Advanced Topics in Theoretical Computer Science

Part 4: Computability and (Un-)Decidability (II)

16.01.2014

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: sofronie@uni-koblenz.de

Last time

- Recall: Turing machines and Turing computability
- Register machines (LOOP, WHILE, GOTO)
- Recursive functions
- The Church-Turing Thesis
- Computability and (Un-)decidability
- Complexity
- Other computation models: e.g. Büchi Automata

Computability and (Un-)decidability

Known undecidable problems (Theoretical Computer Science I)

- The halting problem for Turing machines
- The equivalence problem

Consequences:

- All problems about programs (TM) which are non-trivial (in a certain sense) are undecidable (Theorem of Rice)
- Identify undecidable problems outside the world of Turing machines
  - Validity/Satisfiability in First-Order Logic
  - The Post Correspondence Problem

Computability and (Un-)decidability

Known undecidable problems (Theoretical Computer Science I)

- The halting problem for Turing machines
- The equivalence problem

Consequences:

- All problems about programs (TM) which are non-trivial (in a certain sense) are undecidable (Theorem of Rice)
- Identify undecidable problems outside the world of Turing machines
  - Validity/Satisfiability in First-Order Logic
  - The Post Correspondence Problem TODAY

Decidability and Undecidability results

Formal languages

- The Post Correspondence Problem and its consequences

Post Correspondence Problem

Idea: We consider non-empty strings over the alphabet $\{a, b\}$.

For example "aaabba".

Assume that we have n pairs of strings $(x_1, y_1), \ldots, (x_n, y_n)$.

Post correspondence problem:

Determine whether there is a set of indices $i_1, \ldots, i_m$ such that

$$x_{i_1} x_{i_2} \ldots x_{i_m} = y_{i_1} y_{i_2} \ldots y_{i_m}.$$

This can contain repeated indices, miss certain indices, ...

Post Correspondence Problem

Definition

A correspondence system (CS) P is a finite rule set over an alphabet Σ.

$$P = \{(p_1, q_1), \ldots, (p_n, q_n)\} \text{ with } p_i, q_i \in \Sigma^*$$

An index sequence $I = i_1 \ldots i_m$ of P is a sequence with $1 \leq i_k \leq n$ for all k.
For every index sequence I we denote $p_I = p_{i_1} \ldots p_{i_m}$ and $q_I = q_{i_1} \ldots q_{i_m}$.

A partial solution is an index set I such that

p_I is a prefix of q_I or q_I is an prefix of p_I.

A solution is an index set I such that $p_I = q_I$.

A (partial) solution with given start is a (partial) solution in which the first index i_1 is given.

The Post correspondence problem (PCP) is the question whether a given correspondence system P has a solution.

Post Correspondence Problem

Example:

Let $P = \{(a, ab), (b, ca), (ca, a), (abc, c)\}$.

- $I = 1, 2, 3, 1, 4$ is a solution:

  $p_I = p_1 p_2 p_3 p_1 p_4 = a\, b\, ca\, a\, abc = abcaaabc = ab\, ca\, a\, ab\, c = q_1 q_2 q_3 q_1 q_4 = q_I$

- $J = 1, 2, 3$ is a partial solution:

  $$p_J = p_1 p_2 p_3 = abca \text{ is a prefix of} q_J = abcaa$$

- There are no solutions with given start 2, 3 or 4.

Plan

We will show that the Post correspondence problem is undecidable.

The proof consists of the following steps:

- We identify two types of "rewrite" systems
  Semi-Thue systems (STS) and Post Normal Systems (PNS).
- We show that the TM computable functions are also STS/PNS computable.
- We define $Trans_G = \{(v, w) \mid v \Rightarrow^* w, v, w \in \Sigma^+\}$ and show that there exist STS/PNS G such that $Trans_G$ is undecidable.
- We assume (to derive a contradiction) that a version of the Post correspondence problem is decidable and show that then also $Trans_G$ is decidable (which is clearly impossible).

STS and PNS

Set of rules. A set of rules over an alphabet Σ is a finite subset $R \subseteq \Sigma^* \times \Sigma^*$. We also write $u \rightarrow_R v$ for $(u, v) \in R$.

R is ϵ-free if for all $(u, v) \in R$ we have $u \neq \epsilon$ and $v \neq \epsilon$.

STS and PNS

Set of rules. A set of rules over an alphabet Σ is a finite subset $R \subseteq \Sigma^* \times \Sigma^*$. We also write $u \to_R v$ for $(u, v) \in R$.

