SAARLAND UNIVERSITY
COMPUTER SCIENCE

Undecidability of the Post Correspondence Problem in Coq

Bachelor Talk

Edith Heiter
Advisors: Prof. Dr. Gert Smolka, Yannick Forster

August 23, 2017

What to Expect?

- Formalized decision problems:
  - Post correspondence problem (PCP)
  - modified Post correspondence problem (MPCP)
  - word problem in string-rewriting systems
  - halting problem for Turing machines
- Formal definition and verification of reductions from the literature proving PCP undecidable:
  - Hopcroft et al. (2006)
  - Davis et al. (1994)
  - Wim H. Hesselink (2015)
- constructive Coq development

The Post Correspondence Problem

print	*dog*	*eats*
sprint	*doge*	*at*

Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance P of type $\texttt{pcp} := \mathbf{L}\,(\Sigma^* \times \Sigma^*)$

The Post Correspondence Problem

print	*dog*	*eats*
sprint	*doge*	*at*

dog	*eats*	*print*
doge	*at*	*sprint*

Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance P of type $\mathtt{pcp} := \mathbf{L}\,(\Sigma^* \times \Sigma^*)$

The Post Correspondence Problem

$\frac{\textit{print}}{\textit{sprint}}$ $\frac{\textit{dog}}{\textit{doge}}$ $\frac{\textit{eats}}{\textit{at}}$

$\frac{\textit{dog}}{\textit{doge}}$ $\frac{\textit{eats}}{\textit{at}}$ $\frac{\textit{print}}{\textit{sprint}}$

$\frac{\textit{dogeatsprint}}{\textit{dogeatsprint}}$

Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance P of type $\texttt{pcp} := \mathbf{L}\,(\Sigma^* \times \Sigma^*)$

The Post Correspondence Problem

$$\boxed{\frac{print}{sprint}}\ \boxed{\frac{dog}{doge}}\ \boxed{\frac{eats}{at}}$$

$$\boxed{\frac{dog}{doge}}\boxed{\frac{eats}{at}}\boxed{\frac{print}{sprint}}$$

$$\frac{dogeatsprint}{dogeatsprint}$$

Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance P of type $\mathsf{pcp} := \mathbf{L}\,(\Sigma^* \times \Sigma^*)$
- S is a match if
  $\mathsf{concat}\,(\mathsf{map}\,\pi_1\,S) = \mathsf{concat}\,(\mathsf{map}\,\pi_2\,S)$,
  abbreviated as $C_1\,S = C_2\,S$
- S is a match for P if $S \neq [\,]$, $S \subseteq P$, and S is a match

The Post Correspondence Problem

$\frac{print}{sprint}$ $\frac{dog}{doge}$ $\frac{eats}{at}$

$\frac{dog}{doge}$ $\frac{eats}{at}$ $\frac{print}{sprint}$

$\frac{dogeatsprint}{dogeatsprint}$

Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance P of type $\mathsf{pcp} := \mathbf{L}\,(\Sigma^* \times \Sigma^*)$
- S is a match if $\mathsf{concat}\,(\mathsf{map}\,\pi_1\,S) = \mathsf{concat}\,(\mathsf{map}\,\pi_2\,S)$, abbreviated as $C_1\,S = C_2\,S$
- S is a match for P if $S \neq [\,]$, $S \subseteq P$, and S is a match

Definition (Post correspondence problem)

$\mathsf{PCP}\,P := \exists S.\, S$ is a match for P

The Modified Post Correspondence Problem

$\boxed{\frac{\textit{print}}{\textit{sprint}}}$ $\boxed{\frac{\textit{dog}}{\textit{doge}}}$ $\boxed{\frac{\textit{eats}}{\textit{at}}}$

Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance (d, P) of type $\mathsf{mpcp} := (\Sigma^* \times \Sigma^*) \times \mathsf{pcp}$

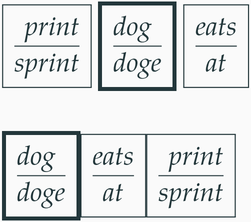
The Modified Post Correspondence Problem

$\frac{\textit{print}}{\textit{sprint}}$ $\boxed{\frac{\textit{dog}}{\textit{doge}}}$ $\frac{\textit{eats}}{\textit{at}}$

Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance (d, P) of type $\mathsf{mpcp} := (\Sigma^* \times \Sigma^*) \times \mathsf{pcp}$

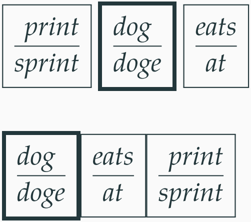
The Modified Post Correspondence Problem



Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance (d, P) of type $\mathsf{mpcp} := (\Sigma^* \times \Sigma^*) \times \mathsf{pcp}$

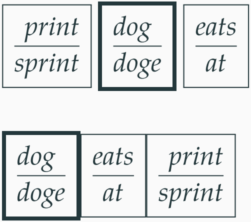
The Modified Post Correspondence Problem



Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance (d, P) of type $\mathsf{mpcp} := (\Sigma^* \times \Sigma^*) \times \mathsf{pcp}$
- S is a match if $C_1\,S = C_2\,S$
- S is a match for P if $S \neq [\,]$, $S \subseteq P$, and S is a match

The Modified Post Correspondence Problem



Assume a fixed alphabet Σ.

- strings $\Sigma^* := \mathbf{L}\,\Sigma$
- instance (d, P) of type $\mathsf{mpcp} := (\Sigma^* \times \Sigma^*) \times \mathsf{pcp}$
- S is a match if $C_1\,S = C_2\,S$
- S is a match for P if $S \neq [\,]$, $S \subseteq P$, and S is a match

Definition (Modified Post correspondence problem)

$\mathsf{MPCP}\,(d, P) := \exists S.\,(d :: S)$ is a match for $(d :: P)$

Undecidability in Coq

Definition (Undecidability)

A class $P : X \to \mathbb{P}$ is undecidable if the halting problem (Halt) reduces to P.

Undecidability in Coq

Definition (Undecidability)

A class $P : X \to \mathbb{P}$ is undecidable if the halting problem (Halt) reduces to P.

Definition (Reduction)

Let $P : X \to \mathbb{P}$ and $Q : Y \to \mathbb{P}$ be two classes. A reduction of P to Q is a function $f : X \to Y$ such that $\forall x.\, P\,x \leftrightarrow Q\,(f\,x)$.

