Automates et Logique

Roger Villemaire

Département d'informatique
UQAM

Séminaire de Logique
5 décembre 2008

Plan

Plan

Alphabets, mots et langages

- Un *alphabet* $\mathcal{A}$ est un ensemble fini, par exemple : $\{ouvrir, lire, fermer\}$
- Un mot sur l'alphabet $\mathcal{A}$ est une suite d'éléments de $\mathcal{A}$, par exemple $ouvrir\ fermer\ lire\ fermer$.
- Un *langage* sur un alphabet $\mathcal{A}$ est un ensemble de mots, par exemple l'ensemble des suites d'actions correctes où on doit avoir *ouvrir* avant *lire* et *fermer* avant un nouvel *ouvrir*.

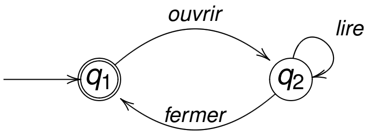
Alphabets, mots et langages

- Un *alphabet* $\mathcal{A}$ est un ensemble fini, par exemple : $\{\textit{ouvrir}, \textit{lire}, \textit{fermer}\}$
- Un mot sur l'alphabet $\mathcal{A}$ est une suite d'éléments de $\mathcal{A}$, par exemple *ouvrir fermer lire fermer*.
- Un *langage* sur un alphabet $\mathcal{A}$ est un ensemble de mots, par exemple l'ensemble des suites d'actions correctes où on doit avoir *ouvrir* avant *lire* et *fermer* avant un nouvel *ouvrir*.

Alphabets, mots et langages

- Un *alphabet* $\mathcal{A}$ est un ensemble fini, par exemple : $\{\mathit{ouvrir}, \mathit{lire}, \mathit{fermer}\}$
- Un mot sur l'alphabet $\mathcal{A}$ est une suite d'éléments de $\mathcal{A}$, par exemple *ouvrir fermer lire fermer*.
- Un *langage* sur un alphabet $\mathcal{A}$ est un ensemble de mots, par exemple l'ensemble des suites d'actions correctes où on doit avoir *ouvrir* avant *lire* et *fermer* avant un nouvel *ouvrir*.

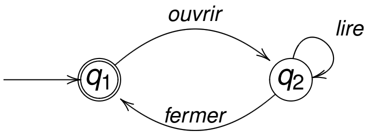
Un automate sur l'alphabet $\mathcal{A}$ est formé de :



- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Un fonction de transition $Q \times \mathcal{A} \rightarrow Q$
- Un ensemble d'états finals $F \subseteq Q$.

Un mot sur $\mathcal{A}$ est *accepté* par l'automate si après sa lecture on se trouve dans un état final.
Un langage est *reconnaissable* si c'est l'ensemble des mots reconnus par un automate.

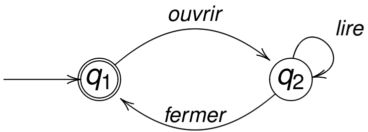
Un automate sur l'alphabet $\mathcal{A}$ est formé de :



- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Un fonction de transition $Q \times \mathcal{A} \to Q$
- Un ensemble d'états finals $F \subseteq Q$.

Un mot sur $\mathcal{A}$ est *accepté* par l'automate si après sa lecture on se trouve dans un état final.
Un langage est *reconnaissable* si c'est l'ensemble des mots reconnus par un automate.

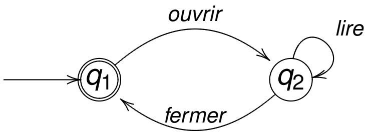
Un automate sur l'alphabet $\mathcal{A}$ est formé de :



- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Un fonction de transition $Q \times \mathcal{A} \rightarrow Q$
- Un ensemble d'états finals $F \subseteq Q$.

Un mot sur $\mathcal{A}$ est *accepté* par l'automate si après sa lecture on se trouve dans un état final.
Un langage est *reconnaissable* si c'est l'ensemble des mots reconnus par un automate.

