Démonstration automatique

Chantal Keller

Maths en mouvement - 2 décembre 2023

Avant tout

N'hésitez pas si vous avez des questions ou des remarques !

Pourquoi faire prouver des théorèmes par des ordinateurs ?

À vous !

Pourquoi faire prouver des théorèmes par des ordinateurs ?

À vous !

- pour remplacer les mathématicien·nes ?

Pourquoi faire prouver des théorèmes par des ordinateurs ?

À vous !

- pour remplacer les mathématicien·nes ?
- parce que c'est amusant ?

Pourquoi faire prouver des théorèmes par des ordinateurs ?

À vous !

- pour remplacer les mathématicien·nes ?
- parce que c'est amusant ?
- parce que ça permet de mieux comprendre les fondements des preuves ?

Pourquoi faire prouver des théorèmes par des ordinateurs ?

À vous !

- pour remplacer les mathématicien·nes ?
- parce que c'est amusant ?
- parce que ça permet de mieux comprendre les fondements des preuves ?
- pour des applications à la preuve et aux programmes ?

Des théorèmes nombreux mais « simples »

Preuve de programmes :

- génération de centaines d'obligations de preuve
- « simples » : *si* $i < n$ *alors* $i + 1 \leqslant n$

Démonstration interactive :

- preuve faite de raisonnement et de cas triviaux que l'on n'écrirait même pas sur le papier
- « simples » : *le nombre d'éléments de l'ensemble vide est égal à 0*

Des théorèmes nombreux mais « simples »

Preuve de programmes :

- génération de centaines d'obligations de preuve
- « simples » : *si* $i < n$ *alors* $i + 1 \leqslant n$

Démonstration interactive :

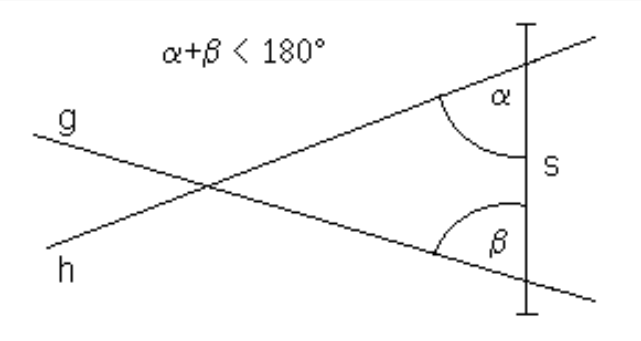
- preuve faite de raisonnement et de cas triviaux que l'on n'écrirait même pas sur le papier
- « simples » : *le nombre d'éléments de l'ensemble vide est égal à 0*

Que peut-on faire ?

Incomplétude, indécidabilité, complexité

Postulats d'Euclide - ∼-300 av. J.-C.

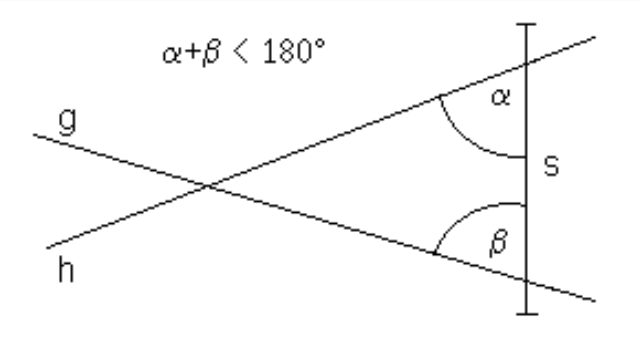
Cinq « demandes »
sur lesquelles repose la géométrie :



1. Entre deux points on peut toujours tracer une droite.
2. On peut toujours prolonger indéfiniment une droite tracée entre deux points.
3. Partant d'un point et d'une longueur donnés, on peut toujours tracer un cercle.
4. Tous les angles droits sont égaux entre eux.
5. Si une droite, tombant sur deux droites, fait les angles intérieurs du même côté plus petits que deux droits, ces droites, prolongées à l'infini, se rencontreront du côté où les angles sont plus petits que deux droits.

Postulats d'Euclide - ∼-300 av. J.-C.

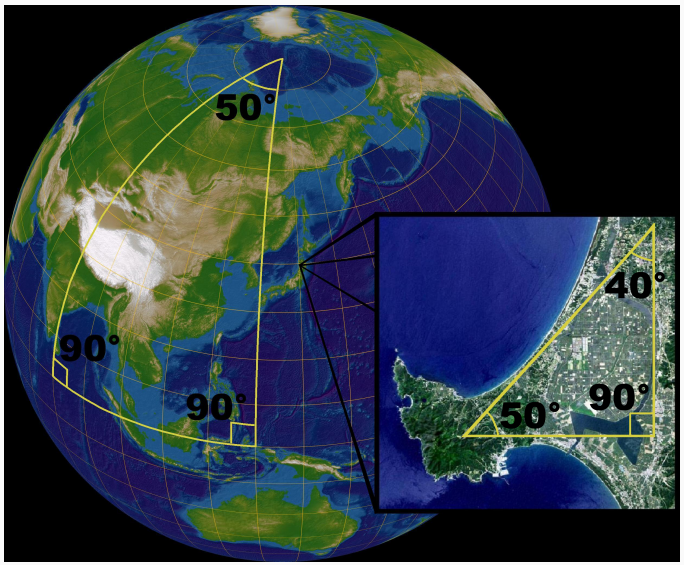
Cinq « demandes »
sur lesquelles repose la géométrie :

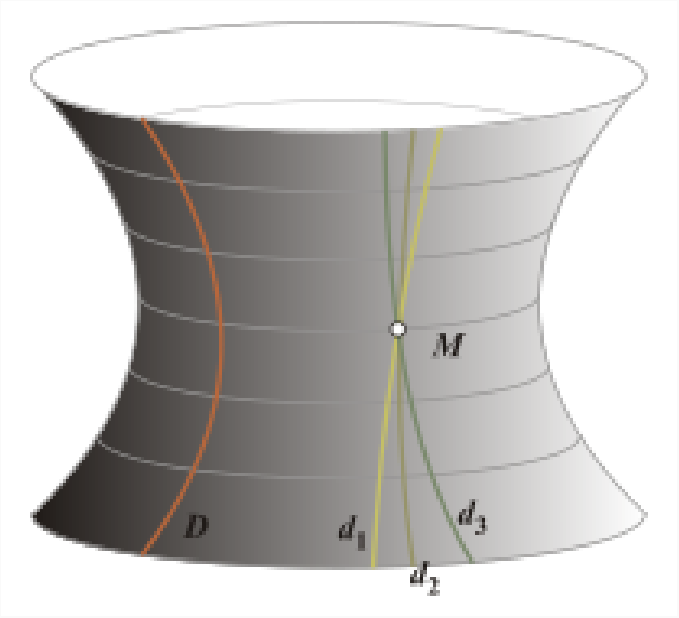


1. Entre deux points on peut toujours tracer une droite.
2. On peut toujours prolonger indéfiniment une droite tracée entre deux points.
3. Partant d'un point et d'une longueur donnés, on peut toujours tracer un cercle.
4. Tous les angles droits sont égaux entre eux.
5. **Si une droite, tombant sur deux droites, fait les angles intérieurs du même côté plus petits que deux droits, ces droites, prolongées à l'infini, se rencontreront du côté où les angles sont plus petits que deux droits.**

Géométries non euclidiennes - XIX^{e} siècle

Autres géométries validant les demandes 1 à 4 et invalidant la 5





En résumé

Dans un système logique (ex : les demandes 1 à 4)

Il y a des énoncés qui ne sont ni vrai, ni faux (ex : la demande 5)

Ils ne sont donc pas prouvable (et a fortiori pas par un ordinateur)

On les appelle **indépendants du système logique**

Incomplétude ou Indécidabilité logique

🥲 La plupart des systèmes « intéressants » sont incomplets (XX^{e} siècle) :

- *Éléments* d'Euclide
- système logique de Lean
- arithmétique
- ...

🙂 En pratique :

pas le point bloquant pour l'automatisation

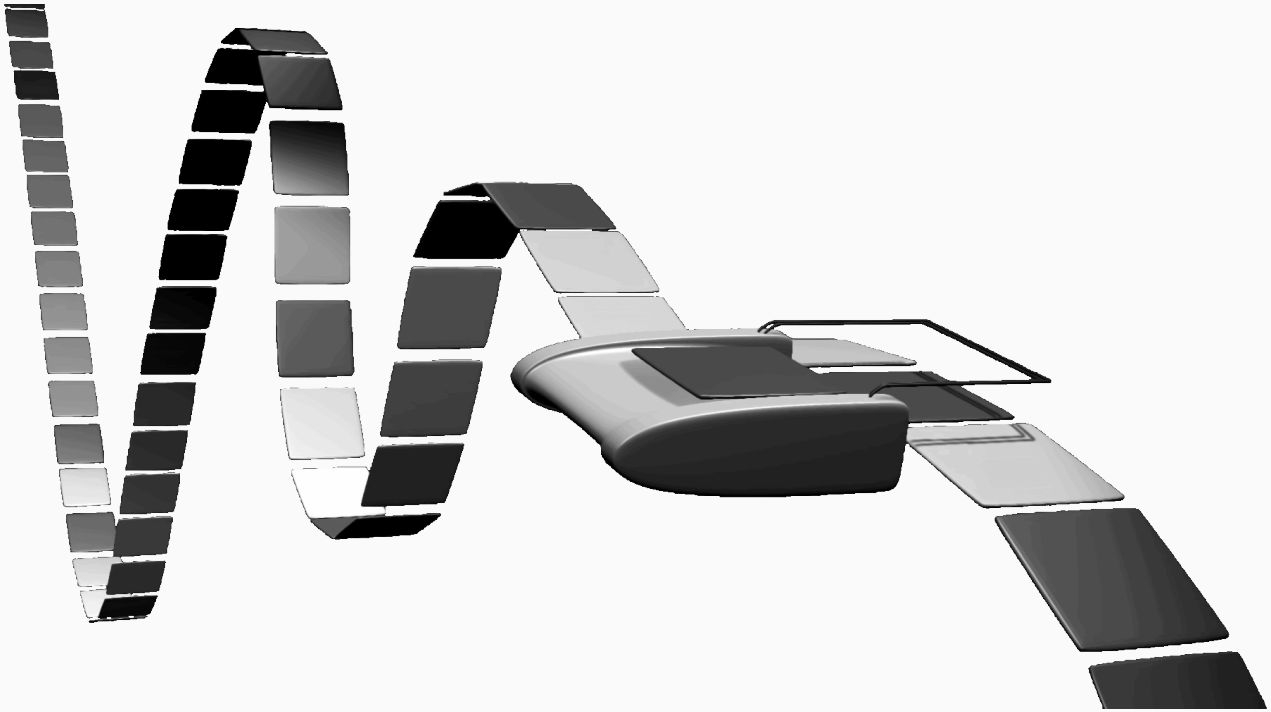
Question

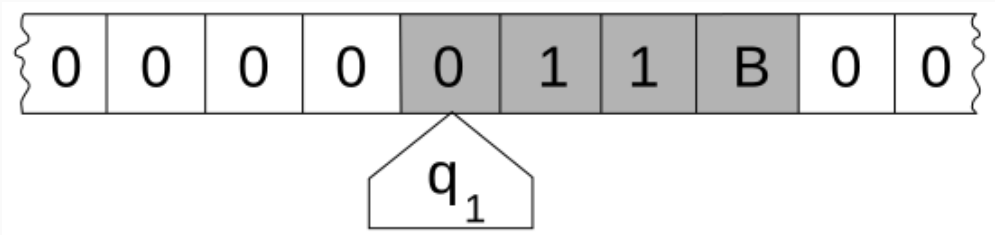
On se fixe un système logique.