Undecidability in Coq

Definition (Undecidability)

A class $P : X \to \mathbb{P}$ is undecidable if the halting problem (Halt) reduces to P.

Definition (Reduction)

Let $P : X \to \mathbb{P}$ and $Q : Y \to \mathbb{P}$ be two classes. A reduction of P to Q is a function $f : X \to Y$ such that $\forall x.\, P\, x \leftrightarrow Q\,(f\, x)$.

Hopcroft et al. (2006)

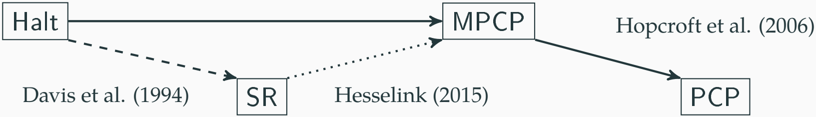
Undecidability in Coq

Definition (Undecidability)

A class $P : X \to \mathbb{P}$ is undecidable if the halting problem (Halt) reduces to P.

Definition (Reduction)

Let $P : X \to \mathbb{P}$ and $Q : Y \to \mathbb{P}$ be two classes. A reduction of P to Q is a function $f : X \to Y$ such that $\forall x. P\, x \leftrightarrow Q\, (f\, x)$.



Undecidability in Coq

Definition (Undecidability)

A class $P : X \to \mathbb{P}$ is undecidable if the halting problem (Halt) reduces to P.

Definition (Reduction)

Let $P : X \to \mathbb{P}$ and $Q : Y \to \mathbb{P}$ be two classes. A reduction of P to Q is a function $f : X \to Y$ such that $\forall x. P\, x \leftrightarrow Q\, (f\, x)$.

String-Rewriting Systems

$\Sigma := \{a, b\}$	finite alphabet of symbols
$R := \{ab/ba, aa/ab\}$	finite set of rewrite rules

String-Rewriting Systems

$\Sigma := \{a, b\}$ finite alphabet of symbols

$R := \{ab/ba, aa/ab\}$ finite set of rewrite rules

$a\mathbf{ab} \Rightarrow_R a\mathbf{ba}$

$$\frac{u/v \in R}{xuy \Rightarrow_R xvy}$$

String-Rewriting Systems

$\Sigma := \{a, b\}$ finite alphabet of symbols

$R := \{ab/ba, aa/ab\}$ finite set of rewrite rules

$a\mathbf{ab} \Rightarrow_R a\mathbf{ba}$

$aab \Rightarrow_R^* bab$

$$\frac{u/v \in R}{xuy \Rightarrow_R xvy} \qquad \frac{}{z \Rightarrow_R^* z} \qquad \frac{x \Rightarrow_R y \quad y \Rightarrow_R^* z}{x \Rightarrow_R^* z}$$

String-Rewriting Systems

$\Sigma := \{a, b\}$ finite alphabet of symbols

$R := \{ab/ba, aa/ab\}$ finite set of rewrite rules

$a\mathbf{ab} \Rightarrow_R a\mathbf{ba}$

$aab \Rightarrow_R^* bab$

$$\frac{u/v \in R}{xuy \Rightarrow_R xvy} \qquad \frac{}{z \Rightarrow_R^* z} \qquad \frac{x \Rightarrow_R y \quad y \Rightarrow_R^* z}{x \Rightarrow_R^* z}$$

Definition: Word problem in string-rewriting systems

$\mathsf{SR}\,(R, x, y) := x \Rightarrow_R^* y$

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$$

$$\boxed{\frac{\$}{\$aab\star}}$$

$$\frac{\$}{\$aab\star}$$

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$$

$\left[\frac{\$}{\$aab\star}\right]$ $\left[\frac{a}{a}\right]$

$\frac{\$}{\$aab\star}$

- *copy dominoes* transfer unchanged symbols to the next string

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$$

$$\boxed{\frac{\$}{\$aab\star}} \quad \boxed{\frac{a}{a}} \quad \boxed{\frac{ab}{ba}}$$

$$\frac{\$}{\$aab\star}$$

- *copy dominoes* transfer unchanged symbols to the next string
- *rewrite dominoes* simulate a single rewrite

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$$

$\frac{\$}{\$aab\star}$ $\frac{a}{a}$ $\frac{ab}{ba}$ $\frac{\star}{\star}$

$$\frac{\$aab\star}{\$aab \star aba\star}$$

- *copy dominoes* transfer unchanged symbols to the next string
- *rewrite dominoes* simulate a single rewrite
- consecutive strings are separated by $\star$

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$$

$$\left[\frac{\$}{\$aab\star}\right] \left[\frac{a}{a}\right] \left[\frac{ab}{ba}\right] \left[\frac{\star}{\star}\right] \left[\frac{ab}{ba}\right] \left[\frac{a}{a}\right] \left[\frac{\star}{\star}\right]$$

$$\frac{\$aab \star aba\star}{\$aab \star aba \star baa\star}$$

- *copy dominoes* transfer unchanged symbols to the next string
- *rewrite dominoes* simulate a single rewrite
- consecutive strings are separated by $\star$

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$$

$$\left[\frac{\$}{\$aab\star}\right] \left[\frac{a}{a}\right] \left[\frac{ab}{ba}\right] \left[\frac{\star}{\star}\right] \left[\frac{ab}{ba}\right] \left[\frac{a}{a}\right] \left[\frac{\star}{\star}\right] \left[\frac{b}{b}\right] \left[\frac{aa}{ab}\right] \left[\frac{\star}{\star}\right]$$

$$\frac{\$aab \star aba \star baa\star}{\$aab \star aba \star baa \star bab\star}$$

- *copy dominoes* transfer unchanged symbols to the next string
- *rewrite dominoes* simulate a single rewrite
- consecutive strings are separated by $\star$

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$$

$$\boxed{\frac{\$}{\$aab\star}}\ \frac{a}{a}\ \frac{ab}{ba}\ \frac{\star}{\star}\ \frac{ab}{ba}\ \frac{a}{a}\ \frac{\star}{\star}\ \frac{b}{b}\ \frac{aa}{ab}\ \frac{\star}{\star}\ \frac{bab\star\$}{\$}$$

$$\frac{\$aab \star aba \star baa \star bab \star \$}{\$aab \star aba \star baa \star bab \star \$}$$

- *copy dominoes* transfer unchanged symbols to the next string
- *rewrite dominoes* simulate a single rewrite
- consecutive strings are separated by $\star$