Un automate sur l'alphabet $\mathcal{A}$ est formé de :

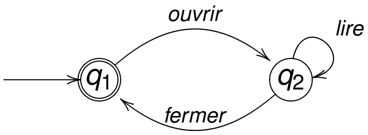


- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Un fonction de transition $Q \times \mathcal{A} \to Q$
- Un ensemble d'états finals $F \subseteq Q$.

Un mot sur $\mathcal{A}$ est *accepté* par l'automate si après sa lecture on se trouve dans un état final.

Un langage est *reconnaissable* si c'est l'ensemble des mots reconnus par un automate.

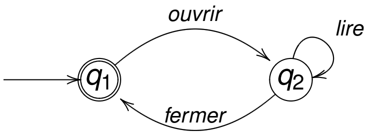
Un automate sur l'alphabet $\mathcal{A}$ est formé de :



- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Un fonction de transition $Q \times \mathcal{A} \rightarrow Q$
- Un ensemble d'états finals $F \subseteq Q$.

Un mot sur $\mathcal{A}$ est *accepté* par l'automate si après sa lecture on se trouve dans un état final.
Un langage est *reconnaissable* si c'est l'ensemble des mots reconnus par un automate.

Un automate sur l'alphabet $\mathcal{A}$ est formé de :

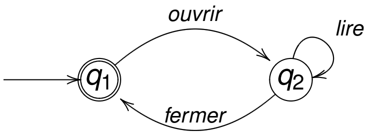


- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Un fonction de transition $Q \times \mathcal{A} \to Q$
- Un ensemble d'états finals $F \subseteq Q$.

Un mot sur $\mathcal{A}$ est *accepté* par l'automate si après sa lecture on se trouve dans un état final.

Un langage est *reconnaissable* si c'est l'ensemble des mots reconnus par un automate.

Un automate sur l'alphabet $\mathcal{A}$ est formé de :



- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Un fonction de transition $Q \times \mathcal{A} \to Q$
- Un ensemble d'états finals $F \subseteq Q$.

Un mot sur $\mathcal{A}$ est *accepté* par l'automate si après sa lecture on se trouve dans un état final.
Un langage est *reconnaissable* si c'est l'ensemble des mots reconnus par un automate.

Expression régulière

Une expression régulière est une représentation d'un langage à partir de :

- a qui représente le singleton $\{a\}$ pour $a \in \mathcal{A}$.
- $L_1 \cdot L_2$ qui représente la *concaténation* des langages L_1 et L_2
- $L_1 + L_2$ qui représente l'union des langages L_1 et L_2
- L^* qui représente l'*itération* du langage L

Un langage est *rationnel* s'il est représentable par une expression régulière.

Expression régulière

Une expression régulière est une représentation d'un langage à partir de :

- a qui représente le singleton $\{a\}$ pour $a \in \mathcal{A}$.
- $L_1 \cdot L_2$ qui représente la *concaténation* des langages L_1 et L_2
- $L_1 + L_2$ qui représente l'union des langages L_1 et L_2
- L^* qui représente l'*itération* du langage L

Un langage est *rationnel* s'il est représentable par une expression régulière.

Expression régulière

Une expression régulière est une représentation d'un langage à partir de :

- a qui représente le singleton $\{a\}$ pour $a \in \mathcal{A}$.
- $L_1 \cdot L_2$ qui représente la *concaténation* des langages L_1 et L_2
- $L_1 + L_2$ qui représente l'union des langages L_1 et L_2
- L^* qui représente l'*itération* du langage L

Un langage est *rationnel* s'il est représentable par une expression régulière.

Expression régulière

Une expression régulière est une représentation d'un langage à partir de :

- a qui représente le singleton $\{a\}$ pour $a \in \mathcal{A}$.
- $L_1 \cdot L_2$ qui représente la *concaténation* des langages L_1 et L_2
- $L_1 + L_2$ qui représente l'union des langages L_1 et L_2
- L^* qui représente l'*itération* du langage L

Un langage est *rationnel* s'il est représentable par une expression régulière.

