Simpler is Faster: Multi-Dimensional Lock-Free Arrays for Multithreaded Mode-Directed Tabling in Prolog

Miguel Areias and Ricardo Rocha
CRACS & INESC-TEC LA
Faculty of Sciences, University of Porto, Portugal
miguel-areias@dcc.fc.up.pt *ricroc@dcc.fc.up.pt*

Tabling in Prolog Systems

- **Tabling** is an **implementation technique** that is used in some **Prolog** systems to improve their performance in **dynamic programming problems**:
  - Tabled subgoal calls are evaluated by storing their answers in an appropriate data space, called the **table space**.
  - Repeated calls to tabled subgoals are resolved by **consuming** the answers already stored in the table instead of **being re-evaluated** against the program clauses.

Tabling in Prolog Systems

- **Tabling** is an **implementation technique** that is used in some **Prolog** systems to improve their performance in **dynamic programming problems**:
  - Tabled subgoal calls are evaluated by storing their answers in an appropriate data space, called the **table space**.
  - Repeated calls to tabled subgoals are resolved by **consuming** the answers already stored in the table instead of **being re-evaluated** against the program clauses.

- Implementations of **Tabling** are currently available in systems like:
  - XSB Prolog, **Yap Prolog**, B-Prolog, ALS-Prolog, Mercury, Ciao Prolog and more recently Picat.
- **Multithreading** combined with **Tabling**:
  - XSB Prolog
  - **YapTab-Mt**.

YapTab-Mt - Advantages

- An **abstraction layer** with **high-level constructors** that provide access to the **dynamic programming (tabling)** support:
  - Instruction: **:- table predicate/arity.**

YapTab-Mt - Advantages

- An **abstraction layer** with **high-level constructors** that provide access to the **dynamic programming (tabling)** support:
  - Instruction: **:- table predicate/arity.**

- **Thread API** is **POSIX Threads compliant**:
  - **Management** - creating, joining , yielding, etc.
  - **Monitoring** - statistics, properties, etc.
  - **Synchronization** - mutex creation, statistics, etc.

YapTab-Mt - Advantages

- An **abstraction layer** with **high-level constructors** that provide access to the **dynamic programming (tabling)** support:
  - Instruction: **:- table predicate/arity.**
- **Thread API** is **POSIX Threads compliant**:
  - **Management** - creating, joining , yielding, etc.
  - **Monitoring** - statistics, properties, etc.
  - **Synchronization** - mutex creation, statistics, etc.
- Write complex **dynamic programming** applications using the **Prolog** programming language.
  - **Procedures** in **Prolog** can be written as **logical specifications**, which are closer to a **mathematical notation**.

Table Space - Internal Architecture

- **Table Entry**: stores generic about the predicates.
  - **:-table ks(index, index, max)**.

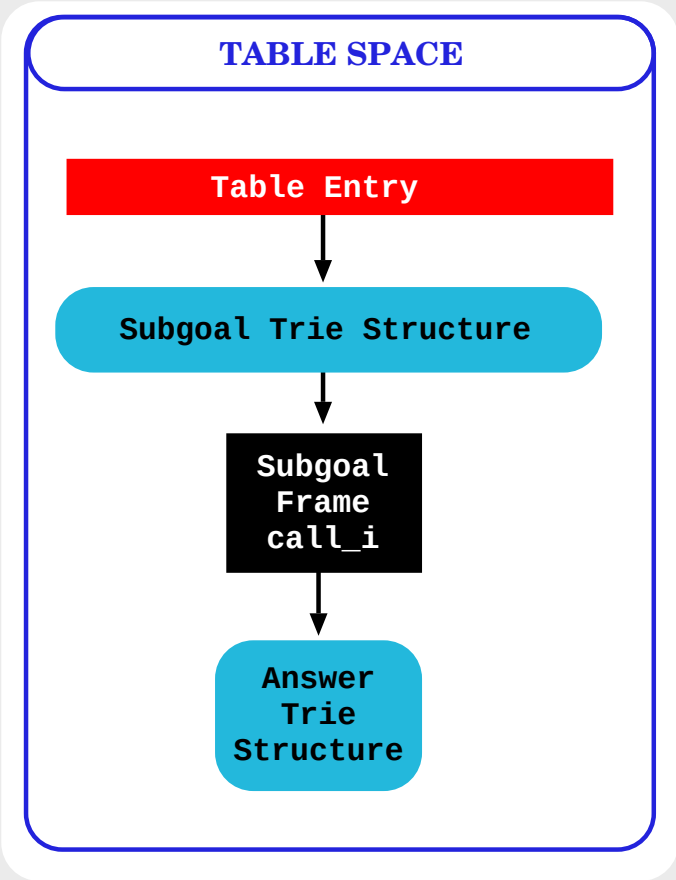


Table Space - Internal Architecture

- **Table Entry**: stores generic about the predicates.
  - **:-table ks(index, index, max)**.
- **Subgoal Trie Structure**: stores the **identifier** of the computations.
  - **ks**(**item**, **capacity**, profit).

