Nondeterministic Automata and JSL-dfas

Robert S.R. Myers

July 14, 2020

1 Introduction

Here’s a summary of our results.

- Section 2 provides background on finite join-semilattices and describes an equivalent category $\text{Dep}$. The latter has the finite relations as objects/morphisms; its self-duality takes the converse of objects/morphisms. This section serves as a succinct version of our paper ‘Representing Semilattices as Relations’.

- Section 3 introduces the concept of Dependency Automaton i.e. two nfas with a relation between their states satisfying compatibility conditions. They are essentially deterministic automata interpreted in $\text{Dep}$, or equivalently deterministic finite automata interpreted in join-semilattices. The state-minimal JSL-dfa accepting $L$ amounts to the left quotients of $L$. As a dependency automaton it can be represented as the state-minimal dfas for $L$ and $L'$ related by the dependency relation $\text{DR}_{L}(u^{-1}L, v^{-1}L') : \iff uv \in L$.

- We also go into some detail concerning various canonical JSL-dfas and their corresponding dependency automata. For example, Polak’s syntactic semiring is the transition semiring of the state-minimal JSL-dfa. Also, the power semiring of the syntactic monoid dualises the closure of $L$ under left/right quotients and boolean operations.

- Section 4 contains many results concerning the Kameda-Weiner algorithm. They lack a unifying thread, although they’re all concerned with the same topic. The reader might skip to the final subsection. There it is proved that an nfa $N$ is ‘subatomic’ iff the transition monoid of $\text{rsc}(\text{rev}(N))$ is syntactic.

2 Relations and semilattices

The Kameda-Weiner algorithm is not an easy read [KW70]. It searches for an edge-covering of a bipartite graph by complete bipartite graphs, where each covering induces a nondeterministic automaton. The best known lower-bound techniques for nondeterministic automata involve such edge-coverings [GH06]. Thus we begin with a structural theory at this underlying level. Our approach is based on the work of Moshier and Jipsen [Jip12]. Our category $\text{Dep}$ is a variant of their category $\text{Ctxt}$, and we denote its composition by $\cdot$ as they did.

2.1 Biclique edge-coverings as morphisms

Notation 2.1.1 (Relations and graphs).

1. Given a subset $X \subseteq Z$ then its relative complement is written $\overline{X} \subseteq Z$. Given $z \in Z$ we may write $\overline{z} := \{z\}$. The collection of all subsets of $Z$ is denoted $\mathcal{P}Z$.

2. A relation is a subset of a specified cartesian product $R \subseteq X \times Y$. We denote its domain by $R_d := X$ and its codomain by $R_c := Y$. The relational composition $R; S \subseteq R_d \times S_c$ is defined whenever $R_c = S_d$, as follows:

$$R; S(x, z) : \iff \exists y \in R_d. R(x, y) \land S(y, z).$$

Each set $Z$ has the identity relation $\Delta_Z \subseteq Z \times Z$ defined $\Delta_Z(z_1, z_2) : \iff z_1 = z_2$.

The image of $X \subseteq R_d$ under $R$ is denoted $R[X] := \{y \in R_c : \exists x \in X. R(x, y)\}$; we may write $R[\{z\}]$ as $R[z]$. $R|_{X,Y} := R \cap X \times Y$ is the domain-codomain restriction of the relation $R$. The converse relation is defined $R^c(x, y) : \iff R(y, x)$, in particular $R^c = R^c$ and $R = R^c$. 

1
3. An **undirected graph** (or just **graph**) \((V, \mathcal{E})\) is a finite set \(V\) and an irreflexive and symmetric relation \(\mathcal{E} \subseteq V \times V\).

A **bipartition** for a graph \((V, \mathcal{E})\) is a pair \((X, Y)\) where \(X \cap Y = \emptyset\), \(X \cup Y = V\) and \(\mathcal{E} = \mathcal{E}|_{X \times Y} \cup \mathcal{E}|_{Y \times X}\). A graph is said to be **bipartite** if it has a bipartition.

**Note 2.1.2** (Bipartitioned graphs as binary relations). A bipartite undirected graph \((V, \mathcal{E})\) with bipartition \((X, Y)\) amounts to a relation \(\mathcal{E}|_{X \times Y} \subseteq X \times Y\). This completely captures its structure. Every relation between finite sets arises from a bipartitioned graph, modulo bijective relabelling of its domain (or codomain).

From this perspective, complete bipartite graphs (bicliques) are **cartesian products**. Covering the edges of a bipartite graph by bicliques amounts to **factorising** \(\mathcal{R} = \mathcal{S}; \mathcal{T}\). This relationship is well-known [GPJL91]. The number of bicliques is the cardinality of the set \(S_t = T_s\) factorised through. The minimum possible cardinality is the **bipartite dimension** of the respective bipartite graph i.e. the minimum number of bicliques needed to cover the edges.

**Definition 2.1.3** (Biclique edge-coverings).

1. A **biclique** of a relation \(\mathcal{R}\) is a cartesian product \(X \times Y \subseteq \mathcal{R}\).

2. A **biclique edge-covering** of a relation \(\mathcal{R}\) is a factorisation \(\mathcal{R} = \mathcal{S}; \mathcal{T}\). Its **underlying bicliques**:

\[ C_{S, T} := \{ \mathcal{S}[x] \times \mathcal{T}[x] : x \in S_t = T_s \} \]

satisfy the equality \(\cup C_{S, T} = \mathcal{R}\).

3. The **bipartite dimension** \(\dim(\mathcal{R})\) is the minimal cardinality \(|C_{S, T}|\) over all factorisations \(\mathcal{R} = \mathcal{S}; \mathcal{T}\).

**Notation 2.1.4** (Lower/upper bipartition). When a binary relation \(\mathcal{R}\) is viewed as a bipartitioned graph, we may refer to its domain as the **lower bipartition** and its codomain as the **upper bipartition**.

**Example 2.1.5** (Biclique edge-coverings).

1. Each relation \(\mathcal{R}\) has two canonical biclique edge-coverings i.e. \(\mathcal{R} = \Delta_{\mathcal{R}} ; \mathcal{R}\) and \(\mathcal{R} = \mathcal{R} ; \Delta_{\mathcal{R}}\). Viewed as a bipartite graph, the stars centered at each vertex of the lower bipartition cover the edges. Alternatively we can take each star centered at a vertex in the upper bipartition. Consequently \(\dim(\mathcal{R}) \leq \min(|\mathcal{R}_s|, |\mathcal{R}_t|)\).

2. Each undirected graph \((V, \mathcal{E})\) provides an irreflexive symmetric relation \(\mathcal{E} \subseteq V \times V\). From our viewpoint, this relation defines a bipartitioned graph. It is known as the **bipartite double cover of** \((V, \mathcal{E})\) i.e. take two copies of \(V\) and connect \(e_1(u)\) to \(e_2(v)\) iff \(\mathcal{E}(u, v)\). Starting with a complete graph on vertices \(V\) yields the relation \(\Delta V \subseteq V \times V\). Interestingly, \(\dim(\Delta V) \approx \lceil \log_2(|V|) \rceil\) by applying Sperner’s theorem [BFRK08].

3. Each finite poset \(P = (P, \leq_P)\) provides an order relation \(\leq_P \subseteq P \times P\). Viewed as a bipartitioned graph, paths amount to alternating relationships \(p_1 \leq_P p_2 \geq_P p_3 \leq_P p_4 \cdots\). The edge-coverings from (1) above are optimal i.e. \(\dim(\leq_P) = |P|\). The lower/upper bipartition’s stars correspond to principal up/downsets.

4. Consider \(C_{2n} = (\{0, \ldots, 2n - 1\}, \mathcal{E}_{2n})\) where \(\mathcal{E}_{2n}(i, j) : \iff |j - i| = 1\) modulo \(2n\) i.e. the \(2n\)-cycle. It has precisely two bipartitions if \(n \geq 1\). Assuming 0 is in the lower bipartition, the respective relation \(\mathcal{E}_{2n}|_{X \times Y}\) relates evens to odds and has bipartite dimension \(|X| = |Y| = n\).

5. Consider \(P_n = (\{0, \ldots, n\}, \mathcal{E}_n)\) where \(\mathcal{E}_n(i, j) : \iff |j - i| = 1\) i.e. the path of edge-length \(n\). It has precisely two bipartitions if \(n \geq 1\). Assuming 0 is in the lower bipartition, the respective relation \(\mathcal{E}_n|_{X \times Y}\) relates evens to odds. Moreover \(\dim(\mathcal{E}_{2n}|_{X \times Y}) = \dim(\mathcal{E}_{2n-1}|_{X \times Y}) = n\).

**Definition 2.1.6** (The category Dep). The objects of the category Dep are the relations \(\mathcal{G} \in \mathcal{G}_s \times \mathcal{G}_t\) between finite sets. A morphism \(\mathcal{R} : \mathcal{G} \to \mathcal{H}\) is a relation \(\mathcal{R} \subseteq \mathcal{G}_s \times \mathcal{H}_t\) such that the diagram:

\[
\begin{array}{ccc}
\mathcal{G}_t & \xrightarrow{(\mathcal{R}_s)^{-1}} & \mathcal{H}_t \\
\mbox{\small \mathcal{G}_s} & \xrightarrow{\mathcal{R}} & \mathcal{H} \\
\end{array}
\]

2
commutes in $\text{Rel}_{f}$ for some $R_{l} \subseteq G_{s} \times H_{s}$ and $R_{u} \subseteq H_{t} \times G_{t}$\footnote{The category whose objects are the finite sets and whose morphisms are the binary relations, composed via relational composition.}. The identity morphisms are $id_{G} := G$ and composition $R \triangleright S : G \rightrightarrows H \rightarrowtail I$ is defined:

$$
\begin{array}{ccc}
\circlearrowleft & \circlearrowright \\
G_{t} & \stackrel{(S_{u})^{-}}{\rightarrow} & H_{t} \quad \downarrow \quad S \quad \downarrow \quad I_{t} \\
G_{s} & \stackrel{(R_{u})^{-}}{\rightarrow} & H_{s} \quad \downarrow \quad S \quad \downarrow \quad I_{s}
\end{array}
$$

That is, $R \triangleright S := R_{l}; S_{t} = R_{l}; S = R_{l}; H_{t} ; (S_{u})^{-} = R_{l} ; (R_{u})^{-} ; (S_{u})^{-}$ is any of the five equivalent relational compositions starting from the bottom left and ending at the top right.

Then a Dep-morphism $R : G \rightarrow H$ is a relation factorising through $G$ on the left and $H$ on the right. In view of Note 2.1.2, it amounts to two biclique edge-coverings of $R$.

**Lemma 2.1.7.** Dep is a well-defined category.

**Proof.** Concerning identity morphisms, $\Delta_{G} ; G = G ; \Delta_{G}^{\cdot}$; graph-theoretically we are using the star-coverings from Example 2.1.5.1. Concerning composition, $R \triangleright S$ is well-defined: (i) the commuting rectangle provides witnessing relations $R_{l}; S_{t}$ and $S_{u}; R_{u}$, (ii) $R \triangleright S$ is independent of the witnesses for $R$ and $S$ by considering the 5 relational compositions. We have $R \triangleright id_{G} = R; \Delta_{G} = R$ and $id_{G} \triangleright R = \Delta_{G} ; R = R$. Composition is associative because relational composition is.

**Example 2.1.8** (Dep-morphisms).

1. **Dep-morphisms are closed under converse and union.**

   Given $R : G \rightarrow H$ then $\hat{R} : H \rightarrow G$ by taking the converse of the commutative square, which actually swaps the witnessing relations. We have $\emptyset : G \rightarrow H$ via empty witnessing relations. Given $R, S : G \rightarrow H$ then $R \cup S : G \rightarrow H$ by (i) unioning the respective witnessing relations, (ii) the bilinearity of relational composition w.r.t. union.

2. **Bipartite graph isomorphisms $\beta : G_{1} \rightarrow G_{2}$ induce Dep-isomorphisms.**

   Suppose we have a bipartite graph isomorphism $\beta : G_{1} \rightarrow G_{2}$ where each $G_{i} = (V_{i}, E_{i})$, so $E_{1}(x,y) \iff E_{2}(\beta(x), \beta(y))$. Given any bipartition $(X,Y)$ of $G_{1}$ we obtain a bipartition $(\beta[X], \beta[Y])$ of $G_{2}$. Setting $G_{i} := E_{i}[X \times Y]$ provides the Dep-morphism below left:

$$
\begin{array}{ccc}
Y & \stackrel{\beta[Y \times \beta[Y]]}{\rightarrow} & \beta[Y] \\
\circlearrowleft & \circlearrowright \\
G_{i} & \circlearrowleft & \circlearrowright \\
X & \stackrel{\beta[X \times \beta[X]]}{\rightarrow} & \beta[X]
\end{array}
$$

The bijective inverse $\beta^{-1} = \hat{\beta}$ provides witnessing relations in the opposite direction i.e. the Dep-morphism $S : G_{2} \rightarrow G_{1}$ above right. These morphisms are mutually inverse: $G_{1}$ is Dep-isomorphic to $G_{2}$.

3. **The canonical quotient poset of a preorder defines a Dep-isoamorphism.**

Let $G \subseteq X \times X$ be a transitive and reflexive relation. There is a canonical way to construct a poset $P = (X/\mathcal{E}, \leq_{P})$ via the equivalence relation $\mathcal{E}(x_{1}, x_{2}) := \iff G(x_{1}, x_{2}) \land G(x_{2}, x_{1})$, where $[x_{1}]_{\mathcal{E}} \leq_{P} [x_{2}]_{\mathcal{E}} := \iff G(x_{1}, x_{2})$.

Consider the Rel-diagram:

$$
\begin{array}{ccc}
X & \xrightarrow{\delta} & \{ G[x] : x \in X \} = \{ \mathcal{U} \uparrow_{P} \{ x \}_{\mathcal{E}} : x \in X \} \\
\downarrow \quad \circlearrowleft & \circlearrowright \\
\{ G[x] : x \in X \} & \xrightarrow{\lambda x \in \cdot G[x]} & \{ \mathcal{U} \uparrow_{P} \{ x \}_{\mathcal{E}} : x \in X \} \\
\downarrow \quad \circlearrowleft & \circlearrowright \\
X & \xrightarrow{\lambda x \in \cdot G[x]} & \{ G[x] : x \in X \} \quad \downarrow \quad \leq_{P} \\
\circlearrowleft & \circlearrowright \\
\{ G[x] : x \in X \} & \xrightarrow{\lambda x \in \cdot G[x]} & \{ \mathcal{U} \uparrow_{P} \{ x \}_{\mathcal{E}} : x \in X \} \\
\downarrow \quad \circlearrowleft & \circlearrowright \\
X & \xrightarrow{\lambda x \in \cdot G[x]} & \{ G[x] : x \in X \} \quad \leq_{P} \\
\end{array}
$$

\footnote{The converse symbol $(R_{u})^{-}$ is intentional. It provides symmetry later on.}
Note that $\mathcal{G}[x]$ is the ‘upwards closure’ i.e. the union of the upwards closure $\uparrow_P [x]_{\mathcal{E}}$, whereas $\overline{\mathcal{G}}[x]$ is the ‘downwards closure’ in a similar manner. The left square commutes for completely general reasons, defining the $\text{Dep}$-morphism:

$$\mathcal{R}(x_1, \overline{\mathcal{G}}[x_2]) : \iff \exists x \in X. [\mathcal{G}(x_1, x) \land \mathcal{G}(x, x_2)] \iff \mathcal{G}(x_1, x_2).$$

The right square involves bijections via (i) identifying elements of $P$ with principal up/downsets, (ii) the disjointness of equivalence classes. It also commutes:

$$\bigcup \uparrow_P [x_1]_{\mathcal{E}} \notin \bigcup \downarrow_P [x_2]_{\mathcal{E}} \iff \exists x \in X. [\uparrow_P [x_1]_{\mathcal{E}} \cap \downarrow_P [x_2]_{\mathcal{E}} \neq \emptyset]$$

In fact, $\mathcal{R} : \mathcal{G} \rightarrow \mathcal{G}$ is an instance of the natural isomorphism $\text{red}_\mathcal{G}$ from Theorem 2.2.14 further below, and the right square defines a $\text{Dep}$-isomorphism by Example 2 above. Thus $\mathcal{G} \cong \mathcal{G}_p$, although whenever $|X| > |X/\mathcal{E}|$ this isomorphism cannot arise from a bipartite graph isomorphism.

4. **Monotonicity can be characterised by $\text{Dep}$-morphisms.**

Given finite posets $P$ and $Q$, a function $f : P \rightarrow Q$ is monotonic iff the following $\text{Rel}$-diagram commutes:

$$\begin{array}{ccc}
P & \xrightarrow{f} & Q \\
\downarrow_{\leq_P} & & \downarrow_{\leq_Q} \\
P & \xrightarrow{f} & Q
\end{array}$$

as the reader may verify. Actually, $f$ is monotonic iff $f ; \leq_P : \leq_P \rightarrow \leq_Q$ is a $\text{Dep}$-morphism. Indeed, given that $f ; \leq_Q : \leq_P \rightarrow \leq_Q$ is a $\text{Dep}$-morphism we’ll prove that $f$ is monotonic in Example 2.2.4 further below.

5. **Biclique edge-coverings amount to $\text{Dep}$-monos.**

Generally speaking, $\text{Dep}$-morphisms represent two edge-coverings of a bipartitioned graph. A single edge-covering amounts to a $\text{Dep}$-mono of a special kind:

$$\begin{array}{ccc}
\mathcal{G}_s & \xrightarrow{\Delta_{\mathcal{G}_s}} & \mathcal{G}_s \\
\downarrow_{\mathcal{G}} & & \downarrow_{\mathcal{H}} \\
\mathcal{G}_s & \xrightarrow{\mathcal{G}} & \mathcal{H}_s
\end{array}$$

i.e. morphisms $\mathcal{G} : \mathcal{G} \rightarrow \mathcal{H}$ with the additional assumption $\mathcal{G}_s = \mathcal{H}_s$. It will follow later that any mono $\mathcal{R} : \mathcal{G} \rightarrow \mathcal{I}$ induces such a $\mathcal{G} : \mathcal{G} \rightarrow \mathcal{H}$ where $|\mathcal{H}_s| \leq |\mathcal{I}_s|$ and $|\mathcal{H}_t| \leq |\mathcal{I}_t|$ i.e. see Theorem ??.

6. **Biclique edge-coverings amount to $\text{Dep}$-epis.**

Analogous to the previous example, a single edge-covering can be represented as a $\text{Dep}$-epi $\mathcal{G} : \mathcal{H} \rightarrow \mathcal{G}$ where $\mathcal{G}_s = \mathcal{H}_s$. This will follow from self-duality i.e. epis are precisely the converses of monos.

2.2 **The categorical equivalence**

Some of the examples above are order-theoretic in nature. Indeed, the main result of this section is:

$\text{Dep}$ is categorically equivalent to the finite join-semilattices equipped with join-preserving morphisms.

This result will characterise $\text{Dep}$-objects modulo isomorphism. They are the union-free or reduced relations. Algebraically they correspond to the finite lattices.

**Definition 2.2.1** (Image, preimage, closure, interior). For any binary relation $\mathcal{R} \subseteq \mathcal{R}_s \times \mathcal{R}_t$ define:

$$\begin{align*}
\mathcal{R}^1 : \mathcal{P}\mathcal{R}_s & \rightarrow \mathcal{P}\mathcal{R}_t \\
\mathcal{R}^1(X) & := \mathcal{R}[X] \\
\mathcal{R}^1 : \mathcal{P}\mathcal{R}_s & \rightarrow \mathcal{P}\mathcal{R}_s \\
\mathcal{R}^1(Y) & := \{ x \in \mathcal{R}_s : \mathcal{R}[x] \subseteq Y \} \\
\mathcal{cl}_\mathcal{R} & := \mathcal{R}^1 \circ \mathcal{R}^1 : (\mathcal{P}\mathcal{R}_s, \subseteq) \rightarrow (\mathcal{P}\mathcal{R}_s, \subseteq) \\
\mathcal{in}_\mathcal{R} & := \mathcal{R}^1 \circ \mathcal{R}^1 : (\mathcal{P}\mathcal{R}_t, \subseteq) \rightarrow (\mathcal{P}\mathcal{R}_t, \subseteq)
\end{align*}$$
where \( \mathcal{P}Z \) is the collection of all subsets of \( Z \). The fixed points of the closure operator \( \text{cl}_\mathcal{R} \) are \( C(\mathcal{R}) := \{ \mathcal{R}^i(Y) : Y \in \mathcal{R}_t \} \). The fixed-points of the interior operator\(^3\) \( \text{in}_\mathcal{R} \) are \( O(\mathcal{R}) := \{ \mathcal{R}[X] : X \in \mathcal{R}_s \} \) and are called the \( \mathcal{R} \)-open sets.

**Definition 2.2.2** (Component relations). For each \( \text{Dep-morphism} \ \mathcal{R} : \mathcal{G} \to \mathcal{H} \) define:

\[
\mathcal{R}_- := \{(g_s, h_s) \in \mathcal{G}_s \times \mathcal{H}_s : h_s \in \mathcal{H}^i(\mathcal{R}[g_s])\} \quad \mathcal{R}_+ := \{(h_t, g_t) \in \mathcal{H}_t \times \mathcal{G}_t : g_t \in \mathcal{G}^i(\mathcal{R}[h_t])\}
\]

called the lower/upper components respectively.

Importantly, \( \mathcal{R} \)'s component relations are witnesses and contain all other witnesses.

**Lemma 2.2.3** (Morphisms characterisation and maximum witnesses).

Let \( \mathcal{R} \subseteq \mathcal{G}_s \times \mathcal{H}_t \) be any relation between finite sets.

1. \( \mathcal{R}^i(X) \subseteq Y \iff X \subseteq \mathcal{R}^i(Y) \) for all subsets \( X \subseteq \mathcal{G}_s, Y \subseteq \mathcal{H}_t \).

2. The following labelled equalities hold:

\[
(\uparrow \downarrow) \quad \mathcal{R}^i \circ \mathcal{R}^i \circ \mathcal{R}^i = \mathcal{R}^i \quad \mathcal{R}^i \circ \mathcal{R}^i \circ \mathcal{R}^i = \mathcal{R}^i \quad (\downarrow \uparrow)
\]

\[
(\neg \uparrow \neg) \quad \neg \mathcal{G}_t \circ \mathcal{R}^i \circ \neg \mathcal{G}_s = \mathcal{R}^i \quad \neg \mathcal{G}_s \circ \mathcal{R}^i \circ \neg \mathcal{G}_t = \mathcal{R}^i \quad (\neg \downarrow \neg)
\]

3. \( \text{cl}_\mathcal{R} \) (resp. \( \text{in}_\mathcal{R} \)) is a well-defined closure (resp. interior) operator, in fact \( \text{in}_\mathcal{R} = \neg \mathcal{R}_- \circ \text{cl}_\mathcal{R} \circ \neg \mathcal{R}_- \).

4. \( \mathcal{R} \) defines a \( \text{Dep-morphism} \ \mathcal{G} \to \mathcal{H} \iff \mathcal{R}^i \circ \text{cl}_\mathcal{G} = \mathcal{R}^i = \text{in}_\mathcal{H} \circ \mathcal{R}^i \), or equivalently \( \mathcal{R}^i \circ \text{cl}_\mathcal{G} = \text{in}_\mathcal{H} \circ \mathcal{R}^i \).

5. Each \( \mathcal{R} : \mathcal{G} \to \mathcal{H} \) has the maximum witnesses \((\mathcal{R}_-, \mathcal{R}_+)\) i.e.
   - \( \mathcal{R}_- : \mathcal{H} = \mathcal{G} ; \mathcal{R}_+^o \).
   - whenever \( \mathcal{R}_t : \mathcal{H} = \mathcal{G} ; \mathcal{R}_+^o \) then \( \mathcal{R}_t \subseteq \mathcal{R}_- \) and \( \mathcal{R}_t \subseteq \mathcal{R}_+ \).

6. For every \( \mathcal{G} \)-open \( Y \in \mathcal{O}(\mathcal{G}) \) and every \( g_t \in \mathcal{G}_t \):

\[
Y \subseteq \text{in}_\mathcal{G}(Y) \iff g_t \notin Y.
\]

**Proof.** See background paper i.e.

- Lemma 4.1.7, Relating \((-)^i \) and \((-)^i \) and also Lemma 4.2.7.

- Lemma 4.1.10, Morphism characterisation and maximum witnesses.

\)[/]

**Example 2.2.4.**

1. **Characterizing monotonicity.**

   Recalling Example 2.1.8.4, suppose \( f : P \to Q \) is a function and \( f : \leq_Q : \leq_P \to \leq_Q \) is a \( \text{Dep-morphism} \). Since \( \text{cl}_{\leq_P} \) constructs the upwards closure in \( P \), by Lemma 2.2.3.4 for any \( p \in P \):

\[
(f : \leq_Q)^i(\uparrow_P p) = (f : \leq_Q)[p] \quad \text{or equivalently} \quad \uparrow_Q f[\uparrow_P p] = \uparrow_Q f(p).
\]

   Thus whenever \( p \leq_P p' \) we know \( \uparrow_Q f(p') \subseteq \uparrow_Q f(p) \) i.e. \( f(p) \leq_Q f(p') \).

2. **One-sided maximal bicliques.**

   When searching for small edge-coverings by bicliques one can restrict to maximal ones i.e. \( X \times Y \subseteq \mathcal{G} \) where \( (X,Y) \) is pairwise-maximal w.r.t. inclusion \cite{Orl77}. Then Lemma 2.2.3.4 says:

   \( \text{Dep-morphisms} \ \mathcal{R} : \mathcal{G} \to \mathcal{H} \) are two one-sided maximal edge-coverings, i.e. \( \mathcal{R}_-^o[h_s] \times \mathcal{H}^o[h_s] \) is left-maximal and \( (\mathcal{G}^o[g_t] \times \mathcal{R}_+^o[g_t])_{g_t \in \mathcal{G}_s} \) is right-maximal.

\(^3\)Interior operators are also known as co-closure operators: monotone, idempotent and co-extensive i.e. \( \text{in}(Y) \subseteq Y \).
Observe we cannot in general pass to maximal bicliques without changing the morphism’s domain/codomain. ■

**Definition 2.2.5 (JSLf).** A join-semilattice is a set with a binary operation $\lor$ and a nullary operation $\bot$, satisfying:

\[
\bot \lor x = x = \bot \lor x \\
x \lor (y \lor z) = (x \lor y) \lor z \\
x \lor y = y \lor x \\
x \lor x = x.
\]

We write them $S = (S, \lor_S, \bot_S)$ where $S$ is the underlying set, $\lor_S : S \times S \to S$ is a function and $\bot_S \in S$. A join-preserving morphism $f : S \to T$ is a function $f : S \to T$ such that $f(s_1 \lor_S s_2) = f(s_1) \lor_T f(s_2)$ and $f(\bot_S) = \bot_T$. Finally, $\text{JSL}_f$ is the category of finite join-semilattices and join-preserving morphisms. ■

**Example 2.2.6 (Clarifying join-semilattices).**

1. Join-semilattices are precisely the commutative and idempotent monoids. Consequently each $S$ has a Cayley-representation as endofunctions $(- \lor_S s : S \to S)_{s \in S}$ closed under functional composition.

2. More importantly, the join-semilattices are precisely the partially-ordered sets with all finite suprema. The binary operation $\lor_S$ is the binary join, $\bot_S$ is the bottom element. Inductively, $\lor_S X$ exists for all finite $X \subseteq S$.

3. The finite join-semilattices are precisely the finite bounded lattices: the finite partially-ordered sets with all finite suprema and infima. Indeed, every finite join-semilattice is complete i.e. has all joins, hence has all meets too. That is, $\lor_S X$ exists for any finite subset $X \subseteq S$.

4. By (3), each finite $S := (S, \lor_S, \bot_S)$ can be flipped yielding the order-dual join-semilattice $S^{\text{op}} := (S, \land_S, \top_S)$.

5. The join-semilattice isomorphisms are precisely the bijective join-semilattice morphisms. They are also precisely the order-isomorphisms between the underlying posets i.e. bijections preserving and reflecting the ordering. For finite join-semilattices they are precisely the bounded lattice isomorphisms by (3).

6. Let $2 = ((0, 1), \lor_2, 0)$ be the two element set with ordering $0 \leq_2 1$. Modulo isomorphism there is only one join-semilattice with two elements. ■

Each binary relation $\mathcal{G}$ induces two isomorphic join-semilattices: the in\$G\$-fixpoints $(O(\mathcal{G}), \lor, \varnothing)$ and the cl\$G\$-fixpoints $(C(\mathcal{G}), \lor_{\text{cl}\mathcal{G}}, \text{cl}\mathcal{G}(\varnothing))$. The latter’s join constructs the closure of the union, whereas its meet is simply intersection. This situation is well-known within the area of Formal Concept Analysis.

**Theorem 2.2.7 (Bounded lattice isomorphism of a bipartitioned graph).** For any $\mathcal{G} \in \mathcal{G}_\times \mathcal{G}_t$, we have the isomorphism:

\[
\alpha_\mathcal{G} : (C(\mathcal{G}), \lor_{\text{cl}\mathcal{G}}, \text{cl}\mathcal{G}(\varnothing)) \to (O(\mathcal{G}), \lor, \varnothing) \\
\alpha_\mathcal{G}(X) := \mathcal{G}[X] \\
\alpha_\mathcal{G}^{-1}(Y) := \mathcal{G}^i(Y).
\]

*Proof.* See background paper i.e. Lemma 4.2.5, The bounded lattices of $\mathcal{G}$-open/closed sets and their irreducibles. ■

**Note 2.2.8 (More about join-semilattices).**

1. **Join and meet-irreducibles.**

   Fix a join-semilattice $(S, \lor_S, \bot_S)$. An element $s \in S$ is join-irreducible if whenever $s = \lor_S X$ for finite $X \subseteq S$ we have $s \in X$. They are denoted $J(S) \subseteq S$. Likewise $s \in S$ is meet-irreducible if whenever $s$ is a finite meet $\land_S X$ then $s \in X$; they are denoted $M(S) \subseteq S$.

2. **Adjoint morphisms.**

   Each $\text{JSL}_f$-morphism $f : S \to T$ has an adjoint $f_\ast : T^{\text{op}} \to S^{\text{op}}$ defined $f_\ast(t) := \lor_S \{s \in S : f(s) \leq_T t\}$. It is uniquely determined by the adjoint relationship $f(s) \leq_T t \iff s \leq_S f_\ast(s)$, and preserves all finite meets in $T$. We’ve already seen examples i.e. $\mathcal{R}^i = (\mathcal{R}^i : (\mathcal{PR}_s, \lor, \varnothing) \to (\mathcal{PR}_t, \lor, \varnothing))_\ast$.

3. **Self-duality of $\text{JSL}_f$.**

   Adjoint morphisms actually define an equivalence functor $(-)_\ast : \text{JSL}_f^{\text{op}} \to \text{JSL}_f$ where $S_\ast := S^{\text{op}}$ is the order-dual join-semilattice and $f_\ast$ is the adjoint morphism. It is witnessed by the natural isomorphism $\lambda : \text{Id}_{\text{JSL}_f} \Rightarrow (-)_\ast \circ ((-)_\ast)^{\text{op}}$ where $\lambda_S := \text{id}_S$. 

6
4. **Monos and epis.**

The \(JSL_f\)-monomorphisms are precisely the injective ones and the epimorphisms are precisely the surjective ones. The latter situation is unlike the case of distributive lattices where \(3 \to 2 \times 2\) is epic. Injective \(JSL_f\)-morphisms \(f\) are also order-embeddings i.e. \(f(s_1) \leq_f f(s_2) \iff s_1 \leq_S s_2\). Generally speaking, injective monotone functions needn’t be order-embeddings e.g. take a bijection from a 2-antichain to a 2-chain.

**Note 2.2.9** (Irreducibles).

1. A bottom element is the empty join and hence never join-irreducible. The top element of a finite join-semilattice is the empty meet, hence never meet-irreducible.

2. The join-irreducibles of \((PX, \cup, \emptyset)\) are the singleton sets \(\{x\}\), the meet-irreducibles their relative complements.

3. Each element of a join-semilattice is the join of those join-irreducibles below it. In fact, \(J(S) \subseteq S\) is the minimal subset generating \(S\) under joins. Order dually, \(M(S) \subseteq S\) is the minimal subset generating \(S\) under meets.

4. Finite distributive lattices \(S\) are determined by their subposet of join-irreducibles. That is, they are isomorphic to the downwards-closed subsets of \((J(S), \leq_S)\), equipped with union (binary join) and intersection (binary meet). Every join-irreducible is actually join-prime i.e. \(j \leq_S \bigvee_S S \iff \exists s \in S. j \leq_S s\).

5. For finite distributive lattices \(S\), the subposet of join-irreducibles is order-isomorphic to the subposet of meet-irreducibles via \(\tau_S : J(S) \to M(S)\) with action \(j \mapsto \bigvee_S \bar j\) and inverse \(m \mapsto \bigwedge_S \bar m\). This fails for non-distributive lattices.

We now have enough structure to define the functorial translation between relations and algebras.

**Definition 2.2.10** (The equivalence functors).

1. \(\text{Open} : \text{Dep} \to JSL_f\) constructs the semilattice of \(G\)-open sets:

\[
\text{Open}G := (O(G), \cup, \emptyset) \quad \text{Open}R := \lambda Y. \mathcal{R}_+^\ast[Y].
\]

2. \(\text{Pirr} : JSL_f \to \text{Dep}\) constructs Markowsky’s poset of irreducibles [Mar75].

\[
\text{Pirr}S := \{j \in J(S) \times M(S) \mid f(j) \not\leq_1 m\} \quad \text{Pirr}f(j, m) : \iff f(j) \not\leq_1 m
\]

\[
(\text{Pirr})_\ast(j_1, j_2) : \iff j_2 \not\leq_1 f(j_1) \quad (\text{Pirr})_\ast(m_1, m_2) : \iff f_\ast(m_1) \leq_S m_2,
\]

where \(\text{Pirr}f\)’s components are also described above.

\(\text{Open}G\) is the inclusion-ordered set of neighbourhoods \(G[X]\) of the lower bipartition i.e. particular subsets of the upper bipartition. In the other direction, \(\text{Pirr}S\) is the domain/codomain restriction of \(\not\leq_S \subseteq S \times S\) to join/meet-irreducibles respectively. We have extended the concept studied by Markowsky to morphisms.

**Note 2.2.11** (Concerning \(\text{Open}\)’s action on morphisms). Given \(\mathcal{R} : G \to H\) we defined \(\text{Open}\mathcal{R}\) as \(\lambda Y. \mathcal{R}_+^\ast[Y]\). It may equivalently be defined \(\lambda Y. \mathcal{R}_+^\ast \circ G^\ast(Y)\) i.e. one needn’t compute the maximal witness \(\mathcal{R}_+\). It may also be defined \(\lambda Y. \mathcal{R}_u[X]\) where \(\mathcal{R}_u \subseteq \mathcal{R}_+\) is any upper witness, since \(\mathcal{R}_+[G[X]] = G; \mathcal{R}_u[X] = G; \mathcal{R}_u[X] = \mathcal{R}_u[G[X]]\).

**Example 2.2.12** (Semilattices as binary relations).

1. **Boolean lattices correspond to identity relations.**

   Observe \(\text{Open}\Delta_X = (PX, \cup, \emptyset)\) for any finite set \(X\). Applying \(\text{Pirr}\) yields the bijection \(\{x\} \mapsto \bar x\), which is bipartite isomorphic to \(\Delta_X\) and hence \(\text{Dep}\)-isomorphic.

2. **Distributive lattices correspond to order relations.**

   Given any order-relation \(\leq_S \subseteq P \times P\) then \(\text{Open} \leq_P\) consists of all upwards closed subsets of \(P\) ordered by inclusion. Since they are closed under unions and intersections, \(\text{Open} \leq_P\) is a distributive lattice. Conversely if \(S\) is distributive one can show \(\text{Pirr}S; \tau_S^\ast = \leq_{\mathcal{S}^\ast;} |J(S) \times J(S)|\), using notation from Note 2.2.9.5. See the background paper Lemma 2.2.3.14 ‘Standard order-theoretic results’. Then \(\text{Pirr}S\) is bipartite isomorphic to an order-relation and hence \(\text{Dep}\)-isomorphic too.
3. **Partition lattices represented via functional composition.**

Recall the inclusion-ordered lattice \( \mathbb{E} \mathbb{R}(X) \) of equivalence relations on a finite non-empty set \( X \). Meets are intersections, whereas joins are constructed by taking the transitive closure of the union. Viewed as a join-semilattice, there is a natural binary relation \( G \) such that \( \text{Open} G \cong \mathbb{E} \mathbb{R}(X) \):

\[
G \subseteq \text{Set}(2,X) \times \text{Set}(X,2) \quad G(f,g) : \iff g \circ f = id_2
\]

where \( 2 := \{0, 1\} \) and \( \text{Set}(A,B) \) is the set of functions from \( A \) to \( B \). Notice we have:

\[
|X|^2 = |G_x| \geq |\mathbb{E} \mathbb{R}(X)| = \binom{|X|}{2} \quad 2^{|X|} = |G_t| = |\mathbb{E} \mathbb{R}(X)|.
\]

4. **A concrete example.**

Let \( P_6 \) be the path of edge-length 6 with vertices \( \{0, \ldots, 6\} \). One of its bipartitions amounts to \( G \subseteq \{1, 3, 5\} \times \{0, 2, 4, 6\} \) where \( G(x, y) : \iff |x - y| = 1 \). Applying \( \text{Open} \) yields:

\[
\begin{array}{ccc}
\{0, 2, 4, 6\} & \xrightarrow{f} & \{2, 4, 6\} \\
\{0, 2\} & \xrightarrow{} & \{2, 4\} \\
\{\emptyset\} & \xrightarrow{} & \{4, 6\}
\end{array}
\]

this being the smallest join-semilattice \( S \) such that \( |J(S)| < |M(S)| \).

**Example 2.2.13** (\( \text{JSL}_f \)-morphisms as \( \text{Dep} \)-morphisms).

Given \( f : S \to T \) we have the following \( \text{Dep} \)-morphism of type \( \varepsilon_S \to \varepsilon_T \),

\[
\begin{array}{c}
S \xrightarrow{f^*} T \\
\varepsilon_S \downarrow \quad \downarrow \varepsilon_T \\
S \xrightarrow{f} T
\end{array}
\]

via the adjoint relationship \( f(s) \varepsilon_T t \iff s \varepsilon_S f_*(t) \) in contrapositive form. \( \text{Pirrf} \) arises by restricting the domain/codomain and passing to the maximum witnesses. Importantly, there is an equivalence functor \( \text{Nleq} : \text{JSL}_f \to \text{Dep} \) with \( \text{Nleq} S := \varepsilon_S \) and \( \text{Nleq}(s, t) : \iff f(s) \varepsilon_T t \). In a precise sense, \( \text{Pirr} \) is the smallest restriction possible.

We now explicitly describe the equivalence between semilattices and graphs, including the relevant component relations. From one perspective we represent each \( S \) as inclusion-ordered subsets of \( M(S) \); from another we show each bipartitioned graph is \( \text{Dep} \)-isomorphic to its reduction – a kind of union-free normal form.

**Theorem 2.2.14** (Categorical equivalence). \( \text{Open} : \text{Dep} \to \text{JSL}_f \) and \( \text{Pirr} : \text{JSL}_f \to \text{Dep} \) define an equivalence of categories via natural isomorphisms:

\[
\begin{array}{ll}
\text{rep} : \text{id}_{\text{JSL}_f} & \Rightarrow \text{Open} \circ \text{Pirr} \\
\text{rep}_S & := \lambda s \in S \{ m \in M(S) : s \varepsilon_S m \} \\
\text{rep}_S^{-1} & := \lambda Y \downarrow M(S) \downarrow Y \\
\text{red} : \text{id}_{\text{Dep}} & \Rightarrow \text{Pirr} \circ \text{Open} \\
\text{red}_g & := \{(g_s, Y) \in G_x \times M(\text{Open} G) : G[g_s] \varepsilon Y\} \\
\text{red}_g^{-1} & := \varepsilon \subseteq J(\text{Open} G) \times G_t
\end{array}
\]

where \( \text{red}_g \) and its inverse have associated component relations:

\[
\begin{array}{ll}
(\text{red}_g)_- & := \{(g_s, X) \in G_x \times J(\text{Open} G) : X \subseteq G[g_s]\} \\
(\text{red}_g)_+ & := \varepsilon \subseteq M(\text{Open} G) \times G_t \\
(\text{red}_g^{-1})_- & := \{(X, g_s) \in J(\text{Open} G) \times G_s : G[g_s] \subseteq X\} \\
(\text{red}_g^{-1})_+ & := \{(g_t, Y) \in G_t \times M(\text{Open} G) : \text{in}_G(G_t) \subseteq Y\}
\end{array}
\]
Proof. See background paper i.e. Theorem 4.2.10, Dep is equivalent to JSL$_f$. 

So rep$_S$ represents a join-semilattice as neighbourhoods of the relation Pirr$S$. Its inverse is relatively clear: every element arises uniquely as the meet of those meet-irreducibles above it. Concerning the other natural isomorphism,

red$_G$ reduces a bipartitioned graph $G$ by discarding vertices whose neighbourhood is a union of other vertices’ neighbourhoods in a canonical manner.

It is worth clarifying the above statement. Firstly,

$$J(PirrOpenG) \subseteq \{ G[g_s] : g_s \in G_s \} \quad M(PirrOpenG) \subseteq \{ in_G(\overline{r}) : g_t \in G_t \}$$

because the supersets join/meet-generate Open$G$ respectively. The join-irreducible $G[g_s]$’s correspond to those $g_s$ whose neighbourhood is not a union of others. Less obviously the meet-irreducible in$_G(\overline{r})$’s correspond to those $g_t$ whose neighbourhood $\overline{G}[g_t]$ is not a union of others:

\[
\begin{align*}
in_G(\overline{r}) &= in_G(g_t) \land in_G(g_2) \iff in_G(\overline{r}) = in_G(in_G(g_t) \cap in_G(g_2)) \quad \text{(definition of } \land \text{in}) \nonumber \\
G^1(\overline{r}) &= G^1(in_G(g_t) \cap in_G(g_2)) \quad \text{(by Lemma 2.2.7)} \nonumber \\
G^1(\overline{r}) &= G^1(in_G(g_t) \cap in_G(g_2)) \quad \text{($G^1$ preserves } \cap \text{)} \nonumber \\
G^1(\overline{r}) &= G^1(in_G(g_t) \cap in_G(g_2)) \quad (\iff) \nonumber \\
\neg_{G^1} \circ G^1(\overline{r}) &= \neg_{G^1} \circ G^1(in_G(g_t) \cap in_G(g_2)) \quad (\neg_{G^1} \text{ is bijective}) \nonumber \\
\overline{G}[g_t] &= \overline{G}[g_t] \cup \overline{G}[g_2] \quad (\neg \downarrow \neg) \nonumber
\end{align*}
\]

Then finally we have:

$$G[g_s] \notin in_G(\overline{r}) \quad \text{(1)} \quad g_s \notin G^1 \circ G^1(\overline{r}) \quad \text{(2)} \quad G[g_s] \notin \overline{r} \quad \text{for any } G \in G_s \times G_t$$

where (1) and (3) follow by the adjoint relationship in Lemma 2.2.3.1. So reduction discards ‘degenerate’ vertices and every relation is Dep-isomorphic to its reduction. This is a form of union-freeness. Importantly:

**Proposition 2.2.15** (Reduction preserves bipartite dimension). dim$(G) = dim(PirrOpenG)$ for any $G \in G_s \times G_t$.

