paradoxe à lui-même.
4. Conclure à la contradiction : arret n'existe pas.

À VÉRIFIER

- Ai-je défini le vocabulaire ?
- Mon code est-il testé sur un exemple ?
- Ai-je traité les cas limites ?
- Ai-je justifié la complexité ou le résultat ?

À RETENIR

- Un **programme est une donnée** : interpréteurs, compilateurs, antivirus.
- **Calculabilité** ; indépendance vis-à-vis du langage (thèse de Church-Turing).
- **Problème de l'arrêt** : indécidable (Turing, 1936) ; preuve par l'absurde.