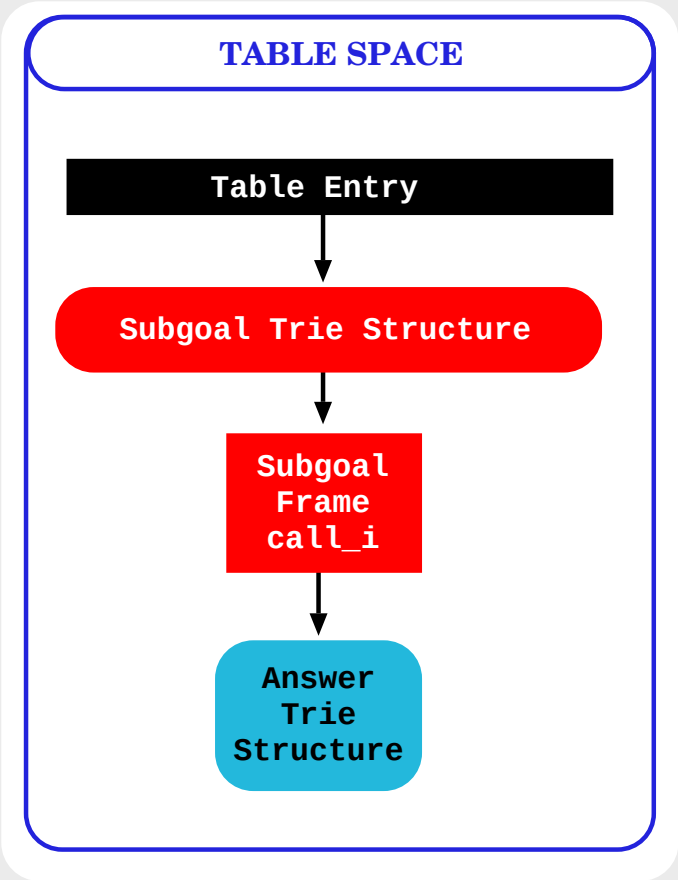
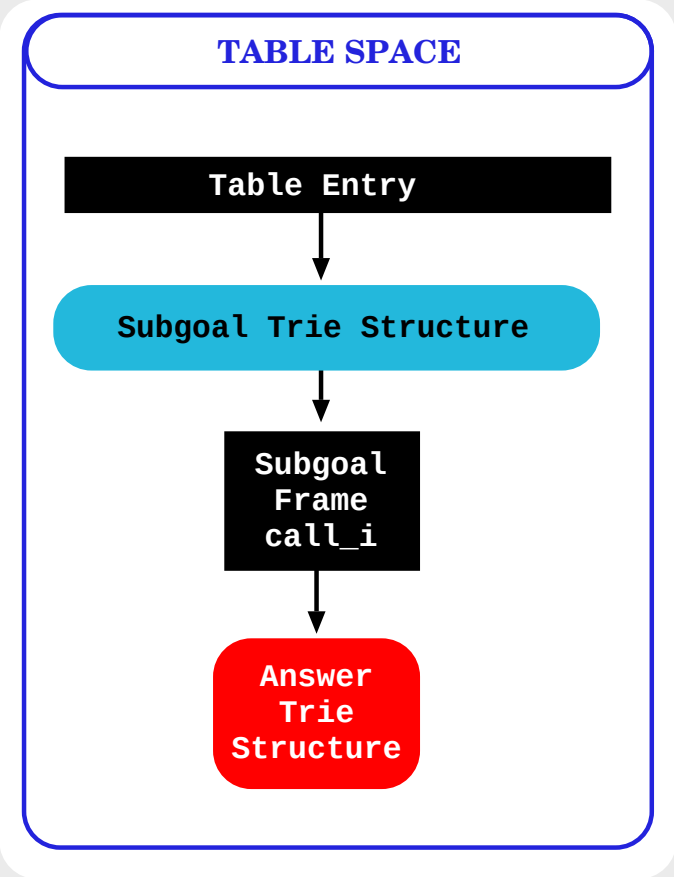
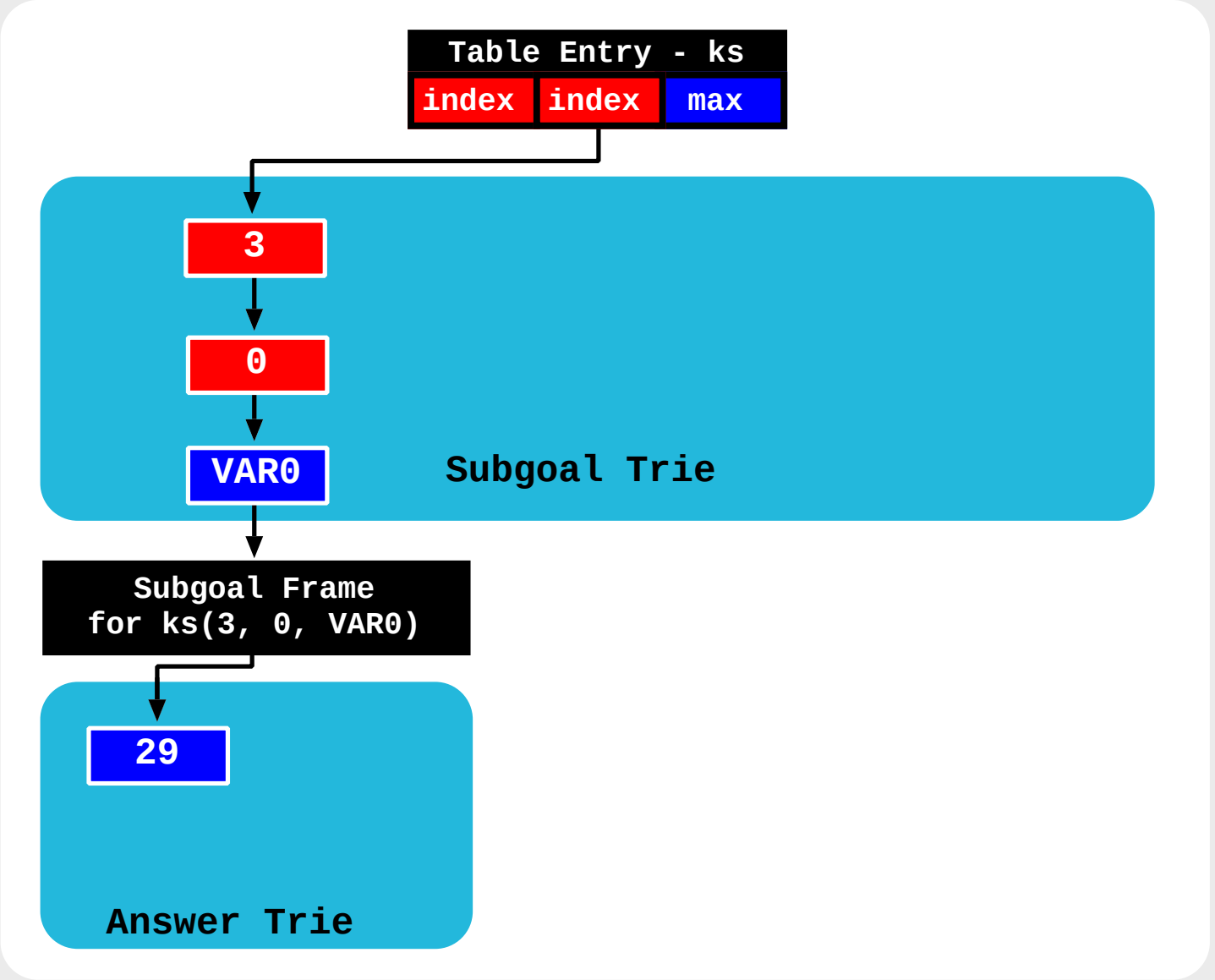


