Chapter 1

Schedule Properties and Argumentation

Schedule properties such as efficiency are modelled using frameworks to explain the satisfaction of properties. In particular, the definitions of efficiency and fixed decision frameworks extend from the definition of the feasibility framework. This is generalised to reason about arbitrary number of properties, using a commonly-extended framework. Using this extended reasoning, we apply argumentation to a variant interval scheduling to illustrate extensions of argumentation.

1.1 Extensions

We use stability over other notions of good extensions to accurately model schedule constraints.

1.2 Frameworks

In order to reason about arbitrary number of properties, we inductively build the expressible properties over a commonly-extended framework, denoted by $\rightsquigarrow_0$. A property P is modelled by the framework $\rightsquigarrow_P$. To be correct, we must preserve the stability of some extension E on $\langle Args, \rightsquigarrow_0 \cup \rightsquigarrow_P \rangle$ if E is also stable on $\langle Args, \rightsquigarrow_0 \rangle$ and $P(S)$ is true. Let $\rightsquigarrow_1 \subseteq Args^2$ and $\rightsquigarrow_2 \subseteq Args^2$ be arbitrary frameworks.

Definition 1. A framework $\rightsquigarrow$ stability-models a schedule property P iff for all extensions E and corresponding schedules S, E is stable on $\langle Args, \rightsquigarrow \rangle \Leftrightarrow P(S)$

Definition 2. A framework $\rightsquigarrow$ conflict-models a schedule property P iff for all extensions E and corresponding schedules S, E is conflict-free on $\langle Args, \rightsquigarrow \rangle \Leftrightarrow P(S)$

Definition 3. A schedule property P is stability-modelable iff there exists a framework that stability-models P.

Lemma 1. E is conflict-free on $\langle Args, \rightsquigarrow_1 \rangle$ and on $\langle Args, \rightsquigarrow_2 \rangle$ iff E is conflict-free on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2 \rangle$.

Proof. To prove the forward implication, assume E is conflict-free on $\langle Args, \rightsquigarrow_1\rangle$ and on $\langle Args, \rightsquigarrow_2\rangle$. To aim for a contradiction, assume E is not conflict-free on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2\rangle$. Then there exists $e_1, e_2 \in E$ such that $e_1(\rightsquigarrow_1 \cup \rightsquigarrow_2)e_2$. Then $e_1 \rightsquigarrow_1 e_2$ or $e_1 \rightsquigarrow_2 e_2$. Both cases lead to a contradiction, so E is conflict-free on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2\rangle$.

To prove the backward implication, assume E is conflict-free on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2\rangle$. To aim for a contradiction, assume E is not conflict-free on $\langle Args, \rightsquigarrow_1\rangle$. Then there exists $e_1, e_2 \in E$ such that $e_1 \rightsquigarrow_1 e_2$. Then $e_1(\rightsquigarrow_1 \cup \rightsquigarrow_2)e_2$, which contradicts the most recent assumption. Therefore, E is conflict-free on $\langle Args, \rightsquigarrow_1\rangle$, and also conflict-free on $\langle Args, \rightsquigarrow_2\rangle$ by similar argument. $\square$

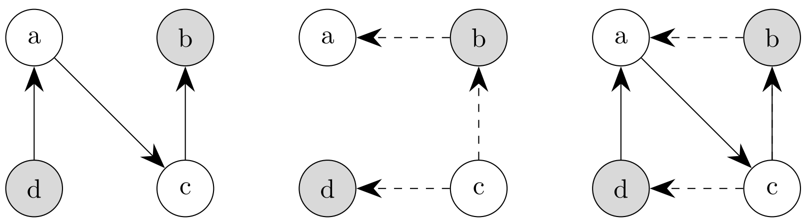

Figure 1.1: Lemma 1 states that given a conflict-free extension over two attack sets on the same arguments, the extension is conflict-free on the merged framework. The figure illustrates this by merging the left and middle frameworks to produce the right framework.

Lemma 2. If E is stable on $\langle Args, \rightsquigarrow_1\rangle$ and E is conflict-free on $\langle Args, \rightsquigarrow_2\rangle$, then E is stable on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2\rangle$.