Peut-on écrire un programme :

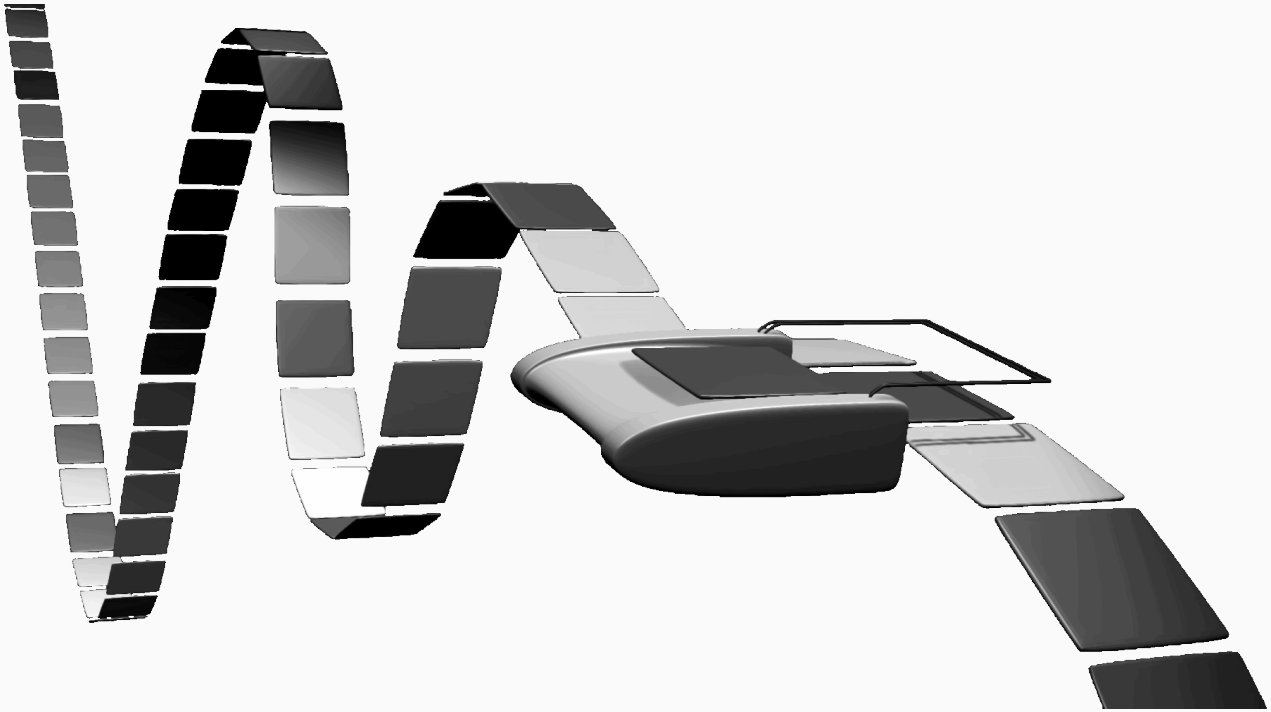
- entrée : énoncé e
- sortie : e est vrai/e est faux/e est indépendant du système logique
- et terminant ?

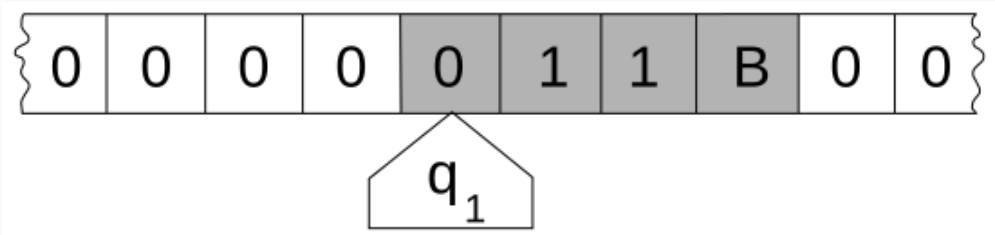
Un modèle de calcul : la machine de Turing





Un modèle de calcul : la machine de Turing





un programme = une machine de Turing

Modèle de calcul $\rightarrow$ question formelle

Définition

Une machine de Turing est un quintuplet $(Q, \Gamma, q_0, \delta, F)$ où Q est un ensemble d'états, Γ est ...

Étant donné un système logique, existe-t-il une machine de Turing :

- entrée : énoncé e
- sortie : e est vrai/e est faux/e est indépendant du système logique
- et terminant ?

Modèle de calcul $\rightarrow$ question formelle

Définition

Une machine de Turing est un quintuplet $(Q, \Gamma, q_0, \delta, F)$ où Q est un ensemble d'états, Γ est ...

Étant donné un système logique, existe-t-il une machine de Turing :

- entrée : énoncé e
- sortie : e est vrai/e est faux/e est indépendant du système logique
- et terminant ?

Réponse : cela dépend du système logique !

Indécidabilité algorithmique

🥲 Beaucoup de systèmes logiques indécidables :

- logique que vous utilisez en maths
- Lean
- logique du premier ordre
- arithmétique

🙂 Des systèmes intéressants décidables :

- arithmétique sans la multiplication
- logique propositionelle (sans « pout tout » et « il existe »)
- arithmétique modulo (utile pour des programmes manipulant des entiers bornés)

En résumé

Verre à moitié vide	Verre à moitié plein
Indécidabilité logique	Aucun impact en pratique
Indécidabilité algorithmique	Certaines classes décidables

En résumé

Verre à moitié vide	Verre à moitié plein
Indécidabilité logique	Aucun impact en pratique
Indécidabilité algorithmique	Certaines classes décidables
Complexité algorithmique	Algorithmes souvent efficaces en pratique

En résumé

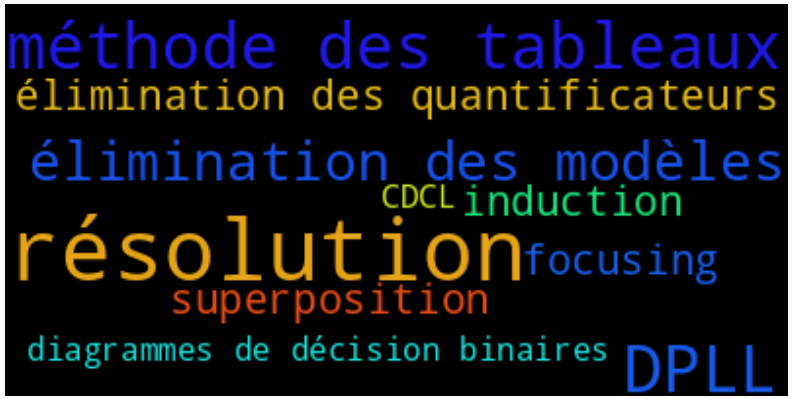
Verre à moitié vide	Verre à moitié plein
Indécidabilité logique	Aucun impact en pratique
Indécidabilité algorithmique	Certaines classes décidables
Complexité algorithmique	Algorithmes souvent efficaces en pratique

Nombreux sujets de recherche :

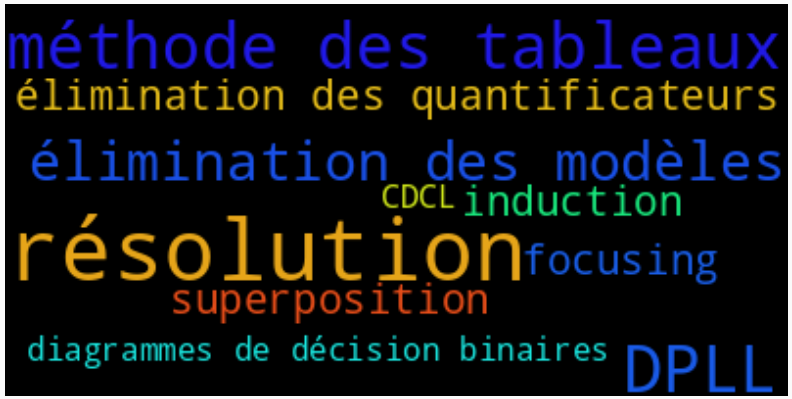
- théorie de la calculabilité et de la complexité
- algorithmes performants

Démonstration automatique en pratique

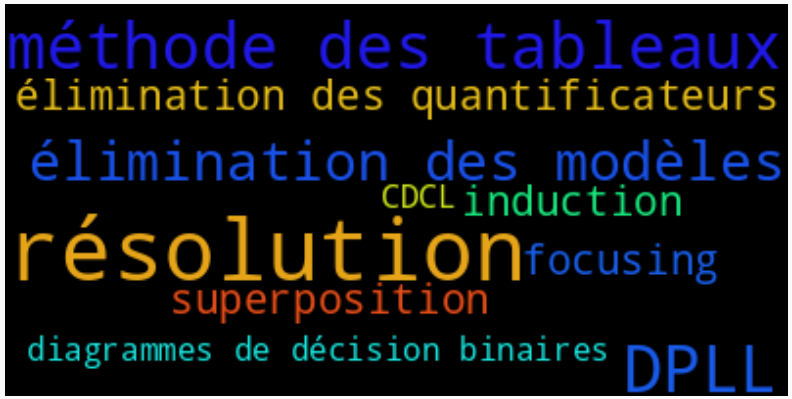
De très nombreux algorithmes. . .



