BỘ GIÁO DỤC
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

PHẠM VĂN ĐĂNG

ĐẠI SỐ STREAM DÙNG CHO KHOA HỌC PHÂN TÍCH DỮ LIỆU LỚN DẠNG LIVESTREAM

TÓM TẮT LUẬN ÁN TIẾN SĨ MÁY TÍNH

Ngành: Khoa học máy tính

Mã số: 9 48 01 01

Hà Nội - 2025

Công trình được hoàn thành tại: Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Người hướng dẫn khoa học:

1. **Người hướng dẫn 1:** PGS.TS. Phan Công Vinh, Trường Đại học Nguyễn Tất Thành, Thành phố Hồ Chí Minh
2. **Người hướng dẫn 2:** TS. Trần Trọng Toàn, Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Phản biện 1: GS.TS. Huỳnh Xuân Hiệp, Đại học Cần Thơ

Phản biện 2: TS. Nguyễn Như Sơn, Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Luận án được bảo vệ trước Hội đồng đánh giá luận án tiến sĩ cấp Học viện họp tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam vào hồi 08 giờ 30 phút, ngày 17 tháng 12 năm 2025.

Có thể tìm hiểu luận án tại:

1. Thư viện Học viện Khoa học và Công nghệ
2. Thư viện Quốc gia Việt Nam

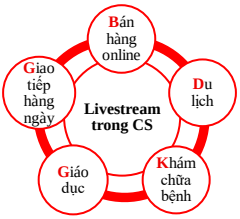
MỞ ĐẦU

1. Lý do chọn đề tài luận án

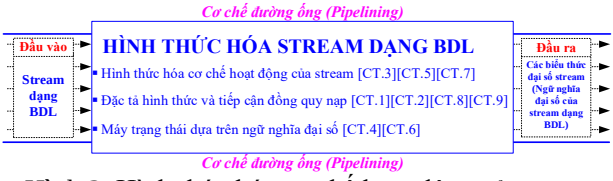
Dữ liệu lớn dạng livestream (BDL) tồn tại ở nhiều hình thức, hiện tại tác giả chưa tìm thấy nền tảng toán học nào định nghĩa, giải thích và chứng minh cơ chế hoạt động của BDL.

Livestream là hình thức phổ biến hỗ trợ nhiều lĩnh vực (Hình 1), tạo ra lượng lớn dữ liệu như văn bản, hình ảnh, âm thanh, video thời gian thực. Sự phát triển mạnh mẽ của livestream như vậy nhưng chưa có cơ sở lý thuyết nào giải thích một cách hình thức về cơ chế hoạt động của livestream.

Luận án xây dựng các phép toán stream, tổ hợp các phép toán này thành các biểu thức stream mới, các khía cạnh đại số, cơ chế lập lịch, mô hình ngữ nghĩa đại số để hình thức hóa cơ chế hoạt động của stream BDL (Hình 2).



Hình 1. Các hình thức livestream phổ biến tạo ra stream dạng BDL trong CS



Hình 2. Hình thức hóa cơ chế hoạt động của stream dạng BDL trong CS

2. Thách thức khoa học

Dựa vào mô tả và phân tích hình 1 và 2, và lý thuyết stream, luận án gặp phải những thách thức cho việc hình thức hóa stream dạng BDL như sau:

- Thách thức về tính trừu tượng của mô hình hóa stream dạng BDL.
- Thách thức về phân tích tính chất của mô hình hóa stream dạng BDL.
- Thách thức về khả năng kiểm chứng hình thức stream dạng BDL.
- Thách thức về tính chất tự nhiên của stream dạng BDL.

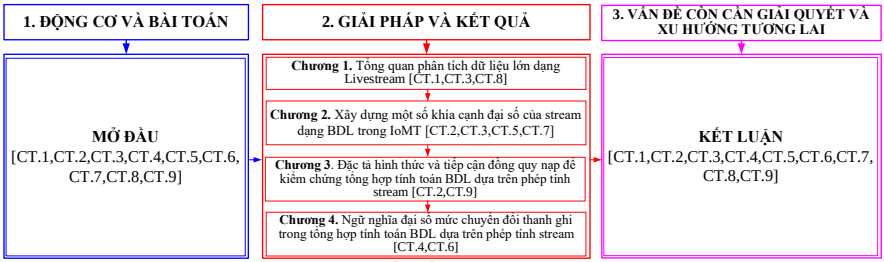
3. Mục tiêu nghiên cứu

Mục tiêu nghiên cứu của luận án là xây dựng một số khía cạnh đại số, tối ưu hóa lập lịch, và mô hình ngữ nghĩa đại số (ASM) để hình thức hóa (mô hình hóa, phân tích tính chất và kiểm chứng hình thức) cơ chế hoạt động của stream nhằm đáp ứng ngữ nghĩa đại số của stream dạng BDL trong CS.

4. Những đóng góp mới của luận án

- Xây dựng mười phép toán xử lý stream, tổ hợp các phép này thành các biểu thức stream mới, mở rộng đại số stream và đồng đại số stream, cấu trúc đại số monoid, và tính chất monoid dùng cho stream dạng BDL trong IoMT [CT.1][CT.2][CT.3][CT.5][CT.7][CT.8].
- Xây dựng cơ chế lập lịch xử lý stream dạng BDL và tối ưu hóa lập lịch theo cơ chế đường ống dùng cho stream BDL [CT.2][CT.9].
- Xây dựng mô hình ngữ nghĩa đại số (ASM) để đặc tả mối quan hệ giữa ASA_{RTL} và ASA_{SPEC} trong kiểm chứng kết quả tổng hợp RTL và SPEC [CT.4][CT.6].

5. Cấu trúc của luận án



Hình 3. Sơ đồ cấu trúc của luận án

CHƯƠNG 1. TỔNG QUAN PHÂN TÍCH DỮ LIỆU LỚN DẠNG LIVESTREAM

1.1. Phát sinh dữ liệu lớn trực tuyến

Dữ liệu là trung tâm thúc đẩy nghiên cứu và đổi mới, xoay quanh quy trình, thuật toán, mô hình toán học và hình thức hóa. Sự phổ biến của thiết bị thông minh, mạng xã hội và tài nguyên web làm gia tăng dữ liệu lớn trực tuyến, đặc biệt là dữ liệu lớn dạng livestream (BDL).

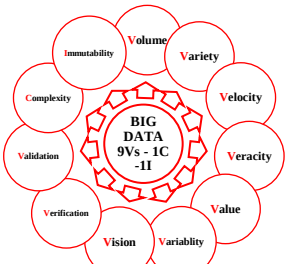
1.2. Một số tính chất đặc trưng của dữ liệu lớn

Một số thách thức kỹ thuật và công nghệ liên quan đến mười một tính chất đặc trưng của dữ liệu lớn, gồm: 9Vs - 1C - 1I [CT.1] (Hình 1.1).

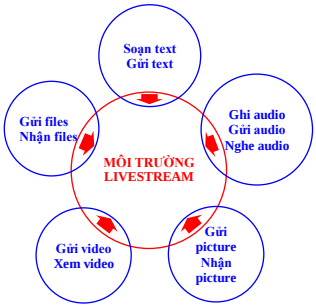
1.3. Hoạt động livestream

Phân tích stream dạng BDL xử lý dữ liệu liên tục, phù hợp với các nguồn đa dạng (dữ liệu có cấu trúc, bán cấu trúc, phi cấu trúc) và truyền theo

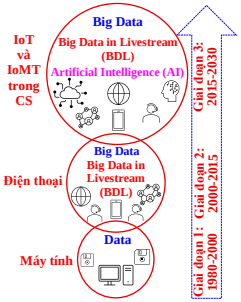
thời gian thực. Livestream là phương thức phổ biến hiện nay (Hình 1.2).



Hình 1.1. Một số tính chất đặc trưng của dữ liệu lớn 9Vs – 1C – 1I



Hình 1.2. Các hoạt động livestream trên các nền tảng mạng xã hội



Hình 1.6. Lịch sử phát triển của các thiết bị tạo dữ liệu [CT.7]

1.4. Dữ liệu lớn dạng livestream (BDL)

Sự phát triển của thiết bị điện tử gắn liền với nguồn dữ liệu lớn qua từng giai đoạn (Hình 1.6) [CT.7]. BDL trong BD đối mặt với thách thức lý thuyết và kỹ thuật từ 11 tính chất ở hình 1.1, và BDL được xem như dãy frame tuyến tính gồm văn bản, hình ảnh, âm thanh, video, v.v. [CT.1][CT.2].

1.5. Sự khác nhau và giống nhau giữa BD và BDL

Bảng 1.1. So sánh sự khác nhau và giống nhau giữa BD và BDL

BD		Tính chất	BDL	
♦		7**V**, 1**C**, 1**I**	♦	
Stream trừu tượng	✓	**V**ariablity	• Stream biến đổi nhanh thời gian thực • Stream lớn thời gian thực • Stream có tính biến động • Stream có tính xen kẽ các frame	✓
		Vision	• Stream trực quan online • Stream đáp ứng tính cảm nhận của thị giác • Stream có tính biến động thời gian thực • Stream đáp ứng tính ra quyết định	✓
Ghi chú: ✓ là thể hiện sự khác nhau; ♦ là thể hiện sự giống nhau				

1.6. Cơ chế đường ống trong xử lý stream dạng BDL

1.7. Lý thuyết stream dùng cho stream dạng BDL

1.7.1. Các khái niệm cơ bản

a) Dữ liệu frame trong stream dạng BDL

Trong [CT.1][CT.2], các ký hiệu và kiến thức cơ bản về stream dạng BDL được định nghĩa như sau: Các ký hiệu $\sigma, \tau, \rho, \alpha, ...$ biểu diễn stream, với

$\sigma \in \mathbb{A}^{\omega}$ là dãy $(\sigma(0), \sigma(1), \ldots)$ và đạo hàm $\sigma' = (\sigma(1), \sigma(2), \ldots)$. Các phép toán gồm: drop, take, zip, split, rev, merge, $\times$, $\bullet$, $\blacksquare$, và $:=$. Với $\mathbb{A}^{\omega}$ là tập các stream, còn $\mathbb{A}$ là tập các frame (văn bản, hình ảnh, âm thanh, video, v.v.).

b) Cơ chế lập lịch (Scheduling)

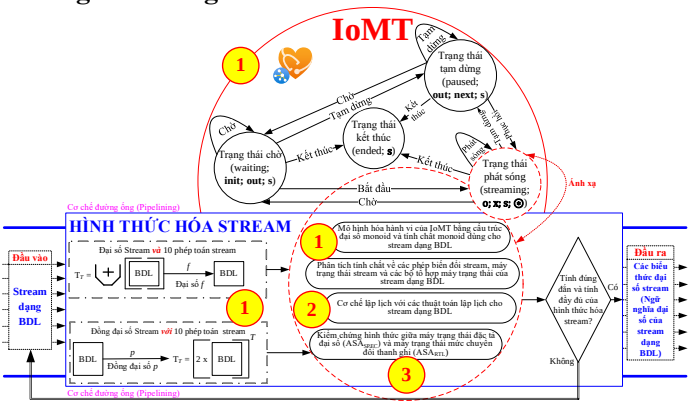
c) Bước điều khiển (Control step - CStep)

1.7.2. Thế mạnh và hạn chế của đại số stream

1.7.3. Thế mạnh và hạn chế của đồng đại số stream

1.7.4. Vai trò của mười phép toán xử lý stream dạng BDL

1.8. Đặt những vấn đề nghiên cứu

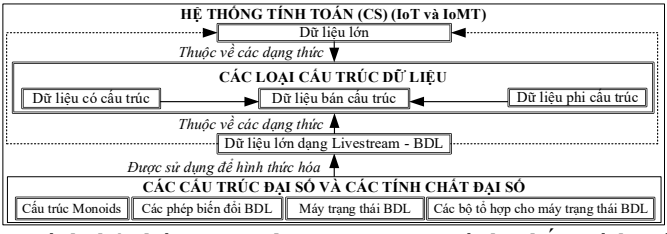


Hình 1.10. Khung hình thức hóa stream thể hiện 03 vấn đề nghiên cứu của luận án về hình thức hóa cơ chế hoạt động của stream dạng BDL trong IoMT

CHƯƠNG 2. XÂY DỰNG MỘT SỐ KHÍA CẠNH ĐẠI SỐ CỦA STREAM DẠNG BDL TRONG IoMT

2.1. Giới thiệu hệ thống tính toán

Sự bùng nổ thiết bị điện tử tạo ra dữ liệu lớn, gồm stream BDL có tính chất phức tạp (Hình 1.1, 1.6). Dựa trên đặc trưng stream (Tiểu mục 1.2, Hình 1.1, 2.1) và hạn chế hình thức hóa (Hình 1), luận án xây dựng các khía cạnh đại số để hình thức hóa hoạt động stream, đáp ứng ngữ nghĩa đại số cho BDL.