Proof. Assume E is stable on $\rightsquigarrow_1$ and E is conflict-free on $\rightsquigarrow_2$. By definition of stability, $\forall a \in Args \setminus E\ \exists e \in E\ e \rightsquigarrow_1 a$. Then $\forall a \in Args \setminus E\ \exists e \in E\ e(\rightsquigarrow_1 \cup \rightsquigarrow_2)a$. So every argument not in E is attacked by some argument in E. E is conflict-free on $\rightsquigarrow_1$ because E is stable on $\rightsquigarrow_1$. Since E is conflict-free on $\rightsquigarrow_1$ and on $\rightsquigarrow_2$, we use Lemma 1 to show that E is also conflict-free on $(\rightsquigarrow_1 \cup \rightsquigarrow_2)$. Therefore E is stable on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2\rangle$. $\square$

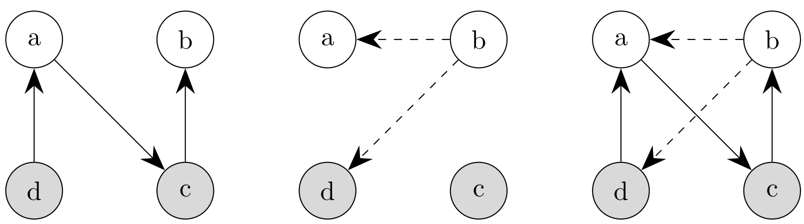

Figure 1.2: Lemma 2 allows stable extensions of attacks to grow with conflict-free attacks. Note that the right, merged framework is stable while the left, base framework is stable.

Lemma 3. If E is stable on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2\rangle$, then E is conflict-free on $\langle Args, \rightsquigarrow_1\rangle$.

Proof. Assume E is stable on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2 \rangle$. By definition of stability, E is conflict-free on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2 \rangle$. By Lemma 1, E is conflict-free on $\langle Args, \rightsquigarrow_1 \rangle$. $\square$

Figure 1.3: Lemma 3 states that given a stable extension, removing attacks preserves the extension's conflict-freeness, as shown from left to right.

Lemma 4. If E is stable on $\langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2 \rangle$ and $\forall a \in Args \setminus E\ (\exists e \in E\ e \rightsquigarrow_2 a) \implies (\exists e \in E\ e \rightsquigarrow_1 a)$, then E is stable on $\langle Args, \rightsquigarrow_1 \rangle$.

Proof.

$$
\begin{aligned}
& E \text{ is stable on } \langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2 \rangle \\
& \wedge \forall a \in Args \setminus E\ (\exists e \in E\ e \rightsquigarrow_2 a) \implies (\exists e \in E\ e \rightsquigarrow_1 a) \\
\implies & E \text{ is conflict-free on } \langle Args, \rightsquigarrow_1 \cup \rightsquigarrow_2 \rangle \\
& \wedge \forall a \in Args \setminus E\ \exists e \in E\ e(\rightsquigarrow_1 \cup \rightsquigarrow_2) \\
& \wedge \forall a \in Args \setminus E\ (\exists e \in E\ e \rightsquigarrow_2 a) \implies (\exists e \in E\ e \rightsquigarrow_1 a) \\
\implies & E \text{ is conflict-free on } \langle Args, \rightsquigarrow_1 \rangle \\
& \wedge \forall a \in Args \setminus E\ \exists e \in E\ (e \rightsquigarrow_1 a \vee e \rightsquigarrow_2 a) \\
& \wedge \forall a \in Args \setminus E\ (\exists e \in E\ e \rightsquigarrow_2 a) \implies (\exists e \in E\ e \rightsquigarrow_1 a) \\
\implies & E \text{ is conflict-free on } \langle Args, \rightsquigarrow_1 \rangle \\
& \wedge \forall a \in Args \setminus E\ ((\exists e \in E\ e \rightsquigarrow_1 a) \vee (\exists e \in E\ e \rightsquigarrow_2 a)) \\
& \wedge \forall a \in Args \setminus E\ (\exists e \in E\ e \rightsquigarrow_2 a) \implies (\exists e \in E\ e \rightsquigarrow_1 a) \\
\implies & E \text{ is conflict-free on } \langle Args, \rightsquigarrow_1 \rangle \\
& \wedge \forall a \in Args \setminus E\ ((\exists e \in E\ e \rightsquigarrow_1 a) \vee (\exists e \in E\ e \rightsquigarrow_1 a)) \\
\implies & E \text{ is conflict-free on } \langle Args, \rightsquigarrow_1 \rangle \\
& \wedge \forall a \in Args \setminus E\ \exists e \in E\ e \rightsquigarrow_1 a \\
\implies & E \text{ is stable on } \langle Args, \rightsquigarrow_1 \rangle
\end{aligned}
$$

$\square$

Figure 1.4: Lemma 4 states that given a stable extension, removing attacks on multi-attacked arguments preserves the extension's stability, as shown from left to right.

