The iteration constraint in the algebra of trees

Alain Colmerauer
and
Thi-Bich-Hanh Dao

Laboratoire d'Informatique de Marseille
CNRS et Université de Provence et
Université de la Méditerranée
France

This work has been inspired by a paper of Pawel Mielniczuk from the University of Wroclaw, Poland.

Introduction

• The n-iterated of the first-order formula $P(x,y)$ is written an defined by:

$$P^n(x,y) \stackrel{\text{def}}{=} \left[\begin{array}{l} \exists u_0 \ldots \exists u_n \\ x = u_0 \wedge \\ P(u_0,u_1) \wedge \\ P(u_1,u_2) \wedge \\ \ldots \\ P(u_{n-1},u_n) \wedge \\ u_n = y \end{array}\right]$$

• Let $\alpha(n)$ be the integer defined by

$$\alpha(n) \stackrel{\text{def}}{=} \underbrace{2^{\cdot^{\cdot^{\cdot}\left(2^{\left(2^2\right)}\right)}}}_{n}$$

that is,

$$\alpha(0) = 1, \quad \alpha(n+1) = 2^{\alpha(n)}.$$

• We will show how to express $P^{\alpha(n)-1}(x,y)$ by a formula, equivalent in the algebra of trees, whose size is of the form $a + bn$, where a, b do not depend on n. This result has consequences with respect to the expressiveness and the complexity of constraints in the theory of trees.

• **Notice** that $\alpha(5) = 2^{65536}$. Thus $\alpha(5)$ is greater than 10^{20000}, a number probably much greater than the number of atoms of the universe or the number of nanoseconds which elapsed since its creation!

Tree constraints

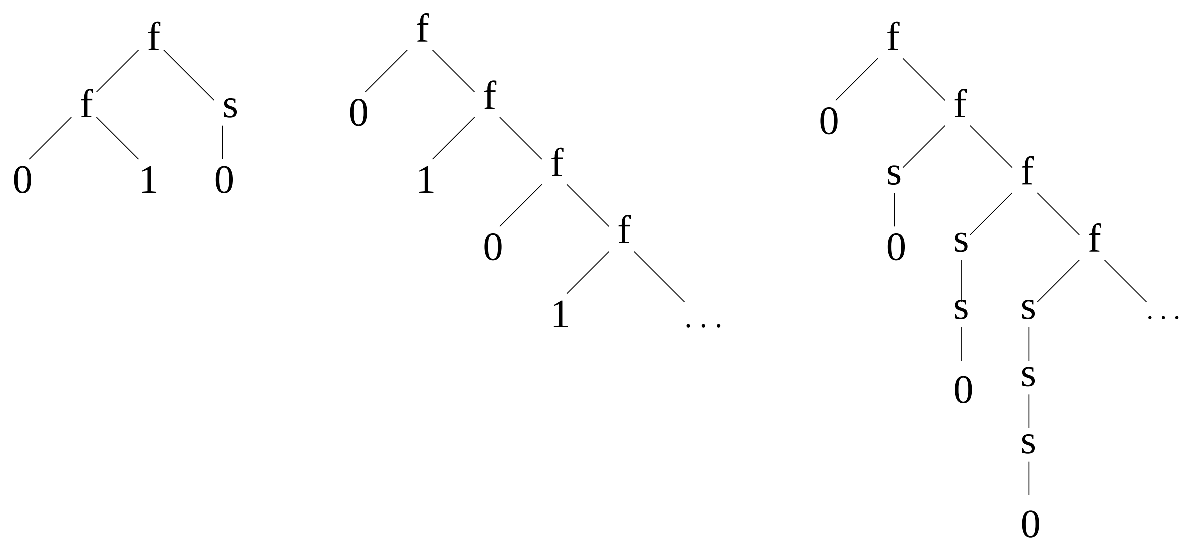
- **Syntax** First order formulae constructed on the alphabet

$$V \cup F \cup \{=, \mathit{true}, \mathit{false}, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, \exists, \forall\},$$

