Description Logics for Solving Scheduling Problems

Pok-Son Kim
Kookmin University, College of Natural Sciences
Department of Mathematics
Seoul 136-702, Korea

Abstract

We introduce new methods for representing and solving general classes of non-preemptive resource-constrained project scheduling problems that minimize the project makespan. These methods are based on a new approach to represent scheduling problems as descriptions (activity-terms) of terminological languages called $\mathcal{RSV}$ and $\mathcal{RCPSV}$, which allow nested expressions using $\mathbf{pll}$, $\mathbf{seq}$, $\mathbf{xor}$ and $\mathbf{hnet}$, $\mathbf{xor}$ respectively. The activity-terms of $\mathcal{RSV}$ and $\mathcal{RCPSV}$ are similar to concepts in a description logic. The languages $\mathcal{RSV}$ and $\mathcal{RCPSV}$ generalize previous approaches to scheduling with variants insofar as it permits $\mathbf{xor}$'s not only of atomic activities but also of arbitrary activity-terms. Specific semantics that assign their set of active schedules to activity-terms show correctness of calculuses normalizing activity-terms of $\mathcal{RSV}$ and $\mathcal{RCPSV}$ respectively similar to propositional DNF-computation.

Overview

1. $\mathcal{RSV}$- and $\mathcal{RCPSV}$-Scheduling Problems

2. The Scheduling Language $\mathcal{RSV}$

3. A Calculus for $\mathcal{RSV}$

4. The Algorithm $\mathcal{A}_{\mathcal{RSV}}$

5. The Scheduling Language $\mathcal{RCPSV}$

6. A Calculus for $\mathcal{RCPSV}$

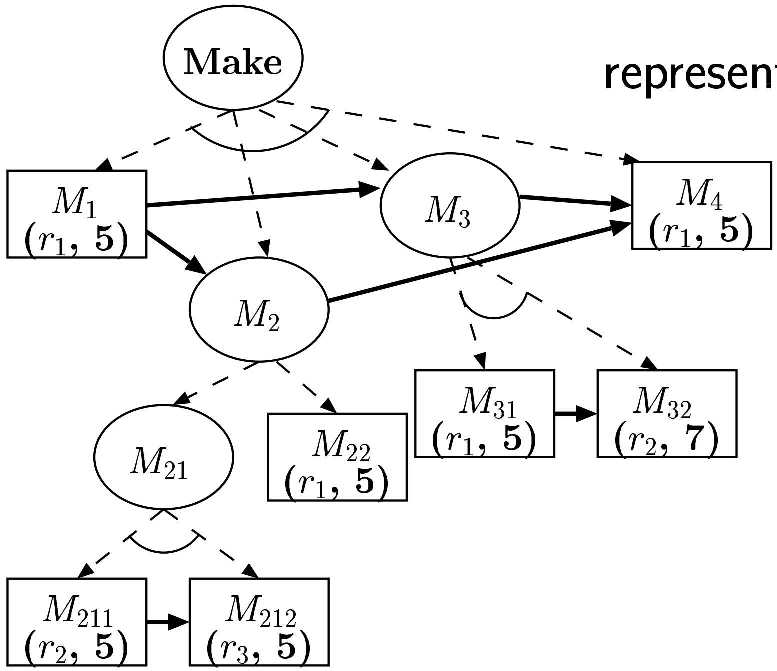
$\mathcal{RCPSV}$-Problems

$\mathcal{RC}$: resource constrained

$\mathcal{PS}$: project scheduling

$\mathcal{V}$: with variants

Graph, Example



represented as ANDORI-graph

- **variant processes**
- **precedence constraints**
- **parallel processing**
- **resource constraints**

Methods for formalizing and Solving $\mathcal{RCPSV}$-Problems

- Definition of the logic-based Scheduling languages $\mathcal{RSV}$ and $\mathcal{RCPSV}$.
- Diagram-based Solution Algorithms
  - $\mathcal{RSV}$-Diagram Calculation
  - $\mathcal{RCPSV}$-Diagram Calculation

The Scheduling Language $\mathcal{RSV}$ - Definition

The Syntax of the Language $\mathcal{RSV}$

- **ground (atomic) activities**

$$P(r,t) \qquad \begin{array}{l} P: \text{ predicate symbol} \\ r: \text{ resource (human, machine, asset, } \cdots) \\ t \in \mathbb{N}: \text{ duration} \end{array}$$

- **operators**
  - **seq**: sequential processing of an activity term or activity terms
  - **xor**: alternative activity terms
  - **pll**: parallel processing of activity terms