De très nombreux algorithmes. . . et outils

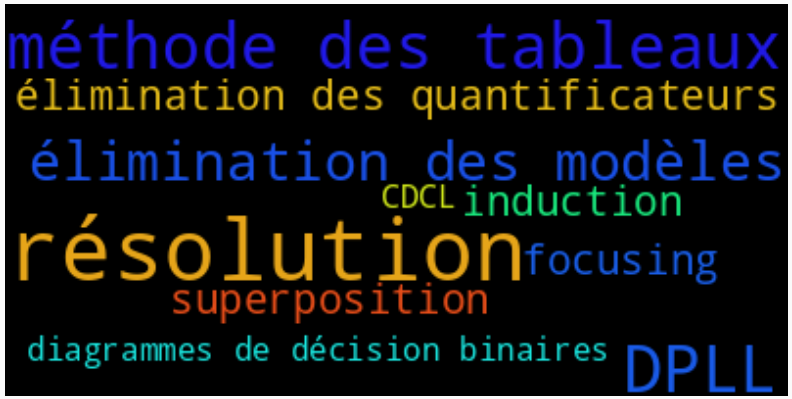


De très nombreux algorithmes. . . et outils



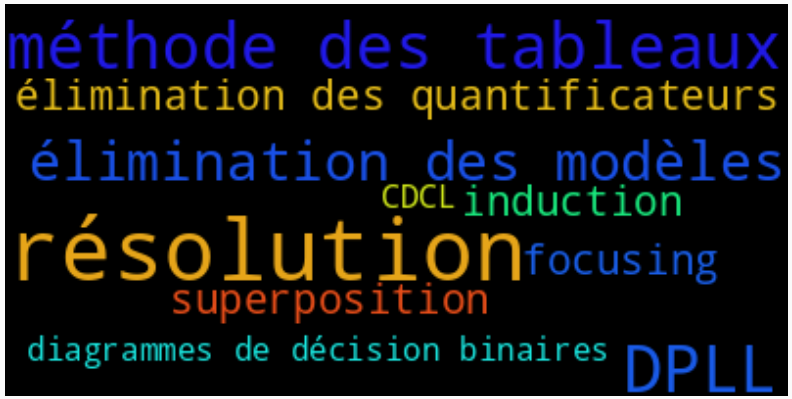
CVC5

De très nombreux algorithmes. . . et outils



De très nombreux algorithmes. . . et outils

VeriT

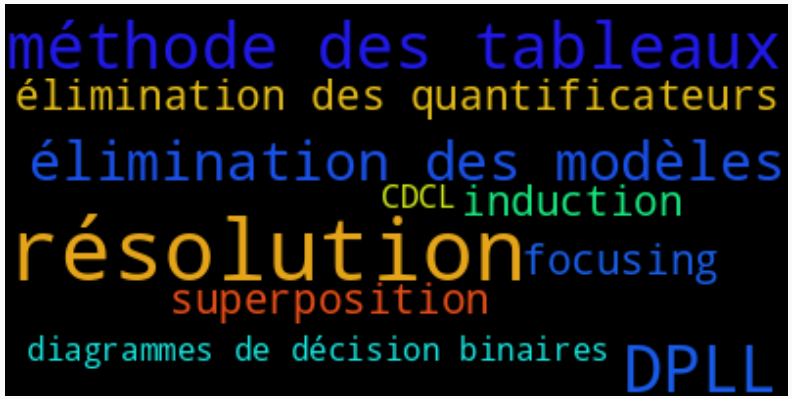


CVC5

De très nombreux algorithmes. . . et outils

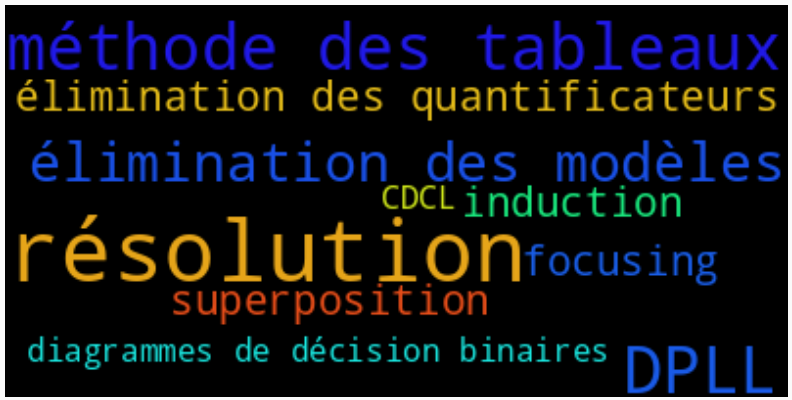
spass

veriT



CVC5

De très nombreux algorithmes. . . et outils

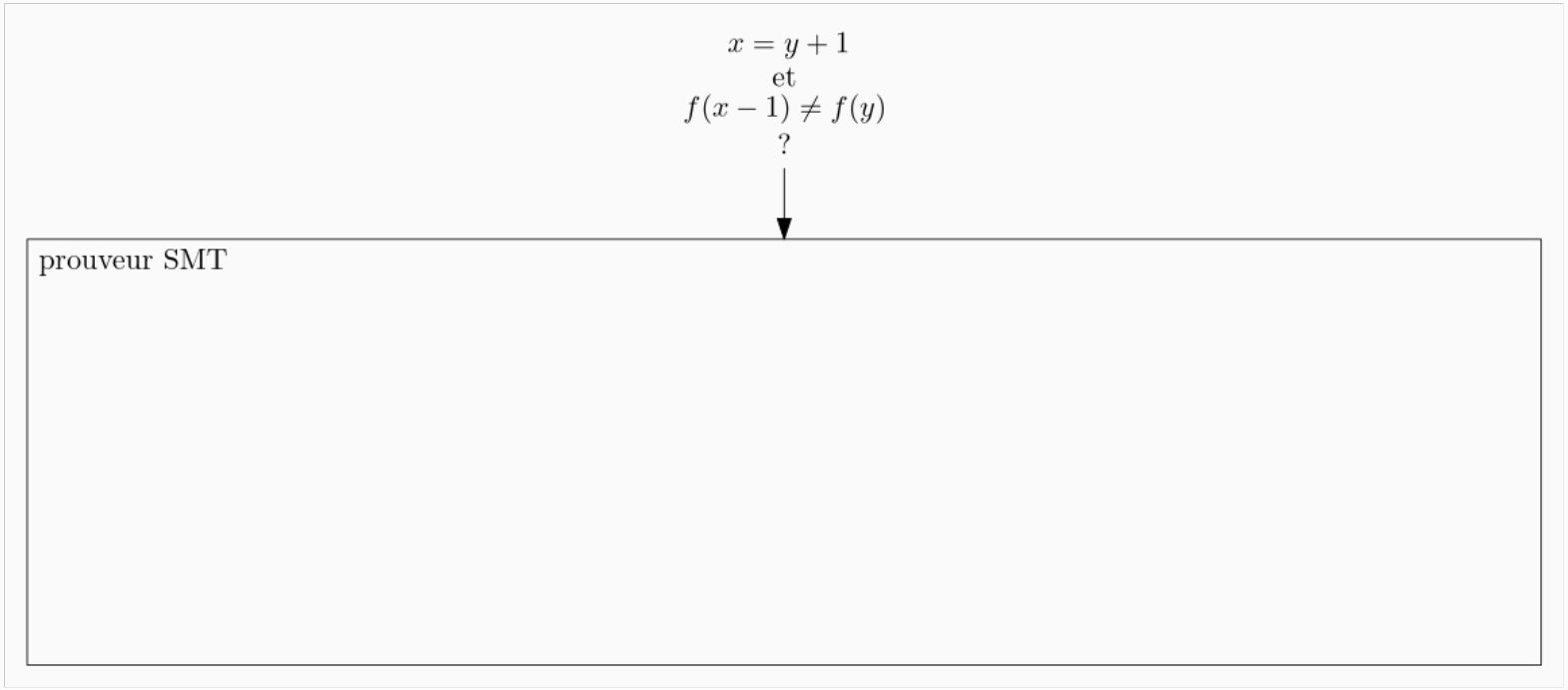


Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?

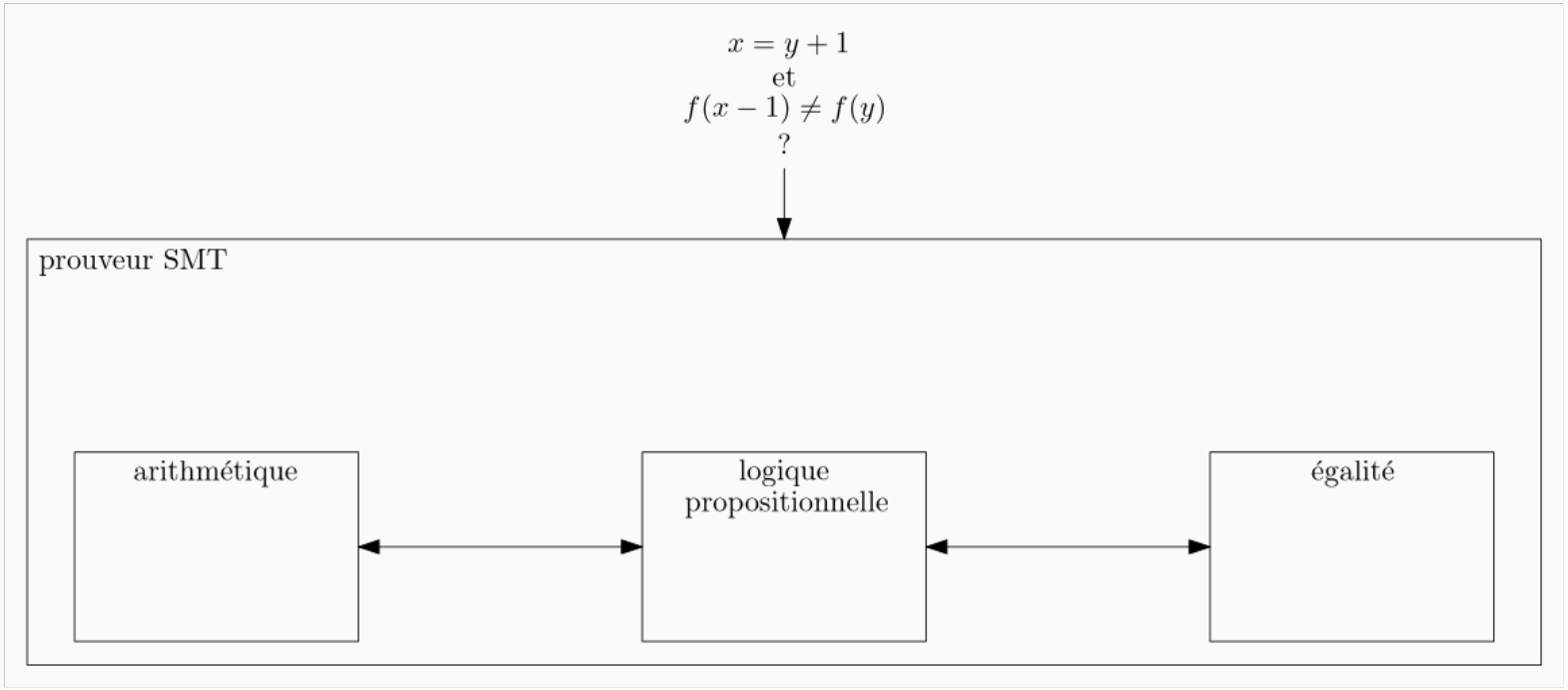
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



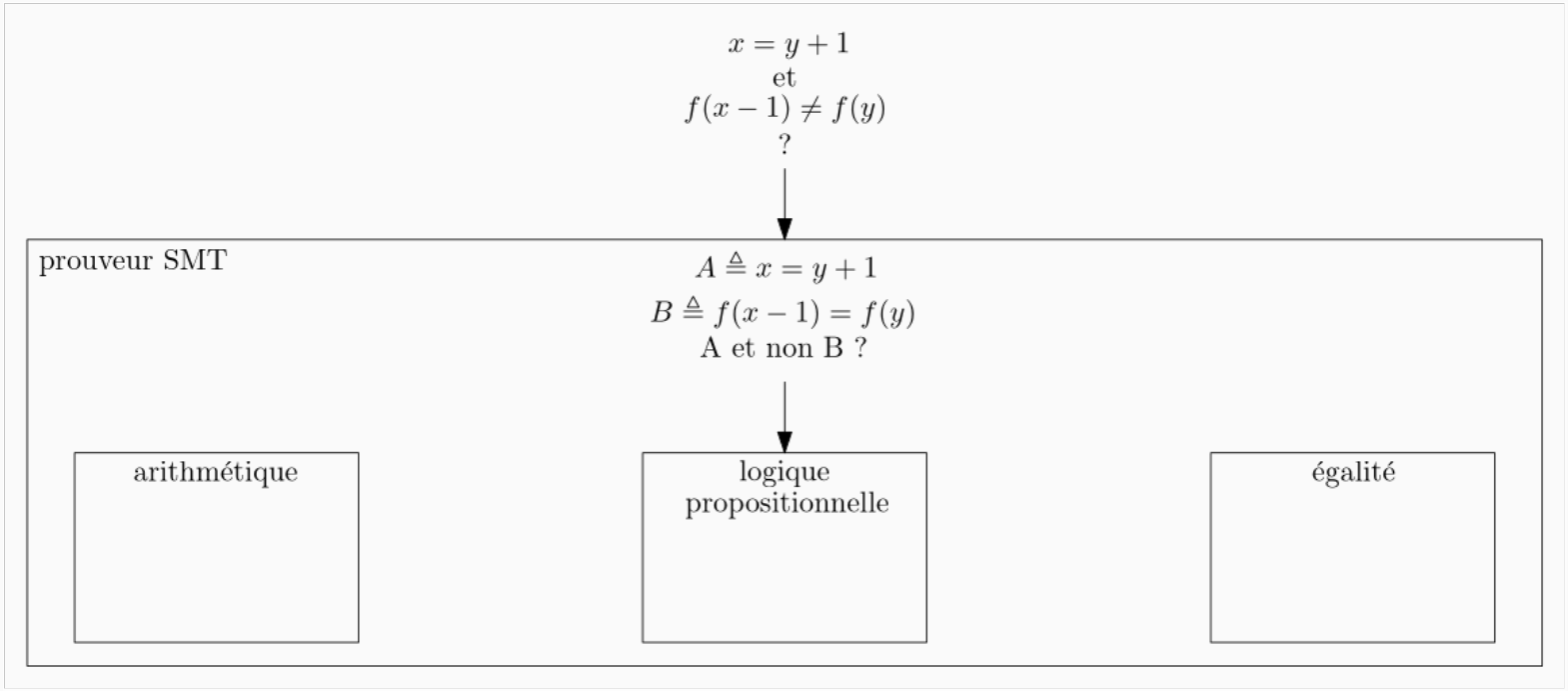
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