R is ϵ-free if for all $(u, v) \in R$ we have $u \neq \epsilon$ and $v \neq \epsilon$.

Semi-Thue System. In a semi-Thue System, a word w is transformed in a word w' by applying one of the rules (u, v) in R.

Definition. A semi-Thue System (STS) is a pair $G = (\Sigma, R)$ consisting of an alphabet Σ and a set of rules R. G is ϵ-free if R is ϵ-free.

$$w \Rightarrow_G w' \quad \text{iff} \quad \exists u \to_R v, \exists w_1, w_2 \in \Sigma^*(w = w_1 u w_2 \text{ and } w' = w_1 v w_2)$$

Example

Let G be the following semi-Thue system:

$$G = (\{a, b\}, \{ab \to bba, ba \to aba\})$$

$\underline{ab}aba \Rightarrow bba\underline{ab}a \Rightarrow bbabbaa$

$a\underline{ba}ba \Rightarrow aab\underline{ab}a \Rightarrow aabbbaa.$

The rule application in not deterministic.

STS and PNS

Definition. A Post Normal System (PNS) is a pair $G = (\Sigma, R)$ where Σ is an alphabet and a set of rules R. G is ϵ-free if R is ϵ-free.

It differs from a semi-Thue system in the way $\Rightarrow_G$ is defined:

$$w \Rightarrow_G w' \quad \text{iff} \quad \exists u \rightarrow_R v, \exists w_1 \in \Sigma^*(w = uw_1 \text{ and } w' = w_1 v)$$

Definition. A computation in a STS or a PNS G is a sequence $w_1, \ldots, w_n$ with $w_i \Rightarrow_G w_{i+1}$ for all $i \in \{1, \ldots, n-1\}$.
The computation does not continue if there exists no w_{n+1} with $w_n \Rightarrow_G w_{n+1}$.

If there exists $n \geq 1$ with $w_1 \Rightarrow_G \cdots \Rightarrow_G w_n$ we write: $w_1 \Rightarrow_G^* w_n$.

Example

Let G be the following Post Normal System:

$$G = (\{a, b\}, \{ab \to bba, ba \to aba, a \to ba\})$$

Then:

$\underline{ab}aba \Rightarrow \underline{a}babba \Rightarrow \underline{ba}bbaba \Rightarrow bbabaaba$

$\underline{a}baba \Rightarrow \underline{ba}baba \Rightarrow \underline{ba}baaba \Rightarrow \underline{ba}abaaba \Rightarrow \underline{a}baabaaba \Rightarrow \ldots$

(infinite computation)

Post Correspondence Problem

Definition. A partial function $f : {\Sigma_1}^* \to {\Sigma_2}^*$ is STS computable (PNS-computable) iff there exists a STS (a PNS) G s.t. for all $w \in \Sigma_1^*$

- $\forall u \in \Sigma_2^*$, $[w] \Rightarrow_G^* [u\rangle \quad \text{iff} \quad f(w) = u$
- $\not\exists v \in \Sigma_2^*$, $[w] \Rightarrow_G^* [v\rangle \quad \text{iff} \quad f(w)$ undefined.

Note: $[,], \rangle$ are special symbols

F_{STS}^{part} : the family of all (partial) STS computable functions

F_{PNS}^{part} : the family of all (partial) PNS computable functions

Post Correspondence Problem

Theorem $TM^{\text{part}} \subseteq F^{\text{part}}_{STS}$; $TM^{\text{part}} \subseteq F^{\text{part}}_{PNS}$.

Proof:

Idea: show that we can simulate the way a TM works using a suitable STS. We then show that we can slightly change the STS and obtain a PNS which simulates the TM.

From the proof it can be seen that we can simulate any TM using a ϵ-free STS and ϵ-free PNS.

The full proof is rather long and is not presented here.
It can be found on pages 309-311 in the book "Theoretische Informatik" (3. Auflage) by Erk and Priese.

Post Correspondence Problem

$Trans_G = \{(v, w) \mid v \Rightarrow_G^* w \wedge v, w \in \Sigma^+\}$

Theorem.
There exists an ϵ-free STS G such that $Trans_G$ is undecidable.

There exists an ϵ-free PNS G such that $Trans_G$ is undecidable.

Proof.
We can reduce $K = \{n \mid M_n \text{ halts on input } n\}$ to $Trans_G$ for a certain STS (PNS) G.