Table Space - Internal Architecture

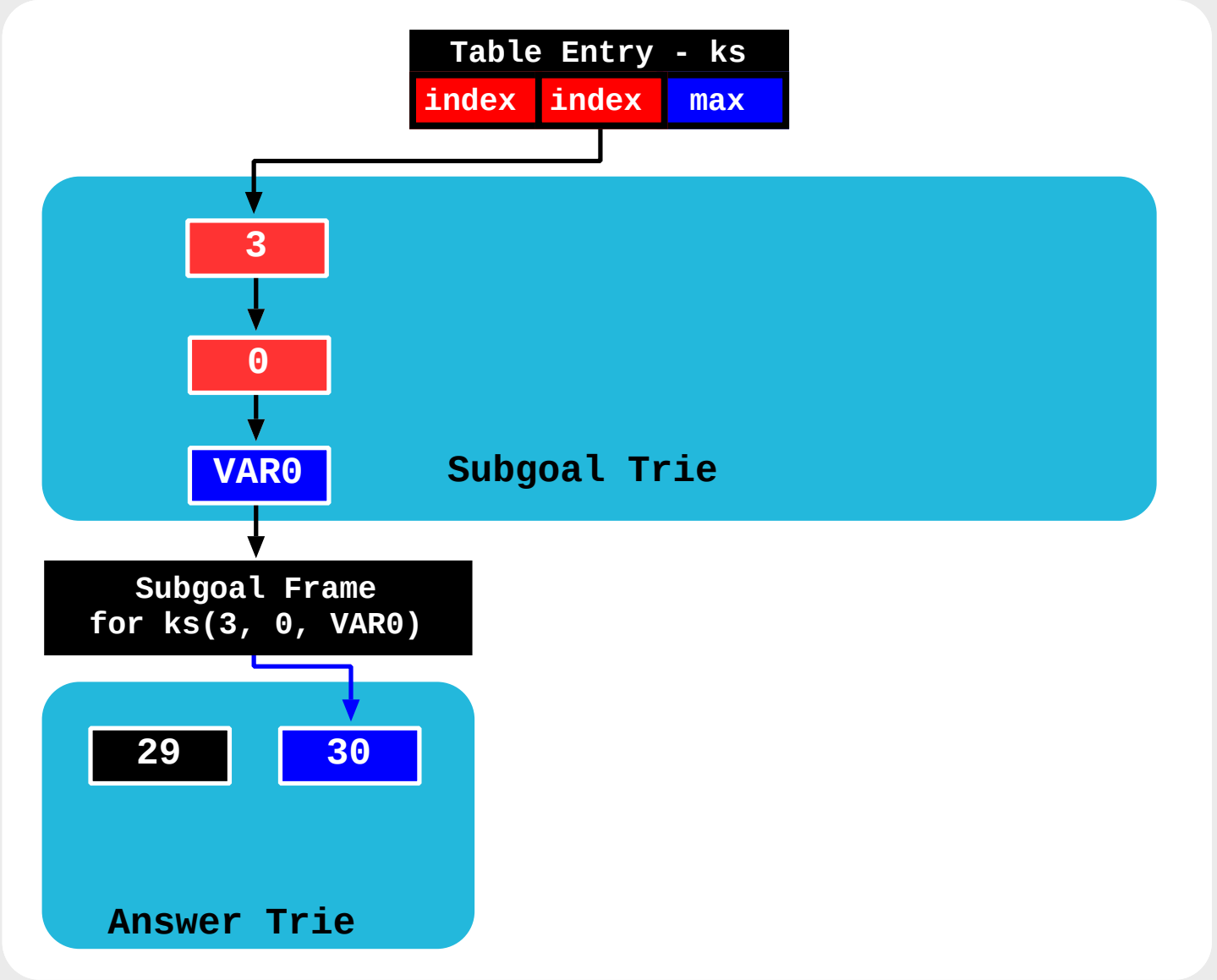
- **Table Entry**: stores generic about the predicates.
  - **:-table ks(index, index, max)**.
- **Subgoal Trie Structure**: stores the **identifier** of the computations.
  - **ks**(**item**, **capacity**, profit).
- **Answer Trie Structure**: stores the **answers** of the computations.
  - **ks**(item, capacity, **profit**).