The Scheduling Language $\mathcal{RSV}$ - Definition

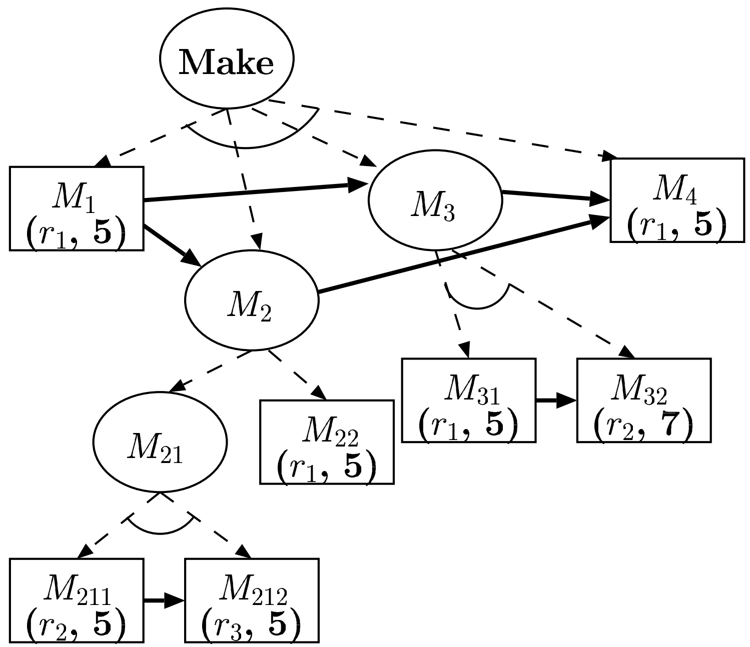
- **activity terms**

  1. Each ground activity is an activity term
  2. $P_1, P_2, \cdots, P_k$: activity terms, then
     $(\mathbf{seq}\, P_1, \cdots, P_k)$,
     $(\mathbf{xor}\, P_1, \cdots, P_k)$,
     $(\mathbf{pll}\, P_1, \cdots, P_k)$ are also activity terms.

Example 1:

$$\begin{array}{ll}(\mathbf{seq} & (\mathbf{xor}\, P_1(r_1, 15),\ (\mathbf{pll}\, P_2(r_2, 2),\ P_3(r_2, 2))) \\ & (\mathbf{pll}\, P_4(r_3, 10),\ P_5(r_4, 12)))\end{array}$$

Activity-terms; Example 2



$$\mathbf{seq} M_1, (\mathbf{pll}(\mathbf{seq} M_{31}, M_{32}), (\mathbf{xor}(\mathbf{seq} M_{211}, M_{212}), M_{22})), M_4$$

Reduced Activity Terms - Definition

A, B: $\mathcal{RSV}$-expressions

B is a *reduced activity subterm* of A, if B can be derived from A by repeatedly replacing subterms of the form $(\mathbf{xor}\, C_1, \cdots, C_n)$ by exactly one C_i $(i = 1, \cdots$ or $n)$ so that B is $\mathbf{xor}$-free.

Example:
The 2 reduced activity terms of the previous activity-term:

$$\begin{aligned}(\mathbf{seq}\quad & P_1(r_1, 15),\\ & (\mathbf{pll}\, P_4(r_3, 10),\ P_5(r_4, 12)))\end{aligned}$$

$$\begin{aligned}(\mathbf{seq}\quad & (\mathbf{pll}\, P_2(r_2, 2),\ P_3(r_2, 2)),\\ & (\mathbf{pll}\, P_4(r_3, 10),\ P_5(r_4, 12)))\end{aligned}$$

Active Schedules - Definition

A: a *reduced* activity term

$g(A) = \{A_1, \cdots, A_n\}$: the set consisting of all ground activities occurring in A

An active schedule for A is a set of starting times of ground activities $\{t_{A_i} \in \mathbb{N} | A_i \in g(A)\}$ such that:

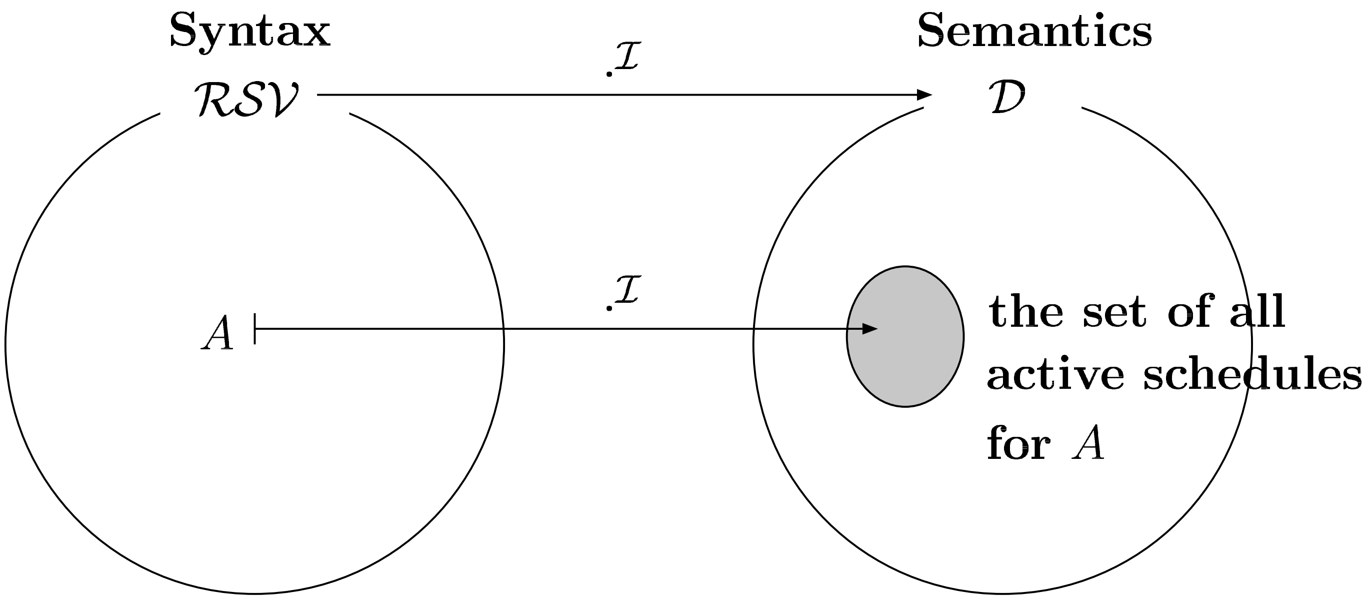
- The precedence constraints are satisfied,
- The resource constraints are satisfied and
- No ground activity can be started earlier without changing other start times.

Semantics of the Language $\mathcal{RSV}$ - Definition

Interpretation $\mathcal{I} = (\mathcal{D}, \cdot^{\mathcal{I}})$

$\mathcal{D}$: the set consisting of all active schedules derived from activity terms in $\mathcal{RSV}$

$\cdot^{\mathcal{I}}$: the interpretation function of $\mathcal{I}$



The $\mathcal{RSV}$-Calculus

$$\frac{(\mathbf{seq}\ A_1, A_2, \cdots, A_k, (\mathbf{seq}\ B_1, B_2, \cdots, B_l), A_{k+2}, A_{k+3}, \cdots, A_n)}{(\mathbf{seq}\ A_1, A_2, \cdots, A_k, B_1, B_2, \cdots, B_l, A_{k+2}, A_{k+3}, \cdots, A_n)} \quad (1)$$

$$\frac{(\mathbf{xor}\ A_1, A_2, \cdots, A_k, (\mathbf{xor}\ B_1, B_2, \cdots, B_l), A_{k+2}, A_{k+3}, \cdots, A_n)}{(\mathbf{xor}\ A_1, A_2, \cdots, A_k, B_1, B_2, \cdots, B_l, A_{k+2}, A_{k+3}, \cdots, A_n)} \quad (2)$$

$$\frac{(\mathbf{pll}\ A_1, A_2, \cdots, A_k, (\mathbf{pll}\ B_1, B_2, \cdots, B_l), A_{k+2}, A_{k+3}, \cdots, A_n)}{(\mathbf{pll}\ A_1, A_2, \cdots, A_k, B_1, B_2, \cdots, B_l, A_{k+2}, A_{k+3}, \cdots, A_n)} \quad (3)$$

$$\frac{(\mathbf{seq}\ A_1, A_2, \cdots, A_k, (\mathbf{xor}\ B_1, B_2, \cdots, B_l), A_{k+2}, A_{k+3}, \cdots, A_n)}{\begin{array}{r}(\mathbf{xor}\ (\mathbf{seq}\ A_1, A_2, \cdots, A_k, B_1, A_{k+2}, A_{k+3}, \cdots, A_n),\\ (\mathbf{seq}\ A_1, A_2, \cdots, A_k, B_2, A_{k+2}, A_{k+3}, \cdots, A_n),\\ \vdots\\ (\mathbf{seq}\ A_1, A_2, \cdots, A_k, B_l, A_{k+2}, A_{k+3}, \cdots, A_n))\end{array}} \tag{4}$$

$$\frac{(\mathbf{pll}\ A_1, A_2, \cdots, A_k, (\mathbf{xor}\ B_1, B_2, \cdots, B_l), A_{k+2}, A_{k+3}, \cdots, A_n)}{\begin{array}{r}\mathbf{xor}\ ((\mathbf{pll}\ A_1, A_2, \cdots, A_k, B_1, A_{k+2}, A_{k+3}, \cdots, A_n),\\ (\mathbf{pll}\ A_1, A_2, \cdots, A_k, B_2, A_{k+2}, A_{k+3}, \cdots, A_n),\\ \vdots\\ (\mathbf{pll}\ A_1, A_2, \cdots, A_k, B_l, A_{k+2}, A_{k+3}, \cdots, A_n))\end{array}} \tag{5}$$

