5.2 Inseparability

Lemma: Let W be perfect.
Then $L_{\mathbb{Z},sj}(W) \not\mid L_{\mathbb{Z},dy}(W) \Rightarrow L_{sj}(W) \not\mid D_n$.

Approach: Use a classical equivalence on finite words due to Büchi:
It equates words that lead to the same state changes in a given finite automaton.

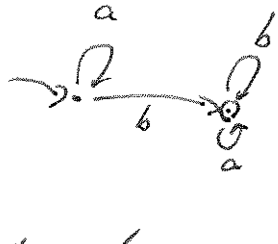
Definition:
Let A be an NFA.
Define $\sim_A \subseteq \Sigma^* \times \Sigma^*$ by

$$u \sim_A v, \text{ if } \forall p,q \in A.Q.\ p \xrightarrow{u} q \text{ iff } p \xrightarrow{v} q$$

Lemma:
- $\sim_A$ is of finite index, namely $|\Sigma^*/_{\sim_A}| \leq 2^{|A.Q|^2}$.
- The equivalence classes $[u]_{\sim_A} = \{v \in \Sigma^* \mid v \sim_A u\}$ are regular languages, namely

$$[u]_{\sim_A} = \bigcap_{\substack{p,q \in A.Q \\ p \xrightarrow{u} q}} L(A_{p,q}) \cap \bigcap_{\substack{p,q \in A.Q \\ p \not\xrightarrow{u} q}} \overline{L(A_{p,q})}.$$

Example:
A



$\varepsilon \sim_A a \sim_A a^2 \sim_A a^3 \sim_A \dots$

$b \sim_A ab \sim_A ba \sim_A bb \sim_A abb \sim_A aba \sim_A \dots$

To prove the lemma, we proceed by contradiction.
Assume $L_{\mathbb{Z},sj}(W) \not\mid L_{\mathbb{Z},dy}(W)$, but A separates $L_{sj}(W)$ and D_n.
Wlog, assume A is a DFA.

Strategy: · We will construct words

$\sigma_{sj} \in L_{sj}(W)$ and $\sigma_{dy} \in L_{dy}(W) \subseteq D_n$

with $\sigma_{sj} \sim_A \sigma_{dy}$.

This yields the contradiction:

↳ $\sigma_{sj} \in L(A) \Rightarrow \sigma_{dy} \in L(A)$,

↯ then $L(A) \cap D_n \neq \emptyset$,

and so A is not a separator

↳ $\sigma_{sj} \notin L(A)$

↯ then $L_{sj}(W) \not\subseteq L(A)$

and so A is not a separator.

· For the construction of σ_{sy} and σ_{dy},

we will use Lambert's iteration lemma (twice):

$$\sigma_{sy} = \lambda(u_0^c . g_0 . w_{sj,0}^c . v_0^c . t_1 \ldots t_h . u_h^c . g_h^c . w_{sj,h}^c . v_h^c) \in L_{sj}(W)$$

$$\sigma_{dy} = \lambda(u_0^c . h_0 . w_{dy,0}^c . v_0^c . t_1 \ldots t_h . u_h^c . h_h . w_{dy,h}^c . v_h^c) \in L_{dy}(W)$$

Note that by perfectness,

the pumping sequences u_0, v_0 to u_h, v_h

are shared between the subject and the Dyck side.

Hence, if we can guarantee

$\lambda(g_i) \sim_A \lambda(h_i)$ for all $i \leq h$

and $\lambda(w_{sj,i}) \sim_A \lambda(w_{dy,i})$ for all $i \leq h$,

then $\sigma_{sy} \sim_A \sigma_{dy}$ follows.

Note: we can select c large enough to work for both sides.

Construction of $\lambda(g_i) \sim_A \lambda(h_i)$:

Idea: - g_i and h_i are the $\mathbb{Z}$-runs
that solve the characteristic equations.

- The purpose of $u_i, w_{sd,i}, v_i$ is just
to turn them into $\mathbb{N}$-runs —
very much like in the proof for reachability.
- To find indistinguishable $\mathbb{Z}$-runs,
we can rely on the lemma's premise $L_{\mathbb{Z},sj}(W) \not| L_{\mathbb{Z},dy}(W)$.