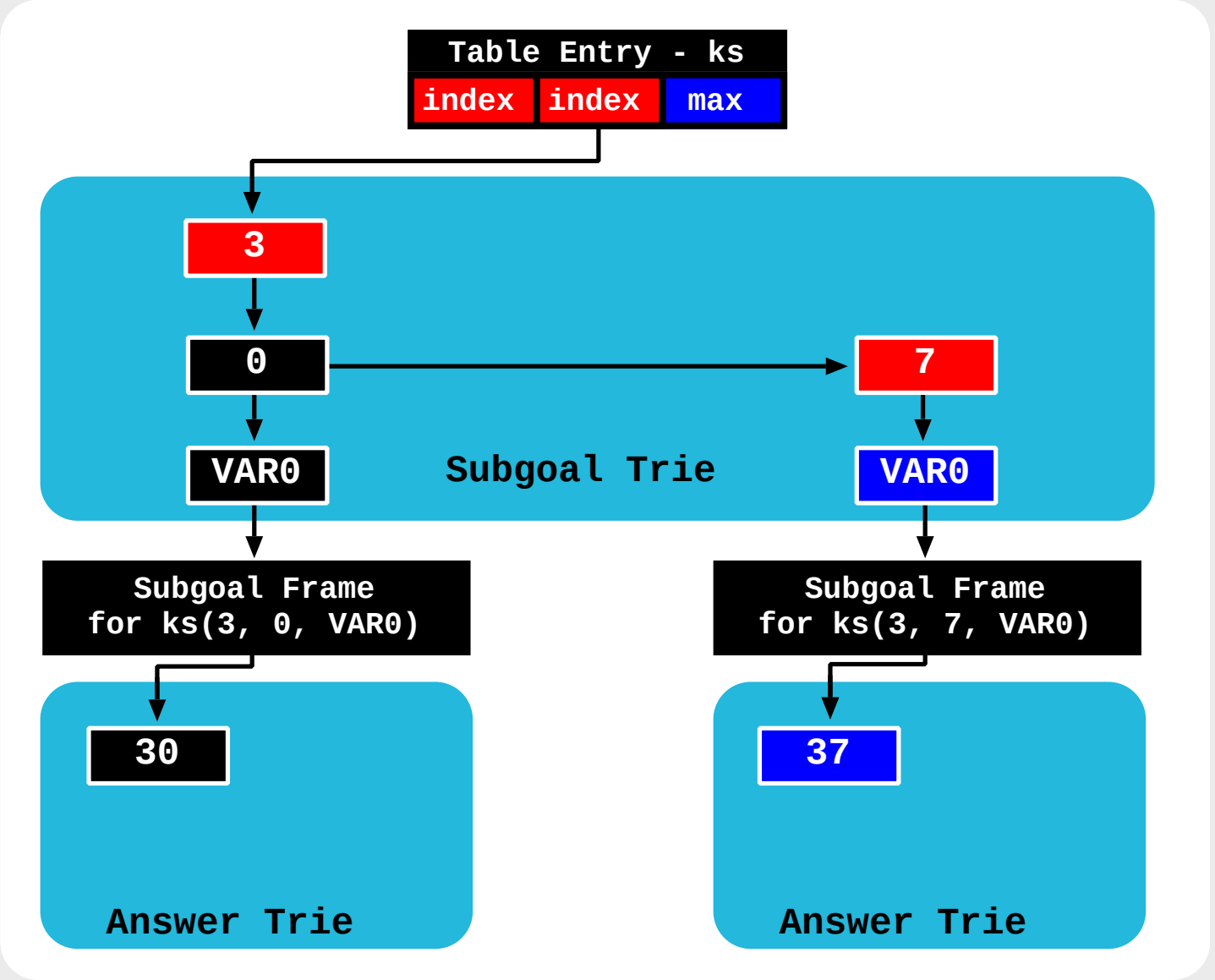
Original Design (Sequential)