Hình 2.1. Hình thức hóa stream dạng BDL trong các hệ thống tính toán (CS)

2.2. Lý thuyết stream

2.2.1. Đại số stream dùng cho phân tích stream dạng BDL

Stream dạng BDL gồm có giá trị khởi đầu $\sigma(0)$, đạo hàm stream σ', hành vi và phương trình vi phân. Luận án xây dựng 10 phép toán xử lý stream (drop, take, zip, split, rev, merge, $\times$, $\bullet$, $\blacksquare$, và $:=$) từ tiểu mục 2.3.3.1–2.3.3.10 và tổ hợp chúng thành biểu thức stream ở tiểu mục 2.3.4 của chương 2.

Phép toán xử lý stream một ngôi $\mathbb{A}^\omega \to \mathbb{A}^\omega$, ký hiệu là $\boxdot \in \{drop, take, zip, split, rev\}$ bằng phương trình sau: $\boxdot: \sigma \in \mathbb{A}^\omega \to \boxdot\ (\sigma) \in \mathbb{A}^\omega$. Phép toán xử lý stream hai ngôi $\mathbb{A}^\omega \times \mathbb{A}^\omega \to \mathbb{A}^\omega$, ký hiệu là $\circledast \in \{+, \times, :=\}$ bằng phương trình sau: $\circledast: \sigma \in \mathbb{A}^\omega \times \tau \in \mathbb{A}^\omega \to \sigma \circledast \tau \in \mathbb{A}^\omega$ [CT.2]. Hàm stream được luận án đề xuất là $\sigma: \mathbb{N} \to \mathbb{A}^\omega$, với $\sigma = (\sigma(0),\ \sigma(1), \dots)$, $\mathbb{A}^\omega$ là tập các stream và $\mathbb{A}$ là tập các frame [CT.1][CT.2].

Định lý 2.1: $\forall \sigma \in \mathbb{A}^\omega$, $\mathbb{A}^\omega$ là tập các stream, ta có phương trình stream như sau: $\sigma = \sigma(0) + (X \times \sigma')$.

2.2.2. Tập các stream dùng cho phân tích BDL

$\mathbb{A}^\omega$ được định nghĩa trên $\mathbb{A}$, $\mathbb{A}^\omega = \{\sigma \mid \sigma: \mathbb{N} \to \mathbb{A}\}$ là tập các stream biểu diễn stream dạng BDL, với $\sigma = (\sigma(0), \sigma(1), \dots) \in \mathbb{A}^\omega$, giá trị khởi đầu $\sigma(0)$, đạo hàm $\sigma' = (\sigma(1), \sigma(2), \dots)$. Với mọi $n \geq 0$, đạo hàm bậc cao được định nghĩa: $\sigma^{(0)} = \sigma$ và $\sigma^{(n+1)} = \left(\sigma^{(n)}\right)'$ [CT.1][CT.2].

2.2.3. Đồng đại số stream dùng cho stream dạng BDL

Định nghĩa 2.14 (Máy trạng thái stream dạng BDL): Cho một tập hữu hạn hoặc vô hạn các phép toán stream ký hiệu T. Một stream dạng BDL trong IoMT là một máy trạng thái stream có kiểu $\langle IoMT, \langle o_{IoMT}, e_{IoMT}\rangle\rangle$ bao gồm T, một tập các trạng thái IoMT, một hàm đầu ra o_{IoMT}: $IoMT \to (T \to 2)$ và một hàm tiến hóa e_{IoMT}: $IoMT \to (T \to IoMT)$.

Định nghĩa 2.15 (Đồng đại số stream dạng BDL): Cho tập các stream ký hiệu $\{\langle IoMT, \langle o_{IoMT}, e_{IoMT}\rangle\rangle\}$ dạng BDL của tất cả các vật mang stream hữu hạn hoặc vô hạn có một cấu trúc đồng đại số gồm một cặp phép toán $\langle O, E\rangle$ dạng $\{\langle IoMT, \langle o_{IoMT}, e_{IoMT}\rangle\rangle\} \xrightarrow[\langle O,E\rangle]{} o_{IoMT} \times \{\langle IoMT, \langle o_{IoMT}, e_{IoMT}\rangle\rangle\}$ và phần tử stream $\langle IoMT, \langle o_{IoMT}, e_{IoMT}\rangle\rangle \mapsto \langle \langle IoMT, \langle o_{IoMT}, e_{IoMT}\rangle\rangle(0), \langle IoMT, \langle o_{IoMT}, e_{IoMT}\rangle\rangle'\rangle$.

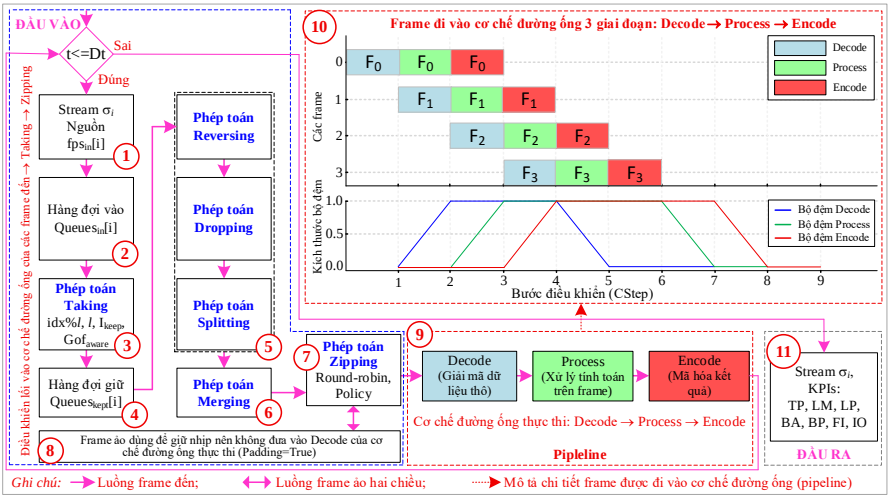
2.2.4. Bài toán mô phỏng và kiểm chứng đánh giá các phép toán stream

a) Đặt bài toán cho các phép toán cần kiểm chứng

Đặt bài toán: Cho biết một stream σ_i có nhiều frame (Mỗi frame là một bộ ⟨text, image, audio, video, v.v⟩) đầu vào sáu phép toán stream (Take, Reverse, Drop, Split, Merge, và Zip). Luận án đề xuất kịch bản sau đây:

Kịch bản: Kiểm chứng ngữ nghĩa bảo toàn tính đồng nhất của stream gốc, giữ tập con, và lượng hóa các chỉ số hiệu năng chính KPIs (TP, LA, LP, BA, BP, FI, IO) khi biết sáu phép toán stream. Tức là, cho biết sáu phép toán xử stream, luận án kiểm chứng ngữ nghĩa và lượng hóa các KPIs, với một stream σ_i có nhiều frame.

b) Phân tích đầu vào và ra cho thuật toán mô phỏng các phép toán



Hình 2.10. Sơ đồ điều khiển lối vào Pipeline của các frame đến

Hình 2.10 mô phỏng cơ chế điều khiển lối vào pipeline gồm 11 khối: Khối ① các nguồn sinh stream σ_i; Khối ② đưa frame vào $\text{Queues}_{\text{in}}[i]$; Khối ③ phép toán taking chọn frame theo nguồn; Khối ④ lưu frame được chấp nhận vào $\text{Queues}_{\text{kept}}[i]$; Khối ⑤ có phép toán reversing, dropping, splitting; Khối ⑥ phép toán merging trộn stream; Khối ⑦ phép toán zipping nén đa nguồn trước pipeline; Khối ⑧ padding tạo frame ảo khi nguồn rỗng; Khối ⑨–⑩ pipeline thực thi nhận các frame F_0–F_3 mỗi bước thời gian. Ba lớp chính Taking ③, Zipping ⑦, và Pipeline ⑨ phối hợp chi phối Khối ⑪, nơi hình thành các KPIs (TP, LM, LP, BA, BP, FI và IO).

c) Thuật toán mô phỏng các phép toán xử lý stream

d) Vai trò của các phép toán xử lý stream trong kịch bản mô phỏng

e) Dữ liệu cho các tham số cấu hình của các kịch bản

Luận án cung cấp dữ liệu các tham số cấu hình mô phỏng ở bảng 2.23. Môi trường giả lập: colab.research.google.com và ngôn ngữ lập trình Python.

Bảng 2.23. Dữ liệu của các tham số cấu hình cho từng kịch bản

Kịch bản	Các tham số cấu hình cho 6 phép toán xử lý stream												
	Take và Zip							Rev	Drop		Split		Merge
	T	M	fps_{in}	l	I_{keep}	Gof_{aware}	P	l	l	k	l	k	Use
KB1	5	1	[60]	4	[0,1,2,3]	False	Skip	5	8	2	8	[0,2]	False
KB2	5	2	[60,45]	4	[1,3]	False	Skip	4	4	2	4	[0,2]	True
KB3	5	3	[60,45,30]	4	[1,3]	True	Skip	3	3	2	3	[0,2]	True
KB4	5	4	[60, 45, 30, 20]	4	[0,2]	False	Skip	5	5	2	5	[0,2]	True
KB5	5	5	[60, 45, 30, 20, 15]	4	[1,3]	False	Skip	6	6	2	6	[0,2]	True
KB6	5	5	[60, 30, 45, 15, 10]	4	[1,3]	False	Skip	7	7	2	7	[0,2]	True

f) Chạy mô phỏng và thu thập kết quả KPIs

Lấy các tham số cấu hình ở bảng 2.23 để chạy mô phỏng thu được kết quả lượng hóa KPIs. Kết quả mô phỏng của 6 kịch bản được trình bày chi tiết ở bảng 2.24.

Bảng 2.24. Các KPIs so sánh giữa các KPIs của các kịch bản khi chạy mô phỏng

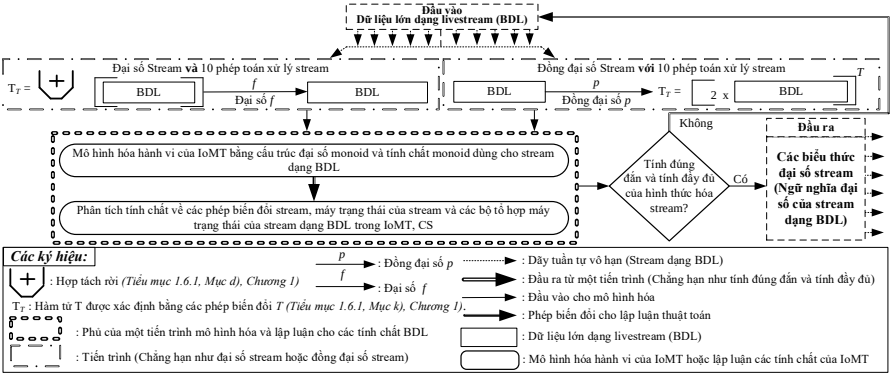
Kịch bản	Các chỉ số hiệu năng (KPIs) được lượng hóa dựa vào tham số cấu hình						
	TP	LM	LP	BA	BP	FI	IO
KB1	60.0	35.0000	35.0	2.0949	3.0	1.0000	0.0
KB2	52.4	36.2023	40.0	1.8921	5.0	0.9794	0.5
KB3	77.4	37.2739	45.0	2.8551	5.0	0.9276	0.5
KB4	65.0	38.3692	45.0	2.3926	5.0	0.7958	0.5
KB5	84.8	40.0825	50.0	3.2927	6.0	0.8073	0.5
KB6	79.8	38.7093	45.0	3.0200	6.0	0.7463	0.5

Phân tích kết quả KPIs của bảng 2.24 cho thấy rằng: KPIs phản ánh rõ tác động từng chính sách cho taking và zipping lên hiệu quả vận hành của pipeline. Cụ thể, đặt 5 *câu hỏi/trả lời* để phân tích sự thay đổi trong cấu hình mô phỏng làm thay đổi đáng kể TP, LM, LP, IO cũng như BA và BP.