Expression régulière

Une expression régulière est une représentation d'un langage à partir de :

- a qui représente le singleton $\{a\}$ pour $a \in \mathcal{A}$.
- $L_1 \cdot L_2$ qui représente la *concaténation* des langages L_1 et L_2
- $L_1 + L_2$ qui représente l'union des langages L_1 et L_2
- L^* qui représente l'*itération* du langage L

Un langage est *rationnel* s'il est représentable par une expression régulière.

Expression régulière

Une expression régulière est une représentation d'un langage à partir de :

- *a* qui représente le singleton $\{a\}$ pour $a \in \mathcal{A}$.
- $L_1 \cdot L_2$ qui représente la *concaténation* des langages L_1 et L_2
- $L_1 + L_2$ qui représente l'union des langages L_1 et L_2
- L^* qui représente l'*itération* du langage L

Un langage est *rationnel* s'il est représentable par une expression régulière.

Exemple : $(ouvrir \cdot lire^{*} \cdot fermer)^{*}$

Théorème (de Kleene)

Un langage est rationnel si et seulement s'il est reconnaissable.

Exemple : $(ouvrir \cdot lire^* \cdot fermer)^*$

Théorème (de Kleene)

Un langage est rationnel si et seulement s'il est reconnaissable.

Plan

Références

- J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.
- *Handbook of Theoretical Computer Science*, volume B, 1990.
- A. Nies, *Describing Groups*, BSL vol.13, no 3, September 2007.
- S. Rubin, *Automata Presenting Structures : A Survey of the finite String Case*, BSL vol.14, no. 2, June 2008
- L. Libkin, *Elements of Finite Model Theory*, Springer, 2004.

Références

- J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.
- *Handbook of Theoretical Computer Science*, volume B, 1990.
- A. Nies, *Describing Groups*, BSL vol.13, no 3, September 2007.
- S. Rubin, *Automata Presenting Structures : A Survey of the finite String Case*, BSL vol.14, no. 2, June 2008
- L. Libkin, *Elements of Finite Model Theory*, Springer, 2004.

Références

- J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.
- *Handbook of Theoretical Computer Science*, volume B, 1990.
- A. Nies, *Describing Groups*, BSL vol.13, no 3, September 2007.
- S. Rubin, *Automata Presenting Structures : A Survey of the finite String Case*, BSL vol.14, no. 2, June 2008
- L. Libkin, *Elements of Finite Model Theory*, Springer, 2004.

Références

- J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.
- *Handbook of Theoretical Computer Science*, volume B, 1990.
- A. Nies, *Describing Groups*, BSL vol.13, no 3, September 2007.
- S. Rubin, *Automata Presenting Structures : A Survey of the finite String Case*, BSL vol.14, no. 2, June 2008
- L. Libkin, *Elements of Finite Model Theory*, Springer, 2004.

Références

- J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.
- *Handbook of Theoretical Computer Science*, volume B, 1990.
- A. Nies, *Describing Groups*, BSL vol.13, no 3, September 2007.
- S. Rubin, *Automata Presenting Structures : A Survey of the finite String Case*, BSL vol.14, no. 2, June 2008
- L. Libkin, *Elements of Finite Model Theory*, Springer, 2004.

Applications

- Compilation
- Machines à états finis (circuits, protocoles)
- Vérification formelle
- Description de systèmes (UML)
- Vérification de structures infinies
- Logique (formules sur une structure)

Applications

- Compilation
- Machines à états finis (circuits, protocoles)
- Vérification formelle
- Description de systèmes (UML)
- Vérification de structures infinies
- Logique (formules sur une structure)

Applications

- Compilation
- Machines à états finis (circuits, protocoles)
- Vérification formelle
- Description de systèmes (UML)
- Vérification de structures infinies
- Logique (formules sur une structure)

Applications

- Compilation
- Machines à états finis (circuits, protocoles)
- Vérification formelle
- Description de systèmes (UML)
- Vérification de structures infinies
- Logique (formules sur une structure)