Lemma:

Let A be an NFA so that
for all pairs of words

$w_0(a_1,\#)\ldots(a_k,\#)w_k \in L_{\mathbb{Z},sj}(W)$

$v_0(a_1,\#)\ldots(a_k,\#)v_k \in L_{\mathbb{Z},dy}(W)$

there is $i \le k$ with

$w_i \not\sim_A v_i$.

Then $L_{\mathbb{Z},sj}(W) \mid L_{\mathbb{Z},dy}(W)$.

Proof: We define the regular (see lemma on $\sim_A$) language

finite union as $\sim_A$ has finite index

$$S := \bigcup_{w_0(a_1,\#)\ldots(a_k,\#)w_k \in L_{\mathbb{Z},sj}(W)} [w_0]_{\sim_A}(a_1,\#)\ldots(a_k,\#).[w_k]_{\sim_A} \quad \text{regular}.$$

We claim that S separates $L_{\mathbb{Z},sj}(W)$ and $L_{\mathbb{Z},dy}(W)$.

- Clearly, $L_{\mathbb{Z},sj}(W) \subseteq S$ by definition.
- Assume towards a contradiction

$S \cap L_{\mathbb{Z},dy}(W)$.

Then for $v_0 (a_1, \#) \dots (a_n, \#) v_h \in L_{\mathbb{Z}, dy}(W)$

there is $w_0 (a_1 \#) \dots (a_n \#) v_h \in L_{\mathbb{Z}, sj}(W)$

with

$$v_i \sim_A w_i \quad \text{for all } i \leq h.$$ ↯ to the premise of the lemma. $\square$

<u>Remark:</u>

Here is where we use the $\#$ symbol.

Without it, we could not conclude $v_i \sim_A w_i$.

For $\lambda(g_i) \sim_A \lambda(h_i)$,

we apply the above lemma in contraposition to $L_{\mathbb{Z}, sj}(W) \times L_{\mathbb{Z}, dy}(W)$.

This yields

$$c_0 (a_1, \#) \dots (a_n, \#) c_h \in L_{\mathbb{Z}, sj}(W)$$

and $$b_0 (a_1, \#) \dots (a_h, \#) b_h \in L_{\mathbb{Z}, dy}(W)$$

with $$c_i \sim_A b_i \quad \text{for all } i \leq h.$$

By membership in the languages,

there are loops g_i and h_i in every precovering graph G_i of W

so that

$$\lambda(g_i) = c_i \quad \text{and} \quad \lambda(h_i) = b_i \quad \text{for all } i \leq h.$$

Moreover,

$$\sum_{i \leq h} \psi(g_i) \quad \text{solves} \quad \mathrm{Char}_{sj}(W)$$

$$\sum_{i \leq h} \psi(h_i) \quad \text{solves} \quad \mathrm{Char}_{dy}(W).$$

We thus have solutions to the characteristic equations

as required by Lambert's iteration lemma.

Construction of $\lambda(w_{sj,i}) \sim_A \lambda(w_{dy,i})$:

Approach: · By perfectness, there are
$$u_i' \in (S_{up}(G_i))$$
$$v_i' \in (S_{down}(G_i))$$
for every precovering graph G_i in U.

· We not only construct $w_{sj,i}$ and $w_{dy,i}$, but along with them new covering sequences u_i and v_i.

· The construction will make sure
$$(u_i, w_{sj,i}, v_i)$$ ish and $$(u_i, w_{dy,i}, v_i)$$ ish
are such that we can invoke Lambert's iteration lemma.

Notation: · From now on, we fix the precovering graph G_i and refer to the
$$u_i, v_i, w_{sj,i}, \text{ and } w_{dy,i}$$
that we wish to construct as
$$u, v, w_{sj}, \text{ and } w_{dy}.$$

Recall that the separating automaton A is a DFA.

Let $N := |A.Q|$.