Theorem 1 (**Union of modelling frameworks**)**.** Let $P_0, ..., P_K$ be schedule properties with K properties. Let $P_{[\![i,j]\!]}$ be an aggregate schedule property where for all schedules S, $P_{[\![i,j]\!]}(S) \Leftrightarrow \forall k \in [\![i,j]\!]\ P_k(S)$.

If $\rightsquigarrow_0$ stability-models P_0, and $\forall k \in [\![1,K]\!]\ \rightsquigarrow_k$ conflict-models P_k, and for all extensions E, $\forall a \in Args \setminus E\ \forall k \in [\![1,K]\!]\ \big((\exists e \in E\ e \rightsquigarrow_k a) \implies (\exists e \in E\ e \rightsquigarrow_0 a)\big)$, then $\left(\bigcup_{k=0}^{K} \rightsquigarrow_k\right)$ stability-models $P_{[\![0,K]\!]}$.

Proof. Take arbitrary $K \in \mathbb{N}$. To prove forward implication:

1. $\rightsquigarrow_0$ stability-models P_0 — given
2. $\forall k \in [\![1,K]\!]\ \rightsquigarrow_k$ conflict-models P_k — given
3. $\forall a \in Args \setminus E\ \forall k \in [\![1,K]\!]\ \big((\exists e \in E\ e \rightsquigarrow_k a) \implies (\exists e \in E\ e \rightsquigarrow_0 a)\big)$ — given
4. E is stable on $\langle Args, \bigcup_{k=0}^{K} \rightsquigarrow_k \rangle$ — assumption
5. $\forall a \in Arg \setminus E\ \left(\left(\exists e \in E\ e \left(\bigcup_{k=0}^{K} \rightsquigarrow_k\right) a\right) \implies (\exists e \in E\ e \rightsquigarrow_0 a)\right)$ — 3
6. E is stable on $\langle Args, \rightsquigarrow_0 \rangle$ — lemma 4, 4, 5
7. $P_0(S)$ — 1, 6
8. Take arbitrary $k \in [\![1,K]\!]$
    9. E is conflict-free on $\langle Args, \rightsquigarrow_k \rangle$ — lemma 3, 4
    10. P_k(S) — 2, 9
11. $\forall k \in [\![1,K]\!]\ P_k(S)$ — 8, 10
12. $P_{[\![0,K]\!]}(S)$ — 7, 11

To prove backward implication:

1. $\rightsquigarrow_0$ stability-models P_0 — given
2. $\forall k \in [\![1,K]\!]\ \rightsquigarrow_k$ conflict-models P_k — given

3. $P_{[\![0,K]\!]}(S)$ — assumption
4. $P_0(S)$ — 3
5. E is stable on $\langle Args, \rightsquigarrow_0 \rangle$ — 1, 4
6. Recursively over $k \in [\![1, K]\!]$
    7. $P_k(S)$ — 3
    8. E is conflict-free on $\langle Arg, \rightsquigarrow_k \rangle$ — 2, 7
    9. E is stable on $\langle Arg, \bigcup_{k'=0}^{k} \rightsquigarrow_{k'} \rangle$ — lemma 2, 5, 8
10. E is stable on $\langle Arg, \bigcup_{k=0}^{K} \rightsquigarrow_k \rangle$ — 6, 9

$\square$

This theorem is a key statement in framing argumentation semantics for arbitrary scheduling problems.