Applications

- Compilation
- Machines à états finis (circuits, protocoles)
- Vérification formelle
- Description de systèmes (UML)
- Vérification de structures infinies
- Logique (formules sur une structure)

Applications

- Compilation
- Machines à états finis (circuits, protocoles)
- Vérification formelle
- Description de systèmes (UML)
- Vérification de structures infinies
- Logique (formules sur une structure)

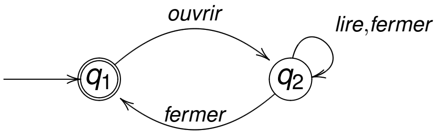
Applications

- Compilation
- Machines à états finis (circuits, protocoles)
- Vérification formelle
- Description de systèmes (UML)
- Vérification de structures infinies
- Logique (formules sur une structure)

Plan

1. Introduction
   - Définitions
   - Contexte et applications
   - Automates non-déterministes et synchrones
2. Logigue du premier ordre
   - Structures reconnaissables
   - Quelques exemples
3. Conclusion

Un automate *non-déterministe* sur l'alphabet $\mathcal{A}$ est formé de :

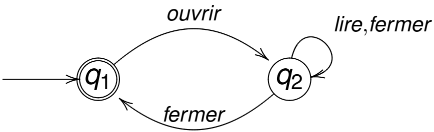


- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Une *relation* de transition $T \subseteq Q \times \mathcal{A} \times Q$
- Un ensemble d'états finals $F \subseteq Q$.

À la lecture de chaque lettre, il y maintenant possiblement *plusieurs* états suivants.

Un mot est *accepté* par un automate non-déterministe, si *au moins une* exécution se termine dans un état final.

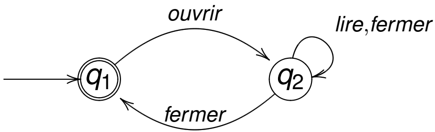
Un automate *non-déterministe* sur l'alphabet $\mathcal{A}$ est formé de :



- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Une *relation* de transition $T \subseteq Q \times \mathcal{A} \times Q$
- Un ensemble d'états finals $F \subseteq Q$.

À la lecture de chaque lettre, il y maintenant possiblement *plusieurs* états suivants.
Un mot est *accepté* par un automate non-déterministe, si *au moins une* exécution se termine dans un état final.

Un automate *non-déterministe* sur l'alphabet $\mathcal{A}$ est formé de :

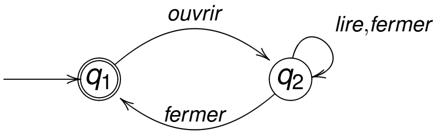


- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Une *relation* de transition $T \subseteq Q \times \mathcal{A} \times Q$
- Un ensemble d'états finals $F \subseteq Q$.

À la lecture de chaque lettre, il y maintenant possiblement *plusieurs* états suivants.

Un mot est *accepté* par un automate non-déterministe, si *au moins une* exécution se termine dans un état final.

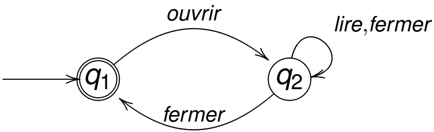
Un automate *non-déterministe* sur l'alphabet $\mathcal{A}$ est formé de :



- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Une *relation* de transition $T \subseteq Q \times \mathcal{A} \times Q$
- Un ensemble d'états finals $F \subseteq Q$.

À la lecture de chaque lettre, il y maintenant possiblement *plusieurs* états suivants.
Un mot est *accepté* par un automate non-déterministe, si *au moins une* exécution se termine dans un état final.

Un automate *non-déterministe* sur l'alphabet $\mathcal{A}$ est formé de :

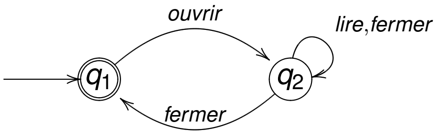


- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Une *relation* de transition $T \subseteq Q \times \mathcal{A} \times Q$
- Un ensemble d'états finals $F \subseteq Q$.