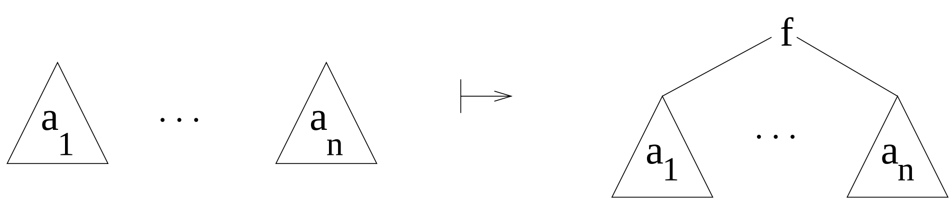
where V is an infinite set of variables and F an infinite set of functionnal symbols.

- **Semantics** Algebra of trees

The domain is the set of all trees, possibly infinite, whose nodes all labelled by elements of F, like



The operation linked to any n-ary functional symbol f is the mapping



Examples of tree constraints

- Example of a tree constraint $P(u, v, w, x, y)$

$$\exists z \begin{bmatrix} z = f(x, f(y, f(x, f(y, f(x, k))))) \; \wedge \\ z = f(u, f(v, f(w, f(u, f(v, k))))) \end{bmatrix}$$

- Solving the preceeding constraint we obtain

$$u = x \;\wedge\; v = x \;\wedge\; w = x \;\wedge\; y = x$$

- Example of a constraint $Q(u, v, w, x, y)$, explicitly involving infinite trees

$$\exists z \begin{bmatrix} z = f(x, f(y, z)) \; \wedge \\ z = f(u, f(v, f(w, z))) \end{bmatrix}$$

- Solving the preceeding constraint we also obtain

$$u = x \;\wedge\; v = x \;\wedge\; w = x \;\wedge\; y = x$$

Iteration theorem

• **Definition** For any formula $P(x,y)$ and any $k \geq 0$, we introduce the tree constraint:

$$\textit{iterated}_k(x,y) \stackrel{\text{def}}{=} \exists z\ \textit{triangle}_k(3,x,z,y)$$

with,

$$\textit{triangle}_0(t,x,z,y) \stackrel{\text{def}}{=} z = x \wedge z = y$$

$$\textit{triangle}_{k+1}(t,x,z,y) \stackrel{\text{def}}{=} \left[\begin{array}{l} [\exists u_1\, \exists u_2\, z = f(x,u_1,u_2,y)] \\ \wedge \\ \left[\begin{array}{l} \forall t'\, \forall y'\, \forall z' \\ \left[\begin{array}{l} (t' = 1 \vee t' = 2) \wedge \\ \textit{triangle}_k(t',z,y',z') \end{array}\right] \rightarrow \\ \left[\begin{array}{l} (t' = 1 \wedge \textit{form1}(z')) \vee \\ (t' = 2 \wedge \left[\begin{array}{l} \exists u\, \exists v\ \textit{form2}(u,z',v) \wedge \\ (t = 1 \rightarrow \textit{son}(u,v)) \wedge \\ (t = 2 \rightarrow \textit{son}(u,v) \vee u = v) \wedge \\ (t = 3 \rightarrow P(u,v)) \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right]$$

and

$$\textit{form1}(x) \stackrel{\text{def}}{=} \exists u_1 \ldots \exists u_4\, x = f(u_1, f(u_2,u_2,u_2,u_2), f(u_3,u_3,u_3,u_3), u_4)$$

$$\textit{form2}(x,z,y) \stackrel{\text{def}}{=} \exists u_1 \ldots \exists u_6\, z = f(u_1, f(u_1,u_2,u_3,x), f(y,u_4,u_5,u_6), u_6)$$

$$\textit{son}(x,y) \stackrel{\text{def}}{=} \exists u_1 \ldots \exists u_4\, x = f(u_1,u_2,u_3,u_4) \wedge (y = u_2 \vee y = u_3)$$

• **Theorem**

1. $\textit{iterated}_k(x,y) \leftrightarrow P^{\alpha(k)-1}(x,y)$,
2. $|\textit{iterated}_k(x,y)| = 9 + k(155 + |P(x,y)|)$.