**Example 2.2.16** (Reduction and bipartite dimension).

1. Isolated points have empty neighbourhoods and so are ‘discarded’ by red$_G$. The bipartite dimension is preserved because it is defined in terms of edges.

2. If two points have the same neighbourhood, only one representative occurs in the reduction Pirr$OpenG$. The square $C_4$ arises as $G \in \{ 0, 2 \} \times \{ 1, 3 \}$ where $G[0] = G[2] = G$. Its reduction $\{ G \} \times \{ G \}$ is bipartite graph isomorphic to a single edge $P_1$. Concerning bipartite dimension, if two vertices have the same neighbourhood we may assume they reside in the same bicliques.

3. A vertex’s neighbourhood can be a non-degenerate union of others e.g. $G[2] = G[0] \cup G[4]$ below:

```
0 1 2 3 4
```

Applying red$_G$ we obtain two disjoint edges. This preserves the bipartite dimension because we can add 2 to each biclique involving 0 or 4. This method extends to the general cases $G[x] = G[X]$ and $\overline{G}[y] = \overline{G}[Y]$.

4. Suppose $G$ is a disjoint union of bicliques i.e. $G = \bigcup_{i,j} X_i \times Y_j$ where $X_i \cap X_j = \emptyset = Y_i \cap Y_j$ whenever $i \neq j$. Then reduction is a special case of Example 2.1.8.3 i.e. a preorder whose quotient poset is discrete.

5. Example 2.2.12.3 described a natural bipartitioned graph which was not reduced. In the automata-theoretic section we’ll see many important examples.
We described the self-duality of \( \text{JSL}_f \) in Note 2.2.8.3. In \( \text{Dep} \), this self-duality simply takes the converse relation on both objects and morphisms. Furthermore, the associated component relations are simply swapped.

**Theorem 2.2.17** (Self-duality). We have the self-duality functor \((-)^\vee : \text{Dep}^{\text{op}} \to \text{Dep}:

\[
\mathcal{G}^\vee := \mathcal{G} \\
\begin{array}{c}
\mathcal{R} : \mathcal{G} \to \mathcal{H} \\
(\mathcal{R}^{\text{op}})^\vee := \mathcal{R} : \mathcal{H} \to \mathcal{G}
\end{array}
\]

with witnessing natural isomorphism \( \alpha : \text{id}_{\text{Dep}} \Rightarrow (\mathcal{R}^{\text{op}})^\vee \) defined \( \alpha_{\mathcal{G}} := \text{id}_{\mathcal{G}} = \mathcal{G} \).

**Proof.** See background paper i.e. Theorem 4.1.13, Self-duality of \( \text{Dep} \).

There is also an important natural isomorphism connecting the two self-dualities.

**Theorem 2.2.18** (Self-duality transfer).

1. If \( \partial : (\mathcal{R}) \circ \text{Open}_{\text{op}} = \text{Open} \circ (\mathcal{R})^\vee \) with \( \partial_{\mathcal{G}} := \lambda Y, \mathcal{G}[\mathcal{T}] \) is a natural isomorphism with inverse \( \bar{\partial}_{\mathcal{G}} := \lambda Y, \mathcal{G}[\mathcal{T}] \).

   In fact, if \( \mathcal{R} : \mathcal{G} \to \mathcal{H} \) is a Dep-morphism then \( (\text{Open}\mathcal{R}) \circ \mathcal{H} \circ \mathcal{H}^\vee \circ \mathcal{H}^\vee \circ \mathcal{R}^\vee \circ \mathcal{Y} \).

2. \( \lambda : (\mathcal{R})^\vee \circ \text{Pirr}_{\text{op}} = \text{Pirr} \circ (\mathcal{R})^\vee \), where \( \lambda_S := \text{id}_{\text{Pirr}_S} = \text{Pirr}_S \) is a self-inverse natural isomorphism.

**Proof.**

1. See background paper Theorem 4.6.7 i.e. \( \partial \) defines a natural isomorphism.

2. Given any \( \text{JSL}_f \)-morphism \( f : S \to \mathcal{T} \) we need to establish the following square commutes:

\[
\begin{array}{c}
\text{Pirr}(\mathcal{S})^\vee \\
\downarrow \text{id}_\mathcal{S} \\
\text{Pirr}(\mathcal{S}^{\text{op}})
\end{array}
\]

Indeed for any \( s \in S \) and \( t \in T \),

\[
\text{Pirr}(f)(t,s) : \Longleftrightarrow f(t) \notin s \Longleftrightarrow s \notin f(t) \Longleftrightarrow f(s) \notin t \Longleftrightarrow \text{Pirr}f(s,t) \Longleftrightarrow (\text{Pirr}f)^\vee (t,s)
\]

via the usual adjoint relationship.

**Note** that \( \text{JSL}_f \) has enough projectives, using category-theoretic parlance.

**Proposition 2.2.19** (\( \text{JSL}_f \) and \( \text{Dep} \) have enough projectives).

Let \( Z \) be a finite set and \( S \) a finite join-semilattice.

1. \( \text{Open}_{\mathcal{S}} = (\mathcal{P}Z, \cup, \emptyset) \) is the free join-semilattice on \( |Z| \)-generators.

2. \( \varepsilon_S : \text{Open}_{\mathcal{J}(S)} \to S \) where \( \varepsilon_S(X) := \forall \subseteq X \) is surjective and extends \( J(S) \to S \). Correspondingly, \( \text{Dep} \) has epimorphisms \( \mathcal{G} : \Delta_{\mathcal{G}} \to \mathcal{G} \).

3. Given \( f : \text{Open}_{\mathcal{G}} \to \mathcal{T} \) and surjective \( g : S \to \mathcal{T} \) then \( f = g \circ g \) where \( g : \text{Open}_{\mathcal{S}} \to S \) extends \( \lambda z, q \in f(\{z\}) \).

Since the self-duality preserves freeness, \( \text{JSL}_f \) has enough injectives too. The witnessing embeddings \( \varepsilon_S := \varepsilon \circ \text{rep}_S : S \to \text{Open}_{\mathcal{J}(S)} \) first represent and then include into a powerset. In \( \text{Dep} \) they amount to monomorphisms \( \mathcal{G} : \mathcal{G} \to \Delta_{\mathcal{G}} \).

Concerning both projectivity and injectivity, there is an important special case involving endomorphisms.

**Corollary 2.2.20** (Endomorphism representations).

The \( \text{JSL}_f \)-definitions below commute for any \( S \)-endomorphism \( \delta \),

\[
\begin{array}{c}
S \xrightarrow{\delta} S \\
\varepsilon_S \downarrow \downarrow \varepsilon_S \\
\text{Open}_{\mathcal{J}(S)} \xrightarrow{\text{Pirr}\delta} \text{Open}_{\mathcal{J}(S)}
\end{array}
\]

Concerning \( \text{Dep} \), \( \mathcal{R} : \mathcal{G} \to \mathcal{G} \) induces both \( \mathcal{R}_+ : \Delta_{\mathcal{G}} \to \Delta_{\mathcal{G}} \) and \( \mathcal{R}_- \Delta_{\mathcal{G}} \to \Delta_{\mathcal{G}} \).
Next, recall \textbf{PirrS} restricts to the join/meet-irreducibles of \( S \). It turns out one can instead pass to any join/meet generators. Roughly speaking, one can extend the domain/codomain of \textbf{PirrS} and \textbf{Pirrf} in the ‘obvious’ way.

\begin{proposition}[Dep generator isomorphisms] \label{proposition:2.2.21} \( \text{Let } f : S \to \mathbb{T} \text{ be a join-semilattice morphism, } J_S, M_S \subseteq S \text{ be join/meet generators for } S, \text{ and } J_T, M_T \subseteq \mathbb{T} \text{ be join/meet generators for } \mathbb{T}. \)

1. We have the Dep-isomorphism \( \mathcal{I}_S : \mathcal{S} |_{J_S \times M_S} \to \text{PirrS}, \)

\[
\mathcal{I}_S := \mathcal{S} |_{J_S \times M_S} (\mathcal{I}_S)_{(x, y)} : \iff j \leq x \quad (\mathcal{I}_S)^+_{(m, y)} : \iff m \leq y
\]

2. The following Dep-diagram commutes where \( \mathcal{I}_f(s, t) : \iff f(s) \mathcal{S} t: \)

\[
\begin{array}{ccc}
\mathcal{S} |_{J_T \times M_T} & \xrightarrow{\mathcal{I}_f} & \text{Pirr} \mathbb{T} \\
\mathcal{I}_f \downarrow & & \downarrow \text{Pirrf} \\
\mathcal{S} |_{J_S \times M_S} & \xrightarrow{\mathcal{I}_S} & \text{Pirr} S \\
\end{array}
\]

\end{proposition}

\begin{example}[Applying Proposition 2.2.21] \label{example:2.2.22} \( \text{Applying Proposition 2.2.21 we see that Example 2.2.13 is essentially Pirrf.} \)

\end{example}

So far we’ve seen that Dep is well-behaved w.r.t. bipartite dimension. However, aside from that, connections with graph theory have been a bit thin on the ground. So before proceeding to the automata-theoretic constructions we mention some additional relationships.

\begin{example}[Further graph-theoretic connections] \label{example:2.2.23} \( \text{(Further graph-theoretic connections).} \)

1. \textit{Discarding vertices.}

Let \( G \) be a reduced relation. Discarding a vertex \( g_s \in G_s \) in the lower bipartition amounts to generating a sub join-semilattice \( (J(\mathbb{S}) \setminus \{j\})_S \). Discarding \( g_t \in G_t \) amounts to constructing a quotient \( (\mathbb{M}(\mathbb{S}) \setminus \{m\})_{\mathbb{S}op} \).

2. \textit{Kronecker product over boolean semiring.}

One can combine binary relations via \( G \odot H ((g_s, h_s), (g_t, h_t)) : \iff G(g_s, g_t) \land H(h_s, h_t) \) i.e. the Kronecker product over the boolean semiring \([ Wat01]\). It defines a functor \( - \odot - : \text{Dep} \times \text{Dep} \to \text{Dep} \) whose corresponding join-semilattice functor is the tight tensor product \( S \otimes_1 \mathbb{T} := \mathbb{T}[\mathbb{S}op, \mathbb{T}] \). To explain briefly,

(a) \( \mathbb{T}[-, -] : \text{JSL}_f \text{op} \times \text{JSL}_f \to \text{JSL}_f \) restricts the usual hom-functor to morphisms which factor through a boolean lattice. The join-semilattice structure on morphisms is defined pointwise.

(b) There is a universal property w.r.t. bilinearity via a natural isomorphism \( ut : \mathbb{T}[- \otimes_1 -, -] \Rightarrow \mathbb{T}[-, \mathbb{T}[-, -]] \).

(c) The tight tensor product is distinct from the tensor product \([ GW05]\); they coincide on distributive lattices.

3. \textit{Extension to non-bipartite graphs.}

We’ve seen that reduced relations correspond to finite join-semilattices. This categorical equivalence can be extended to \textit{reduced undirected graphs} versus \textit{finite De Morgan algebras} i.e. bounded lattices with an order-reversing involution where distributivity is not assumed.

(a) By undirected graph we mean a symmetric relation \( E = \tilde{E} \) i.e. a standard undirected graph where self-loops are now permitted.

(b) The algebras may be axiomatised by extending join-semilattices with a unary operation satisfying \( \sigma(x \lor y) \leq \sigma(x) \) and \( \sigma(\sigma(x)) = x \). A morphism is a join-semilattice morphism preserving \( \sigma \).

(c) Given \( (V, E) \) we construct the De Morgan algebra \( \partial_E : \text{OpenE} \to (\text{OpenE})^{op} \) where \( \partial_E(X) := E[X] \). Given a De Morgan algebra \( \sigma : \mathbb{S} \to \mathbb{S}^{op} \) we construct the undirected graph \((J(\mathbb{S}), \text{Pirr} \sigma)\).

\end{example}
3 Dependency Automata

3.1 From Nondeterministic to Dependency Automata

Definition 3.1.1 (Nondeterministic finite automaton).

1. A nondeterministic finite automaton (or nfa) is a tuple $\mathcal{N} = (I, Z, \mathcal{N}_a, F)$ where:
   - $Z$ is a finite set,
   - $I, F \subseteq Z$ are subsets, and
   - $\mathcal{N}_a \subseteq Z \times Z$ for each $a \in \Sigma$.

The elements of $Z$, $I$, $F$ are called states, initial states and final states respectively. Each $\mathcal{N}_a$ is called the $a$-transition relation. We often reuse the symbol denoting the nfa (e.g. $\mathcal{N}$) to denote the transitions (e.g. $\mathcal{N}_a$).

We may also denote the states, initial states and final states by $Z$, $I, F$ respectively.

2. For $w \in \Sigma^*$ inductively define $\mathcal{N}_w$ as $\mathcal{N}_e := \Delta_Z$, $\mathcal{N}_a := \mathcal{N}_a; \mathcal{N}_w$. Then we say $\mathcal{N}$ accepts the language:

$$L(\mathcal{N}) := \{ w \in \Sigma^* : \mathcal{N}_w[I] \cap F \neq \emptyset \}.$$

3. Constructions on nondeterministic automata $\mathcal{N} = (I, Z, \mathcal{N}_a, F)$.

a. Given $S \subseteq Z$ then $\mathcal{N}_{@S} := (S, Z, \mathcal{N}_a, F)$ is the nfa with its initial states changed to $S$. Notice that $L(\mathcal{N}_{@S}) = \bigcup_{z \in S} L(\mathcal{N}_{@z})$.

b. $\mathcal{N}$’s reverse nfa is:

$$\text{rev}(\mathcal{N}) := (F, Z, (\mathcal{N}_a)^\sim, I)$$

and accepts the reverse language $(L(\mathcal{N}))^\sim$.

c. There are various concepts relating to reachability:

$$\text{rs}(\mathcal{N}) := \{ \mathcal{N}_w[I] : w \in \Sigma^* \}$$

$$\text{reach}(\mathcal{N}) := \bigcup \text{rs}(\mathcal{N}) \subseteq Z$$

$$\text{rsc}(\mathcal{N}) := (\{ I \}, \text{rs}(\mathcal{N}), \lambda X. \mathcal{N}_a[X], \{ X \in \text{rs}(\mathcal{N}) : X \cap F \neq \emptyset \})$$

That is, $\text{rs}(\mathcal{N})$ consists of $\mathcal{N}$’s reachable subsets, $\text{reach}(\mathcal{N})$ consists of $\mathcal{N}$’s reachable states and finally $\text{rsc}(\mathcal{N})$ is the famous reachable subset construction. The latter is a dfa – see Definition 3.1.3 below.

d. If $I \subseteq X \subseteq Z$ and $\mathcal{N}_a[X] \subseteq X$ (for $a \in \Sigma$) the nfa $\mathcal{N} \cap X := (I, X, \mathcal{N}_a|_{X \times X}, F \cap X)$ accepts $L(\mathcal{N})$. Then:

$$\text{reach}(\mathcal{N}) := \text{reach}(\mathcal{N}) \cap \mathcal{N}$$

is the reachable part of $\mathcal{N}$.

e. The coreachable part of $\mathcal{N}$ also accepts $L(\mathcal{N})$:

$$\text{coreach}(\mathcal{N}) := \text{rev}(\text{reach}(\text{rev}(\mathcal{N}))).$$

f. An nfa isomorphism $f : \mathcal{M} \to \mathcal{N}$ is a bijection $f : Z_M \to Z_N$ which preserves and reflects the initial states, the final states, and also the transitions. That is:

$$z \in I_M \iff f(z) \in I_N \iff M_a(z_1, z_2) := N_a(f(z_1), f(z_2)) \iff z \in F_M \iff f(z) \in F_N$$

for each $z, z_1, z_2 \in Z$ and $a \in \Sigma$. We may also write $\mathcal{M} \cong \mathcal{N}$.

g. Each nfa has an associated join-semilattice of accepted languages by varying the initial states:

$$\text{langs}(\mathcal{N}) := (\{ L(\mathcal{N}_{@S}) : S \subseteq Z \}, \cup, \emptyset).$$

Equivalently, $\text{langs}(\mathcal{N}) := \text{langs}(\det(\text{dep}(\mathcal{N})))$ is the join-semilattice of languages accepted by the full subset construction – see Definition 3.4.1.2 and Note 3.2.7.

4. We say $\mathcal{N}$ is state-minimal if there is no nfa accepting $L(\mathcal{N})$ with strictly fewer states.
Example 3.1.2 (Some small nfas).

1. \( L = a(b + c) + b(a + c) + c(a + b) \) from [ADN92] is a language with two state-minimal nfas.\(^4\)

2. \( L = a + aa \) from [LRT09] is an example of a language which is not ‘biresidual’ [Tam10, LRT09]. It has 5 state-minimal nfas:

3. \( L = (ab)^* + (abc)^* \) has a unique state-minimal nfa shown below left.

4. Consider the language \( L_n = (a + b)^*a(a + b)^n \) for any \( n \geq 0 \). If \( n = 3 \) then the state-minimal nfa with the greatest (resp. least) number of transitions is shown below on the left (resp. right).

Every regular language has a unique state-minimal partial deterministic machine, shown for this \( L \) above right.

We now recall deterministic finite automata and their associated canonical construction i.e. the state-minimal deterministic machine for a regular language.

Definition 3.1.3 (Deterministic finite automaton).

1. A deterministic finite automaton (or dfa) is an nfa \( (I, Z, N_a, F) \) where \( |I| = 1 \) and each \( N_a \) is a function. We may write them as \( \delta = (i, Z, \delta_a, F) \) where \( i \in Z \). For each \( w \in \Sigma^* \) we inductively define the endofunction \( \delta_w : Z \rightarrow Z \) as follows: \( \delta_e := id_Z \) and \( \delta_{ua} := \delta_a \circ \delta_u \) for each \( (u, a) \in \Sigma^* \times \Sigma \).

2. Given a dfa \( \delta = (z_0, Z, \delta_a, F) \) accepting \( L \) and \( u \in \Sigma^* \),

\[
L(\delta_{\delta_a(i)}) = \{ z \in Z^* : \delta_w(z) = z_0 \ \text{for some} \ w \in \Sigma^* \}.
\]

In other words, the unique \( u \)-successor of \( z_0 \) accepts \( u^{-1}L \). The latter set is the left word quotient of \( L \) by \( u \) and is also known as the Brzozowski derivative [Brz64].

\(^4\)Here, \( \overset{b,c}{\longrightarrow} \) indicates there is one \( b \)-labelled edge and another parallel \( c \)-labelled one. Initial states are indicated by 1, final states by 0.
3. Fix any regular $L \subseteq \Sigma^*$ and let $\mathcal{LW}(L) := \{ u^{-1}L : u \in \Sigma^* \}$ be $L$’s left word quotients. Then:

$$\text{dfa}(L) := (L, \mathcal{LW}(L), \lambda X. a^{-1}X, \{X \in \mathcal{LW}(L) : \varepsilon \in X \})$$

is the state-minimal dfa accepting $L$. It is well-defined because $a^{-1}(u^{-1}L) = (ua)^{-1}L$.

4. A dfa morphism $f : (x_0, X, \gamma_a, F_X) \rightarrow (y_0, Y, \delta_a, F_Y)$ is a function $f : X \rightarrow Y$ such that for $a \in \Sigma$:

$$f \circ \delta_a = \gamma_a \circ f \quad f(x_0) = y_0 \quad f^{-1}(F_Y) = F_X.$$

The final condition asserts that the final states are both preserved and reflected, noting that $f^{-1} : \mathcal{P}Y \rightarrow \mathcal{P}X$ is the preimage function. Importantly, dfa morphisms always preserve the accepted language.

5. Each dfa $\delta := (z_0, Z, \delta_a, F)$ has a dfa of accepted languages:

$$\text{simple}(\delta) := (L, \text{langs}(\delta), \lambda X. a^{-1}X, \{X \in \text{langs}(\delta) : \varepsilon \in X \})$$

where $\text{langs}(\delta) := \{L(\delta_{az}) : z \in Z\}$. There is a surjective dfa morphism $\text{acc}_\delta : \delta \rightarrow \text{simple}(\delta)$ defined $\text{acc}_\delta(z) := L(\delta_{az})$ i.e. the acceptance map. The word simple is non-standard yet well-motivated: every surjective dfa morphism $f : \text{simple}(\delta) \rightarrow \gamma$ is bijective.

6. An ordered dfa $(p_0, P, \delta_a, F)$ consists of a partially ordered set $P = (P, \preceq_P)$ and a dfa $(p_0, P, \delta_a, F)$ whose deterministic transitions are respectively monotonic $\delta_a : P \rightarrow P$. An ordered dfa morphism is a dfa morphism between ordered dfas which is also monotonic.

Note 3.1.4 (Concerning dfa($L$)). State-minimal dfas are often introduced via Hopcroft’s algorithm. One takes the reachable part of a given dfa, afterwards identifying states accepting the same language. The latter uses Hopcroft’s partition refinement, essentially constructing the Myhill-Nerode congruence. There are two ‘representation independent’ ways of defining it: (1) as equivalence classes of the Myhill-Nerode congruence $\mathcal{MN}_L \subseteq \Sigma^* \times \Sigma^*$ for $L$, (2) as the left word quotients $u^{-1}L$ also known as Brzozowski derivatives [Brz64].

Note 3.1.5 (Concerning simple($\delta$)). The dfa simple($\delta$) has no more states than $\delta$. Each state $z$ of the latter accepts $L(\delta_{az})$ (by definition), as does the state $L(\delta_{az})$ in simple($\delta$). This construction is defined for dfas but not nfas. However, later we’ll introduce a related construction simple,$_p$(N) for each nfa $N$ – see Definition 4.2.4.

We now introduce dependency automata i.e. two nfas compatible w.r.t. a bipartitioned graph.

Definition 3.1.6 (Dependency automaton). A dependency automaton is a triple $(\mathcal{N}, \mathcal{G}, \mathcal{N}')$ where:

1. $\mathcal{G} \subseteq \mathcal{G}_s \times \mathcal{G}_t$ is a binary relation (bipartitioned graph).
2. $\mathcal{N} := (I_{\mathcal{N}}, \mathcal{G}_s, \mathcal{N}_a, F_{\mathcal{N}})$ is an nfa over the lower bipartition.
3. $\mathcal{N}' := (I_{\mathcal{N}'}, \mathcal{G}_t, \mathcal{N}_a', F_{\mathcal{N}'})$ is an nfa over the upper bipartition.
4. $\mathcal{N}_a; \mathcal{G} = \mathcal{G}; (\mathcal{N}_a')^\ast$ for each $a \in \Sigma$.
5. $F_{\mathcal{N}'} = \mathcal{G}[I_{\mathcal{N}'}]$ and $F_{\mathcal{N}} = \mathcal{G}'[I_{\mathcal{N}'}]$.

Condition (4) induces Dep-endomorphisms which we denote by $\mathcal{N}_a^\ast : \mathcal{G} \rightarrow \mathcal{G}$ for each $a \in \Sigma$. A dependency automaton $(\mathcal{N}, \mathcal{G}, \mathcal{N}')$ accepts the language $L(\mathcal{N}, \mathcal{G}, \mathcal{N}') := L(\mathcal{N})$ i.e. the language accepted by the lower nfa.

Each nfa induces a dependency automaton with only linear blowup.

Definition 3.1.7 (Nfa’s associated dependency automaton). Given an nfa $\mathcal{N}$ with states $Z$,

$$\text{dep}(\mathcal{N}) := (\mathcal{N}, \Delta_Z, \text{rev}(\mathcal{N}))$$

is its associated dependency automaton.

For well-definedness consider Definition 3.1.6 when $\mathcal{G} = \Delta_Z$. Then (4) amounts to taking the converse relation and (5) to swapping the initial/final states. So each nfa can be viewed as a dependency automaton. Very importantly, each regular language has an associated dependency automaton too.
Definition 3.1.8 (Canonical dependency automaton). Given a regular language \( L \subseteq \Sigma^* \),
\[
\text{dep}(L) := (\text{dfa}(L), \mathcal{DR}_L, \text{dfa}(L^*)) \quad \text{where} \quad \mathcal{DR}_L(u^{-1}L, v^{-1}L^*) : \iff uv^* \in L
\]
is the respective canonical dependency automaton.

Example 3.1.9 (\text{dep}(L)). If \( L = a + aa \) so \( L = L^* \) then \( \text{dfa}(L) = \text{dfa}(L^*) \) is \( i \xrightarrow{a} o \xrightarrow{a} o \) excluding the sink. The canonical dependency automaton takes the form:

![Dependency Automaton Diagram]

excluding the sink state from the top and bottom (which are isolated in \( \mathcal{DR}_L \)).

Lemma 3.1.10. \( \text{dep}(L) \) is a well-defined dependency automaton.

Proof. Concerning (4),
\[
(\lambda X. a^{-1}X) : \mathcal{DR}_L[u^{-1}L] = \mathcal{DR}_L[(ua)^{-1}L] = \{v^{-1}L^r : v \in \Sigma^*, uv^r \in L\}
\]
\[
\mathcal{DR}_L; (\lambda X. a^{-1}X)^* [u^{-1}L] = (\lambda X. a^{-1}X)^* [(v^{-1}L^r : v \in \Sigma^*, uv^r \in L)]
\]
\[
= \{v^{-1}L^r : v \in \Sigma^*, u(va)^* \in L\}
\]
\[
= \{v^{-1}L^r : v \in \Sigma^*, uv^r \in L\}.
\]

Concerning (5),
\[
v^{-1}L^r \in \mathcal{DR}_L[L] \iff v^r \in L \iff v \in L^r \iff \varepsilon \in v^{-1}L^r
\]
\[
u^{-1}L \in (\mathcal{DR}_L)^c[L^r] \iff u \in L \iff \varepsilon \in u^{-1}L.
\]

In both the classes of examples so far, the upper nfa accepts a word iff the lower nfa accepts its reverse. This situation holds generally for all dependency automata.

Lemma 3.1.11. If \((N, G, N')\) is a dependency automaton then \(L(N') = (L(N))^r\).

Proof. Since \(N_a : G = G; (N'_a)^r\) we have \(N'_a : G \to G\) and composing yields \(N'_w : G = G; (N'_w)^r\) for \(w \in \Sigma^*\). Then:
\[
w \in L(N') \iff N'_w[I_{N'}] \cap F_{N'} = \emptyset
\]
\[
\iff N'_w[I_{N'}] \notin \overline{F_{N'}} = G[I_{N'}] \quad \text{(by def.)}
\]
\[
\iff N'_w[I_{N'}] \notin \overline{G}'(I_{N'}) \quad \text{(-)}
\]
\[
\iff N'_w[I_{N'}] \notin I_{N'} \quad \text{(adjoints)}
\]
\[
\iff \overline{G}; N'_w[I_{N'}] \notin I_{N'} \quad \text{(see above)}
\]
\[
\iff N'_w[F_{N'}] \notin I_{N'} \quad \text{(by def.)}
\]
\[
\iff w \in L(\text{rev}(N))
\]
\[
\iff w \in (L(N))^r.
\]

Definition 3.1.12 (The category \text{aut}_{\text{Dep}}). Its objects are the dependency automata. Take any two of them:
\[
(M, F, M') \quad \text{where} \quad M = (I_M, F_s, M_a, F_M) \quad \text{and} \quad M' = (I'_M, F_{t}, M'_a, F'_{M})
\]
\[
(N, G, N') \quad \text{where} \quad N = (I_N, G_s, N_a, F_N) \quad \text{and} \quad N' = (I'_N, G_t, N'_a, F'_{N}).
\]

An \(\text{aut}_{\text{Dep}}\)-morphism \(\mathcal{R} : (M, F, M') \to (N, G, N')\) is a \text{Dep}-morphism \(\mathcal{R} : F \to G\) such that for each \(a \in \Sigma\),
\[
M_a : \mathcal{R} = \mathcal{R}; (N'_a)^r \quad \mathcal{R}[I_M] = F_{N'} \quad \mathcal{R}[I_{N'}] = F_M.
\]
Composition is inherited from \text{Dep}. The leftmost condition can be written \(M_a : \mathcal{R} = \mathcal{R} ; N'_a\).


Lemma 3.1.13. $\text{autDep}$ is a well-defined category.

Proof. Identity morphisms $id_{(N', G, N')}: = id_G = G$ are well-defined. Indeed, the conditions concerning dependency automata state precisely that $G$ is an $\text{autDep}$-endomorphism of $(N, G, N')$. It remains to verify that compatible $\text{autDep}$-morphisms are closed under Dep-composition. To this end, take a dependency automaton $(O, H, O')$ where $O = (I_O, H_O, O_A, F_O)$ and $O' = (I_{O'}, H_{O'}, O_{A'}, F_{O'})$ and also a morphism $S: (N, G, N') \to (O, H, O')$. The $\text{autDep}$-morphisms inform us that $M^i_a \triangleright R = R \triangleright N^i_a$ and $N^i_a \triangleright S = S \triangleright N^i_a$, so that:

$M^i_a \triangleright (R \triangleright S) = (M^i_a \triangleright R) \triangleright S = (R \triangleright N^i_a) \triangleright S = R \triangleright (N^i_a \triangleright S) = (R \triangleright S) \triangleright O^i_a$.

Finally,

\[
\begin{align*}
R \triangleright S[I_M] &= R; S^\ast[I_M] = S^\ast[R[I_M]] = S^\ast[F_{O'}], \\
S^\ast[G[I_N]] &= G; S^\ast[I_N] = S[I_N] = F_{O'},
\end{align*}
\]

(Def. of Dep and $\text{autDep}$)

\[
(R \triangleright S)[O'_M] = (R \triangleright S)^\ast[I'_O] = (S^\ast; R^\ast[I'_O], \\
= S^\ast; R^\ast[I'_O], \\
= S; R^\ast[I'_O], \\
= R^\ast[S[I'_O]], \\
= R^\ast[F_{O'}], \\
= R^\ast[G[I'_N]], \\
= R^\ast[G[I'_N] = F_M (\text{Def. of } \text{autDep} \text{ and } \text{Dep})].
\]

\[\Box\]

Theorem 3.1.14 (Self-duality of $\text{autDep}$). We have the self-duality functor $\text{Rev}: \text{autDep}^\circ \to \text{autDep}$:

$\text{Rev}(N, G, N') := (N', \bar{G}, N)$, $\text{Rev}R := R^\ast$.

recalling the self-duality of $\text{Dep}$ from Theorem 3.1.14.

Proof. Its action on objects is well-defined: (4) holds because we know $N_a; \bar{G} = \bar{G}; (N'_a)^\ast$ and hence $N'_a; \bar{G} = \bar{G}; (N'_a)^\ast$; (5) holds because it is invariant under swapping the lower/upper nfa. Its action on morphisms is well-defined by a similar argument, recalling that $(R: F \to G)^\ast = \bar{R}: \bar{G} \to \bar{F}$. Then it is a functor because $(-)^\ast$ is. It is an equivalence functor for the same reason.

Next we specify a way in which dependency automata can be isomorphic i.e. via distinct pairs of witnessing relations $(N_a, N'_a)$ of the same $\text{Dep}$-endomorphism $N_a; \bar{G} = N'_a; \bar{G}$: $(N'_a)^\ast$. In other words, there can be too few transitions relative to the inclusion-maximal components $(N'_a)^\ast$ and $(N'_a)^\ast$. There can also be too few initial/final states (these sets correspond to $\text{Dep}$-morphisms too).

Proposition 3.1.15 ($\text{autDep}$ transition-based isomorphisms). Given $(N, G, N')$ and $(M, G, M')$ such that:

$N_a; G = M_a; G$ (for $a \in \Sigma$, $G[I_N] = G[I_M]$ and $\bar{G}[I_{N'}] = G[I_{M'}]$)

then $id_G: (N, G, N') \to (M, G, M')$ is an $\text{autDep}$-isomorphism.

Proof. $G$ certainly defines a $\text{Dep}$-morphism $id_G: G \to G$. It is an $\text{autDep}$-isomorphism $G: (N, G, N') \to (M, G, M')$ because:

$M_a; G = N_a; G$ (by assumption)

$G; N'_a ((N, G, N')$ a dependency automaton)

and similarly $G[I_M] = G[I_M] = G[I_M]$ and $\bar{G}[I_{N'}] = G[I_{M'}] = F_M$. By a symmetric argument we infer $G: (M, G, M') \to (N, G, N')$ is well-defined, and also the inverse because $id_G; id_G = id_G$ in $\text{Dep}$.

\[\Box\]

Proposition 3.1.16 (Polytime canonical dependency automaton). Given dfas $\alpha, \beta$ s.t. $L(\beta) = (L(\alpha))^x$ one can build $L(\alpha)'s$ canonical dependency automaton in polytime.
Proof. Minimising α in polytime yields γ := (x₀, X, γₐ, F), minimising β yields δ := (y₀, Y, δₐ, Fₙ). Construct \( G \subseteq X \times Y \),
\[
G(γ_u(x₀), δ_v(y₀)) : \iff γ_v(γ_u(x₀)) ∈ Fγ \iff uν ∈ L
\]
noting that γ and δ are reachable. Then we have the bipartite graph isomorphism:
\[
\begin{array}{ccc}
Y & \xrightarrow{β_2} & LW(L') \\
G & \xrightarrow{β_1} & L_{DEP} \\
X & \xrightarrow{β_1} & LW(L)
\end{array}
\]
where \( β_1(γ_u(x₀)) := u⁻¹L \) and \( β_2(δ_v(y₀)) := ν⁻¹L' \). It induces a \( DEP \)-isomorphism – in fact an \( auto_{DEP} \)-isomorphism. □

3.2 Deterministic automata over join-semilattices

Just as each nfa has a reverse, each dependency automaton \((N, G, N')\) has a reverse \( Rev(N, G, N')\). It swaps the lower/upper nfa and takes the converse of \( G \) (equivalently, swaps the bipartitions). This construction arose from the self-duality of \( DEP \). We now focus on lifting the categorical equivalence \( DEP \cong JSL \) to one between dependency automata and deterministic finite automata interpreted in join-semilattices. In the process we’ll generalise the subset construction to dependency automata.

**Definition 3.2.1 (\( dfa_{JSL} \)).** A JSL-dfa is a 4-tuple \((s, S, γₐ, F)\) where \( S = (S, ∨_S, ↓_S) \) is a finite join-semilattice, \( s ∈ S \) is an element, \( δₐ : S → S \) is an join-semilattice morphism for \( a ∈ Σ \), and \( F := ↓_S' T ∈ S \) for some \( t ∈ S \). It accepts the language its underlying dfa does. A JSL-dfa morphism is a dfa morphism which is also a join-semilattice morphism i.e. preserves all joins. Given \( u ∈ Σ^* \) we inductively define endomorphisms \( δ_z := id_S \) and \( δ_w := δ_a o δ_v \). The category \( dfa_{JSL} \) consists of the JSL-dfas with their morphisms, where composition is functional. □

Importantly, JSL-dfas are deterministic finite automata interpreted in join-semilattices.⁵

| states         | classical dfa | JSL-dfa |
|----------------|--------------|---------|
| initial state  | \( α : \{*\} → Z \) | \( α : 2 → S \) |
| transitions    | \( δₐ : Z → Z \) | \( δₐ : S → S \) |
| final states   | \( ω : Z → \{0, 1\} \) | \( ω : S → 2 \) |

Indeed, viewing sets as algebras for the empty signature then \( \{*\} \) is free 1-generated, just as 2 is the free 1-generated join-semilattice. Morphisms from such algebras amount to picking a single element. On the other hand, \( \{0, 1\} \) and 2 are the unique (modulo isomorphism) two-element algebras of their respective varieties. Morphisms to such algebras amount to subsets i.e. the elements sent to 1. Permitting every function \( ω : Z → \{0, 1\} \) permits any set of final states. Morphisms \( ω : S → 2 \) must have a largest element sent to 0, so that \( ω⁻¹(\{\{\}\}) = ↓_S' t \) for some \( t ∈ S \).

The following Lemma provides further clarification. That is, a join of states accepts the union of the languages accepted by its summands. As a special case, the bottom element accepts the empty language.

**Lemma 3.2.2.** For any JSL-dfa \( δ = (s₀, S, δₐ, F) \) and \( X ⊆ S \),
\[
L(δ_0 ∨_S X) = \bigcup_{s∈X} L(δ_0 s).
\]

**Proof.** Let \( t = ∨_S F \) be the largest non-final state. Each \( δ_w : S → S \) is an endomorphism so \( δ_w(∨_S X) ≤_S t \iff ∨_S \{δ_w(x) : x ∈ X\} ≤_S t \iff ∃x ∈ X. δ_w(x) ≤_S t \). □

Next, the category of JSL-dfas is self-dual. That is, one can take adjoints and exchange the initial state with the largest non-final state.

⁵Other varieties where dfas can be interpreted include pointed sets, distributive lattices, boolean algebras and vector spaces over \( F_2 \).
Proof. Let \( \alpha : 2 \to S \) where \( \alpha(1) := s_0 \) and \( \omega : S \to 2 \) where \( \omega^{-1}(\{1\}) = F \). Then we calculate:

\[
\begin{align*}
\omega \circ \delta_w \circ \alpha &= id_2 \\
\alpha_+ \circ (\delta_w)_+ \circ \omega_+ &= (id_2)_+ = id_{2^S} \quad \text{(consider action on } \top_2) \\
\alpha_+ \circ (\delta^*)_w \circ \omega_+ &= id_{2^S} \quad \text{(by def. of } (\delta^*)_w) \\
\alpha_+ \circ (\delta^*)_w \circ \omega_+(0) &= 0 \quad \text{(since } \top_2 = 0) \\
\alpha_+ \circ (\delta^*)_w \circ \omega_+(1) &= 1 \quad \text{(adjoints)} \\
(\delta^*)_w \circ (\omega^*)_S \circ (\alpha_+)_S(1) &= \omega(0) \quad \text{(since } (\alpha_+)_S = \alpha) \\
\omega^* \in L(\delta^*) 
\end{align*}
\]

Importantly, dependency automata and dfas interpreted in semilattices are two sides of the same coin.

Definition 3.2.5 (Equivalence functors for automata).

1. \( \text{Det} : \text{aut}_{\text{Dep}} \to \text{dfa}_{\text{JSL}} \) determinises a dependency automaton:

\[
\text{Det}(N, G, N') := (F_{N'}, \text{Open}G, \lambda Y.(N'_a)^* [Y], \{ Y \in O(G) : Y \cap I_{N'} \neq \emptyset \})
\]

and acts on morphisms as \( \text{Open} \) does (see Definition 2.2.10.1).

2. \( \text{Airr} : \text{dfa}_{\text{JSL}} \to \text{aut}_{\text{Dep}} \) constructs a dependency automaton over the semilattice’s irreducibles:

\[
\text{Airr}(s, S, \delta_a, F) := (\mathcal{N}, \text{Pirr}S, \mathcal{N}')
\]

\[
\mathcal{N} := (J(S) \cap \downarrow S, J(S), (\text{Pirr}\delta_a)_-, J(S) \cap F)
\]

\[
\mathcal{N}' := (M(S) \cap \downarrow S \uparrow S F, M(S), (\text{Pirr}\delta_a)_+, M(S) \cap \downarrow S S).
\]

It acts on morphisms as \( \text{Pirr} \) does (see Definition 2.2.10.2).

Theorem 3.2.6 (Automata-theoretic categorical equivalence). \( \text{Det} : \text{aut}_{\text{Dep}} \to \text{dfa}_{\text{JSL}} \) and \( \text{Airr} : \text{dfa}_{\text{JSL}} \to \text{aut}_{\text{Dep}} \) define an equivalence of categories with natural isomorphisms inherited from Theorem 2.2.14:

\[
\begin{align*}
\text{rep} : Id_{\text{dfa}_{\text{JSL}}} &\Rightarrow \text{Det} \circ \text{Airr} \\
\text{rep}_{(s, S, \delta_a, F)} &:= \text{rep}_S \\
\text{red} : Id_{\text{aut}_{\text{Dep}}} &\Rightarrow \text{Airr} \circ \text{Det} \\
\text{red}_{(N, G, N')} &:= \text{red}_G
\end{align*}
\]

Proof.

1. \( \text{Det} \) is well-defined.

Since \( \text{Open} \) is a well-defined functor we need only show \( \text{Det} \) is well-defined on objects and morphisms. Concerning objects, first recall:

\[
\text{Det}(N, G, N') := (F_{N'}, \text{Open}G, \lambda Y.(N'_a)^* [Y], \{ Y \in O(G) : Y \cap I_{N'} \neq \emptyset \})
\]

Then \( F_{N'} = G[I_N] \in O(G) \) is an element of the join-semilattice \( \text{Open}G \), as required. Since \( N_a : G = N'_a = G; (N'_a)^* \) we have \( N'_a : G \to G \); applying \( \text{Open} \) yields an endomorphism of \( \text{Open}G \) with action \( \lambda Y \in O(G).((N'_a)^*)_Y \). The
latter can be rewritten $\lambda Y.(N^1_a)^*[Y]$ because $G; (N^1_a)^*_s = G; (N^1_a)^*$ and each $Y = G[X]$ for some $X$. Finally, the non-final states $\{Y \in O(G) : Y \cap I_{N'} = \emptyset\}$ have a largest element $\text{in}_G(I_{N'})$. Then the JSL-dfa is well-defined.