À la lecture de chaque lettre, il y maintenant possiblement *plusieurs* états suivants.

Un mot est *accepté* par un automate non-déterministe, si *au moins une* exécution se termine dans un état final.

Un automate *non-déterministe* sur l'alphabet $\mathcal{A}$ est formé de :

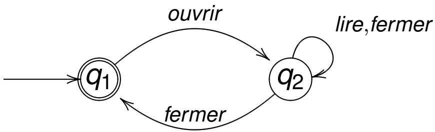


- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Une *relation* de transition $T \subseteq Q \times \mathcal{A} \times Q$
- Un ensemble d'états finals $F \subseteq Q$.

À la lecture de chaque lettre, il y maintenant possiblement *plusieurs* états suivants.

Un mot est *accepté* par un automate non-déterministe, si *au moins une* exécution se termine dans un état final.

Un automate *non-déterministe* sur l'alphabet $\mathcal{A}$ est formé de :



- Un nombre fini d'états Q
- Un ensemble d'états initiaux $I \subseteq Q$.
- Une *relation* de transition $T \subseteq Q \times \mathcal{A} \times Q$
- Un ensemble d'états finals $F \subseteq Q$.

À la lecture de chaque lettre, il y maintenant possiblement *plusieurs* états suivants.
Un mot est *accepté* par un automate non-déterministe, si *au moins une* exécution se termine dans un état final.

Déterminisation

Théorème

Un langage est reconnaissable par un automate déterministe si et seulement s'il est reconnaissable par un automate non-déterministe.

Néanmoins l'automate déterministe peut avoir un nombre d'états exponentiellement plus grand que l'automate non-déterministe.

Déterminisation

Théorème

Un langage est reconnaissable par un automate déterministe si et seulement s'il est reconnaissable par un automate non-déterministe.

Néanmoins l'automate déterministe peut avoir un nombre d'états exponentiellement plus grand que l'automate non-déterministe.

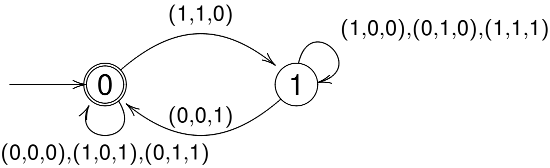
Déterminisation

Théorème

Un langage est reconnaissable par un automate déterministe si et seulement s'il est reconnaissable par un automate non-déterministe.

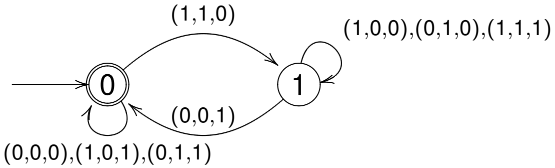
Néanmoins l'automate déterministe peut avoir un nombre d'états exponentiellement plus grand que l'automate non-déterministe.

Automates synchrones



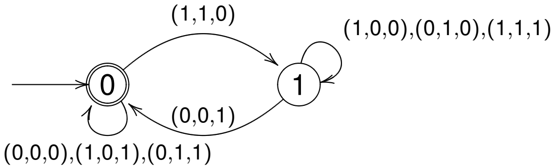
- Pour pouvoir lire des n-tuplets de mots, on considère des automates sur $\mathcal{A}^n$.
- On représente par exemple $(101, 100)$ par $(1, 1)(0, 0)(1, 0)$.
- On préfixe par un nouveau caractère $\#$ si nécessaire (0 pour l'arithmétique).

Automates synchrones



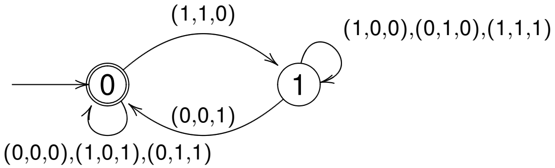
- Pour pouvoir lire des n-tuplets de mots, on considère des automates sur $\mathcal{A}^n$.
- On représente par exemple $(101, 100)$ par $(1,1)(0,0)(1,0)$.
- On préfixe par un nouveau caractère $\#$ si nécessaire (0 pour l'arithmétique).