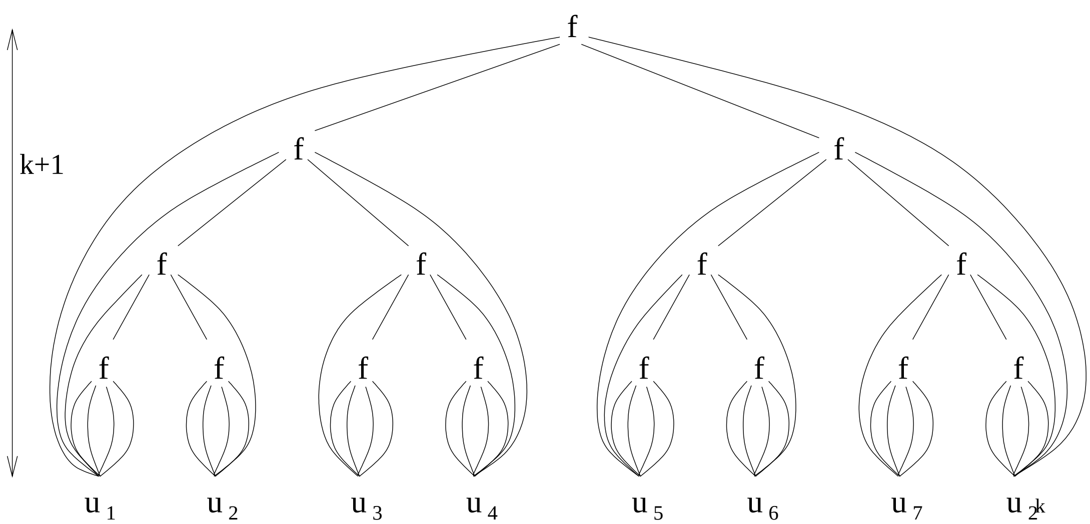
Onion tree

- **Definition** For all $k \geq 1$,

$$
\mathit{onion}_k(z) \stackrel{\text{def}}{=}
\left[
\begin{array}{l}
\left[\exists u_1 \ldots \exists u_4\, z = f(u_1, u_2, u_3, u_4)\right] \\
\wedge \\
\left[
\begin{array}{l}
\forall z' \\
\mathit{son}^{k-1}(z, z') \rightarrow \\
\mathit{form1}(z')
\end{array}
\right] \\
\wedge \\
\left[
\begin{array}{l}
\forall z' \\
\left[\bigvee_{i=0}^{k-1} \mathit{son}^{i}(z, z')\right] \rightarrow \\
\left[\exists u \exists v\, \mathit{form2}(u, z', v)\right]
\end{array}
\right]
\end{array}
\right]
$$

- Tree x such that *form1* (x) and trees x, y, z such that *form2* (x, z, y)