Correctness of the $\mathcal{RSV}$-Calculus

Lemma *The $\mathcal{RSV}$-Calculus is a correct calculus*

Proof. idea: A rule in the form

$$\frac{A}{B}$$

is "correct" iff the interpretation of the upper expression A and the lower expression B is identical ($A^{\mathcal{I}} = B^{\mathcal{I}}$). We can show this for each of the 5 rules.

Theorem *For any activity description A of $\mathcal{RSV}$ all operators '$\mathbf{xor}$' in the interior of A always can be moved to the leftmost position such that A is transformed to a semantically equivalent, normalized activity-term A' in which the operator '$\mathbf{xor}$' can occur uniquely once in the leftmost position combining all reduced activity-terms derived from A.*

Proof. idea:

- All derivations terminate in a normalized expression (This is the case when no further $\mathcal{RSV}$-rules can be applied.).
- In every expression containing '$\mathbf{xor}$'-operator, it can be shifted to the topmost position.

Example

Calculus that may be used to transform any $\mathcal{RSV}$-expression A into a semantically equivalent *normalized* $\mathcal{RSV}$-expression B:

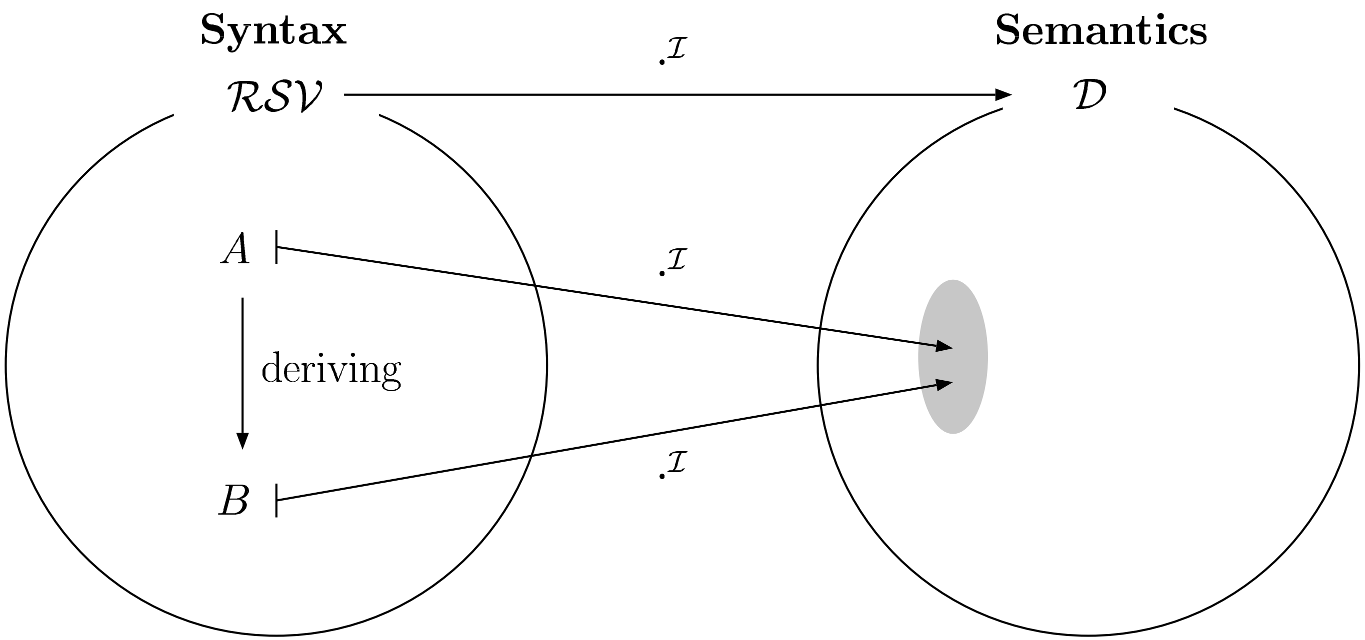
$$(\textbf{pll}\ (\textbf{seq}\ P_3(c,15), P_4(c,16)), (\textbf{xor}\ P_5(c,3), P_6(d,5)))$$

$$\Big\downarrow\ \textbf{normalized by the } \mathcal{RSV}\textbf{-calculus}$$

$$\begin{aligned}(\textbf{xor}\ \ &(\textbf{pll}\ (\textbf{seq}\ P_3(c,15), P_4(c,16)), P_5(c,3)),\\ &(\textbf{pll}\ (\textbf{seq}\ P_3(c,15), P_4(c,16)), P_6(d,5)))\end{aligned}$$