Automates synchrones



- Pour pouvoir lire des n-tuplets de mots, on considère des automates sur $\mathcal{A}^n$.
- On représente par exemple $(101, 100)$ par $(1, 1)(0, 0)(1, 0)$.
- On préfixe par un nouveau caractère $\#$ si nécessaire (0 pour l'arithmétique).

Automates synchrones



- Pour pouvoir lire des n-tuplets de mots, on considère des automates sur $\mathcal{A}^n$.
- On représente par exemple $(101, 100)$ par $(1,1)(0,0)(1,0)$.
- On préfixe par un nouveau caractère $\#$ si nécessaire (0 pour l'arithmétique).

Plan

Structures reconnaissables

Une structure $S = \langle U; R_1, \ldots, R_n \rangle$ est *reconnaissable* ou *FA-présentable* si

- il existe un alphabet $\mathcal{A}$ et une représentation des éléments de U par des mots sur $\mathcal{A}$ tel que
  - L'image de U est reconnaissable.
  - Toutes les relations R_i (incluant $=$) sont reconnaissables.

L'arithmétique de Pressburger $\langle N, + \rangle$ est reconnaissable par automate.

Structures reconnaissables

Une structure $S = \langle U; R_1, \ldots, R_n \rangle$ est *reconnaissable* ou *FA-présentable* si

- il existe un alphabet $\mathcal{A}$ et une représentation des éléments de U par des mots sur $\mathcal{A}$ tel que
  - L'image de U est reconnaissable.
  - Toutes les relations R_i (incluant $=$) sont reconnaissables.

L'arithmétique de Pressburger $\langle N, + \rangle$ est reconnaissable par automate.

Structures reconnaissables

Une structure $S = \langle U; R_1, \ldots, R_n \rangle$ est *reconnaissable* ou *FA-présentable* si

- il existe un alphabet $\mathcal{A}$ et une représentation des éléments de U par des mots sur $\mathcal{A}$ tel que
  - L'image de U est reconnaissable.
  - Toutes les relations R_i (incluant $=$) sont reconnaissables.

L'arithmétique de Pressburger $\langle N, + \rangle$ est reconnaissable par automate.

Structures reconnaissables

Une structure $S = \langle U; R_1, \ldots, R_n \rangle$ est *reconnaissable* ou *FA-présentable* si

- il existe un alphabet $\mathcal{A}$ et une représentation des éléments de U par des mots sur $\mathcal{A}$ tel que
  - L'image de U est reconnaissable.
  - Toutes les relations R_i (incluant $=$) sont reconnaissables.

L'arithmétique de Pressburger $\langle N, + \rangle$ est reconnaissable par automate.

Relation définissables

Théorème

Les relations définissables en logique du premier-ordre sur une structure reconnaissable sont reconnaissables.

Une structure reconnaissable a donc une théorie du premier-ordre décidable.

Relation définissables

Théorème

Les relations définissables en logique du premier-ordre sur une structure reconnaissable sont reconnaissables.

Une structure reconnaissable a donc une théorie du premier-ordre décidable.

Plan

Arithmétique

- $\langle N, + \rangle$ est reconnaissable.
- (Büchi, Bruyère) En fait, une relation est reconnaissable en base k si et seulement si elle est définissable dans $\langle N, +, V_k \rangle$, où $V_k(x)$ est la plus grande puissance de k qui divise x.
- Si k et l ont des puissances communes ($k^n = l^m$, pour $n, m \in N$), V_l est définissable dans $\langle N, +, V_k \rangle$ et donc $\langle N, +, V_k \rangle = \langle N, +, V_l \rangle$.
- (RV) Sinon la multiplication est définissable dans $\langle N, +, V_k, V_l \rangle$ qui est indécidable et donc non-reconnaissable.
- (Cobham, Semenov) Une relation qui est définissable dans $\langle N, +, V_k \rangle$ et $\langle N, +, V_l \rangle$, pour k et l sans puissances communes est définissable dans $\langle N, + \rangle$.