Let G be an ϵ-free STS or PNS which computes the function of the TM

$$M = M_K M_{\text{delete}}$$

where M_K is the TM which accepts K and M_{delete} deletes the band after M_K halts (such a TM can easily be constructed because $M_K = M_{\text{prep}} U_0$; the halting configurations of the universal TM U_0 are of the form $h_U, \#|^n\#|^m\underline{\#}$).

Input v: M_K halts iff M_v halts on v. If M_K halts, M_{delete} deletes the tape.

Post Correspondence Problem

Proof. (ctd.)

Assume Trans_G decidable. We show how to use G and the decision procedure for Trans_G to decide K:

For $v = [\underbrace{| \dots |}_{n \text{ times}}]$ and $w = [\epsilon\rangle$ we have:

$$
\begin{array}{lll}
(v, w) \in \mathit{Trans}_G & \text{iff} & (v \Rightarrow_G^* w) \\
& \text{iff} & M = M_K M_{\text{delete}} \text{ halts for input } |^n \text{ with } \# \\
& \text{iff} & M_K \text{ halts for input } |^n \\
& \text{iff} & n \in K.
\end{array}
$$

Post Correspondence Problem

Theorem For every ϵ-free semi-Thue System G and every pair of words $w', w'' \in \Sigma^+$ there exists a Post Correspondence System $P_{G,w',w''}$ such that

$$P_{G,w',w''} \text{ has a solution with given start} \quad \text{iff} \quad w' \Rightarrow_G^* w''.$$

Proof: Assume that we are given

- G an ϵ-free STS $G = (\Sigma, R)$ with $|\Sigma| = m$ and $R = \{u_1 \to v_1, \ldots, u_n \to v_n\}$ with $u_i, v_i \in \Sigma^+$
- $w', w'' \in \Sigma^+$

We construct the correspondence system $P_{G,w',w''} = \{(p_i, q_i) \mid 1 \leq i \leq k\}$ with $k = n + m + 3$ over the alphabet $\Sigma_X = \Sigma \cup X$ with:

- the first n rules are the rules in R
- the rule $n + 1$ is $(X, Xw'X)$; the rule $n + 2$ is $(w''XX, X)$
- the rules $n + 2 + 1, \ldots, n + 2 + m$ are (a, a) for every $a \in \Sigma$
- the last rule is (X, X)
- the index for the given start is $n + 1$.

Example

$G = (\Sigma, R)$ with $\Sigma = \{a, b, c\}$ and $R = \{ca \rightarrow ab, ab \rightarrow c, ba \rightarrow a\}$.

For the word pair $w' = caaba$, $w'' = abc$ we have

$$w' = ca\underline{ab}a \Rightarrow_2 ca\underline{ca} \Rightarrow_1 ca\underline{ab} \Rightarrow_2 \underline{ca}c \Rightarrow_1 abc = w''$$

$$\begin{aligned} P_{G,w',w''} = \{ \quad & (ca, ab), (ab, c), (ba, a), (X, XcaabaX), (abcXX, X), \\ & (a, a), (b, b), (c, c), (X, X)\} \end{aligned}$$

We can see that $P_{G,w',w''}$ has a solution with start $n+1$ iff $w' \Rightarrow_G^* w''$

$$p_4 \qquad X \qquad = XcaabaX \qquad = q_4$$

Example

$G = (\Sigma, R)$ with $\Sigma = \{a, b, c\}$ and $R = \{ca \to ab, ab \to c, ba \to a\}$.

For the word pair $w' = caaba$, $w'' = abc$ we have

$$w' = ca\underline{ab}a \Rightarrow_2 ca\underline{ca} \Rightarrow_1 ca\underline{ab} \Rightarrow_2 \underline{ca}c \Rightarrow_1 abc = w''$$

$$\begin{aligned} P_{G,w',w''} = \{ \quad & (ca, ab), (ab, c), (ba, a), (X, XcaabaX), (abcXX, X) \\ & (a, a), (b, b), (c, c), (X, X)\} \end{aligned}$$

We can see that $P_{G,w',w''}$ has a solution with start $n+1$ iff $w' \Rightarrow_G^* w''$

$$p_{486} \qquad = Xca \qquad\qquad = XcaabaXca \qquad\qquad = q_{486}$$

Example

$G = (\Sigma, R)$ with $\Sigma = \{a, b, c\}$ and $R = \{ca \rightarrow ab, ab \rightarrow c, ba \rightarrow a\}$.