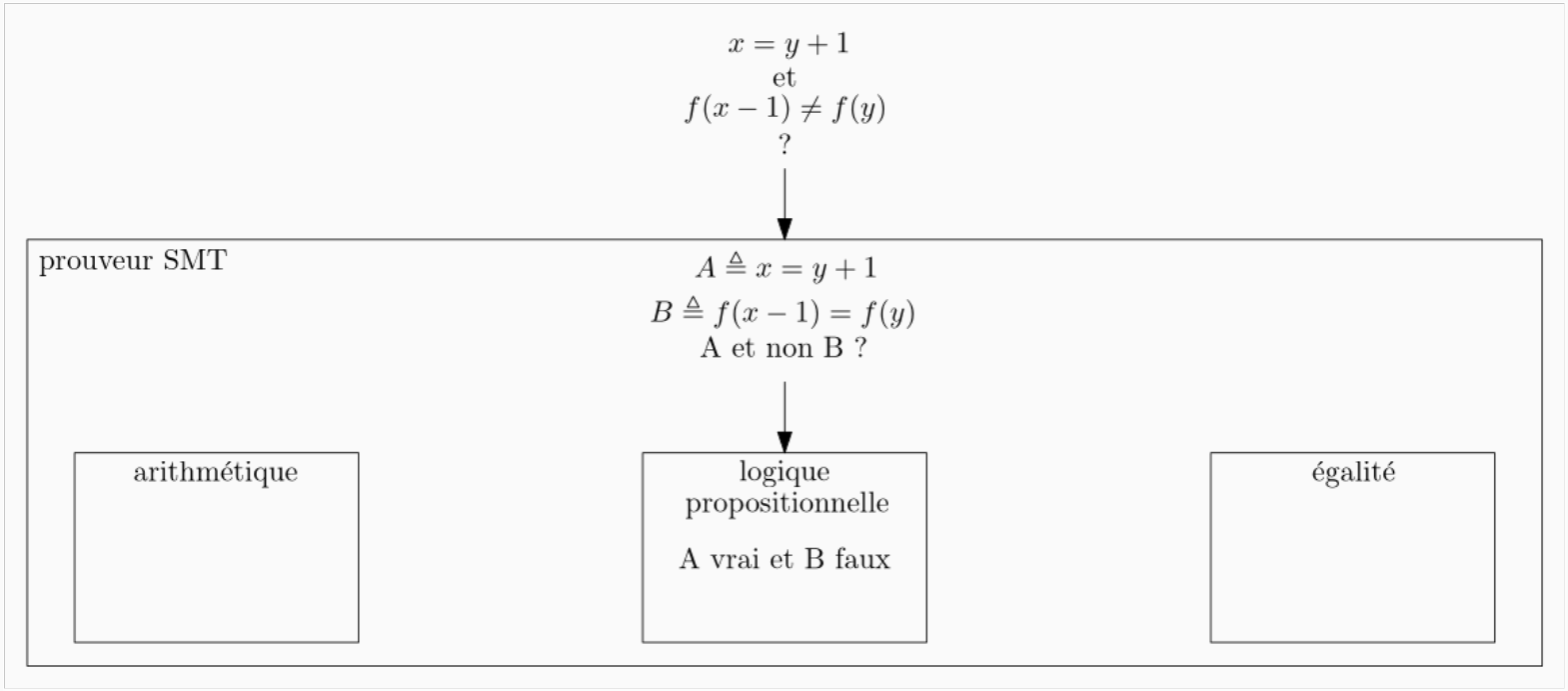
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



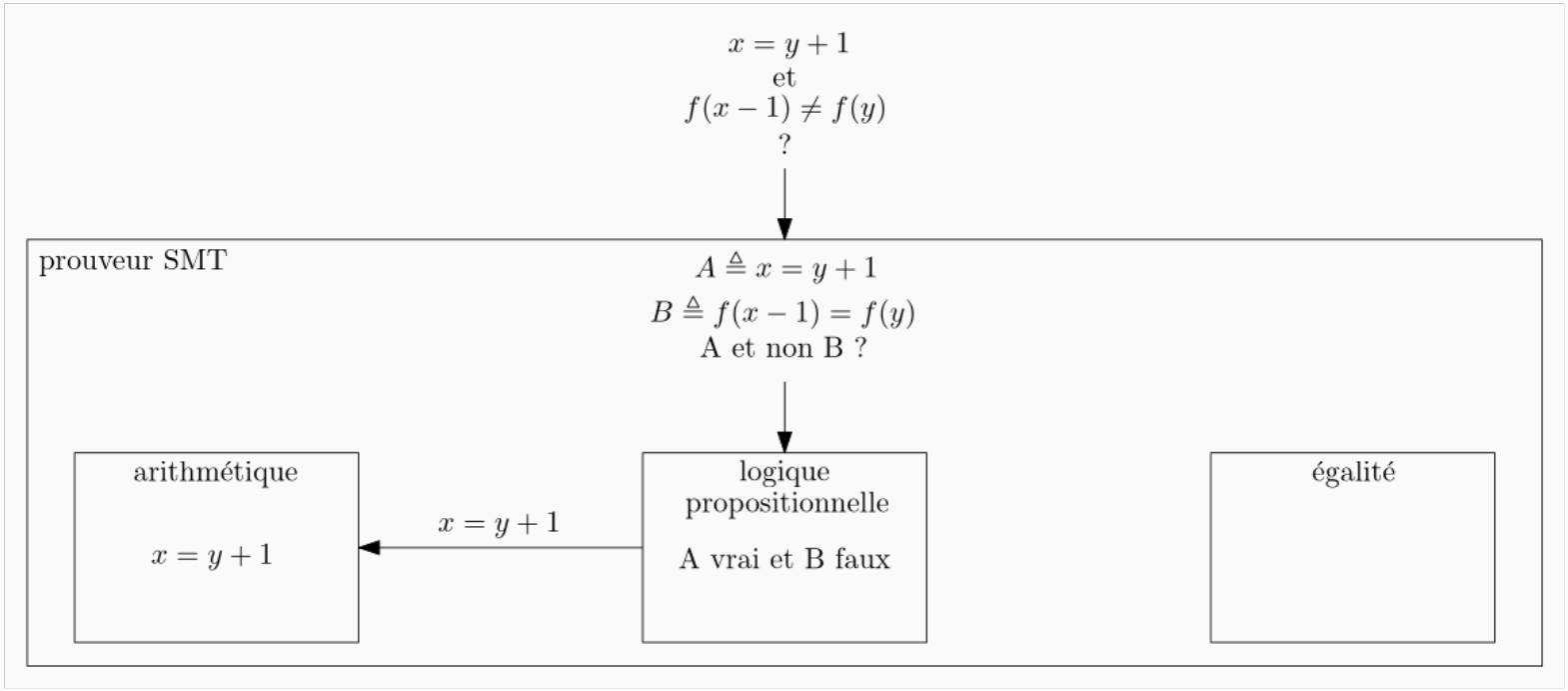
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



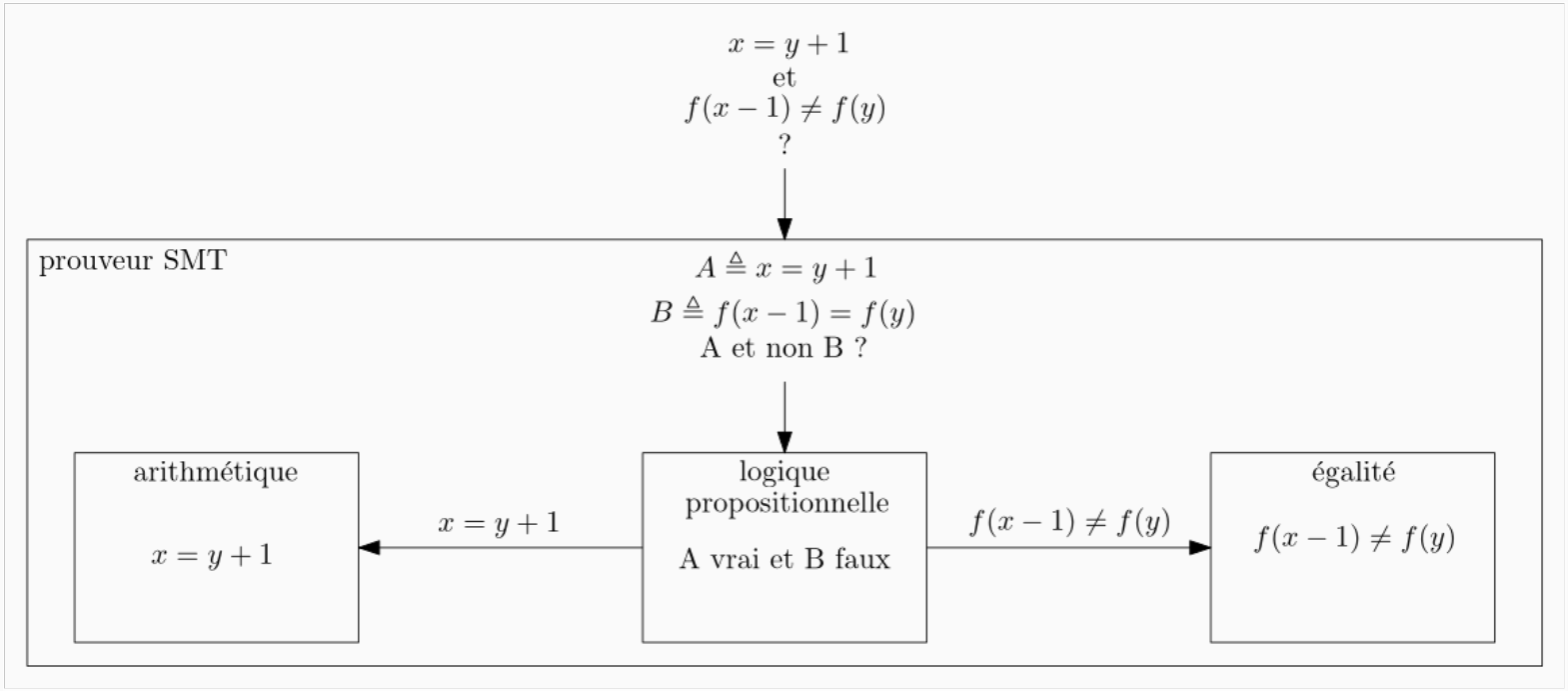
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