Arithmétique

- $\langle N, +\rangle$ est reconnaissable.
- (Büchi, Bruyère) En fait, une relation est reconnaissable en base k si et seulement si elle est définissable dans $\langle N, +, V_k\rangle$, où $V_k(x)$ est la plus grande puissance de k qui divise x.
- Si k et l ont des puissances communes ($k^n = l^m$, pour $n, m \in N$), V_l est définissable dans $\langle N, +, V_k\rangle$ et donc $\langle N, +, V_k\rangle = \langle N, +, V_l\rangle$.
- (RV) Sinon la multiplication est définissable dans $\langle N, +, V_k, V_l\rangle$ qui est indécidable et donc non-reconnaissable.
- (Cobham, Semenov) Une relation qui est définissable dans $\langle N, +, V_k\rangle$ et $\langle N, +, V_l\rangle$, pour k et l sans puissances communes est définissable dans $\langle N, +\rangle$.

Arithmétique

- $\langle N, +\rangle$ est reconnaissable.
- (Büchi, Bruyère) En fait, une relation est reconnaissable en base k si et seulement si elle est définissable dans $\langle N, +, V_k\rangle$, où $V_k(x)$ est la plus grande puissance de k qui divise x.
- Si k et l ont des puissances communes ($k^n = l^m$, pour $n, m \in N$), V_l est définissable dans $\langle N, +, V_k\rangle$ et donc $\langle N, +, V_k\rangle = \langle N, +, V_l\rangle$.
- (RV) Sinon la multiplication est définissable dans $\langle N, +, V_k, V_l\rangle$ qui est indécidable et donc non-reconnaissable.
- (Cobham, Semenov) Une relation qui est définissable dans $\langle N, +, V_k\rangle$ et $\langle N, +, V_l\rangle$, pour k et l sans puissances communes est définissable dans $\langle N, +\rangle$.

Arithmétique

- $\langle N, +\rangle$ est reconnaissable.
- (Büchi, Bruyère) En fait, une relation est reconnaissable en base k si et seulement si elle est définissable dans $\langle N, +, V_k\rangle$, où $V_k(x)$ est la plus grande puissance de k qui divise x.
- Si k et l ont des puissances communes ($k^n = l^m$, pour $n, m \in N$), V_l est définissable dans $\langle N, +, V_k\rangle$ et donc $\langle N, +, V_k\rangle = \langle N, +, V_l\rangle$.
- (RV) Sinon la multiplication est définissable dans $\langle N, +, V_k, V_l\rangle$ qui est indécidable et donc non-reconnaissable.
- (Cobham, Semenov) Une relation qui est définissable dans $\langle N, +, V_k\rangle$ et $\langle N, +, V_l\rangle$, pour k et l sans puissances communes est définissable dans $\langle N, +\rangle$.

Arithmétique

- $\langle N, +\rangle$ est reconnaissable.
- (Büchi, Bruyère) En fait, une relation est reconnaissable en base k si et seulement si elle est définissable dans $\langle N, +, V_k\rangle$, où $V_k(x)$ est la plus grande puissance de k qui divise x.
- Si k et l ont des puissances communes ($k^n = l^m$, pour $n, m \in N$), V_l est définissable dans $\langle N, +, V_k\rangle$ et donc $\langle N, +, V_k\rangle = \langle N, +, V_l\rangle$.
- (RV) Sinon la multiplication est définissable dans $\langle N, +, V_k, V_l\rangle$ qui est indécidable et donc non-reconnaissable.
- (Cobham, Semenov) Une relation qui est définissable dans $\langle N, +, V_k\rangle$ et $\langle N, +, V_l\rangle$, pour k et l sans puissances communes est définissable dans $\langle N, +\rangle$.

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ^1_1-complet).
  - $\langle Q^+, \times \rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times \rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, + \rangle$ est reconnaissable ?