2.3. Khung hình thức hóa dùng cho stream dạng BDL trong IoMT

Hình thức hóa stream dạng BDL gặp thách thức trong đại số stream và đồng đại số stream. Luận án xây dựng các khía cạnh đại số để mô hình hóa cơ chế hoạt động của stream [CT.1][CT.2][CT.3][CT.5][CT.7]. Tính toán stream dưới dạng một cấu xạ $IoMT^{\omega} \rightarrow IoMT^{\omega}$ để tạo ra cơ chế đường ống

nhằm đáp ứng ngữ nghĩa đại số của stream BDL trong IoMT (Hình 2.11).



Hình 2.11. Khung hình thức hóa dùng cho stream dạng BDL trong IoMT

2.4. Cấu trúc monoid của BDL trong IoMT

2.4.1. Cấu trúc monoid dùng cho stream dạng BDL trong IoMT

Áp dụng phép toán stream trong cấu trúc monoid là tiếp cận nền tảng để xử lý stream BDL. Luận án ký hiệu $\circledast$ cho các phép toán stream hai ngôi, dùng để mô hình hóa tiến trình tự động xử lý từng frame trong IoMT [CT.2].

Định nghĩa 2.16 (Cấu trúc monoid): Một monoid là một bộ ba $(IoMT, \circledast, 1)$, ở đó $IoMT$ là một tập hợp các trạng thái hữu hạn hoặc kiểu các stream dữ liệu, cụ thể là BDL. $\circledast: IoMT \times IoMT \to IoMT$ là một phép toán hai ngôi trên $IoMT$, gọi là phép nối các frame trong bối cảnh stream. $1 \in IoMT$ là phần tử đơn vị, sao cho hai tiên đề (2.23) và (2.24) thỏa mãn:

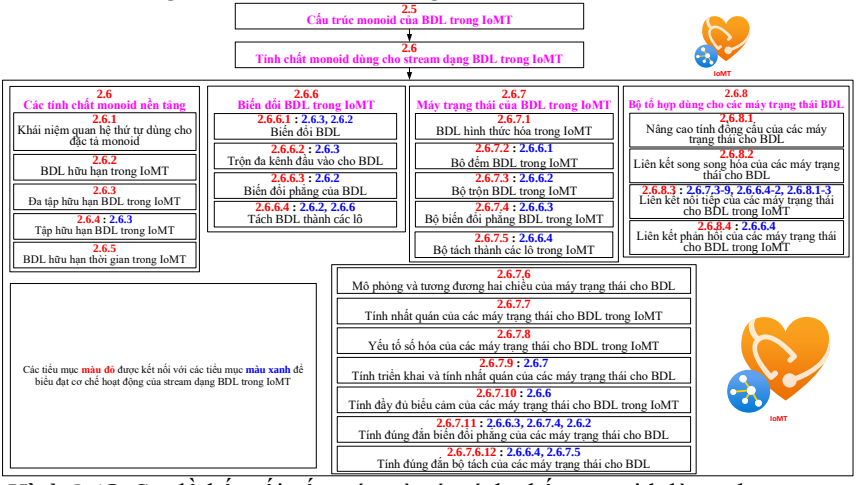
- Tính chất kết hợp: $(x \circledast y) \circledast z = x \circledast (y \circledast z), \forall x, y, z \in IoMT$ (2.23)
- Tính chất đồng nhất: $1 \circledast x = x \circledast 1 = x, \forall x \in IoMT.$ (2.24)

Định nghĩa 2.17 (Tích hai monoid): Cho $(IoMT_{1,2}, \circledast_{IoMT_{1,2}}, 1_{IoMT_{1,2}})$ là hai cấu trúc monoid. Khi đó, tích của hai monoid này là một monoid $(IoMT_1 \times IoMT_2, \circledast, 1)$, trong đó: phép toán stream $\circledast$ được cho bởi: $(x, y) \circledast (x', y') = (x \circledast_{IoMT_1} x', y \circledast_{IoMT_2} y')$, $\forall x, x' \in IoMT_1$ và $\forall y, y' \in IoMT_2$, và phần tử đơn vị 1 là $1 = (1_{IoMT_1}, 1_{IoMT_2})$.

Định nghĩa 2.18 (Đồng cấu monoid cho stream): Cho $(IoMT_{1,2}, \circledast, 1)$ là hai monoid. Một ánh xạ $h: IoMT_1 \to IoMT_2$ được gọi là một đồng cấu monoid nếu thỏa mãn hai điều kiện: Bảo toàn phần tử đơn vị: $h(1) = 1$ và bảo toàn phép toán stream: $h(x \circledast y) = h(x) \circledast h(y), \forall x, y \in IoMT_{1,2}$.

2.4.2. Hình thức hóa stream dạng BDL trong IoMT

Luận án mô tả quan hệ cấu trúc và tính chất monoid để hình thức hóa cơ chế hoạt động của stream BDL trong IoMT, được mô tả ở sơ đồ hình 2.13.



Hình 2.13. Sơ đồ kết nối cấu trúc và các tính chất monoid dùng cho stream

- Monoid (IoMT, $\circledast$, 1) là cấu trúc đại số dùng để hình thức hóa cơ chế hoạt động của stream BDL trong IoMT, cùng với các tính chất được trình bày *ở tiểu mục 2.5.1 và 2.6, chương 2 của luận án.*
  - Dùng cho tập trạng thái hữu hạn (*Tiểu mục 2.6.7, chương 2 của luận án).*
  - Dùng cho kiểu các stream dữ liệu.
  - Dùng để mô tả cơ chế hoạt động của stream dạng BDL trong IoMT.
  - Ý nghĩa khoa học và thực tiễn:
    - Xử lý stream dữ liệu dạng BDL trong IoMT theo thời gian thực.
    - Duy trì livestream ngay cả khi có lỗi dữ liệu.
    - Giảm tải băng thông và tăng tốc độ xử lý.
    - Đảm bảo tính toàn vẹn và bảo mật dữ liệu trong IoMT.
  - Quan hệ thứ tự dùng cho đặc tả monoid *(Tiểu mục 2.6.1, chương 2 của luận án)*: Được áp dụng trong các tính chất của monoid thỏa mãn về thứ tự của frame, stream, trạng thái trong IoMT.
  - Tính chất monoid *(Tiểu mục 2.6.2-2.6.5, chương 2 của luận án)*: Đặc tả tính hữu hạn BDL trong tập các trạng thái hữu hạn của máy trạng thái.
  - Biến đổi BDL trong IoMT ($IoMT_{1,2}$) *(Tiểu mục 2.6.6, chương 2 của luận*

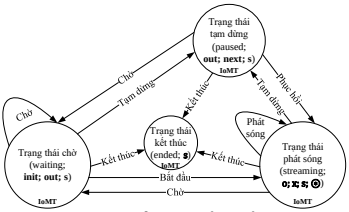
án): BDL được biến đổi từ tập trạng thái $IoMT_1$ sang tập trạng thái $IoMT_2$.

- Máy trạng thái của BDL trong IoMT *(Tiểu mục 2.6.7, trong chương 2 của luận án)*: Đề xuất các máy trạng thái, gồm: *Hình thức hóa*; *Bộ đếm BDL*; *Bộ trộn BDL*; *Bộ biến đổi phẳng BDL*; *Bộ tách BDL* thành các lô.
- 04 bộ tổ hợp dùng cho các máy trạng thái stream dạng BDL (*Tiểu mục 2.6.8, trong chương 2 của luận án)*: Mô tả tính nối tiếp, song song, trộn, và phản hồi của các máy trạng dùng cho frame hoặc stream dạng BDL.

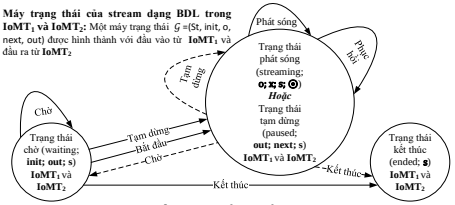
2.4.3. Máy trạng thái của stream dạng BDL trong IoMT

Luận án mô hình hóa cơ chế hoạt động và tính toán stream xử lý BDL trong IoMT bằng các ký hiệu (init, out, s, o, x, next, $\circledast$) trên sơ đồ trạng thái (Hình 2.14), với đầu vào và đầu ra là các frame thuộc hai monoid $(IoMT_1, \circledast, 1)$ và $(IoMT_2, \circledast, 1)$.

Luận án đề xuất máy trạng thái $\mathcal{G} = (\mathrm{St}, \mathrm{init}, \mathrm{o}, \mathrm{next}, \mathrm{out})$ để xử lý stream dạng BDL, bắt đầu từ $\mathrm{init} \in \mathrm{St}$, tạo frame đầu ra $\mathrm{o} \in IoMT_2$, và tại mỗi bước tiêu thụ dần các frame đầu vào $x \in IoMT_1$, phát sinh đầu ra $\mathrm{out}(s, x) \in IoMT_2$ và chuyển sang trạng thái tiếp theo $\mathrm{next}(s, x) \in \mathrm{St}$.



Hình 2.14. Sơ đồ chuyển đổi trạng thái của một hệ thống livestream (IoMT)



Hình 2.15. Sơ đồ chuyển đổi trạng thái cho hệ thống livestream (IoMT_1 và IoMT_2)

2.4.4. Mô phỏng và tương đương hai chiều của máy trạng thái BDL

Máy trạng thái $\mathcal{G} = (\mathrm{St}, \mathrm{init}, \mathrm{o}, \mathrm{next}, \mathrm{out})$ có đầu vào từ $IoMT_1$ và đầu ra từ $IoMT_2$ (Hình 2.15) với:

- Một mô phỏng hai chiều $R \subseteq \mathrm{St} \times \mathrm{St}$ cho $\mathcal{G}$, nếu với $\forall s, t \in \mathrm{St}$ và $x \in IoMT_1$, thì $(s, t) \in R$ chỉ ra $\mathrm{out}(s, x) = \mathrm{out}(t, x) \in R$ và $(\mathrm{next}(s, x), \mathrm{next}(t, x)) \in R$.
- sRt chỉ ra sự tương đương hai chiều $s \sim t$ nếu tồn tại một mô phỏng hai chiều R cho $\mathcal{G}$ sao cho sRt.
- $\forall s, t \in \mathrm{St}$ và $x \in IoMT$, $s \sim t$ đảm bảo $\mathrm{next}(s, x) \sim \mathrm{next}(t, x)$.

CHƯƠNG 3. ĐẶC TẢ HÌNH THỨC VÀ TIẾP CẬN ĐỒNG QUY NẠP ĐỂ KIỂM CHỨNG TỔNG HỢP TÍNH TOÁN BDL DỰA TRÊN PHÉP TÍNH STREAM

3.1. Giới thiệu

Livestream từ CS liên tục tạo ra stream dạng BDL thời gian thực, đòi hỏi sự hỗ trợ của công nghệ và kỹ thuật phân tích stream hiệu quả. Xử lý stream đóng vai trò quan trọng trong việc đặc tả các hệ thống dữ liệu lớn, tiếp cận lập lịch và kiểm chứng stream dạng BDL vẫn còn bỏ ngỏ.