Reducing String Rewriting to MPCP

Word problem $aab \Rightarrow_R^* bab$ with $R = \{ab/ba, aa/ab\}$

$$aab \Rightarrow aba \Rightarrow baa \Rightarrow bab$$

$$\left[\frac{\$}{\$aab\star}\right]\left[\frac{a}{a}\right]\left[\frac{ab}{ba}\right]\left[\frac{\star}{\star}\right]\left[\frac{ab}{ba}\right]\left[\frac{a}{a}\right]\left[\frac{\star}{\star}\right]\left[\frac{b}{b}\right]\left[\frac{aa}{ab}\right]\left[\frac{\star}{\star}\right]\left[\frac{bab\star\$}{\$}\right]$$

$$\frac{\$aab \star aba \star baa \star bab \star \$}{\$aab \star aba \star baa \star bab \star \$}$$

- *copy dominoes* transfer unchanged symbols to the next string
- *rewrite dominoes* simulate a single rewrite
- consecutive strings are separated by $\star$

$$f(R, x, y) := \left\{ \left[\frac{\$}{\$x\star}\right], \left[\frac{y\star\$}{\$}\right], \left[\frac{\star}{\star}\right] \right\} \cup \left\{ \left[\frac{a}{a}\right] \middle| a : \Sigma \right\} \cup \left\{ \left[\frac{u}{v}\right] \middle| u/v \in R \right\}$$

Correctness Proof $x \Rightarrow_R^* y \leftrightarrow \mathsf{MPCP}\,(f\,(R, x, y))$

Let x, y and z be strings over Σ and R a set of rewrite rules.

Lemma

If $x \Rightarrow_R^* y$, then there is a match for the MPCP instance $f\,(R, x, y)$.

Correctness Proof $x \Rightarrow_R^* y \leftrightarrow \mathsf{MPCP}\,(f\,(R, x, y))$

Let x, y and z be strings over Σ and R a set of rewrite rules.

Lemma

If $x \Rightarrow_R^* y$, then there is a match for the MPCP instance $f\,(R, x, y)$.

Lemma

Let $A \subseteq f\,(R, x, y)$. If $C_1\,A = z \star (C_2\,A)$, then $z \Rightarrow_R^* y$.

Proof. Size induction on A with a generalized claim for all z. A more general lemma yields either

- $z \Rightarrow_R^* y$ or
- $z \Rightarrow_R^* m$ and $C_1\,A' = m \star (C_2\,A')$ for a smaller list A'. The inductive hypothesis yields $m \Rightarrow_R^* y$.

Correctness Proof $x \Rightarrow_R^* y \leftrightarrow \mathsf{MPCP}\,(f\,(R, x, y))$

Let x, y and z be strings over Σ and R a set of rewrite rules.

Lemma

If $x \Rightarrow_R^* y$, then there is a match for the MPCP instance $f\,(R, x, y)$.

Lemma

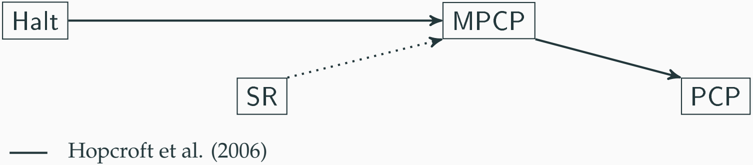
Let $A \subseteq f\,(R, x, y)$. If $C_1\,A = z \star (C_2\,A)$, then $z \Rightarrow_R^* y$.

Proof. Size induction on A with a generalized claim for all z. A more general lemma yields either

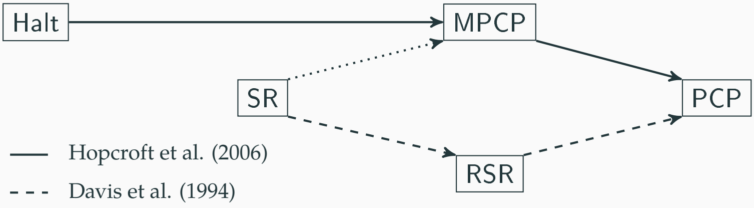
- $z \Rightarrow_R^* y$ or
- $z \Rightarrow_R^* m$ and $C_1\,A' = m \star (C_2\,A')$ for a smaller list A'. The inductive hypothesis yields $m \Rightarrow_R^* y$.

Theorem (SR reduces to MPCP) $\mathsf{SR}\,(R, x, y) \leftrightarrow \mathsf{MPCP}\,(f\,(R, x, y))$

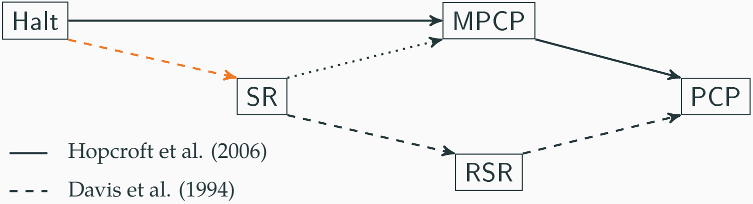
Intermediate Result



Intermediate Result



Intermediate Result



Turing Machines[1] and the Halting Problem

$$
\begin{array}{lccccc}
 & \underset{\uparrow}{\emptyset} & {}_{\uparrow}aA & \underset{\uparrow}{B}aA & Ba_{\uparrow} & \\
\mathsf{tape} := & \emptyset & \mid \mathsf{leftof}\; a\, A & \mid \mathsf{midtape}\; B\, a\, A & \mid \mathsf{rightof}\; a\, B & (a : \Sigma)\; (A\; B : \Sigma^*)
\end{array}
$$

[1] Andrea Asperti and Wilmer Ricciotti (2015)

Turing Machines[1] and the Halting Problem

$$\text{tape} := \underset{\uparrow}{\emptyset} \;\emptyset \quad \mid \text{leftof}\; a\, A \;\; (\uparrow aA) \quad \mid \text{midtape}\; B\, a\, A \;\; (B\underset{\uparrow}{a}A) \quad \mid \text{rightof}\; a\, B \;\; (Ba\uparrow) \qquad (a : \Sigma)\, (A\ B : \Sigma^*)$$

- Turing machine $M := (Q, \delta, q_0, H)$ over finite alphabet Σ
  - transition function $\delta : Q \times \Sigma_\perp \to Q \times \Sigma_\perp \times \{L, N, R\}$
  - halting function $H : Q \to \mathbb{B}$

[1] Andrea Asperti and Wilmer Ricciotti (2015)