To see $\text{Det}$ is well-defined on morphisms we'll show the respective JSL-morphisms preserve the additional structure. Given $R : (M, G, M') \rightarrow (N, H, N')$ we have $\text{Open}R : \text{Open}G \rightarrow \text{Open}H$. The initial state is preserved:

$$\text{Open}G(F_{M'}) = \text{Open}G(G[I_M]) = G; R^*_a[I_M] = R[I_M] = F_{N'}.$$ 

The transitions are preserved because for any $Y = G[X]$,

$$\text{Open}R((M'_a)^*[Y]) = \mathcal{R}^*_a((M'_a)^*[Y]) \quad (\text{def. of } \text{Open})$$

$$= G; (M'_a)^*; \mathcal{R}^*_a[X] \quad (\text{def. of } \text{aut}_\text{Dep})$$

$$= M_a; G; \mathcal{R}^*_a[X] \quad (\text{def. of } \text{aut}_\text{Dep})$$

$$= \mathcal{R}; (N^1_a)^*[X] \quad (\text{def. of } \text{aut}_\text{Dep})$$

$$= G; \mathcal{R}^*_a; (N^1_a)^*[X] \quad (\text{def. of } \text{aut}_\text{Dep})$$

$$= \mathcal{R}^*_a; (N^1_a)^*[Y] \quad (\text{def. of } \text{Open})$$

To see the final states are preserved, observe $I_{M'}$ determines the $\text{Dep}$-morphism:

$$\begin{array}{ccc}
\text{G} & I_{M'} \times (0) & \text{Pirr}2 \\
\text{G}_s & 0 \\
\text{G}_{F_{M'}} & \text{Pirr}2 \times (1) & 1
\end{array}$$

Fix $Y = G[X] \in O(G)$. Then $Y$ is final iff $Y \cap I_{M'} \neq \emptyset$ iff $X \cap F_{M} \neq \emptyset$. Moreover $\text{Open}R(Y) = \mathcal{R}[X]$ is final iff $\mathcal{R}[X] \cap I_{N'} \neq \emptyset$. By assumption $\mathcal{R}[I_{N'}] = F_{M}$, so $Y$ is final iff $X \cap \mathcal{R}[I_{N'}] \neq \emptyset$ iff $\text{Open}R(Y)$ is final.

2. $\text{Airr}$ is well-defined.

Since $\text{Pirr}$ is a well-defined functor it suffices to show $\text{Airr}$ is well-defined on objects and morphisms. Concerning objects, $\text{Airr}(s, S, \delta_s, F) := (N, \text{Pirr}S, N')$ and both $N$ and $N'$ are well-defined nfas. Condition (4) holds:

$$(\text{Pirr} \delta_a) ; \text{Pirr}S = \text{Pirr} \delta_a ; \text{id}_{\text{Pirr}S} = \text{Pirr} \delta_a = \text{id}_{\text{Pirr}S} ; \text{Pirr} \delta_a = \text{Pirr}S ; (\text{Pirr} \delta_a).$$

Finally, condition (5) holds:

$$\text{Pirr}S[I_{N'}] = \text{Pirr}S[(j \in J(S) : j \leq_S s)]$$

$$= \{m \in M(S) : \exists j \in J(S) \lfloor j \leq_S s \text{ and } j \not\leq_S m \}$$

$$= \{m \in M(S) : s \not\leq_S m \} = F_{N'}.$$ 

$$(\text{Pirr}S)^*[I_{N'}] = (\text{Pirr}S)^*[(m \in M(S) : \forall_S F \leq_S m)]$$

$$= \{j \in J(S) : \exists m \in M(S) \lfloor j \not\leq_S m \text{ and } \forall_S F \leq_S m \}$$

$$= \{j \in J(S) : j \not\leq_S \forall_S F \} = F_{N'}.$$ 

To see $\text{Airr}$ is well-defined on morphisms, take $f : (s_0, S, \gamma_a, F_S) \rightarrow (t_0, T, \delta_a, F_T)$ so we have $\text{Pirr}f : \text{Pirr}S \rightarrow \text{Pirr}T$. Then let us verify the required identities:

$$(\text{Pirr} \gamma_a) ; \text{Pirr}f = \text{Pirr} \gamma_a ; \text{Pirr}f$$

$$= \text{Pirr} (\gamma_a \circ f)$$

$$= \text{Pirr} (f \circ \delta_a)$$

$$= \text{Pirr}f ; \text{Pirr} \delta_a$$

$$= \text{Pirr}f ; (\text{Pirr} \delta_a)^*.$$
Moreover if $\text{Airr}: (M, \text{Pirr}S, M') \to (N, \text{Pirr}T, N')$ then,  
\[
\text{Pirr}[I_M] = \text{Pirr}\{\{j \in J(S) : j \in S s_0\}\} \\
= \{m \in M(T) : \exists j \in J(S). (f(j) \notin T m \text{ and } j \in S s_0)\} \\
= \{m \in M(T) : \exists j \in J(S). (j m \notin f_\circ (m) \text{ and } j \in S s_0)\} \\
= \{m \in M(T) : s_0 \notin j m \bigcirc_\circ (m)\}. \\
= \{m \in M(T) : f(s_0) \notin T m\}. \\
= \{m \in M(T) : t_0 \notin T m\}. \\
= F'_{N'}.
\]
\[
(\text{Pirr})^-[I_{N'}] = \{j \in J(S) : \exists m \in M(T). (f(j) \notin T m \text{ and } V_S \overline{T} F \leq T m)\} \\
= \{j \in J(S) : f(j) \notin V_S \overline{T} F S\} \\
= \{j \in J(S) : j \notin V_S \overline{T} F S\} = F_N.
\]

3. $\text{rep}$ restricts to a natural isomorphism as claimed.
Recall the natural isomorphism $\text{rep} : Id_{\mathbb{J}S} \Rightarrow \text{Open} \circ \text{Pirr}$ where $\text{rep}_s : = \lambda s \in S. \{m \in M(S) : s \notin S m\}$. Then given any $\mathbb{J}S$-dfa $\delta = (s_0, S, \delta_a, F)$ it suffices to establish the typing $\text{rep}_s : \delta \to \text{DetAirr}\delta$ where:
\[
\text{DetAirr}\delta = \langle \{m \in M(S) : s_0 \notin S m\}, \text{OpenPirr}S, \lambda Y. (\text{Pirr}\delta_a)_\circ \delta\{Y\}, \{Y \in O(\text{Pirr}S) : \sqrt{S} \notin S Y\} \rangle.
\]
The initial state is clearly preserved. Next, the deterministic transitions are preserved:
\[
\text{rep}_s \circ \delta_a (s) = \{m \in M(S) : \delta_a (s) \notin S m\} = \{m \in M(S) : s \notin S \delta_a \circ (m)\}
\]
\[
(\text{Pirr}\delta_a)_\circ \delta s = (\text{Pirr}\delta_a)_\circ \{m \in M(S) : s \notin S m\} \\
= \{m \in M(S) : \exists m' \in M(S). (s \notin S m' \text{ and } \delta_a \circ (m'))\} \\
= \{m \in M(S) : \exists s \notin S \delta_a \circ (m)\}.
\]
Concerning the final states we know $\text{rep}_s[F] = \{M(S) \cap \notin S [s] : s \in F\}$, so given $s \in F$ let $Y_s := \{m \in M(S) : s \notin S m\} \in O(\text{Pirr}S)$. Then:
\[
V_S \overline{T} \notin S Y_s \iff \forall m \in M(S), (s \notin S m \Rightarrow V_S \overline{T} \notin S m) \\
\iff \forall m \in M(S), (V_S \overline{T} \notin S m \Rightarrow s \notin S m) \\
\iff s \notin V_S \overline{T} \notin S \notin F
\]

4. $\text{red}$ restricts to a natural isomorphism as claimed.
Recall the natural isomorphism $\text{red} : Id_{\mathbb{M} \text{Dep}} \Rightarrow \text{Pirr} \circ \text{Open}$. Take any dependency automaton $\mathbb{M} := (M, G, M')$. It suffices to establish the typing $\text{red}_G : \mathbb{M} \Rightarrow \text{AirrDetAirr} \mathbb{M}$ whose codomain is $(N, \text{Pirr}OpenG, N')$ where:
\[
N := \langle \{j \in J(\text{Open}G) : j \notin F_{M'}\}, J(\text{Open}G), N_\circ \delta, \{Y \in J(\text{Open}G) : Y \cap I_{M'} = \emptyset\} \rangle
\]
\[
N' := \langle \{m \in M(\text{Open}G) : \text{in}_G(I_{M'}) \in m\}, M(S), N'_\circ \delta, \{m \in M(S) : F_{M'} \notin m\} \rangle.
\]
Firstly by $\text{Dep}$-composition and naturality,
\[
M_\circ \text{red}_G = M'_\circ \delta ; \text{red}_G = \text{red}_G ; \text{Pirr} \text{Open} M'_\circ = \text{red}_G ; (\text{Pirr} \text{Open} M'_\circ)^- = \text{red}_G ; (N'_\circ)^-.
\]
Finally we establish the two remaining conditions:
\[
\text{red}_G[I_M] = \{Y \in M(\text{Open}G) : \exists g_s \in I_{M'}G, G[g_s] \notin Y\} \\
= \{Y \in M(\text{Open}G) : G[I_M] \notin Y\} \\
= \{Y \in M(\text{Open}G) : F_{M'} \notin Y\} \\
= F_{N'} \\
\text{(def. of } Aut_{\text{Dep}}) \\
\text{(see above)}
\]
\[
(\text{red}_G)^-[I_{N'}] = \{g_s \in G_s : \exists Y \in M(\text{Open}G), G[g_s] \notin Y \text{ and } \text{in}_G(I_{M'}) \subseteq Y\} \\
= \{g_s \in G_s : G[g_s] \notin \text{in}_G(I_{M'})\} \\
= \{g_s \in G_s : (g_s) \notin G_\circ \text{in}_G(I_{M'})\} \\
= \{g_s \in G_s : g_s \notin G(I_{M'})\} \\
= \bar{G}[I_M] \\
= F_M.
\]
Recall that each nfa $\mathcal{N}$ induces a dependency automaton $\det(\mathcal{N}) = \det(\mathcal{N}; \Delta Z, \rev(\mathcal{N}))$.

**Note 3.2.7** ($\det(\mathcal{N}; \Delta_Z, \rev(\mathcal{N}))$) is $\mathcal{N}$’s full subset construction). Given any nfa $\mathcal{N} = (z_0, Z, \mathcal{N}_a, F)$,

$$\det(\det(\mathcal{N})) = \det(\mathcal{N}; \Delta Z, \rev(\mathcal{N})) = (I_{\rev(\mathcal{N})}, \open_{\Delta Z}, \lambda X. (\mathcal{N}_a)^\ast [X], \{X \subseteq Z : X \cap I_{\rev(\mathcal{N})} \neq \emptyset\})$$

i.e. the full subset construction for $\mathcal{N}$ endowed with inclusion ordering. This explains Definition 3.2.8 below.

**Definition 3.2.8** (Full subset construction $\sc(\mathcal{N})$). For any nfa $\mathcal{N}$ define:

$$\sc(\mathcal{N}) := \det(\det(\mathcal{N})) = (I_N, (PZ, \cup, \emptyset), \lambda X. \mathcal{N}_a[X], \{X \subseteq Z : X \cap F_N \neq \emptyset\}).$$

This is $\mathcal{N}$’s full subset construction endowed with its JSL-dfa structure.

**Note 3.2.9** ($\det(\mathcal{N}, G, \mathcal{N}')$ restricts $\rev(\mathcal{N})$’s full subset construction). Generally speaking, $\det(\mathcal{N}, G, \mathcal{N}')$ is obtained from $\rev(\mathcal{N})$’s full subset construction by restricting to $\open G \subseteq \open_{\Delta G}$. This generalises Note 3.2.7.

**Note 3.2.10** ($\det$ and $\airr$ preserve the accepted language). Given any dependency automaton $(\mathcal{N}, G, \mathcal{N}')$, the classically reachable part of $\det(\mathcal{N}, G, \mathcal{N}')$ has a classical description too:

$$\text{reach}(\det(\mathcal{N}, G, \mathcal{N}')) = \rsc(\rev(\mathcal{N}')).$$

It follows by Lemma 3.1.11 that $\det$ preserves the accepted language. The natural isomorphism $\text{red} : I_{\text{aut}} \Rightarrow \airr \circ \det$ informs us that $\airr$ also preserves the accepted language.

**Corollary 3.2.11** (Language correspondence). Let $\delta = (s_0, S, \delta_a, F)$ be a JSL-dfa and $\mathcal{N}$ the lower nfa of $\airr \delta$. Then:

$$L(\mathcal{N}_{\delta} Z) = \text{acc}_\delta(\bigvee Z) \quad \text{for every } Z \subseteq J(S).$$

In particular each individual state $j \in J(S)$ of $\mathcal{N}$ accepts $\text{acc}_\delta(j)$.

**Proof.** By Note 3.2.10 $\det$ preserves the accepted language. Given $\airr \delta = (\mathcal{N}, G, \mathcal{N}')$ and $Z \subseteq J(S)$ then:

$$(\mathcal{N}_{\delta} Z, G, \mathcal{M}) \quad \text{where } \mathcal{M} := \rev(\mathcal{N}')_{\delta G}[Z]$$

is a well-defined dependency automaton. Applying $\det$ yields the JSL-dfa $(\det \airr \delta)_{\delta G}[Z]$ which accepts $L(\mathcal{N}_{\delta} Z)$. Finally $\text{rep}_\delta^{-1}$ provides a language-preserving isomorphism $(\det \airr \delta)_{\delta G}[Z] \rightarrow \delta_{\delta \cup \vee Z}$.

**Example 3.2.12** (Dualising the full subset construction). Via relative complement we have the JSL-dfa isomorphism:

$$(\sc(\mathcal{N}))^* \equiv \sc(\rev(\mathcal{N}))$$

which follows by considering $(\sc(\mathcal{N}))^* = (T, \open_{\Delta Z}, \mathcal{R}_a, \{X \subseteq Z : X \cap T \neq \emptyset\})$. In other words, the dual of the full subset construction for $\mathcal{N}$ is the full subset construction for $\rev(\mathcal{N})$. This isomorphism instantiates the natural isomorphism $\hat{\partial}$ described below.

The self-duality transfers of Theorem 2.2.18 generalise naturally to the automata-theoretic setting.

**Theorem 3.2.13** (Automata-theoretic self-duality transfer).

1. $\hat{\partial} : (-)^* \circ \det^{op} \Rightarrow \det \circ \rev$ restricts $\partial$ from Theorem 2.2.18.1.

2. $\hat{\lambda} : \rev \circ \airr^{op} \Rightarrow \airr \circ (-)^*$ restricts $\lambda$ from Theorem 2.2.18.2.

**Proof.**
1. Given a dependency automaton \((N, G, N')\) it suffices to show \(\partial_{\delta} : (\text{Open} G)^{\text{op}} \rightarrow \text{Open} G\) defines an \(\text{dfa}_{JSL}\)-morphism of type \((\text{Det}(N, G, N'))^* \rightarrow \text{DetRev}(N, G, N')\).

Concerning preservation of the initial state,

\[
\partial_{\delta}(\{ Y \in O(G) : Y \cap I_{N'} \neq \varnothing \}) = \partial_{\delta}(\{ Y \in O(G) : Y \cap I_{N'} = \varnothing \})
\]

\[
= \partial_{\delta}(\text{in}_{G}(I_{N'})) (\text{def. of } \partial_{\delta})
\]

\[
= G^{\dagger}(\text{in}_{G}(I_{N'})) (\text{De Morgan duality})
\]

\[
= G^{\dagger}(I_{N'}) (\text{adjoints})
\]

\[
= G[I_{N'}] (\text{def. of } (\text{N}, G, N')).
\]

The marked equality holds because \(\text{in}_{G}(I_{N'})\) is the largest \(G\)-open in \(I_{N'}\). Next, the final states are preserved and reflected:

\[
X \in \partial_{\delta}^{-1}(\{ Y \in O(G) : Y \cap I_{N} \neq \varnothing \}) \iff G[X] \cap I_{N} \neq \varnothing
\]

\[
\iff G^{\dagger}(X) \cap I_{N} \neq \varnothing (\text{De Morgan duality})
\]

\[
\iff I_{N} \notin G^{\dagger}(X) (\text{def. of } (\text{N}, G, N'))
\]

\[
\iff G[I_{N}] \notin X (\text{adjoints})
\]

\[
\iff F_{N} \notin X (\text{def. of } (\text{N}, G, N')).
\]

\[
\iff X \in \text{Open}_{G} F_{N'}.
\]

Finally, preservation of the deterministic transitions follows by the naturality of \(\partial\).

2. Given a \(JSL\)-dfa \(\delta = (s_0, S, \delta_a, F)\) it suffices to show \(\lambda_{S} : (\text{Pirr} S)^{\text{op}} \rightarrow \text{Pirr}(S^{\text{op}})\) defines an \(\text{aut}_{\text{Dep}}\)-morphism of type \(\text{RevAirr} \delta \rightarrow \text{Airr}(\delta^{*})\). It types correctly, and since \(\lambda\)’s components are identity morphisms we are done.

\(\square\)

\(JSL_f\) and \(\text{Dep}\) have enough projectives/injectives by Proposition 2.2.19, as do the automata-theoretic categories.

**Proposition 3.2.14** (\(\text{dfa}_{JSL}\) and \(\text{aut}_{\text{Dep}}\) have enough projectives).

Let \(Z\) be a finite set and \(\gamma := (s_0, S, \gamma_a, F_{\gamma})\) be a \(JSL\)-dfa.

1. Given any nfa \((I, Z, R_a, F)\) we have the \(JSL\)-dfa \((I, \text{Open}_{\Delta Z}, R_a^I, F_{\gamma})\).

2. \(\varepsilon_{S} : (J(S) \cap I_{S}) \rightarrow \gamma\) where \(\varepsilon_{S}(X) := \forall_{S} X\) is a surjective \(JSL\)-dfa morphism.

3. Given \((I, \text{Open}_{\Delta Z}, R_a^I, F) \xrightarrow{f} (t_0, T, \delta_a, F_{\gamma}) \xrightarrow{g} \gamma\) then \(f = g \circ g\) where \(g\) uniquely extends \(\lambda z q_{*}(f(\{z\}))\).

The self-duality of \(\text{dfa}_{JSL}\) preserves the freeness of the join-semilattice, so we immediately deduce:

**Corollary 3.2.15.** \(\text{dfa}_{JSL}\) and \(\text{aut}_{\text{Dep}}\) have enough injectives.

Recall Proposition 2.2.21. Choosing any join/meet-generators for \(S\) we can construct \(\xi_{S} |_{J \times M} \cong \text{Pirr} S\). Likewise, given any \(JSL\)-dfa \(\delta\) over \(S\), choosing such generators yields a dependency automaton \(\text{aut}_{\text{Dep}}\)-isomorphic to \(\text{Airr} \delta\).

**Proposition 3.2.16** (\(\text{aut}_{\text{Dep}}\) generator-based isomorphisms).

In the notation of Proposition 2.2.21,

1. \(I_{S} : \mathcal{R}_{S} \rightarrow \text{Airr}(s_0, S, \gamma_a, F_{S})\) defines an \(\text{aut}_{\text{Dep}}\)-isomorphism where \(\mathcal{R}_{S} := (N, \xi_{S} |_{J \times M}, N')\),

\[
\mathcal{N} := (J_{S} \cap I_{S} s_{0}, J_{S}, \mathcal{N}_{a}, J_{S} \cap F_{S})
\]

\[
\mathcal{N}_{a}(x_{1}, x_{2}) : \iff x_{2} \leq_{S} \delta_{a}(x_{1})
\]

\[
\mathcal{N}' := (M_{S} \cap I_{S} \forall_{S} \mathcal{F}_{S}, M_{S}, M_{S} \cap F_{S}, M_{S} \cap I_{S} s_{0})
\]

\[
\mathcal{N}_{a}(y_{1}, y_{2}) : \iff (\delta_{a})*_{S}(y_{1}) \leq_{S} y_{2}.
\]

2. Suppose \(f : (s_0, S, \gamma_a, F_{S}) \rightarrow (t_0, T, \delta_a, F_{T})\) is a \(JSL\)-dfa morphism. Then \(I_{f} : \mathcal{R}_{S} \rightarrow \mathcal{R}_{T}\) defines an \(\text{aut}_{\text{Dep}}\)-isomorphism.
The full powerset construction $\mathcal{S}(N) := \text{Det}(\text{dep}(N))$ also defines a free construction.

**Theorem 3.2.17** (Free JSL-dfa on a dfa). If $\gamma$ is a dfa, $\delta = (t_0, \top, \delta_\top, F_\top)$ is a JSL-dfa and $f : \gamma \to \delta$ is a dfa morphism,

$$\text{JSL} \prod f(S) : \text{sc}(\gamma) \to \delta$$

is a well-defined JSL-dfa morphism.

**Proof.** Let $\gamma = (z_0, Z, \gamma_a, F_\gamma)$ and denote the candidate JSL-dfa morphism by $\hat{f}$. Concerning the initial state, $\hat{f}(\{z_0\}) = f(z_0) = t_0$ because $f$ is a dfa morphism by assumption. Concerning final states $\forall \gamma f[S] \in F_\delta \iff \exists \gamma \in S.f(s) \in F_\delta$ (since there is a largest non-final state) iff $S \cap F_\delta \neq \emptyset$ (since $f$ is a dfa morphism). Finally for each $S \subseteq Z$,

$$\hat{f} \circ \alpha_{\top}(S) = \forall \gamma f[\gamma_a[S]]$$

$$= \forall \gamma \{f \circ \alpha_a(s) : s \in S\}$$

$$= \forall \gamma \{\delta_a \circ f(s) : s \in S\} \quad (f \text{ a dfa morphism})$$

$$= \delta_a(\forall \gamma \{f(s) : s \in S\}) \quad (\delta_a \text{ preserves } \top\text{-joins})$$

$$= \delta_a \circ \hat{f}(S).$$

There is also a free construction for ordered dfsas, recalling Definition 3.1.3.6.

**Theorem 3.2.18** (Free JSL-dfa on an ordered dfa). Let $\gamma = (p_0, P, \gamma_a, F_\gamma)$ be an ordered dfa, $\delta = (t_0, \top, \delta_\top, F_\top)$ a JSL-dfa and $f : \gamma \to \delta$ an ordered dfa morphism. Then we have the well-defined JSL-dfa morphism,

$$\text{JSL} \prod f[S] : \text{Det}(\gamma_\top, \geq_P, \text{rev}(\gamma_\top)) \to \delta$$

where $\gamma_\top := (\downarrow_P, p_0, P, \gamma_a, \geq_P, F_\gamma)$ is an nfa.

**Proof.** We first verify $(\gamma_\top, \geq_P, \text{rev}(\gamma_\top))$ is a dependency automaton. Concerning transitions, $(\gamma_a; \geq_P) ; \geq_P = \gamma_a; \geq_P$ by transitivity and $\geq_P; (\gamma_a; \geq_P) = \gamma_a; \geq_P$ by Example 2.1.8.4 (via $\gamma_a : P^{op} \to P^{op}$). Concerning the remaining conditions:

$$\geq_P[I_{\text{dfsa}(L')} \gamma_\top] = \geq_P[I_{\text{dfsa}(L')} \gamma_\top] = \downarrow_P = \downarrow_P p_0 = F_{\text{rev}(\text{dfsa}(L'))},$$

$$\geq_P[I_{\text{rev}(\text{dfsa}(L'))}] = \leq_P \{Y \in \text{LW}(L') : \varepsilon \in Y\} = \{Y \in \text{LW}(L') : \varepsilon \in Y\} = F_{\text{dfsa}(L')},$$

Denote the candidate JSL-dfa morphism by $\hat{f}$. Recall $\text{Det}(\gamma_\top, \geq_P, \text{rev}(\gamma_\top))$ is $\text{sc}(\gamma_\top)$ restricted to $\text{open } \geq_P$. Concerning the initial state, $\hat{f}(\downarrow_P p_0) = f(p_0) = t_0$ by monotonocity and the fact that $f$ is a dfa morphism. Concerning final states, $\forall \gamma f[S] \in F_\delta \iff \exists \gamma \in S.f(s) \in F_\delta$ iff $S \cap F_\delta \neq \emptyset$ (since $f$ is a dfa morphism). Finally for each down-closed $S \subseteq P$,

$$\hat{f} \circ (\gamma_a; \geq_P)^\top(S) = \forall \gamma f[\gamma_a; \geq_P[S]]$$

$$= \forall \gamma \{f \circ \gamma_a(s) : s \in S\} \quad(f \circ \gamma_a \text{ monotonlic})$$

$$= \forall \gamma \{\delta_a \circ f(s) : s \in S\} \quad(f \text{ a dfa morphism})$$

$$= \delta_a(\forall \gamma \{f(s) : s \in S\}) \quad(\delta_a \text{ preserves } \top\text{-joins})$$

$$= \delta_a \circ \hat{f}(S).$$

3.3 Canonical dependency automata

Previously we described the canonical dependency automaton for $L \subseteq \Sigma^*$ in Definition 3.1.8. We now describe the state-minimal JSL-dfa for $L$. These two machines are actually the same object modulo categorical equivalence.

**Definition 3.3.1** (Left and right quotients). Fix any regular language $L \subseteq \Sigma^*$ and recall the left word quotients $\text{LW}(L)$ from Definition 3.1.3.3.

1. For $U, V \subseteq \Sigma^*$, $U^{-1}L := \{w \in \Sigma^* : \exists u \in U. uw \in L\}$ is a left quotient, $LV^{-1} := \{w \in \Sigma^* : \exists v \in V. vw \in L\}$ is a right quotient.

2. Let $LQ(L) := \{U^{-1}L : U \subseteq \Sigma^*\}$ and $RQ(L) := \{LV^{-1} : V \subseteq \Sigma^*\}$. Then $\text{LW}(L) \subseteq LQ(L)$ and likewise we may write $L v^{-1}$ instead of $L(\{v\})^{-1}$.
3. We have finite join-semilattice $\mathbb{L}Q(L) := (LQ(L), \cup, \emptyset)$.

4. $J(\mathbb{L}Q(L)) \subseteq LW(L)$ because the latter generate $LQ(L)$ under unions. Thus $J(\mathbb{L}Q(L))$ consists of those left word quotients which are not unions of others, so in particular are non-empty.

**Definition 3.3.2 (State-minimal JSL-dfa).** Let $\mathcal{d}fa(L) := (L, \mathbb{L}Q(L), \lambda X, a^{-1} X, \{ X \in LQ(L) : \varepsilon \in X \})$.

Observe that the state-minimal $\mathcal{d}fa(L)$ is obtained from $\mathcal{d}fa(L)$ by restricting to left word quotients $u^{-1} L$. Conversely, every left quotient $U^{-1} L$ is a finite union of left word quotients and $a^{-1}(-)$ preserves unions.

**Lemma 3.3.3.** $\mathcal{d}fa(L)$ is the state-minimal JSL-dfa accepting $L$.

**Proof.** It accepts $L$ – the reachable part of its underlying dfa is precisely the state-minimal dfa $\mathcal{d}fa(L)$. Concerning the state-minimality of $\mathcal{d}fa(L)$, take any JSL-dfa $\delta = (s_0, S, \delta, F)$ accepting $L$ and consider the languages accepted by varying the initial state, noting $|\text{langs}(\delta)| \leq |S|$. By Lemma we know $\mathcal{JSL}(L) \subseteq \text{langs}(\delta)$, hence $|\mathcal{JSL}(L)| \leq |S|$.

**Example 3.3.4 ($\mathcal{d}fa_1(L)$).** Applying Proposition 3.2.16 to $\mathcal{d}fa(L)$ with join-generating subset $J_{\mathbb{L}Q(L)} := LW(L)$ yields a dependency automaton, whose lower nfa takes the following form:

$$\mathcal{d}fa_1(L) := \{ (X \in LW(L) : X \subseteq L), LW(L), N_a, \{ X \in LW(L) : \varepsilon \in X \} \}
\quad N_a(X_1, X_2) : \iff X_2 \subseteq a^{-1} X_1.$$

It accepts $L$ by the witnessing aut$_{\text{dep}}$-isomorphism (see Note 3.2.10). Importantly, we’ll use it to represent the canonical distributive JSL-dfa further below.

Recall the self-duality $(-)^*: \mathcal{d}fa_{\text{op}} \rightarrow \mathcal{d}fa_{\text{JSL}}$ of Theorem 3.2.3, itself arising from the self-duality of JSL$_f$ in Note 2.2.8.3. We now describe an important representation of $\mathcal{d}fa(L)$ which explains its meet structure.

**Lemma 3.3.5.** $[L^{-1}X^{-1}]L = \{ w \in \Sigma^* : Lw^{-1} \subseteq LX^{-1} \}$ for any subsets $X, L \subseteq \Sigma^*$.

**Proof.**

$$[L^{-1}X^{-1}]L = \{ w \in \Sigma^* : \forall v \in \Sigma^*, (v \in L^{-1} \Rightarrow vw \notin L) \}
= \{ w \in \Sigma^* : \forall v \in \Sigma^*, (vw \in L \Rightarrow v \notin LX^{-1}) \}
= \{ w \in \Sigma^* : \forall v \in \Sigma^*, (v \in Lw^{-1} \Rightarrow v \notin LX^{-1}) \}
= \{ w \in \Sigma^* : Lw^{-1} \subseteq LX^{-1} \}.$$

**Theorem 3.3.6 (Fundamental dualising isomorphism $dr_L$).** For each regular $L \subseteq \Sigma^*$ we have the JSL-dfa isomorphism:

$$(\mathcal{d}fa(L'))* \xrightarrow{dr_L} \mathcal{d}fa(L) \quad dr_L(X) := [X']^{-1} L \quad dr_L^{-1} := dr_{L'}^*,$$

noting that reversal/complement of languages commute. There is also an alternative description:

$$dr_L(U^{-1} L') := \bigcup \{ X \in LW(L) : X \cap U' = \emptyset \}.$$

**Proof.**

1. We first establish the underlying join-semilattice isomorphism $dr_L : (LQ(L'))^\text{op} \rightarrow \mathbb{L}Q(L)$. Now, $dr_L$ is certainly a well-defined function. It is monotone because $X \subseteq Y \subseteq LW(L')$ implies $X \subseteq Y$ and hence $[Y']^{-1} L \subseteq [X']^{-1} L$. Likewise $dr_L^{-1}$ is a well-defined monotone function. Next, given any $X \in LW(L')$,

$$dr_L^{-1} \circ dr_L(X) = dr_L^{-1}([X']^{-1} L) = [[L^{-1}X^{-1}]L]^{-1} L'
= [L'X']^{-1} L'
= \{ w \in \Sigma^* : L'w^{-1} \notin L'X' \}
= \{ w \in \Sigma^* : \exists u \in \Sigma^*, [uw \in L' \text{ and } \forall v \in \Sigma^*, (uv \in L' \Rightarrow v \notin X)] \}
= \{ w \in \Sigma^* : \exists u \in \Sigma^*, [w \in u^{-1} L' \text{ and } \forall v \in \Sigma^*, (u^{-1}L'v \Rightarrow v \notin X)] \}
= \{ w \in \Sigma^* : \exists u \in \Sigma^*, [w \in u^{-1} L' \text{ and } u^{-1} L' \subseteq X] \}
= X$$

(since $X \in LQ(L)$).
It immediately follows that $dr_L \circ dr_L^\dagger(Y) = Y$ by substituting $L \mapsto L'$. Thus $dr_L$ is a bijective order-preserving and order-reflecting function, hence a bounded-lattice isomorphism and in particular a $\mathbb{JSL}_f$-morphism. Finally we establish the alternative action:

\[
\begin{align*}
    dr_L(U^{-1}L') & = [(U^{-1}L')^{-1}L] \\
    & = [L(U^{-1})^{-1}]^{-1}L \\
    & = \{ w \in \Sigma^* : Lw^{-1} \notin L(U')^{-1} \} \\
    & = \{ w \in \Sigma^*: \exists x \in \Sigma^*, \{x \in L \land \forall y \in \Sigma^*:\{xy \in L \Rightarrow y \notin U'\}\} \\
    & = \{ w \in \Sigma^*: \exists x \in \Sigma^*, \{w \in x^{-1}L \land \forall y \in \Sigma^*:\{y \in x^{-1}L \Rightarrow y \notin U'\}\} \\
    & = \{ w \in \Sigma^*: \exists x \in \Sigma^*, \{w \in x^{-1}L \land x^{-1}1L \notin U'\} \} \\
    & = \{ w \in \Sigma^*: \exists x \in \Sigma^*, \{w \notin x^{-1}L \land x^{-1}L \cap U' = \emptyset\} \} \\
    & = \bigcup \{ X \in LW(L) : X \cap U' = \emptyset \}.
\end{align*}
\]

2. It remains to establish that the join-semilattice isomorphism $dr_L$ defines a dfa morphism. Concerning preservation of the initial state:

\[
\begin{align*}
    dr_L(i_{\delta_{f\alpha(L')}}) & = dr_L(\bigvee_{LQ(L')} \{ X \in LW(L') : \varepsilon \notin X \}) \\
    & = dr_L(\bigcup \{ X \in LW(L') : \varepsilon \notin X \}) \\
    & = dr_L(L^{-1}L') \\
    & = \bigcup \{ X \in LW(L) : X \cap {}^{-} = \emptyset \} \\
    & = \bigcup \{ X \in LW(L) : X \subseteq L \} \\
    & = L = i_{\delta_{f\alpha(L)}}.
\end{align*}
\]

Concerning transitions, let $\gamma_a : \mathbb{L}Q(L') \rightarrow \mathbb{L}Q(L')$ and $\delta_a : \mathbb{L}Q(L) \rightarrow \mathbb{L}Q(L)$ be the deterministic $a$-transitions for the respective machines (both have action $\lambda X.a^{-1}X$). It suffices to show $(\gamma_a)_* = dr_L^{-1} \circ \delta_a \circ dr_L$:

\[
\begin{align*}
    dr_L^{-1} \circ \delta_a \circ dr_L(X) & = dr_L^{-1}(a^{-1}(\overline{X'}^{-1}L)) \\
    & = dr_L^{-1}(\overline{X a}^{-1}L) \\
    & = \bigcup \{ X \in LW(L') : Y \cap aX = \emptyset \} \\
    & = \bigcup \{ X \in LW(L') : a^{-1}Y \subseteq X \} \\
    & = (\gamma_a)_*(X).
\end{align*}
\]

The final states are preserved and reflected:

\[
\begin{align*}
    X \in dr_L^{-1}(F_{\delta_{\alpha(L')}}) & \iff X \in dr_L^{-1}(\{ Y \in LW(L) : \varepsilon \notin Y \}) \\
    & \iff \varepsilon \in dr_L(X) \\
    & \iff \varepsilon \in \overline{X'^{-1}L} \\
    & \iff \overline{X'} \cap L \neq \emptyset \\
    & \iff \overline{X'} \cap L' \neq \emptyset \\
    & \iff L' \notin X \\
    & \iff X \in F_{\delta_{\alpha(L')}}.
\end{align*}
\]

\[ \square \]

**Note 3.3.7.**

1. $dr_L$ provides a bijection between $L$’s left word quotients $U^{-1}L$ and right word quotients $LV^{-1} = ((U')^{-1}L')^r$.

2. If $L = L'$ we have an order-reversing involutive isomorphism $dr_L : (\mathbb{L}Q(L))^\circ \rightarrow \mathbb{L}Q(L)$. Thus $\mathbb{L}Q(L)$ is a De Morgan algebra whose bounded lattice structure needn’t be distributive. This holds for any unary language. $\blacksquare$

**Corollary 3.3.8 (Meet-generating $\mathbb{L}Q(L)$).**

1. $\mathbb{L}Q(L)$ is join-generated by $LW(L)$ and meet-generated by $dr_L[LW(L')] = \{ \overline{Lw^{-1}}^{-1}L : v \in \Sigma^* \}$.

2. $Y \in dr_L(v^{-1}L') \iff v^r \notin Y$ for each $Y \in \mathbb{L}Q(L)$. 

25
3. Each \( Y \in \mathcal{LQ}(L) \) arises as an intersection:

\[
Y = \bigwedge_{L \in \mathcal{LQ}(L)} (dr_L(v^{-1}L') : v' \notin Y)
= \cap \{dr_L(v^{-1}L') : v' \notin Y\}.
\]

Proof.

1. That \( \mathcal{LQ}(L) \) is generated by \( \mathcal{LW}(L) \) under finite unions follows via Definition 3.3.1. Concerning the new claim, the order isomorphism \( dr_L \) from Theorem 3.3.6 preserves/reflects meet-irreducibles. Then recalling Definition 3.3.1 we have \( M(\mathcal{LQ}(L))^{op} = J(\mathcal{LQ}(L)) \subseteq \mathcal{LW}(L') \), so applying \( dr_L \) we obtain a meet-generating set.

2. Given \( Y \subseteq dr_L(v^{-1}L') \) then since \( dr_L(v^{-1}L') = \bigcup \{X \in \mathcal{LW}(L) : v' \notin X\} \) doesn’t contain \( v' \) we immediately deduce \( v' \notin Y \). Conversely if \( v' \notin Y \) then for every \( Y \subseteq X \in \mathcal{LW}(L) \) we have \( v' \notin X \) hence \( X \subseteq dr_L(v^{-1}L') \), so that \( Y \subseteq dr_L(v^{-1}L') \) too.

3. By (1) each \( Y \in \mathcal{LW}(L) \) is the meet of those \( K \in dr_L[\mathcal{LW}(L')] \subseteq \mathcal{LQ}(L) \) above it. By (2) \( Y \subseteq dr_L(v^{-1}L') \) iff \( v' \notin Y \), which implies the first equality. Finally this meet is actually an intersection: given \( w \in \cap \{dr_L(v^{-1}L') : v' \notin Y\} \) then if \( w \notin Y \) we obtain the contradiction \( w \in dr_L((v')^{-1}L') \).

With reference to the Corollary 3.3.8, the next Lemma explains the strong connection between the state-minimal JSL-dfa and the canonical dependency automaton \((\text{dfa}(L), \mathcal{D}\mathcal{R}_L, \text{dfa}(L'))\) where \( \mathcal{D}\mathcal{R}_L(u^{-1}L, v^{-1}L') : \iff \ u'v' \in L \).

Lemma 3.3.9 (Dependency Lemma). For any regular \( L \subseteq \Sigma^* \) and words \( u, v \in \Sigma^* \),

\[
u^{-1}L \notin [Lv^{-1}]^{-1}L \iff uv \in L \quad \text{or equivalently} \quad u^{-1}L \notin dr_L(v^{-1}L') \iff u'v' \in L \iff \mathcal{D}\mathcal{R}_L(u, v).
\]

Proof. We calculate:

\[
u^{-1}L \notin [Lv^{-1}]^{-1}L \iff \exists y \in \Sigma^*, [uy \in L \text{ and } y \notin [Lv^{-1}]^{-1}L]
\iff \exists y \in \Sigma^*, [uy \in L \text{ and } \forall x \in \Sigma^*, [xy \notin L \Rightarrow x \notin Lv^{-1}]]
\iff \exists y \in \Sigma^*, [uy \in L \text{ and } \forall x \in \Sigma^*, [x \notin Ly^{-1} \Rightarrow x \notin Lv^{-1}]]
\iff \exists y \in \Sigma^*, [u \notin Ly^{-1} \text{ and } Ly^{-1} \subseteq Lv^{-1}]
\iff \exists u \in Ly^{-1}
\iff \exists u \in Lv^{-1}
\iff uv \in L.
\]

If \( L \) is regular we may rewrite this in terms of \( dr_L \) and \( \mathcal{D}\mathcal{R}_L \) as above, see Theorem 3.3.6 and Definition 3.1.8.

We are now ready for the main result of this subsection.

Theorem 3.3.10 (Dependency Theorem). The state-minimal JSL-dfa is isomorphic to the determinisation of the canonical dependency automaton.

\[
\alpha : \text{dfa}(L) \to \text{det}(\text{dfa}(L), \mathcal{D}\mathcal{R}_L, \text{dfa}(L'))
\]

\[
\alpha(X) := \{v^{-1}L' : v \in X\} \quad \alpha^{-1}(Y) := \left( (\bigcap Y) \right)^{-1}L
\]

Proof.

1. We first establish the underlying isomorphism \( \alpha : \mathcal{S} \to \text{Open}\mathcal{D}\mathcal{R}_L \) where \( \mathcal{S} := \mathcal{LQ}(L) \).

By Theorem 2.2.14 we have the isomorphism \( \text{reps} : \mathcal{S} \to \text{OpenPirr}\mathcal{S} \) where \( \text{reps}(X) := \{Y \in M(\mathcal{S}) : X \notin Y\} \). Proposition 2.2.21 permits one to extend the domain/codomain of \( \text{Pirr}\mathcal{S} \) to join/meet generators, which by Corollary 3.3.8 can be \( J_S := \mathcal{LW}(L) \) and \( M_S := \{L^{-1}v^{-1}L : v \in \Sigma^*\} \). Then we obtain the \( \text{Dep}\)-isomorphism:

\[
\mathcal{I}_S^{-1} := \mathcal{S} \mid_{J(S) \times M_S} : \text{Pirr}\mathcal{S} \to \mathcal{S} \mid_{J_S \times M_S} \quad \text{where } (\mathcal{I}_S^{-1})^{-1} : (y, m) : \iff y \leq m
\]

and thus \( \text{Open}\mathcal{I}_S^{-1} : \text{OpenPirr}\mathcal{S} \to \text{Open} \mathcal{S} \mid_{J_S \times M_S} \) is an isomorphism with action \( \lambda X. \mathcal{LW}(L') \cap \downarrow P X \). Next, we’ll establish the bipartite graph isomorphism:

\[
\begin{array}{ccc}
M_S & \xrightarrow{\alpha} & \mathcal{LW}(L') \\
\downarrow \lambda X. \mathcal{D}\mathcal{R}_L(X) & & \downarrow \mathcal{D}\mathcal{R}_L \\
J_S & \xrightarrow{id_{\mathcal{LW}(L)}} & \mathcal{LW}(L)
\end{array}
\]
The upper bijective witness is well-defined because \( M_S = d_{LW}(L') \); this Rel-diagram commutes by the Dependency Lemma 3.3.9. It defines a Dep-isomorphism \( \mathcal{DR}_L : \xi_S |_{J_S \times M_S} \rightarrow \mathcal{DR}_L \) and hence a JSL-\( \tau \)-isomorphism \( \text{Open} \mathcal{DR}_L := \lambda X. d_{LW}(L') [X] \) recalling that any upper witness can be used by Note 2.2.11. Then we have the composite isomorphism:

\[
S \xrightarrow{\text{rep}_S} \text{OpenPirr} S \xrightarrow{\text{rev}_L} \text{Open} \xi_S |_{J_S \times M_S} \xrightarrow{\text{Open} \mathcal{DR}_L} \text{Open} \mathcal{DR}_L
\]

with action and inverse action:

- \( X \in \text{LQ}(L) \quad \Rightarrow \{ Y \in M(S) : X \notin Y \} \) (apply \( \text{rep}_S \))
- \( \Rightarrow \{ Z \in \text{LQ}(L') : \exists Y \in M(S). (X \notin Y \text{ and } Z \subseteq Y) \} \) (apply \( \text{Open} \xi_S^{-1} \))
- \( = \{ Z \in \text{LQ}(L') : X \notin Z \} \) (apply \( \text{Open} \mathcal{DR}_L \))
- \( \Rightarrow d_{LW}(L') \{ Z \in \text{LQ}(L') : X \notin Z \} \) (apply \( \text{Open} \mathcal{DR}_L \))
- \( = \{ v^{-1}L' : v \in X^r \} \) (apply \( \text{Open} \mathcal{DR}_L \))

Concerning (1), each \( \mathcal{DR}_L \)-open set \( S \) is up-closed in \( (\text{LQ}(L'), \subseteq) \) so \( d_{LW}(L') \) is down-closed in \( (M_S, \subseteq) \) recalling \( M(S) \subseteq M_S \). Concerning (2), if \( v^{-1}L' \in S \) some join-irreducible \( v^{-1}L' \subseteq w^{-1}L' \) must also lie in \( S \).