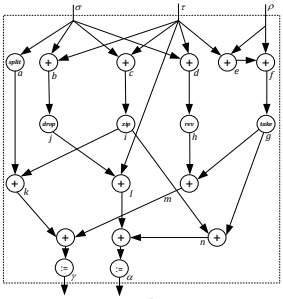
3.2. Đặc tả hình thức tính toán stream dạng BDL

Biểu diễn tính toán stream bằng phép toán stream cho phép đặc tả hiệu quả đường ống nhằm giảm độ trễ, tăng thông lượng, cân bằng tải và tối ưu tài nguyên. Đặc tả này là một hàm từ $\mathbb{A}^\omega$ đến $\mathbb{A}^\omega$, hỗ trợ biểu diễn các mạng RTL. Chương 2 đã phân loại các phép toán xử lý stream một ngôi $\mathbb{A}^\omega \rightarrow \mathbb{A}^\omega$ và hai ngôi $\mathbb{A}^\omega \times \mathbb{A}^\omega \rightarrow \mathbb{A}^\omega$ nhằm phân hoạch vào thuật toán 3.1.

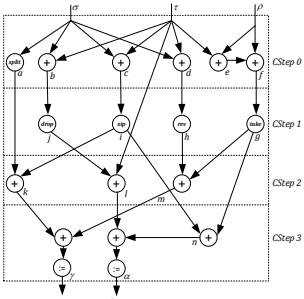
3.3. Mức chuyển đổi thanh ghi với thuật toán 3.1

Thuật toán 3.1 được dùng để khảo sát mức chuyển đổi thanh ghi, minh họa quá trình tổng hợp BDL qua ánh xạ ba stream đầu vào $\sigma, \tau, \rho \in \mathbb{A}^\omega$ thành hai đầu ra $\alpha, \gamma \in \mathbb{A}^\omega$.

3.3.1. Phân hoạch các phép toán stream dạng BDL



Hình 3.3. Sơ đồ stream phân hoạch phép toán stream



Hình 3.4. Sơ đồ phân hoạch phép toán stream cho tính toán BDL thành các CStep

Phép toán xử lý stream được phân hoạch vào sơ đồ stream (Hình 3.3), tương ứng với mã thuật toán 3.1, với các kết quả trung gian được gán tên trực tiếp trên sơ đồ gồm các stream: $a, b, c, d, e, f, g, h, i, j, k, l, m$, và n.

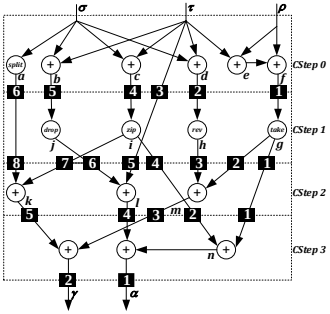
3.3.2. Lập lịch cho tính toán BDL

Lập lịch tính toán BDL dựa trên phép toán stream là tiến trình phân phối

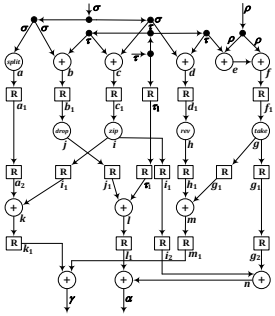
các phép toán stream vào các CStep được đánh số 0, 1, 2,..., k (Hình 3.4).

3.3.3. Cấp phát và liên kết thanh ghi

Trong cấp phát và liên kết thanh ghi gồm bước tổng hợp lập trình stream thực thi hai nhiệm vụ như sau (Hình 3.6): *Nhiệm vụ 1*, cấp phát là xác định thanh ghi cần để lưu trữ kết quả trung gian giữa 2 CStep; *Nhiệm vụ 2*, liên kết là ánh xạ các kết quả trung gian đến các thanh ghi cho mỗi CStep.



Hình 3.5. Sơ đồ cấp phát và liên kết thanh ghi của thuật toán 3.1



Hình 3.6. Sơ đồ tính toán stream biểu diễn cho RTL

Quá trình cấp phát và liên kết thanh ghi ảnh hưởng trực tiếp đến tài nguyên phần cứng, giúp tối ưu nếu thực hiện hiệu quả. Với các stream $\sigma, \tau, \rho, \alpha, \gamma \in \mathbb{A}^{\omega}$ (Hình 3.5), áp dụng định lý 2.1 thu được phương trình vi phân: $\alpha = X^3 \times (2\sigma + 4\tau + 2\rho)$ và $\gamma = X^3 \times (3\sigma + 3\tau + 2\rho)$.

Đặt stream $\mu = (2\sigma + 4\tau + 2\rho)$ và $\delta = (3\sigma + 3\tau + 2\rho)$ để tạo ra hai stream đầu ra là $\alpha = X^3 \times \mu$ và $\gamma = X^3 \times \delta$. Stream μ và δ được tính từ σ, τ, ρ, rồi dùng để tạo α và γ qua các phương trình vi phân như sau:

Phương trình vi phân stream	Giá trị ban đầu
$\alpha^{(3)} = \mu, \gamma^{(3)} = \delta$	$\alpha(0) = \alpha(1) = \alpha(2), \gamma(0) = \gamma(1) = \gamma(2)$

Sơ đồ stream ở hình 3.6 cũng được xem là sơ đồ stream của RTL trong hình 3.5. Nếu tính toán được lập lịch trong n CStep ($n \geq 0$), thì stream đầu ra α và γ được xác định: $\alpha = X^n \times \mu$ và $\gamma = X^n \times \delta$, hai stream đầu ra α và γ sau:

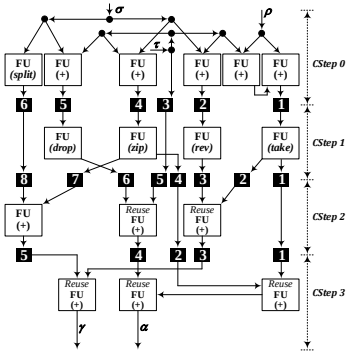
Phương trình vi phân stream	Giá trị ban đầu
$\alpha^{(n)} = \mu$	$\alpha^{(i)} = \mu(0), 0 \leq i \leq n-1$
$\gamma^{(n)} = \delta$	$\gamma^{(i)} = \delta(0), 0 \leq i \leq n-1$

Một quan hệ mô phỏng hai chiều B trên $\mathbb{A}^{\omega}$ là một quan hệ $B \subseteq \mathbb{A}^{\omega} \times \mathbb{A}^{\omega}$ sao cho $\forall \varphi, \beta \in \mathbb{A}^{\omega}$, nếu $\varphi \; B \; \beta$ thì $\varphi(0) = \beta(0)$ và $\varphi' \; B \; \beta'$.

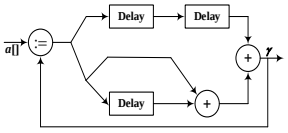
***Mệnh đề 3.1* (Mô phỏng hai chiều):** $\forall n \geq 0$, tất cả các stream đầu ra dạng BDL $\alpha = X^n \times \mu$ (hoặc $\gamma = X^n \times \delta$) đều là mô phỏng hai chiều theo cặp với stream α (hoặc với stream γ).

***Hệ quả 3.1* (Đồng quy nạp):** Với mọi stream đầu ra $X^n \times \mu$ trong $\mathbb{A}^\omega$, nếu $X^i \times \mu \sim X^j \times \mu$ $(0 \leq i, j \leq n)$, thì $X^i \times \mu = X^j \times \mu$. Nói cách khác, nếu hai stream đầu ra $X^i \times \mu$ và $X^j \times \mu$ là tương đương $\sim$, thì $X^i \times \mu$ và $X^j \times \mu$ cũng bằng nhau.

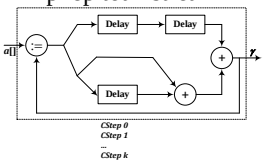
3.3.4. Cấp phát và liên kết đơn vị chức năng



Hình 3.7. Sơ đồ cấp phát và liên kết FUs



Hình 3.11. Sơ đồ tính toán BDL dựa trên phép toán stream



Hình 3.12. Sơ đồ lập lịch một CStep

Trong việc cấp phát và liên kết các đơn vị chức năng, ở đó một đơn vị chức năng (FU) thực hiện hai nhiệm vụ như sau:

- *Cấp phát là gán các phép toán xử lý stream vào từng CStep để thực thi.*
- *Liên kết là triển khai các phép toán xử lý stream trên các sơ đồ stream.*

FUs được tổ chức trong thư viện để ánh xạ chức năng với các phép toán stream, hỗ trợ đơn hoặc đa dụng qua tín hiệu điều khiển. Hình 3.7 minh họa 8 FU đơn dụng triển khai các phép toán: $drop, take, zip, split, rev, +, \bullet, \blacksquare$.

3.3.5. Bài toán tối ưu hóa lập lịch theo cơ chế đường ống tính toán stream

Đặt bài toán: Cho biết 3 stream $\sigma, \tau, \rho \in \mathbb{A}^\omega$ đầu vào hai biểu thức đại số stream của thuật toán 3.1: $\alpha := \big(zip(\sigma + \tau) + take\big(\rho + (\tau + \rho)\big) + (drop(\sigma + \tau) + \tau)\big)$ và $\gamma := \Big(split(\sigma) + zip(\sigma + \tau) + \Big(rev(\sigma + \tau) + take\big(\rho + (\tau + \rho)\big)\Big)\Big)$. Luận án đề xuất kịch bản sau đây:

Kịch bản: Tối ưu hóa lập lịch (Giảm độ trễ, tăng thông lượng, cân bằng tải, và tối ưu tài nguyên) theo cơ chế đường ống dựa trên quy hoạch tuyến

tính khi biết hai biểu thức đại số stream. Tức là, cho biết hai biểu thức đại số stream, luận án tối ưu hóa lập lịch, với 3 stream σ, τ, ρ đầu vào.

Môi trường giả lập: colab.research.google.com và ngôn ngữ lập trình Python.

d) Xây dựng mô hình tối ưu hóa lập lịch theo cơ chế đường ống

$$
\begin{cases}
C_{max} \geq S_k, \forall k = 1,2,3,\dots,13 \\
S_4 \geq S_1 \oplus 1 \\
S_5 \geq S_2 \oplus 1, S_5 \geq S_3 \oplus 2 \\
S_2 \geq S_3 \oplus 1 \\
S_6 \geq S_1 \oplus 1 \\
S_7 \geq S_6 \oplus 1 \\
S_9 \geq S_1 \oplus 1 \\
S_{10} \geq S_1 \oplus 1 \\
S_{11} \geq S_{10} \oplus 1, S_{11} \geq S_5 \oplus 1 \\
S_{12} \geq S_4 \oplus 1, S_{12} \geq S_5 \oplus 1, S_{12} \geq S_7 \oplus 1 \\
S_{13} \geq S_8 \oplus 1, S_{13} \geq S_9 \oplus 1, S_{13} \geq S_{11} \oplus 1 \\
|S_4 - S_9| \geq 1 \\
|S_9 - S_{10}| \geq 1 \\
|S_6 - S_{10}| \geq 1 \\
|S_6 - S_9| \geq 1 \\
|S_4 - S_{10}| \geq 1
\end{cases}
$$

Hình 3.10. Sơ đồ tối ưu hóa lập lịch theo cơ chế đường ống [CT.9]

LP-based PSOM được đặc tả qua các CStep ứng với biến S_k tại mỗi nút $k \in \mathbb{Z}^+$. Mục tiêu là tối thiểu hóa độ trễ $C_{max} = \max(\{S_1, S_2, \dots, S_{13}\})$, với $C_{max} \geq S_k$ và ràng buộc không chồng lấp $|S_i - S_j| \geq 1$ hoặc $|S_j - S_i| \geq 1$. Từ đây, luận án xác định được 29 ràng buộc phụ thuộc và 5 ràng buộc không chồng lấp cho LP-based PSOM.

e) Tối ưu hóa lập lịch theo cơ chế đường ống

f) Hiện thực mô hình tối ưu hóa lập lịch theo cơ chế đường ống

Luận án hiện thực LP-based PSOM để thu được sơ đồ tối ưu hóa lập lịch ở Hình 3.10 bằng thuật toán 3.2: Mỗi dòng là một phép toán stream S_i, bắt đầu tại CStep. Có 5 màu thể hiện nhánh: xám (dùng chung), xám nhạt (trung gian), xanh lam (nhánh α), cam (nhánh γ), xanh lá (dùng chung cho α và γ).