- **Example** Onion tree for $k = 3$

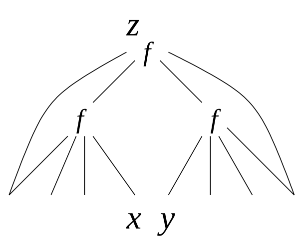


Onion lemma

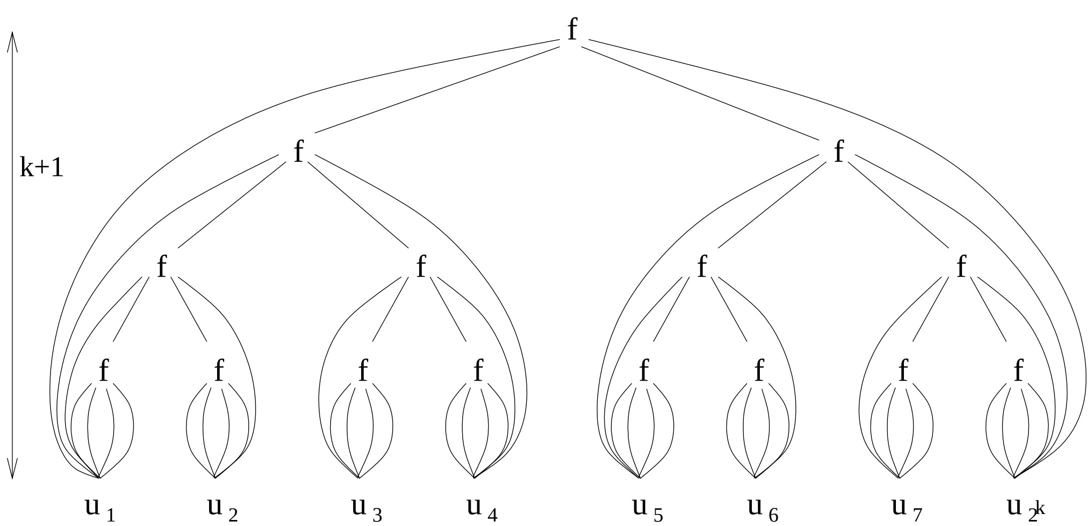
- **Lemma** For any constraint $P(x,y)$,

$$P^{2^k-1}(x,y) \;\leftrightarrow\; \exists z \left[\begin{array}{l} \textit{onion}_k(z) \\ \wedge \\ \left[\exists u_2 \exists u_3\, z = f(x, u_2, u_3, y)\right] \\ \wedge \\ \left[\begin{array}{l} \forall z' \\ \left[\bigvee_{i=0}^{k-1} \textit{son}^{\,i}(z,z')\right] \rightarrow \\ \\ \left[\begin{array}{l} \exists u \exists v\ \textit{form2}\,(u,z',v) \wedge \\ P(u,v) \end{array}\right] \end{array}\right] \end{array}\right]$$

- Trees x, y, z such that *form2* (x,z,y)



- Onion tree for $k=3$



Proof of the iteration theorem

• Given the definition of $\textit{iterated}_k$, it is suffficient to prove the last of the three equivalences:

$$
\begin{aligned}
(\exists z \; \textit{\textbf{triangle}}_k(1,x,z,y)) &\leftrightarrow \textit{\textbf{son}}^{\alpha(k)-1}(x,y)\\
(\exists z \; \textit{\textbf{triangle}}_k(2,x,z,y)) &\leftrightarrow \bigvee_{i=0}^{i=\alpha(k)-1} \textit{\textbf{son}}^{i}(x,y)\\
(\exists z \; \textit{\textbf{triangle}}_k(3,x,z,y)) &\leftrightarrow P^{\alpha(k)-1}(x,y)
\end{aligned}
$$

• By induction k, we will prove the three equivalences. The three equivalences hold for $k = 0$. Assuming that they hold for a given $k \geq 0$, let us prove that they hold for $k+1$.

By introducing an existential quantifiquation on z in both side of the definition of $\textit{triangle}_{k+1}$ and by splitting up the $t' = 1$ case from the $t' = 2$ case, we get

$$
\exists z \; \textit{\textbf{triangle}}_{k+1}(t,x,z,y) \leftrightarrow \exists z
\begin{bmatrix}
[\exists u_1 \, \exists u_2 \; z = f(x,u_1,u_2,y)] \\
\wedge \\
\begin{bmatrix}
\forall z' \\
(\exists y' \; \textit{\textbf{triangle}}_k(1,z,y',z')) \rightarrow \\
\textit{\textbf{form1}}(z')
\end{bmatrix} \\
\wedge \\
\begin{bmatrix}
\forall z' \\
(\exists y' \; \textit{\textbf{triangle}}_k(2,z,y',z')) \rightarrow \\
\begin{bmatrix}
\exists u \, \exists v \; \textit{\textbf{form2}}(u,z',v) \wedge \\
Q(u,v)
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