2. It remains to establish that \( \alpha \) is a dfa morphism:

\[
\alpha : \delta \text{fa}(L) \rightarrow (F_{\text{dfa}(L')}, \text{Open} \mathcal{DR}_L, \delta_a, \{ L' \}) \quad \delta_a := \lambda S. \{ Y \in \text{LW}(L') : \exists X \in S a^{-1}Y = X \}
\]

The initial state is preserved because \( \alpha(L) = \{ v^{-1}L' : v \in L' \} = \{ X \in \text{LW}(L') : \exists X \in X \} = F_{\text{dfa}(L')} \). Next we show the transitions are preserved, denoting the domain dfa’s transitions by \( \gamma_a := \lambda X. a^{-1}X \). Given any \( X \in \text{LW}(L) \),

\[
\alpha \circ \gamma_a(X) = \alpha(a^{-1}X) = \{ v^{-1}L' : v \in (a^{-1}X)^r \} = \{ v^{-1}L' : v \in X^r a^{-1} \}
\]

\[
\delta_a \circ \alpha(X) = \delta_a(\{ w^{-1}L' : w \in X^r \}) = \{ v^{-1}L' : \exists w \in X^r a^{-1}(v^{-1}L' \subseteq w^{-1}L') \} = \{ v^{-1}L' : \exists w \in X^r (va)^{-1}L' \subseteq w^{-1}L' \}.
\]

Let us establish \( (a) = (b) \) via mutual inclusions.

- \( (a) \subseteq (b) \): Given \( v \in X^r a^{-1} \) we deduce \( w := va \in X^r \), hence \( v^{-1}L' \) resides in \( (b) \).
- \( (b) \subseteq (a) \): We may assume \( X := (w')^{-1}L' \); we know there exists \( w \in X^r \) such that \( (va)^{-1}L' \subseteq w^{-1}L' \). Then \( wu \in L' \) hence \( u \in w^{-1}L' \) so that \( va \in L' \). Thus \( va \in L' a^{-1} \) i.e. \( v \in X^r a^{-1} \) so \( v^{-1}L' \) resides in \( (a) \).

Lastly the final states are preserved and reflected:

\[
X \in \alpha^{-1}(\{ L' \}) \iff \alpha(X) = L' \iff \varepsilon \in X^r \iff \varepsilon \in X.
\]

Note 3.3.11 (Canonical dependency automaton as canonical residual automata).

27
By the Dependency Theorem 3.3.10, the canonical dependency automaton corresponds to the state-minimal dfa interpreted in join-semilattices. On the other hand, the categorical equivalence \( \text{dfa}_{\text{JSL}} \equiv \text{aut}_{\text{Dep}} \) of Theorem 3.2.6 already provides the component isomorphism:

\[
\text{rep}_{\text{dfa}(L)} : \text{dfa}(L) \to \text{Det} ( \text{Airr}( \text{dfa}(L) )) = \text{Det}( N_L, \text{Pirr} L(L), N') .
\]

The lower \( N_L \) is precisely the canonical residual automaton of [DLT01]. That is, let \( \text{ILQ}(L) := J( \text{ILQ}(L) ) \subseteq \text{LW}(L) \) be the irreducible left quotients i.e. those left word quotients not arising as the union of others (so, non-empty). Then:

\[
N_L = ( \text{ILQ}(L) \cap \downarrow \text{ILQ}(L)) L, \text{ILQ}(L), N_a, \{ X \in \text{ILQ}(L) : x \in X \} \quad N_a(X_1, X_2) : \iff X_2 \subseteq a^{-1} X_1.
\]

Relabelling the upper bipartition we obtain \( (N_L, \text{DR}_L |_{\text{ILQ}(L)} \cap \downarrow \text{ILQ}(L'), N_L') \). The upper nfa is the canonical residual nfa for \( L' \). The bipartitioned graph is obtained by restricting the dependency relation to irreducibles. It is necessarily aut_{Dep}-isomorphic to the canonical dependency automaton, and actually constructable from it in polytime. It is never larger than our chosen description and potentially far smaller.

### 3.4 Explaining Brzozowski’s algorithm

Recalling Definition 3.1.3, the minimisation of a classical dfa \( \delta := (z_0, Z, \delta_a, F) \) can be understood as follows:

\[
\begin{array}{c}
\delta \\
\downarrow \delta_1 \\
\text{reach}(\delta) \\
\downarrow \delta_2 \\
\text{reach}(\text{simple}(\delta)) \\
\downarrow \delta_3 \\
\text{simple}(\text{reach}(\delta)) \\
\downarrow \delta_4 \\
\text{dfa}(L)
\end{array}
\]

Traditionally one first takes \( \text{reach}(\delta) \) by restricting to states reachable from \( z_0 \) via the underlying directed graph \( \cup_{\delta S} : \delta \in Z \times Z \). From the perspective of dfa morphisms we construct the minimal sub-dfa of \( \delta \) (i.e. the inclusion \( \iota_1 \) above). Secondly one can apply Hopcroft’s algorithm to compute a partition of the states i.e. the equivalence classes over which the state-minimal dfa can then be defined. From the perspective of dfa morphisms we construct the largest quotient-dfa of \( \delta \) (i.e. the surjection \( \text{acc}_{\text{reach}(\delta)} \) above)\(^6\). The latter sends a state to the language it accepts, yielding precisely the state-minimal machine \( \text{dfa}(L) \). Notice the other way to minimise \( \delta \): quotient first; restrict to reachable second.

We expressed minimisation in terms of dfa morphisms because one has exactly the same situation in \( \text{dfa}_{\text{JSL}} \), whose morphisms must also preserve the join-semilattice structure. For any JSL-dfa \( \delta := (s_0, S, \delta_a, F) \),

\[
\begin{array}{c}
\delta \\
\downarrow \delta_1 \\
\text{reach}(\delta) \\
\downarrow \delta_2 \\
\text{reach}(\text{simple}(\delta)) \\
\downarrow \delta_3 \\
\text{simple}(\text{reach}(\delta)) \\
\downarrow \delta_4 \\
\text{dfa}(L)
\end{array}
\]

We’ve already seen the state-minimal \( \text{dfa}(L) \) and its close connection to the canonical dependency automaton. We now introduce the corresponding concepts of reachability and simplicity, recalling the notation of Definition 3.1.3.

**Definition 3.4.1 (JSL-reachability and simplicity).** Let \( \delta := (s_0, S, \delta_a, F) \) be a JSL-dfa.

1. \( \delta \) is JSL-reachable if it has no proper sub JSL-dfas: every injective JSL-dfa morphism \( f : \gamma \to \delta \) is an isomorphism.

   Given any \( R \subseteq S \) with \( s_0 \in R \) and \( \delta_a(R) \subseteq R \) for \( a \in \Sigma \), then \( R \cap \delta := (s_0, R, \delta_a\mid_{R \times R}, F \cap R) \) is a JSL-dfa accepting \( L(\delta) \). In particular,

   \[
   \text{reach}(\delta) := \text{reach}(\delta) \cap \delta \quad \text{where} \quad \text{reach}(\delta) := \langle \text{reach}(\delta) \rangle_S \quad \text{is the reachable sub JSL-dfa of} \ \delta.
   \]

\(^6\)By largest quotient we mean the respective equivalence relation is the largest w.r.t. inclusion. The respective quotient-dfa actually has the least possible number of states amongst other such quotients.

\(^7\)By \( \langle \text{reach}(\delta) \rangle_S \) we mean the sub join-semilattice of \( S \) generated by \( \text{reach}(\delta) \).
2. \( \delta \) is simple if it has no proper quotient JSL-dfas: every surjective JSL-dfa morphism \( f : \delta \to \gamma \) is an isomorphism. We have the join-semilattice of accepted languages \( \text{langs}(\delta) := (\text{langs}(\delta), \cup, \emptyset) \) by Definition 3.1.3.5 and Lemma 3.2.2. Then the simple JSL-dfa:

\[
\text{simple}(\delta) := (L, \text{langs}(\delta), \lambda X. a^{-1}X, \{ X \in \text{langs}(\delta) : \varepsilon \in X \})
\]

is the largest quotient JSL-dfa of \( \delta \) via \( \text{acc}_4 \). Finally, a JSL-dfa \( \delta \) is simplified if \( \text{simple}(\delta) = \delta \).

**Lemma 3.4.2** (Well-definedness of \( \text{reach}(\cdot) \) and \( \text{simple}(\cdot) \)).

1. \( \text{reach}(\delta) \) is the JSL-reachable sub-dfa of \( \delta \).
2. \( \text{simple}(\delta) \) is the simple quotient dfa of \( \delta \).
3. \( L(\text{simple}(\delta)_{@X}) = X \) for each \( X \in \text{simple}(\delta) \).

**Proof.**

1. \( R \cap \delta \) is a well-defined JSL-dfa: (a) the conditions ensure each \( \delta_a : S \to S \) restricts to an \( R \)-endomorphism, (b) just as \( F = h^{-1}(\{1\}) \) for some \( h : S \to S \). Concerning well-definedness of \( \text{reach}(\delta) \), \( R := \text{reach}(\delta) \) is the reachable part of the underlying classical dfa closed under all \( S \)-joins. Certainly \( R \subseteq S \) and \( a \in R \). Next, \( \delta_0[R] \subseteq R \) because \( \delta_0 \) preserves all joins, so applying \( \delta_a \) to joins of classically reachable states is the same as taking the join of \( a \)-successors of classically reachable states. It accepts \( L \) because the reachable part of its underlying classical dfa is precisely \( \text{reach}(\delta) \).

Finally, \( \text{reach}(\delta) \) is JSL-reachable because any sub JSL-dfa must at least contain the underlying reachable part and be closed under the algebraic structure.

2. Concerning well-definedness of \( \text{simple}(\delta) \), \( \text{langs}(\delta) \) is closed under arbitrary unions by Lemma 3.2.2. Certainly \( L \in \text{langs}(\delta) \) and the transitions are well-defined by Definition 3.1.3.2. The final states are well-defined because the union of all languages sans \( \varepsilon \) does not contain it either.

It accepts \( L \) because the reachable part of its underlying classical dfa is precisely \( \text{dfa}(L) \). Finally, \( \text{acc}_3 : \delta \to \text{langs}(\delta) \) is additionally a join-semilattice morphism by Lemma 3.2.2. It is simple because each state \( X \in \text{langs}(\delta) \) accepts \( X \) i.e. distinct states accept distinct languages, so there can be no quotient dfa and thus also no quotient JSL-dfa.

3. Follows via Definition 3.1.8.2. \( \square \)

However, the self-duality of \( \text{dfa}_{\text{JSL}} \) provides an additional relationship.

**Theorem 3.4.3** (JSL-reachability is dual to simplicity). Let \( \delta := (S_0, S, \delta_a, F) \) be a JSL-dfa.

1. \( \delta \) is JSL-reachable iff every join-irreducible \( j \in J(S) \) is classically reachable.
2. \( \delta \) is simple iff distinct states accept distinct languages.
3. \( \delta \) is JSL-reachable iff its dual \( \delta^* \) is simple.

**Proof.** \( \delta \) is JSL-reachable iff \( \text{reach}(\delta) = \delta \) iff every state is a join of classically reachable states. Since \( J(S) \) is the minimal join-generating set we infer (1). Concerning (2), \( \delta \) is simple iff \( \text{acc}_4 : \delta \to \text{langs}(\delta) \) is bijective iff distinct states accept distinct languages. Finally, the concepts of JSL-reachable and simple are categorically dual, recalling JSL-\( \gamma \)-monos are precisely the injective morphisms and JSL-\( \gamma \)-epis are precisely the surjective ones (see Note 2.2.8.4). \( \square \)

We also mention a basic characterisation of simplified JSL-dfas.

**Lemma 3.4.4** (Simplified JSL-dfas). For any JSL-dfa \( \gamma \) t.f.a.e.

1. \( \gamma \) is simplified i.e. \( \text{simple}(\gamma) = \gamma \).
2. There exists a finite set of regular languages \( S \ni L(\gamma) \), closed under unions and left-letter quotients s.t.

\[
\gamma = (L(\gamma), (S, \cup, \emptyset), \lambda X. a^{-1}X, \{ K \in S : \varepsilon \in K \}).
\]
Proof. Given (1) then (2) follows by choosing $S := \text{langs}(\gamma)$, recalling $L(\gamma_a(z)) = a^{-1}L(\gamma_a)$ by Lemma 3.1.3.5. Given (2), the specified quadruple is a well-defined JSL-dfa because $a^{-1}(-)$ preserves unions and there is a largest non-final state $\cup \{K \in S : z \in S \}$. □

Corollary 3.4.5 (simple(−) is the De Morgan dual of reach(−)).

\[ \text{acc}(\text{reach}(\delta)^*) \colon (\text{reach}(\delta^*))^* \to \text{simple}(\delta) \]

is an isomorphism for any JSL-dfa $\delta$.

Proof. Given $\delta := (s_0, S, \delta_a, F)$ there is an injective JSL-dfa morphism $\iota : \text{reach}(\delta^*) \to \delta^*$ by Lemma 3.4.2. By Theorem 3.2.3 we have:

\[ \delta \xrightarrow{\lambda_{\delta}} (\delta^*)^* \overset{\iota_*}{\rightarrow} (\text{reach}(\delta^*))^* \]

where the identity function $\lambda_{\delta} := \text{id}_{\delta}$ is a component of the natural isomorphism witnessing self-duality, and $\iota_*$ is surjective by Note 2.2.8.4. By Theorem 3.4.3.3 the codomain is simple, so the surjective morphism $\text{acc}(\text{reach}(\delta^*))^*$ is an isomorphism. $\text{langs}(\text{reach}(\delta^*))^* = \text{langs}(\delta)$ because $\iota_*$ is surjective, hence $\text{acc}(\text{reach}(\delta^*))^*$ has codomain $\text{simple}(\delta)$. □

Example 3.4.6 (Dualising the reachable subset construction).

In Example 3.2.12 we described the dual of the full subset construction. Again letting $\delta = \text{Det}(\mathcal{N}, \Delta, \text{rev}(\mathcal{N}))$, we now provide a description of $\gamma^*$ where $\gamma := \text{reach}(\delta)$. By Corollary 3.4.5 we have the isomorphism:

\[ \text{acc}_{\gamma^*} : \gamma^* \to \text{simple}(\delta^*) \]

sending unions of reachable subsets to their accepted language via the JSL-dfa $\gamma^*$. We now describe this isomorphism in more detail. First we write $\gamma = (I, S, \lambda X, N_\gamma[X], \{X \in S : X \cap F \neq \emptyset\})$, so that:

\[ \gamma^* = (\text{reach}(\mathcal{N}) \cap \mathcal{F}, \mathcal{S}^{\mathcal{F}}, \beta_\gamma, \{X \in S : I \nsubseteq X\}) \]

where $\beta_\gamma := (\gamma_a)^*$. Then $\beta_w = (\gamma_w)^* = \lambda Y. \bigcup \{X \in \text{rs}(\mathcal{N}) : N_{w^*}[X] \subseteq Y\}$ since the reachable subsets $\text{rs}(\mathcal{N})$ join-generate $\gamma$. Next,

\[ \begin{align*}
  w \in \text{acc}_{\gamma^*}(Y) & \iff \beta_w(Y) \in F_{\gamma^*} \quad \text{(by def.)} \\
  & \iff I \notin \beta_w(Y) \\
  & \iff -(I \subseteq \bigcup \{X \in \text{rs}(\mathcal{N}) : N_{w^*}[X] \subseteq Y\}) \quad \text{(see above)} \\
  & \iff -(N_{w^*}[I] \subseteq Y) \\
  & \iff N_{w^*}[I] \nsubseteq Y.
\end{align*} \]

Concerning the marked equivalence, $\iff$ follows immediately because $I \in \text{rs}(\mathcal{N})$. Conversely if for each $z \in I$ we have $z \in X_z \in \text{rs}(\mathcal{N})$ with $N_{w^*}[X_z] \subseteq Y$ then $N_{w^*}[I] \subseteq Y$ as well. Thus we obtain a more explicit description of the isomorphism i.e. $\text{acc}_{\gamma^*}(Y) = \{w \in \Sigma^* : N_{w^*}[I] \nsubseteq Y\}$. □

Corollary 3.4.7. reach(−) preserves simplicity and simple(−) preserves JSL-reachability.

Proof. If $\delta$ is simple then it is isomorphic to $\gamma := \text{simple}(\delta)$ so distinct states accept distinct languages by Theorem 3.4.3.2. Ignoring the join-structure, reach($\gamma$) is a sub-dfa of $\gamma$, so distinct states continue to accept distinct languages and reapplying Theorem 3.4.3.2 we deduce simplicity. The second statement follows by duality i.e. Theorem 3.4.3.3. □

Corollary 3.4.8 (Characterisation of dfaJSL-minimality). A JSL-dfa $\delta$ is JSL-reachable and simple iff $\text{acc}_\delta : \delta \rightarrow \text{dfa}(L(\delta))$ is a well-defined JSL-dfa isomorphism.

Proof. Let $L := L(\delta)$. Suppose $\text{acc}_\delta : \delta \rightarrow \text{dfa}(L)$ has correct typing and is an isomorphism. The notions of ‘JSL-reachable’ and ‘simple’ are invariant under isomorphism, so we show $\gamma := \text{dfa}(L)$ is JSL-reachable and simple. Firstly, reach($\gamma$) = $\gamma$ because the left quotients $\text{LQ}(L)$ arise from $\text{LW}(L)$ via finite unions. Finally, $\gamma$ is simple because $\gamma^* \nsubseteq \text{dfa}(L^*)$ by Theorem 3.3.6 which is JSL-reachable by the preceding argument, so $\gamma$ is simple by Theorem 3.4.3.3.

Conversely let $\delta$ be JSL-reachable and simple. Since $\delta$ is simple, the surjection $\text{acc}_\delta : \delta \rightarrow \text{simple}(\delta)$ is an isomorphism. Since $\delta$ is JSL-reachable, by Theorem 3.4.3.1 and Lemma 3.2.2 we have $\text{langs}(\delta) = \text{LQ}(L)$, hence $\text{simple}(\delta) = \text{dfa}(L(\delta))$. □
Corollary 3.4.9 (Meet-generators for JSL-dfas). Let \( \gamma = (s_0, S, \gamma_a, F) \) be a JSL-dfa.

1. If \( \gamma \) is simplified it is meet-generated by \( \{ \cup \{ j \in J(\text{langs}(\gamma)) : w \notin j \} : w \in \Sigma^* \} \).

2. If \( \gamma \) is JSL-reachable it is meet-generated by \( \{ \forall z \in J_\gamma(s_0) : w \notin j \} : j \in J(\text{langs}(\gamma^*)) \} \).

Proof.

1. By Lemma 3.4.4 there exists a set \( S \) of union and left-letter-quotient closed languages s.t.

\[
\gamma = (L, (S, \cup, \emptyset), \lambda x . a^{-1} x, \{ K \in S : \varepsilon \notin K \}) \quad \text{where} \quad L := L(\gamma).
\]

By Theorem 3.4.3 we know \( \gamma^* \) is JSL-reachable, hence join-generated by elements \( (\gamma_w)_*(K_1) \) where \( K_1 := \cup \{ K \in S : \varepsilon \notin K \} \). Finally observe that:

\[
(\gamma_w)_*(K_0) = \cup \{ j \in J(S) : \gamma_w(j) \subseteq K_0 \} = \cup \{ j \in J(S) : \varepsilon \notin \gamma_w(j) \} = \cup \{ j \in J(S) : w \notin j \}
\]

so \( S \) is meet-generated by these elements.

2. By Theorem 3.4.3 we may assume (modulo isomorphism) that \( \gamma = \delta^* \) where \( \delta = (t_0, \emptyset, \lambda x . a^{-1} x, \{ K \in T : \varepsilon \notin K \}) \) is simplified. Consequently \( s_0 = \cup \{ K \in T : \varepsilon \notin K \} \) and we calculate:

\[
\forall z \{ \gamma_w(s_0) : w \notin j \} = \Lambda \{ (\delta_w)_*(s_0) : w \notin j \} = \Lambda \{ (\cup \{ j' \in J(T) : w^r \notin j' \} : w \notin j \} = \Lambda \{ (\cup \{ j' \in J(T) : w \notin j' \} : w \notin j \} = j
\]

Concerning the marked equality: \( \subseteq \) follows because if \( w \notin j \) then \( w \notin \{ j' \in J(T) : w \notin j' \} \); \( \supseteq \) follows because whenever \( w \notin j \) we know \( j \notin \{ j' \in J(T) : w \notin j' \} \). Finally, \( J(T) \) join-generates \( T \) and thus meet-generates \( S \).

\( \square \)

The self-duality of \( \text{df}_{\text{JSL}} \) (Theorem 3.2.3) corresponds to the self-duality of \( \text{aut}_{\text{Dep}} \) (Theorem 3.1.14). But what does JSL-reachability correspond to at the level of dependency automata? Our next result shows it is a combination of the classical reachable subset construction and the classical reachable nfa construction.

Theorem 3.4.10 (aut_{Dep}-reachability). We have the \( \text{aut}_{\text{Dep}} \)-isomorphism:

\[ \widetilde{\varepsilon} : \text{Airr}(\text{reach}(\text{dep}(N))) \rightarrow (\text{rsc}(N), \varepsilon, \text{rev}(\text{reach}(N))). \]

for each nfa \( N = (z_0, Z, N_0, F) \).

Proof. By Note 3.2.7 the determinisation \( \delta := \text{Det}(\text{dep}(N)) \) is \( N \)'s full subset construction endowed with its join-semilattice structure \( \text{Open}_{DZ} = (PZ, \cup, \emptyset) \). Consider:

\[ \text{reach}(\delta) = (I, S, \gamma_a, F_v), \quad S := \text{reach}(\delta) = (\text{rs}(N))_{\text{Open}_{DZ}}, \quad \iota : S \rightarrow \text{Open}_{DZ}. \]

Then \( S \) is join-generated by \( J_S := \text{rs}(N) \) but what about a meet-generating set? The surjective adjoint \( \iota_a : (PZ, \cap, Z) \rightarrow S_{\text{op}} \) provides one:

\[ M(S) = J(S_{\text{op}}) \subseteq \iota_a(J(PZ, \cap, Z)) = \{ M_z : z \in Z \} \quad \text{where} \quad M_z := \iota_a(\overline{z}). \]

Since \( \text{rs}(N) \) join-generates \( S \) we know \( M_z = \cup \{ X \in \text{rs}(N) : z \notin X \} \) is the union of reachable subsets without \( z \). It follows that \( M_S := \{ M_z : z \in \text{reach}(N) \} \supseteq M(S) \) because if \( z \) is unreachable then \( M_z = \text{reach}(N) = T_\varepsilon \notin M(S) \) cannot contribute. Now, Airr\(\delta \) is isomorphic to \( (M, \varepsilon, M') \) by Proposition 3.2.16 where:

\[
\begin{align*}
M_a(X_1, X_2) : & \iff X_2 \subseteq N_a[X_1] \\
M_a'(M_{z_1}, M_{z_2}) : & \iff (\delta_a)(M_{z_1}) \subseteq M_{z_2}.
\end{align*}
\]
The lower \( N \) turns out to be \( \text{rsc}(N) \) with some additional degenerate structure, we'll come back to this point. Concerning the upper \( N \), the calculations:

\[
\begin{align*}
\mathcal{F} \cap S & \subseteq M_z \quad \text{(adjoints)} \quad \iff \mathcal{I}(\mathcal{F} \cap S) \subseteq \mathcal{I} \quad \iff \ z \notin \mathcal{F} \cap S \iff \ z \in F \cap \mathcal{R}(N) \\
I & \notin M_z \quad \text{(adjoints)} \quad \iff \mathcal{I}(I) \notin \mathcal{I} \iff \ z \in I.
\end{align*}
\]

\((\delta_a),(M_{z_1}) \subseteq M_{z_2} \iff \mathcal{I}((\delta_a),(M_{z_1})) \subseteq \mathcal{I} \quad \text{(adjoints)} \quad \iff \ z_2 \notin (\delta_a),(M_{z_1}) \\
\iff \forall X \in \text{rs}(N).[\gamma_2(X) \subseteq M_{z_1} \Rightarrow z_2 \notin X] \\
\iff \forall X \in \text{rs}(N).[z_2 \notin X \Rightarrow \gamma_a(X) \notin M_{z_1}] \\
\iff \forall X \in \text{rs}(N).[z_2 \notin X \Rightarrow z_1 \notin \gamma_a(X)] \quad \text{(via adjoints)}
\]

\( \mathcal{N}_a(z_{2},z_{1}) \quad \text{(since \( z_2 \) reachable)} \)

show that it is essentially \( \text{rev}(\mathcal{R}(N)) \). More precisely we have the bipartite graph isomorphism:

\[
\begin{array}{ccc}
M_{\Sigma} & \xrightarrow{\beta} & \mathcal{R}(N) \\
\varepsilon & \downarrow & \text{id}_{\mathcal{R}(N)} \\
\text{rs}(N) & \xrightarrow{\text{rs}(\mathcal{N})} & \mathcal{R}(N)
\end{array}
\]

where the bijection \( \beta \) has action \( M_{\Sigma} \rightarrow z \). Indeed \( X \notin M_z \iff X \notin \mathcal{I} \iff z \in X \iff \bar{\varepsilon}(X,z) \). Then it follows from the earlier calculations that we have the \( \text{auto}_{\text{dep}} \)-isomorphism \( \varepsilon : \text{Airr} \delta \rightarrow (M,\varepsilon,\text{rev}(\mathcal{R}(N))) \). Instantiating Proposition 3.1.15 provides the isomorphism \( \bar{\varepsilon} : (M,\varepsilon,\text{rev}(\mathcal{R}(N))) \rightarrow (\text{rsc}(N),\bar{\varepsilon},\text{rev}(\mathcal{R}(N))) \). This follows by the calculations \( M_a;\bar{\varepsilon}(X,z) = 3X' \in \text{rs}(N).[X' \subseteq \mathcal{N}_a[X] \land z \in X'] \iff z \in \mathcal{N}_a[X] = \mathcal{N}_a[N_a[Y]];\bar{\varepsilon}(X,z) \) and \( \bar{\varepsilon}([X \in \text{rs}(N) : X \subseteq I] = I = \bar{\varepsilon}([\{I\}]. \) The third requirement in Proposition 3.1.15 is trivial because both dependency automata have the same upper \( N \). Composing these two \( \text{auto}_{\text{dep}} \)-isomorphisms yields:

\[
\bar{\varepsilon} \circ \varepsilon = \text{id}_{\text{rs}(N)};\bar{\varepsilon} = \bar{\varepsilon} : \text{Airr}(\text{rsc}(\text{Det}(N),\Delta_Z,\text{rev}(N))) \rightarrow (\text{rsc}(N),\varepsilon,\text{rev}(\mathcal{R}(N)))
\]

i.e. relate a \( \text{join-irreducible reachable subset} \ Y \) to its elements \( z \in Y \) – all classically reachable in the \( N \).

**Note 3.4.11** (Reachability in \( \text{auto}_{\text{dep}} \)). Given the full subset construction \( \delta = \text{Det}(N,\Delta_Z,\text{rev}(N)) \), Theorem 3.4.10 describes \( \text{trach}(\delta) \) as a dependency automaton. What about for arbitrary JSL-dfas? In a sense we’ve already covered the general case via Corollary 3.2.15. The JSL-dfas with carrier \( \text{open}_{\Delta_Z} = (PZ,\cup,\emptyset) \) are injective objects and every JSL-dfa embeds into one.

**Theorem 3.4.12** (\( \text{auto}_{\text{dep}} \)-simplicity). We have the \( \text{auto}_{\text{dep}} \)-isomorphism:

\[
\mathcal{I} : (\text{coreach}(N),\varepsilon,\text{rsc}(\mathcal{R}(N))) \rightarrow \text{Airr}(\text{simple}(\text{dep}(N)))
\]

\[
\mathcal{I}(z,Y) \iff L(N_{\Delta Z}) \cap \overline{Y} ^* \neq \emptyset.
\]

**Proof.** Let \( \delta = \text{Det}(\text{dep}(N)) \) and apply the duality of Theorem 3.1.14 to the isomorphism of Theorem 3.4.10:

\[
\mathcal{R} := \varepsilon : (\text{rev}(\mathcal{R}(N)),\varepsilon,\text{rs}(N)) \rightarrow \text{Rev}(\text{Airr}(\text{rsc}(\mathcal{R}(\delta))))
\]

Observe \( \mathcal{R} \) has bijective lower witness \( \alpha := \lambda z.\cup\{X \in \text{rs}(N) : z \notin X\} \) by inspecting the proof of Theorem 3.4.10. By Theorem 3.2.13.2 we have \( \lambda : \text{Rev} \circ \text{Airr} \circ \rightarrow \text{Airr} \circ (-)^* \) and hence the component:

\[
\hat{\lambda}_{\text{rach}(\delta)} = \text{id}_{\text{Airr}(\text{rsc}(\mathcal{R}(\delta)))} = \text{Pirr}(\text{rs}(N))_{\text{open}_{\Delta_Z}}
\]

whose domain is the codomain of \( \mathcal{R} \) and whose codomain is \( \text{Airr}(\text{rsc}(\mathcal{R}(\delta)))^* \). Corollary 3.4.5 provides the isomorphism:

\[
f := \text{acc}(\text{rach}(\delta))^* : (\text{rach}(\delta))^* \rightarrow \text{simple}(\delta^*)
\]

32
and thus the $\text{aut}_{\text{Dep}}$-isomorphism $\text{Airr}_f$. By Example 3.2.12 we know $\delta^* \cong \text{Det}(\text{Rev}(\text{dep}(N)))$ hence $\text{simple}(\delta^*)$ exactly equals $\text{simple}(\text{Det}(\text{Rev}(\text{dep}(N))))$. Composing these three $\text{Dep}$-isomorphisms yields:

\[
\begin{align*}
\mathcal{R} \; \text{Airr}_f(z, Y) & \iff \alpha; \text{Airr}_f(z, Y) \\
& \iff \{ w \in \Sigma^* : w \in \text{rs}(N) \} \subseteq \text{rs}(\delta^*) Y \\
& \iff \{ w \in \Sigma^* : w \notin \text{rs}(N) \} \subseteq \text{rs}(\delta^*) Y \quad \text{(by Example 3.4.6)}
\end{align*}
\]

Finally we reparameterise via $N \mapsto \text{rev}(N)$ recalling that $\text{coreach}(N) = \text{rev}(\text{reach}(\text{rev}(N)))$ by definition.

We can now explain the original motivation for the above results.

**Theorem 3.4.13** (Brzozowski construction of state-minimal dfa). We have the $\text{dfa}$-isomorphism:

\[
\text{acc}_{\text{rsc}((\text{rev}(\text{rsc}(\text{rev}(N))))))} : \text{rsc}(\text{rev}(\text{rsc}(\text{rev}(N)))) \rightarrow \text{dfa}(L(N))
\]

for any nfa $N = (I, Z, N_a, F)$.

**Proof.** Consider the dependency automaton $(N, \Delta, \text{rev}(N))$. By Theorem 3.4.12 its simplification amounts to $\mathcal{R} := (\text{coreach}(N), \epsilon, \text{rsc}(\text{rev}(N)))$. Then $\text{Det}\mathcal{R}$ is a simple $\text{JSL}$-dfa. By Corollary 3.4.7, $\text{reach}(\text{Det}\mathcal{R})$ is both simple and $\text{JSL}$-reachable, hence isomorphic to $\text{simple}(L(N))$ by Corollary 3.4.8. Thus the classically reachable part $\text{reach}(\text{Det}\mathcal{R})$ is isomorphic to $\text{dfa}(L(N))$. Finally by Note 3.2.10 we have $\text{reach}(\text{Det}\mathcal{R}) = \text{rsc}(\text{rev}(\text{rsc}(\text{rev}(N))))$.

3.5 Minimal boolean and distributive machines

Recall the state-minimal machine $\text{dfa}(L)$ from Definition 3.1.3. Its states $\text{lw}(L)$ are the left word quotients $u^{-1}L$, also known as Brzozowski derivatives [Brz64, Con71].

**Definition 3.5.1** (Minimal boolean/distributive $\text{JSL}$-dfa). Fix a regular language $L \subseteq \Sigma^*$.

1. $L$’s left predicates and state-minimal boolean $\text{JSL}$-dfa.

   Let $\text{LP}(L)$ be the closure of $\text{lw}(L)$ under all intersections and unions. The subsets define a distributive lattice with underlying join-semilattice $\text{LP}(L) := (\text{LP}(L), \cup, \emptyset)$. Then $J(\text{LP}(L))$ is a distributive lattice with underlying join-semilattice $\text{LD}(L) := (\text{LD}(L), \cup, \emptyset)$. Meet is intersection and its top element is $\Sigma^*$. The canonical boolean $\text{JSL}$-dfa for $L$ is defined:

   $\text{df}_a_\lambda(L) := (L, \text{LP}(L), \lambda X. a^{-1} X, \{ K \in \text{LP}(L) : \epsilon \in K \})$.

2. $L$’s positive left predicates and state-minimal distributive $\text{JSL}$-dfa.

   Let $\text{LD}(L)$ be the closure of $\text{lw}(L)$ under all intersections and unions. The subsets define a distributive lattice with underlying join-semilattice $\text{LD}(L) := (\text{LD}(L), \cup, \emptyset)$. Meet is intersection and its top element is $\Sigma^*$. The canonical distributive $\text{JSL}$-dfa for $L$ is defined:

   $\text{df}_a_\lambda(L) := (L, \text{LD}(L), \lambda X. a^{-1} X, \{ K \in \text{LD}(L) : \epsilon \in K \})$.

**Note 3.5.2** (Canonicity of $\text{JSL}$-dfas).

We briefly explain the sense in which these $\text{JSL}$-dfas are canonical, see [MAMU14].

- $\text{df}_a_\lambda(L)$ is the underlying $\text{JSL}$-dfa of the state-minimal BA-dfa.
- $\text{df}_a_a_\lambda(L)$ is the underlying $\text{JSL}$-dfa of the state-minimal DL-dfa.
In the remainder of this subsection, we’ll describe the canonical boolean/distributive JSL-dfas as dependency automata. This immediately provides representations of their dual JSL-dfas. The next subsection is dedicated to the transition-semiring of an nfa. These admit a JSL-dfa structure. In particular, the canonical syntactic JSL-dfa \( \mathcal{Jfa}_{\text{syn}}(L) \) is the dual of syntactic semiring for \( L' \) [Pol01].

**Lemma 3.5.3** (Concerning atoms and finality).

1. The atoms \( J(\mathbb{L}P(L)) \) are pairwise disjoint and their union is \( \Sigma^* \).
2. Given \( u \in \alpha \in J(\mathbb{L}P(L)) \) and \( Y \in \mathbb{L}P(L) \) then \( u \in Y \iff \alpha \subseteq Y \).
3. For any \( Y \in \mathbb{L}P(L) \) we have \( \varepsilon \in Y \iff Y \notin d\mathbb{L}(L') \).

**Proof.**

1. The atoms \( J(\mathbb{L}P(L)) \) are pairwise-disjoint because their meet (intersection) is the bottom element \( \emptyset \). The union of all atoms is the top element i.e. the empty intersection \( \Sigma^* \).
2. Fix any \( Y \in \mathbb{L}P(L) \) and \( u \in \alpha \in J(\mathbb{L}P(L)) \). Given \( \alpha \subseteq Y \) then certainly \( u \in Y \). Conversely if \( u \in Y \) then it must lie in some atom, which is unique by disjointness, hence \( \alpha \subseteq Y \).
3. If \( \varepsilon \in Y \) then certainly \( Y \notin d\mathbb{L}(L') \) because the latter does not contain \( \varepsilon \) (see Theorem 3.3.6). Conversely if \( Y \notin d\mathbb{L}(L') \) there exists \( u \in Y \) such that \( \forall X \in L\mathbb{W}(L)(u \in X \iff \varepsilon \in X) \) i.e. \( \alpha \) and \( \varepsilon \) reside in the same atom, so \( \varepsilon \in Y \) by (2).

**□**

**Lemma 3.5.4** (Well-definedness of canonical boolean/distributive JSL-dfas).

\( \mathcal{Jfa}_{\alpha}(L) \) and \( \mathcal{Jfa}_{\kappa}(L) \) are well-defined fixpoints of simple(\(-\)) which accept \( L \) and have \( \mathcal{Jfa}(L) \) as a sub JSL-dfa.

**Proof.** The join-semilattice \( \mathbb{L}P(L) \) is closed under unions, thus well-defined. We have \( L \in \mathbb{L}Q(L) \subseteq \mathbb{L}P(L) \) and \( a^{-1}(\cdot) \) preserves unions. The final states are well-defined by Lemma 3.5.3.3. Each state \( K \) accepts \( K \) i.e. \( w \in K \iff \varepsilon \in w^{-1}K \iff w^{-1}K \notin d\mathbb{L}(L') \) where the latter corresponds to JSL-dfa acceptance. Then it is a fixpoint of simple(\(-\)) as claimed and clearly has the sub JSL-dfa \( \mathcal{Jfa}(L) \). Finally \( \mathcal{Jfa}_{\alpha}(L) \) is sandwiched between them via JSL-dfa inclusion morphisms, with well-defined final states by Lemma 3.5.3.3.

**□**

Generally speaking, \( L' \)'s left word quotients biject with \( \mathbb{L}P(L) \)'s atoms.

**Theorem 3.5.5** (Quotient-atom bijection). Each regular \( L \) has the canonical bijection:

\[
\kappa_L : L\mathbb{W}(L') \rightarrow J(\mathbb{L}P(L)) \quad \kappa_L(v^{-1}L') := \llbracket v^{-1} \rrbracket \varepsilon_L \quad \kappa_L^{-1}(X) := [X^{-1}]^{-1}L',
\]

and respective relationship:

\[
\kappa_L(x^{-1}L') \subseteq a^{-1}\kappa_L(y^{-1}L') \iff (xa)^{-1}L' = y^{-1}L' \quad \text{for any } x, y \in \Sigma^*.
\]

**Proof.**

1. We first verify \( \kappa_L \) is a well-defined function:

\[
\begin{align*}
v^{-1}_1L' = v^{-1}_2L' & \iff \forall w \in \Sigma^*, [v_1w \in L' \iff v_2w \in L'] \\
& \iff \forall w \in \Sigma^*, [wv^{-1}_1 \in L \iff wv^{-1}_2 \in L] \\
& \iff \forall w \in \Sigma^*, [v^{-1}_1w \in L' \iff v^{-1}_2w \in L'] \\
& \iff \llbracket v^{-1}_1 \rrbracket \varepsilon_L = \llbracket v^{-1}_2 \rrbracket \varepsilon_L \quad \text{(by Lemma 4.4.3.6).}
\end{align*}
\]

It is clearly surjective and also injective by reversing the argument above. The action of \( \kappa_L^{-1} \) is well-defined because \( \kappa_L \) is injective.

2. Suppose \( (xa)^{-1}L' = y^{-1}L' \) so that \( [ax \cdot x] \varepsilon_L = [y \cdot y] \varepsilon_L \) by applying \( \kappa_L \). Since \( x \varepsilon \in a^{-1} [ax \cdot x] \varepsilon_L \) we deduce \( [x^{-1}] \varepsilon_L \subseteq a^{-1} [ax \cdot x] \varepsilon_L = a^{-1} [x \cdot x] \varepsilon_L \). Conversely suppose the inclusion \( [x^{-1}] \varepsilon_L \subseteq a^{-1} [y \cdot y] \varepsilon_L \) holds. Then \( ax \varepsilon \in [y \cdot y] \varepsilon_L \) and consequently \( [ax \cdot x] \varepsilon_L = [y \cdot y] \varepsilon_L \), so applying \( \kappa_L^{-1} \) we infer \( (xa)^{-1}L' = y^{-1}L' \).
Note 3.5.6 (Canonicity of $\kappa_L$). $\kappa_L$ arises from the duality between Set-dfas (classical dfas) and BA-dfas i.e. finite deterministic automata interpreted in boolean algebras [MAMU14]. In particular, the dual of the state-minimal BA-dfa for $L$ is isomorphic to the state-minimal Set-dfa for $L'$.

Theorem 3.5.7 (Canonical boolean dependency automaton). We have the aut$\alpha$-isomorphism:

$$\preceq_{\text{L}(L)} \circ \kappa_L : \text{dep}(\text{dfa}(L')) \to \text{Airr}(\text{dfa}_L(L))$$

with action $\lambda v^{-1} L'.[[v]]_{E_L}$ and inverse $\kappa_L^{-1}$.