3.4. Mức chuyển đổi thanh ghi với thuật toán 3.3

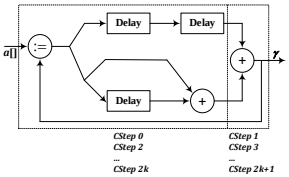
Tổng hợp tính toán BDL theo thuật toán 3.3 được minh họa qua mảng stream a, với $a[0] := drop(\sigma)$, $a[1] := zip(a[0])$, $a[2] := a[1] + a[0]$, và với $i \geq 3$, lặp: $a[i] := a[i-1] + a[i-2] + a[i-3]$, các stream này được ánh xạ đến đầu ra γ.

3.4.1. Phân hoạch các phép toán stream dạng BDL

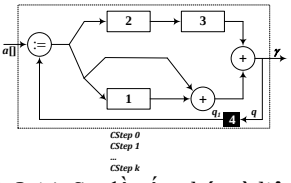
Sơ đồ tính toán BDL của thuật toán 3.3 (Hình 3.11) phân hoạch ba phép toán: $:=$, $+$, và độ trễ đơn vị, trong đó độ trễ đơn vị dùng ô nhớ lưu stream σ, phát ra $(n-1)^{th}$ giá trị và lưu giá trị n^{th} mới tại mỗi bước với $n \geq 1$.

3.4.2. Lập lịch cho tính toán BDL

Lập lịch tính toán BDL theo phép toán stream phụ thuộc vào tốc độ (số CStep) và tài nguyên phần cứng. Thuật toán 3.3 có thể lập lịch với một CStep (Hình 3.12) hoặc lập lịch với hai CStep (Hình 3.13).



Hình 3.13. Sơ đồ biểu diễn lập lịch cho hai CStep

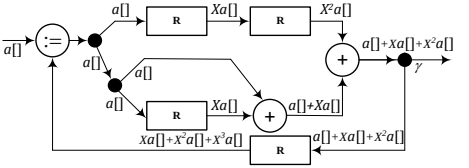


Hình 3.14. Sơ đồ cấp phát và liên kết thanh ghi cho việc lập lịch một CStep

3.4.3. Cấp phát và liên kết thanh ghi

Khi lập lịch một CStep, cần một thanh ghi bổ sung có nhãn số 4 để cấp phát và liên kết ở cuối mỗi CStep. Ba độ trễ đơn vị được thay thế bằng ba thanh ghi (có nhãn 1, 2, 3) trong hình 3.14 được mô tả như sau:

Phương trình vi phân stream	Giá trị ban đầu
$a^{(3)}[] = a^{(2)}[] + a^{(1)}[] + a[]$	$a[0] = drop(\sigma)$, $a^{(1)}[0] = zip(a[0])$ $a^{(2)}[0] = a^{(1)}[0] + a[0]$



Hình 3.15. Sơ đồ tính toán BDL trên phép tính stream cho việc lập lịch một CStep

Sơ đồ stream ở hình 3.15 cũng được xem như là một triển khai của thuật toán 3.3 và hàm hành vi xác định hành vi stream đầu ra γ.

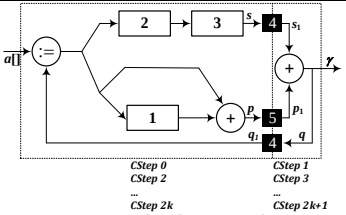
Phương trình vi phân hành vi	Giá trị ban đầu
$\gamma = a[] + X \times a[] + X^2 \times a[]$	$a[0] = drop(\sigma)$, $a^{(1)}[0] = zip(a[0])$ $a^{(2)}[0] = a^{(1)}[0] + a[0]$

Mệnh đề 3.2 **(Mô phỏng hai chiều):** Hai stream dạng BDL đầu ra $a[] + X \times a[] + X^2 \times a[]$ và $X \times a[] + X^2 \times a[] + X^3 \times a[]$ là tương đương theo mô phỏng hai chiều.

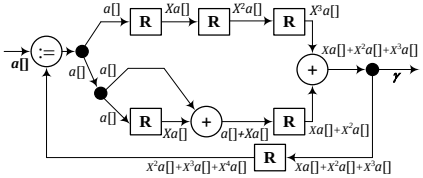
Hệ quả 3.2 **(Đồng quy nạp):** Nếu $(a[] + X \times a[] + X^2 \times a[]) \sim (X \times a[] + X^2 \times a[] + X^3 \times a[])$, thì $(a[] + X \times a[] + X^2 \times a[]) = (X \times a[] + X^2 \times a[] + X^3 \times a[])$.

Khi lập lịch hai CStep, ba đơn vị độ trễ được thay bằng ba thanh ghi (1, 2, 3). Hai thanh ghi bổ sung (4, 5) được cấp phát ở cuối mỗi CStep chẵn, trong đó thanh ghi 4 được tái sử dụng ở cuối mỗi CStep lẻ (hình 3.16). Phương trình vi phân hành vi của thuật toán 3.3 được mô tả như sau:

Phương trình vi phân hành vi	Giá trị ban đầu
$a^{(3)}[] = X \times \left(a^{(2)}[] + a^{(1)}[] + a[]\right)$	$a[0] = drop(\sigma),\ a^{(1)}[0] = zip(a[0])$ $a^{(2)}[0] = a^{(1)}[0] + a[0]$



Hình 3.16. Sơ đồ về việc cấp phát và liên kết thanh ghi cho lập lịch đôi



Hình 3.17. Sơ đồ tính toán BDL trên phép tính stream mô tả việc lập lịch đôi

Sơ đồ stream ở hình 3.17 cũng được xem như là một triển khai thuật toán 3.3 và xác định hành vi stream đầu ra γ dạng BDL như sau:

Phương trình vi phân hành vi	Giá trị ban đầu
$\gamma = X \times a[] + X^2 \times a[] + X^3 \times a[]$	$a[0] = \sigma,\ a^{(1)}[0] = a[0]$ $a^{(2)}[0] = a^{(1)}[0] + a[0]$

Thay thế $a[] = 1 + X + 2X^2 + \left(4X^4 + 3X^5 + 2X^6 \times Rev_4(1 - X^2 - X^3 - X^4)\right)$ cho phương trình vi phân này sẽ cung cấp hành vi stream đầu ra γ dưới dạng: $\gamma = 1 + X + 2X^2 + \left(4X^4 + 3X^5 + 2X^6 \times Rev_4(1 - X^2 - X^3 - X^4)\right) \times (X + X^2 + X^3)$.

Mệnh đề 3.3 **(Mô phỏng hai chiều):** Hai stream dạng BDL đầu ra: $X \times a[] + X^2 \times a[] + X^3 \times a[]$ và $X^2 \times a[] + X^3 \times a[] + X^4 \times a[]$ là tương đương theo mô phỏng hai chiều.

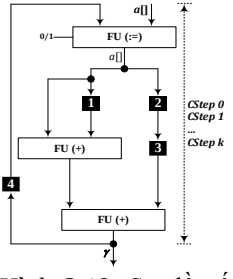
Hệ quả 3.3 **(Đồng quy nạp):** Nếu $(X \times a[] + X^2 \times a[] + X^3 \times a[]) \sim (X^2 \times a[] + X^3 \times a[] + X^4 \times a[])$, thì $(X \times a[] + X^2 \times a[] + X^3 \times a[]) = (X^2 \times a[] + X^3 \times a[] + X^4 \times a[])$.

3.4.4. Cấp phát và liên kết đơn vị chức năng

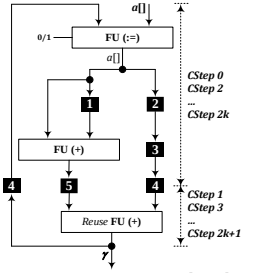
Với lập lịch CStep đơn (Hình 3.18), hai đơn vị chức năng thực hiện hai phép toán $+$ và một phép toán $:=$, điều khiển bằng tín hiệu $0/1$ để chọn

luồng xử lý. Nếu tín hiệu là 0, stream đầu vào của $a[]$ được xuất ra; nếu là 1, $a[]$ được gán stream đầu vào thứ hai. Hình 3.18 minh họa việc cấp phát và liên kết các đơn vị chức năng trong thuật toán 3.3.

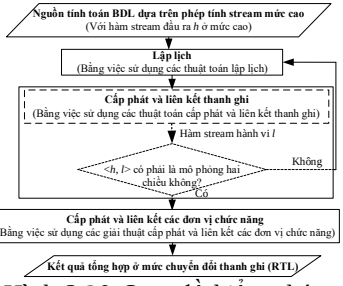
Với lập lịch CStep đôi (Hình 3.18), chỉ cần hai đơn vị chức năng: CStep chẵn thực hiện $+$ và $:=$, CStep lẻ tái sử dụng đơn vị trộn.



Hình 3.18. Sơ đồ cấp phát và liên kết FU cho việc lập lịch CStep đơn



Hình 3.19. Sơ đồ cấp phát và liên kết FU xử lý lập lịch CStep đôi



Hình 3.20. Lưu đồ kiểm chứng quá trình tổng hợp của tính toán BDL dựa trên phép tính stream

3.5. Tiếp cận đồng quy nạp để kiểm chứng việc tổng hợp tính toán BDL

3.5.1. Kiểm chứng tổng hợp

Kỹ thuật kiểm chứng quá trình tổng hợp BDL dựa trên phép tính stream được thực hiện sau khi chạy xong, nhằm tích hợp trực tiếp vào các bước tổng hợp. Khác với kiểm chứng sau tổng hợp, phương pháp trong Hình 3.20 dùng tiếp cận đồng quy nạp để đảm bảo tính đúng đắn.

3.5.2. Tiếp cận đồng quy nạp để kiểm chứng tổng hợp tính toán BDL

h	$\sim$	l
σ (hoặc τ)		$X^n \times \mu$ (hoặc $X^n \times \delta$), với $n \geq 0$
$a[] + X \times a[] + X^2 \times a[]$		$X \times a[] + X^2 \times a[] + X^3 \times a[]$
$X \times a[] + X^2 \times a[] + X^3 \times a[]$		$X^2 \times a[] + X^3 \times a[] + X^4 \times a[]$

Kỹ thuật kiểm chứng dùng quan hệ mô phỏng hai chiều giữa hàm stream đầu ra h từ nguồn tính toán BDL ở mức cao và hàm stream hành vi l sau cấp phát và liên kết thanh ghi (Hình 3.20). Mối quan hệ này, tích hợp trong quá trình tổng hợp, được minh họa qua các mệnh đề 3.1–3.3.

Suy ra, mối quan hệ mô phỏng hai chiều tổng quát giữa h và l như sau:

h với $m \in \mathbb{N}$	$\sim$	l với $n \in \mathbb{N}$
$X^m \times a[] + X^{m+1} \times a[] + X^{m+2} \times a[]$		$X^n \times a[] + X^{n+1} \times a[] + X^{n+2} \times a[]$

CHƯƠNG 4. NGỮ NGHĨA ĐẠI SỐ MỨC CHUYỂN ĐỔI THANH GHI TRONG TỔNG HỢP TÍNH TOÁN BDL DỰA TRÊN PHÉP TÍNH STREAM

4.1. Giới thiệu

Việc kiểm chứng tính đúng đắn trong xử lý stream trên dữ liệu livestream rất phức tạp, đặc biệt khi kiểm chứng máy trạng thái ASA_{RTL} và ASA_{SPEC} dựa trên ngữ nghĩa đại số trong quá trình tổng hợp.

4.2. Khái niệm không gian Chu

4.2.1. Không gian Chu

4.2.2. Ngữ nghĩa đại số như không gian Chu

Trong ngữ nghĩa đại số, Chu $C = (X, r, A)$, với $r(e, s) = 1$ nếu sự kiện e xảy ra tại trạng thái s, ngược lại $r(e, s) = 0$. Với mỗi $e \in A$, trạng thái $s_j \in 2^A$ được xác định bởi $s_j = \{e \mid r(e, s_j) = 1\}$.