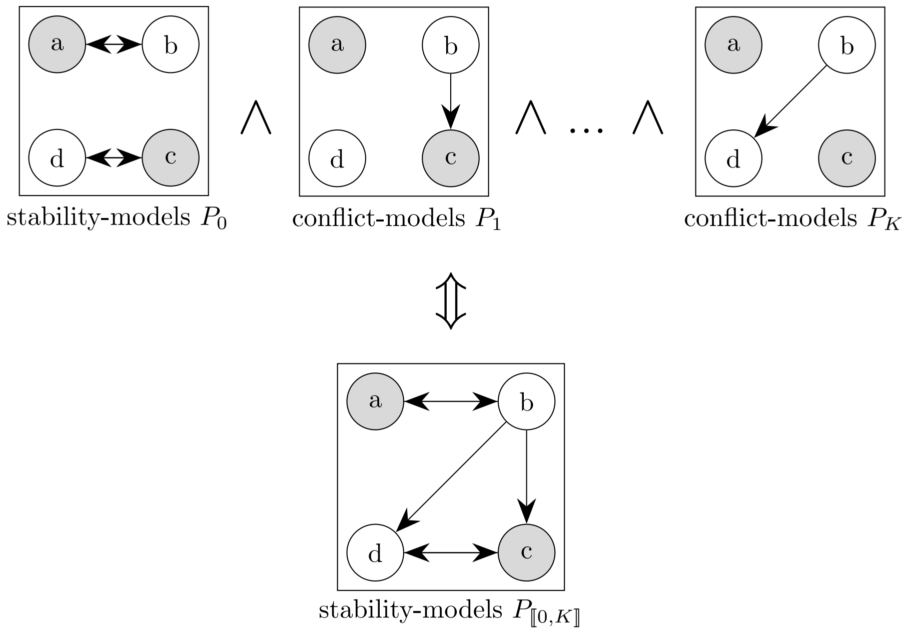

Figure 1.5: Theorem 1 allows construction and deconstruction of an aggregate property, $P_{[\![0,K]\!]}$ from carefully overlaying frameworks, while preserving stability.

1.3 Interval Scheduling

Makespan schedules are extended to discrete time-indexed interval scheduling. Let T be the exclusive upper-bound of indexed time where $\mathcal{T} = \{0, ..., T-1\}$. The assignment matrix $\mathbf{x} \in \mathcal{M} \times \mathcal{J} \times \mathcal{T}$ is extended such that $x_{i,j,t} = 1$ iff job j is starts work on machine i at time t. Each job j has a start time $s_j \in \mathcal{T}$

and finish time $f_j \in \{0, ..., T\}$, where j must be completed within the $[s_j, f_j)$ interval. The objective is to minimise the total completion time.

$$\min_{\mathbf{x}} C_{\max} \text{ subject to:}$$
$$\forall i \in \mathcal{M}\ \forall j \in \mathcal{J}\ \forall t \in \mathcal{T}\ C_{\max} \geq x_{i,j,t}(t + p_j)$$
$$\forall j \in \mathcal{J}\ \sum_{i \in \mathcal{M}} \sum_{t \in \mathcal{T}} x_{i,j,t} = 1 \qquad \alpha$$
$$\forall i \in \mathcal{M}\ \forall t \in \mathcal{T}\ \sum_{j \in \mathcal{J}} \sum_{t' = \max\{t - p_j + 1, 0\}}^{t} x_{i,j,t'} \leq 1 \qquad \beta$$
$$\forall i \in \mathcal{M}\ \forall j \in \mathcal{J}\ \forall t \in \{0, ..., s_j - 1\}\ x_{i,j,t} = 0 \qquad \gamma$$
$$\forall i \in \mathcal{M}\ \forall j \in \mathcal{J}\ \forall t \in \{f_j - p_j + 1, ..., T - 1\}\ x_{i,j,t} = 0 \qquad \delta$$
$$\forall \langle i, j \rangle \in D^-\ \forall t \in \mathcal{T}\ x_{i,j,t} = 0 \qquad \varepsilon$$
$$\forall \langle i', j \rangle \in D^+\ \forall i \in \mathcal{M} \setminus \{i'\}\ \forall t \in \mathcal{T}\ x_{i,j,t} = 0 \qquad \zeta$$

α models feasibility, that all jobs must be allocated. β models that machines cannot process multiple jobs at the same time. γ and δ models the restriction of start and end times respectively. ε and ζ models negative and positive fixed decisions respectively. Equivalently, ζ can be modelled by $\forall \langle i, j \rangle \in D^+\ \exists t \in \mathcal{T}\ x_{i,j,t} = 1$. ζ is defined as such to simplify the proof that the union of these properties is modelable. For interval scheduling, let $Args = \mathcal{M} \times \mathcal{J} \times \mathcal{T}$.

Definition 4. Let $\rightsquigarrow_\alpha$ be the base-feasibility framework such that $\langle i, j, t \rangle \rightsquigarrow_\alpha \langle i', j', t' \rangle \Leftrightarrow i \neq i' \wedge j = j' \wedge t \neq t'$

Lemma 5. $\rightsquigarrow_\alpha$ stability-models α.