Proof. Consider the dependency automaton of irreducibles:

$$\text{Airr}(\text{dfa}_L(L)) = (N, \preceq_{\text{dfa}_L(L)}), M$$

$$N = ([x]_{E_L} : x \in L), J(\text{dfa}_L(L)), \mathcal{N}_a, ([e]_{E_L}) \quad \mathcal{N}_a([x]_{E_L}, [x]) \iff (x^*a)^{-1} L' = (x^*)^{-1} L'$$

$$M = ([e]_{E_L}, M(\text{dfa}_L(L)), \mathcal{M}_a, [x]_{E_L} : x \in L) \quad \mathcal{M}_a([x]_{E_L}, [x]) \iff \mathcal{N}_a([x]_{E_L}, [x])$$

To explain, $N$'s description follows by unwinding the definitions and the relationship $[x_2]_{E_L} \preceq a^{-1}[x_1]_{E_L} \iff (x^*_2a)^{-1} L' = (x^*_1)^{-1} L'$ from Theorem 3.5.5. Likewise $M$ follows via the definitions and the following calculation, where $\gamma_a := \lambda X.a^{-1}X : \mathbb{L}(L) \to \mathbb{L}(L)$:

$$\mathcal{M}_a([x_1]_{E_L}, [x_2]_{E_L}) \iff (\gamma_a)([x_1]_{E_L}) \subseteq [x_2]_{E_L} \quad \text{(by definition)}$$

$$\quad \quad \cup ([x]_{E_L} : a^{-1}[x]_{E_L} \subseteq [x_1]_{E_L}) \subseteq [x_2]_{E_L}$$

$$\quad \quad \cup [x_2]_{E_L} \not\subseteq \cup ([x]_{E_L} : a^{-1}[x]_{E_L} \subseteq [x_1]_{E_L})$$

$$\quad \quad a^{-1}[x_2]_{E_L} \not\subseteq [x_1]_{E_L}$$

$$\quad \quad [x_1]_{E_L} \not\subseteq a^{-1}[x_2]_{E_L}$$. We now verify the claimed dependency automaton isomorphism using the canonical bijection $\kappa_L$ from Theorem 3.5.5. First of all, $\alpha := \preceq_{\text{L}(L)} \circ \kappa_L$ defines a Dep-isomorphism via the bijective witnesses:

$$\mathbb{L}(L') \xrightarrow{\preceq_{\text{L}(L)} \circ \kappa_L} \mathbb{M}(\mathbb{L}(L)) \xrightarrow{\kappa_L} \mathbb{J}(\mathbb{L}(L))$$

It remains to verify the constants from Definition 3.1.12. Let $\delta := \text{dfa}(L')$ be the classical state-minimal dfa so that $\delta_a(Y_1, Y_2) \iff Y_2 = a^{-1} Y_1$. Then we calculate:

$$\delta'_a(a^{-1} L', [y]_{E_L}) \iff \exists w \in \Sigma^* [(wa)^{-1} L' = v^{-1} L' \land \alpha(w^{-1} L') = [y]_{E_L}]$$

$$\quad \quad \exists w \in \Sigma^* [(wa)^{-1} L' = v^{-1} L' \land [w']_{E_L} = [y]_{E_L}]$$

$$\quad \quad \exists w \in \Sigma^* [\exists v'.[v']_{E_L} \not\subseteq \alpha^{-1}[v']_{E_L} \land [w']_{E_L} = [y]_{E_L}] \quad \text{(by Theorem 3.5.5)}$$

$$\quad \quad \exists y_{E_L} \subseteq a^{-1}[v']_{E_L}$$

$$\quad \quad M_a([y]_{E_L}, [v']_{E_L})$$

$$\quad \quad \alpha_a([v^{-1} L', [y]_{E_L}])$$

 Concerning the remaining conditions, $\alpha[I_{M}] = \{\kappa_L^{-1}([e]_{E_L})\} = \{L'\} = F_{\text{rev}(\delta)}$ and finally:

$$\alpha[I_{\text{rev}(\delta)}] = \alpha\{Y \in \mathbb{L}(L') : e \in Y\}$$

$$\quad = \alpha\{[v^{-1} L' : v \in L']\}$$

$$\quad = \{[v^*]_{E_L} : v^* \in L\}$$

$$\quad = F_{M}.$$
Corollary 3.5.8 (Dualising δfα_.(L)). We have the JSL-dfa isomorphism:
\[\theta : \text{se}(\text{dfa}(L')) \to (\text{dfa}_-(L'))^* \quad \theta(S) := \bigcup\{[v']_L : v^{-1}L' \notin S\}\]

**Proof.** First recall the inverse of the isomorphism from Theorem 3.5.7,
\[f := \lambda X.[X']^{-1}L' : \text{Airr}(\text{dfa}_-(L)) \to \text{M} \quad \text{M} := \text{dep}(\text{dfa}(L'))\]

It has upper witness \(\beta := \kappa_L^{-1} \circ \text{rep}_-\) so that \(\text{Det}f = \lambda S.\beta[S]\) and also \((\text{Det}f)^* = \lambda S.\beta^{-1}[S]\), since the adjoint of an isomorphism acts like the inverse. Recall the natural isomorphism \(\hat{\theta}\) from Theorem 3.2.13 and rep_ from Theorem 3.2.6. Then we have the composite join-semilattice isomorphism:
\[
\text{Det}(\text{dep}(\text{dfa}(L'))) \xrightarrow{\hat{\theta}_{\text{dep}(\text{dfa}(L'))}^{-1}} (\text{Det})^* \xrightarrow{(\text{Det})^*} (\text{DetAirr}(\text{dfa}_-(L)))^* \xrightarrow{\text{rep}_{\text{dfa}_-(L)}^*} (\text{dfa}_-(L))^*
\]

which acts on \(S \subseteq LW(L')\) as follows:
\[
S \quad \mapsto \quad \hat{\theta}_{\text{dep}(\text{dfa}(L'))}^{-1}(S)
= \quad \Delta_{LW(L')} \bigl(\{S\}\bigr)
\mapsto \quad (\text{Det})^*(\{S\})
= \quad \{[v']_L : v^{-1}L' \in S\} \\ \\
\quad \mapsto \quad \text{rep}_{\text{dfa}_-(L)}^*(\{[v']_L : v^{-1}L' \notin S\})
= \quad \cap\{[v']_L : v^{-1}L' \in S\} \\ \\
\quad \mapsto \quad \cup\{[v']_L : v^{-1}L' \notin S\}.
\]

We now turn our attention to positive predicates, recalling \(dr_L\) from Theorem 3.3.6.

**Lemma 3.5.9** (Concerning irreducibles in \(\text{LD}(L)\)).
1. \((\text{LD}(L))^\text{op} \cong \text{LD}(L)\) via relative complement.
2. \(\text{dr}_L(v^{-1}L') = \cap\{X \in LW(L) : v^r \in X\}\).
3. \(|J(\text{LD}(L))| = |M(\text{LD}(L))| = |LW(L')|\) where:
\[
J(\text{LD}(L)) = \overline{\{\text{dr}_L(v^{-1}L') : v^{-1}L' \in LW(L')\}} \\ \\
M(\text{LD}(L)) = \{\text{dr}_L(v^{-1}L') : v^{-1}L' \in LW(L')\}.
\]
4. \(S \subseteq \text{dr}_L(v^{-1}L') \iff v^r \notin S\), for each \(S \in \text{LD}(L)\).
5. The canonical bijection \(\tau_{\text{LD}(L)} : J(\text{LD}(L)) \to M(\text{LD}(L))\) has action:
\[
\tau_{\text{LD}(L)}(\text{dr}_L(v^{-1}L')) := \text{dr}_L(v^{-1}L') \quad \text{see Note 2.2.9.}
\]

**Proof.**
1. Consider \(\theta := \lambda X.\overline{X} : (\text{LD}(L))^\text{op} \to \text{LD}(L)\). It is a well-defined bijection by the set-theoretic De Morgan laws and \(v^{-1}L = u^{-1}L\). It is an order-isomorphism because \(X \subseteq Y \iff \overline{Y} \subseteq \overline{X}\), hence a join-semilattice isomorphism too.
2. We calculate:
\[
\text{dr}_L(v^{-1}L') = \text{dr}_L(v^{-1}L') \\
= \cap\{X \in LW(L') : v^r \notin X\} \\
= \cap\{\overline{x} : v^r \in X \in LW(L')\} \\
= \cap\{X \in LW(L') : v^r \in X\}.
\]
3. We first show $M(\mathbb{L}(L))$ has the claimed description. By Corollary 3.3.8 each $X \in \mathbb{L}(L)$ is an intersection of $dr_L(v^{-1}L^r)$’s, so every $S \in \mathbb{L}(L)$ is an intersection of them too. Then these elements meet-generate $\mathbb{L}(L)$. To see they are all meet-irreducible, fix $v_0 \in \Sigma^*$. We’ll show $dr_L(v_0^{-1}L^r)$ has the following unique cover in $\mathbb{L}(L)$:

$$K_{v_0} := \bigcap\{dr_L(Y) : dr_L(v_0^{-1}L^r) \subseteq dr_L(Y), Y \in \mathbb{L}(L)\}.$$ 

Certainly $dr_L(v_0^{-1}L^r) \subseteq K_{v_0}$. Crucially if $dr_L(v_0^{-1}L^r) \subseteq dr_L(Y)$ then by strictness we know $dr_L(Y) \subseteq dr_L(v_0^{-1}L^r)$, hence $v_0 \in dr_L(Y)$ by Corollary 3.3.8. Then we have the strict inclusion $dr_L(v_0^{-1}L^r) \subseteq K_{v_0}$. Since $K_v$ is the meet of all meet-irreducibles strictly greater than $dr_L(v^{-1}L^r)$ it is also the unique cover of the latter.

The description of $J(\mathbb{L}(L))$ follows by (1) i.e. they are the relative complements of the meet-irreducibles in $\mathbb{L}(L)$. Finally, both sets have cardinality $|\mathbb{L}(L^r)|$.

4. For any $v_0 \in \Sigma^*$ we first establish:

$$\bigcap\{X \in \mathbb{L}(L^r) : v_0 \in X\} \subseteq dr_L(v^{-1}L^r) \iff \forall X \in \mathbb{L}(L).[v_0 \in X \implies X \subseteq dr_L(v^{-1}L^r)] \quad (A)$$

$$\iff \forall X \in \mathbb{L}(L).[v_0 \in X \implies v^r \in X] \quad \text{(Corollary 3.3.8)}$$

$$\iff v^r \in \bigcap\{X \in \mathbb{L}(L^r) : v_0 \in X\}.$$ 

Concerning (A), the implication ($\implies$) follows because $X \subseteq dr_L(v^{-1}L^r)$ would yield a contradiction, whereas ($\iff$) holds because $X \not\subseteq dr_L(v^{-1}L^r)$ implies $v^r \not\in X$ by Corollary 3.3.8, so the intersection contains $v^r$ too. Then, invoking (2) and (3), we’ve established the original claim whenever $S \in J(\mathbb{L}(L))$. In the general case $S = \bigcup J$ where $J \subseteq J(\mathbb{L}(L))$,

$$S \subseteq dr_L(v^{-1}L^r) \iff \forall K \in J. K \subseteq dr_L(v^{-1}L^r)$$

$$\iff \forall K \in J. v^r \not\in K$$

$$\iff v^r \not\in S.$$ 

5. Each join-irreducible takes the form $j_v := \overline{dr_L(v^{-1}L^r)}$ where $v \in \Sigma^*$. By Note 2.2.9 we know $j_v$ is join-prime, so for any $J \subseteq J(\mathbb{L}(L))$ we have $j_v \not\subseteq \bigcup J \iff \forall j \in J. j_v \not\subseteq j$. Then we calculate:

$$\tau_{\mathbb{L}(L)}(j_v) = \bigcup\{S \in \mathbb{L}(L) : j_v \not\subseteq S\}$$

$$= \bigcup\{j_v \in J(\mathbb{L}(L)) : j_v \not\subseteq j_v\} \quad (j_v \text{ join-prime})$$

$$= \bigcup\{j_v : \overline{dr_L(v^{-1}L^r)} \not\subseteq \overline{dr_L(v_0^{-1}L^r)}\} \quad (\text{Corollary 3.3.8})$$

$$= \bigcup\{j : j \subseteq J(\mathbb{L}(L)) : v_0 \not\subseteq j\}$$

$$= dr_L(v_0^{-1}L^r) \quad \text{(by (4)).}$$

Lemma 3.5.9 provides a natural bijection $\mathbb{L}(L^r) \cong J(\mathbb{L}(L))$ akin to the quotient-atom bijection.

**Theorem 3.5.10** (Quotient-intersection bijection). Each regular $L$ has the canonical bijection,

$$\lambda_L : \mathbb{L}(L^r) \to J(\mathbb{L}(L)) \quad \lambda_L(Y) := \overline{dr_L(Y)} \quad \lambda_L^{-1} := \lambda_L,$$

and respective relationship:

$$\lambda_L(x^{-1}L^r) \subseteq a^{-1} \lambda_L(y^{-1}L^r) \iff y^{-1}L^r \subseteq (xa)^{-1}L^r \quad \text{for any } x, y \in \Sigma^*.$$ 

**Proof.** The bijection follows by Lemma 3.5.9. Concerning the relationship,

$$y^{-1}L^r \subseteq (xa)^{-1}L^r \iff \forall v \in \Sigma^*. [yv \in L^r \implies xav \in L^r]$$

$$\iff \forall v \in \Sigma^*. [v^ry \in L \implies v^r ax^r \in L]$$

$$\iff \forall v \in \Sigma^*. [y^r \in [v^r]^{-1}L \implies x^r \in [(ya)^{-1}]L]$$

$$\iff \forall X \in \mathbb{L}(L). [y^r \in X \implies x^r \in a^{-1}X]$$

$$\iff \bigcap\{X \in \mathbb{L}(L) : x^r \in X\} \subseteq a^{-1} \bigcap\{X \in \mathbb{L}(L) : y^r \in X\}.$$
Note 3.5.11 (Canonicity of $\lambda_L$). It arises from the duality between Poset-dfas and DL-dfas, see [MAMU14].

Recall the nfa $\text{dfa}_1(L)$ from Example 3.3.4. It arises from the state-minimal deterministic machine $\text{dfa}(L)$ by extending the initial states and transitions.

Theorem 3.5.12 (Canonical distributive dependency automaton). We have the $\text{aut}_{\text{Dep}}$-isomorphism:

\[ D: (\text{rev}(\text{dfa}_1(L')), \leq, \text{dfa}_1(L')) \to \text{Airr}(\text{dfa}_1(L)) \quad D(v_1^{-1}L', dr_L(v_2^{-1}L')) : \iff v_1^{-1}L' \subseteq v_2^{-1}L' \]

with inverse $E(S, v^{-1}L') : \iff \lambda_L^{-1}(S) \subseteq v^{-1}L'$.

Proof. To see $D$’s domain is a well-defined dependency automaton, observe that its dual $(\text{dfa}_1(L'), \geq, \text{rev}(\text{dfa}_1(L')))$ is well-defined by Theorem 3.2.18. Next we establish the commuting relations:

\[
\begin{align*}
\text{LW}(L') & \xrightarrow{\tau_{LD}(L)} M(\text{LD}(L)) \\
\text{LW}(L') & \xrightarrow{\lambda_L} J(\text{LD}(L))
\end{align*}
\]

via the following calculation:

\[
\begin{align*}
\lambda_L(v_1^{-1}L') \notin dr_L(v_2^{-1}L') & \iff dr_L(v_1^{-1}L') \notin dr_L(v_2^{-1}L') \quad \text{(def. of $\lambda$)} \\
\iff v_2' \notin dr_L(v_1^{-1}L') \quad \text{(by Lemma 3.5.9.4)} \\
\iff v_2' \notin dr_L(v_1^{-1}L') \quad \text{(by Lemma 3.5.9.4)} \\
\iff v_2' \notin dr_L(v_1^{-1}L') \\
\iff v_2' \notin dr_L(v_1^{-1}L') \\
\iff v_1^{-1}L' \subseteq v_2^{-1}L' \\
\iff v_1^{-1}L' \subseteq v_2^{-1}L'.
\end{align*}
\]

Since the witnesses are bijections we’ve established that $D$ underlying $\text{Dep}$-morphism is an isomorphism. Concerning the remaining conditions, $D$’s domain has lower nfa $\mathcal{N}_a := \text{rev}(\text{dfa}_1(L'))$ with transitions $\mathcal{N}_a(Y_1, Y_2) : \iff Y_1 \subseteq a^{-1}Y_2$.

Furthermore $\text{Airr}(\text{dfa}_1(L'))$’s upper nfa $\mathcal{M}$ has transitions:

\[
\begin{align*}
\mathcal{M}_a(dr_L(v_1^{-1}L'), dr_L(v_2^{-1}L')) & \iff (\gamma_a)_*(dr_L(v_1^{-1}L')) \subseteq dr_L(v_2^{-1}L') \\
& \iff \bigcup \{ j \in J(\text{LD}(L)) : a^{-1}j \in dr_L(v_1^{-1}L') \} \subseteq dr_L(v_2^{-1}L') \\
& \iff \bigcup \{ j \in J(\text{LD}(L)) : a^{-1}j \in dr_L(v_2^{-1}L') \} \subseteq dr_L(v_2^{-1}L') \quad \text{(by Lemma 3.5.9.4)} \\
& \iff dr_L((v_1a)^{-1}L') \subseteq dr_L(v_2^{-1}L') \quad \text{(by Lemma 3.5.9.4)} \\
& \iff v_2^{-1}L' \subseteq (v_1a)^{-1}L' \\
& \iff v_2^{-1}L' \subseteq (v_1a)^{-1}L' \quad \text{(dr$_L$ an order-iso)}
\end{align*}
\]

where $\gamma_a := \lambda X. a^{-1}X : \text{LD}(L) \to \text{LD}(L)$. Then we verify:

\[
\begin{align*}
\mathcal{N}_a; D(v_1^{-1}L', dr_L(v_2^{-1}L')) & \iff \exists v \in \Sigma*. [v_1^{-1}L' \subseteq (va)^{-1}L' \land D(v_1^{-1}L', dr_L(v_2^{-1}L'))] \\
& \iff \exists v \in \Sigma*. [v_1^{-1}L' \subseteq (va)^{-1}L' \land v_1^{-1}L' \subseteq v_2^{-1}L'] \\
& \iff \exists v \in \Sigma*. [v_1^{-1}L' \subseteq v_1^{-1}L' \land v_1^{-1}L' \subseteq (v_2a)^{-1}L'] \\
& \iff \exists v \in \Sigma*. [D(v_1^{-1}L', dr_L(v_2^{-1}L')) \land \mathcal{M}_a(dr_L(v_2^{-1}L'), dr_L(v_2^{-1}L'))] \\
& \iff D; M_a(v_1^{-1}L', dr_L(v_2^{-1}L')).
\end{align*}
\]

Concerning (A), $(\Rightarrow)$ follows because $a^{-1}(-)$ preserves inclusions so we can choose $v := v_1$; $(\Leftarrow)$ follows analogously,
choosing \( v := v_2 \). Finally we verify:

\[
\begin{align*}
\mathcal{D}[I_N] &= \mathcal{D}[\{Y : \varepsilon \in Y \in \mathcal{LW}(L')\}] \\
&= \{dr_L(Y) : \varepsilon \in Y \in \mathcal{LW}(L')\} \\
&= \{dr_L(v^{-1}L') : v \in L'\} \\
&= \{dr_L(v^{-1}L') : L \nsubseteq dr_L(v^{-1}L')\} \\
&= F_M \quad \text{(by Corollary 3.3.8)}
\end{align*}
\]

\[
\begin{align*}
\mathcal{D}[I_M] &= \mathcal{D}[\{dr_L(v^{-1}L') : dr_L(L') \subseteq dr_L(v^{-1}L')\}] \\
&= \mathcal{D}[\{dr_L(v^{-1}L') : v^{-1}L' \subseteq L'\}] \\
&= \{v^{-1}L' : \exists v \in \Sigma^*, [\varepsilon v^{-1}L' \subseteq v^{-1}L' \land v^{-1}L' \subseteq L']\} \\
&= \{v^{-1}L' : v^{-1}L' \subseteq L'\} \\
&= F_N. \quad \text{(see Definition 3.2.5)}
\end{align*}
\]

\[\square\]

**Corollary 3.5.13** (Dualising \( \mathfrak{d} \mathfrak{a}_\lambda(L) \)). We have the JSL-dfa isomorphism,

\[
\varrho_L : \text{Det}(\text{dfa}_\lambda(L'), \geq, \text{rev}(\text{dfa}_\lambda(L'))) \rightarrow (\mathfrak{d} \mathfrak{a}_\lambda(L))^* \\
\varrho_L := \lambda S \cap \{dr_L(Y) : Y \in S\} \quad \varrho_L^1 := \lambda K \{Y \in \mathcal{LW}(L') : K \subseteq dr_L(Y)\}.
\]

**Proof.** First recall the isomorphism from Theorem 3.5.12,

\[
\mathcal{E} : \text{Airr}(\mathfrak{d} \mathfrak{a}_\lambda(L)) \rightarrow \text{Rev}\mathfrak{M} \quad \text{where} \quad \mathfrak{M} := (\text{dfa}_\lambda(L'), \geq, \text{rev}(\text{dfa}_\lambda(L'))).
\]

Since \( \mathcal{E} \) has bijective upper witness \((\tau_{\mathcal{LW}(L)} \circ \lambda_L)^{-1}\) and join-semilattice adjoints act as the inverse, it follows that \((\text{Det}\mathcal{E})^* = \lambda X. \tau_{\mathcal{LW}(L)} \circ \lambda_L[X]\). Further recall the natural isomorphism \( \delta \). (Theorem 3.2.13) and \( \text{rep}_\lambda \) (Theorem 3.2.6). Then we have the composite join-semilattice isomorphism:

\[
\begin{align*}
\text{Det}\mathfrak{M} & \xrightarrow{\delta_{\mathfrak{M}}^{-1}} (\text{DetRev}\mathfrak{M})^* \xrightarrow{(\text{Det}\mathcal{E})^*} (\text{DetAirr}(\mathfrak{d} \mathfrak{a}_\lambda(L)))^* \xrightarrow{\text{rep}_{\lambda}^*(\mathfrak{d} \mathfrak{a}_\lambda(L))} (\mathfrak{d} \mathfrak{a}_\lambda(L))^*.
\end{align*}
\]

Given any subset \( S \subseteq \mathcal{LW}(L') \) upwards-closed w.r.t. inclusion,

\[
\begin{align*}
S & \rightarrow \delta_{\mathfrak{M}}^{-1}(S) \\
& \supseteq \{S\} \\
& = \mathcal{S} \quad \text{(} \mathcal{S} \text{ down-closed)} \\
& \rightarrow (\text{Det}\mathcal{E})^*(\mathcal{S}) \\
& = \tau_{\mathcal{LW}(L)} \circ \lambda_L[\mathcal{S}] \\
& = \{dr_L(v^{-1}L') : v^{-1}L' \not\in S\} \\
& \rightarrow \text{rep}_{\lambda}^*(\mathfrak{d} \mathfrak{a}_\lambda(L))(\{dr_L(v^{-1}L') : v^{-1}L' \not\in S\}) \\
& = \{dr_L(v^{-1}L') : v^{-1}L' \in S\}.
\end{align*}
\]

Finally the action of the inverse follows by the bijectivity of \( \varrho_L \). \[\square\]

### 3.6 Minimal boolean syntactic machine

We start by recalling the syntactic monoid of a regular language and the transition monoid of a classical dfa.

**Definition 3.6.1** (Transition monoids and syntactic monoids).

1. Given any set \( \Sigma \) we have the free \( \Sigma \)-generated monoid \( \Sigma^* := (\Sigma^*, \cdot, \varepsilon) \) where multiplication is concatenation.

2. Given a dfa \( \delta = (Z_0, Z, \delta_a, F) \), its transition monoid is defined \( \text{TM}(\delta) := (\{\delta_w : w \in \Sigma^*\}, \circ, id_Z) \) where \( \circ \) is functional composition and \( \delta_e = id_Z \) (see Definition 3.1.3). It admits a natural dfa structure accepting \( L \):

\[
\text{dfa}_{\text{TM}(\delta)} := (id_Z, \{\delta_w : w \in \Sigma^*\}, \lambda f, \delta_a \circ f, \{f : f(z_0) \in F\}).
\]

Finally we have \([\cdot]_{\text{TM}(\delta)} : \Sigma^* \rightarrow \text{TM}(\delta)\) where \([w]_{\text{TM}(\delta)} := \delta_w\).
3. The syntactic monoid of a regular language $L \subseteq \Sigma^*$ is the quotient $\text{Syn}(L) := \Sigma^*/S_L$ by the syntactic congruence $S_L := \{(u, v) \in \Sigma^* \times \Sigma^* : \forall x, y \in \Sigma^*. [x u y \in L \iff x v y \in L]\}$. It admits a natural dfa structure accepting $L$:

$$\text{dfa}_\text{Syn}(L) := ([\varepsilon]_{S_L}, \Sigma^*/S_L, \lambda x. x \cdot [a]_{S_L}, \{[u]_{S_L} : u \in L\}).$$

We also denote the underlying set by $\text{Syn}(L) := \Sigma^* \setminus S_L$.

Lemma 3.6.2 (The syntactic/transition monoid are well-defined).

1. $\text{TM}(\delta)$ is a well-defined finite monoid and $L(\text{dfa}_\text{TM}(\delta)) = L(\delta)$.

2. $\text{Syn}(L)$ is a well-defined monoid and $L(\text{dfa}_\text{Syn}(L)) = L$.

Proof.

1. Fix a dfa $\delta = (z_0, Z, \delta_0, F)$. The set of all endofunctions on a set equipped with functional composition define a finite monoid; $\text{TM}(\delta)$ defines a submonoid. Finally:

$$w \in L(\text{dfa}_\text{TM}(\delta)) \iff \delta_w \in F(\text{dfa}_\text{TM}(\delta)) \iff \delta_w(z_0) \in F \iff w \in L(\delta).$$

2. To see $S_L \subseteq \Sigma^* \times \Sigma^*$ is a congruence for $(\Sigma^*, \cdot, \varepsilon)$, given $S_L(u_1, u_2)$ and $S_L(v_1, v_2)$,

$$x(u_1, v_1) y \in L \iff x(u_1) v_1 y \in L \iff x(u_2) v_1 y \in L \iff x u_2(v_1) y \in L \iff x u_2 v_2 y \in L.$$Thus $\text{Syn}(L)$ is a well-defined monoid. It is finite because the equivalence classes are precisely the atoms of the set-theoretic boolean algebra generated by the finite set $\{x^{-1}y^{-1} : x, y \in \Sigma^*\}$. Finally:

$$w \in L(\text{dfa}_\text{Syn}(L)) \iff [w]_{S_L} \in F(\text{dfa}_\text{Syn}(L)) \iff w \in L.$$The final equivalence follows because if $w \in L$ then $[w]_{S_L} \subseteq L$. Indeed if $u \in [w]_{S_L}$ then choosing $x = y = \varepsilon$ we have $x y \in L \iff x y \in L$ i.e. $u \in L$.

As is well-known, $L$’s syntactic monoid is isomorphic to the transition monoid of $L$’s state-minimal dfa.

Theorem 3.6.3 ($\text{Syn}(L) \cong \text{TM}(\text{dfa}(L))$). We have the monoid isomorphism:

$$\lambda [w]_{S_L} X. w^{-1} X : \text{Syn}(L) \rightarrow \text{TM}(\text{dfa}(L)).$$

Proof. The function is well-defined and injective because:

$$[u_1]_{S_L} = [u_2]_{S_L} \iff \forall x, y \in \Sigma^*. [x u_1 y \in L \iff x u_2 y \in L] \iff \forall x \in \text{LW}(L), y \in \Sigma^*. [x u_1 y \in X \iff x u_2 y \in X] \iff \forall x \in \text{LW}(L), y \in \Sigma^*. [x u_1 y \in X \iff x u_2 y \in X] \iff \forall x \in \text{LW}(L), u_1^{-1} X = u_2^{-1} X \iff \lambda X. x \in \text{LW}(L), u_1^{-1} X = \lambda X \in \text{LW}(L), u_2^{-1} X.$$It is surjective because $\text{dfa}(L)$’s transition monoid consists of the functions $\{\lambda X. x \in \text{LW}(L), w^{-1} X : w \in \Sigma^*\}$. Finally it is a monoid morphism because $\lambda X. x^{-1} X = id_{\text{LW}(L)}$ and $(uv)^{-1} X = v^{-1}(u^{-1} X)$.

We can now introduce another canonical JSl-dfa and its equivalent dependency automaton.

Definition 3.6.4 (L’s minimal boolean syntactic JSL-dfa). Let $\text{LRW}(L) := \{u^{-1} L v^{-1} : u, v \in \Sigma^*\}$ be the left-right-word-quotients and $\text{LRP}(L)$ the closure of $\text{LRW}(L)$ under the set-theoretic boolean operations. Then:

$$\text{dfa}_\text{Syn}(L) := (L, \text{LRP}(L), \lambda X. a^{-1} X, \{K : \varepsilon \in K\})$$
is the canonical boolean syntactic JSL-dfa over the join-semilattice $\text{LRP}(L) := (\text{LRP}(L), \cup, \emptyset)$.

Lemma 3.6.5 ($J(\text{LRP}(L)) = \text{Syn}(L)$). $\text{LRP}(L)$’s atoms are the equivalence classes of the syntactic congruence $S_L$.

Proof. An equivalence class amounts to $\cap_i u_i^{-1} L v_i^{-1} \cap \cap_j u_j^{-1} L v_j^{-1}$ involving every left-right-word-quotient $u^{-1} L v^{-1}$. 

Next we describe the minimal boolean syntactic JSL-dfa as a dependency automaton.

**Theorem 3.6.6 (Canonical boolean syntactic dependency automaton).** We have the \( \text{aut}_{\text{Dep}} \)-isomorphism:

\[
\lambda X.\overline{X} : \text{dep}(\text{rev}(\text{dfa}_{\text{syn}}(L'))) \to \text{Airr}(\text{dfa}_{\text{syn}}(L)),
\]
whose inverse has action \( \lambda X.X' \).

**Proof.** We have the bijection \( \lambda X.X' : \text{Syn}(L') \to \text{Syn}(L) \) because \( \forall x, y \in \Sigma^* \cdot [xuv \in L \iff xuvy \in L'] \) is equivalent to \( \forall x, y \in \Sigma^* \cdot [xuv' \in L' \iff xuv'y \in L'] \). Then we have the \( \text{Dep} \)-isomorphism \( f := \lambda X.\overline{X} \),

\[
\begin{align*}
\text{Syn}(L') & \xrightarrow{\Delta_{\text{Syn}(L')}} \overline{X} : X \in \text{Syn}(L) \\
\text{Syn}(L') & \xrightarrow{\lambda X.X'} \text{Syn}(L)
\end{align*}
\]

where \( \neg_{\text{LFS}(L)} \) constructs the relative complement in \( \Sigma^* \). It is a \( \text{Dep} \)-isomorphism because the witnesses are bijections. It remains to verify the other constraints. Denote the transitions of the left (resp. right) dependency automaton’s lower (resp. upper) nfa by \( \mathcal{N} \) (resp. \( \mathcal{M} \)). Then:

\[
\begin{align*}
\mathcal{N}_a([u_1]_{S_{L}}, [u_2]_{S_{L}}) & \iff [u_2a]_{S_{L}} = [u_1]_{S_{L}} \\
\mathcal{M}_a([u_1]_{S_{L}}, [u_2]_{S_{L}}) & \iff ([u_1]_{S_{L}}) \cup ([u_2]_{S_{L}}) \in [[u_1 u_2]_{S_{L}}] \\
& \iff ([u_1]_{S_{L}}) : a^{-1} [u_1]_{S_{L}} \subseteq [u_1]_{S_{L}} \\
& \iff ([u_1]_{S_{L}}) : a^{-1} [u_1]_{S_{L}} \subseteq [u_2]_{S_{L}} \\
& \iff ([u_1]_{S_{L}}) : a^{-1} [u_1]_{S_{L}} \subseteq [u_2]_{S_{L}} \\
& \iff [u_1]_{S_{L}} \subseteq [u_2]_{S_{L}} \\
& \iff [u_2]_{S_{L}} \subseteq [u_1]_{S_{L}}.
\end{align*}
\]

where \( \gamma_a := \lambda X.a^{-1}X : \text{dfa}_{\text{syn}}(L) \to \text{dfa}_{\text{syn}}(L). \) We now verify the condition concerning transitions:

\[
\begin{align*}
\mathcal{N}_a; f([u_1]_{S_{L}}, [u_2]_{S_{L}}) & \iff \exists u \in \Sigma^* \cdot [u,a]_{S_{L}} = [u_1]_{S_{L}} \land [u_2]_{S_{L}} = [u_1]_{S_{L}} \\
& \iff \exists u \in \Sigma^* \cdot [u,a]_{S_{L}} = [u_1]_{S_{L}} \land [u]_{S_{L}} = [u_2]_{S_{L}} \\
& \iff [u]_{S_{L}} = [u_1]_{S_{L}} \\
& \iff \exists u \in \Sigma^* \cdot [u]_{S_{L}} = [u_1]_{S_{L}} \land [u]_{S_{L}} = [a u]_{S_{L}} \\
& \iff \exists u \in \Sigma^* \cdot [f([u]_{S_{L}})] = [u]_{S_{L}} \land \mathcal{M}_a([u]_{S_{L}}, [u]_{S_{L}}) \\
& \iff f; \mathcal{M}_a([u_1]_{S_{L}}, [u_2]_{S_{L}}).
\end{align*}
\]

Finally we calculate:

\[
\begin{align*}
F_{\text{rev}(\text{dfa}_{\text{syn}}(L'))} & = f(F_{\text{dfa}_{\text{syn}}(L')}) \\
& = f(F_{\text{rev}(\text{dfa}_{\text{syn}}(L'))}) \\
& = F_{\text{M}_a} \\
\end{align*}
\]

\[
\begin{align*}
\tilde{f} \{ [u]_{S_{L}} : \varepsilon \notin L \} & = \tilde{f} \{ [u]_{S_{L}} : \varepsilon \notin L \} \\
& = \{ [\varepsilon]_{S_{L}} : \varepsilon \notin L \} \\
& = F_{\mathcal{N}}.
\end{align*}
\]

\[\Box\]

**Note 3.6.7 (Canonical distributive syntactic JSL-dfa).**

\[
\mathcal{R} : (\text{rev}(\text{dfa}_{\text{syn}}(L')), \mathcal{S}_L, (\text{dfa}_{\text{syn}}(L'))) \to \text{Airr}(\text{dfa}_{\text{syn}}(L))
\]

\[\blacksquare\]
3.7 Transition semirings of JSL-dfas

Whilst classical dfas induce monoids, JSL-dfas induce idempotent semirings.

Definition 3.7.1 (Transition semiring of a JSL-dfa).

1. \( \mathcal{P}_f \Sigma^* := (\mathcal{P}_f \Sigma, \cup, \emptyset, \{ \varepsilon \}) \) is the free \( \Sigma \)-generated idempotent semiring where \( \mathcal{P}_f \Sigma^* \) is the set of finite languages, its multiplication being sequential composition of languages.

2. Fix a JSL-dfa \( \gamma = (s_0, S, \gamma_a, F) \) and recall the composites \( \gamma_w : S \rightarrow S \) for \( w \in \Sigma^* \) from Definition 3.2.1. More generally for any \( K \subseteq \Sigma^* \) we can construct the pointwise-join of \( \{ \gamma_w : w \in K \} \),

\[
\gamma_K := \lambda s. \bigvee_{s} \{ \gamma_w(s) : w \in K \} : S \rightarrow S.
\]

Then \( \gamma \)'s transition semiring is the idempotent semiring \( TS(\gamma) := (S_\gamma, \circ, id_{S_\gamma}) \) where:

\[
S_\gamma := (S_\gamma, \vee_{S_\gamma}, \lambda X. \downarrow_S) \quad S_\gamma := \{ \gamma_K : K \subseteq \Sigma^* \} \quad \gamma U \vee_{S_\gamma} \gamma V := \gamma U \cup V.
\]

3. Since \( TS(\gamma) \) is \( \Sigma \)-generated by \( \{ \gamma_a : a \in \Sigma \} \) we have the unique extension \([-\]\( TS(\gamma) : \mathcal{P}_f \Sigma^* \rightarrow TS(\gamma) \) i.e. a surjective idempotent semiring morphism.

4. Finally the semiring \( TS(\gamma) \) has a natural associated JSL-dfa structure:

\[
ts_\gamma(\gamma) := (id_{S_\gamma}, S_\gamma, \lambda f. \gamma_a \circ f, \{ \gamma : f \not\in S_\gamma, \gamma_T \})
\]

\[\text{accepting } L := L(\gamma).\]

Lemma 3.7.2 (\( TS(\gamma) \) and \( ts(\gamma) \) well-defined).

1. \( TS(\gamma) \) is a well-defined idempotent semiring.

2. \( ts(\gamma) \) is a JSL-reachable JSL-dfa accepting \( L(\gamma) \).

Proof. Let \( \gamma = (s_0, S, \gamma_a, F) \) be a JSL-dfa.

1. \( S_\gamma \) defines an ‘additive’ idempotent commutative monoid; \( (S_\gamma, \circ, id_{S_\gamma}) \) defines a ‘multiplicative’ monoid. Multiplication left/right distributes over addition and \( \downarrow_{S_\gamma} \) annihilates multiplication because composition of join-semilattice morphisms is bilinear w.r.t. pointwise-joins.

2. We first establish \( ts(\gamma) \) is a well-defined JSL-dfa. The transition endomorphisms are well-defined functions, and preserve the join by bilinearity. The final states are well-defined by construction since \( \gamma_T \in S_\gamma \). This JSL-dfa accepts \( L \) because \( \gamma_w \downarrow_{S_\gamma}, \gamma_T \implies w \in L \), as we now show.

- \( (\Rightarrow) \): contrapositive follows because if \( w \in T \) then \( \gamma_T \) is a join of morphisms including \( \gamma_w \).

- \( (\Leftarrow) \): \( w \in L \) implies \( \gamma_w(s_0) \in F \) whereas \( \gamma_T(s_0) \notin F \).

Finally it is JSL-reachable because (i) each \( \gamma_w \) is classically reachable from the identity function \( id_{S_\gamma} \), (ii) each \( \gamma_K \) is the join of \( \gamma_w \)'s.

\[\square\]

Lemma 3.7.3. dfa\( \downarrow_\text{TM}(\text{reach}(N)) \cong \text{reach}(\text{ts}(\text{reach}(\text{sc}(N)))) \) for any nfa \( N \).

Proof. Let \( S := \text{reach}(\text{sc}(N)) \) i.e. the closure of the reachable subsets \( \text{rs}(N) \) under unions. Then we need to establish the dfa isomorphism \( \lambda_{\gamma_w.\delta_w} : \gamma \rightarrow \delta \) where:

\[
\gamma = (id_{\text{rs}(N)}, \{ \gamma_w : \text{rs}(N) \rightarrow \text{rs}(N), w \in \Sigma^* \}, \lambda f. \gamma_a \circ f, \{ f : f(I) \cap F \neq \emptyset \})
\]

\[
\delta = (id_{S}, \{ \delta_w : S \rightarrow S, w \in \Sigma^* \}, \lambda f. \delta_a \circ f, \{ f : f \notin \delta_T \})
\]

42
and both $\gamma_w$ and $\delta_w$ have action $\lambda X.\mathcal{N}_w[X]$. The candidate isomorphism is a well-defined bijection because $\delta_w$ is uniquely determined by the domain-codomain restriction $\gamma_w$. It clearly preserves the initial state and preserves/reflects the transitions. Finally,

$$\delta_w \neq \delta_w^L \iff \exists u \in \Sigma^* \cdot \delta_w(N_u[I]) \neq \delta_w^L(N_u[I])$$

$$\iff \exists u \in \Sigma^*, z \in \mathcal{N}_w(z \in \mathcal{N}_w[N_u[I]] \land z \notin \mathcal{N}_w^L[N_u[I]])$$

$$\iff w \in L$$

$$\iff \gamma_w(I) \cap F \neq \emptyset.$$  \hfill (A)

Concerning (A), (\Rightarrow) is immediate whereas (\Leftarrow) follows by choosing $u := \varepsilon$. \hfill $\square$

**Definition 3.7.4** (Power semiring and syntactic semiring).

1. The finitary power semiring of a monoid $M := (M, \cdot, 1_M)$ is the idempotent semiring:

$$\mathcal{P}_f M := ((\mathcal{P}_f M, \cup, \emptyset), \cdot, \{1_M\}) \quad S_1 \cdot S_2 := \{m_1 \cdot m_2 : m_1 \in S_1, m_2 \in S_2\}$$

where $\mathcal{P}_f M$ is the set of finite subsets of $M$. If $M$ is a finite monoid we may instead write $\mathcal{P}M$.

2. Given any set $\Sigma$ then $\mathcal{P}_f \Sigma^*$ is the free $\Sigma$-generated idempotent semiring.

3. The syntactic semiring $\text{Syn}_v(\Sigma) := \mathcal{P}_f \Sigma^*/\mathcal{S}_L^\Sigma_v$ of a regular language $L \subseteq \Sigma^*$ is the quotient of the free $\Sigma$-generated idempotent semiring by $L$’s syntactic semiring congruence $\mathcal{S}_L^\Sigma_v \subseteq \mathcal{P}_f \Sigma^* \times \mathcal{P}_f \Sigma^*$ [Pol01]:

$$\mathcal{S}_L^\Sigma_v(U, V) := \iff \forall x, y \in \Sigma^*.[\{x\} \cdot U \cdot \{y\} \subseteq \overline{L} \iff \{x\} \cdot V \cdot \{y\} \subseteq \overline{L}].$$

It admits a natural JSL- DFA structure accepting $L$,

$$\text{Syn}(L) := ([\{\varepsilon\}]_{\mathcal{S}_L^\Sigma_v}, (\mathcal{P}_f \Sigma^*/\mathcal{S}_L^\Sigma_v, \cdot, \text{Syn}_v(\Sigma), \emptyset, \mathcal{S}_L^\Sigma_v), \lambda X.\mathcal{N}_w[\text{Syn}_v(\Sigma), \emptyset, \mathcal{S}_L^\Sigma_v, \{[a]\}, [U]]_{\mathcal{S}_L^\Sigma_v} : U \cap L \neq \emptyset).$$

\hfill $\blacksquare$

**Lemma 3.7.5** (Power/syntactic semirings are well-defined).

1. $\mathcal{P}_f M$ is a well-defined idempotent semiring.

2. $S_L(u, v) \iff S_L(\{u\}, \{v\})$ for all $u, v \in \Sigma^*$.

3. $\text{Syn}_v(\Sigma)$ is a well-defined finite idempotent semiring.

4. $\text{Syn}(L)$ is a well-defined JSL-DFA accepting $L$.

**Proof.**

1. Let $M = (M, \cdot, 1_M)$ be a monoid. Firstly, $(\mathcal{P}_f M, \cup, \emptyset)$ is the free join-semilattice on $M$. Secondly, the multiplication $\cdot$ is respectively bilinear by construction.