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ^1_1-complet).
  - $\langle Q^+, \times\rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times\rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, +\rangle$ est reconnaissable ?

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ_1^1-complet).
  - $\langle Q^+, \times \rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times \rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, + \rangle$ est reconnaissable ?

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ_1^1-complet).
  - $\langle Q^+, \times \rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times \rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, + \rangle$ est reconnaissable ?

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ_1^1-complet).
  - $\langle Q^+, \times \rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times \rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, + \rangle$ est reconnaissable ?

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ_1^1-complet).
  - $\langle Q^+, \times \rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times \rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, + \rangle$ est reconnaissable ?

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ_1^1-complet).
  - $\langle Q^+, \times \rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times \rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, + \rangle$ est reconnaissable ?

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ^1_1-complet).
  - $\langle Q^+, \times \rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times \rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, + \rangle$ est reconnaissable ?

Classification des structures reconnaissables

- (Khoussainov, Nies, Rubin, Stephan)
  - Une algèbre de Boole infinie est reconnaissable si et seulement si elle est isomorphe à $(B_{fin-cof})^n$ pour un certain n.
  - L'isomorphisme de graphes reconnaissables est indécidable (Σ_1^1-complet).
  - $\langle Q^+, \times \rangle$ n'est pas reconnaissable.
- (Oliver, Thomas) Tout groupe abélien-par-fini finiment engendré est reconnaissable.
- (Nies, Thomas) Les seuls anneaux sans diviseur de zéro (commutatifs ou non) reconnaissables sont les corps finis.
- (Blumensath, Grädel) $\langle N, \times \rangle$ n'est par reconnaissable.

(Problème ouvert) Est-ce que $\langle Q, + \rangle$ est reconnaissable ?

Conclusion

- Les automates sont des structures centrales en informatique.
- Des liens fondamentaux existent avec la logique.
  - Nous avons parlé du lien avec la logique du premier ordre.
  - Il y a aussi un lien important avec les familles de structures finies et la logique monadique du second ordre (voir livre de Libkin).
  - Il y aussi un regain d'intérêt en informatique (vérification de modèles infinis) dû au fait qu'une structure reconnaissable a une représentation finie (par les automates).

Conclusion

- Les automates sont des structures centrales en informatique.
- Des liens fondamentaux existent avec la logique.
  - Nous avons parlé du lien avec la logique du premier ordre.
  - Il y a aussi un lien important avec les familles de structures finies et la logique monadique du second ordre (voir livre de Libkin).
  - Il y aussi un regain d'intérêt en informatique (vérification de modèles infinis) dû au fait qu'une structure reconnaissable a une représentation finie (par les automates).

Conclusion

- Les automates sont des structures centrales en informatique.
- Des liens fondamentaux existent avec la logique.
    - Nous avons parlé du lien avec la logique du premier ordre.
    - Il y a aussi un lien important avec les familles de structures finies et la logique monadique du second ordre (voir livre de Libkin).
    - Il y aussi un regain d'intérêt en informatique (vérification de modèles infinis) dû au fait qu'une structure reconnaissable a une représentation finie (par les automates).

Conclusion

- Les automates sont des structures centrales en informatique.
- Des liens fondamentaux existent avec la logique.
  - Nous avons parlé du lien avec la logique du premier ordre.
  - Il y a aussi un lien important avec les familles de structures finies et la logique monadique du second ordre (voir livre de Libkin).
  - Il y aussi un regain d'intérêt en informatique (vérification de modèles infinis) dû au fait qu'une structure reconnaissable a une représentation finie (par les automates).

Conclusion

- Les automates sont des structures centrales en informatique.
- Des liens fondamentaux existent avec la logique.
  - Nous avons parlé du lien avec la logique du premier ordre.
  - Il y a aussi un lien important avec les familles de structures finies et la logique monadique du second ordre (voir livre de Libkin).
  - Il y aussi un regain d'intérêt en informatique (vérification de modèles infinis) dû au fait qu'une structure reconnaissable a une représentation finie (par les automates).