For the word pair $w' = caaba$, $w'' = abc$ we have

$$w' = ca\underline{ab}a \Rightarrow_2 ca\underline{ca} \Rightarrow_1 ca\underline{ab} \Rightarrow_2 \underline{ca}c \Rightarrow_1 abc = w''$$

$$\begin{aligned} P_{G,w',w''} = \{ \quad & (ca, ab), (ab, c), (ba, a), (X, XcaabaX), (abcXX, X) \\ & (a, a), (b, b), (c, c), (X, X)\} \end{aligned}$$

We can see that $P_{G,w',w''}$ has a solution with start $n+1$ iff $w' \Rightarrow_G^* w''$

$$p_{4862} \qquad = Xcaab \qquad\qquad = XcaabaXcac \qquad\qquad = q_{4862}$$

Example

$G = (\Sigma, R)$ with $\Sigma = \{a, b, c\}$ and $R = \{ca \to ab, ab \to c, ba \to a\}$.

For the word pair $w' = caaba$, $w'' = abc$ we have

$$w' = ca\underline{ab}a \Rightarrow_2 ca\underline{ca} \Rightarrow_1 ca\underline{ab} \Rightarrow_2 \underline{ca}c \Rightarrow_1 abc = w''$$

$$\begin{aligned} P_{G,w',w''} = \{ \quad & (ca, ab), (ab, c), (ba, a), (X, XcaabaX), (abcXX, X), \\ & (a, a), (b, b), (c, c), (X, X)\} \end{aligned}$$

We can see that $P_{G,w',w''}$ has a solution with start $n+1$ iff $w' \Rightarrow_G^* w''$

$$p_{486269} \quad = XcaabaX \qquad\qquad = XcaabaXcacaX \qquad\qquad = q_{486269}$$

Example

$G = (\Sigma, R)$ with $\Sigma = \{a, b, c\}$ and $R = \{ca \to ab, ab \to c, ba \to a\}$.

For the word pair $w' = caaba$, $w'' = abc$ we have

$$w' = ca\underline{ab}a \Rightarrow_2 ca\underline{ca} \Rightarrow_1 ca\underline{ab} \Rightarrow_2 \underline{ca}c \Rightarrow_1 abc = w''$$

$$\begin{aligned} P_{G,w',w''} = \{ \quad & (ca, ab), (ab, c), (ba, a), (X, XcaabaX), (abcXX, X) \\ & (a, a), (b, b), (c, c), (X, X)\} \end{aligned}$$

We can see that $P_{G,w',w''}$ has a solution with start $n + 1$ iff $w' \Rightarrow^*_G w''$

$$p_{48626986} = XcaabaXca \qquad = XcaabaXcacaXca \qquad = q_{48626986}$$

Example

$G = (\Sigma, R)$ with $\Sigma = \{a, b, c\}$ and $R = \{ca \to ab, ab \to c, ba \to a\}$.

For the word pair $w' = caaba$, $w'' = abc$ we have

$$w' = ca\underline{ab}a \Rightarrow_2 ca\underline{ca} \Rightarrow_1 ca\underline{ab} \Rightarrow_2 \underline{ca}c \Rightarrow_1 abc = w''$$

$$\begin{aligned} P_{G,w',w''} = \{ \quad & (ca, ab), (ab, c), (ba, a), (X, XcaabaX), (abcXX, X) \\ & (a, a), (b, b), (c, c), (X, X)\} \end{aligned}$$

We can see that $P_{G,w',w''}$ has a solution with start $n+1$ iff $w' \Rightarrow_G^* w''$

$p_{4862698619} = XcaabaXcacaX \qquad = XcaabaXcacaXcaabX \quad = q_{4862698619}$

Example

$G = (\Sigma, R)$ with $\Sigma = \{a, b, c\}$ and $R = \{ca \to ab, ab \to c, ba \to a\}$.