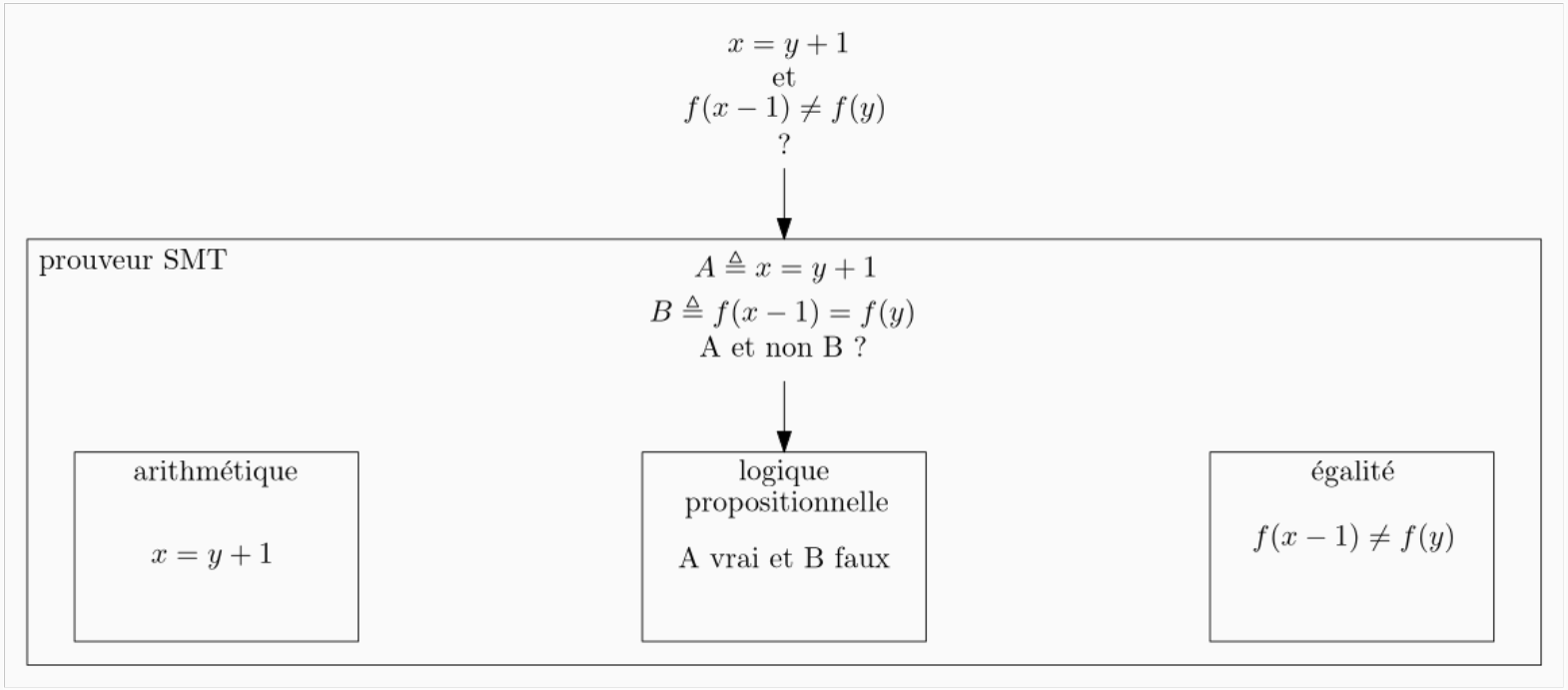
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



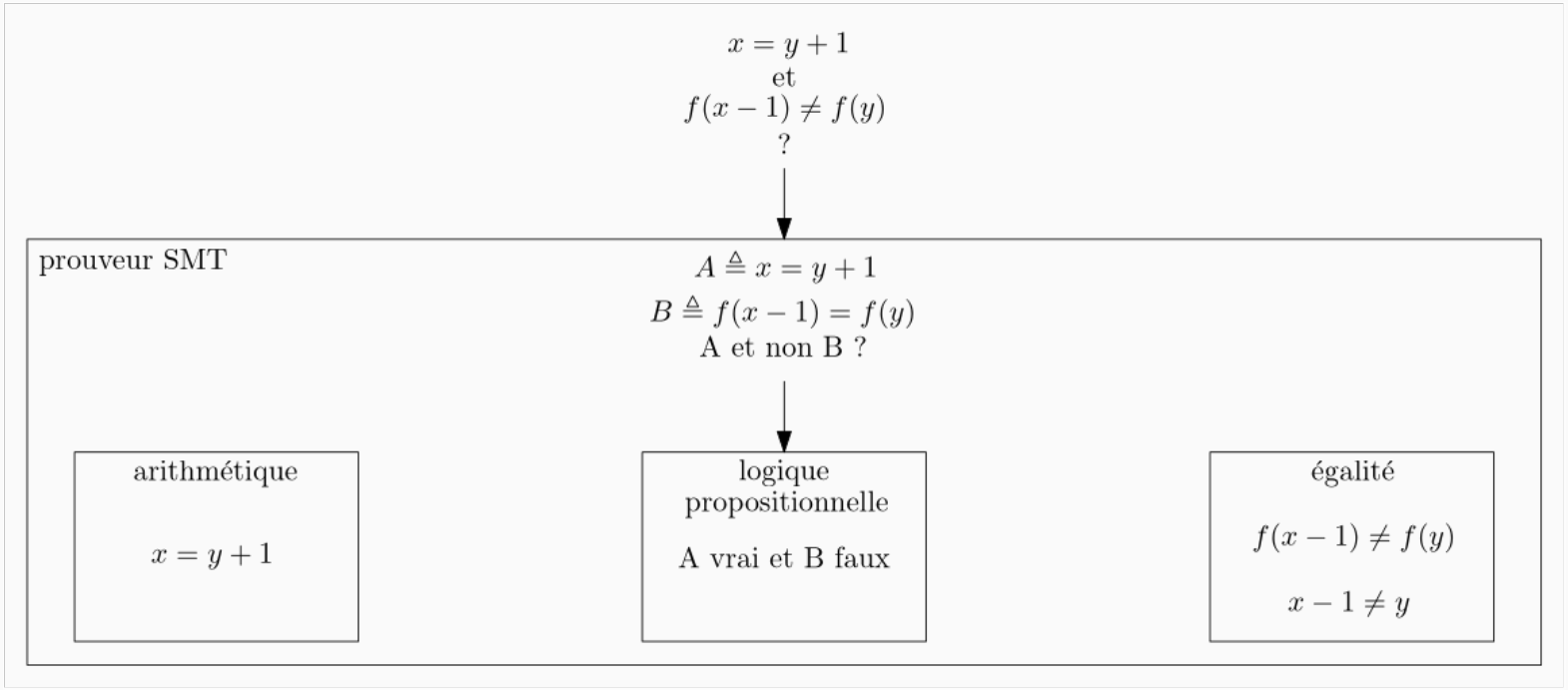
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



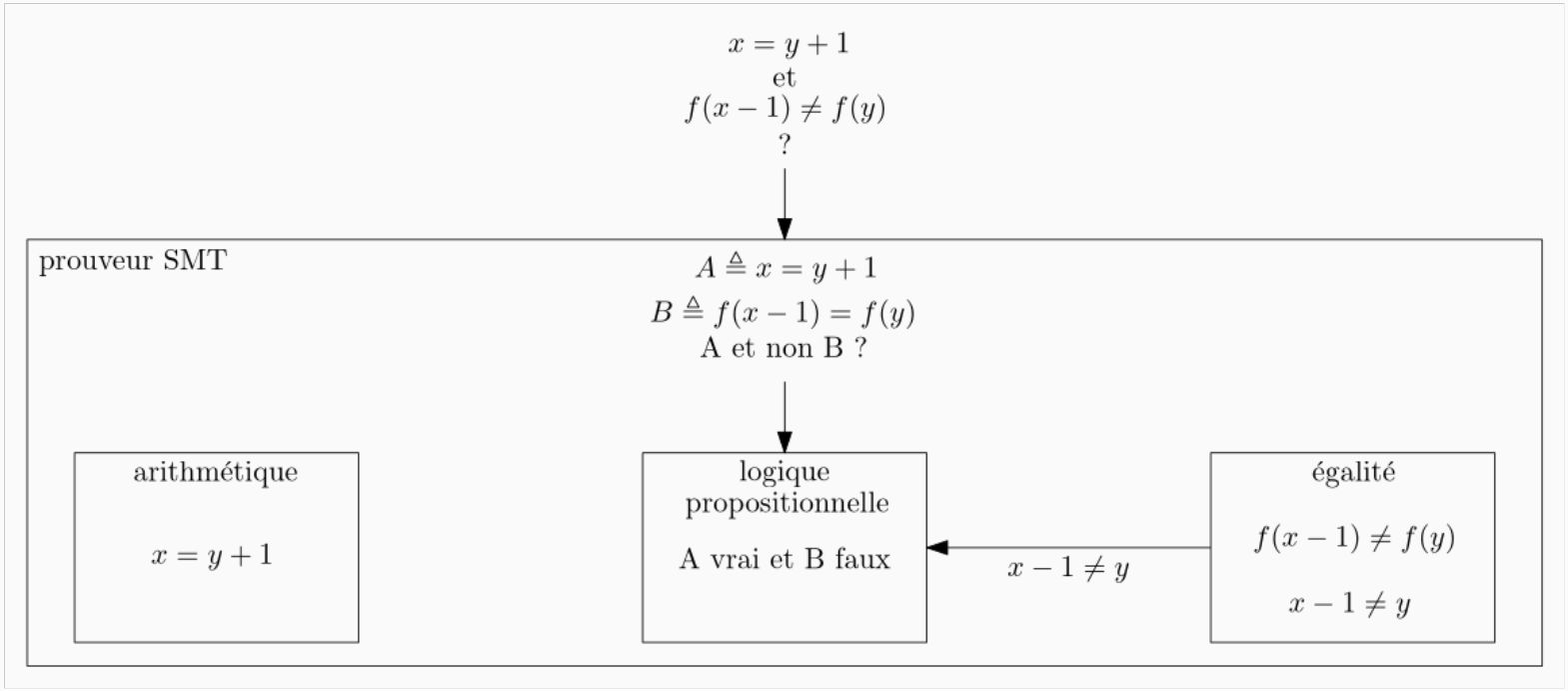
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



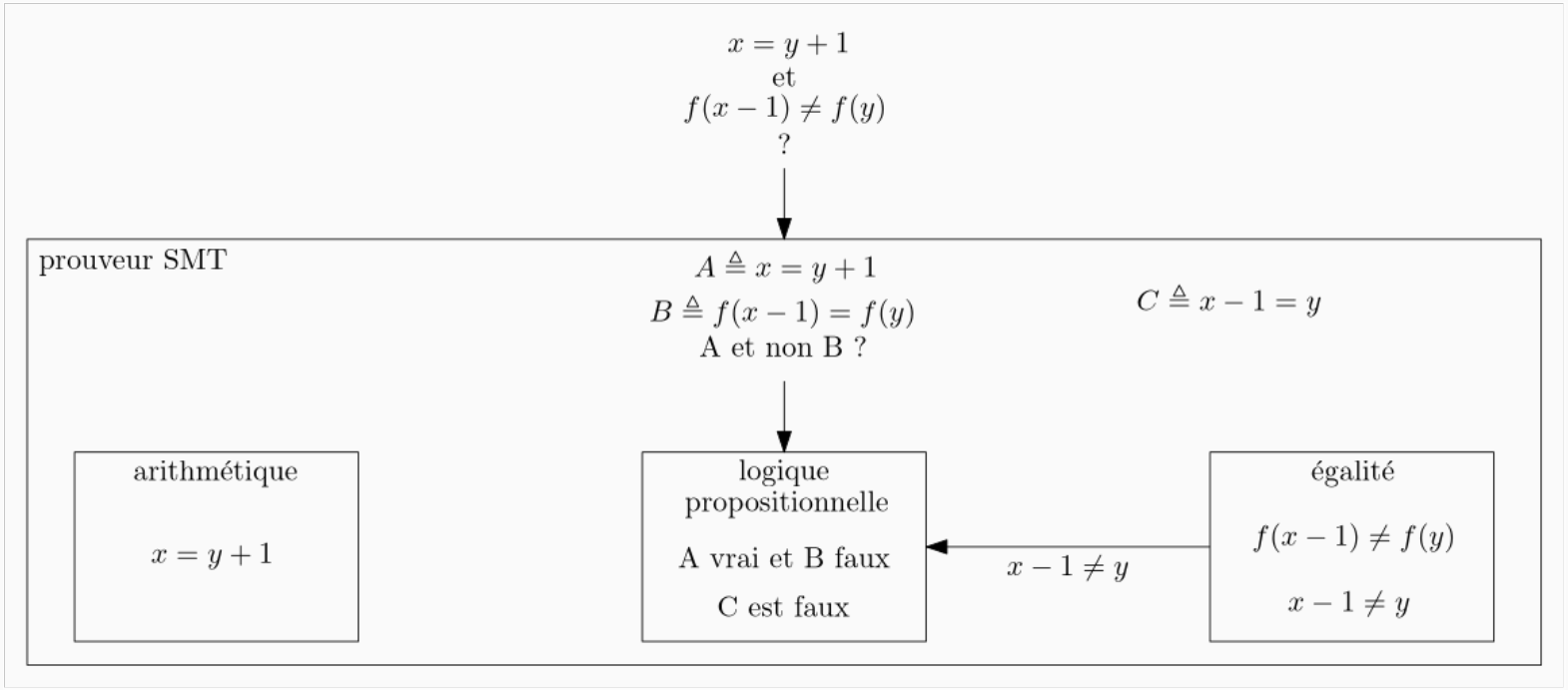
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



