6. Decomposition

Goal: Prove the following result.

Lemma (Decomposition):

We can decompose a faithful DMGTS W into finite sets Perf and Fin so that:

(i) $\forall\ S \in Perf.\ S$ is perfect

(ii) $\forall\ J \in Fin.\ L_{sj}(J) \mid Dn.$

(iii) $L_{sj}(W) = L_{sj}(Perf \cup Fin).$

Approach: · Capture a single decomposition step, rely on well-foundedness.

· A single decomposition step works as described next.

Lemma (Step):

There is a function $dec(-)$, computable with elementary resources, that takes DMGTS W as input, which satisfies the following:

· faithful,

· imperfect,

· $sol(Chars_{sj}(W)) \neq \emptyset \neq sol(Chars_{dy}(W)).$

It returns finite sets $(X, Y) = dec(W)$ with

(a) $\forall\ W' \in X.\ W'$ is faithful and $W' < W$

(b) $\forall\ W' \in Y.\ L_{sj}(W') \mid Dn$

1- (c) $L_{sj}(W) = L_{sj}(X \cup Y).$

We now use the step lemma to prove the decomposition lemma. In the remainder of the section, we prove the step lemma.

<u>Proof (of the decomposition lemma)</u>:

```
algo (input: a faithful DMGTS W)
begin
   if W is perfect then
      return Perf = {W}, Fin = ∅;
   if sol(C(Ch_wsj(W))) = ∅ then
      return Perf = ∅, Fin = ∅;
   if sol(C(Ch_wdy(W))) = ∅ then
      return Perf = ∅, Fin = {W};
   else
      (X, Y) = dec(W);                  // decomposition
      Perf := ∅
      Fin := Y;
      for all S ∈ X do
         (Perf_S, Fin_S) = algo(S);     // recursion
         Perf := Perf ∪ Perf_S;
         Fin := Fin ∪ Fin_S;
      od
      return (Perf, Fin);
   end else
end
```

Correctness:

- We reason about correctness by induction on the height of the call-tree.
- The height is finite as every recursive call decreases the well-founded order and each node has a finite outdegree. This means König's lemma applies.

W perfect: ✓

sol(Chars$_{sj}$(W)) = ∅ :
- Then $L_{sj}(W) = \emptyset$, and so (iii) is trivial.
- Also (i) and (ii) are trivial.

sol(Chars$_{dy}$(W)) = ∅ :
- Then $L_{\mathbb{Z},dy}(W) = \emptyset$, and so Σ_n^* separates $L_{\mathbb{Z},sj}(W)$ and $L_{\mathbb{Z},dy}(W)$. The separability transfer result shows $L_{sj}(W) \mid D_n$. This proves (ii).
- The points (i) and (iii) are again trivial.

Induction step:

(i) Perfectness for the elements in Perf holds by the induction hypothesis.

(ii) Separability for the elements in Fin holds by the step lemma (for Ψ) and by the induction hypothesis (for Fin$_S$).

(iii) By the step lemma,
we have

$$L_{sj}(W) = L_{sj}(X \cup Y).$$

By the induction hypothesis,

$$L_{sj}(S) = L_{sj}(Perf_S \cup Fin_S)$$

for all $S \in X$.

Hence,

$$L_{sj}(W) = L_{sj}\left(\bigcup_{S \in X}(Perf_S \cup Fin_S) \cup Y\right)$$

$$= \underbrace{L_{sj}\left(\bigcup_{S \in X} Perf_S\right)}_{Perf} \cup \underbrace{L_{sj}\left(\bigcup_{S \in X} Fin_S \cup Y\right)}_{Fin}.$$

$\square$

To prove the step lemma,
we proceed by a case distinction:

A faithful DMGTS W is <u>not perfect</u>
if and only if

$\exists$ precovering graph G in W.

(i) $\exists sd \in \{sj, dy\}.\ \exists j \in sd.\ \exists cio \in \{G.cin, G.cout\}.$

$cio[j] = \omega$ but $x[G.cio, j] \notin supp(Char_{sd}(W))$,

$\vee$ (ii) $\exists sd \in \{sj, dy\}.\ \exists e \in G.E.$

$x[e] \notin supp(Char_{sd}(W))$,

$\vee$ (iii) $CS_{up}(G) = \emptyset$ or $CS_{down}(G) = \emptyset$.