2. We calculate:

$$S_L^\Sigma_v(\{u\}, \{v\}) \iff \forall x, y \in \Sigma^*.[\{x\} \cdot \{u\} \cdot \{y\} \subseteq \overline{L} \iff \{x\} \cdot \{v\} \cdot \{y\} \subseteq \overline{L}]$$

$$\iff \forall x, y \in \Sigma^*.[xuv \in \overline{L} \iff xuv \in \overline{L}]$$

$$\iff \forall x, y \in \Sigma^*.[xuv \in \overline{L} \iff xuv \in \overline{L}]$$

$$\iff S_L(u, v).$$

3. We’ll show $S_L^\Sigma_v$ is a congruence for the free idempotent semiring $\mathcal{P}_f \Sigma^*$. First observe:

$$S_L^\Sigma_v(U, V) \iff \forall X, Y \in \mathcal{P}_f \Sigma^*.[X \cdot U \cdot Y \subseteq \overline{L} \iff X \cdot V \cdot Y \subseteq \overline{L}]. \quad (*)$$
Concerning final states, Proof. It is well-defined and injective because:

1. Firstly, defined. First observe if are well-defined because multiplication in with its dfa structure.
2. Indeed:

\[ \{ x \} \cdot (U_1 \cup U_2) \cdot \{ y \} \subseteq \overline{L} \iff \forall i \in \{1,2\}, \{ x \} \cdot U_i \cdot \{ y \} \subseteq \overline{L} \]
\[ \iff \forall i \in \{1,2\}, \{ x \} \cdot V_i \cdot \{ y \} \subseteq \overline{L} \]
\[ \iff \{ x \} \cdot (V_1 \cup V_2) \cdot \{ y \} \subseteq \overline{L} \]

3. \[ \{ x \} \cdot (U_1 \cdot U_2) \cdot \{ y \} \subseteq \overline{L} \iff \{ x \} \cdot U_1 \cdot (U_2 \cdot \{ y \}) \subseteq \overline{L} \]
\[ \iff \{ x \} \cdot V_1 \cdot (U_2 \cdot \{ y \}) \subseteq \overline{L} \] (via *)
\[ \iff \{ \{ x \} \cdot V_1 \} \cdot U_2 \cdot \{ y \} \subseteq \overline{L} \] (via *)
\[ \iff \{ x \} \cdot (V_1 \cdot V_2) \cdot \{ y \} \subseteq \overline{T}. \]

To see \( \text{Syn}_\lambda(L) \) is finite, recall the syntactic monoid is finite by Lemma 3.6.2 and consider the mapping:

\[ q := \lambda[u]_{S_L} : u \in U \in \mathcal{P}_f \Sigma^*. \{U\}_{S_L} : \mathcal{P}_{\text{Syn}}(L) \rightarrow \text{Syn}_\lambda(L). \]

Well-definedness follows via (2) and it is clearly surjective, hence \( \text{Syn}_\lambda(L) \) is finite.

4. We show \( \text{syn}(L) \) is a well-defined JSL-dfa. It is finite because the syntactic semiring is finite – see (3). Its join-semilattice structure is well-defined because \( S_L^\lambda \) is a well-defined congruence. Its deterministic transitions are well-defined because multiplication in \( \text{Syn}_\lambda(L) \) is bilinear. It remains to show the final states are well-defined. First observe if \([U]_{S_L^\lambda} = [V]_{S_L^\lambda} \) and \( U \notin \overline{L} \) then \( V \notin \overline{L} \) by choosing \( x = y = \varepsilon \). Secondly, the non-finals \( \{[V]_{S_L^\lambda} : U \in \overline{L} \} \) are closed under joins because given (finitely many) \( U_i \in \overline{L} \) then \( \bigcup U_i \in \overline{L} \) too. This well-defined JSL-dfa accepts \( L \) because its classically reachable part is isomorphic to the syntactic monoid \( \text{Syn}(L) \) endowed with its dfa structure.

\[ \square \]

Analogous to Theorem 3.6.3, \( \text{dfa}(L) \)’s transition semiring is isomorphic to \( L \)’s syntactic semiring.

**Theorem 3.7.6 (\( \text{Syn}_\lambda(L) \equiv \text{TS}(\text{dfa}(L)) \)).** We have the idempotent semiring isomorphism:

\[ \alpha := \lambda[U]_{S_L^\lambda} : \lambda X \cdot U^{-1} X : \text{Syn}_\lambda(L) \rightarrow \text{TS}(\text{dfa}(L)). \]

It also defines a JSL-dfa isomorphism \( \text{syn}(L) \rightarrow \text{ts}(\text{dfa}(L)) \).

**Proof.** It is well-defined and injective because:

\[ [U_1]_{S_L^\lambda} = [U_2]_{S_L^\lambda} \iff \forall x, y \in \Sigma^*. [xU_1y \in \overline{L} \iff xU_2y \in \overline{L}] \]
\[ \iff \forall x, y \in \Sigma^*. [xU_1y \notin \overline{L} \iff xU_2y \notin \overline{L}] \]
\[ \iff \forall x, y \in \Sigma^*. [xU_1y \cap L \neq \emptyset \iff xU_2y \cap L \neq \emptyset] \]
\[ \iff \forall x, y \in \Sigma^*. [xU_1y \cap L \neq \emptyset \iff U_2y \cap Ux^\lambda \neq \emptyset] \]
\[ \iff \forall x \in \text{LW}(L), y \in \Sigma^*. [U_1y \cap X \neq \emptyset \iff U_2y \cap Ux^\lambda \neq \emptyset] \]
\[ \iff \forall x \in \text{LW}(L), y \in \Sigma^*. [y \in [U_1]^{-1} X \iff y \in [U_2]^{-1} X] \]
\[ \iff \forall x \in \text{LW}(L)[[U_1]^{-1} X = [U_2]^{-1} X] \]
\[ \iff \lambda X \in \text{LQ}(L), [U_1]^{-1} X = \lambda X \in \text{LQ}(L), [U_2]^{-1} X. \]

Concerning the final equivalence, \( \text{LW}(L) \) join-generates \( \text{LQ}(L) \) and each \( U^{-1}(\cdot) \) preserves unions. Next, \( \alpha \) is surjective because \( \text{dfa}(L) \)’s transition semiring consists of the endomorphisms \( \lambda X \cdot U^{-1} X \) for \( U \in \Sigma^* \), or equivalently where \( U \in \mathcal{P}_f \Sigma^* \) since \( \text{LW}(L) \) is finite. Next, \( \alpha \) is a monoid morphism because \( \lambda X \cdot \varepsilon X \) is the identity function and \( (UV)^{-1} X = V^{-1} (U^{-1} X) \). Finally it preserves the join structure because \( \varepsilon X \neq \emptyset \) and \( (U \cup V)^{-1} X = U^{-1} X \cup V^{-1} X \).

Finally we establish the claimed JSL-dfa isomorphism. The transitions follow because \((Ua)^{-1}(X) = a^{-1}(U^{-1} X)\). Concerning final states, \( \alpha \) is a join-semilattice isomorphism hence an order isomorphism, so it suffices to show \( \alpha \) preserves the largest non-final state. Then we must prove the marked equality below:

\[ \lambda X, [U]^{-1} X \updownarrow \lambda X, [\overline{L}]^{-1} X \quad \text{where} \quad [U]_{S_L^\lambda} := \bigvee_{\text{Syn}_\lambda(L)} \{ \{ u \} \}_{S_L^\gamma} : u \notin L \} . \]

Firstly, \( U \notin \overline{L} \) by well-definedness. Conversely each \( u_0 \in \overline{L} \) has some \( u \in U \) s.t. \( \{ u_0 \} \}_{S_L^\lambda} = \{ \{ u \} \}_{S_L^\gamma} \). Then by an earlier calculation we know \( \lambda X, u_0^{-1} X = \lambda X, u^{-1} X \), so the marked equality follows.

\[ \square \]
Corollary 3.7.7. \( \syn(L) \equiv ts(\dfaf(L)). \)

Proof. The join-semilattice isomorphism and transitions follows via Theorem 3.7.6. The initial state is preserved i.e. \( [(\varepsilon)]_{L}^{\delta} \rightarrow id_{\dfaf(L)}. \) Lastly the final states are preserved/reflect:

\[
\lambda X. U^{-1} X \notin \gamma_r \iff \exists w \in \Sigma^*, U^{-1}(w^{-1}L) \notin \bigcup_{x \in L} x^{-1}(w^{-1}L) \\
\iff \exists w, v \in \Sigma^*, u \in U, (wuv \in L \land wLv \cap L = \emptyset) \\
\iff U \cap L \neq \emptyset.
\]

\( \square \)

Corollary 3.7.8. \( \text{dfa}_{\syn(L)} \equiv \text{reach}(\syn(L)). \)

Proof. By Corollary 3.7.7 we know \( \syn(L) \equiv ts(\dfaf(L)). \) Recall that \( \dfaf(L) = (L, LW(L), \gamma, F_\gamma) \) and \( \dfaf(L) = (L, LQ(L), \delta, F_\delta) \) where both \( \gamma \) and \( \delta \) have action \( \lambda X. a^{-1}X. \) Observe that:

\[
\text{reach}(ts(\dfaf(L))) = (id_{LQ(L)}, \{ \delta_w : w \in \Sigma^* \}, \lambda f. \delta_n \circ f, \{ f : f \notin \delta_r \}).
\]

By Theorem 3.6.3 it suffices to establish the dfa isomorphism \( \lambda \gamma_w. \delta_w : \text{dfa}_{TM(\dfaf(L))} \rightarrow \text{reach}(ts(\dfaf(L))). \) It is a well-defined bijection because \( \delta_w : LQ(L) \rightarrow LQ(L) \) is completely determined by its domain-codomain restriction \( \gamma_w. \) The initial state and transitions of the two dfsas are defined in the same way. Finally,

\[
\gamma_w(L) \in F_\gamma \iff \varepsilon \in w^{-1}L \\
\iff w \in L \\
\iff \exists u \in \Sigma^* [\delta_w(u^{-1}L) \notin \delta_r(u^{-1}L)] (A) \\
\iff \delta_w \notin \delta_r.
\]

Concerning (A), \( (\Rightarrow) \) follows by contradiction whereas \( (\Leftarrow) \) holds by choosing \( u := \varepsilon \) and observing \( \varepsilon \in [L]^{-1}L. \) \( \square \)

In order to dualise the above constructions one needs the notion of right-quotient closure (see Definition 3.3.1).

Definition 3.7.9 (Right-quotient closure).

1. A JSL-dfa \( \delta \) is right-quotient closed if \( K \in \longq(\delta) \) and \( V \subseteq \Sigma^* \) implies \( KV^{-1} \in \longq(\delta). \)

2. The right-quotient closure of a JSL-dfa \( \gamma \) is the simplified JSL-dfa:

\[
\mathfrak{rqc}(\gamma) := (L(\gamma), (T, \varepsilon, \emptyset), \lambda X. a^{-1}X, \{ K \in T : \varepsilon \in K \})
\]

where \( T \) is the closure of \( \{ jv^{-1} : j \in J(\longq(\gamma)), v \in \Sigma^* \} \) under unions. \( \blacksquare \)

Lemma 3.7.10 (The right-quotient closure is well-defined). Fix any JSL-dfa \( \gamma. \)

1. \( \mathfrak{rqc}(\gamma) \) is a simplified JSL-dfa accepting \( L(\gamma). \)

2. \( \mathfrak{rqc}(\gamma) \) is the smallest right-quotient closed JSL-dfa \( \delta \) such that \( \longq(\gamma) \subseteq \longq(\delta). \) \( \blacksquare \)

Proof.

1. \( T \) contains \( L(\gamma) \) and is closed under unions. It also closed under left-letter-quotients:

\[
a^{-1}(\bigcup_{i \in I} j_i v_i^{-1}) = \bigcup_{i \in I} (a^{-1}j_i)v_i^{-1} = \bigcup_{i \in I}( \bigcup_{k \in K_i} j_{i,k})v_i^{-1} = \bigcup_{i \in I, k \in K_i} j_{i,k}v_i^{-1}.
\]

Then \( \mathfrak{rqc}(\gamma) \) is a well-defined simplified JSL-dfa accepting \( L(\gamma) \) by Lemma 3.4.4.

2. We’ll show \( \mathfrak{rqc}(\gamma) \) is right-quotient closed by showing \( T \) is right-word-quotient closed (recall \( T \) is union-closed):

\[
(\bigcup_{i \in I} j_i v_i^{-1}) v^{-1} = \bigcup_{i \in I}(j_i v_i^{-1}) v^{-1} = \bigcup_{i \in I} j_i (v_i v_i^{-1})^{-1}.
\]

Since \( \mathfrak{rqc}(\gamma) \) is simplified by (1) we deduce \( \longq(\gamma) \subseteq \longq(\mathfrak{rqc}(\gamma)). \) Finally is it the smallest such JSL-dfa because every state is the union of right-quotients of languages in \( J(\longq(\gamma)). \) \( \square \)

45
Theorem 3.7.11 (Transition-semiring dualises right-quotient closure). If $\delta$ is a JSL-reachable JSL-dfa then:

$$\text{acc}(\text{ts}(\delta))^* : (\text{ts}(\delta))^* \rightarrow \text{rqe}(\delta^*)$$

is a JSL-dfa isomorphism.

Proof. Firstly $\gamma := \text{ts}(\delta)$ is JSL-reachable by Lemma 3.7.2, so its dual $\gamma^*$ is simple by Theorem 3.4.3. Then $\text{acc}_{\gamma^*}$ defines a JSL-dfa isomorphism to its simplification $\text{simple}(\gamma^*)$. We'll show the latter is precisely $\text{rqe}(\delta^*)$. Fix $\delta = (t_0, T, \delta_a, F_\delta)$ and $L := L(\delta^*)$. Then by definition $\gamma = (id_T, S_\delta, \gamma_a, F_\gamma)$ where:

$$S_\delta := \{ \delta_K : T \rightarrow T : K \subseteq \Sigma^* \} \quad S_\delta := (S_\delta, \cup, \varnothing) \quad \gamma_a := \lambda f. \delta_a \circ f.$$

Let us break the argument down into steps.

1. We'll show $\text{langs}(\delta^*) \subseteq \text{simple}(\gamma^*)$. Fixing any element of $\gamma^*$ we can rewrite acceptance as follows:

$$u \in \text{acc}_{\gamma^*}(\delta_K) \iff id_T \notin (\gamma_w^*)^*(\delta_K)$$

(by definition)

$$\gamma_w^*(id_T) \neq \delta_K$$

(adjoints)

$$\exists v \in \Sigma^*, \delta_w : (\delta_w(t_0)) \notin \delta \delta_k(\delta_w(t_0))$$

($\delta$ is JSL-reachable)

$$\exists v \in \Sigma^*, \delta_w : (\delta_w(t_0)) \neq \delta \delta_k(\delta_w(t_0))$$

($\delta$ is JSL-reachable)

Recalling Corollary 3.4.9.2, for each $j \in J(\text{langs}(\delta^*))$ we'll show $\delta_j^*$ accepts $j$. Fixing $j_0$, first observe:

$$v \notin [j_0]^{-1} j \iff \forall x \in \Sigma^*, [x \in j_0 \Rightarrow x v \notin j]$$

$$\forall x \in \Sigma^*, [x \in j_0 \Rightarrow x \notin j^v]$$

$$\forall x \in \Sigma^*, [x \in j_0 \Rightarrow x \notin j^v]$$

$$j_0 \subseteq j^v$$

$$j^v \subseteq j_0.$$

(B).

$$u \in \text{acc}_{\gamma^*}(\delta_j) \iff \exists v \in \Sigma^*, j \in J(\text{langs}(\delta^*)), [u \in j^{v_1} \land v \notin [j_0]^{-1} j]$$

(by A)

$$\exists v \in \Sigma^*, z \in Z, [u \in j^{v_1} \land j^{v_1} \subseteq j_0]$$

(by B)

$$u \in j_0.$$ 

Thus $\gamma^*$ accepts every language in $\text{langs}(\delta^*)$ via closure under joins.

2. Next we show $\gamma^*$ is right-quotient closed. Aside from the composite endomorphisms $\gamma_w := \lambda f. \delta_w \circ f$ we also have $\phi_w := \lambda f. f \circ \delta_w : S_\delta \rightarrow S_\delta$. They are well-defined because the composition of join-semilattice morphisms is bilinear. Their adjoints witness right-word-quotient closure:

$$u \in \text{acc}_{\gamma^*}(\delta_K) \iff id_T \notin (\phi_w)^* (\delta_K)$$

(by definition)

$$\gamma_w^*(id_T) \neq \delta_K$$

(adjoints)

$$\phi_w \circ \gamma_w^*(id_T) \neq \delta_K$$

(adjoints)

$$\delta_{wu} \neq \delta_{wu} \delta_K$$

(eq)
Corollary 3.7.12 (Right-quotient closed vs. finite $\Sigma$-generated idempotent semirings).

1. If $\delta$ is a simple right-quotient closed JSL-dfa, $\delta^* \cong \text{ts}(\delta^*)$ is a $\Sigma$-generated idempotent semiring acting on itself.

2. If $S = \langle S, \cdot, 1_S \rangle$ is a finite $\Sigma$-generated idempotent semiring and $s_1 \in S$ then $(1_S, S, \lambda s.s \cdot s, a, \{ s \in S : s \not\in \Sigma s_1 \})^*$ is a simple right-quotient closed JSL-dfa.

Proof.

1. Modulo isomorphism $\delta$ is simplified. Then $\delta = \text{tqc}(\delta) \cong (\text{ts}(\delta^*))^*$ by Theorem 3.7.11, so that $\delta^* \cong \text{ts}(\delta^*)$.

2. First, $\gamma := (1_S, S, \lambda s.s \cdot s, a, \{ s \in S : s \not\in \Sigma s_1 \})$ is a well-defined JSL-dfa because right-multiplication preserves joins. It is JSL-reachable because $S$ is $\Sigma$-generated. Next we'll show $\lambda f.f(1_S) : \text{ts}(\gamma) \to \gamma$ is a JSL-dfa isomorphism. It is a well-defined function because right-multiplication preserves joins. Both $\gamma$ is $\Sigma$-generated and $\gamma_U(1_S) = \llbracket U \rrbracket_S$. It is injective because each $\gamma_U = \lambda s.s \cdot S \llbracket U \rrbracket_S$ acts as right-multiplication by $\llbracket U \rrbracket_S$. Finally it preserves joins and multiplication. Then $\gamma^* \cong (\text{ts}(\gamma))^* \cong \text{tqc}(\gamma^*)$ is simple and right-quotient closed by Theorem 3.7.11.

Corollary 3.7.13 (Quotients of finite idempotent semirings).

1. Given a JSL-dfa inclusion morphism $\iota : \gamma \to \delta$ between simplified right-quotient closed JSL-dfas,

$$\lambda \llbracket U \rrbracket_{T S(\delta^*)}, \llbracket U \rrbracket_{T S(\gamma^*)} : T S(\delta^*) \to T S(\gamma^*)$$

is a well-defined surjective semiring morphism.

2. Let $f : \langle S, \cdot, 1_S \rangle \to \langle \top, \cdot, 1_\top \rangle$ be a surjective semiring morphism where $S$ is a finite $\Sigma$-generated idempotent semiring. Given any $s_0 \in S$ we have the JSL-dfa embedding:

$$f_* : (id_\top, \top, \lambda t. t \cdot T f(a), \{ t : t \not\in \llbracket f(s_0) \rrbracket \})^* \to (id_S, S, \lambda s.s \cdot S a, \{ s : s \not\in \Sigma s_0 \})^*$$

between simple right-quotient closed JSL-dfas.

Proof.

1. Firstly $\iota_* : \delta^* \to \gamma^*$ is a surjective JSL-dfa morphism by Theorem 3.2.3. Since $\gamma$ and $\delta$ are right-quotient closed, $\text{ts}(\delta^*) \cong (\text{tqc}(\delta))^* = (\delta)^* \cong (\text{tqc}(\gamma))^* \cong \text{ts}(\gamma^*)$ by applying Theorem 3.7.11. Then we have the surjective JSL-dfa morphism $f : \text{ts}(\delta^*) \to \text{ts}(\gamma^*)$. It is a join-semilattice morphism preserving the unit (initial state) and right-multiplication by generators. Then $f((\delta_0)_*) := (\gamma_0)_*$ and thus $f((\delta_U)_*) = (\gamma_U)_*$ by induction over words and joins. Then $f$ preserves the multiplication too:

$$f((\delta_V)_* \circ (\delta_U)_*) = f((\delta_V \circ \delta_U)_*) = f((\delta_{V \cup U})_*) = (\gamma_{V \cup U})_* = (\gamma_V \circ \gamma_U)_* = (\gamma_V)_* \circ (\gamma_U)_*$$

so it is a surjective semiring morphism. Finally it preserves the generators so has the claimed description.

2. The surjective semiring morphism also defines a JSL-dfa morphism:

$$f : (id_S, S, \lambda s.s \cdot S a, \{ s : s \not\in \Sigma s_0 \}) \to (id_T, \top, \top, t \cdot T f(a), \{ t : t \not\in \llbracket f(s_0) \rrbracket \})$$

because right-multiplication preserves joins. Both JSL-dfas are JSL-reachable because $f$ is surjective, so that $f[ \Sigma ]$ generates $T$. Then its adjoint defines an injective JSL-dfa morphism between simple JSL-dfas. Finally each JSL-dfa is right-quotient closed via closure under left multiplication on the dual side.

Next we dualise the syntactic semiring.
**Definition 3.7.14** (L’s minimal syntactic JSL-dfa). The closure of LRW(L) := \{u^{-1}Lv^{-1} : u, v \in \Sigma^*\} under unions defines the minimal syntactic JSL-dfa \(\mathcal{D}_{\text{syn}}(L)\) i.e. the smallest right-quotient closed JSL-dfa accepting \(L\).

**Note 3.7.15.** The minimal syntactic JSL-dfas satisfies \(\mathcal{D}_{\text{syn}}(L) = \mathcal{R}(\mathcal{D}_{\text{syn}}(L))\).

**Corollary 3.7.16** (Dualising the syntactic semiring). We have the JSL-dfa isomorphism:

\[
\text{acc}_{(\text{syn}(L'))*} : (\text{syn}(L'))* \rightarrow \mathcal{D}_{\text{syn}}(L)
\]

**Proof.** First let \(\delta := \mathcal{D}_{\text{syn}}(L')\). By Theorem 3.7.6 we know \(\text{syn}(L') \cong \text{ts}(\delta)\) so that \((\text{syn}(L'))* \cong (\text{ts}(\delta))*\). By Theorem 3.7.11 we know \((\text{ts}(\delta))* \cong \mathcal{R}(\delta)\) because \(\delta\) is JSL-reachable (see Corollary 3.4.8). Finally by Theorem 3.3.6 we have \(\delta* \cong \mathcal{D}_{\text{syn}}(L)\), so that \((\text{syn}(L'))* \cong \mathcal{R}(\mathcal{D}_{\text{syn}}(L))\). Since \(\mathcal{R}(\mathcal{D}_{\text{syn}}(L))\) is simplified this isomorphism must be the acceptance map.

Finally we describe the dual of the power semiring of the syntactic monoid. It is essentially the canonical boolean syntactic JSL-dfa from Definition 3.6.4.

**Corollary 3.7.17** (Dualising \(\mathcal{P}_{\text{Syn}}(L)\)). Let \(\delta := \mathcal{R}(\mathcal{D}_{\text{syn}}(L))\).

1. We have the semiring isomorphism \(\lambda \delta_U \cdot \{[u]_{\mathcal{S}_L} : u \in U\} : \mathcal{T}_{\delta} \rightarrow \mathcal{P}_{\text{Syn}}(L)\).
2. We have the JSL-dfa isomorphism \(\text{acc}_{(\text{ts}(\delta))*} : (\text{ts}(\delta))* \rightarrow \mathcal{D}_{\text{syn}}(L)\).

**Proof.**

1. Denote the candidate isomorphism by \(\alpha\). Given \(\gamma := \mathcal{D}_{\text{syn}}(L)\) then \(\delta_U = \lambda \gamma_U \cdot \gamma_{\gamma}[S]\). Given \(\delta_{U_1} = \delta_{U_2}\) then applying them to \(\{[e]_{\mathcal{S}_L}\}\) we see \(\alpha\) is a well-defined injective function. It is clearly surjective and also preserves joins i.e. \(\alpha(\delta_{U_1 \cup U_2}) = \alpha(\delta_{U_1}) \cup \alpha(\delta_{U_2})\). Finally \(\alpha(\delta_{\emptyset}) = \emptyset\) and the multiplication is also preserved:

\[
\alpha(\delta_V \cdot \delta_U) = \alpha(\delta_{U \cdot V}) = \{[x]_{\mathcal{S}_L} : x \in U \cdot V\} = \{[u]_{\mathcal{S}_L} : u \in U\} \cdot \{[v]_{\mathcal{S}_L} : v \in V\}
\]

2. By Example 3.2.12 \(\delta* \cong \mathcal{R}(\mathcal{D}_{\text{syn}}(L))\) hence \(\delta* \cong \mathcal{D}_{\text{syn}}(L)\) by Theorem 3.6.6. Applying Theorem 3.7.11 yields:

\[
(\text{ts}(\delta))* \cong \mathcal{R}(\delta) \cong \mathcal{D}_{\text{syn}}(L) = \mathcal{D}_{\text{syn}}(L),
\]

since the latter is simplified and right-quotient closed by construction.

---

4 The Kameda-Weiner Algorithm and Beyond

4.1 L-coverings

An L-covering is an edge-covering of the dependency relation \(\mathcal{D}_{\text{RL}}\) (Definition 3.1.8) by left-maximal bicliques. That is, each biclique \(A \times B \subseteq \mathcal{D}_{\text{RL}}\) is inclusion-maximal on the left. Importantly, they can be defined as certain \(\text{Dep}\)-morphisms.

**Definition 4.1.1** (L-coverings). Fix any regular language \(L \subseteq \Sigma^*\).

1. An L-covering is a \(\text{Dep}\)-morphism \(\mathcal{D}_{\text{RL}} : \mathcal{D}_{\text{RL}} \rightarrow \mathcal{H}\) such that \(\mathcal{H}_L = \mathcal{LW}(L')\).

The \(\text{Dep}\)-morphism is determined by \(\mathcal{H}\), so it may be denoted \((L, \mathcal{H})\). We may also refer to the L-covering via the relation \(\mathcal{H} \subseteq \mathcal{H}_s \times \mathcal{LW}(L')\) alone.

2. Given an L-covering \(\mathcal{H}\), Definition 2.2.2 provides \((L, \mathcal{H})_1 \subseteq \mathcal{LW}(L) \times \mathcal{H}_s\). But we may also directly define:

\[
(L, \mathcal{H}_s)_1(\mathcal{H}(h, Y) \Rightarrow \mathcal{D}_{\text{RL}}(u^{-1}L, Y))
\]

without knowing \(\mathcal{H} \subseteq \mathcal{H}_s \times \mathcal{LW}(L')\) is an L-covering.
**Definition 4.1.2** *(L-covering constructions).* Fix an L-covering $\mathcal{DR}_L : \mathcal{DR}_L \to \mathcal{H}$. 

1. $\mathcal{H}$'s biclique-form is the L-covering $\mathcal{H}^0 \subseteq \mathcal{H}^0_s \times \mathcal{LW}(L')$ where:
   
   $$\mathcal{H}^0_s := \{(L, \mathcal{H})_{[h_s]} \times \mathcal{H}[h_s] : h_s \in \mathcal{H}_s\} \quad \mathcal{H}^0(A \times B, Y) : \Longleftrightarrow Y \in B.$$

   It turns out that $(L, \mathcal{H}^0)_-(X, A \times B) \iff X \in A$. Finally, we say $\mathcal{H}$ is in biclique-form if $\mathcal{H} = \mathcal{H}^0$.

2. $\mathcal{H}$'s induced nfa $\mathcal{N}_\mathcal{H}$ has states $\mathcal{H}_s$ and is defined:
   
   $$I_{\mathcal{N}_\mathcal{H}} = (L, \mathcal{H})_{[L]} \quad F_{\mathcal{N}_\mathcal{H}} := \{h \in \mathcal{H}_s : \varepsilon \in \bigcap \{L, \mathcal{H}\}_{[h]}\}$$

   $$\mathcal{N}_{\mathcal{H}, a}(h_1, h_2) : \iff \forall X \in \mathcal{LW}(L).((L, \mathcal{H})_-(X, h_1) \Rightarrow (L, \mathcal{H})_-(a^{-1}X, h_2)).$$

   Just as $(L, \mathcal{H})$ is completely determined by $\mathcal{H}$, the induced nfa $\mathcal{N}_{\mathcal{H}}$ is completely determined by $(L, \mathcal{H})$.

3. An $L$-covering $\mathcal{H}'$ extends another $L$-covering $\mathcal{H}$ if $\mathcal{H}_s = \mathcal{H}'_s$, $\mathcal{H} \subseteq \mathcal{H}'$ and $(L, \mathcal{H})_+ = (L, \mathcal{H}')_+$. We say $\mathcal{H}$ is maximal if its only extension is itself.

4. $\mathcal{H}$ is legitimate if $L(\mathcal{N}_\mathcal{H}) = L$, see [KW70, Definition 16].

5. $\mathcal{H}$'s dual is the $L'$-covering $\mathcal{H}^o := (L, \mathcal{H})_{-} \subseteq \mathcal{H}_s \times \mathcal{LW}(L').$

**Note 4.1.3** *(Concerning extensions of L-coverings).* Given any two $L$-extensions satisfying $\mathcal{H}_s = \mathcal{H}'_s$ and $\mathcal{H} \subseteq \mathcal{H}'$ we necessarily have $(L, \mathcal{H})_+ \subseteq (L, \mathcal{H}')_+$. Then Definition 4.1.2.3 could equivalently require $(L, \mathcal{H})_+ \subseteq (L, \mathcal{H}')_+$, which is more in-keeping with ‘maximality’.

We now prove various basic facts concerning $L$-coverings.

**Lemma 4.1.4** *(L-coverings).*

1. $\mathcal{H} \subseteq \mathcal{H}_s \times \mathcal{LW}(L')$ is an $L$-covering iff $\mathcal{DR}_L = I; \mathcal{H}$ for some $I \subseteq \mathcal{LW}(L) \times \mathcal{H}_s$.

2. Each $L$-covering is a Dep-monomorphism via the witnesses:

   $$\begin{array}{ccc}
   \mathcal{LW}(L') & \xrightarrow{\Delta_{\mathcal{LW}(L')}} & \mathcal{LW}(L') \\
   \mathcal{DR}_L & \downarrow & \mathcal{H} \\
   \mathcal{LW}(L) & \xleftarrow{(L, \mathcal{H})_-} & \mathcal{H}_s
   \end{array}$$

3. If $\mathcal{H}$ is an $L$-covering then so is its biclique-form $\mathcal{H}^0$; moreover $(L, \mathcal{H}^0)_-(X, A \times B) \iff X \in A$.

4. If $\mathcal{H}$ is an $L$-covering in biclique-form then its induced nfa satisfies:

   $$A \times B \in I_{\mathcal{N}_\mathcal{H}} \iff L \in A \quad A \times B \in F_{\mathcal{N}_\mathcal{H}} \iff \varepsilon \in \bigcap A$$

   $$\mathcal{N}_{\mathcal{H}, a}(A_1 \times B_1, A_2 \times B_2) \iff \gamma_a[A_1] \subseteq A_2 \quad \text{where } \gamma_a : \lambda X.a^{-1}X.$$

5. $L(\mathcal{N}_\mathcal{H}) = L(\mathcal{N}_{\mathcal{H}})$ because $q := \lambda h_s.(L, \mathcal{H})_{[h_s]} \times \mathcal{H}[h_s] : \mathcal{H}_s \to \mathcal{N}_{\mathcal{H}}$ is a surjection which preserves/reflects initial states, final states and transitions. If $\mathcal{N}_{\mathcal{H}}$ is state-minimal then $q$ is an nfa isomorphism.

6. $L(\mathcal{N}_\mathcal{H}) \subseteq L$ for each $L$-covering $\mathcal{H}$.

7. Let $\mathcal{H}$ be an $L$-covering.

   a. $\mathcal{H}^o$ is a well-defined maximal $L'$-covering.
   b. $\mathcal{H} \subseteq \mathcal{H}^{oo}$ and $(L, \mathcal{H})_+ = (\mathcal{H}^o)^_+ = (L, \mathcal{H}^{oo})_+$, so $\mathcal{H}^{oo}$ extends $\mathcal{H}$.
   c. $\mathcal{H}$ is maximal iff $\mathcal{H} = \mathcal{H}^{oo}$.

---

8 However, even when $\mathcal{N}_{\mathcal{H}}$ is not state-minimal $q$ is almost the same thing as an isomorphism.
Proof.

1. If $\mathcal{DL}_L: \mathcal{DR}_L \rightarrow \mathcal{H}$ is an L-covering it is a Dep-morphism, so $(L, \mathcal{H})_{\rightarrow} \mathcal{H} = \mathcal{DR}_L$ via the maximum witnesses (Lemma 2.2.3). Conversely if $\mathcal{DR}_L = \mathcal{I}; \mathcal{H}$ we know $\mathcal{I}; \mathcal{H} = \mathcal{DR}_L = \mathcal{DR}_L; \mathcal{I}_{\mathcal{LW}(L)}$ hence $\mathcal{DR}_L$ is a Dep-morphism.

2. The commuting diagram follows via maximum witnesses. It defines a Dep-mono because the upper witness is a bijective function, recalling Dep-composition from Definition 2.1.6.

3. Since $\mathcal{H}$ is an L-covering we know $(L, \mathcal{H})_{\rightarrow} \mathcal{H} = \mathcal{DR}_L$. For completely general reasons $\bigcup \mathcal{H} = \mathcal{DR}_L$ i.e. the union of the cartesian products $(L, \mathcal{H})_{\rightarrow} \mathcal{H}[h_x]$ is $L$’s dependency relation (see Note 2.1.2). If we define $\mathcal{I} \subseteq \mathcal{LW}(L) \times \mathcal{H}_x$ as $\mathcal{I}(X, A \times B) : \iff X \in A$ then:

$$\mathcal{I}; \mathcal{H}^t(X, Y) \iff \exists A \times B \in \mathcal{H}^t.\Bigr[ X \in A \land Y \in B \Bigr]$$

$$\iff \exists h_x \in \mathcal{H}_x.\Bigr[ (X, Y) \in (L, \mathcal{H})_{\rightarrow} \mathcal{H}[h_x] \Bigr]$$

$$\iff (X, Y) \in \bigcup \mathcal{H}_x^t$$

$$\iff \mathcal{DL}_L(X, Y).$$

Then by (1) $\mathcal{H}^t$ is a well-defined L-covering. It remains to establish $(L, \mathcal{H}^t)_{\rightarrow} \mathcal{I}$. To this end, let $A \times B = \mathcal{S}^t[h_x] \times \mathcal{H}[h_x]$ and consider:

$$(L, \mathcal{H}^t)_{\rightarrow} (X, A \times B) : \iff \mathcal{H}^t[A \times B] \subseteq \mathcal{DR}_L[X]$$

(by def. of $\mathcal{H}^t$).

4. Concerning the transitions:

$$\mathcal{N}_{\mathcal{H}^t}(A_1 \times B_1, A_2 \times B_2)$$

: $\iff \forall X \in \mathcal{LW}(L).[(L, \mathcal{H})_{\rightarrow} (X, A_1 \times B_1) \Rightarrow (L, \mathcal{H})_{\rightarrow} (a^{-1}X, A_2 \times B_2)]$

: $\iff \forall X \in \mathcal{LW}(L).[X \in A_1 \Rightarrow a^{-1}X \in A_2]$

(by (3))

5. Consider the well-defined surjection $q: \mathcal{H}_x \rightarrow \mathcal{H}^t_x$ with action $\lambda h_x.(L, \mathcal{H})_{\rightarrow} \mathcal{H}[h_x]$. It preserves and reflects the initial states and also the final states:

$$h_x \in I_{\mathcal{N}_\mathcal{H}} \iff h_x \in (L, \mathcal{H})_{\rightarrow} [L] \iff L \in (L, \mathcal{H})_{\rightarrow} [h_x] \iff q(h_x) \in I_{\mathcal{N}_\mathcal{H}^t}.$$}

The characterisations of the initial/final states follow easily.

6. By (5) we may assume the L-covering is in biclique-form. The induced nfa $\mathcal{N}_\mathcal{H}$ is described in (4). If $w \in L(\mathcal{N}_\mathcal{H})$ then by induction we have $\gamma_w[A_1] \subseteq A_n$ where $L \in A_1$, $A_n \times B_n \subseteq \mathcal{DR}_L$ and $\varepsilon \in \bigcap A_n$. Thus $\varepsilon \in w^{-1}L$ so $w \in L$. 

50
7. a. By Lemma 4.1.4.1 we know \(<L, \mathcal{H}> \subseteq \mathcal{H} = \mathcal{DR}_L\) hence \(\mathcal{H}: (\mathcal{L}, \mathcal{H})^- = (\mathcal{DR}_L)^- = \mathcal{DR}_L^\sim\). Thus \(\mathcal{H}^\circ := (\mathcal{L}, \mathcal{H})^\circ\) is an \(L^\sim\)-covering by applying Lemma 4.1.4.1 again. Finally \(\mathcal{H}^\circ\) is maximal because its converse is a maximal lower witness.

b. Below on the left we’ve depicted \(\mathcal{H}\) together with the respective lower component \((\mathcal{L}, \mathcal{H})_-\).

\[
\begin{align*}
\text{LW}(L') & \xrightarrow{\Delta_{\text{LW}(L')}} \text{LW}(L') \\
\mathcal{DR}_L & \xrightarrow{(\mathcal{L}, \mathcal{H})} \mathcal{H}_s \quad \text{LW}(L) & \xrightarrow{\Delta_{\text{LW}(L)}} \text{LW}(L) \\
\text{LW}(L') & \xrightarrow{\Delta_{\text{LW}(L')}} \text{LW}(L') \\
\mathcal{DR}_L & \xrightarrow{(\mathcal{L}, \mathcal{H})_-} \mathcal{H}_s
\end{align*}
\]

The central diagram shows \(\mathcal{H}\)’s dual \(L^\sim\)-covering \(\mathcal{H}^\circ := (\mathcal{L}, \mathcal{H})^\circ\) and the respective lower component \((\mathcal{L}, \mathcal{H}^\circ)_-\). Then the central diagram arises by dualising \(\mathcal{H}\) and the right-most diagram arises by dualising \(\mathcal{H}^\circ\). Notice that since \(\mathcal{H}^\circ\) is already maximal, the right-most square swaps and reverses both relations. In particular:

- from left to right \((\mathcal{L}, \mathcal{H})_- = (\mathcal{H}^\circ)^- = (\mathcal{L}, \mathcal{H}^\circ^\sim)_-\).
- \(\mathcal{H}^\circ = (\mathcal{L}, \mathcal{H})_-\) implies \(\mathcal{H} \subseteq (L^\sim, \mathcal{H}^\circ)^- = \mathcal{H}^\circ\).

c. If \(\mathcal{H}\) is maximal then \(\mathcal{H} = \mathcal{H}^\circ\) by (b). Conversely if \(\mathcal{H} = (\mathcal{H}^\circ)^\circ\) then it is maximal by (a).

d. If \(\mathcal{H}\) is maximal then \(\mathcal{H}\) amounts to a bijective relabelling of \(\mathcal{H}_s\), so it is also maximal.

e. Since \((\mathcal{L}, \mathcal{H})_- = (\mathcal{L}, \mathcal{H}^\circ)_-\), we know \(\mathcal{N}_\mathcal{H} = \mathcal{N}_\mathcal{H}^\circ\), hence \(\mathcal{H}^\circ\) is also legitimate.

f. We have \(\mathcal{H}^\circ = (L^\sim, \mathcal{H}^\circ)_-\) and also \((\mathcal{L}, \mathcal{H}^\circ)_- = (\mathcal{H}^\circ)^\circ\) by (b). Then constructing the clique-form of \(\mathcal{H}^\circ\) amounts to constructing cliques:

\[
(L^\sim, \mathcal{H}^\circ)_-\cdot [h_s] = \mathcal{H}^\circ\cdot [h_s] \times (L, \mathcal{H}^\circ)_-\cdot [h_s]
\]

and the claim the follows.

8. Given \(A \times B \in (\mathcal{N}_\mathcal{H})_u(I_{\mathcal{N}_\mathcal{H}})\) we’ll prove \(u^{-1}L \in A\) by induction on \(u\). If \(u = \varepsilon\) this holds by definition of \(I_{\mathcal{N}_\mathcal{H}}\). If \(u = u_0a\) we have \(A_0 \times B_0 \in (\mathcal{N}_\mathcal{H})_u(I_{\mathcal{N}_\mathcal{H}})\) and \(a^{-1}[A_0] \subseteq A_1\) by Lemma 4.1.4.4. Then by induction \(u_0^{-1}L \in A_0\) and hence \((u_0a)^{-1}L \in A\), so we are done.

\(\square\)

### 4.2 Saturated machines

There are various ways an nfa can be have many initial/final states and transitions.

**Definition 4.2.1** (Locally/intersection-saturated and transition-maximality). Let \(\mathcal{N} = (I, Z, \mathcal{N}_a, F)\) be an nfa.

1. \(\mathcal{N}\) is **locally-saturated** if for all \(a \in \Sigma\) and \(z, z_1, z_2 \in Z\),

\[
(z \in I \iff L(\mathcal{N}_a) \subseteq L(\mathcal{N}) \quad \mathcal{N}_a(z_1, z_2) : \iff L(\mathcal{N}_a z_2) \subseteq a^{-1}L(\mathcal{N}_a z_1).
\]

2. \(\mathcal{N}\) is **intersection-saturated** if for all \(z, z_1, z_2 \in Z\),

\[
\mathcal{N}_a(z_1, z_2) \iff \forall u \in \Sigma^* \cdot (z_1 \in \mathcal{N}_u) \Rightarrow z_2 \in \mathcal{N}_u(z_1)\)
\[
z \in F_{\mathcal{N}} \iff \forall u \in \Sigma^* \cdot (z \in \mathcal{N}_u \Rightarrow \mathcal{N}_u \cap F_{\mathcal{N}} \neq \varnothing).
\]

These are the conditions for transitions and final states from Kameda and Weiner’s intersection rule [KW70].

3. \(\mathcal{N}\) is **transition-maximal** if adding transitions or colouring additional initial/final states changes the accepted language. More formally, an nfa \(\mathcal{M}\) extends \(\mathcal{N}\) if \(\mathcal{M} = (I_{\mathcal{M}}, Z, \mathcal{M}_a, F_{\mathcal{M}})\) where \(I \subseteq I_{\mathcal{M}}\), each \(\mathcal{N}_a \subseteq \mathcal{M}_a\), \(F \subseteq F_{\mathcal{M}}\) and finally \(L(\mathcal{M}) = L(\mathcal{N})\). Then \(\mathcal{N}\) is transition-maximal if its only extension is itself.

The concept of being locally-saturated arises naturally from canonical constructions, as we’ll see. It is ‘local’ because one can enforce it without changing the languages accepted by the individual states. It is worth clarifying the second concept straight away.
Note 4.2.2. An nfa $\mathcal{N}$ is intersection-saturated iff the following hold:

- whenever for every $u$-path to $z_1$ there exists a $ua$-path to $z_2$ then $\mathcal{N}_a(z_1, z_2)$.
- whenever for every $u$-path to $z$ we have $u \in L(\mathcal{N})$ then $z$ is final.