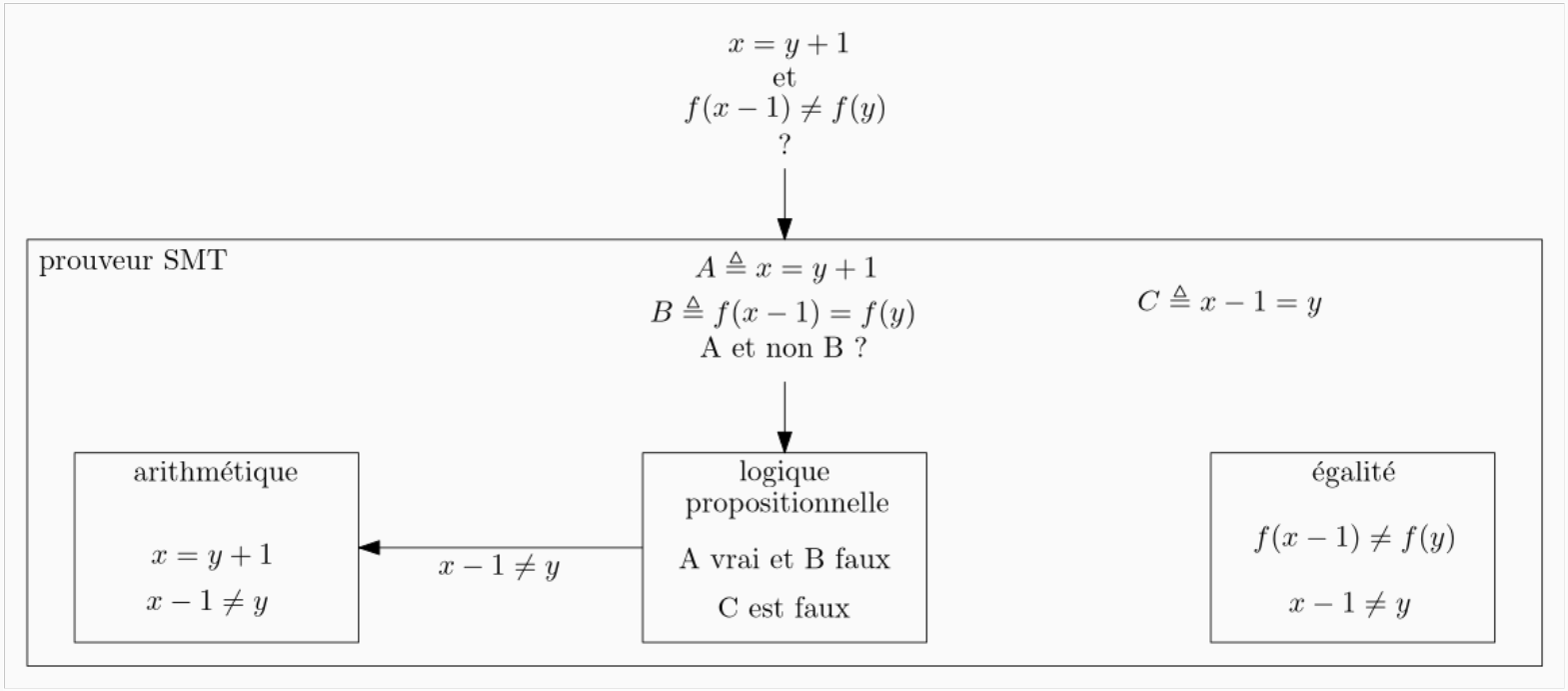
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



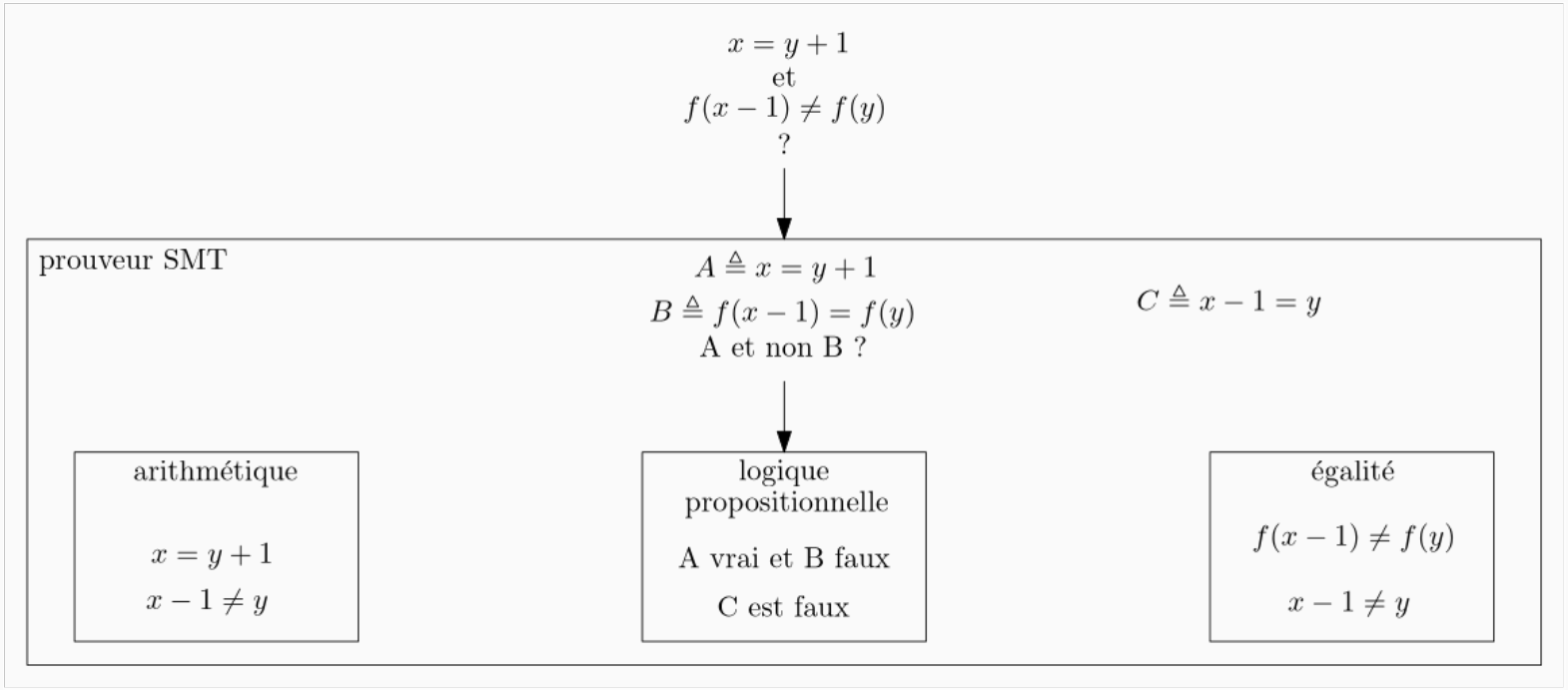
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



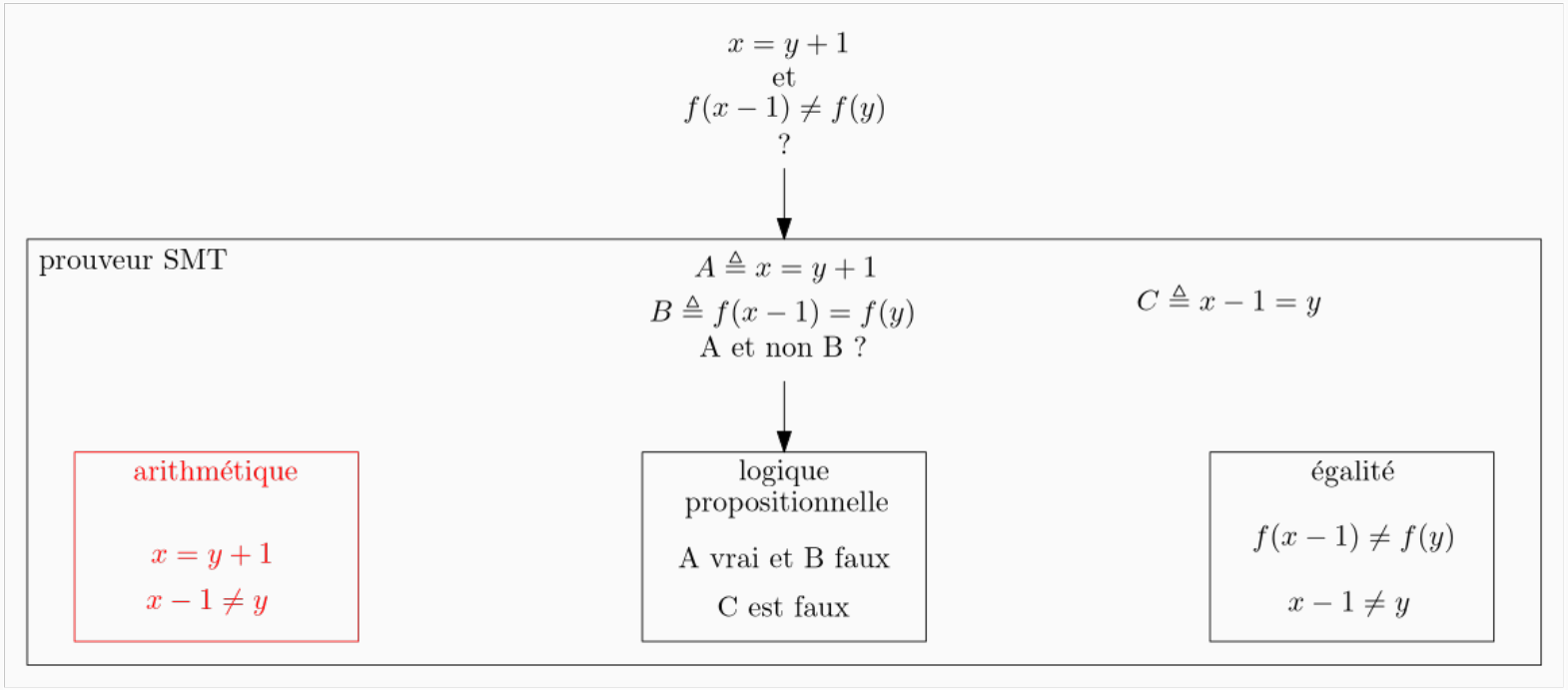
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