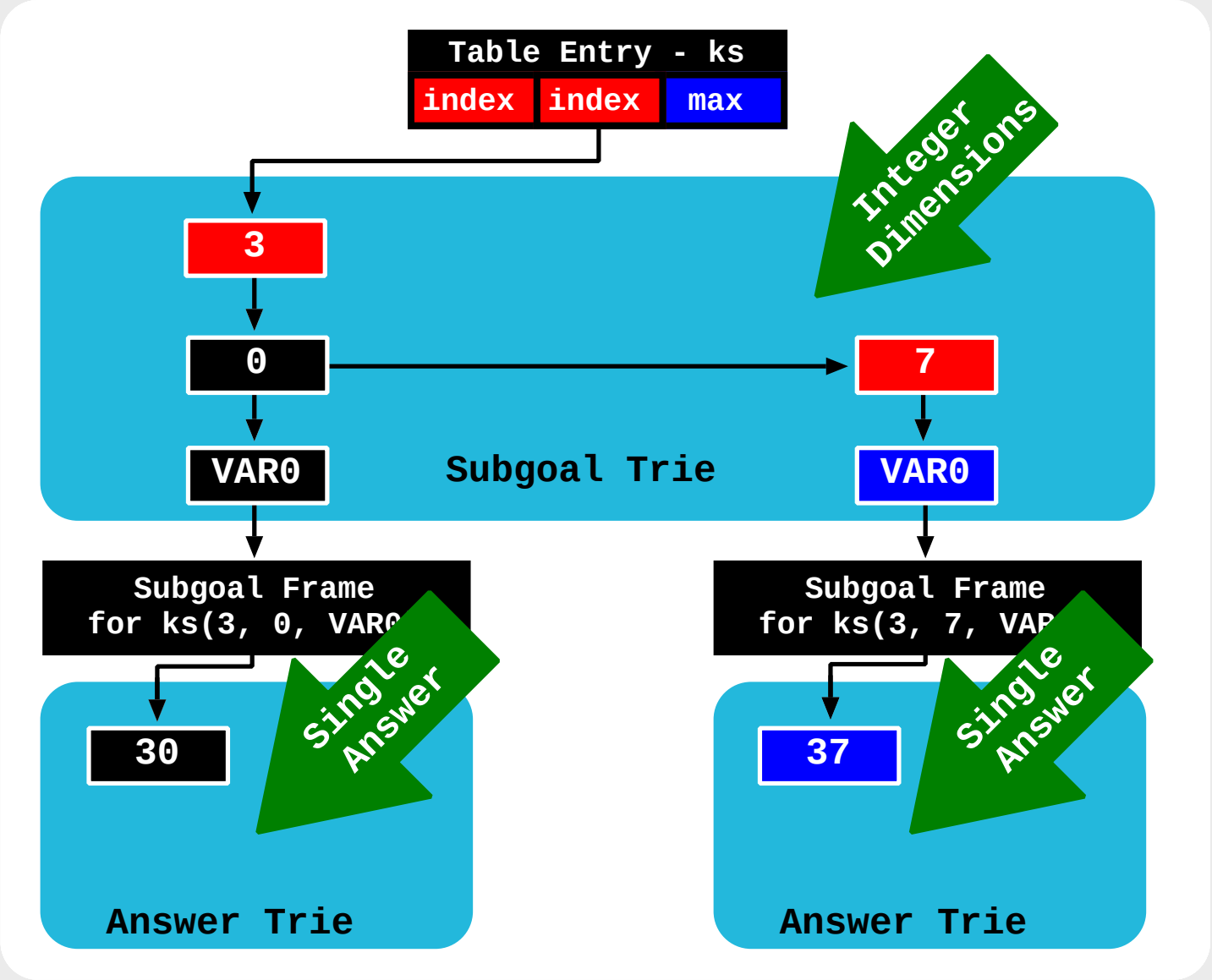
Original Design (Sequential)



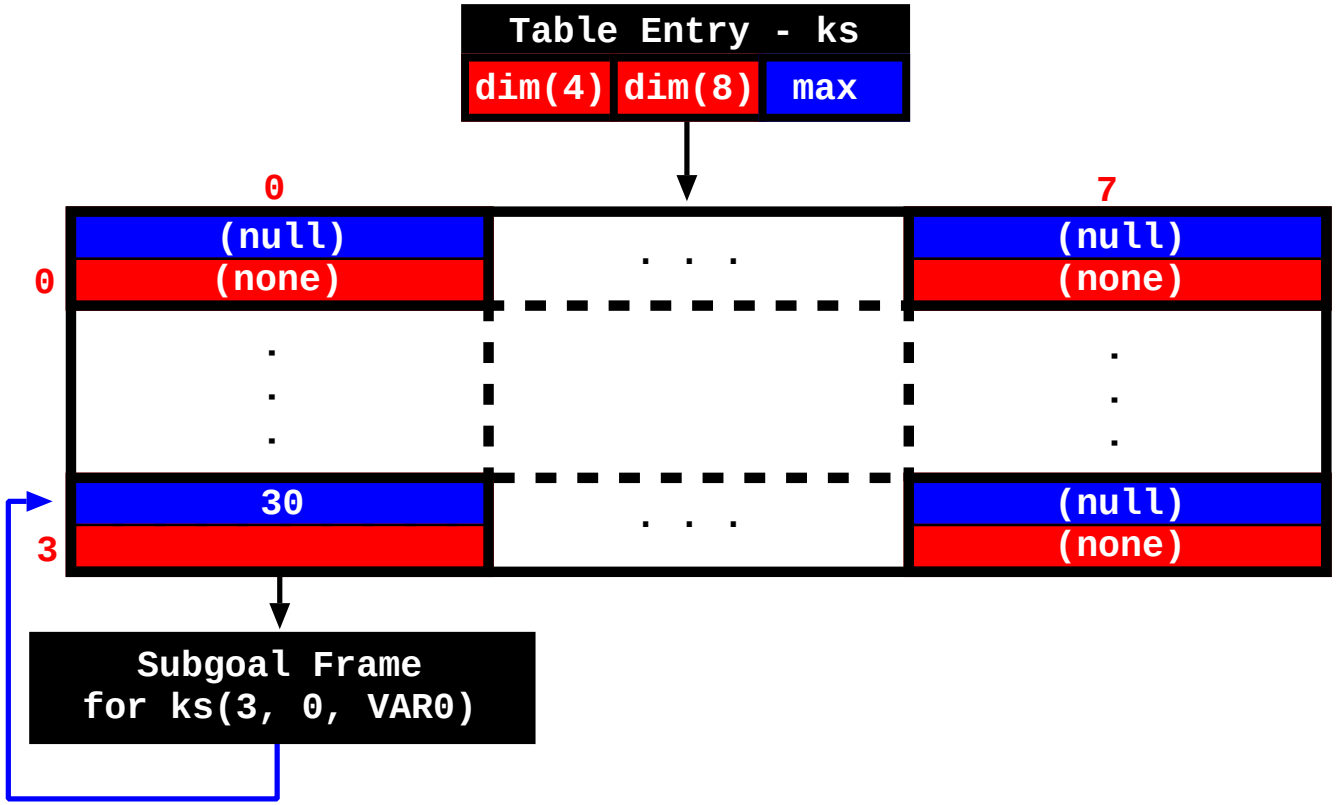
Original Design (Sequential)