For the word pair $w' = caaba, w'' = abc$ we have

$$w' = ca\underline{ab}a \Rightarrow_2 ca\underline{ca} \Rightarrow_1 ca\underline{ab} \Rightarrow_2 \underline{ca}c \Rightarrow_1 abc = w''$$

$$P_{G,w',w''} = \{ \quad (ca, ab), (ab, c), (ba, a), (X, XcaabaX), (abcXX, X) \\ (a, a), (b, b), (c, c), (X, X)\}$$

We can see that $P_{G,w',w''}$ has a solution with start $n+1$ iff $w' \Rightarrow_G^* w''$

$p_{4862698619} = XcaabaXcacaX \qquad = XcaabaXcacaXcaabX \;= q_{4862698619}$

The successive application of rules $2, 1, 2, 1$ corresponds to the solution
$I = \underline{\underline{4}}, 8, 6, \underline{2}, 6, 9, 8, 6, \underline{1}, 9, 8, 6, \underline{2}, 9, \underline{1}, 8, 9, \underline{\underline{5}}$

4,4: begin/end; Underlines: rule applications. Remaining numbers: copy symbols such that rule applications at the desired position. X separates the words in G-derivations.

$p_I = XcaabaXcacaXcaabXcacXabcXX = q_I$

Post Correspondence Problem

Theorem For every ϵ-free semi-Thue System G and every pair of words $w', w'' \in \Sigma^+$ there exists a Post Correspondence System $P_{G,w',w''}$ such that

$$P_{G,w',w''} \text{ has a solution with given start iff } w' \Rightarrow_G^* w''.$$

Proof: Assume that we are given

- G an ϵ-free STS $G = (\Sigma, R)$ with
  $|\Sigma| = m$ and $R = \{u_1 \to v_1, \ldots, u_n \to v_n\}$ with $u_i, v_i \in \Sigma^+$
- $w', w'' \in \Sigma^+$

We construct the correspondence system $P_{G,w',w''} = \{(p_i, q_i) \mid 1 \leq i \leq k\}$ with $k = n + m + 3$ over the alphabet $\Sigma_X = \Sigma \cup X$ with:

- the first n rules are the rules in R
- the rule $n + 1$ is $(X, Xw'X)$; the rule $n + 2$ is $(w''XX, X)$
- the rules $n + 2 + 1, \ldots, n + 2 + m$ are (a, a) for every $a \in \Sigma$
- the last rule is (X, X)
- the index for the given start is $n + 1$.

Post Correspondence Problem

Proof (ctd.) We show that $P_{G,w',w''}$ has a solution iff $w \Rightarrow_G^* w''$.

Occurrences of $X \mapsto$ In the solution index $n+2$ must occur.

Assume $(n+1)I'(n+2)I''$ is a solution in which I' does not contain $n+1$, nor $n+2$. By careful analysis of the equality $p_{(n+1)I'(n+2)I''} = q_{(n+1)I'(n+2)I''}$ we note the following:

(1) no XX in $p_{(n+1)I'}, q_{(n+1)I'}$;

(2) $p_{(n+1)I'(n+2)}$, and $q_{(n+1)I'(n+2)}$ end on XX

(3) $p_{(n+1)I'(n+2)I''} = Xp_{I'}w''XXp_{I''} = Xw'Xq_{I'}Xq_{I''}$, so:

- I' starts with $I_1, (n+m+3)$ with $p_{I_1(n+m+3)} = w'X$.
- Then $q_{I_1,n+m+3} = w_2X$ for some $w_2 \neq \epsilon$.
- I_1 contains only indices in $\{1, \ldots, n\} \cup \{n+3, \ldots, n+2+m\}$.
- Therefore, $w' \Rightarrow_G^* w_2$.

Post Correspondence Problem

Proof (ctd.)

From (1) and (2) it follows that $p_{(n+1)I'(n+2)} = q_{(n+1)I'(n+2)}$.

Thus, if $P_{G,w',w''}$ has a solution then it has a solution of the form $(n+1)I'(n+2)$, such that I' does not contain $(n+1)$ or $(n+2)$.

From (3), by induction, we can show that

$$I' = I_1, (n+m+3), I_2, (n+m+3), \dots, I_k, (n+m+3),$$

where I_j contains only indices in $\{1, \dots, n\} \cup \{n+3, \dots, n+2+m\}$.

Then $p_{I'} = w'Xw_2X \dots Xw_{l-1}X$ and $q_{I'} = w_2X \dots Xw_lX$
for words $w_2, \dots, w_l$ with

$$w' \Rightarrow_G^* w_2 \Rightarrow_G^* \dots \Rightarrow_G^* w_l$$

Post Correspondence Problem

Proof (ctd.)

Thus, for every solution $I = (n+1)I'(n+2)$ we have:

$$p_I = Xw'Xw_2 \dots Xw_{l-1}Xw''XX = q_I$$

with $w' \Rightarrow_G^* w_2 \Rightarrow_G^* \dots \Rightarrow_G^* w_l = w''$.