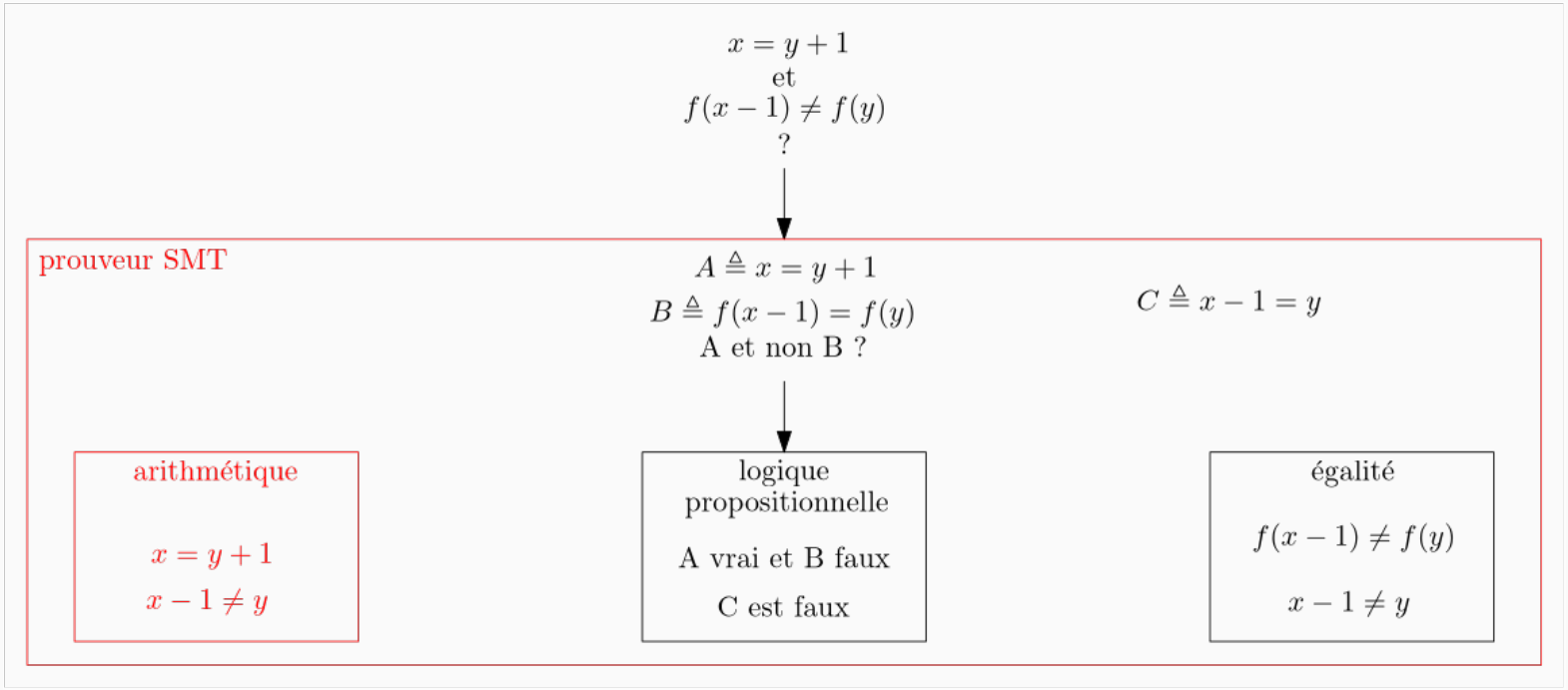
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



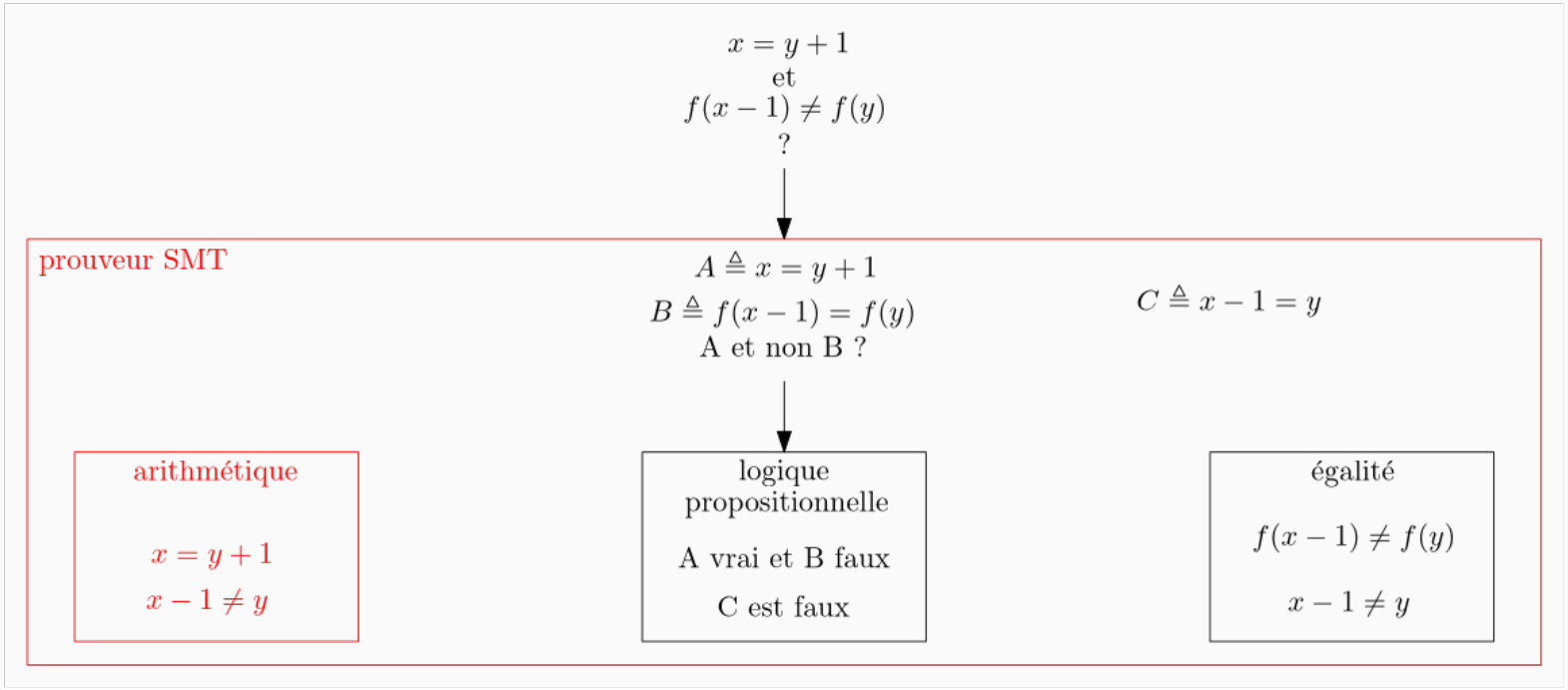
Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



Exemple des prouveurs de Satisfiabilité Modulo Théories (SMT)

Peut-on avoir à la fois $x = y + 1$ et $f(x - 1) \neq f(y)$?



NON !

Super !

Super !

Oui, mais. . .

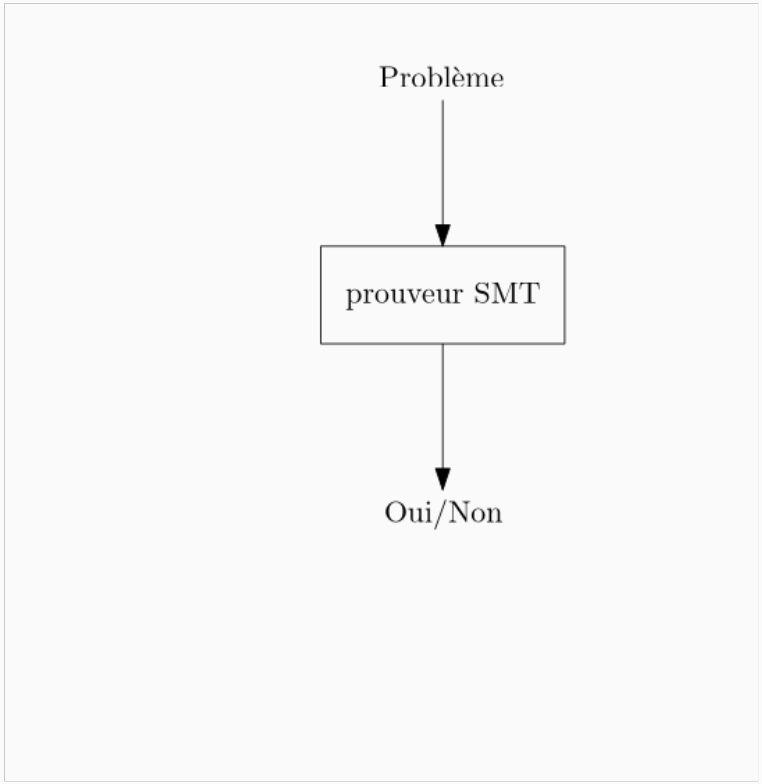
Super !

Oui, mais. . .

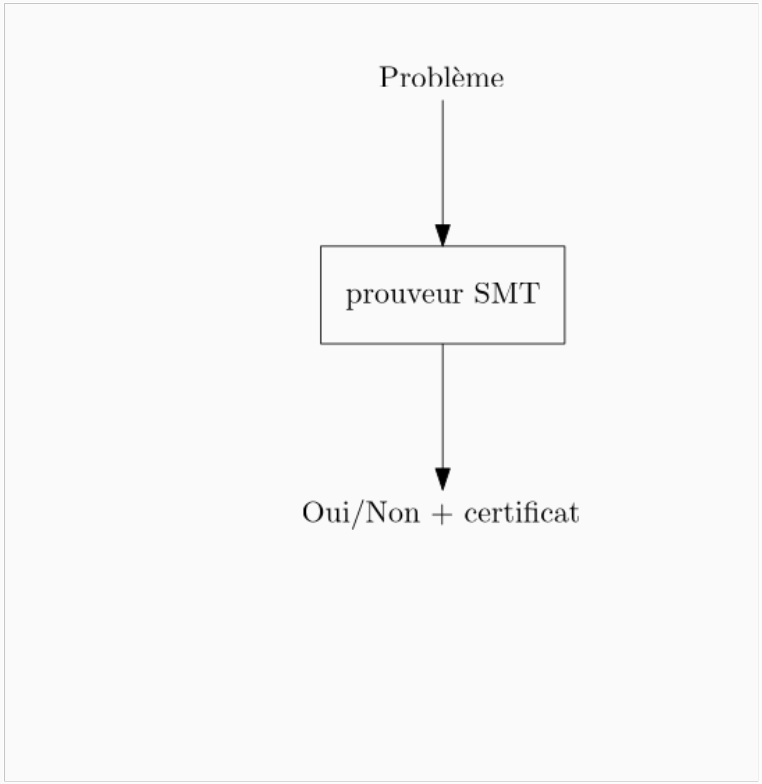
comment lui faire confiance ? ?