Then each transition relation $\mathcal{N}_a$ can be reconstructed from the deterministic transitions $\mathcal{N}_u[I] \rightarrow a \mathcal{N}_{ua}[I]$ of the reachable subset construction. Soon we’ll prove an nfa $\mathcal{N}$ is intersection-saturated iff $\text{rev}(\mathcal{N})$ is locally-saturated. □

Perhaps unsurprisingly, transition-maximal machines are both locally-saturated and intersection-saturated. We now provide various examples of nfas in different classes.

Example 4.2.3 (Comparing notions of saturation).

1. **Locally but not intersection-saturated (via final states).** The nfa below accepts $a + aa$ and is locally-saturated e.g. there is no transition from the left-most state to the right-most because $\{\varepsilon\} \notin a^{-1}\{aa\}$. However it is not intersection-saturated because the central state should be final by Note 4.2.2.

```
 i  a  i  a  o
```

2. **Locally but not intersection-saturated (via transitions).** This locally-saturated nfa accepts $a(bb^* + cc^*)$:

```
 b  ↓  ↓  c  ↓  ↓  o
 o  b  ←  ←  i  a  ↓  ↓  a  ←  ←  c  →  →  o
 b  ↑  ↑  c  →  →  o
```

It is not intersection-saturated because by Note 4.2.2 it should have the dashed transitions too.

3. **Intersection-saturated but not locally-saturated.** Take the reverse nfa of either (1) or (2). This follows by Theorem 4.2.9 further below.

4. **Locally-saturated, not transition-maximal.** Example (2) is locally-saturated but not transition-maximal.

5. **Locally and intersection-saturated, not transition-maximal.** This nfa accepts $L := a + b + (a^* + b^*)c^*$ and is locally-saturated e.g. there is no dashed $c$-transition because $c^* \notin c^{-1}\{\varepsilon\}$. It is also intersection-saturated.

```
 a  ↓  ↓  a  ↓  ↓  b  ↓  ↓  b
 a  ↓  ↓  a  ↓  ↓  b  ↓  ↓  b
```

However, it is not transition-maximal – adding the dashed $c$-transition preserves the accepted language.

6. The nfa $\text{dfa}_L(L)$ from Example 3.3.4 is always transition-maximal, as the reader may verify. □

There is a canonical way to locally saturate an nfa.

Definition 4.2.4 (Irreducible simplification). We define the irreducible simplification of an nfa $\mathcal{N} = (I, Z, R_a, F)$ as:

$$\text{simple}_\varepsilon(\mathcal{N}) := (\{X \in J(\mathcal{S}) : X \subseteq L(\mathcal{N})\}, J(\mathcal{S}), \lambda X.\, a^{-1}X, \{X \in J(\mathcal{S}) : \varepsilon \in X\})$$

where $\mathcal{S} := \text{langs}(\text{dep}(\mathcal{N}))$ is the join-semilattice of languages accepted by $\mathcal{N}$. □

Note 4.2.5 (Irreducible simplification is canonical). $\text{simple}_\varepsilon(\mathcal{N})$ is the lower nfa of $\text{Airr}(\text{simple}(\text{dep}(\mathcal{N})))$. □
Lemma 4.2.6 (Concerning irreducible simplifications).

1. simple

2. L((simple

3. simple

4. simple

Proof.

1. Det

2. Each state

3. Let

4. Follows by (2).

Lemma 4.2.7. simple

is locally-saturated with no more states than

Proof. Recall Definition 4.2.4 and let

Simple

is locally-saturated via their initial states and transition structure (Definition 3.2.5) because each state

accepts

by Lemma 3.4.2. Finally, the lower nfa of

is locally-saturated by (1), and this property is preserved by the nfa isomorphism. Recall the surjective join-semilattice morphism

where

is the principal downset generated by

is idempotent.

Finally, irreducible simplifications are precisely those nfas which are both locally-saturated and ‘union-free’.

Theorem 4.2.8 (Characterizing irreducible simplifications). The following statements are equivalent:

1. N \cong simple

2. \lambda z.L(N_{\overline{z}}) : N \rightarrow simple(N) defines an nfa isomorphism.

3. N is locally-saturated and satisfies:

∀ z ∈ Z. ∀ S ≤ Z. [(L(N_{\overline{z}}) = L(N_{\overline{z}S})) ⇒ z ∈ S] (union-free)

Proof.

1. (1 \iff 2): given (1) then each state accepts a distinct language, so there is only one possible nfa isomorphism.

2. (2 \implies 3): Suppose \lambda z.L(N_{\overline{z}}) defines an nfa isomorphism. Then N is locally-saturated because simple

is locally-saturated by (1), and this property is preserved by the nfa isomorphism. Recall the join-semilattice of accepted languages

and also the relationship

from Definition 3.1.1. Then (union-free) holds via Lemma 4.2.6.2 because it asserts each

accepts

for each

union-free).
3. \(3 \implies 2\): Suppose \(\mathcal{N}\) is locally-saturated and satisfies (union-free). Firstly, \(f := \lambda z. L(N_{\emptyset z}) : Z \to J(\text{langs}(\mathcal{N}))\) is a well-defined function because by (union-free) we know each \(L(N_{\emptyset z}) \in J(\text{langs}(\mathcal{N}))\). Furthermore \(f\) is injective for otherwise (union-free) would fail, and surjective because \(J(\text{langs}(\mathcal{N}))\) is the minimal join-generating subset of \(\text{langs}(\mathcal{N})\) (see Note 2.2.8.4). Concerning the nfa isomorphism, \(z\) is final iff \(\varepsilon \in L(N_{\emptyset z})\) hence \(f\) preserves and reflects final states. The initial states and transitions are preserved and reflected because \(\mathcal{N}\) is locally-saturated.

We now characterize the intersection-saturated nfas.

**Theorem 4.2.9.** An nfa \(\mathcal{N}\) is intersection-saturated iff \(\text{rev}(\mathcal{N})\) is locally-saturated.

**Proof.** Let \(\mathcal{N} = (I, Z, N_a, F)\) and fix any \(z \in Z\). For completely general reasons:

\[
L(\text{rev}(\mathcal{N}))_{\emptyset z} = L((z), Z, N^*_a, I) = (L(I, Z, N_a, \{z\}))^* = (\{u \in \Sigma^*: z \in N_a[I]\})^*.
\]

Assuming \(\text{rev}(\mathcal{N})\) is locally-saturated we prove the condition concerning transitions:

\[
N_a(z_1, z_2) \iff N^*_a(z_2, z_1) \iff L((\text{rev}(\mathcal{N}))_{\emptyset z_2}) \subseteq a^{-1}L((\text{rev}(\mathcal{N}))_{\emptyset z_1}) \tag{\text{rev}(\mathcal{N}) \text{ locally-saturated}}
\]

\[
\{u \in \Sigma^*: z_1 \in N_a[I]\} \subseteq a^{-1}(\{u \in \Sigma^*: z_2 \in N_a[I]\}) \tag{\text{above}}
\]

\[
\{u \in \Sigma^*: z_1 \in N_a[I]\} \subseteq (\{u \in \Sigma^*: z_2 \in N_a[I]\})a^{-1} \tag{since \((a^{-1}X)^* = X^*a^{-1})
\]

\[
\forall u \in \Sigma^*(z_1 \in N_a[I] \Rightarrow z_2 \in N_a[I]).
\]

Finally \(\forall u \in \Sigma^*(z \in N^*_a[F_N] \Rightarrow N^*_a[F_N] \cap I_N \neq \emptyset)\) is equivalent to requiring \(L(N_{\emptyset z}) \subseteq L\), which follows by local saturation. Conversely if \(\mathcal{N}\) is intersection-saturated it is locally-saturated by reversing the above arguments.

Then there is also a canonical way to intersection saturate an nfa.

**Corollary 4.2.10.** \(\text{rev}(\text{simple}_a(\text{rev}(\mathcal{N})))\) is an intersection-saturated nfa accepting \(L(\mathcal{N})\), no larger than \(\mathcal{N}\).

**Proof.** \(\text{rev}(\text{simple}_a(\text{rev}(\mathcal{N})))\) accepts the same language because \(\text{rev}(\mathcal{N})\) reverses it and \(\text{simple}_a(\mathcal{N})\) preserves it (Lemma 4.2.6.1). Moreover \(\text{rev}(\mathcal{N})\) preserves the number of states and \(\text{simple}_a(\mathcal{N})\) never increases it by Lemma 4.2.7. By the same Lemma we know \(\text{simple}_a(\text{rev}(\mathcal{N}))\) is locally-saturated, hence its reverse satisfies the intersection rule by Theorem 4.2.9.

Finally we collect a few results concerning transition-maximal nfas. Given any nfa, there is a non-canonical way to construct a transition-maximal extension: keep adding initial/final states and transitions whenever doing so preserves the accepted language. Let us formally state this basic fact, an instantiation of Zorn’s Lemma in the finite setting.

**Lemma 4.2.11.** Every nfa \(\mathcal{N}\) has a transition-maximal extension (see Definition 4.2.1.2).

**Lemma 4.2.12.** \(\text{rev}(\mathcal{N})\) preserves transition-maximality.

**Proof.** Holds because an nfa \(\mathcal{M}\) extends \(\text{rev}(\mathcal{N})\) iff \(\text{rev}(\mathcal{M})\) extends \(\mathcal{N}\).

Transition-maximal transitions are determined by the order-structure of \(\text{langs}(\mathcal{N}) \supseteq \text{LW}(L(\mathcal{N}))\).

**Lemma 4.2.13 (Transition-maximal transitions and finality).** If an nfa \(\mathcal{N} = (I, Z, N_a, F)\) is transition-maximal,

\[
N_a(z_1, z_2) \iff \forall X \in \text{LW}(L(\mathcal{N})). (L(N_{\emptyset z_2}) \subseteq X \Rightarrow L(N_{\emptyset z_1}) \subseteq a^{-1}X)
\]

\[
z \in F \iff \forall X \in \text{LW}(L(\mathcal{N})). (L(N_{\emptyset z_1}) \subseteq X \Rightarrow \varepsilon \in X).
\]

**Proof.** Let \(L := L(\mathcal{N})\).
1. We’ll prove (T). Given an nfa $N$ where $N_a(z_1, z_2)$ then $L(N_{a_{z_2}z_1}) \subseteq a^{-1}L(N_{a_{z_2}z_1})$, so $(\Rightarrow)$ holds generally because $a^{-1}(-)$ is monotonic w.r.t. inclusions. We’ll refer to (T)’s right hand side by (RHS).

Suppose $N$ is transition-maximal and (RHS) holds for specific $z_1, z_2$. For a contradiction assume $(z_1, z_2) \notin N_a$, letting $M$ be $N$ with the new transition. We know $L \subseteq L(M)$ and we’ll show the converse, contradicting transition-maximality. Consider:

$I \ni u \rightarrow_N z_1 \stackrel{a}{\rightarrow} z_2 \stackrel{v}{\rightarrow}_M f \in F$

where the $v$-path uses the new transition $n \geq 0$ times. We know $L(N_{a_{z_2}z_1}) \subseteq u^{-1}L$ and may write $v = (\prod_{1 \leq i \leq n} v_i a_i)w$ where $N_{v_i}(z_2, z_1)$ for $1 \leq i \leq n$ and $w \in L(N_{a_{z_2}z_1})$. Then it suffices to establish $L(N_{a_{z_2}z_1}) \subseteq (ua(\prod_{1 \leq i \leq n} v_i a_i))^{-1}L$ by induction. For $n = 0$ we just apply (RHS). For the inductive case $n + 1$ we combine:

$L(N_{a_{z_2}z_1}) \subseteq (ua(\prod_{1 \leq i \leq n} v_i a_i))^{-1}L$ \quad $L(N_{a_{z_2}z_1}) \subseteq (v_{n+1})^{-1}L(N_{a_{z_2}z_1})$

to infer $L(N_{a_{z_2}z_1}) \subseteq (ua(\prod_{1 \leq i \leq n} v_i a_i)v_{n+1})^{-1}L$ and finally $L(N_{a_{z_2}z_1}) \subseteq (ua(\prod_{1 \leq i \leq n} v_i a_i)v_{n+1}a_i)^{-1}L$ via (RHS).

2. We’ll prove (F). The implication $(\Rightarrow)$ is trivial because $z \in F$ implies $z \in L(N_{a_{z_2}z_1})$. Conversely we’ll use transition-maximality. Assuming (RHS), and given any $u$-path $I \ni z_1 \rightarrow_a z_n = z$ through $N$, since $L(N_{a_{z_2}z_1}) \subseteq u^{-1}L$ we infer $z \in u^{-1}L$ i.e. $u \in L$, so by transition-maximality $z \in F$.

\[\square\]

**Corollary 4.2.14.** If $N$ is transition-maximal it is locally-saturated and intersection-saturated.

**Proof.** Given transition-maximal $N = (I, Z, N_a, I)$ we first show $N$ is locally-saturated. Given $L(N_{a_{z_2}z_1}) \subseteq L(N)$ then $z \in I$ by transition-maximality; the converse is trivial. Concerning transitions, $N_a(z_1, z_2)$ certainly implies $L(N_{a_{z_2}z_1}) \subseteq a^{-1}L(N_{a_{z_2}z_1})$. Conversely if the latter holds, then whenever $L(N_{a_{z_2}z_1}) \subseteq X \subseteq \text{LW}(L)$ we infer $L(N_{a_{z_2}z_1}) \subseteq a^{-1}X$ because $a^{-1}(-)$ is monotonic w.r.t. inclusions. Thus $N_a(z_1, z_2)$ by Lemma 4.2.13, so $N$ is locally-saturated. Finally, $\text{rev}(N)$ is transition-maximal by Lemma 4.2.12 hence locally-saturated, so $N$ is intersection-saturated by Theorem 4.2.9.

\[\square\]

**Corollary 4.2.15.** If $N$ is transition-maximal and union-free then $N \equiv \text{simple}_v(N)$.

**Proof.** By Corollary 4.2.14 $N$ is locally-saturated, so by union-freeness $N \equiv \text{simple}_v(N)$ via Theorem 4.2.8.

\[\square\]

**Lemma 4.2.16.** $\text{simple}_v(-)$ preserves transition-maximality.

**Proof.** Given $N = (I, Z, N_a, F)$ we have the full subset construction $\delta := \text{det}(\text{decp}(N))$, the quotient JSL-dfa $acc_\delta : \delta \rightarrow \text{simple}(\delta)$ and also the irreducible simplification $\text{simple}_v(N)$. If $N_a(z_1, z_2)$ then for completely general reasons $L(N_{a_{z_2}z_1}) \subseteq a^{-1}L(N_{a_{z_2}z_1})$, or equivalently $acc_\delta(\{z_2\}) \subseteq a^{-1}acc_\delta(\{z_1\})$ i.e. $acc_\delta(\{z_1\}) \rightarrow_a acc_\delta(\{z_2\})$ in $\text{simple}_v(N)$.

1. One cannot add an initial state to $\text{simple}_v(N)$ whilst preserving acceptance, because by local saturation (Lemma 4.2.7), any additional state accepts $K \notin L(N)$.

2. For a contradiction suppose adding a final state $K$ to $\text{simple}_v(N)$ preserves acceptance. Then $N' := (I, Z, N_a, F \cup acc_\delta^{-1}(\{K\}))$ accepts $L(N)$ (which is a contradiction) because any additional accepting $N'$-path $I \ni z_0 \rightarrow_{a_0} \cdots \rightarrow_{a_n} z_{n+1} \in acc_\delta^{-1}(\{K\})$ directly induces $I_{\text{simple}_v(N)} \ni acc_\delta(\{z_0\}) \rightarrow_{a_0} \cdots \rightarrow_{a_n} acc_\delta(\{z_{n+1}\}) = K$ in $\text{simple}_v(N)$’s extension.

3. It remains to show no additional transitions can be added. For a contradiction, assume $N'$ obtained by adding a single new transition $X_1 \rightarrow_{a_0} X_2$ to $\text{simple}_v(N)$ satisfies $L(N') = L(\text{simple}_v(N)) = L(N)$. Consider the nfa:

\[
M := (I, Z, M_a, F) \quad M_a := \begin{cases} N_a \cup acc_\delta^{-1}(\{X_1\}) \times acc_\delta^{-1}(\{X_2\}) & \text{if } a = a_0 \\ N_a & \text{otherwise.} \end{cases}
\]

Let us show $M$ has strictly more transitions than $N$. Firstly $Y := acc_\delta^{-1}(\{X_1\}) \times acc_\delta^{-1}(\{X_2\})$ is non-empty because $acc_\delta$ is surjective. Secondly $N_a \cap Y = \emptyset$ for otherwise $X_1 \rightarrow_{a_0} X_2$ would already be in $\text{simple}_v(N)$.

\[55\]
Certainly \(L(N) \subseteq L(M)\). For a contradiction we establish the converse. Given an accepting \(M\)-path shown below left:

\[
I \ni z_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} z_n \in F \quad I_{\text{simple}}(N) \ni \text{acc}_\delta(\{z_0\}) \xrightarrow{a_1} \cdots \xrightarrow{a_n} \text{acc}_\delta(\{z_n\}) \in F_{\text{simple}}(N)
\]

there is a respective accepting \(N^\prime\)-path shown above right. Indeed, if \(N_{a,i+1}(z_i, z_{i+1})\) then \(\text{acc}_\delta(\{z_i\}) \rightarrow_{a_{i+1}} \text{acc}_\delta(\{z_{i+1}\})\) in \(\text{simple}_\delta(N)\) and hence \(N^\prime\). Otherwise \((z_i, z_{i+1}) \in \text{acc}_\delta^{-1}(\{X_1\}) \times \text{acc}_\delta^{-1}(\{X_2\})\) is covered by the single extra transition in \(N^\prime\).

\[
\square
\]

### 4.3 \(L\)-extensions

**Definition 4.3.1** (\(L\)-extension). Recall the transitions of the state-minimal JSL-dfa \(\delta_\alpha(L)\) i.e. \(\gamma_a = \lambda X.a^{-1}X : LQ(L) \rightarrow LQ(L)\) from Definition 3.3.2. An \(L\)-extension \(\epsilon : LQ(L) \rightarrow (\mathbb{T}, \delta_a)\) is an injective JSL\(_f\)-morphism \(\epsilon : LQ(L) \rightarrow \mathbb{T}\) together with \(\mathbb{T}\)-endomorphisms \(\delta_a\) such that \(\epsilon \circ \gamma_a = \delta_a \circ \epsilon\) for each \(a \in \Sigma\).

Then an \(L\)-extension is a join-preserving order-embedding of \(LQ(L)\) into \(\mathbb{T}\). Additionally each endomorphism \(\lambda X.a^{-1}X\) of the former is extended by the endomorphism \(\delta_a\) of the latter.

**Note 4.3.2** (Representation theory). By Theorem 3.7.6 one can view an \(L\)-extension as a representation of \(L\)'s syntactic semiring [Pol01]. Then we are considering the ‘representation theory’ of finite idempotent semirings.

**Example 4.3.3** (\(L\)-extensions).

1. Given \(\gamma_a = \lambda X.a^{-1}X\) we have two bijective \(L\)-extensions:

   \[
   id_{LQ(L)} : LQ(L) \rightarrow (LQ(L), \gamma_a) \quad dr_L^1 : LQ(L) \rightarrow (LQ(L^r)^{\mathcal{op}}, (\gamma_a)\).
   \]

   The second one follows by Theorem 3.3.6. They are essentially the same extension i.e. they are isomorphic when viewed as algebras with \(|\Sigma|\)-many unary operations.

2. \(LQ(\Sigma^*) := (\emptyset, \Sigma^*, \cup, \emptyset)\) where each \(\lambda X.a^{-1}X = id_{LQ(\Sigma^*)}\). Any \(S \in \text{JSL}_f\) has endomorphism:

   \[
   e_{\tau_S} := \lambda X : \begin{cases} 1_S & s = 1_S \\ \tau_S & \text{otherwise.} \end{cases}
   \]

   If \(\Sigma \neq \emptyset\), the number of injective \(\epsilon : LQ(\Sigma^*) \rightarrow S\) is \(|S| - 1\). Each \(\epsilon\) defines an \(L\)-extension \(LQ(\Sigma^*) \rightarrow (S, e_{\tau_S})\).

3. Let \(T\) be a finite union-closed set of languages such that (a) \(L \in T\) and (b) \(X \in T \implies a^{-1}X \in T\) for all \(a \in \Sigma\). Then the inclusion \(\iota : LQ(L) \rightarrow (T, \cup, \emptyset), \lambda X.a^{-1}X\) is an \(L\)-extension.

There is a direct translation from an \(L\)-extension to a JSL-dfa: inherit the initial state and extend the final states of \(\delta_\alpha(L)\) (see below). Conversely each JSL-dfa induces an \(L\)-extension by first simplifying and then forgetting the initial state and final states.

**Definition 4.3.4** (Translation between \(L\)-extensions and JSL-dfas).

1. The induced JSL-dfa of an \(L\)-extension \(\epsilon : LQ(L) \rightarrow (\mathbb{T}, \delta_a)\) is:

   \[
   j\text{dfa}(\epsilon) := (\epsilon(L), \mathbb{T}, \delta_a, \iota_{\mathbb{T}}(\epsilon(dr_L(L^r)))) \quad \text{and accepts } L.
   \]

2. Conversely given any JSL-dfa \(\delta = (s_0, S, \delta_a, F)\) then:

   \[
   \text{lext}(\delta) := \iota : LQ(L) \rightarrow (\mathbb{L}\mathcal{angs}(\delta), \lambda X.a^{-1}X).
   \]

   is its induced \(L(\delta)\)-extension.

**Note 4.3.5**.
1. Concerning \( \text{jdfa}(-) \), the largest non-final state in \( LQ(L) \) is \( \bigcup \{ X \in \text{LW}(L) : \varepsilon \notin L \} = dr_L(L') \). Then by Definition 3.2.1 it is well-defined JSL-dfa. It accepts \( L \) because the embedding \( \varepsilon \) restricts to a dfa-isomorphism from \( \text{dfa}(L) \) i.e. the classical state-minimal dfa which is a sub dfa of \( \text{dfa}(L) \).

2. Concerning \( \text{lext}(-) \), the join-semilattice of accepted language \( \text{langs}(\delta) \) is from Definition 3.4.1. It was used to define the simplification \( \text{simple}(\delta) \) of the JSL-dfa \( \delta \). We remarked in Example 4.3.3.3 that such structures are well-defined \( L \)-extensions.

**Definition 4.3.6** (Simplicity, reachability, transition-maximality, state-minimality). Fix an \( L \)-extension \( e \).

1. \( e \) is simple if \( \text{jdfa}(e) \) is simple (Definition 3.4.1.2). Then \( e \)'s simplification is \( \text{simple}(e) := \text{lext}(\text{jdfa}(e)) \).

2. \( e \) is reachable if the lower nfa of \( \text{Airr}(\text{jdfa}(e)) \) is reachable (Definition 3.1.1).

3. \( e \) is transition-maximal if the lower nfa of \( \text{Airr}(\text{jdfa}(e)) \) is transition-maximal (Definition 4.2.1).

4. \( e \) is state-minimal if the lower nfa of \( \text{Airr}(\text{jdfa}(e)) \) is state-minimal.

To simplify an \( L \)-extension one views it as a JSL-dfa, simplifies it, and finally forgets the initial state and final states. Well-definedness follows because \( \text{jdfa}(e) \) accepts \( L \), so that \( \text{lext}(\text{jdfa}(e)) \) is an \( L \)-extension. The notions of simplicity and simplification are inherited from JSL-dfas, whereas the notions of reachability, transition-maximality and state-minimality are inherited from nfas.

### 4.3.1 Transition-maximal \( L \)-extensions

**Note 4.3.7.** The results in this section are currently not being used elsewhere.

**Lemma 4.3.8** (Reachability degeneracy). Let \( e \) be a simple transition-maximal \( L \)-extension and \( \text{Airr}(\text{jdfa}(e)) = (\mathcal{M}, \mathcal{G}, \mathcal{M}') \). Then \( \mathcal{M} \) has at most one unreachable state, accepting \( \Sigma^* \) if it exists.

**Proof.** By assumption the lower nfa \( \mathcal{M} \) is transition-maximal. Then those states not reachable from an initial state are all final and have transitions to every other state by transition-maximality. Thus they all accept \( \Sigma^* \), so by simplicity there is at most one of them. \( \square \)

We now come to another important notion of ‘maximality’ definable purely in terms of an \( L \)-extension’s structure.

**Definition 4.3.9** (Meet-maximality). An \( L \)-extension \( e : LQ(L) \rightarrow (\overline{T}, \delta_e) \) is meet-maximal if:

\[
\begin{align*}
  j &= \bigwedge_{\overline{\tau}} \{ e(X) : X \in \text{LW}(L), j \leq_{\overline{\tau}} e(X) \} \\
  \delta_e(j) &= \bigwedge_{\overline{\tau}} \{ e(a^{-1}X) : X \in \text{LW}(L), j \leq_{\overline{\tau}} e(X) \}
\end{align*}
\]

for all \( j \in J(\overline{T}) \) and \( a \in \Sigma \).

Then in meet-maximal \( L \)-extensions each \( j \in J(\overline{T}) \) is the meet of those embedded left word quotients of \( L \) above it. Moreover, the endomorphism extensions \( \delta_e : \overline{T} \rightarrow \overline{T} \) preserve these special meets. Importantly, each transition-maximal nfa induces a meet-maximal \( L \)-extension.

**Lemma 4.3.10.** If \( \mathcal{N} \) is a transition-maximal nfa, \( \text{lext}(\text{det}(\text{dep}(\text{reach}(\mathcal{N})))) \) is a simple, reachable and transition-maximal \( L(\mathcal{N}) \)-extension.

**Proof.** Setting \( L := L(\mathcal{N}) \) then the specified \( e \) is an \( L \)-extension because each operation preserves acceptance. It is simple because \( \text{lext}(-) \) first simplifies and then forgets the initial state and final states. Concerning reachability, if \( \mathcal{M} := \text{reach}(\mathcal{N}) \) then the lower nfa of \( \text{Airr}(\text{jdfa}(e)) \) is precisely the irreducible simplification \( \text{simple}_e(\mathcal{M}) \) (Definition 4.2.4) and the latter is reachable by Lemma 4.2.6.3. Concerning transition-maximality, \( \mathcal{M} = \text{reach}(\mathcal{N}) \) is transition-maximal for otherwise \( \mathcal{N} \) wouldn’t be, hence \( \text{simple}_e(\mathcal{M}) \) is transition-maximal by Lemma 4.2.10. \( \square \)

**Theorem 4.3.11** (Meet-maximality). If an \( L \)-extension is simple and transition-maximal it is meet-maximal.
Proof. We may assume \( e \) is simplified i.e. \( e = \text{simple}(e) \). Then it is an inclusion \( e : \mathbb{L}Q(L) \to (\mathbb{T}, \lambda X.x^{-1}X) \) where \( \mathbb{T} = (T, \cup, \emptyset) \) and \( T \) is the set of languages accepted by the individual states of \( \delta := \text{jdfa}(e) \). Let \( \mathcal{M} \) be the lower nfa of \text{Airr}^T which is transition-maximal by assumption, hence locally-saturated by Lemma 4.2.7. Since each \( j \in J(\mathbb{T}) \) accepts \( j \) by Lemma 4.3.2.3, invoking Lemma 4.2.13 yields:

\[
\mathcal{M}_a(j_1, j_2) \iff \forall X \in \mathbb{L}W(L). [j_1 \subseteq X \Rightarrow j_2 \subseteq a^{-1}X].
\]

(T)

Furthermore by Lemma 4.2.13 \( \mathcal{M} \)'s final states are:

\[
F_{\mathcal{M}} = \{ j \in J(\mathbb{T}) : \varepsilon \in \bigcap \{ X \in \mathbb{L}W(L) : j \subseteq X \} \}.
\]

(F)

We're ready to prove meet-maximality, so fix any \( j \in J(\mathbb{T}) \) and let \( M_j := \bigwedge \{ X \in \mathbb{L}W(L) : j \subseteq X \} \). Certainly \( j \in M_j \).

For the reverse inclusion, first observe \( \forall j \in J(\mathbb{T}) \), \( \mathcal{M}_a(j_0, j_2) \Rightarrow \mathcal{M}_a(j, j_2) \) via (T). In particular, given \( j \in X \in \mathbb{L}W(L) \) then \( j_0 \subseteq M_j \subseteq X \) so we deduce \( j_2 \subseteq a^{-1}X \) by assumption.

So every word accepted by \( j_0 \) is accepted by \( j \) i.e. \( j_0 \subseteq j \); moreover \( M_j \subseteq Y \) because \( j_0 \subseteq M \) was arbitrary. Then we've established:

\[
j = \bigwedge \{ X \in \mathbb{L}W(L) : j \subseteq X \} \text{ for each } j \in J(\mathbb{T}).
\]

Fixing any \( a \in \Sigma \) and \( j \in J(\mathbb{T}) \) it remains to establish:

\[
a^{-1}j = \bigwedge \{ a^{-1}X : j \subseteq X \in \mathbb{L}W(L) \}.
\]

Indeed if \( j' \in J(\mathbb{T}) \) lies below the (RHS) i.e. \( \forall X \in \mathbb{L}W(L). [j \subseteq X \Rightarrow j' \subseteq a^{-1}X] \) then by (T) we infer \( \mathcal{M}_a(j, j') \) and hence \( j' \subseteq a^{-1}j \) because \( \mathcal{M} \) is locally-saturated. Finally \( a^{-1}j \) is itself a lower bound for (RHS) because \( a^{-1}(-) \) is monotone w.r.t. inclusion.

Are these special meets of left word quotients \( u^{-1}L \) always their intersection? The answer is \text{no}.

Example 4.3.12 (Meet-maximal meets needn’t be intersections). In [BT14, Theorem 7] a language \( L \) is implicitly provided s.t. if an nfa \( \mathcal{N} \) accepts \( L \) and each \( L(N_{\delta(z)}) \) is a set-theoretic boolean combination of \( \mathbb{L}W(L) \) then \( \mathcal{N} \) is not state-minimal. Given a transition-maximal extension of a state-minimal nfa we obtain a meet-maximal \( L \)-extension by Theorem 4.3.11. If the special meets \( \bigwedge \{ j : j \subseteq X \in \mathbb{L}W(L) \} \) were intersections we’d obtain a contradiction via the lower nfa of \( \text{Airr}(\text{jdfa}(e)) \) – which is also a state-minimal nfa accepting \( L \).

We finally mention some related properties.

Lemma 4.3.13. If \( e : \mathbb{L}Q(L) \to (\mathbb{T}, \delta) \) is transition-maximal and simplified,

\[
j \subseteq u^{-1}L \iff \bigcap \{ X \in \mathbb{L}W(L) : j \subseteq X \} \subseteq u^{-1}L \text{ for any } j \in J(\mathbb{T}) \text{ and } u \in \Sigma^*.
\]

Proof. The implication \((\Rightarrow)\) is immediate. Conversely we know \( e \) is meet-maximal by Theorem 4.3.11, so that \( j = \bigwedge \{ X \in \mathbb{L}W(L) : j \subseteq X \} \subseteq \bigcap \{ X \in \mathbb{L}W(L) : j \subseteq X \} \subseteq u^{-1}L \).

Lemma 4.3.14. If \( e : \mathbb{L}Q(L) \to (\mathbb{T}, \delta) \) is transition-maximal and simplified then for any \( j \in J(\mathbb{T}) \),

\[
j_2 \subseteq a^{-1}j_1 \iff \bigcap \{ X \in \mathbb{L}W(L) : j_2 \subseteq X \} \subseteq \bigcap \{ a^{-1}X \in \mathbb{L}W(L) : j_1 \subseteq X \}.
\]

Proof. We calculate:

\[
j_2 \subseteq a^{-1}j_1 \iff \forall X \in \mathbb{L}W(L). [j_1 \subseteq X \Rightarrow j_2 \subseteq a^{-1}X] \quad \text{(by Theorem 4.3.11)}
\]
\[
\iff \forall X \in \mathbb{L}W(L). [j_1 \subseteq X \Rightarrow \bigcap \{ X \in \mathbb{L}W(L) : j_2 \subseteq X \} \subseteq a^{-1}X] \quad \text{(by Lemma 4.3.13)}
\]
\[
\iff \bigcap \{ X \in \mathbb{L}W(L) : j_2 \subseteq X \} \subseteq \bigcap \{ a^{-1}X \in \mathbb{L}W(L) : j_1 \subseteq X \}.
\]

\( \square \)
4.3.2 Reversing $L$-extensions

Note 4.3.15. The results in this section are currently not being used elsewhere.

Definition 4.3.16 (Reversal of an $L$-extension). Given an $L$-extension $e$ let $\mathcal{N}$ be the lower nfa of $\text{Airr}(\text{jdfa}(e))$. Then $e$’s reversal is the $L’$-extension:

$$\text{rev}(e) = \text{lext}(\text{det}(\text{dep}(\text{rev}(\mathcal{N}))))).$$

It is union-generated by the languages $\text{rev}_e(j) := L(\text{rev}(\mathcal{N})_{\delta_0}) = \{w \in \Sigma^* : j \in \mathcal{N}_{\delta_0}[I_{\mathcal{N}}]\} \text{ where } j \in J(\mathcal{T})$.

Note 4.3.17 (Alternative descriptions of $\text{rev}(e)$).

1. It is $\text{simple}(\text{det}(\text{dep}(\text{rev}(\mathcal{N}))))$ without the initial state or final states.
2. By Corollary 3.4.5, it is isomorphic to $(\text{reach}(\text{det}(\text{dep}(\mathcal{N}))))^*$ without the initial state or final states.

Lemma 4.3.18 (Reversing $L$-extensions). Fix any $L$-extension $e : \mathcal{Q}(L) \Rightarrow (T, \delta_0)$.

1. $\text{rev}(e)$ is a well-defined simplified $L’$-extension.
2. If $e$ is simple then $\text{rev}(e)$ is reachable.
3. If $e$ is transition-maximal then so is $\text{rev}(e)$. Similarly if $e$ is state-minimal then so is $\text{rev}(e)$.
4. If $e$ is simple, transition-maximal and $\text{rev}_e(j_0) = \bigcup \text{rev}_e(j) : j \in J \big)$ for some $j_0 \in J(\mathcal{T})$, $J \subseteq J(\mathcal{T})$ then $j_0 = \bigwedge J$.
5. If $e$ is simplified, transition-maximal and state-minimal then $\text{rev}(\text{rev}(e)) = e$.

Proof.

1. Well-definedness follows by construction, noting that $\text{dep}(-)$, $\text{det}(-)$ and $\text{lext}(-)$ preserve the accepted language $L’$. Likewise $\text{rev}(e)$ is simplified by construction.
2. If $e$ is simple then $\mathcal{N}$ is coreachable because each $j \in J(\mathcal{T})$ accepts a non-empty language. Then $\text{rev}(\mathcal{N})$ is reachable and hence so is $\text{simple}(\text{rev}(\mathcal{N}))$ by Lemma 4.2.6.
3. If $\mathcal{N}$ is transition-maximal then $\text{rev}(\mathcal{N})$ is too by Lemma 4.2.12, hence so is $\text{simple}(\text{rev}(\mathcal{N}))$ by Lemma 4.2.16. If $e$ is state-minimal then the lower nfa of $\text{Airr}(\text{jdfa}(e))$ is a state-minimal nfa $\mathcal{M}$ accepting $L$. Since the lower nfa of $\text{Airr}(\text{jdfa}(\text{rev}(e)))$ accepts $L’$ and has no more states than $\mathcal{M}$ it is also state-minimal, so $\text{rev}(e)$ is state-minimal.
4. We may assume $e = \text{simple}(e)$ is simplified. Let $\delta := \text{jdfa}(e)$ and $\mathcal{N}$ be the lower nfa of $\text{Airr}\delta$. Suppose $\text{rev}_e(j_0) = \bigcup \text{rev}_e(j) : j \in J \big)$ and $\forall \alpha \in \Sigma, \forall j \in J, (j \in \mathcal{N}_\alpha[I_{\mathcal{N}}] \Rightarrow j_0 \in \mathcal{N}_\alpha[I_{\mathcal{N}}])$. Now, since $\mathcal{N}$ is transition-maximal the intersection rule holds by Corollary 4.2.14, so that:

$$\forall j \in J, \forall \alpha \in \Sigma, \mathcal{N}_\alpha[j_0] \subseteq \mathcal{N}_\alpha[j] \Rightarrow$$

because whenever $\mathcal{N}_\alpha(j_0, j’)$ and there is a $u$-path to $j$ there is a $ua$-path to $j’$. Furthermore:

$$j_0 \text{ is final iff every } j \in J \text{ is final.}$$

Indeed, if $j_0$ is final and $j \in J$ then for every $u$-path to $j$ we have a $u$-path to $j_0$ and hence $u \in L$, so by transition-maximality $j$ is final. Similarly, if every $j \in J$ is final then every $u$-path to $j_0$ satisfies $u \in L$ so $j_0$ is final by transition-maximality. Then by (A) and (B) we deduce $j_0 \subseteq \bigwedge J$.

To establish $j_0 = \bigwedge J$ we fix any $j’ \subseteq \bigwedge J$ and prove $j’ \subseteq j_0$. Certainly $\forall j \in J, j’ \subseteq j$ hence:

$$\forall j \in J, \mathcal{N}_\alpha[j’] \subseteq \mathcal{N}_\alpha[j] \Rightarrow$$

because $j” \subseteq a^{-1}j’$ implies $j” \subseteq a^{-1}j$. We now aim to prove $\mathcal{N}_\alpha[j’] \subseteq \mathcal{N}_\alpha[j_0]$. Given $\mathcal{N}_\alpha(j’, j”)$ we certainly know $\forall j \in J, \mathcal{N}_\alpha(j’, j”) \Rightarrow \forall j \in J, \mathcal{N}_\alpha’(j’, j) \subseteq \mathcal{N}_\alpha[j]$. Equivalently $\forall j \in J, \mathcal{N}_\alpha’(j’, j) (\text{in } \text{rev}(\mathcal{N}))$ and thus $\forall j \in J, \mathcal{N}_\alpha’(j’, j) \subseteq a^{-1} \text{rev}_e(j’)$, where the latter uses a general property of nfas. Then:

$$\text{rev}_e(j_0) = \bigcup_{j \in J} \text{rev}_e(j) \subseteq a^{-1} \text{rev}_e(j’).$$
In other words, for every $u^r$-path to $j_0$ in $N$ there exists an $(au)^r$-path to $j''$. Applying the intersection-rule we deduce $N_\alpha(j_0,j'')$ as desired i.e. we have established $N_\alpha[j'] \subseteq N_\alpha[j_0]$. Furthermore if $j'$ is final then every $j \in J$ is final, so that $j_0$ is final by (B). Then we’ve proved that $j' \in j$ and we’re done.

5. An nfa is state-minimal iff its reverse is state-minimal. Then since $rev(e) : L(\emptyset) \to (U, \phi_\alpha)$ accepts $L'$ we deduce $\alpha := \lambda j . rev_\alpha(j) : J(\emptyset) \to J(U)$ is bijective, for otherwise we’d contradict state-minimality. Let $N$ be the lower nfa of $\text{Airr}(\text{dfa}(\alpha))$ and $M$ be the lower nfa of $\text{Airr}(\text{dfa}(\text{rev}(\alpha)))$. Then we can bijectively relabel $M$ to obtain the nfa:

$$M' := (\alpha^{-1}[M], J(\emptyset), M'_\alpha, \alpha^{-1}[F_M])$$

$$M'_\alpha(j_1, j_2) := M_\alpha(\text{rev}(\alpha)(j_1), \text{rev}(\alpha)(j_2)).$$

We’re going to show that $\text{rev}(M')$ extends $N$. By (2) we know $M$ is transition-maximal, hence:

$$\text{(rev}(M'))_\alpha(j_1, j_2) \iff M'_\alpha(j_1, j_2) \iff M_\alpha(\text{rev}(\alpha)(j_1), \text{rev}(\alpha)(j_2))$$

$$\iff rev_\alpha(j_1) \in \alpha^{-1} rev_\alpha(j_2)$$

$$\iff \forall \alpha \in \Sigma^*. \exists [j_1 \in N_\alpha[I_N] \Rightarrow j_2 \in N_\alpha[I_N]].$$

Thus $N(j_1, j_2) \Rightarrow (\text{rev}(M'))_\alpha(j_1, j_2)$ because whenever there is a $u^r$-path to $j_1$ we obtain a $ua$-path to $j_2$. Next, if $j$ is initial in $N$ then $j \in I_N = N_\alpha[I_N]$ and hence $e \in rev_\alpha(j)$, so that $j$ is initial in $M'$, thus initial in $\text{rev}(M')$. Finally if $j$ is final in $N$ then $\forall u \in \Sigma^*, \exists [j \in N_\alpha[I_N] \Rightarrow u \in L]$ or equivalently $\text{rev}_\alpha(j) \subseteq L$, so that $j$ is initial in $M'$, thus final in $\text{rev}(M')$. Having established that $\text{rev}(M')$ extends $N$ we immediately deduce $\text{rev}(M') = N$ by transition-maximality. It follows that:

$$\text{rev}(\text{rev}(\alpha)(j)) = L(\text{rev}(M')_{\alpha \alpha \alpha j})$$

$$= L(\text{rev}(M')_{\alpha j})$$

$$= L(N_{\alpha j})$$

Then we can bijectively relabel $M$ to obtain the nfa:

$$M' := (\alpha^{-1}[M], J(\emptyset), M'_\alpha, \alpha^{-1}[F_M])$$

$$M'_\alpha(j_1, j_2) := M_\alpha(\text{rev}(\alpha)(j_1), \text{rev}(\alpha)(j_2)).$$

We’re going to show that $\text{rev}(M')$ extends $N$. By (2) we know $M$ is transition-maximal, hence:

$$\text{(rev}(M'))_\alpha(j_1, j_2) \iff M'_\alpha(j_1, j_2) \iff M_\alpha(\text{rev}(\alpha)(j_1), \text{rev}(\alpha)(j_2))$$

$$\iff rev_\alpha(j_1) \in \alpha^{-1} rev_\alpha(j_2)$$

$$\iff \forall \alpha \in \Sigma^*. \exists [j_1 \in N_\alpha[I_N] \Rightarrow j_2 \in N_\alpha[I_N]].$$

Thus $N(j_1, j_2) \Rightarrow (\text{rev}(M'))_\alpha(j_1, j_2)$ because whenever there is a $u^r$-path to $j_1$ we obtain a $ua$-path to $j_2$. Next, if $j$ is initial in $N$ then $j \in I_N = N_\alpha[I_N]$ and hence $e \in rev_\alpha(j)$, so that $j$ is final in $M'$, thus initial in $\text{rev}(M')$. Finally if $j$ is final in $N$ then $\forall u \in \Sigma^*, \exists [j \in N_\alpha[I_N] \Rightarrow u \in L]$ or equivalently $\text{rev}_\alpha(j) \subseteq L$, so that $j$ is initial in $M'$, thus final in $\text{rev}(M')$. Having established that $\text{rev}(M')$ extends $N$ we immediately deduce $\text{rev}(M') = N$ by transition-maximality. It follows that:

$$\text{rev}(\text{rev}(\alpha)(j)) = L(\text{rev}(M')_{\alpha \alpha \alpha j})$$

$$= L(\text{rev}(M')_{\alpha j})$$

$$= L(N_{\alpha j})$$

Since $\alpha$ is bijective we know every $j \in J(\emptyset)$ has a unique corresponding $\text{rev}_\alpha(j) \in J(U)$. Combining this with the above equality we deduce $\text{rev}(\text{rev}(\alpha)) = e$. □

4.4 The Atomizer

This section is based on recent work of Tamm [Tam16]. Recall the minimal boolean and distributive JSL-dfa from Definition 3.5.1. Fixing $L$, the left predicates $LP(L)$ are those finitely many languages arising as a set-theoretic boolean combination of the left word quotients $LW(L)$. Importantly, any language can be transformed into a left predicate via a closure operator.