with

$$
Q(u,v) \stackrel{\text{def}}{=}
\begin{bmatrix}
(t=1 \rightarrow \textit{\textbf{son}}(u,v)) \wedge \\
(t=2 \rightarrow \textit{\textbf{son}}(u,v) \vee u=v) \wedge \\
(t=3 \rightarrow P(u,v))
\end{bmatrix}
$$

Proof of the iteration theorem, next

- Since we have have assumed the three initial equivalences to hold for k, it follows that

$$\exists z\, \textit{triangle}_{k+1}(t,x,z,y) \;\leftrightarrow\; \exists z \left[\begin{array}{l} [\exists u_2 \exists u_3\, z = f(x,u_2,u_3,y)] \\ \wedge \\ \left[\begin{array}{l} \forall z' \\ \textit{son}^{\alpha(k)-1}(z,z') \to \\ \textit{form1}(z') \end{array}\right] \\ \wedge \\ \left[\begin{array}{l} \forall z' \\ \left[\bigvee_{i=0}^{\alpha(k)-1} \textit{son}^{i}(z,z')\right] \to \\ \left[\begin{array}{l} \exists u\, \exists v\, \textit{form2}(u,z',v) \wedge \\ Q(u,v) \end{array}\right] \end{array}\right] \end{array}\right]$$

- that is, by looking at the definition of $\textit{onion}_k$,

$$\exists z\, \textit{triangle}_{k+1}(t,x,z,y) \;\leftrightarrow\; \exists z \left[\begin{array}{l} \textit{onion}_k(z) \\ \wedge \\ [\exists u_2 \exists u_3\, z = f(x,u_2,u_3,y)] \\ \wedge \\ \left[\begin{array}{l} \forall z' \\ \left[\bigvee_{i=0}^{\alpha(k)-1} \textit{son}^{i}(z,z')\right] \to \\ \\ \left[\begin{array}{l} \exists u \exists v\, \textit{form2}(u,z',v) \wedge \\ Q(u,v) \end{array}\right] \end{array}\right] \end{array}\right] \quad (1)$$

Proof of the iteration theorem, end

- By the onion lemma with Q instead of P, we then get

$$\exists z\, \textit{triangle}_{k+1}(t, x, y, z) \;\leftrightarrow\; Q^{\alpha(k+1)-1}(u, v)$$

- and by giving successively the values $1, 2, 3$ to t and going back to the definition of Q, we obtain

$$(\exists z\; \textit{triangle}_k(1, x, z, y)) \;\leftrightarrow\; \textit{son}^{\alpha(k)-1}(x, y)$$
$$(\exists z\; \textit{triangle}_k(2, x, z, y)) \;\leftrightarrow\; \vee_{i=0}^{i=\alpha(k)-1}\, \textit{son}^{i}(x, y)$$
$$(\exists z\; \textit{triangle}_k(3, x, z, y)) \;\leftrightarrow\; P^{\alpha(k)-1}(x, y)$$

Huge finite tree

• If we define

$$\textit{huge}_k(x) \stackrel{\text{def}}{=} \textit{iterated}_k(x, 0),$$

with

$$P(x, y) \stackrel{\text{def}}{=} x = s(y),$$

• according to the iteration theorem, we then have

1. $\textit{huge}_k(x) \leftrightarrow x = s^{\alpha(k)-1}(0)$,
2. $|\textit{huge}_k(x)| = 9 + 159k$.