SMTCoq

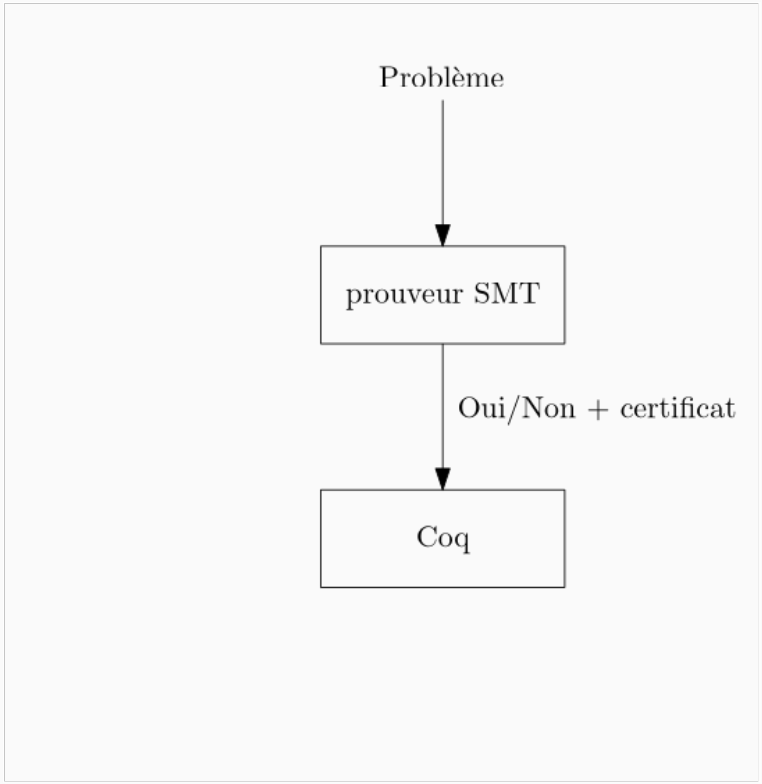
SMTCoq



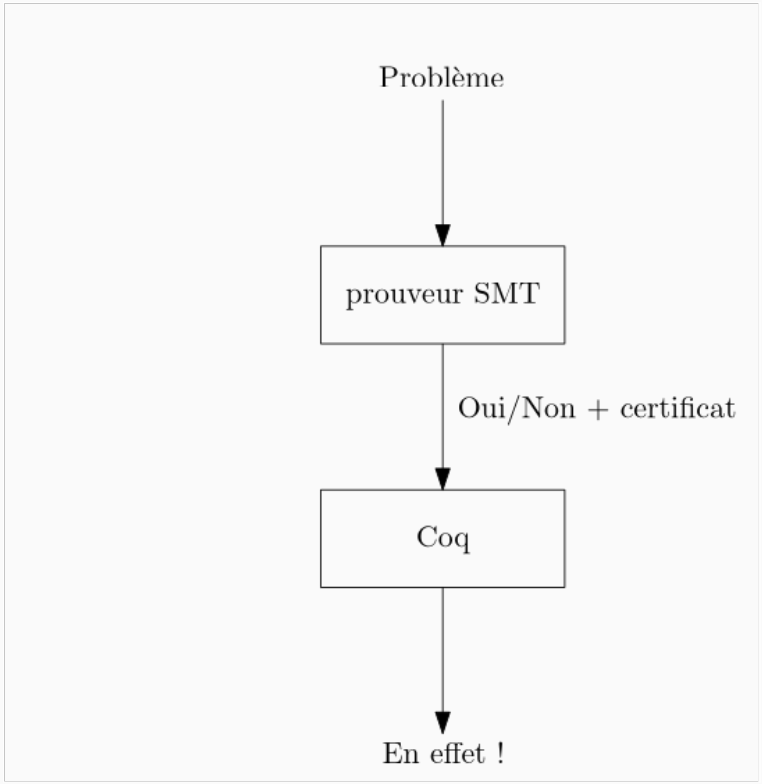
SMTCoq



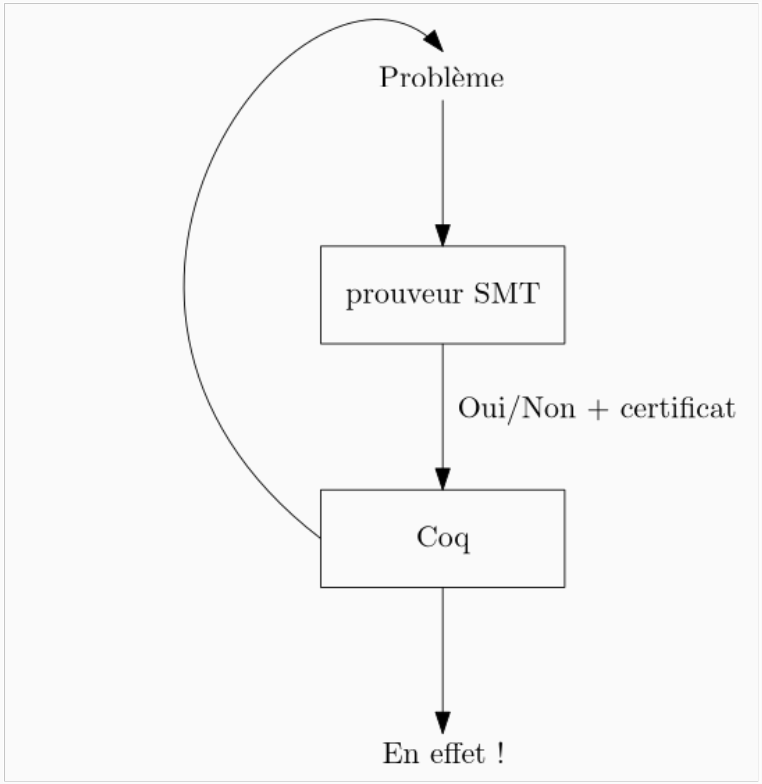
SMTCoq



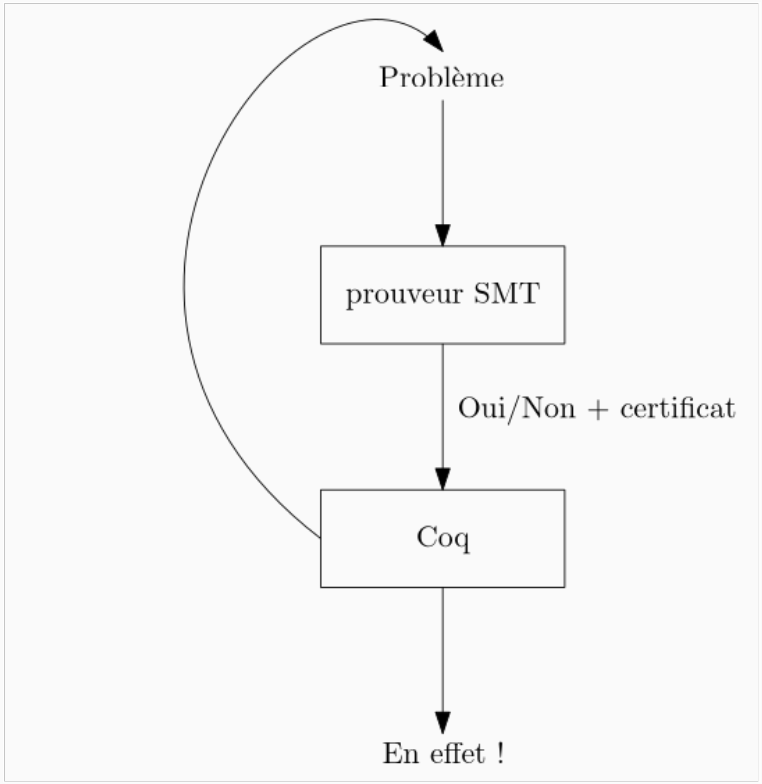
SMTCoq



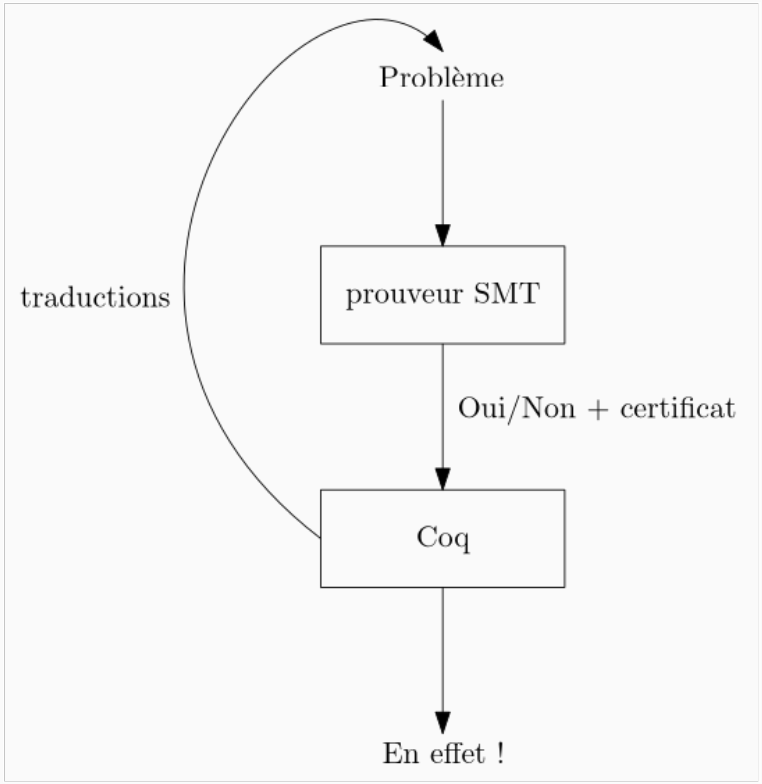
SMTCoq



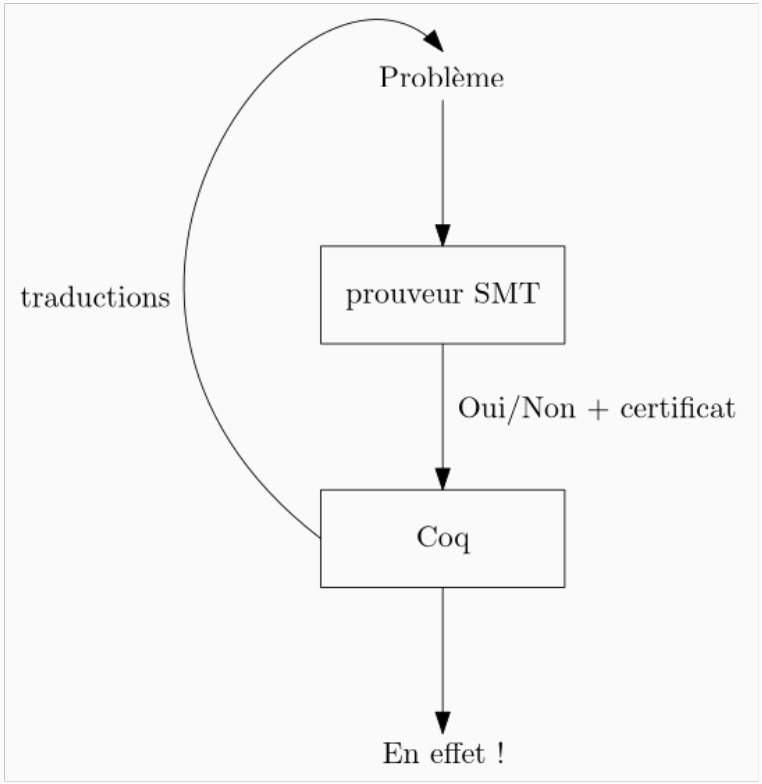
SMTCoq et Sniper



SMTCoq et Sniper



SMTCoq et Sniper

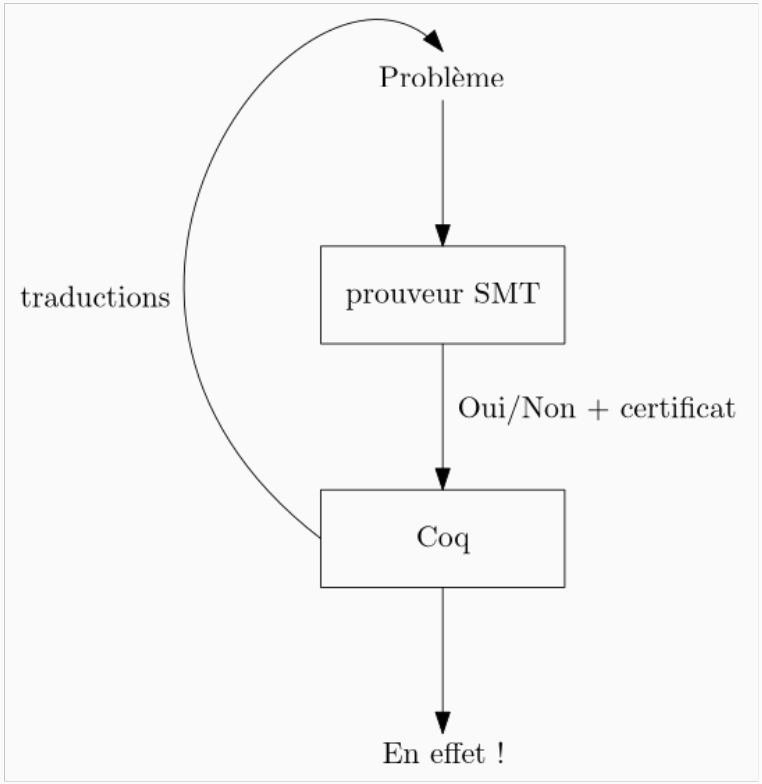


Difficulté : très grande différence de logique

Approche compositionnelle :

- bibliothèques de « petites » transformations
- choix des transformations selon la formule à prouver

SMTCoq et Sniper



Difficulté : très grande différence de logique

Approche compositionnelle :

- bibliothèques de « petites » transformations
- choix des transformations selon la formule à prouver

Remarque : les transformations aussi doivent produire des preuves !

Conclusion

On ne peut pas tout automatiser en théorie. . .

mais on peut quand même faire beaucoup en pratique

Mon sujet de recherche :

combiner preuve automatique et preuve interactive