Trong logic, ma trận được xem là bảng chân trị. Đặc tả công thức luận lý (logic) f_{ASM} của mô hình ngữ nghĩa đại số (ASM) được xác định bởi hàm: $f_{ASM} = \vee_{0 \le i \le n, 0 \le j \le l}(\wedge \{e_j \mid r(e_j, s_i) = 1\}) \wedge \{\wedge \bar{e}_j \mid r(\bar{e}_j, s_i) = 0\}$, với $n = |X| = |\{s_0, s_1, \dots, s_6\}|$, $l = |A| = |\{e_0, e_1, e_2\}|$, $e_j \in A$ và $\bar{e}_j$ là phần bù của e_j.

4.3. Các quan hệ nền tảng giữa trạng thái và sự kiện cho không gian Chu

4.4. Ngữ nghĩa đại số dùng cho RTL của thuật toán 4.2

4.4.1. Mô tả tập sự kiện

Định nghĩa 4.1 (Tính toán): Một dãy tuần tự các hành động hữu hạn được gọi là một quá trình tính toán trên tập các sự kiện A và tập hợp các tính toán được biểu diễn bằng ký hiệu $\mathcal{C}(A)$.

4.4.2. Mô tả tập trạng thái

Gọi X là tập trạng thái, mỗi $s \in X$ xác định qua quan hệ chuyển tiếp $T(e, s_i)$, thường viết $s_i \xrightarrow{e} s_j$ với $e \in A$. Khi đó, $s_j = \{e \mid e \in A: s_i \xrightarrow{e} s_j\}$. Bộ ba (A, X, T) tạo nên ngữ nghĩa đại số. Với mọi $e_i \in A$ và $s_i \in X$, mô hình $\text{ASM}(A, X, T)$ thỏa: $\mathcal{T} = \{e_1, e_2, \dots, e_n \mid s_1 \xrightarrow{e_1} s_2 \xrightarrow{e_2} s_3 \dots \xrightarrow{e_n} s_{n+1}\}$.

4.4.3. Thuật toán 4.2 dùng cho xử lý stream dạng BDL

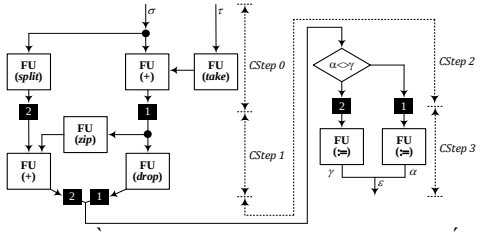
Chương này áp dụng thuật toán udStreamsA [CT.2] để phân phối sự kiện cho các phép toán stream (merge, drop, take, zip, split, assign) trong

một bước CStep để hình thành nên thuật toán 4.2. Quá trình tổng hợp ánh xạ các stream đầu vào $\sigma, \tau \in \mathbb{A}^{\omega}$ thành đầu ra $\alpha, \gamma, \epsilon \in \mathbb{A}^{\omega}$ của thuật toán 4.2.

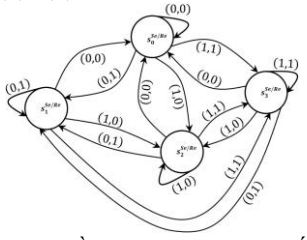
4.4.4. Máy trạng thái đại số của thuật toán 4.2

Định nghĩa 4.2 (Máy trạng thái đại số): Máy trạng thái đại số là bộ ba $\mathcal{A} = (X, s_0, T)$, với X là tập hữu hạn các trạng thái, s_0 là trạng thái khởi đầu, và T là một hàm chuyển tiếp từ $X \times A$ vào $X \cup \{\perp\}$.

Tập tính toán được chấp nhận bởi $\mathcal{A}$, ký hiệu $\mathcal{C}(\mathcal{A})$ (Định nghĩa 4.1), được minh họa qua bốn máy trạng thái đại số cho bốn CStep, mỗi máy trạng thái gồm có bốn trạng thái, trong các Hình 4.10(a)–(c).



(a) Sơ đồ của RTL cho thuật toán 4.2 với bốn CStep cùng với FUs và phép toán



(c) Sơ đồ máy trạng thái đại số cho thuật toàn 4.2 với 4 CStep

	Thanh ghi			Thanh ghi			Thanh ghi			Thanh ghi	
CStep 0	1	2	**CStep 1**	1	2	**CStep 2**	1	2	**CStep 3**	1	2
$0\left(s_0^{Se/Re}\right)$	0	0	$0\left(s_0^{Re/Se}\right)$	0	0	$0\left(s_0^{Re/Se}\right)$	0	0	$0\left(s_0^{Re/Se}\right)$	0	0
$1\left(s_1^{Se/Re}\right)$	0	1	$1\left(s_1^{Re/Se}\right)$	0	1	$1\left(s_1^{Re/Se}\right)$	0	1	$1\left(s_1^{Re/Se}\right)$	0	1
$2\left(s_2^{Se/Re}\right)$	1	0	$2\left(s_2^{Re/Se}\right)$	1	0	$2\left(s_2^{Re/Se}\right)$	1	0	$2\left(s_2^{Re/Se}\right)$	1	0
$3\left(s_3^{Se/Re}\right)$	1	1	$3\left(s_3^{Re/Se}\right)$	1	1	$3\left(s_3^{Re/Se}\right)$	1	1	$3\left(s_3^{Re/Se}\right)$	1	1

(b) Ma trận biểu diễn máy trạng thái đại số thuật toán 4.2 với 4 CStep

Hình 4.10. Biểu diễn máy trạng thái đại số cho thuật toán 4.2 với 4 CStep

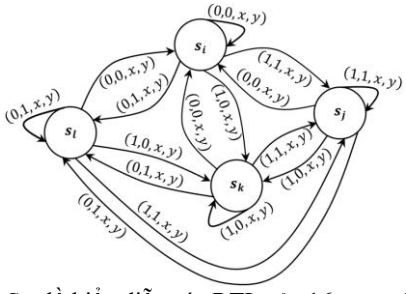
4.4.5. Tích máy trạng thái đại số của thuật toán 4.2

Luận án mô hình hóa hai tiến trình Se và Re bằng các máy trạng thái đại số gửi $\mathcal{S}e = \left(X^{Se}, s_p^{Se}, T^{Se}\right)$ và nhận $\mathcal{R}e = \left(X^{Re}, s_q^{Re}, T^{Re}\right)$ (Hình 4.11(a) và 4.11(b)). Tích $\mathcal{S}e \times \mathcal{R}e$ mô hình hóa việc kết nối giữa hai tiến trình, đảm bảo đồng bộ dữ liệu khi truyền qua thanh ghi. Tích này được định nghĩa trên tập sự kiện A là $\mathcal{S}e \times \mathcal{R}e = \left(X, s_p, T\right)$ với $X = X^{Se} \times X^{Re}$, $s_p = \left(s_p^{Se} \times s_q^{Re}\right)$, trong đó p, q có thể thay bằng i, j, k, hoặc l.

Quan hệ chuyển tiếp T được định nghĩa như sau: với $s_i = \left(s_i^{Se} \times s_j^{Re}\right) \in X$ và $e \in A$, nếu $T^{Se}\left(s_i^{Se}, e\right) = s_{i+1}^{Se}$ và $T^{Re}\left(s_j^{Re}, e\right) = s_{j+1}^{Re}$ thì $T\left(\left(s_i^{Se}, s_j^{Re}\right), e\right) = (s_{i+1}^{Se}, s_{j+1}^{Re})$; ngược lại, $T\left(\left(s_i^{Se}, s_j^{Re}\right), e\right) = \perp$.

	Các thanh ghi			
$\mathcal{S}e \times \mathcal{R}e$	1	2	3	4
$s_i = \left(s_i^{Se} \times s_j^{Re}\right)$	0	0	x	y
$s_j = \left(s_j^{Se} \times s_k^{Re}\right)$	0	1	x	y
$s_k = \left(s_k^{Se} \times s_l^{Re}\right)$	1	0	x	y
$s_l = \left(s_l^{Se} \times s_i^{Re}\right)$	1	1	x	y
Trong đó: $x, y \in \{0,1\}$ và $x \neq y$; $i, j, k, l = \overline{[0,3]} \subset \mathbb{N}^* = \mathbb{N} \cup (0)$				

(a) Ma trận này biểu diễn 16 trạng thái của 4 thanh ghi



(b) Sơ đồ biểu diễn các RTL của 16 trạng thái

Hình 4.11. Sơ đồ kết nối của tiến trình Se và Re và tích máy trạng thái đại số

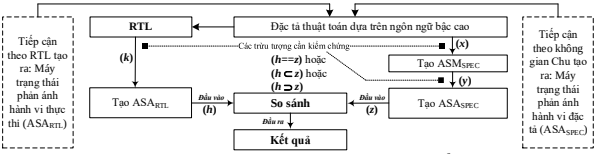
Nếu *đầu ra/đầu vào* của Se tương ứng với *đầu vào/đầu ra* của Re, hai tiến trình được kết nối như Hình 4.11: (a) ma trận 16 trạng thái của bốn thanh ghi và (b) sơ đồ RTL tương ứng. Mỗi trạng thái tích $\mathcal{S}e \times \mathcal{R}e$ là cặp $(\mathcal{S}e, \mathcal{R}e)$ và được chạy qua chuỗi tính toán $\mathcal{T} = \{(1,1,0,1)(1,1,1,0)(1,1,1,1)\}$ sau:

$$(s_3^{Se}, s_0^{Re}) \xrightarrow{(1,1,0,1)} (s_3^{Se}, s_1^{Re}) \xrightarrow{(1,1,1,0)} (s_3^{Se}, s_2^{Re}) \xrightarrow{(1,1,1,1)} (s_3^{Se}, s_3^{Re})$$

Tại trạng thái $\left(s_3^{Se}, s_0^{Re}\right)$ với sự kiện $(1,1,0,1)$, tích máy trạng thái $\mathcal{S}e \times \mathcal{R}e$ được thực thi bằng cách chạy đồng thời $\mathcal{S}e$ từ s_3^{Se} và $\mathcal{R}e$ từ s_0^{Re}, và tiếp tục lặp lại tương tự.

4.5. Thuật toán 4.2 cho các kết quả tổng hợp RTL

4.5.1. Các bước thuật toán kiểm chứng



Hình 4.12. Các bước của thuật toán kiểm chứng ASM

4.5.2. Đặc tả thuật toán dựa trên ngôn ngữ mức cao

Đặc tả thuật toán bằng ngôn ngữ mức cao là phù hợp để mô tả, dẫn đến việc tạo ra một chương trình. Nhiệm vụ chính trong quá trình này là cập nhật các kiểu lập lịch khác nhau. Trong *hình 4.14*, luận án trình bày *thuật toán 4.2* với bảy sự kiện e_0, e_1, e_2,…, e_7, mỗi sự kiện tương ứng với một câu lệnh.

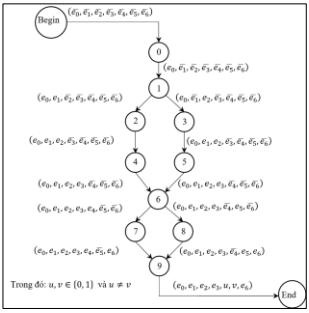
4.5.3. Tạo mô hình dựa trên ngữ nghĩa đại số (ASM)

Tạo đặc tả thứ tự cục bộ và quan hệ phụ thuộc dữ liệu từ thuật toán 4.2 nhằm xây dựng ASA$_{SPEC}$, dựa trên tiểu mục 4.3 và đặc tả hệ thống ngôn ngữ lập trình mức cao (tiểu mục 4.5.2) để tạo $\text{ASM}_{\text{SPEC}} = \{e_0 \prec e_1, e_0 \prec$

$e_2, e_1 \nabla e_2, e_1 \prec e_3, e_2 \prec e_3, e_3 \prec e_4, e_3 \prec e_5, e_4 \# e_5, \text{den}(e_6, e_4, e_5)\}$ (Hình 4.13).