Turing Machines[1] and the Halting Problem

$$\text{tape} := \underset{\uparrow}{\emptyset} \;\mid\; \text{leftof}\ a\ A\ \left({}_{\uparrow}aA\right) \;\mid\; \text{midtape}\ B\ a\ A\ \left(B\underset{\uparrow}{a}A\right) \;\mid\; \text{rightof}\ a\ B\ \left(Ba_{\uparrow}\right) \qquad (a : \Sigma)\ (A\ B : \Sigma^*)$$

- Turing machine $M := (Q, \delta, q_0, H)$ over finite alphabet Σ
  - transition function $\delta : Q \times \Sigma_\perp \to Q \times \Sigma_\perp \times \{L, N, R\}$
  - halting function $H : Q \to \mathbb{B}$
- configurations conf : $Q \times$ tape and step function $\hat{\delta}$: conf $\to$ conf
  - $\hat{\delta}\,(q, \underset{\uparrow}{b}aA) = (q', c\underset{\uparrow}{a}A)$ if $\delta\,(q, \lfloor b \rfloor) = (q', \lfloor c \rfloor, R)$
  - $\hat{\delta}\,(q, {}_{\uparrow}aA) = (q', {}_{\uparrow}aA)$ if $\delta\,(q, \perp) = (q', \perp, L)$

[1] Andrea Asperti and Wilmer Ricciotti (2015)

Turing Machines[2] and the Halting Problem

- final configurations $H_c := H\ (\pi_1\ c) = \texttt{true}$

[2] Andrea Asperti and Wilmer Ricciotti (2015)

Turing Machines[2] and the Halting Problem

- final configurations $H_c := H\ (\pi_1\ c) = \texttt{true}$
- reachability predicate: $$\frac{}{c' \vdash c'} \qquad \frac{\hat{\delta}\, c \vdash c' \quad \neg H_c}{c \vdash c'}$$

[2] Andrea Asperti and Wilmer Ricciotti (2015)

Turing Machines[2] and the Halting Problem

- final configurations $H_c := H\ (\pi_1\ c) = \mathsf{true}$
- reachability predicate: $\dfrac{}{c' \vdash c'}$ $\qquad$ $\dfrac{\hat{\delta}\, c \vdash c' \quad \neg H_c}{c \vdash c'}$

Definition: Reachability

$\mathsf{Reach}\ (M, c_1, c_2) := c_1 \vdash c_2$

[2] Andrea Asperti and Wilmer Ricciotti (2015)

Turing Machines[2] and the Halting Problem

- final configurations $H_c := H\,(\pi_1\, c) = \mathsf{true}$
- reachability predicate: $$\frac{}{c' \vdash c'} \qquad \frac{\hat{\delta}\, c \vdash c' \quad \neg H_c}{c \vdash c'}$$

Definition: Reachability

$\mathsf{Reach}\,(M, c_1, c_2) := c_1 \vdash c_2$

Definition: Halting problem

$\mathsf{Halt}\,(M, t) := \exists\, c_f.\, (q_0, t) \vdash c_f \wedge H_{c_f}$

[2] Andrea Asperti and Wilmer Ricciotti (2015)

Reducing Reachability to String Rewriting

$$f(M, c_1, c_2) := (R, x, y)$$

Reducing Reachability to String Rewriting

$$f\left(M, c_1, c_2\right) := (R, x, y)$$

$$\underset{\substack{\uparrow\\ q_0}}{a}ba \quad \vdash \quad a\underset{\substack{\uparrow\\ q_1}}{b}a \quad \vdash \quad ab\underset{\substack{\uparrow\\ q_1}}{a} \quad \vdash \quad aba\underset{\substack{\uparrow\\ q_0}}{} \quad \vdash \quad ab\underset{\substack{\uparrow\\ q_f}}{a}$$

Reducing Reachability to String Rewriting

$$f\,(M, c_1, c_2) := (R, x, y)$$

$$\underset{q_0}{\underset{\uparrow}{a}}ba \;\vdash\; a\underset{q_1}{\underset{\uparrow}{b}}a \;\vdash\; ab\underset{q_1}{\underset{\uparrow}{a}} \;\vdash\; aba\underset{q_0}{\underset{\uparrow}{}} \;\vdash\; ab\underset{q_f}{\underset{\uparrow}{a}}$$

$$(\!|q_0aba|\!) \Rightarrow (\!|aq_1ba|\!) \Rightarrow (\!|abq_1a|\!) \Rightarrow (\!|abaq_0|\!) \Rightarrow (\!|abq_fa|\!)$$

- string encoding $\langle\cdot\rangle : \mathsf{conf} \to \Gamma^*$ with $\Gamma := q : Q \mid a : \Sigma \mid (\!| \mid |\!)$

c	$(q, \underset{\uparrow}{\emptyset})$	$(q, \underset{\uparrow}{}aA)$	$(q, B\underset{\uparrow}{a}A)$	$(q, Ba\underset{\uparrow}{})$
$\langle c\rangle$	$q(\!||\!)$	$q(\!|aA|\!)$	$(\!|BqaA|\!)$	$(\!|Baq|\!)$

Reducing Reachability to String Rewriting

$$f\,(M, c_1, c_2) := (R, \langle c_1 \rangle, \langle c_2 \rangle)$$

$$\underset{\substack{\uparrow\\ q_0}}{a}ba \;\vdash\; a\underset{\substack{\uparrow\\ q_1}}{b}a \;\vdash\; ab\underset{\substack{\uparrow\\ q_1}}{a} \;\vdash\; aba\underset{\substack{\uparrow\\ q_0}}{} \;\vdash\; ab\underset{\substack{\uparrow\\ q_f}}{a}$$

$$(\!|q_0aba|\!) \Rightarrow (\!|aq_1ba|\!) \Rightarrow (\!|abq_1a|\!) \Rightarrow (\!|abaq_0|\!) \Rightarrow (\!|abq_fa|\!)$$

- string encoding $\langle \cdot \rangle$: conf $\to \Gamma^*$ with $\Gamma := q : Q \mid a : \Sigma \mid (\!| \mid |\!)$

c	$(q, \underset{\uparrow}{\emptyset})$	$(q, \underset{\uparrow}{}aA)$	$(q, B\underset{\uparrow}{a}A)$	$(q, Ba\underset{\uparrow}{})$
$\langle c \rangle$	$q(\!||\!)$	$q(\!|aA|\!)$	$(\!|BqaA|\!)$	$(\!|Baq|\!)$