Conversely, one can prove by induction that if $w' = w_1 \Rightarrow_G^* w_2 \Rightarrow_G^* \dots \Rightarrow_G^* w_l = w''$ is a computation in G then there exists a partial solution I of $P_{G,w',w''}$ with given start $n+1$ and

$$p_I = Xw'Xw_2 \dots Xw_{l-1}X \qquad q_I = Xw'Xw_2 \dots Xw_{l-1}Xw_lX$$

Then $I, (n+2)$ is a solution if $w_l = w''$.

Post Correspondence Problem

Theorem. Assume $|\Sigma| \geq 2$. The Post Correspondence Problem is undecidable.

Proof:

1. We first show that PCP with given start is undecidable.

   Assume that the PCP with given start is decidable. By the previous result it would follow that $Trans_G$ is decidable for every ϵ-free STS G. We showed that there exists at least one ϵ-free STS G for which $Trans_G$ is undecidable. Contradiction. Thus, the PCP with given start is undecidable.

2. We prove that PCP is undecidable.
   For this, we show that for every PCP $P = \{(p_i, q_i) \mid 1 \leq i \leq n\}$ with given start j_0 we can construct a PCP P' such that P has a solution iff P' has a solution.

   Construction: New symbols X, Y; two types of encodings of words:

   $$w = c_1 \dots c_n \mapsto \quad \overline{w} = Xc_1Xc_2 \dots Xc_n; \quad \overline{\overline{w}} = c_1Xc_2 \dots Xc_nX$$

   $P' = \{(\overline{p}_1, \overline{\overline{q_1}}), \dots, (\overline{p}_n, \overline{\overline{q_n}}), (\overline{p}_{j_0}, X\overline{\overline{q_{j_0}}}), (XY, Y)\}$
   A solution of P' can only start with rule $(n+1)$ (only rule where both sides start with same symbol). P has solution with start j_0 iff P' has a solution.

Undecidabile problems in formal languages

Theorem It is undecidable whether a context free grammar is ambiguous.

Proof. Assume that the problem is decidable. Construct algorithm for solving the PCP.

Let $T = \{(u_1, v_1), \dots, (u_n, v_n)\}$ a CS over Σ_1; $\quad \Sigma' = \Sigma_1 \cup \{a_1, \dots, a_n\}$.

$L_{T,1} = \{a_{i_m} \dots a_{i_1} u_{i_1} \dots u_{i_m} | m \geq 1, 1 \leq i_j \leq n\}$ generated by c.f. grammar $G_{T,1}$.

$G_{T,1} = (\{S_1\}, \Sigma', R_1, S_1)$, $R_1 = \{S_1 \rightarrow a_i S_1 u_i \mid 1 \leq i \leq n\} \cup \{S_1 \rightarrow a_i u_i\}$

$L_{T,2} = \{a_{i_m} \dots a_{i_1} v_{i_1} \dots v_{i_m} | m \geq 1, 1 \leq i_j \leq n\}$ generated by c.f. grammar $G_{T,2}$.

$G_{T,2} = (\{S_2\}, \Sigma', R_2, S_2)$, $R_2 = \{S_2 \rightarrow a_i S_2 v_i \mid 1 \leq i \leq n\} \cup \{S_2 \rightarrow a_i v_i\}$

$G_{T,1}, G_{T,2}$ are unambigouus. Let $G_T = (\{S, S_1, S_2\}, \Sigma', R_1 \cup R_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}, S)$.

T has a solution iff $\exists w \in L_{T,1} \cap L_{T,2}$

iff $\exists w \in L(G)$ with two different derivations iff G_T ambiguous.

Undecidable problems in formal languages

Theorem It is undecidable whether the intersection of two

- deterministic context-free languages (DCFL)
- non-ambiguous context-free languages
- context-free languages

is empty.

Proof. Assume that one of the problems is decidable.

Let $T = \{(u_1, v_1), \dots, (u_n, v_n)\}$ a CS over Σ; $\Sigma' = \Sigma \cup \{a_1, \dots, a_n\}$, $c \not\in \Sigma'$.