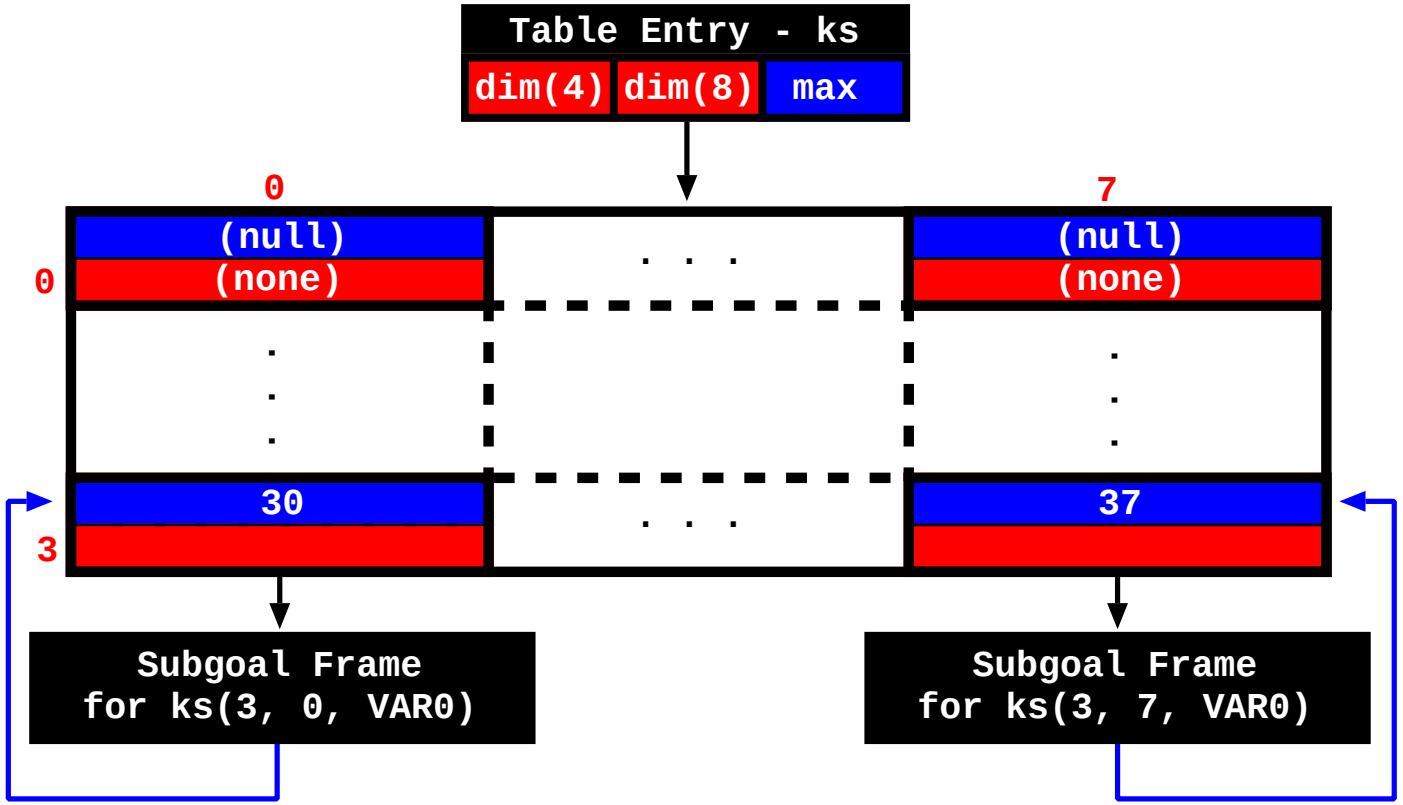
Original Design (Sequential)