ASA_{SPEC}	e_0	e_1	e_2	e_3	e_4	e_5	e_6	Công thức logic (ASM_{SPEC})
s_0	0	0	0	0	0	0	0	$f = f_{e_0<e_1} \wedge f_{e_0<e_2} \wedge f_{e_1 \nabla e_2} \wedge f_{e_1<e_3} \wedge f_{e_2<e_3} \wedge f_{e_3<e_4} \wedge f_{e_3<e_5} \wedge f_{e_4 \# e_5} \wedge f_{\text{den}(e_6,e_4,e_5)} = (e_0 + \bar{e_1}) \wedge (e_0 + \bar{e_2}) \wedge 1 \wedge (e_1 + \bar{e_3}) \wedge (e_2 + \bar{e_3}) \wedge (e_3 + \bar{e_4}) \wedge (e_3 + \bar{e_5}) \wedge (\bar{e_4} + \bar{e_5}) \wedge (\bar{e_6} + e_4 + e_5) = \bar{e_1}\bar{e_2}\bar{e_3}\bar{e_4}\bar{e_5}\bar{e_6} + e_0\bar{e_3}\bar{e_4}\bar{e_5}\bar{e_6} + e_0e_2\bar{e_3}\bar{e_4}\bar{e_5}\bar{e_6} + e_0e_1\bar{e_3}\bar{e_4}\bar{e_5}\bar{e_6} + e_0e_1e_2\bar{e_4}\bar{e_5}\bar{e_6}\ e_0e_1e_2e_3\bar{e_4}\bar{e_6} + e_0e_1e_2e_3\bar{e_4}e_5 + e_0e_1e_2e_3\bar{e_5}\bar{e_6}\ e_0e_1e_2e_3e_4\bar{e_5}$
s_1	1	0	0	0	0	0	0	
s_2	1	0	1	0	0	0	0	
s_3	1	1	0	0	0	0	0	
s_4	1	1	1	0	0	0	0	
s_5	1	1	1	1	0	0	0	
s_6	1	1	1	1	0	1	0	
s_7	1	1	1	1	0	1	1	
s_8	1	1	1	1	1	0	0	
s_9	1	1	1	1	1	0	1	

Hình 4.13. ASA_{SPEC} của thuật toán 4.2



Hình 4.14. ASA của thuật toán 4.2

4.6. Ngữ nghĩa đại số dùng cho RTL của thuật toán 4.4

4.6.1. Thuật toán 4.4 dùng cho stream dạng BDL

Thuật toán 4.4 phân bổ sự kiện cho các phép toán merge, drop, take, zip, split và assign trong một CStep. Quá trình tổng hợp BDL được minh họa qua ánh xạ các stream đầu vào $a[]$ và $\sigma \in \mathbb{A}^\omega$ thành một stream đầu ra $\gamma \in \mathbb{A}^\omega$.

4.6.2. Máy trạng thái đại số của thuật toán 4.4

Hình 4.16 minh họa tiến trình thực hiện các phép tính trên sáu CStep của thuật toán 4.4 qua máy trạng thái. Tập phép tính được chấp nhận, ký hiệu $\mathcal{C}(\mathcal{A})$ (Định nghĩa 4.1), thể hiện chi tiết trong Hình 4.16(a)–(c).

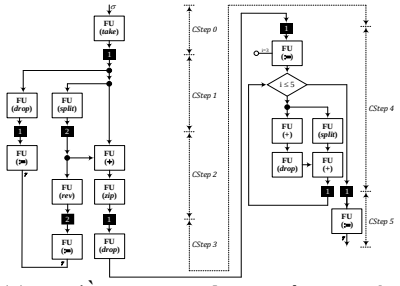
Hình 4.16(a) yêu cầu 15 FUs đơn dụng cho 10 phép toán ($split$, $take$, zip, rev, $drop$, $merge$, $copy$, $assign$, $\leq$, $register$) qua các CStep sau: CStep 0, thực hiện $take$ và lưu kết quả vào thanh ghi 1. CStep 1, thực hiện $copy$, $drop$, $split$, lưu vào thanh ghi 1 và 2. CStep 2, thực hiện $assign$, $copy$, $merge$, $reverse$, zip, lưu lại 1 và 2. CStep 3, thực hiện $assign$, $drop$, lưu vào thanh ghi 1. CStep 4, thực hiện $assign$, $\leq$, $merge$, $take$, $drop$, lưu vào thanh ghi 1. CStep 5, thực hiện $assign$ và gửi kết quả ra ngoài.

4.6.3. Tích máy trạng thái đại số của thuật toán 4.4

Hình 4.17(a) biểu diễn ma trận 32 trạng thái của 5 thanh ghi, Hình 4.17(b) là sơ đồ RTL tương ứng. Mỗi trạng thái trong $Se \times Re$ là cặp trạng thái từ Se và Re. Tích máy trạng thái $Se \times Re$ thực thi chuỗi tính toán $\mathcal{T} = \{(1,1,1,0,0)(1,1,1,0,1)(1,1,1,1,0)\}$ như sau:

$$(s_3^{Se}, s_0^{Re}) \xrightarrow{(1,1,1,0,0)} (s_3^{Se}, s_1^{Re}) \xrightarrow{(1,1,1,0,1)} (s_3^{Se}, s_2^{Re}) \xrightarrow{(1,1,1,1,0)} (s_3^{Se}, s_3^{Re})$$

Tại trạng thái (s_3^{Se}, s_0^{Re}) với sự kiện $(1,1,1,0,0)$, Se và Re chạy song song từ các trạng thái tương ứng và tiếp tục lặp lại như vậy.



(a) Sơ đồ RTL của thuật toán 4.4 với 6 CStep cùng với các FU, phép toán stream

$\mathbf{ASA_{RTL}}$	e_0	e_1	e_2	e_3	e_4	e_5	e_6
s_0	0	0	0	0	0	0	0
s_1	1	0	0	0	0	0	0
s_2	1	0	1	0	0	0	0
s_3	1	1	0	0	0	0	0
s_4	1	1	1	0	0	0	0
s_5	1	1	1	1	0	0	0
s_6	1	1	1	1	1	0	0
s_7	1	1	1	1	0	1	0
s_8	1	1	1	1	0	1	1
s_9	1	1	1	1	1	0	1
s_{10}	1	1	1	1	u	v	1
Trong đó: $u, v \in \{0,1\}$ và $u \neq v$							

Hình 4.15. $\mathrm{ASA_{RTL}}$ của thuật toán 4.2

	Thanh ghi			**Thanh ghi**	
CStep 0	1		CStep 1	1	2
$0\ (s_0^{Se/Re})$	0		$0\ (s_0^{Re/Se})$	0	0
$1\ (s_1^{Se/Re})$	1		$1\ (s_1^{Re/Se})$	0	1
			$2\ (s_2^{Re/Se})$	1	0
			$3\ (s_3^{Re/Se})$	1	1

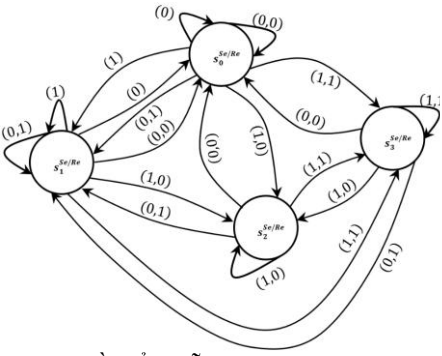
CStep 2	2	1
$0\ (s_0^{Re/Se})$	0	0
$1\ (s_1^{Re/Se})$	0	1
$2\ (s_2^{Re/Se})$	1	0
$3\ (s_3^{Re/Se})$	1	1

CStep 3	1
$0\ (s_0^{Re/Se})$	0
$1\ (s_1^{Re/Se})$	1

CStep 4	1
$0\ (s_0^{Re/Se})$	0
$1\ (s_1^{Re/Se})$	1

CStep 5	1
$0\ (s_0^{Re/Se})$	0
$1\ (s_1^{Re/Se})$	1

(b) Ma trận biểu diễn máy trạng thái đại số của thuật toán 4.4 với 6 CStep



(c) Sơ đồ biểu diễn máy trạng thái đại số các RTL với 6 CStep

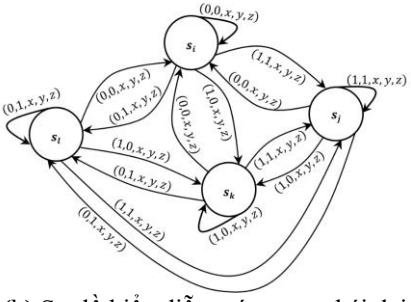
Hình 4.16. Biểu diễn máy trạng thái đại số cho thuật toán 4.4 với sáu CStep

4.6.4. Sự tương đương của các thuật toán

4.6.5. Ngữ nghĩa đại số của thuật toán 4.4

	Các thanh ghi				
$\mathcal{Se} \times \mathcal{Re}$	1	2	3	4	5
$s_i = (s_i^{Se} \times s_j^{Re})$	0	0	x	y	z
$s_j = (s_j^{Se} \times s_k^{Re})$	0	1	x	y	z
$s_k = (s_k^{Se} \times s_l^{Re})$	1	0	x	y	z
$s_l = (s_l^{Se} \times s_i^{Re})$	1	1	x	y	z
Trong đó: $x, y, z \in \{0,1\}$ và $x \neq y$ và $y \neq z$; $i, j, k, l = [\overline{0,3}] \subset \mathbb{N}^* = \mathbb{N} \cup (0)$					

(a) Ma trận này biểu diễn 32 trạng thái của năm thanh ghi



(b) Sơ đồ biểu diễn máy trạng thái đại số các RTL của 32 trạng thái

Hình 4.17. Kết nối các tiến trình Se và Re và tích máy trạng thái đại số

Máy trạng thái đại số được mô hình hóa bởi tập sự kiện A và quan hệ chuyển tiếp T, là tập các quan hệ giữa trạng thái mang ngữ nghĩa đại số. Sự

kết hợp các quan hệ này tạo thành máy trạng thái đại số.

4.7. Thuật toán 4.4 cho các kết quả tổng hợp RTL

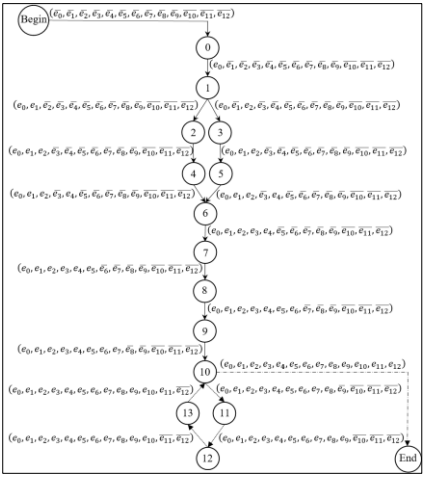
4.7.1. Đặc tả thuật toán dựa trên ngôn ngữ mức cao

4.7.2. Tạo mô hình dựa trên ngữ nghĩa đại số (ASM)

Sử dụng các quan hệ sự kiện để tạo ASM_{SPEC} cho thuật toán 4.4 ở ngữ cảnh mức cao có 13 sự kiện $e_0,\ldots, e_{12}$. Việc kết hợp quan hệ ở tiểu mục 4.3 đã tạo nên $\text{ASM}_{\text{SPEC}} = \{e_0 \prec e_1, e_0 \prec e_2, e_1 \nabla e_2, e_2 \prec e_3, e_0 \prec e_4, e_2 \prec e_4, e_3 \nabla e_4, e_4 \prec e_5, e_5 \nabla e_6, e_6 \prec e_7, e_7 \prec e_8, e_8 \prec e_9, e_6 \prec e_{10}, e_9 \prec e_{10}, e_{10} \prec e_{11}, e_7 \prec e_{11}, e_7 \prec e_{12}\}$ với máy trạng thái đại số hình 4.18.