Reducing Reachability to String Rewriting

$$f\,(M, c_1, c_2) := (R, \langle c_1 \rangle, \langle c_2 \rangle)$$

$$\underset{\substack{\uparrow\\ q_0}}{a}ba \;\vdash\; a\underset{\substack{\uparrow\\ q_1}}{b}a \;\vdash\; ab\underset{\substack{\uparrow\\ q_1}}{a} \;\vdash\; aba\underset{\substack{\uparrow\\ q_0}}{\ } \;\vdash\; ab\underset{\substack{\uparrow\\ q_f}}{a}$$

$$(\!|q_0aba|\!) \Rightarrow (\!|aq_1ba|\!) \Rightarrow (\!|abq_1a|\!) \Rightarrow (\!|abaq_0|\!) \Rightarrow (\!|abq_fa|\!)$$

- string encoding $\langle\cdot\rangle : \text{conf} \to \Gamma^*$ with $\Gamma := q : Q \mid a : \Sigma \mid (\!| \mid |\!)$

c	$(q, \underset{\uparrow}{\emptyset})$	$(q, \underset{\uparrow}{\ }aA)$	$(q, B\underset{\uparrow}{a}A)$	$(q, Ba\underset{\uparrow}{\ })$
$\langle c \rangle$	$q(\!||\!)$	$q(\!|aA|\!)$	$(\!|BqaA|\!)$	$(\!|Baq|\!)$

- each rewrite rule realizes one $\hat{\delta}$-step
  - q_0a/aq_1 represents $\delta\,(q_0, \lfloor a \rfloor) = (q_1, \bot, R)$
  - $aq_0|\!)/q_fa|\!)$ and $q_0(\!|/q_f(\!|$ represent $\delta\,(q_0, \bot) = (q_f, \bot, L)$

Reducing Reachability to String Rewriting

$$f\,(M, c_1, c_2) := (R, \langle c_1\rangle, \langle c_2\rangle)$$

$$\underset{\substack{\uparrow\\ q_0}}{a}ba \;\vdash\; a\underset{\substack{\uparrow\\ q_1}}{b}a \;\vdash\; ab\underset{\substack{\uparrow\\ q_1}}{a} \;\vdash\; aba\underset{\substack{\uparrow\\ q_0}}{\;} \;\vdash\; ab\underset{\substack{\uparrow\\ q_f}}{a}$$

$$\llparenthesis q_0 aba \rrparenthesis \Rightarrow \llparenthesis a q_1 ba \rrparenthesis \Rightarrow \llparenthesis ab q_1 a \rrparenthesis \Rightarrow \llparenthesis aba q_0 \rrparenthesis \Rightarrow \llparenthesis ab q_f a \rrparenthesis$$

- string encoding $\langle\cdot\rangle : \mathsf{conf} \to \Gamma^*$ with $\Gamma := q : Q \mid a : \Sigma \mid \llparenthesis \mid \rrparenthesis$

c	$(q, \underset{\uparrow}{\emptyset})$	$(q, \underset{\uparrow}{\;}aA)$	$(q, B\underset{\uparrow}{a}A)$	$(q, Ba\underset{\uparrow}{\;})$
$\langle c\rangle$	$q\llparenthesis\rrparenthesis$	$q\llparenthesis aA\rrparenthesis$	$\llparenthesis BqaA\rrparenthesis$	$\llparenthesis Baq\rrparenthesis$

- each rewrite rule realizes one $\hat{\delta}$-step
  - $q_0 a / a q_1$ represents $\delta\,(q_0, \lfloor a\rfloor) = (q_1, \bot, R)$
  - $a q_0 \rrparenthesis / q_f a \rrparenthesis$ and $q_0 \llparenthesis / q_f \llparenthesis$ represent $\delta\,(q_0, \bot) = (q_f, \bot, L)$
- Δ contains rules that simulate the result of $\delta\,(q, \lfloor a\rfloor)$ and $\delta\,(q, \bot)$ for all non final states $q : Q$ and symbols $a : \Sigma$

Reducing Reachability to String Rewriting

$$f\,(M, c_1, c_2) := (\Delta, \langle c_1 \rangle, \langle c_2 \rangle)$$

$$\underset{\substack{\uparrow \\ q_0}}{a}ba \;\vdash\; a\underset{\substack{\uparrow \\ q_1}}{b}a \;\vdash\; ab\underset{\substack{\uparrow \\ q_1}}{a} \;\vdash\; aba\underset{\substack{\uparrow \\ q_0}}{\,} \;\vdash\; ab\underset{\substack{\uparrow \\ q_f}}{a}$$

$$\llparenthesis q_0aba \rrparenthesis \Rightarrow \llparenthesis aq_1ba \rrparenthesis \Rightarrow \llparenthesis abq_1a \rrparenthesis \Rightarrow \llparenthesis abaq_0 \rrparenthesis \Rightarrow \llparenthesis abq_fa \rrparenthesis$$

- string encoding $\langle\cdot\rangle$: conf $\to \Gamma^*$ with $\Gamma := q : Q \mid a : \Sigma \mid \llparenthesis \mid \rrparenthesis$

c	$(q, \underset{\uparrow}{\emptyset})$	$(q, \underset{\uparrow}{\,}aA)$	$(q, B\underset{\uparrow}{a}A)$	$(q, Ba\underset{\uparrow}{\,})$
$\langle c \rangle$	$q\llparenthesis\rrparenthesis$	$q\llparenthesis aA\rrparenthesis$	$\llparenthesis BqaA\rrparenthesis$	$\llparenthesis Baq\rrparenthesis$

- each rewrite rule realizes one $\hat{\delta}$-step
  - q_0a/aq_1 represents $\delta\,(q_0, \lfloor a \rfloor) = (q_1, \bot, R)$
  - $aq_0\rrparenthesis/q_fa\rrparenthesis$ and $q_0\llparenthesis/q_f\llparenthesis$ represent $\delta\,(q_0, \bot) = (q_f, \bot, L)$
- Δ contains rules that simulate the result of $\delta\,(q, \lfloor a \rfloor)$ and $\delta\,(q, \bot)$ for all non final states $q : Q$ and symbols $a : \Sigma$