$L_1 = \{wcw^R \mid w \in (\Sigma')^*\}$: non-ambiguous, deterministic.
$L_2 = \{u_{i_1} \dots u_{i_m} a_{i_m} \dots a_{i_1} c a_{j_1} \dots a_{j_l} v_{j_l}^R \dots v_{j_1}^R \mid m, l \geq 1, i_k, j_p \in \{1, \dots, n\}\}$
L_2 non-ambigous, deterministic (see proof in the book by Erk and Priese)

$$\begin{array}{lll} T \text{ has a solution} & \text{iff} & \exists k{\geq}1\, \exists i_1, \dots, i_k\colon u_{i_1} \dots u_{i_k} = v_{i_1} \dots v_{i_k} \\ & \text{iff} & \exists k{\geq}1\, \exists i_1, \dots, i_k\colon u_{i_1} \dots u_{i_k} a_{i_k} \dots a_{i_1} = (a_{i_1} \dots a_{i_k} v_{i_1}^R \dots v_{i_k}^R)^R \\ & \text{iff} & \exists x{\in}L_2 \text{ such that } x = wcw^R \quad \text{iff} \quad \exists x \in L_2 \cap L_1 \end{array}$$

If we can always decide whether $L_1 \cap L_2 = \emptyset$ then PCP decidable!

Undecidable problems in formal languages

Theorem It is undecidable whether for a context free language $L \subseteq \Sigma^*$ with $|\Sigma| > 1$ we have $L = \Sigma^*$.

Proof. Assume that is was decidable whether $L = \Sigma^*$. We show that then it would be decidable whether $L_1 \cap L_2 = \emptyset$ for DCFL.

Let L_1, L_2 DCFL languages over Σ. Then $L_1 \cap L_2 = \emptyset$ iff $\overline{L_1 \cap L_2} = \Sigma^*$ iff $\overline{L_1} \cup \overline{L_2} = \Sigma^*$.

Note that DCFL's are closed under complement. Then $\overline{L_1}, \overline{L_2} \in \mathcal{L}_2$, so $\overline{L_1} \cup \overline{L_2} \in \mathcal{L}_2$.

Then we could use the decision procedure to check whether $\overline{L_1} \cup \overline{L_2} = \Sigma^*$, i.e. to check whether $L_1 \cap L_2 = \emptyset$. This is a contradiction, since we proved that it is undecidable whether the intersection of two DCFLs is empty.

Undecidable problems in formal languages

Theorem The following problems are undecidable for context-free languages L_1, L_2 and regular languages R over every alphabet Σ with at least two elements.

(1) $L_1 = L_2$

(2) $L_2 \subseteq L_1$

(3) $L_1 = R$

(4) $R \subseteq L_1$

Proof: Let L_1 be an arbitrary context-free language. Choose $L_2 = \Sigma_2^*$. Then L_2 is regular and:

- $L_1 = L_2$ iff $L_1 = \Sigma^*$ (1 and 3)
- $L_2 \subseteq L_1$ iff $L_1 = \Sigma^*$ (2 and 3)

Undecidable problems for $\mathcal{L}_2$

decidable	**undecidable**	
$w \in L(G)$	G ambiguous	
$L(G) = \emptyset$	$D_1 \cap D_2 = \emptyset$	
$L(G)$ finite	$L_1 \cap L_2 = \emptyset$	for non-ambiguous languages $L_1.L_2$
$D_1 = \Sigma^*$	$L_1 = \Sigma^*$	if $\lvert\Sigma\rvert \geq 2$
$L_1 \subseteq R$	$L_1 = L_2$	if $\lvert\Sigma\rvert \geq 2$
	$L_1 \subseteq L_2$	if $\lvert\Sigma\rvert \geq 2$
	$L_1 = R$	if $\lvert\Sigma\rvert \geq 2$
	$R \subseteq L_1$	if $\lvert\Sigma\rvert \geq 2$

where L_1, L_2 are context-free languages; D_1, D_2 are DCFL languages

R is a regular language; G is a context-free grammar, $w \in \Sigma^*$.

Contents

- Recall: Turing machines and Turing computability
- Register machines (LOOP, WHILE, GOTO)
- Recursive functions
- The Church-Turing Thesis
- Computability and (Un-)decidability
- Complexity
- Brief outlook: other computation models, e.g. Büchi Automata

Contents

- Recall: Turing machines and Turing computability
- Register machines (LOOP, WHILE, GOTO)
- Recursive functions
- The Church-Turing Thesis
- Computability and (Un-)decidability
- Complexity
- Brief outlook: other computation models, e.g. Büchi Automata