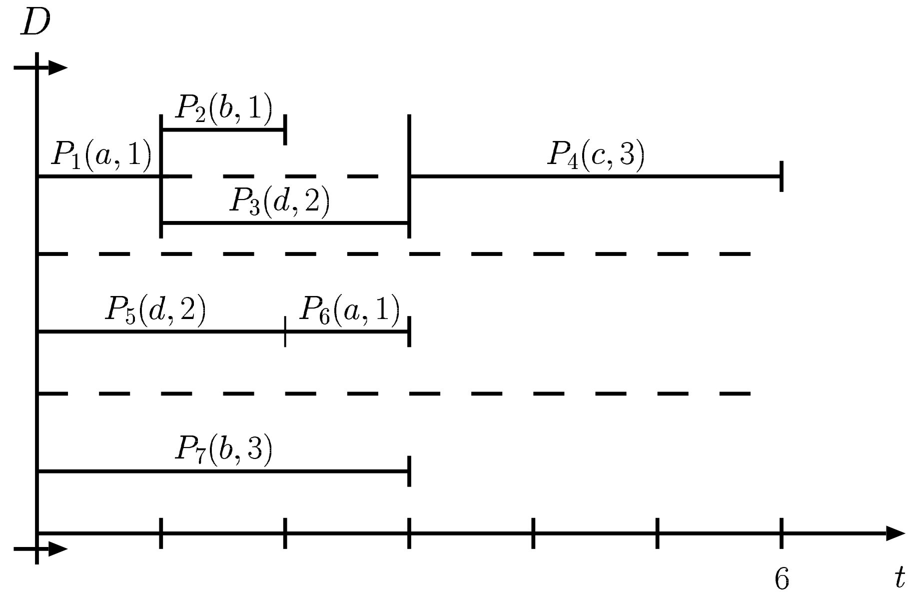
Graphical Representation of a Scheduling Equation

$$A \doteq B$$



A $\mathcal{RSV}$-Diagram
Graphical Representation of a $\mathcal{RSV}$-Term

$$\begin{array}{ll}\mathbf{pll} & (\mathbf{seq}P_1(a,1),(\mathbf{pll}P_2(b,1),P_3(d,2)),P_4(c,3)) \\ & (\mathbf{seq}P_5(d,2),P_6(a,1)) \\ & P_7(b,3)\end{array}$$



Solution Algorithm $\mathcal{A}_{\mathcal{RSV}}$ Based on a Scan-Line Principle

- Attaching start ground activities to the scan-line

Recursive application, only if some unfrozen ground activity exists:

- Moving the scan-line
- Determining and resolving resource conflicts; Freezing all definitely placed ground activities
- Deleting all t_{SL}-time direct scan-line activities from the actual activity term
- Attaching further ground activities to the scan-line