Translating the Transition Function into Rewrite Rules

$$\delta\,(q_1, \bot) = (q_2, \text{write}, \text{move})$$

u	v	u	v	write	move
$q_1 \llparenthesis$	$q_2 \llparenthesis$	$c\, q_1 \rrparenthesis$	$q_2\, c \rrparenthesis$	$\bot$	L
$q_1 \llparenthesis$	$q_2 \llparenthesis$	$q_1 \rrparenthesis$	$q_2 \rrparenthesis$	$\bot$	N
$q_1 \llparenthesis\rrparenthesis$	$q_2 \llparenthesis\rrparenthesis$	$q_1 \rrparenthesis$	$q_2 \rrparenthesis$	$\bot$	R
$q_1 \llparenthesis c$	$\llparenthesis q_1\, c$			$\bot$	R
$q_1 \llparenthesis$	$q_2 \llparenthesis b$	$c\, q_1 \rrparenthesis$	$q_2\, c\, b \rrparenthesis$	$\lfloor b \rfloor$	L
$q_1 \llparenthesis$	$\llparenthesis q_2\, b$	$q_1 \rrparenthesis$	$q_2\, b \rrparenthesis$	$\lfloor b \rfloor$	N
$q_1 \llparenthesis$	$\llparenthesis b\, q_2$	$q_1 \rrparenthesis$	$b\, q_2 \rrparenthesis$	$\lfloor b \rfloor$	R

$$\delta\,(q_1, \lfloor a \rfloor) = (q_2, \text{write}, \text{move})$$

u	v	u	v	write	move
$\llparenthesis q_1\, a$	$q_2 \llparenthesis a$	$c\, q_1\, a$	$q_2\, c\, a$	$\bot$	L
		$q_1\, a$	$q_2\, a$	$\bot$	N
		$q_1\, a$	$a\, q_2$	$\bot$	R
$\llparenthesis q_1\, a$	$q_2 \llparenthesis b$	$c\, q_1\, a$	$q_2\, c\, b$	$\lfloor b \rfloor$	L
		$q_1\, a$	$q_2\, b$	$\lfloor b \rfloor$	N
		$q_1\, a$	$b\, q_2$	$\lfloor b \rfloor$	R

Correctness Proof

Lemmas

- If c is not a final configuration, then $\langle c \rangle \Rightarrow_\Delta \langle \hat{\delta}\, c \rangle$.
- If $\langle c \rangle \Rightarrow_\Delta z$, then $z = \langle \hat{\delta}\, c \rangle$ and c is not a final configuration.

Proof. Both lemmas require large case analyses on the tape of configuration c and the result of transitions.

Theorem (Reach reduces to SR) $c_1 \vdash c_2 \leftrightarrow \langle c_1 \rangle \Rightarrow_\Delta^* \langle c_2 \rangle$

Reducing the Halting Problem to String Rewriting

$$f\,(M, t) := (R, \langle (q_0, t) \rangle, y)$$

Reducing the Halting Problem to String Rewriting

$$f\,(M,t) := (R, \langle (q_0, t)\rangle, \varepsilon)$$

Reducing the Halting Problem to String Rewriting

$$f\,(M, t) := (R, \langle (q_0, t) \rangle, \varepsilon)$$

- $(q_0, t) \vdash c_f$ if and only if $\langle (q_0, t) \rangle \Rightarrow^*_\Delta \langle c_f \rangle$

Reducing the Halting Problem to String Rewriting

$$f(M, t) := (R, \langle (q_0, t) \rangle, \varepsilon)$$

- $(q_0, t) \vdash c_f$ if and only if $\langle (q_0, t) \rangle \Rightarrow_\Delta^* \langle c_f \rangle$
- provide rules enabling $\langle c_f \rangle \Rightarrow^* \varepsilon$ for all final configurations c_f:

$$D := \{(q_f s / q_f), (s q_f / q_f), (q_f / \varepsilon) \mid q_f \in Q_H, s \in \Sigma \cup \{(\!|, |\!)\}\}$$

$$(\!| q_0 aba |\!) \Rightarrow_\Delta^* (\!| abq_f a |\!) \Rightarrow_D (\!| abq_f |\!) \Rightarrow_D (\!| abq_f \Rightarrow_D (\!| aq_f \Rightarrow_D (\!| q_f \Rightarrow q_f \Rightarrow_D \varepsilon$$

Reducing the Halting Problem to String Rewriting

$$f(M,t) := (\Delta \cup D, \langle (q_0,t) \rangle, \varepsilon)$$

- $(q_0,t) \vdash c_f$ if and only if $\langle (q_0,t) \rangle \Rightarrow^*_\Delta \langle c_f \rangle$
- provide rules enabling $\langle c_f \rangle \Rightarrow^* \varepsilon$ for all final configurations c_f:

$D := \{(q_f s/q_f), (sq_f/q_f), (q_f/\varepsilon) \mid q_f \in Q_H, s \in \Sigma \cup \{(\!|, |\!)\}\}$

$(\!|q_0aba|\!) \Rightarrow^*_\Delta (\!|abq_fa|\!) \Rightarrow_D (\!|abq_f|\!) \Rightarrow_D (\!|abq_f \Rightarrow_D (\!|aq_f \Rightarrow_D (\!|q_f \Rightarrow q_f \Rightarrow_D \varepsilon$

Reducing the Halting Problem to String Rewriting

$$f\,(M,t) := (\Delta \cup D, \langle (q_0,t)\rangle, \varepsilon)$$

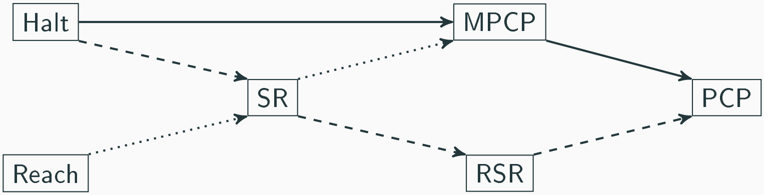
- $(q_0,t) \vdash c_f$ if and only if $\langle (q_0,t)\rangle \Rightarrow^*_\Delta \langle c_f \rangle$
- provide rules enabling $\langle c_f \rangle \Rightarrow^* \varepsilon$ for all final configurations c_f:

$$D := \{(q_f s / q_f), (s q_f / q_f), (q_f/\varepsilon) \mid q_f \in Q_H, s \in \Sigma \cup \{(\!|, |\!)\}\}$$

$$(\!| q_0 aba |\!) \Rightarrow^*_\Delta (\!| abq_f a |\!) \Rightarrow_D (\!| abq_f |\!) \Rightarrow_D (\!| abq_f \Rightarrow_D (\!| aq_f \Rightarrow_D (\!| q_f \Rightarrow q_f \Rightarrow_D \varepsilon$$