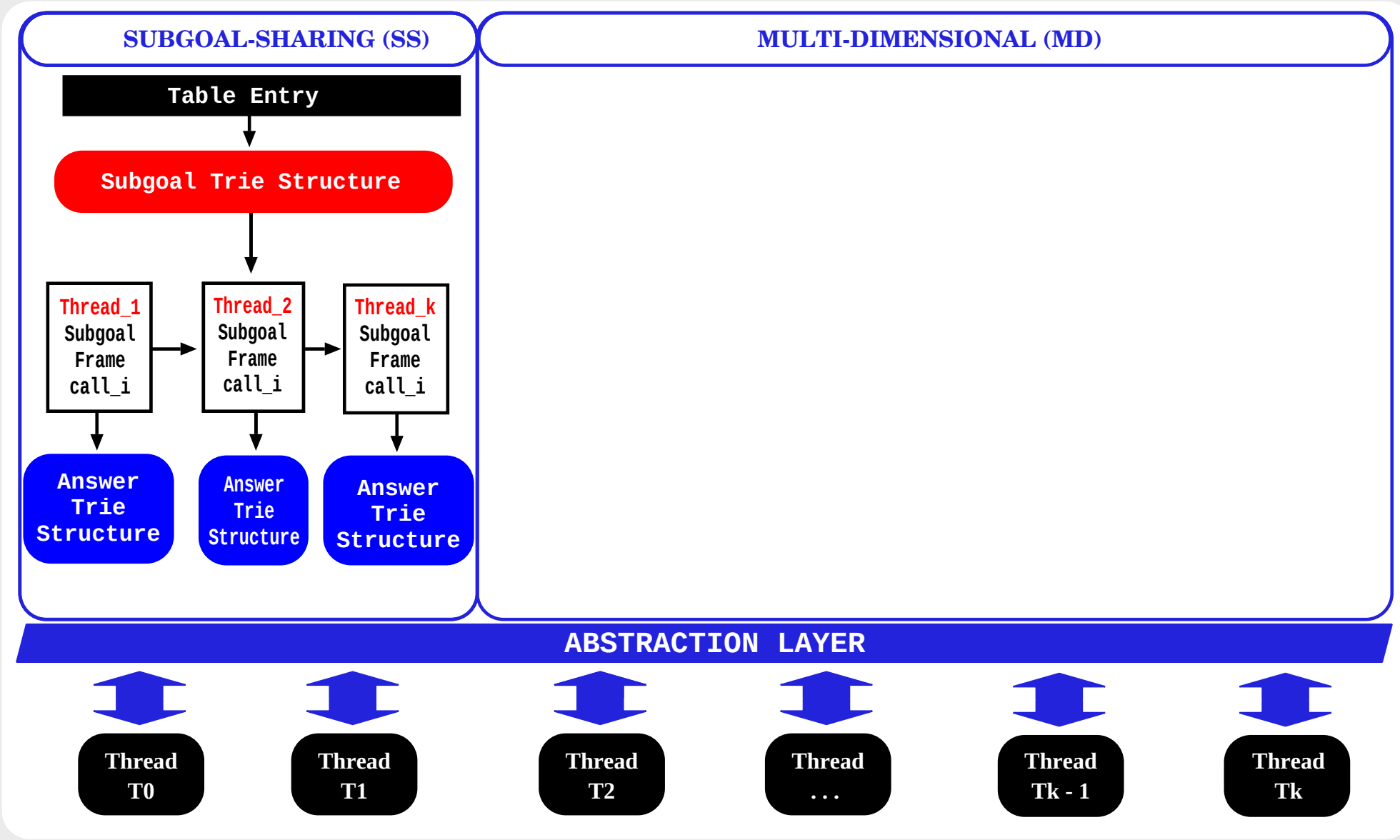
Multi-Dimensional Design (Sequential)



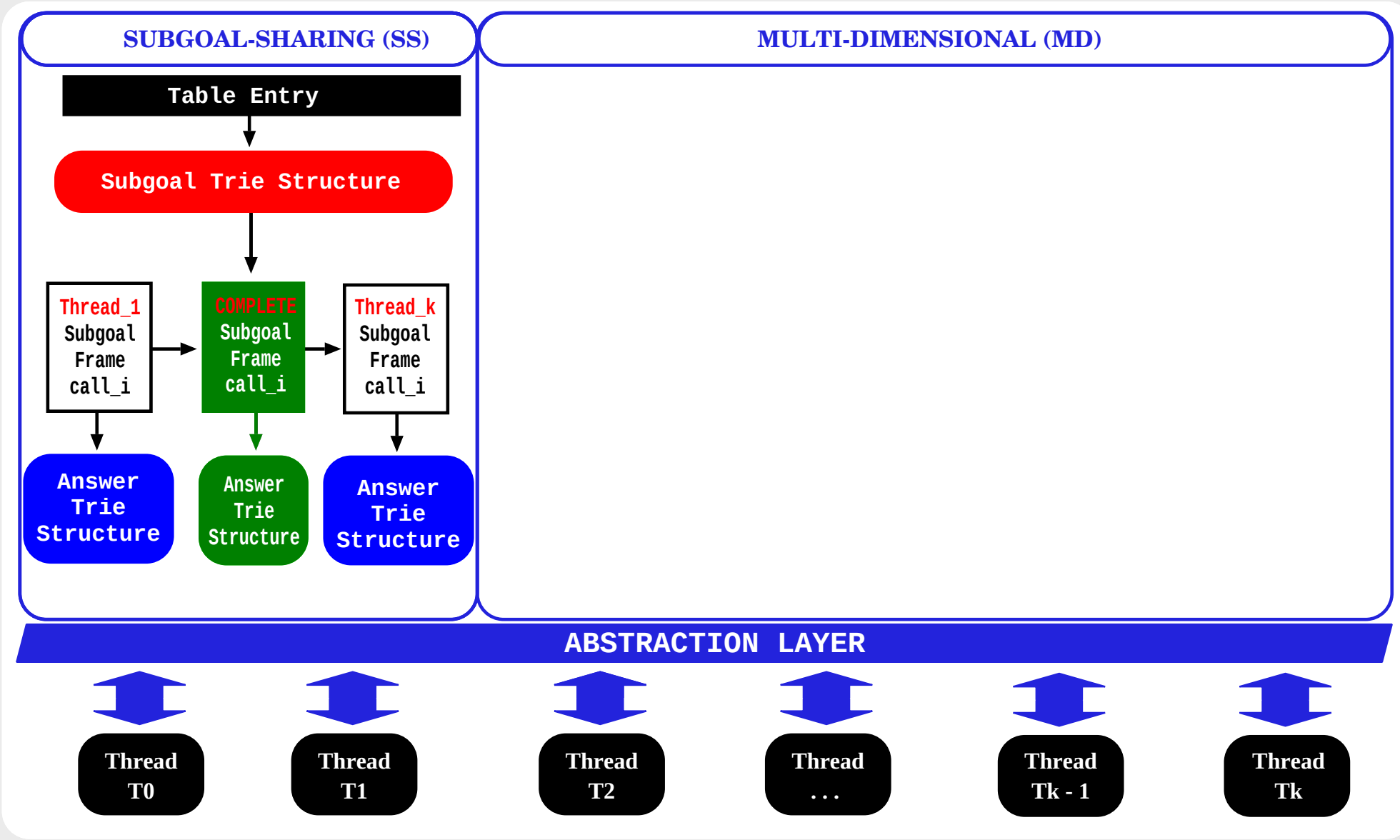
Multi-Dimensional Design (Sequential)