Proof. To prove forward implication: E is stable on $\langle Args, \iff \alpha \rangle$. Take arbitrary $j \in \mathcal{J}$. To aim to contradict, assume $\sum_{i \in \mathcal{M}} \sum_{t \in \mathcal{T}} x_{i,j,t} > 1$. Then $\exists \langle i, j, t \rangle, \langle i', j, t' \rangle \in E$ where $x_{i,j,t} = 1$ and $x_{i',j,t'} = 1$ such that $i \neq i'$ or $t \neq t'$. By definition of $\rightsquigarrow_\alpha$, $\langle i, j, t \rangle \rightsquigarrow_\alpha \langle i', j, t' \rangle$. Hence E is not conflict-free, then E is not stable. By contradiction, $\sum_{i \in \mathcal{M}} \sum_{t \in \mathcal{T}} x_{i,j,t} \leq 1$. To aim to contradict, assume $\sum_{i \in \mathcal{M}} \sum_{t \in \mathcal{T}} x_{i,j,t} = 0$. Then $\forall i \in \mathcal{M}\ \forall t \in \mathcal{T}\ x_{i,j,t} = 0$. Then $\forall i \in \mathcal{M}\ \forall t \in \mathcal{T}\ \langle i, j, t \rangle \notin E$. Then E is not stable. By contradiction, $\sum_{i \in \mathcal{M}} \sum_{t \in \mathcal{T}} x_{i,j,t} > 0$. Therefore α holds.

To prove backward implication: From α, there is exactly one $i \in \mathcal{M}$ and $t \in \mathcal{T}$ such that $x_{i,j,t} = 1$. So E is conflict free. Also, for all j, $\langle i, j, t \rangle \in E$ attacks every other $\langle i, t \rangle$, so E is stable. □

Definition 5. Let $\rightsquigarrow_\beta$ be the sequential-feasibility framework such that $\langle i, j, t \rangle \rightsquigarrow_\beta \langle i', j', t' \rangle \Leftrightarrow i = i' \wedge (t' \leq t \leq t' + p_j \vee t \leq t' < t + p_j)$.

Lemma 6. $\rightsquigarrow_\beta$ conflict-models β.

Proof. To prove forward implication: Take arbitrary $i \in \mathcal{M}$, $t \in \mathcal{T}$. To aim for a contradiction, assume $\sum_{j \in \mathcal{J}} \sum_{t' \in \max\{t - p_j + 1, 0\}} x_{i,j,t'} \geq 2$. Then there exists some $j_1, j_2 \in \mathcal{J}$ and some $t_1, t_2 \in \mathcal{T}$ such that $0 \leq t_1, t_2 \leq t$ and $x_{i,j_2,t_2} +$

$x_{i,j_2,t_2} = 2$. Then $\langle i, j_1, t_1\rangle \in E$ and $\langle i, j_2, t_2\rangle \in E$. By conduction of β, then either $t_1 \leq t_2 \leq t_1 + p_1$ or $t_2 \leq t_1 \leq t_2 + p_2$. By definition of $\rightsquigarrow_\beta$, $\langle i, j_1, t_1\rangle \rightsquigarrow_\beta \langle i, j_2, t_2\rangle$. But this contradicts that E is conflict-free. Therefore β holds.

To prove backward implication: Assume β holds. Take arbitrary $i \in \mathcal{M}$, $t \in \mathcal{T}$. Then there does not exists overlapping jobs j_1 and j_2 such that $x_{i,j_1,t_1}+x_{i,j_2,t_2} = 2$. Then $\langle i, j_1, t_1\rangle \notin E$ and $\langle i, j_2, t_2\rangle \notin E$. Therefore, E is conflict-free. □

Definition 6. Let $\rightsquigarrow_\gamma$ be a start-feasibility framework such that
$\rightsquigarrow_\gamma = \{\langle\langle i,j,t\rangle, \langle i,j,t\rangle\rangle \mid i \in \mathcal{M}, j \in \mathcal{J}, 0 \leq t < s_j\}$.

Definition 7. Let $\rightsquigarrow_\delta$ be a finish-feasibility framework such that
$\rightsquigarrow_\delta = \{\langle\langle i,j,t\rangle, \langle i,j,t\rangle\rangle \mid i \in \mathcal{M}, j \in \mathcal{J}, f_j - p_j < t < T\}$.

Definition 8. Let $\rightsquigarrow_\varepsilon$ be the negative fixed decision feasibility framework such that
$\rightsquigarrow_\varepsilon = \{\langle\langle i,j,t\rangle, \langle i,j,t\rangle\rangle \mid \langle i,j\rangle \in D^-, t \in \mathcal{T}\}$.