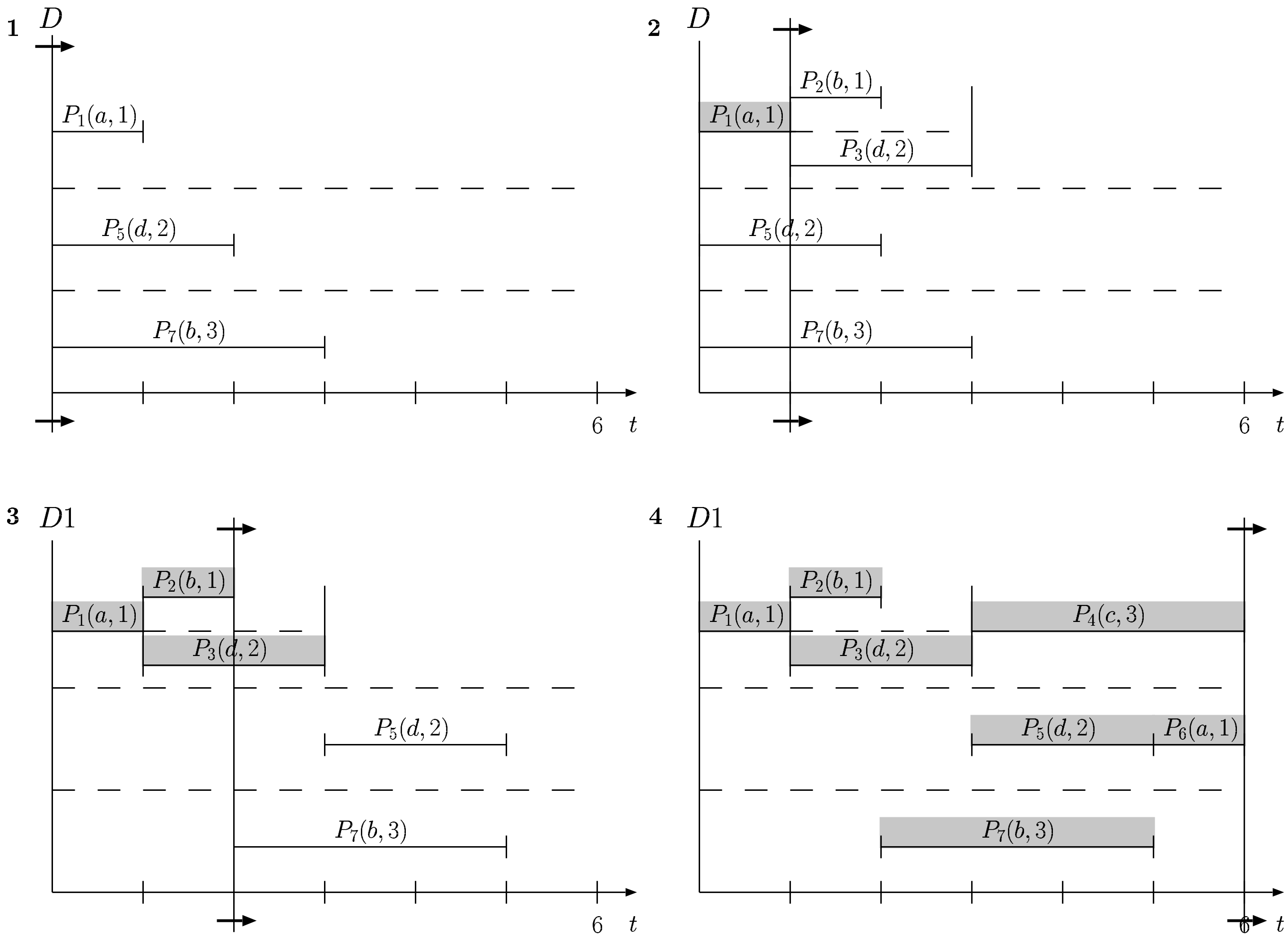


Figure 1: $\mathcal{RSV}$-Diagram-Based Calculation of Active Schedules

Proving Correctness of $\mathcal{A}_{\mathcal{RSV}}$

Theorem: For any given reduced $\mathcal{RSV}$-activity term A, $\mathcal{A}_{\mathcal{RSV}}$ generates nonredundantly all active schedules which may be derived from A.

Complexity of the language $\mathcal{RSV}$
$\mathcal{RSV}$ is $\mathcal{NP}$-complete

The language $\mathcal{RSV}^\star$

$\mathcal{RSV}^\star$ consists of all Pairs (X, t) with $X \in \mathcal{RSV}$ and $t \in \mathbb{N}_0$, where there is an active schedule $P_{a_{ij}}$ with $\mathit{lt}(P_{a_{ij}}) \leq t$ for the term X.

Theorem: $\mathcal{RSV}^\star \in \mathcal{NP}$.

Theorem: $SE \leq \mathcal{RSV}^\star$

The Scheduling Language $\mathcal{RCPSV}$ - Definition

The Syntax of the Language $\mathcal{RCPSV}$

- **h-ground activities**

  $$\{(0, eu\,, 0)\} \cup \{(i, r(i), d(i)) | i = 1, \cdots, n, r(i) \in R, d(i) \in \mathbb{N}_0\}$$

  eu : a dummy-resource

- **operators**

  - **xor**: alternative activity terms
  - **hnet**: structural arrangement of activity terms; The structural arrangement of activity terms of the operator '**hnet**' always has to represent a directed simple and acyclic graph (a scheduling network).

  $$\mathbf{hnet}[let\, t_1 = N_1, \cdots, t_k = N_k; (t_{11}, t_{12}), \cdots, (t_{j1}, t_{j2})]$$