Theorem (Halt reduces to SR)
$(\exists\, c_f.\, (q_0,t) \vdash c_f \wedge H_{c_f}) \leftrightarrow \langle (q_0,t)\rangle \Rightarrow^*_{\Delta \cup D} \varepsilon$

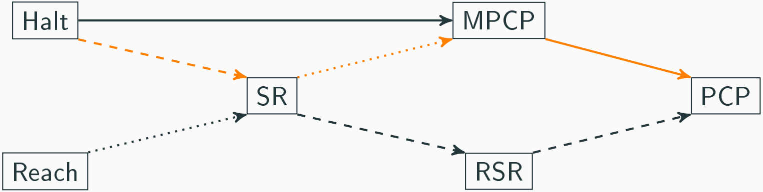
Undecidability Result



— Hopcroft et al. (2006)

- - - Davis et al. (1994)

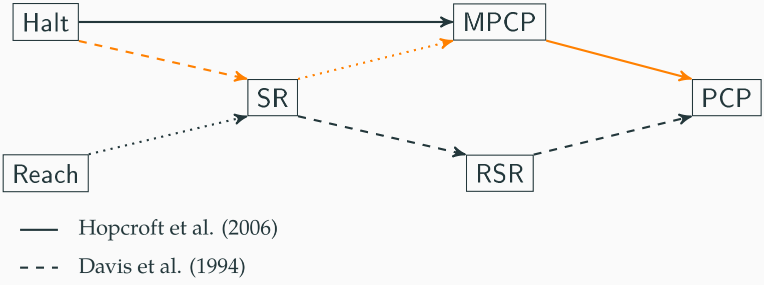
Undecidability Result



— Hopcroft et al. (2006)

- - - Davis et al. (1994)

Undecidability Result



Realization of one Turing machine transition

- reduction via SR: q_0a/aq_1
- direct reduction to MPCP: $\frac{(\!|}{(\!|}\ \frac{q_0a}{aq_1}\ \frac{b}{b}\ \frac{a}{a}\ \frac{|\!)}{|\!)}$

Future Work

- Formalize undecidability proofs based on reductions of PCP:
  - problems related to context-free grammars: inclusion and non-emptiness of intersection (Hopcroft et al. 2006, Hesselink 2015)
  - satisfiability problem for variants of specification formalisms (Finkbeiner and Hahn 2016, Song and Wu 2014)
  - validity of first-oder formulas (Schöning 2009)
  - secrecy problem for security protocols (Tiplea et al. 2005)
- Show PCP λ and Turing undecidable:
  - implement the reductions in the weak call-by-value λ-calculus L (Forster and Smolka 2017)
  - formalize the computational equivalence of L and Turing machines (Dal Lago and Martini 2008)

References

- Andrea Asperti and Wilmer Ricciotti.
  A formalization of multi-tape Turing machines.
  Theoretical Computer Science, 603:23–42, 2015.
- Martin D. Davis, Ron Sigal, and Elaine J. Weyuker.
  Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science.
  Academic Press, 1994.
- Wim H. Hesselink.
  Post's correspondence problem and the undecidability of context-free intersection.
  Manuscript, July 2015.
- John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman.
  Introduction to Automata Theory, Languages, and Computation.
  Addison-Wesley, 2006.
- Emil L Post.
  A variant of a recursively unsolvable problem.
  Bulletin of the American Mathematical Society, 52(4):264–268, 1946.
- Axel Thue.
  Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln.
  J. Dybwad, 1914.

Coq Development

	Spec	Proof	Σ
Definitions	292	121	413
MPCP to PCP	75	145	220
SR to MPCP	50	127	177
Halt to SR	209	349	558
Halt to MPCP	306	517	823
SR to RSR	37	71	108
RSR to PCP	118	328	446
PCP undecidability	9	12	21
	1096	1670	2766

Halt, SR, MPCP, PCP : 955

Halt, SR, RSR, PCP : 1112

Halt, MPCP, PCP : 1043

Proof (SR to MPCP) $\rightarrow$

Lemma

If $z \Rightarrow_R^* y$, then there is some $A \subseteq f\,(R, x, y)$ such that $C_1\,A = z \star (C_2\,A)$.

Proof. Induction on $\Rightarrow *$.

Lemma

If $x \Rightarrow_R^* y$, then there is a match for the MPCP instance $f\,(R, x, y)$.

Proof. The list $\boxed{\frac{\$}{\$x\star}}$ :: A is a match for the MPCP instance.

Proof (SR to MPCP) $\leftarrow$

Lemma

Let $A \subseteq f\,(R, x, y)$. If $C_1\,A = z \star m\,(C_2\,A)$, then either

- $z \Rightarrow_R^* y$ and $m = [\,]$ or
- $A = B +\!\!+ \boxed{\frac{\star}{\star}} :: A'$, $C_1\,B = z$, $C_2\,B = m'$, and $z \Rightarrow_R^* m'$ for some A', B, m'.

Proof. Induction on A for all strings z and m. Let $A = d :: A$.

- $z = [\,]$: $\boxed{\frac{y \star \$}{\$}}$, $\boxed{\frac{u}{v}}$ with $u = [\,]$, and $\boxed{\frac{\star}{\star}}$ are candidates for d
- $z = az'$: $\boxed{\frac{y \star \$}{\$}}$, $\boxed{\frac{u}{v}}$, and $\boxed{\frac{a}{a}}$ are candidates for d

Proof (Reach to SR)

Lemma

If $\langle c \rangle \Rightarrow_\Delta z$, then $z = \langle \hat{\delta}\, c \rangle$ and c is not a final configuration.

Proof. Let $c = (q, t)$. We have $\langle (q, t) \rangle = xuy$ and $z = xvy$ with $u/v \in \Delta$. Case analysis on tape t. Assume $t = \underset{\uparrow}{\emptyset}$.

$\langle (q, \underset{\uparrow}{\emptyset}) \rangle = q(\!|\,|\!) = xuy$. If $u/v = q_1(\!|/(\!|aq_2$ simulating $\delta\,(q_1, \perp) = (q_2, a, R)$, then $q(\!|\,|\!) = xq_1(\!|y$ yields $q = q_1$ and $\langle \hat{\delta}\, c \rangle = (\!|aq_2|\!) = x(\!|aq_2y = z$.

Remark: It is important that $(\!| \neq |\!)$. Assume a configuration $\langle (q_1, \underset{\uparrow}{\emptyset}) \rangle = q_1(\!|\,|\!)$ and $\delta\,(q_1, \perp) = (q_2, \lfloor a \rfloor, R)$.