Definition 9. Let $\rightsquigarrow_\zeta$ be the positive fixed decision feasibility framework such that $\rightsquigarrow_\varepsilon = \{\langle\langle i,j,t\rangle, \langle i,j,t\rangle\rangle \mid i \in \mathcal{M}, \langle i',j\rangle \in D^+, i \neq i', t \in \mathcal{T}\}$.

Lemma 7. Let $\mathcal{A} \subseteq Args$ be the set of arbitrary negative fixed decisions. A schedule S satisfies these decisions if property $P_\mathcal{A}$ holds. Formally $P_\mathcal{A} \iff \forall a \in \mathcal{A}\ x_a = 0$. If $\rightsquigarrow_\mathcal{A}$ is defined by $\rightsquigarrow_\mathcal{A} = \{\langle a, a\rangle \mid a \in \mathcal{A}\}$, then $\rightsquigarrow_\mathcal{A}$ conflict-models $P_\mathcal{A}$.

Proof. To prove forward implication: Assume E is conflict free on $\langle Args, \rightsquigarrow_\mathcal{A}\rangle$. Take arbitrary $a \in \mathcal{A}$. To aim for a contradiction, assume $x_a = 1$. Then $a \in E$. By definition of $\rightsquigarrow_\mathcal{A}$, $a \rightsquigarrow_\mathcal{A} a$. But this contradicts E is conflict-free so $x_a = 0$. Therefore $P_\mathcal{A}(S)$ holds.

To prove backward implication: Assume $P_\mathcal{A}(S)$ holds. Take arbitrary $a \in \mathcal{A}$. To aim for a contradiction, assume $a \rightsquigarrow_\mathcal{A} a$. Then $a \in E$, so $x_a = 1$. This contradicts $P_\mathcal{A}(S)$, so E is conflict-free. □

Lemma 8. For all extensions E, $\forall a \in Args \setminus E\ \forall \lambda \in \{\beta, \gamma, \delta, \varepsilon, \zeta\}\ \big((\exists e \in E\ e \rightsquigarrow_\lambda a) \implies (\exists e \in E\ e \rightsquigarrow_\alpha a)\big)$.

Proof. Take arbitrary extension E and arbitrary $a \in Args \setminus E$. If $\lambda \neq \beta$ and $\exists e \in E\ e \rightsquigarrow_\lambda a$, then $a = e$ from the definition of $\rightsquigarrow_\lambda$. But $e \notin Args \setminus E$. By contradiction, $\lambda = \beta$.

If $m = 0$ or $T = 0$, then $Args = \varnothing$, so the proof is trivial.

If $m = 1$ and $T = 1$, then by definition of $\rightsquigarrow_\beta$, $\rightsquigarrow_\beta = \varnothing$. So $\neg\exists e \in E\ e \rightsquigarrow_\beta a$. As the condition does not hold, $\exists e \in E\ e \rightsquigarrow_\alpha a$.

Otherwise, let $a = \langle i, j, t\rangle$. Because $m > 2$ or $T > 2$, then there exists i and t such that $\langle i, t\rangle \neq \langle i', t'\rangle$. By definition of $\rightsquigarrow_\alpha$, $\langle i', j, t'\rangle \rightsquigarrow_\alpha a$. □

Theorem 2 (Interval scheduling is stability-modelable). Let $\Lambda(S)$ iff $\forall \lambda \in \{\alpha, \beta, \gamma, \delta, \varepsilon, \zeta\}\ \lambda(S)$. Λ is stability-modelable.

Proof.

1. $\rightsquigarrow_\gamma$ conflict-models γ — definition of $\rightsquigarrow_\gamma$, lemma 7
2. $\rightsquigarrow_\delta$ conflict-models δ — definition of $\rightsquigarrow_\delta$, lemma 7
3. $\rightsquigarrow_\varepsilon$ conflict-models ε — definition of $\rightsquigarrow_\varepsilon$, lemma 7
4. $\rightsquigarrow_\zeta$ conflict-models ζ — definition of $\rightsquigarrow_\zeta$, lemma 7
5. $\bigcup_{\lambda \in \{\alpha,\beta,\gamma,\delta,\varepsilon,\zeta\}} \rightsquigarrow_\lambda$ stability-models Λ
   lemma 5, lemma 6, 1, 2, 3, lemma 8, theorem 1
6. Λ is stability-modelable — 5

$\square$