ASA_{SPEC}	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	Công thức luận lý (ASM_{SPEC})
s_0	1	1	1	1	1	1	1	1	1	1	1	0	0	$f = f_{e_0 \prec e_2} \wedge f_{e_0 \prec e_2} \wedge f_{e_2 \prec e_3}$
s_1	1	1	1	1	1	1	1	1	1	1	1	1	0	$\wedge f_{e_0 \prec e_4} \wedge f_{e_2 \prec e_4}$
s_2	1	1	1	1	1	1	1	1	1	1	1	1	1	$\wedge f_{e_3 \nabla e_4} \wedge f_{e_4 \prec e_5}$
s_3	1	1	1	1	1	1	1	1	1	1	0	0	0	$\wedge f_{e_5 \nabla e_6} \wedge f_{e_6 \prec e_7}$
s_4	1	1	1	1	1	1	1	1	1	1	1	0	0	$\wedge f_{e_7 \prec e_8} \wedge f_{e_8 \prec e_9}$
s_5	1	1	1	1	1	1	1	1	1	0	0	0	0	$\wedge f_{e_6 \prec e_{10}} \wedge f_{e_9 \prec e_{10}}$
s_6	1	1	1	1	1	1	1	1	0	0	0	0	0	$\wedge f_{e_{10} \prec e_{11}} \wedge f_{e_7 \prec e_{11}}$
s_7	1	1	1	1	0	1	0	0	0	0	0	0	0	$\wedge f_{e_7 \prec e_{12}} =$
s_8	1	1	1	1	1	1	0	0	0	0	0	0	0	$= (e_0 + \overline{e_1}) \wedge (e_0 + \overline{e_2}) \wedge 1 \wedge (e_2 + \overline{e_3}) \wedge (e_0 + e_4) \wedge (e_2 + \overline{e_4}) \wedge 1 \wedge (e_4 + \overline{e_5}) \wedge 1 \wedge (e_6 + \overline{e_7}) \wedge (e_7 + \overline{e_8}) \wedge (e_8 + \overline{e_9}) \wedge (e_6 + \overline{e_{10}}) \wedge (e_9 + \overline{e_{10}}) \wedge (e_{10} + \overline{e_{11}}) \wedge (e_7 + \overline{e_{11}}) \wedge (e_7 + \overline{e_{12}}) =$
s_9	1	1	1	1	1	1	1	0	0	0	0	0	0	$e_0e_2e_4e_6e_7e_8e_9e_{10} +$
s_{10}	1	0	1	0	0	0	0	0	0	0	0	0	0	$e_0e_2e_4e_6e_7e_8\overline{e_9}e_{10}\overline{e_{11}} +$
s_{11}	1	1	1	0	0	0	0	0	0	0	0	0	0	$e_0e_2e_4e_6e_7e_8\overline{e_9}\overline{e_{10}}e_{11} +$
s_{12}	1	0	0	0	0	0	0	0	0	0	0	0	0	$e_0e_2e_4e_6\overline{e_8}e_9\overline{e_{10}}e_{11}\overline{e_{12}} +$
s_{13}	1	1	0	0	0	0	0	0	0	0	0	0	0	$e_0e_2e_4e_6\overline{e_8}e_9\overline{e_{10}}e_{11}e_{12} +$
s_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	$e_0e_2e_5e_7e_8e_9\overline{e_{10}}e_{11}\overline{e_{12}} +$
														$\overline{e_1}e_2e_3e_4e_5e_7e_8e_9\overline{e_{10}}e_{11}e_{12}$

Hình 4.18. ASA_{SPEC} thuật toán 4.4



Hình 4.19. ASA thuật toán 4.4

4.7.3. Kết quả tổng hợp RTL (Hình 4.19)

4.7.4. Tạo máy trạng thái RTL đại số

ASA được tạo từ kết quả tổng hợp RTL của thuật toán 4.4, như minh họa trong hình 4.19 và thể hiện thành máy trạng thái ASA_{RTL} ở Hình 4.20.

ASA_{RTL}	e_1	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	ASA_{RTL}	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}
s_0	0	0	0	0	0	0	0	0	0	0	0	0	0	s_8	1	1	1	1	1	1	1	0	0	0	0	0	0
s_1	1	0	0	0	0	0	0	0	0	0	0	0	0	s_9	1	1	1	1	1	1	1	1	0	0	0	0	0
s_2	1	0	1	0	0	0	0	0	0	0	0	0	0	s_{10}	1	1	1	1	1	1	1	1	1	0	0	0	0
s_3	1	1	0	0	0	0	0	0	0	0	0	0	0	s_{11}	1	1	1	1	1	1	1	1	1	1	0	0	0
s_4	1	1	1	0	0	0	0	0	0	0	0	0	0	s_{12}	1	1	1	1	1	1	1	1	1	1	1	0	0
s_5	1	1	1	0	1	0	0	0	0	0	0	0	0	s_{13}	1	1	1	1	1	1	1	1	1	1	1	1	0
s_6	1	1	1	1	1	0	0	0	0	0	0	0	0	s_{14}	1	1	1	1	1	1	1	1	1	1	1	1	1
s_7	1	1	1	1	1	1	0	0	0	0	0	0	0														

Hình 4.20. ASA_{RTL} của thuật toán 4.4

4.8. Kết quả so sánh

Việc kiểm chứng tính đúng đắn của kết quả RTL được thực hiện bằng

cách so sánh các tính chất giữa ASA_{SPEC} và ASA_{RTL} (thuật toán 4.2 hoặc 4.4).

Định nghĩa 4.3 (Tính đúng đắn của kết quả tổng hợp RTL): Một kết quả tổng hợp RTL dựa trên ngữ nghĩa đại số là chính xác nếu đáp ứng tất cả các yêu cầu của đặc tả đại số.

Định lý 4.1: Một kết quả tổng hợp RTL dựa trên ngữ nghĩa đại số là chính xác nếu và chỉ nếu $\mathcal{C}(\text{RTL}) = \mathcal{C}(\text{SPEC})$ (Định nghĩa 4.1).

Việc kiểm chứng tính đúng đắn giữa RTL và SPEC được thực hiện bằng cách so sánh ASA_{RTL} với ASA_{SPEC}. Hình 4.13 và 4.15 (Thuật toán 4.2) hoặc Hình 4.18 và 4.20 (Thuật toán 4.4) cho thấy $\mathcal{C}(\text{SPEC})$ tương đương $\mathcal{C}(\text{RTL})$.

KẾT LUẬN

- **Đóng góp khoa học**
  - **Xây dựng cơ sở đại số stream dùng cho stream dạng BDL trong IoMT [CT.1][CT.2][CT.3][CT.5][CT.7][CT.8]**
    - Phân tích lý thuyết stream (Đại số stream và Đồng đại số stream)
    - Xây dựng mười phép toán xử lý stream
    - Tổ hợp các phép toán stream thành biểu thức stream mới
    - Mở rộng đại số stream và đồng đại số stream
    - Cấu trúc đại số monoid và tính chất monoid
    - Xây dựng kịch bản để mô phỏng, kiểm chứng đánh giá các phép toán stream
  - **Xây dựng cơ chế lập lịch xử lý stream dạng BDL [CT.2][CT.9]**
    - Phân hoạch các phép toán xử lý stream
    - Lập lịch cho tính toán stream dạng BDL
    - Cấp phát và liên kết thanh ghi
    - Cấp phát và liên kết đơn vị chức năng
    - Tối ưu hóa lập lịch theo cơ chế đường ống nhằm tính toán stream BDL
  - **Xây dựng mô hình ngữ nghĩa đại số [CT.4]|[CT.6]**
    - Xây dựng mô hình ASM
    - Quan hệ giữa ASA_{RTL} và ASA_{SPEC}
- **Hướng phát triển**
    - Mở rộng các phép toán xử lý stream dạng BDL
    - Xây dựng các cấu trúc đại số mới dùng cho stream dạng BDL
    - Phát triển lý thuyết stream tích hợp với dữ liệu lớn
    - Nghiên cứu tính toán dữ liệu lớn trong hệ thống tính toán
    - Xây dựng mô hình tổng quát tối ưu hóa lập lịch theo cơ chế đường ống
    - Xây dựng một số khía cạnh ngữ nghĩa đại số đặc tả hoạt động của BDL

DANH MỤC CÁC BÀI BÁO ĐÃ XUẤT BẢN LIÊN QUAN ĐẾN LUẬN ÁN

[CT.1] **Dang Van Pham**, Vinh Cong Phan (2023), "Overview of The Stream Theory-Based Big Data in Livestream", Mobile Networks and Applications, ISSN: 1572-8153, Springer, 29(4), pp: 1239–1256, DOI: 10.1007/s11036-023-02180-0, **Scopus indexed(Q1), SCIE IF: 3.8(Q2)**.

[CT.2] **Dang Van Pham**, Vinh Cong Phan, Bao Khang Nguyen (2023), "Formally Specifying and Coinductive Approach to Verifying Synthesis of Stream Calculus-Based Computing Big Data in Livestream", Internet of Things, ISSN: 2542-6605, Elsevier, vol: 23, DOI: 10.1016/j.iot.2023.100878, **Scopus indexed(Q1), SCIE IF: 7.6(Q1)**.

[CT.3] **Dang Van Pham**, Vinh Cong Phan (2024), "Algebraic Aspects of Big Data in Livestream in Internet of Mobile Things", Mobile Networks and Applications, ISSN: 1572-8153, Springer, 29(1), pp: 243–261, DOI: 10.1007/s11036-024-02297-w, **Scopus indexed(Q1), SCIE IF: 3.8(Q2)**.

[CT.4] **Dang Van Pham**, Vinh Cong Phan, Bao Khang Nguyen (2024), "Algebraic Semantics of Register Transfer Level in Synthesis of Stream Calculus-Based Computing Big Data in Livestream", EAI ICTCC 2023 - 9th EAI International Conference on Nature of Computation and Communication. LNICST, ISSN: 1867-8211, Springer, Cham, vol: 586, pp: 19-35, ISBN: 978-3-031-59461-8, DOI: 10.1007/978-3-031-59462-5_2 **(Scopus, Q4)**.

[CT.5] **Dang Van Pham**, Vinh Cong Phan (2024), "Monoids Structure used for Computing Big Data in Livestream in Computing Systems", Hội thảo quốc gia lần thứ XXVII: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – VNICT 2024, Nha Trang, Việt Nam, 11-12/10/2024, trang: 363-369, ISBN: 978-604-67-3029-3.

[CT.6] **Dang Van Pham**, Vinh Cong Phan, Trong Toan Tran (2025), "Relationship between RTL Automata (ASA_{RTL}) and Algorithm Specification (ASA_{SPEC}) Based on Algebraic Semantics in Synthesis of Stream Calculus-Based Computing Big Data in Livestream", EAI ICCASA 2024 - 13th EAI International Conference on Context-Aware Systems and Applications. LNICST, ISSN: 1867-8211, Springer, Cham, vol: 664, ISBN: 978-3-032-12941-3 **(Scopus, Q4)**.

[CT.7] **Dang Van Pham**, Vinh Cong Phan, Toan Trong Tran, Bao Khang Nguyen (2024), "An Algebraic Structure for Computing Big Data in Livestream," 2024 13th International Conference on Control, Automation and Information Sciences (ICCAIS), ISSN: 2475-7896, ISBN: 979-8-3315-4204-7, IEEE, DOI: 10.1109/ICCAIS63750.2024.10814295 **(Scopus)**.

[CT.8] **Dang Van Pham**, Vinh Cong Phan, Trong Toan Tran (2025), "Analyzing and Comparing the Stream Theory: A Review", EAI ICTCC 2024 - 10th EAI International Conference on Nature of Computation and Communication. LNICST, ISSN: 1867-8211, Springer, Cham, vol: 668, ISBN: 978-3-032-12846-1, DOI: 10.1007/978-3-032-12846-1_6 **(Scopus, Q4)**.

[CT.9] **Dang Van Pham**, Vinh Cong Phan (2025), "Developing Linear Programming-Based Pipeline Scheduling Optimization Model Used to Big Data in Livestream", Proceeding of the 18th National Conference on Fundamental and Applied IT Research – FAIR'18, Ha Noi, 21-22/08/2025, ISBN: 978-604-357-455-5, DOI: 10.15625/vap.2025.0298, pp: 275-281 **(HĐCDGSNN)**.