We define runs diff and rem and set
$$w_{sj} := diff^{N}.rem$$
$$w_{dy} := diff^{N+c \cdot N!}.rem.$$
The integer $c \geq 1$ will be defined in a moment.

Claim: $\lambda(w_{sj}) \sim_A \lambda(w_{dy})$

- Consider states $p, q \in A.Q$.

  Since A is a DFA, there is a unique run from p on $\lambda(w_{sj})$.

- Consider the part of the run that reads $\lambda(diff^N)$.

  By the pigeon hole principle, there are

  $$0 \leq i < j \leq N$$

  so that the state reached after reading $\lambda(diff^i)$ equals the state reached after reading $\lambda(diff^j)$.

- This means we can

  repeat $\lambda(diff^{j-i})$ and still arrive at that state.

  In w_{dy}, we repeat $diff^{j-i}$ another

  $$c \cdot \frac{N!}{j-i} \text{ times.}$$

  Note that the factorial function $N!$ guarantees $\frac{N!}{j-i}$ is an integer.

- Since the state reached after reading

  $\lambda(diff^N)$ resp. $\lambda(diff^{N+c \cdot N!})$ coincides,

  we get with $\lambda(w_{sj})$ from p to q

  iff we get with $\lambda(w_{dy})$ from p to q.

Construction of u, v, diff, and rem:

- Let s_{sj}' and s_{dy}' be the full support solutions that satisfy the two conditions in Lambert's iteration lemma:

$$\left.\begin{aligned} & s_{sj}' - \Psi(u') - \Psi(v') \geq 1 \\ & s_{sj}'[G.in, i] + \Delta(u')[i] \geq 1 \\ & s_{sj}'[G.out, i] - \Delta(v')[i] \geq 1 \end{aligned}\right\} \text{Counter } i \text{ with } G.c_{in}[i] = \omega.$$

- We will not only construct the above runs, but also new support solutions s_{sj} and s_{dy}, guided by the equations

$$\Psi(u) + \Psi(v) + \Psi(w_{sj}) = s_{sj}[E] \qquad (1)$$

$$\Psi(u) + \Psi(v) + \Psi(w_{dy}) = s_{dy}[E]. \qquad (2)$$

Recall that

$$w_{sj} = diff^{N}.rem$$
$$w_{dy} = diff^{N + c \cdot N?}.rem.$$

We thus subtract (1) from (2) and get

$$c \cdot N? \cdot \Psi(diff) = (s_{dy} - s_{sj})[E]. \qquad (3)$$

This leads us to define

$$\boxed{s_{sd} := c \cdot N? \cdot s_{sd}'}.$$

- We can then divide (3) by $c \cdot N?$ and get

$$\Psi(diff) = (s_{dy}' - s_{sj}')[E].$$

We thus define diff as a realization of this Parikh vector:

$$\boxed{diff := \langle s_{dy}' - s_{sj}' \rangle}.$$

Note that, to invoke the Euler/Kirchhoff lemma,

we need

$$(s_{dy}' - s_{sj}')[E] \geq 1.$$

We can assume this by just scaling s_{dy}'

by an appropriate factor.

- The new support solutions suggest we should adapt the pumping sequences and define

$$\boxed{\begin{aligned} u &:= (u')^{c \cdot N!} \\ v &:= (v')^{c \cdot N!} \end{aligned}}.$$

- We insert the definitions into Equation (1) and get

$$c \cdot N! \cdot (\psi(u') + \psi(v')) + N \cdot \psi(diff) + \psi(rem) = c \cdot N! \cdot s_{sj}'[E].$$

This yields

$$rem = \langle\, c \cdot N! \underbrace{(s_{sj}'[E] - \psi(u') - \psi(v'))}_{\geq 1} - N \cdot \psi(diff) \,\rangle$$

Here is where we need c.

It has to be large enough

for the above difference to be ≥ 1. $\square$

<u>Remark:</u>

Note that c determined in the last step

goes into the definition of the new support solution.

This means the inequality has to hold

for <u>all</u> precovering graphs, not just for the one at hand.