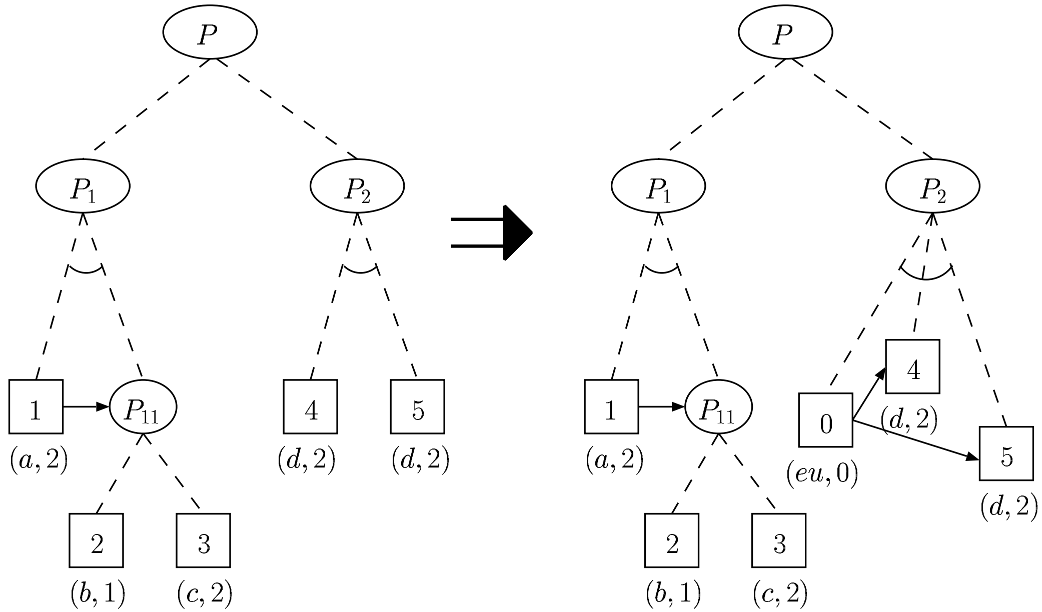
The Scheduling Language $\mathcal{RCPSV}$ - Definition

- h-activity terms
  1. Each h-ground activity is a h-activity term.
  2. $N_1, N_2, \cdots, N_k$: h-activity terms, then
     $(\mathbf{xor}\, N_1, N_2, \cdots, N_k)$ and
     all expressions

$$\mathbf{hnet}[let\, t_1 = N_1, \cdots, t_k = N_k; (t_{11}, t_{12}), \cdots, (t_{j1}, t_{j2})]$$

are h-activity terms, where $t_1, t_2, \cdots$, and t_k correspond to different names (constants) and $[(t_{11}, t_{12}), \cdots, (t_{j1}, t_{j2})]$ represents a scheduling network on $\{t_1, \cdots, t_k\}$.

Example



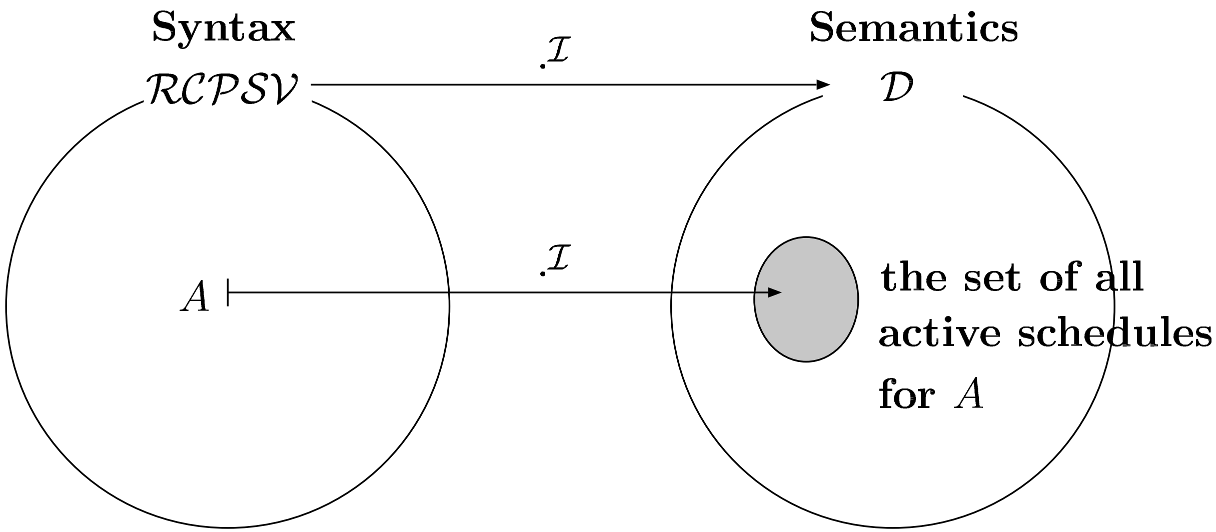
$$\begin{aligned}(\textbf{xor}\quad & \textbf{hnet}[let\, t_1 = (1, a, 2), t_2 = (\textbf{xor}(2, b, 1), (3, c, 2)); (t_1, t_2)], \\ & \textbf{hnet}[let\, k_1 = (0, eu, 0), k_2 = (4, d, 2), k_3 = (5, d, 2); (k_1, k_2), (k_1, k_3)])\end{aligned}$$