Toward a multiplicative constraint

• Consider the Prolog like program

$$\left\{\begin{array}{l} \mathit{Times}(0,v,0) \leftarrow \mathit{Integer}(v), \\ \mathit{Times}(s(u),v,w') \leftarrow \mathit{Times}(u,v,w) \wedge \mathit{Plus}(v,w,w'), \\ \mathit{Plus}(0,v,v) \leftarrow \mathit{True}, \\ \mathit{Plus}(s(u),v,s(w)) \leftarrow \mathit{Plus}(u,v,w), \\ \mathit{Integer}(0) \leftarrow \mathit{True}, \\ \mathit{Integer}(s(u)) \leftarrow \mathit{Integer}(u) \end{array}\right\}$$

which for a query of the form $\mathit{Times}(u,v,w)$ would enumerate the contraints of la form

$$C_m^n(u,v,w) \quad := \quad u = s^m(0) \wedge v = s^n(0) \wedge w = s^{m\times n}(0)),$$

ad infinitum, without forgetting any.

• By analyzing the execution of the program we conclude that the constraint

$$\exists x\, \exists y\, x = \mathit{sq}(\mathit{times}(u,v,w), \mathit{nil}) \ \wedge\ P^k(x,y) \ \wedge\ y = \mathit{nil}$$

est équivalent to the conjunction of the constraints $C_m^n(u,v,w)$ with

$$mn + 2m + n + 2 \leq k$$

where $P(x,y)$, is the transition constraint of the Prolog machine which corresponds to the program.

Transition constraint of the Prolog machine

- Program

$$\left\{\begin{array}{l} \mathit{Times}\,(0, v, 0) \leftarrow \mathit{Integer}\,(v), \\ \mathit{Times}\,(s(u), v, w') \leftarrow \mathit{Times}\,(u, v, w) \wedge \mathit{Plus}\,(v, w, w'), \\ \mathit{Plus}\,(0, v, v) \leftarrow \mathit{True}, \\ \mathit{Plus}\,(s(u), v, s(w)) \leftarrow \mathit{Plus}\,(u, v, w), \\ \mathit{Integer}\,(0) \leftarrow \mathit{True}, \\ \mathit{Integer}\,(s(u)) \leftarrow \mathit{Integer}\,(u) \end{array}\right\}$$

- Transition constraint

$$P(x,y) \stackrel{\text{def}}{=} \left[\begin{array}{l} x = y \vee \\ \left[\begin{array}{l} \exists v\, \exists z \\ x = \mathit{sq}\,(\mathit{times}\,(0, v, 0),\ z)) \wedge \\ y = z \end{array}\right] \vee \\ \left[\begin{array}{l} \exists u\, \exists v\, \exists w\, \exists w'\, \exists z \\ x = \mathit{sq}\,(\mathit{times}\,(s(u), v, w'),\ z) \wedge \\ y = \mathit{sq}\,(\mathit{times}\,(u, v, w),\ \mathit{sq}\,(\mathit{plus}\,(v, w, w'),\ z)) \end{array}\right] \vee \\ \left[\begin{array}{l} \exists v\, \exists z \\ x = \mathit{sq}\,(\mathit{plus}\,(0, v, v),\ z) \wedge \\ y = z \end{array}\right] \vee \\ \left[\begin{array}{l} \exists u\, \exists v\, \exists w\, \exists z \\ x = \mathit{sq}\,(\mathit{plus}\,(s(u), v, s(w)),\ z) \wedge \\ y = \mathit{sq}\,(\mathit{plus}\,(u, v, w),\ z) \end{array}\right] \vee \\ \left[\begin{array}{l} \exists v\, \exists z \\ x = \mathit{sq}\,(\mathit{integer}\,(0),\ z) \wedge \\ y = z \end{array}\right] \vee \\ \left[\begin{array}{l} \exists u\, \exists v\, \exists w\, \exists z \\ x = \mathit{sq}\,(\mathit{integer}\,(s(u)),\ z) \wedge \\ y = \mathit{sq}\,(\mathit{integer}\,(u),\ z) \end{array}\right] \end{array}\right]$$