- The only applicable rewrite rule is $(q_1(\!| \,/\, (\!|aq_2)$ and $\langle \hat{\delta}(q_1, \underset{\uparrow}{\emptyset}) \rangle = \langle (q_2, \underset{\uparrow}{a\;}) \rangle = (\!|aq_2|\!)$.
- If the only one tape delimiter is $\|$, the rule $(q_1\| \,/\, aq_2\|)$ for the right end of the tape is also suitable. But $aq_2\|\| \neq \langle (q_2, \underset{\uparrow}{a\;}) \rangle = \|aq_2\|$.

Proof (Halt to SR)

Lemmas

1. If c_f is a final configuration, then $\langle c_f \rangle \Rightarrow_D^* \varepsilon$.
2. If $\langle c \rangle \Rightarrow_D z$ for some z, then c is a final configuration.
3. If $\langle c \rangle \Rightarrow_{\Delta \cup D}^* \varepsilon$, then $c \vdash c_f$ for some final configuration c_f.

Proof (3). Induction on the derivation $\Rightarrow^*$ with a generalized claim for all c.

- $\langle c \rangle = \varepsilon$ is contradictory.
- $\langle c \rangle \Rightarrow_{\Delta \cup D} z$: If the rewrite rule is from Δ, we use the inductive hypothesis and $z \Rightarrow_{\Delta \cup D}^* \varepsilon$, otherwise the lemma above.

Reducing Restricted String Rewriting to PCP

$$f(R,x,y) := \left\{ \left[\frac{\$}{\$x\star}\right], \left[\frac{y\star\$}{\$}\right], \left[\frac{\star}{\tilde{\star}}\right], \left[\frac{\tilde{\star}}{\star}\right] \right\} \cup \left\{ \left[\frac{a}{\tilde{a}}\right], \left[\frac{\tilde{a}}{a}\right] \;\middle|\; a : \Sigma \right\} \cup \left\{ \left[\frac{u}{\tilde{v}}\right], \left[\frac{\tilde{u}}{v}\right] \;\middle|\; u/v \in R \right\}$$

Example:

$R := \{aa/ab, ab/ba\}$, $x := baa$ and $y := bab$. Since $baa \Rightarrow_R^* bab$ holds, we should be able to construct a match for the PCP instance

$$\left\{ \left[\frac{\$}{\$baa\star}\right], \left[\frac{bab\star\$}{\$}\right], \left[\frac{\star}{\tilde{\star}}\right], \left[\frac{\tilde{\star}}{\star}\right], \left[\frac{a}{\tilde{a}}\right], \left[\frac{\tilde{a}}{a}\right], \left[\frac{b}{\tilde{b}}\right], \left[\frac{\tilde{b}}{b}\right], \left[\frac{aa}{\widetilde{ab}}\right], \left[\frac{\widetilde{aa}}{ab}\right], \left[\frac{ab}{\widetilde{ba}}\right], \left[\frac{\widetilde{ab}}{ba}\right] \right\}$$

$$\left[\frac{\$}{\$baa\star}\right]\left[\frac{b}{\tilde{b}}\right]\left[\frac{aa}{\widetilde{ab}}\right]\left[\frac{\star}{\tilde{\star}}\right]\left[\frac{\tilde{b}}{b}\right]\left[\frac{\tilde{a}}{a}\right]\left[\frac{\tilde{b}}{b}\right]\left[\frac{\tilde{\star}}{\star}\right]\left[\frac{bab\star\$}{\$}\right]$$

$$\frac{\$baa\star\tilde{b}\tilde{a}\tilde{b}\tilde{\star}bab\star\$}{\$baa\star\tilde{b}\tilde{a}\tilde{b}\tilde{\star}bab\star\$}$$

Reducing the Halting Problem to MPCP

tape	$\emptyset$	leftof	midtape	rightof
c	$(q, \underset{\uparrow}{\emptyset})$	$(q, {}_{\uparrow}aA)$	$(q, B\underset{\uparrow}{a}A)$	$(q, Ba{}_{\uparrow})$
$\langle c \rangle$	$\llparenthesis q \sqcup \rrparenthesis$	$\llparenthesis q \sqcup aA \rrparenthesis$	$\llparenthesis BqaA \rrparenthesis$	$\llparenthesis Baq \rrparenthesis$

Encoding of configurations using a blank symbol $\sqcup$.

$$\boxed{\frac{\$}{\$\llparenthesis q_s \sqcup aba \rrparenthesis}}\ \frac{\llparenthesis q_s \sqcup a}{\llparenthesis q_0 a}\ \frac{b}{b}\ \frac{a}{a}\ \frac{\rrparenthesis}{\rrparenthesis}\ \dots\ \frac{\llparenthesis}{\llparenthesis}\ \frac{b}{b}\ \frac{b}{b}\ \frac{bq_0\rrparenthesis}{q_f b\rrparenthesis}\ \frac{\llparenthesis}{\llparenthesis}\ \frac{b}{b}\ \frac{b}{b}\ \frac{q_f b}{q_f}\ \frac{\rrparenthesis}{\rrparenthesis}\ \dots\ \frac{\llparenthesis}{\llparenthesis}\ \frac{bq_f}{q_f}\ \frac{\rrparenthesis}{\rrparenthesis}\ \frac{\llparenthesis q_f \rrparenthesis\$}{\$}$$

- initial domino
- transition dominoes for all non final states
- copy dominoes for all symbols and $\llparenthesis$, $\rrparenthesis$
- deletion dominoes for all final states
- final dominoes for all final states

Reducing MPCP to PCP

$$f\left\{\boxed{\frac{1}{111}},\boxed{\frac{10111}{10}},\boxed{\frac{10}{0}}\right\}=\left\{\boxed{\frac{\$\#1\#0\#1\#1\#1}{\$\#1\#0\#}},\boxed{\frac{\#1}{1\#1\#1\#}},\boxed{\frac{\#1\#0\#1\#1\#1}{1\#0\#}},\boxed{\frac{\#1\#0}{0\#}},\boxed{\frac{\#\$}{\$}}\right\}$$

Both instances are solvable:

10111	1	1	10
10	111	111	0

$#1#0#1#1#1	#1	#1	#1#0	#$
$#1#0#	1#1#1#	1#1#1#	0#	$

- interleave the domino components with # symbols starting to the left of the first symbol in the top string and to the right in the bottom string
- delete empty dominoes since the interleaving has no effect
- provide an additional copy of the first MPCP domino starting at the top and the bottom with $#
- provide an extra domino adding the missing # at the top row