Semantics of the Language $\mathcal{RCPSV}$ - Definition

Interpretation $\mathcal{I} = (\mathcal{D}, \cdot^{\mathcal{I}})$

$\mathcal{D}$: the set consisting of all active schedules derived from activity terms in $\mathcal{RCPSV}$

$\cdot^{\mathcal{I}}$: the interpretation function of $\mathcal{I}$



The $\mathcal{RCPSV}$-Calculus

Calculus that may be used to transform any $\mathcal{RCPSV}$-expression t into a semantically equivalent *normalized* $\mathcal{RCPSV}$-expression s.

$t_1, t_2, \cdots, t_k, s_1, \cdots, s_l, t_{k+2}, \cdots, t_n$: activity-terms

The calculus has the associative rule (6) and distributive rules (7) and (8).

$$\frac{(\mathbf{xor}\, t_1, t_2, \cdots, t_k, (\mathbf{xor}\, s_1, s_2, \cdots, s_l), t_{k+2}, t_{k+3}, \cdots, t_n)}{(\mathbf{xor}\, t_1, t_2, \cdots, t_k, s_1, s_2, \cdots, s_l, t_{k+2}, t_{k+3}, \cdots, t_n)} \tag{6}$$

$$\frac{\begin{array}{r}(\mathbf{hnet}[let\, n_1 = t_1, \cdots, n_{k+1} = (\mathbf{xor}\, s_1, s_2, \cdots, s_l), \cdots, n_n = t_n; (n_{11}, n_{12}), \cdots, (n_{h1}, n_{k+1}),\\ \cdots, (n_{j1}, n_{j2})])\end{array}}{\begin{array}{r}(\mathbf{xor}\, \mathbf{hnet}[let\, n_1 = t_1, \cdots, n_{k+1} = s_1, \cdots, n_n = t_n; (n_{11}, n_{12}), \cdots, (n_{h1}, n_{k+1}), \cdots, (n_{j1}, n_{j2})],\\ \vdots\\ \mathbf{hnet}[let\, n_1 = t_1, \cdots, n_{k+1} = s_l, \cdots, n_n = t_n; (n_{11}, n_{12}), \cdots, (n_{h1}, n_{k+1}), \cdots, (n_{j1}, n_{j2})])\end{array}} \tag{7}$$

$$\frac{\begin{array}{r}(\mathbf{hnet}[let\, n_1 = t_1, \cdots, n_{k+1} = (\mathbf{xor}\, s_1, s_2, \cdots, s_l), \cdots, n_n = t_n; (n_{11}, n_{12}), \cdots, (n_{k+1}, n_{h2}),\\ \cdots, (n_{j1}, n_{j2})])\end{array}}{\begin{array}{r}(\mathbf{xor}\, \mathbf{hnet}[let\, n_1 = t_1, \cdots, n_{k+1} = s_1, \cdots, n_n = t_n; (n_{11}, n_{12}), \cdots, (n_{k+1}, n_{h2}), \cdots, (n_{j1}, n_{j2})],\\ \vdots\\ \mathbf{hnet}[let\, n_1 = t_1, \cdots, n_{k+1} = s_l, \cdots, n_n = t_n; (n_{11}, n_{12}), \cdots, (n_{k+1}, n_{h2}), \cdots, (n_{j1}, n_{j2})])\end{array}} \tag{8}$$

Rules (6), (7), (8)

Proving correctness of the $\mathcal{RCPS}V$-calculus

Lemma The $\mathcal{RCPS}V$-calculus is a correct calculus.

Theorem For any $\mathcal{RCPS}V$-term t all operators '$\mathbf{xor}$' in the interior of t always may be moved to the leftmost position, so that t is transformed to a semantically equivalent, normalized term s in which all nonredundant reduced terms derived from t are being combined by the uniquely occurring operator '$\mathbf{xor}$'.

Solution Algorithm $\mathcal{A}_{\mathcal{RCPSV}}$ Based on a Scan-Line Principle

Conclusion

- The Logic-based Languages $\mathcal{RSV}$ and $\mathcal{RCPSV}$ for Representing and Solving $\mathcal{RCPSV}$-Scheduling Problems
- Diagram-based Solution Algorithms