Multiplicative constraint

• The set of solutions of the constraint

$$\mathit{Times}'(u,v,w) \ := \ \begin{bmatrix} \exists x\, \exists y \\ x = \mathit{sq}\,(\mathit{times}\,(u,v,w), \mathit{nil}\,) \wedge \\ \mathit{iterated}_{\,5}(x,y) \wedge \\ y = \mathit{nil} \end{bmatrix}$$

is a finite set of tuples of the form

$$(s^m(0),\ s^n(0), s^{m\times n}(0))$$

which includes all the cases, when m and n range from 0 to the number of atoms of the universe.

• The size of the constraint $\mathit{Times}'(u,v,w)$ is less then 2000 occurences of symbols.

Turing machine

• Instead of a Prolog machine we can take a Turing machine M, and express by $P(x, y)$ the fact that $x = y$ or that M can move from configuration x to configuration y by executing one instruction. For example, if the set of instructions of M is

$$\begin{Bmatrix} (q_0, 1, 1, q_1, D), \\ (q_0, \sqcup, 1, q_3, G), \\ (q_1, 1, 1, q_0, D), \\ (q_1, \sqcup, \sqcup, q_3, G) \end{Bmatrix}$$

we can take

$$P(x,y) \stackrel{\text{def}}{=} \begin{bmatrix} x = y \lor \\ \begin{bmatrix} \exists u\, \exists v\, \exists w \\ x = q_0(u, \boldsymbol{sq}(1, \boldsymbol{sq}(v, w))) \land \\ y = q_1(\boldsymbol{sq}(u, 1), \boldsymbol{sq}(v, w)) \end{bmatrix} \lor \\ \begin{bmatrix} \exists u\, \exists v\, \exists w \\ x = q_0(\boldsymbol{sq}(u, v), \boldsymbol{sq}(\sqcup, w)) \land \\ y = q_3(u, \boldsymbol{sq}(v, \boldsymbol{sq}(1, w))) \end{bmatrix} \lor \\ \begin{bmatrix} \exists u\, \exists v\, \exists w \\ x = q_1(u, \boldsymbol{sq}(1, \boldsymbol{sq}(v, w)) \land \\ y = q_0(\boldsymbol{sq}(u, 1), \boldsymbol{sq}(v, w)) \end{bmatrix} \lor \\ \begin{bmatrix} \exists u\, \exists v\, \exists w \\ x = q_1(\boldsymbol{sq}(u, v), \boldsymbol{sq}(\sqcup, w)) \land \\ y = q_3(u, \boldsymbol{sq}(v, \boldsymbol{sq}(\sqcup, w))) \end{bmatrix} \end{bmatrix}$$

Quasi-universality of tree constraints

• **Definition** For any word $a := a_1 a_2 \dots a_n$, without the space character $\sqcup$, let $\tau(a)$ be the only tree x such that

$$\exists w \, x = \textit{\textbf{sq}}(u, a_1, \textit{\textbf{sq}}(a_2, \dots \textit{\textbf{sq}}(a_n, w) \dots) \wedge w = \textit{\textbf{sq}}(\sqcup, w))$$

Call a *right-half-tape* any tree of the form $\tau(a)$.

• **Property** Let M be a well defined Turing machine, whose transition constraint is $P(x, y)$. The set of solutions of the constraint

$$\textit{\textbf{machine}}(u, v) \; := \; \begin{bmatrix} \exists x \, \exists y \, \exists w \\ x = q_0(w, u) \wedge \\ \textit{\textbf{iterated}}_5(x, y) \wedge \\ y = q_n(w, v) \wedge \\ w = \textit{\textbf{sq}}(w, \sqcup) \end{bmatrix},$$

whith u restricted to be a right-half-tape, consists in all the pairs of the form

$$(\tau(a), \tau(b))$$

such that, with input a, the machine M outputs b in less then $\alpha(5)$ steps.