**Definition 4.4.1 (Atomic languages, $cl_L$ and $E_L$).**

1. The **atomic closure operator** $cl_L : \mathcal{P} \Sigma^* \to \mathcal{P} \Sigma^*$ is defined:

$$cl_L(X) := \bigcup \{ \alpha \in J(LP(L)) : \alpha \cap X \neq \emptyset \}$$

$$= \bigcap \{ Y \in LP(L) : X \subseteq Y \}.$$

Moreover the equivalence relation $E_L \subseteq \Sigma^* \times \Sigma^*$ is defined:

$$E_L(u,v) : \iff \forall X \in LW(L). [u \in X \iff v \in X]$$

with equivalence classes $[u]_{E_L} \subseteq \Sigma^*$.

2. A language is **atomic** w.r.t $L$ [BT14] if it is a fixpoint of $cl_L$. They are precisely the languages in $LP(L)$

3. A language is **positively atomic** w.r.t $L$ if it lies in $LD(L) \subseteq LP(L)$.

4. A language is **subatomic** w.r.t $L$ if it lies in $LRF(L)$.
**Note 4.4.2** (Compatible definitions of $\text{cl}_L$). The distinct definitions of $\text{cl}_L(X)$ are consistent: each element of $\text{LP}(L)$ is (i) the join (union) of join-irreducibles (atoms) below it, (ii) the meet (intersection) of elements above it.

**Lemma 4.4.3** (Concerning atomic closure). Let $\alpha \in J(\text{LP}(L))$ and $X, Y \subseteq \Sigma^*$.

1. $\text{cl}_L$ is a well-defined closure operator.
2. $\text{cl}_L(X)$ is the smallest left predicate containing $X$.
3. $\alpha \subseteq \text{cl}_L(X)$ iff $\alpha \cap X \neq \emptyset$.
4. $\text{cl}_L(X \cup Y) = \text{cl}_L(X) \cup \text{cl}_L(Y)$.
5. $\text{cl}_L(w^{-1}X) \subseteq w^{-1}\text{cl}_L(X)$ for all $w \in \Sigma^*$.
6. $J(\text{LP}(L)) = \{\llbracket w \rrbracket_{\text{cl}_L} : w \in \Sigma^*\}$.
7. $\text{cl}_L(X) = \bigcup_{w \in X} \llbracket w \rrbracket_{\text{cl}_L}$.

**Proof.**
1. $\text{cl}_L$ is monotone: if $X \subseteq Y$ then $\forall \alpha \in J(\text{LP}(L)) (\alpha \cap X \neq \emptyset \Rightarrow \alpha \cap Y \neq \emptyset)$ hence $\text{cl}_L(X) \subseteq \text{cl}_L(Y)$. Next, $X \subseteq \text{cl}_L(X)$ because the latter is an intersection of supersets of $X$. Finally, $\text{cl}_L \circ \text{cl}_L(X) = \text{cl}_L(X)$ because $\text{cl}_L(X) \in \text{LP}(L)$ is the union of the atoms it includes.
2. Follows by alternate definition.
3. If $\alpha \cap X \neq \emptyset$ then $\alpha \subseteq \text{cl}_L(X)$ by definition. Conversely if $\alpha \subseteq \text{cl}_L(X)$ then some $u \in \alpha$ satisfies $u \in X$ for otherwise we’d know $X \subseteq \overline{\alpha}$ (the latter being a coatom), so that $\text{cl}_L(X) \subseteq \overline{\alpha}$ by the alternate definition (a contradiction).
4. The inclusion ($\supseteq$) follows by monotonicity. Conversely, given an atom $\alpha \subseteq \text{cl}_L(X \cup Y)$ then by (3) there exists $u \in \alpha \cap (X \cup Y)$ and hence w.l.o.g. $u \in \alpha \cap X$ and thus $\alpha \subseteq \text{cl}_L(X)$.
5. Since $X \subseteq \text{cl}_L(X)$ we deduce $w^{-1}X \subseteq w^{-1}\text{cl}_L(X)$ and hence $\text{cl}_L(w^{-1}X) \subseteq w^{-1}\text{cl}_L(X)$ because (a) the former is the least atomic language above $w^{-1}X$, (b) the latter is in $\text{LP}(L)$ because $w^{-1}(\cdot)$ preserves all set-theoretic boolean operations.
6. An atom amounts to specifying $X$ or $\overline{X}$ for each $X \in \text{lw}(L)$ i.e. an $\text{cl}_L$ equivalence-class.
7. Follows by definition via (6).

**Note 4.4.4** ($\text{cl}_L$ preserves unions). The fixpoints (closed sets) of every closure operator are closed under intersections. By Lemma 4.4.3 $\text{LP}(L)$ is also closed under unions, which is not a general property of closure operators. Then the closed sets form a distributive lattice, in fact a boolean lattice because $\text{LP}(L)$ is closed under relative complement.

We’ve now arrived at the main definition of this section.

**Definition 4.4.5** (Atomizer). Each $L$-extension $e : \text{LQ}(L) \Rightarrow (\top, \delta_e)$ has associated join-semilattice morphism:

$$\lambda Y. \text{cl}_L(\text{acc}_e(\text{cl}_L(Y))) : \top \rightarrow \text{LP}(L).$$

Restricting to the image yields the **atomizer** $\text{at}_e : \top \rightarrow \text{Ak}_e$ where $\text{Ak}_e := (\text{At}_e, \cup, \emptyset)$ is the atomized semilattice.

**Note 4.4.6** (Atomizer’s action). The atomizer constructs the closure of the accepted language. We often construct $L$-extensions by simplifying a JSL-dfa, in which case the atomizer is a domain/codomain restriction of $\text{cl}_L$.

**Lemma 4.4.7.** The atomizer is a well-defined join-semilattice morphism.

**Proof.** Fixing an $L$-extension $e : \text{LQ}(L) \Rightarrow (\top, \delta_e)$, $\text{at}_e$ is a well-defined function because $\text{cl}_L(X) \in \text{LP}(L)$. Given $\delta := \text{jdfa}(e)$ then $\text{acc}_\delta : \top \rightarrow \text{long}(\delta)$ is a well-defined surjective JSL$_f$-morphism by Definition 3.4.1. Finally, $\text{cl}_L$ restricts to a morphism $\text{cl}_L : \text{long}(\delta) \rightarrow \text{LP}(L)$ because $\emptyset = \text{cl}_L(\emptyset)$ is atomic and $\text{cl}_L$ preserves binary unions by Lemma 4.4.3.4.

61
Recall the canonical quotient-atom bijection $\kappa_L$ from Theorem 3.5.5. We now use it to represent each atomized semilattice inside Dep.

**Definition 4.4.8** (Atomizer relation $H_e$). Each L-extension $e : \mathbb{L}Q(L) \to (T, \delta_e)$ has an associated atomizer relation,

$$H_e \subseteq J(\mathbb{T}) \times \mathbb{L}W(L^r) \quad H_e(j, v^{-1}L^r) : \iff v^r \in \text{at}_e(j).$$

Furthermore if $e$ is simplified this becomes $H_e(j, v^{-1}L^r) \iff v^r \in \text{cl}_L(j)$. 

This important concept is preserved under simplification of the L-extension.

**Lemma 4.4.9** ($H_e \cong H_{\text{simple}(e)}$). We have the Dep-isomorphism:

$$\begin{array}{cccc}
\mathbb{L}W(L^r) & \overset{\Delta_{jdf}(L)}{\longrightarrow} & \mathbb{L}W(L^r) \\
\mathbb{L}W(L^r) & \overset{\Delta_{jdf}(L)}{\longrightarrow} & \mathbb{L}W(L^r)
\end{array}$$

**Proof.** The diagram commutes by unwinding the definitions, recalling each state $Y$ in $\text{simple}(\text{jdf}(e))$ accepts $Y$. Since $\text{Open}H_e = \text{Open}H_{\text{simple}(e)}$, applying $\text{Open}$ yields an identity morphism (see Note 2.2.11), so this Dep-morphism is actually an isomorphism. \qed

**Theorem 4.4.10** ($\mathbb{A}_e \cong \text{Open}H_e$). For any L-extension $e$ we have the join-semilattice isomorphism:

$$\theta_e : \mathbb{A}_e \to \text{Open}H_e \quad \theta_e(Y) := \{w^{-1}L^r : w \in Y^r\} \quad \theta_e^{-1}(S) := \{w \in \Sigma^r : w^{-1}L^r \in S\}.$$

**Proof.** Recall that the atomized semilattice $\mathbb{A}_e$ is a sub join-semilattice of $\mathbb{L}P(L)$. Concerning the latter, we may instantiate Proposition 2.2.21 with $J_{\mathbb{L}P(L)} := \mathbb{L}P(L)$ (every element) and $M_{\mathbb{L}P(L)} := M(\mathbb{L}P(L))$ (the coatoms). We immediately obtain the Dep-isomorphism:

$$\begin{array}{cccc}
J_{\mathbb{L}P(L)} & \overset{\text{repl}_{\mathbb{L}P(L)}}{\longrightarrow} & \text{OpenPirr}_\mathbb{L}P(L) & \overset{\text{OpenPirr}_{\mathbb{L}P(L)}}{\longrightarrow} & \text{OpenG}
\end{array}$$

and thus the composite join-semilattice isomorphism:

$$\alpha := \begin{array}{cccc}
\mathbb{L}P(L) & \overset{\text{repl}_{\mathbb{L}P(L)}}{\longrightarrow} & \text{OpenPirr}_\mathbb{L}P(L) & \overset{\text{OpenPirr}_{\mathbb{L}P(L)}}{\longrightarrow} & \text{OpenG}
\end{array}$$

To clarify, $\alpha$ acts as $\text{repl}_{\mathbb{L}P(L)}$ because each $Y \in O(\text{Pirr}_\mathbb{L}P(L))$ is downwards-closed in $M(\mathbb{L}P(L))$ w.r.t. inclusion, and $(T_{\mathbb{L}P(L)};Y)$ constructs the downwards-closure (see Proposition 2.2.21). It follows that $\mathbb{A}_e \subseteq \mathbb{L}P(L)$ may be represented as a sub join-semilattice of $\text{OpenG}$. Since $\text{at}_e$ is surjective we know $J := \text{at}_e[J(\mathbb{T})]$ join-generates the atomized semilattice $\mathbb{A}_e$. Then:

$$\alpha \text{ restricts to the isomorphism } \beta : \mathbb{A}_e \to \text{OpenH} \quad \text{where } H := \text{at}_e[J(\mathbb{T})].$$

To explain, each $\text{cl}_L(j) \in J(\mathbb{A}_e) \subseteq J_{\mathbb{L}P(L)}$ satisfies $\text{repl}_{\mathbb{L}P(L)}(\text{cl}_L(j)) = \mathbb{G}[\text{cl}_L(j)],$ and all other open sets are unions of them. To construct the desired isomorphism $\theta$, recall the bijection $\kappa_L : \mathbb{L}W(L^r) \to J(\mathbb{L}P(L))$ from Theorem 3.5.5, and the bijection $J(\mathbb{L}P(L)) \to M(\mathbb{L}P(L))$ between atoms and coatoms (relative complement). Consider the relations:

$$\begin{array}{cccc}
M(\mathbb{L}Q(L)) & \overset{f}{\longrightarrow} & \mathbb{L}W(L^r) \\
J(\mathbb{T}) & \overset{\Delta_{jdf}(L)}{\longrightarrow} & J(\mathbb{T})
\end{array}$$

where the composite bijection $f$ has action $\overline{[w]_{\mathbb{E}_L}} \to \kappa^r_L(\overline{[w]_{\mathbb{E}_L}}) = (w^r)^{-1}L^r$. If they commute we have a Dep-isomorphism because the lower and upper witnesses are bijections. Then let us calculate:

$$\begin{array}{cccc}
\text{H;f}(j, v^{-1}L^r) & \iff & 3w \in \Sigma^r, [\text{at}_e(j) \notin \overline{[w]_{\mathbb{E}_L}} \land f(\overline{[w]_{\mathbb{E}_L}}) = v^{-1}L^r] \\
\iff & \text{at}_e(j) \notin \overline{[v^r]_{\mathbb{E}_L}} \quad \text{atom vs. coatom} \\
\iff & [v^r]_{\mathbb{E}_L} \notin \text{at}_e(j) \quad \text{via Lemma 4.4.3.7} \\
\iff & v^r \in \text{at}_e(j) \quad \text{by definition} \\
\iff & \text{H}_e(j, v^{-1}L^r) \quad \text{(A)}
\end{array}$$

62
Concerning (A), \((\Rightarrow)\) follows because we know \((w^r)^{-1}L^r = v^{-1}L^r\) and thus \([w]_{E_L} = [v^r]_{E_L}\) because \(\kappa_L\) is injective. Conversely \((\Leftarrow)\) follows by choosing \(w := v^r\). So we have the isomorphism \(\theta_e := \text{Open}_e \circ \beta\) with action:

\[
Y \mapsto f[\text{rep}_{L,\beta}(Y)] \quad \text{(by Note 2.2.11)}
\]

\(= \{f([w]_{E_L}) : Y \notin [w]_{E_L}\} \quad \text{(def. of rep)}
\]

\(= \{[w^r]^{-1}L^r : Y \notin [w]_{E_L}\} \quad \text{(def. of } f\text{)}
\]

\(= \{[w^r]^{-1}L^r : w \in Y\} \quad (Y \text{ is atomic}).
\]

\(\square\)

### 4.5 Explaining Kameda-Weiner

Recall the notion of \(L\)-covering \(H\) i.e. Definition 4.1.1. They amount to biclique edge-coverings of the dependency relation \(\mathcal{D}_R_L\). They are legitimate if their induced nfa \(N_H\) (defined over the bicliques) accepts \(L\). Crucially \(H_e\) is a legitimate \(L\)-covering for any \(L\)-extension \(e\).

**Theorem 4.5.1.** \(H_e\) is a legitimate \(L\)-covering for any \(L\)-extension \(e\),

\[
\langle L, H_e \rangle := H_e = \mathcal{D}_R_L
\]

where \(\langle L, H_e \rangle-(u^{-1}L, j) \iff \text{acc}_{jdfa}(e)(j) \subseteq u^{-1}L\).

**Proof.** Denote the acceptance map \(\alpha := \text{acc}_{jdfa}(e)\) for brevity. Observe \(\alpha(j) \subseteq u^{-1}L \iff \text{at}_e(j) \subseteq u^{-1}L\) because \(u^{-1}L\) is atomic w.r.t. \(L\). We first compute \(\langle L, H_e \rangle_\ast\) without knowing \(H_e\) is an \(L\)-covering. Afterwards we’ll verify the claimed equality.

\[
\langle L, H_e \rangle-(u^{-1}L, j) : \iff H_e[j] \subseteq \mathcal{D}_R_L[u^{-1}L]
\]

\(\iff \forall v \in \Sigma^*, [v^r]_{E_L}(j) \Rightarrow uv^r \in L\)

\(\iff \forall v \in \Sigma^*, [v]_{E_L}(j) \Rightarrow v \in u^{-1}L\)

\(\iff \text{at}_e(j) \subseteq u^{-1}L\)

\(\iff \alpha(j) \subseteq u^{-1}L\)

(see above).

\[
\langle L, H_e \rangle-(u^{-1}L, v^{-1}L') : \iff \exists j \in J(\mathcal{F})([\alpha(j) \subseteq u^{-1}L \land v^r \in \text{at}_e(j)]) \quad \text{(by def.)}
\]

\(\iff \exists j \in J(\mathcal{F})([\text{at}_e(j) \subseteq u^{-1}L \land v^r \in \text{at}_e(j)]) \quad \text{(see above)}
\]

\(\iff v^r \in u^{-1}L\)

(A)

\(\iff \mathcal{D}_R_L(u^{-1}L, v^{-1}L').
\]

Concerning (A), \((\Rightarrow)\) follows immediately. As for \((\Leftarrow)\), since \(u^{-1}L = \cup \alpha[S]\) for some \(S \subseteq J(\mathcal{T})\) we deduce \(u^{-1}L = \cup \text{at}_e[S]\), so that \(v^r \in u^{-1}L\) implies \(v^r \in \text{at}_e(j) \subseteq u^{-1}L\) for some \(j \in J(\mathcal{T})\). Then \(H_e\) is an \(L\)-covering by Lemma 4.1.4.1.

It remains to establish the legitimacy of \(H_e\). By Lemma 4.1.4.6 we at least know \(L(N_{H_e}) \subseteq L\), and it remains to prove the reverse inclusion. First let \(M\) be the lower nfa of \(\text{Airr}(jdfa(e))\), which accepts \(L\) by Note 3.2.10. These two nfas have the same states \(J(\text{longs}(e))\); concerning their transitions:

\[
M_\alpha(\alpha(j_1), \alpha(j_2)) \iff \alpha(j_2) \subseteq a^{-1}\alpha(j_1)
\]

\(\iff \forall X \in \mathcal{L}(L).[\alpha(j_1) \subseteq X \Rightarrow \alpha(j_2) \subseteq a^{-1}X]\)

\(\iff \forall X \in \mathcal{L}(L).[\alpha(j_1) \subseteq X \Rightarrow \alpha(j_2) \subseteq \text{at}_e(a)]\)

(see \(\langle L, H_e \rangle_\ast\) above).

Moreover (a) \(I_M = I_{N_{H_e}}\) since \(\alpha(j) \subseteq L \iff \langle L, H_e \rangle-(L, \alpha(j))\) and (b) \(F_M \subseteq F_{N_{H_e}}\) because \(\varepsilon \in \alpha(j) \iff \varepsilon \in \text{at}_e(j) \iff \langle L, H_e \rangle-(L, \varepsilon^r)\). It follows that \(N_{H_e}\) simulates \(M\) i.e. \(L \subseteq L(N_{H_e})\) and we are done.

\(\square\)

**Corollary 4.5.2** (Maximal legitimate \(L\)-coverings). \(H_{e^o}\) is a maximal legitimate \(L\)-covering.

**Proof.** By Theorem 4.5.1 we know \(H_e\) is a legitimate \(L\)-covering. Then \(H_{e^o}\) is a maximal \(L\)-covering by Lemma 4.1.4.7.c and legitimate by Lemma 4.1.4.7.e.

**Corollary 4.5.3.** If \(e\) is simplified and transition-maximal then \(N_{H_e}\) is the lower nfa of \(\text{Airr}(jdfa(e))\).

**Proof.** Let \(M\) be the lower nfa of \(\text{Airr}(jdfa(e))\). In the proof of Theorem 4.5.1 we showed \(N_{H_e}\) is an extension of \(M\). Then by transition-maximality \(M = N_{H_e}\).

\[63\]
Each nfa canonically induces a legitimate $L$-covering – a pattern which the Kameda-Weiner algorithm can recognise. Moreover every transition-maximal union-free nfa (see Theorem 4.2.8) arises as an induced nfa.\footnote{However, induced nfas needn’t be transition-maximal nor union-free.}

**Corollary 4.5.4.** Fix any nfa $\mathcal{N}$ accepting $L$.

1. $\mathcal{H}_{\text{ext}(\mathcal{N})}$ is a legitimate $L$-covering.

2. If $\mathcal{N}$ is transition-maximal and union-free then $\mathcal{N} \cong \mathcal{N}_{\mathcal{H}_{\text{ext}(\mathcal{N})}}$.

**Proof.**

1. The full subset construction $\mathcal{H}(\mathcal{N}) = \text{Det}(\text{dep}(\mathcal{N}))$ accepts $L$ by Note 3.2.10, so the well-defined $L$-extension $\text{ext}(\mathcal{N})$ accepts $L$ by Note 4.3.5. Then the claim follows by Theorem 4.5.1.

2. Since $\mathcal{N}$ is transition-maximal and union-free it is isomorphic to $\text{simple}_\alpha(\mathcal{N})$ by Corollary 4.2.15. Then $\mathcal{N}$ is the isomorphic to the lower nfa of $\text{Airr}(\text{simple}(\mathcal{N}))$, so the claim follows by Corollary 4.5.3.

\[\square\]

### 4.6 Atomic nfas and $L$-extensions

Fixing any regular language $L$, there are finitely languages arising as a union of the atoms $J(L\mathcal{P}(L))$. Recall that these languages are called **atomic** (Definition 4.4.1). Likewise there are **positively atomic** languages (a subclass of the atomic ones) and also the **subatomic** languages (a superclass of the atomic ones).

**Definition 4.6.1** (Atomic, positively atomic and subatomic nfas and $L$-extensions). Fix an nfa $\mathcal{N}$ accepting $L$.

1. $\mathcal{N}$ is **atomic** if each state accepts an atomic language (equiv. $\mathcal{N}$ contains $(L(P(L)))$.

2. $\mathcal{N}$ is **positively atomic** if each state accepts a positively atomic language (equiv. $\mathcal{N}$ contains $(L(D(L)))$.

3. $\mathcal{N}$ is **subatomic** if each individual state accepts a subatomic language (equiv. $\mathcal{N}$ contains $(L(RP(L)))$.

Finally, an $L$-extension $e$ is **atomic** (resp. positively atomic, subatomic) if the lower nfa of $\text{Airr}(\text{jdfa}(e))$ is atomic (resp. positively atomic, subatomic).

**Lemma 4.6.2** (Atomic $L$-extensions). The following statements concerning $L$-extensions are equivalent.

1. $e$ is atomic.

2. $\text{longs}(\text{jdfa}(e)) \subseteq L(P(L))$.

3. $\text{simple}(\text{jdfa}(e))$ is a sub $JSL$-dfas of $\text{dfa}_\alpha(L)$.

4. $\text{at}_e$ defines a surjective $JSL$-dfa morphism to a sub $JSL$-dfa of $\text{dfa}_\alpha(L)$.

**Proof.**

- (1) $\iff$ (2): By Corollary 3.2.11 the languages accepted by the lower nfa (varying over subsets) are precisely those accepted by $\text{jdfa}(e)$ (varying over individual states).

- (2) $\iff$ (3): Follows because the transition structure of the two $JSL$-dfas is defined in the same way.

- (2) $\implies$ (4): We know each state of $\text{jdfa}(e)$ accepts an atomic language, so $\text{at}_e$ acts in the same way as the $JSL$-dfa morphism $\text{at}_e\text{jdafa}(e)$. Then $\text{at}_e$ defines a $JSL$-dfa morphism to a sub $JSL$-dfa of $\text{dfa}_\alpha(L)$.

- (4) $\implies$ (2): The dfa morphism informs us that each state accepts an atomic language.

\[\square\]

**Definition 4.6.3** (Pseudo-atomicity). An $L$-extension $e : L\mathcal{P}(L) \rightarrow (T, \delta_a)$ is **pseudo-atomic** if the kernel of the atomizer $\text{ker}(\text{at}_e) \subseteq T \times T$ is closed under $\lambda(x, y), (\delta_a(x), \delta_a(y))$ for each $a \in \Sigma$.
Then ε is pseudo-atomic if the join-semilattice congruence ker(αε) ⊆ T × T is also a congruence for each unary operation δα : T → T. We’re going to show that atomicity and pseudo-atomicity are equivalent concepts.

Lemma 4.6.4 (Pseudo-atomic L-extensions).
1. Every atomic L-extension is pseudo-atomic.
2. simple(–) preserves pseudo-atomicity.
3. ε : LQ(L) → (T, δα) is pseudo-atomic iff the atomized semilattice admits the L-extension structure:

Proof.
1. By Lemma 4.6.2.4 the dfa morphism αε satisfies αε(δα(t)) = δα(αε(t)) for each α ∈ Σ. The latter implies ε is pseudo-atomic.
2. Let δ := jdfa(ε) and ε0 := simple(ε) recalling Definition 4.3.6. Recalling Lemma 3.4.2.3, αε0(accδ(t)) = clL(accδ(t)) = αε(t) for every t ∈ T. Then given any Y = accδ(t),

\[
\begin{align*}
\alphaε0 (Y1) = \alphaε0 (Y2) & \implies \alphaε(t1) = \alphaε(t2) \\
& \implies \alphaε0(\deltaε(t1)) = \alphaε0(\deltaε(t2)) \\
& \implies \text{cl}_L(\alpha\deltaε\deltaε(t1)) = \text{cl}_L(\alpha\deltaε\deltaε(t2)) \\
& \implies \text{cl}_L(\alpha^{-1}(\alpha\deltaε\deltaε(t1))) = \text{cl}_L(\alpha^{-1}(\alpha\deltaε\deltaε(t2))) \\
& \implies \text{cl}_L(\alpha^{-1}Y1) = \text{cl}_L(\alpha^{-1}Y2) \\
& \implies \alphaε0(\alpha^{-1}Y1) = \alphaε0(\text{cl}_L(\alpha^{-1}Y2)).
\end{align*}
\]

Hence the simplification of ε is also pseudo-atomic.
3. If ε is a well-defined L-extension then whenever αε(t1) = αε(t2) we deduce αε(δε(t1)) = φα(αε(t1)) = φα(αε(t2)) = αε(δε(t2)), so that ε is pseudo-atomic. Conversely, αε is stable under each δε so the endo-morphisms φα : Aε → Aε are well-defined. Since δ := jdfa(ε) accepts L, by varying the initial state it accepts every Y ∈ LQ(L) ⊆ LP(L), so the inclusion ε : LQ(L) → Aε is a well-defined join-semilattice morphism. Observe that each Y ∈ LQ(L) has some tY ∈ T with αε(tY) = Y. Then the calculation:

\[
\begin{align*}
\phiα(ε(Y)) &= \phiα(Y) \\
&= \phiα(αε(tY)) \\
&= \alphaε(δε(tY)) \\
&= \text{cl}_L(\alpha jun\text{ DFA}(ε))(δε(tY)) \\
&= \text{cl}_L(\alpha^{-1}(\alpha\text{ DFA}(ε)(tY))) \\
&= \alpha^{-1}Y \\
&= ε(\alpha^{-1}Y)
\end{align*}
\]

establishes that ε is an L-extension.

\[\qed\]

Theorem 4.6.5. An L-extension is atomic if it is pseudo-atomic.

Proof. If an L-extension is atomic it is pseudo-atomic by Lemma 4.6.4. Conversely given a pseudo-atomic L-extension ε : LQ(L) → (T, δα) then ε0 := simple(ε) is pseudo-atomic by Lemma 4.6.4.2. By Lemma 4.6.4.3 we have the L-extension ε0 : LQ(L) → (Aε, φα), where φα := λX.αX.X : Aε → Aε, since clL(αX.X) = αX.X. Then ε0 is simplified and each state X accepts X ∈ Aε, so ε is atomic.

Tamm and Brzozowski proved an nfa N is atomic iff rev(N)’s reachable subset construction is state-minimal [BT14]. We reprove their result using our terminology and then:

– refine their result i.e. N is positively atomic (see Definition 4.4.1.3) iff the dfa isomorphism is also an order isomorphism.
– generalise their result i.e. $\mathcal{N}$ is subatomic (see Definition 4.4.1.4) iff $\text{rsc}(\text{rev}(\mathcal{N}))$’s transition monoid is suitably isomorphic to $L'$’s syntactic monoid.

**Theorem 4.6.6** (Atomicity and $\text{rsc}(\text{rev}(\mathcal{N}))$). Let $\mathcal{N}$ be an nfa accepting $L$.

1. $\mathcal{N}$ is atomic iff $\text{rsc}(\text{rev}(\mathcal{N})) \cong \text{dfa}(L')$ [BT14].

2. $\mathcal{N}$ is positively atomic iff the dfa isomorphism from (1) is also an order isomorphism w.r.t. inclusion.

3. $\mathcal{N}$ is subatomic iff $\text{rsc}(\text{rev}(\mathcal{N}))$’s transition monoid is isomorphic to $L'$’s syntactic monoid via:

$$\lambda[u]_{\text{TM}(\text{rsc}(\text{rev}(\mathcal{N})))}, \lambda[u]_{\delta_{L'}} : \text{TM}(\text{rsc}(\text{rev}(\mathcal{N}))) \to \text{Syn}(L').$$

**Proof.**

1. Let $\delta := \text{sc}(\text{dfa}(L'))$ be the dual of $\text{dfa}_{\alpha}(L)$ – see Corollary 3.5.8.

Assuming $\mathcal{N}$ is an atomic nfa, we have the composite JSL-dfa morphism:

$$\text{reach}(\delta) \delta \phi \text{reach}(\text{sc}(\text{rev}(\mathcal{N}))).$$

The surjection $q$ arises by dualising $\iota : \text{simple}(\text{sc}(\mathcal{N})) \to \text{dfa}_{\alpha}(L)$ (see Lemma 4.6.2) and applying Corollary 3.4.5 and Corollary 3.5.8. Viewing $\text{reach}(\delta)$ as its underlying dfa, consider its classically reachable part:

$$\text{reach}(\text{reach}(\delta)) = \text{reach}(\delta) = \text{reach}(\text{sc}(\text{dfa}(L'))) = \text{rsc}(\text{rev}(\text{rev}(\text{dfa}(L')))) = \text{rsc}(\text{dfa}(L')) \cong \text{dfa}(L').$$

The above observation provides the injective dfa morphism $\psi$ below:

$$\text{dfa}(L') \psi \text{reach}(\delta) \phi \text{reach}(\text{sc}(\text{rev}(\mathcal{N}))).$$

Since $\text{dfa}(L')$ is state-minimal, the composite dfa morphism $\chi$ is injective. Since $\text{dfa}(L')$ is reachable we obtain the dfa isomorphism $\text{dfa}(L') \cong \text{reach}(\text{reach}(\text{sc}(\text{rev}(\mathcal{N})))) = \text{rsc}(\text{rev}(\mathcal{N}))$ recalling Note 3.2.10.

Conversely fix an nfa $\mathcal{N}$ such that $\text{dfa}(L') \cong \text{rsc}(\text{rev}(\mathcal{N})).$ By definition of $\text{reach}(\delta)$ and Note 3.2.10,

$$\text{reach}(\text{reach}(\text{sc}(\text{rev}(\mathcal{N})))) = \text{reach}(\text{sc}(\text{rev}(\mathcal{N}))) = \text{rsc}(\text{rev}(\mathcal{N})).$$

Then by definition of $\text{reach}(\delta)$ we have an injective dfa morphism $\chi : \text{dfa}(L') \to \text{reach}(\text{sc}(\text{rev}(\mathcal{N}))).$ By taking the free JSL-dfa on a dfa (Theorem 3.2.17) this extends to a JSL-dfa morphism $\tilde{\chi} : \delta \to \text{reach}(\text{sc}(\text{rev}(\mathcal{N}))).$

Applying duality, Corollary 3.5.8 and Corollary 3.4.5 we obtain a JSL-dfa morphism $\text{sc}(\text{sc}(\mathcal{N})) \to \text{dfa}_{\alpha}(L).$ Then every language accepted by $\mathcal{N}$ is atomic, so that $\mathcal{N}$ is itself atomic.

2. Let $\delta := \text{det}(\text{dfa}(L'), \leq, \text{rev}(\text{dfa}(L'))) = \text{sc}(\text{dfa}(L'))$ be the dual of $\text{dfa}_{\alpha}(L)$ – see Corollary 3.5.13.

Assuming $\mathcal{N}$ is positively atomic, we have the composite JSL-dfa morphism:

$$\text{reach}(\delta) \delta \phi \text{reach}(\text{sc}(\text{rev}(\mathcal{N}))).$$

The surjection $q$ arises by dualising $\iota : \text{simple}(\text{sc}(\mathcal{N})) \to \text{dfa}_{\alpha}(L)$ and applying Corollary 3.4.5 and Corollary 3.5.13. Repeating the argument from (1) we obtain the injective dfa morphism $\psi$ below:

$$\text{dfa}(L') \psi \text{reach}(\delta) \phi \text{reach}(\text{sc}(\text{rev}(\mathcal{N}))).$$
Again repeating the argument in (1), we obtain the dfa isomorphism \( \text{dfa}(L') \cong \text{rsc}(\text{rev}(N)) \). To see it is an order-isomorphism w.r.t. inclusion, first observe \( \psi \) has action \( \lambda u^{-1} L'. \{ Y \in \text{lw}(L') : Y \subseteq u^{-1} L' \} \) so it preserves/reflects the inclusion ordering. Finally, \( \phi \) certainly preserves inclusions since it is join-semilattice morphism. It reflects inclusions when restricted to \( \psi(\text{dfa}(L')) \) because simplicity forbids additional inclusions.

Conversely, fix an nfa \( N \) such that \( \text{dfa}(L') \cong \text{rsc}(\text{rev}(N)) \) where this isomorphism also preserves and reflects inclusions. Repeating (1) yields the injective dfa morphism \( \chi : \text{dfa}(L') \rightarrow \text{reach}(\text{sc}(\text{rev}(N))) \), additionally preserving inclusions. In fact, \( \chi \) is an ordered dfa morphism (see Definition 3.1.3) so applying the respective free construction (Theorem 3.2.18) provides \( \chi : \delta \rightarrow \text{reach}(\text{sc}(\text{rev}(N))) \). Applying duality, Corollary 3.5.13 and Corollary 3.4.5 yields \( \text{simple}(\text{sc}(N)) \rightarrow \text{dfa}_r(L) \), so \( N \) is positively atomic.

3. Assume \( N \) is a subatomic nfa. Then we have \( \iota : \gamma \rightarrow \text{rsc}(\text{syn}(L)) \) where \( \gamma : = \text{simple}(\text{sc}(N)) \). Since \( \iota \)'s codomain is right-quotient closed we also have the JSL-dfa inclusion morphism \( \iota_1 : \text{rsc}(\gamma) \rightarrow \text{rsc}(\text{syn}(L)) \). Dualising, and applying Theorem 3.7.11 and Theorem 3.6.6, we obtain a surjective morphism \( q_1 : \text{syn}(\text{dfa}_{\text{syn}}(L')) \rightarrow \text{syn}(\gamma^*) \).

Furthermore applying right-quotient closure to the inclusion \( \text{hin}(L) \rightarrow \gamma \) yields \( \text{dfa}_{\text{syn}}(L) = \text{rsc}(\text{dfa}(L)) \rightarrow \text{rsc}(\gamma) \).

Dualising the latter and applying Corollary 3.7.16 we obtain a surjective morphism \( q_2 : \text{syn}(\gamma^*) \rightarrow \text{syn}(L') \) onto \( L' \)'s syntactic semiring (viewed as a JSL-dfa). Then consider the composite JSL-dfa morphism:

\[
\text{sc}(\text{dfa}_{\text{syn}}(L')) \xrightarrow{q_1} \text{syn}(\gamma^*) \xrightarrow{q_2} \text{syn}(L').
\]

The classically reachable part of the domain JSL-dfa consists of singleton sets and is isomorphic to \( \text{dfa}_{\text{syn}}(L') \). Likewise by Corollary 3.7.8 the reachable part of the codomain is isomorphic to \( \text{dfa}_{\text{syn}}(L') \). The image of a reachable dfa under a dfa morphism is reachable, so the composite morphism restricts to:

\[
\begin{array}{ccc}
\text{reach}(\text{sc}(\text{dfa}_{\text{syn}}(L'))) & \xrightarrow{q_1} & \text{reach}(\text{syn}(\gamma^*)) \\
\downarrow & & \downarrow \\
\text{dfa}_{\text{syn}}(L') & \xrightarrow{q_2} & \text{dfa}_{\text{syn}}(L')
\end{array}
\]

Then \( q_1 \) is bijective and hence a dfa isomorphism, so that \( \text{reach}(\text{syn}(\gamma^*)) \cong \text{dfa}_{\text{syn}}(L') \) too. Importantly,

\[
\text{reach}(\text{syn}(\gamma^*)) \cong \text{dfa}_{\text{TM}}(\text{rsc}(\text{rev}(N)))
\]

because \( \gamma^* \cong \text{reach}(\text{syn}(\text{recl}(N))) \) by Corollary 3.4.5 and Example 3.2.12, so we can apply Lemma 3.7.3. Finally, the action of the dfa isomorphism \( \text{dfa}_{\text{TM}}(\text{rsc}(\text{rev}(N))) \cong \text{dfa}_{\text{syn}}(L') \) defines the desired monoid isomorphism.

Conversely suppose \( \text{TM}(\text{rsc}(\text{rev}(N))) \cong \text{Syn}(L') \) via the generator-preserving mapping \( \lbrack w \rbrack_{\text{TM}(\text{rsc}(\text{rev}(N)))} \rightarrow \lbrack w \rbrack_{\text{Syn}(L')} \). Its action defines a dfa isomorphism \( \text{dfa}_{\text{TM}}(\text{rsc}(\text{rev}(N))) \cong \text{dfa}_{\text{syn}}(L') \), where the conditions concerning the initial state and transitions are obvious. The final states are preserved/reflected because:

\[
\begin{align*}
\lbrack w \rbrack_{\text{TM}(\text{rsc}(\text{rev}(N)))} & \in F_{\text{dfa}_{\text{TM}}(\text{rsc}(\text{rev}(N)))} \iff N_{w}[I_{\text{rev}(N)}] \cap F_{\text{rev}(N)} \neq \emptyset \\
& \iff N_{w}[F_N] \cap I_N \neq \emptyset \\
& \iff w^* \in L \\
& \iff w \in L' \\
& \iff \lbrack w \rbrack_{\text{Syn}(L')} \in F_{\text{dfa}_{\text{syn}}(L')}. 
\end{align*}
\]

Applying Lemma 3.7.3 we deduce \( \text{dfa}_{\text{syn}}(L') \cong \text{dfa}_{\text{TM}}(\text{rsc}(\text{rev}(N))) \cong \text{reach}(\text{syn}(\text{sc}(\text{rev}(N)))) \). Then we have a dfa morphism \( f : \text{dfa}_{\text{syn}}(L') \rightarrow \text{syn}(\text{sc}(\text{rev}(N))) \). Applying the free construction (Theorem 3.2.17) we obtain \( f : \text{sc}(\text{dfa}_{\text{syn}}(L')) \rightarrow \text{syn}(\text{recl}(N)) \). It is actually surjective because \( \text{ts}(\cdot) \) constructs JSL-reachable machines. Dualising this free-extension yields:

\[
\text{rsc}(\text{sc}(N)) \cong (\text{syn}(\text{sc}(\text{rev}(N))))^* \xrightarrow{f^*} (\text{sc}(\text{dfa}_{\text{syn}}(L')))^* \cong \text{ DFA}_{\text{syn}}(L).
\]

The left isomorphism follows by Theorem 3.7.11 and Example 3.4.6, whereas the right one follows by Theorem 3.6.6. Finally since \( \text{simple}(\text{sc}(N)) \rightarrow \text{rsc}(\text{sc}(N)) \) we deduce \( N \) is subatomic.

\( \square \)
References

[ADN92] André Arnold, Anne Dicky, and Maurice Nivat. A note about minimal non-deterministic automata. *Bulletin of the EATCS*, 47:166–169, 1992.

[BFRK08] Sergei Bezrukov, Dalibor Fronek, Steven J. Rosenberg, and Petr Kov. On biclique coverings. *Discrete Mathematics*, 308(2):319 – 323, 2008. Combinatorics04.

[Brz64] Janusz A. Brzozowski. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, October 1964.

[BT14] Janusz Brzozowski and Hellis Tamm. Theory of tomata. *Theoretical Computer Science*, 539:13 – 27, 2014.

[Con71] J. H. Conway. *Regular Algebra and Finite Machines*. Printed in GB by William Clowes & Sons Ltd, 1971.

[DLT01] François Denis, Aurélien Lemay, and Alain Terlutte. *Residual Finite State Automata*, pages 144–157. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[GH06] Hermann Gruber and Markus Holzer. *Finding Lower Bounds for Nondeterministic State Complexity Is Hard*, pages 363–374. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[GPJL91] David A Gregory, Norman J Pullman, Kathryn F Jones, and J. Richard Lundgren. Biclique coverings of regular bigraphs and minimum semiring ranks of regular matrices. *Journal of Combinatorial Theory, Series B*, 51(1):73 – 89, 1991.

[GW05] George Grätzer and Friedrich Wehrung. Tensor products of semilattices with zero, revisited. *arXiv Mathematics e-prints*, page math/0501436, Jan 2005.

[Jip12] Peter Jipsen. Categories of algebraic contexts equivalent to idempotent semirings and domain semirings. In Wolfram Kahl and Timothy G. Griffin, editors, *Relational and Algebraic Methods in Computer Science*, pages 195–206, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[KW70] T. Kameda and P. Weiner. On the state minimization of nondeterministic finite automata. *IEEE Trans. Comput.*, 19(7):617–627, July 1970.

[LRT09] Michel Latteux, Yves Roos, and Alain Terlutte. Minimal nfa and birfsa languages. *RAIRO - Theoretical Informatics and Applications*, 43(2):221–237, 004 2009.

[MAMU14] Robert S. R. Myers, Jiří Adámek, Stefan Milius, and Henning Urbat. *Canonical Nondeterministic Automata*, pages 189–210. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[Mar75] George Markowsky. The factorization and representation of lattices. *Transactions of the American Mathematical Society*, 203:185–200, 1975.

[Orl77] James Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae (Proceedings)*, 80(5):406–424, 1977.

[Pol01] Libor Polák. *Syntactic Semiring of a Language*, pages 611–620. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[Tam10] Hellis Tamm. *Some Minimality Results on Biresidual and Biseparable Automata*, pages 573–584. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[Tam16] Hellis Tamm. New interpretation and generalization of the kameda-weiner method. In *ICALP*, 2016.

[Wat01] Valerie L. Watts. Boolean rank of kronecker products. *Linear Algebra and its Applications*, 336(1):261 – 264, 2001.