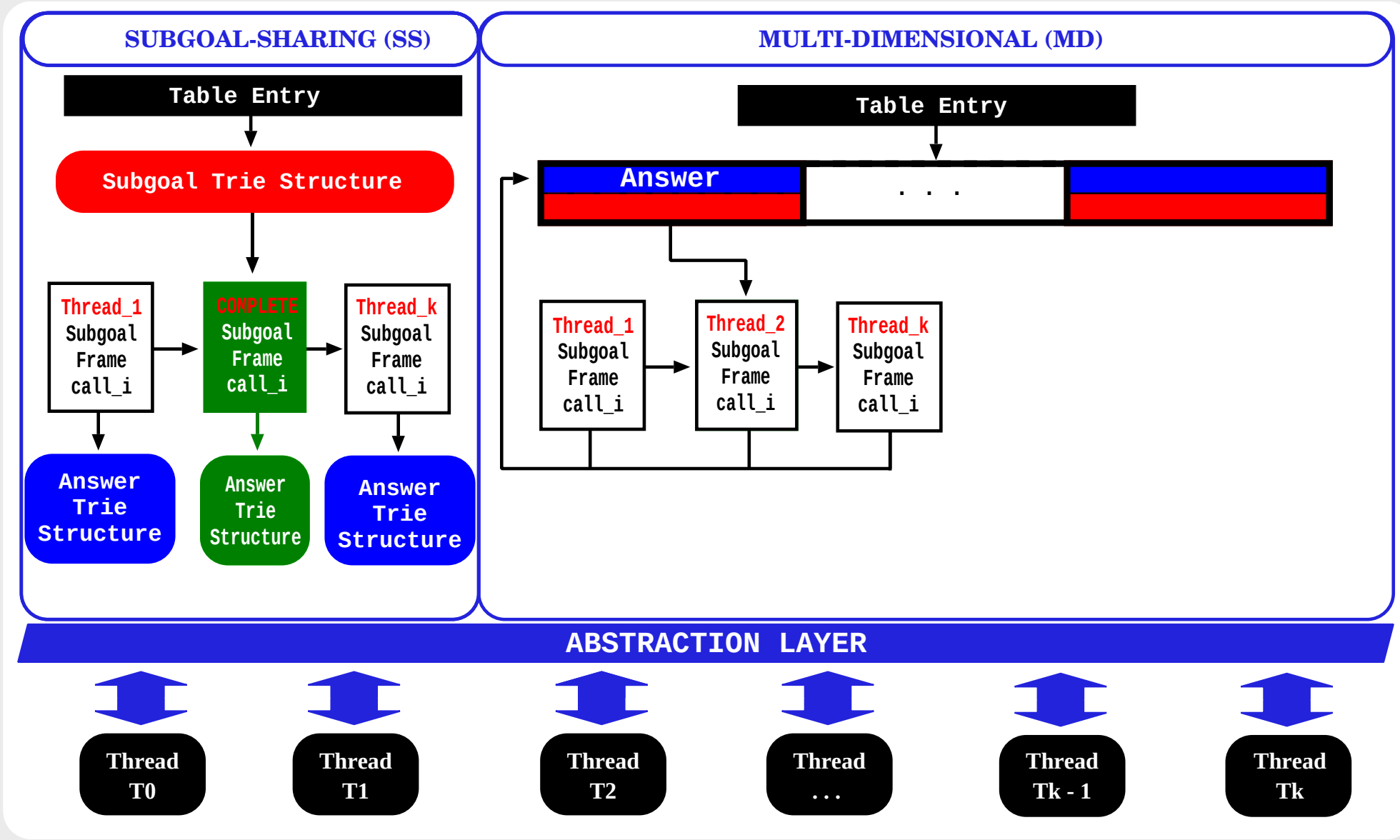
Multi-Dimensional Design (Threaded)



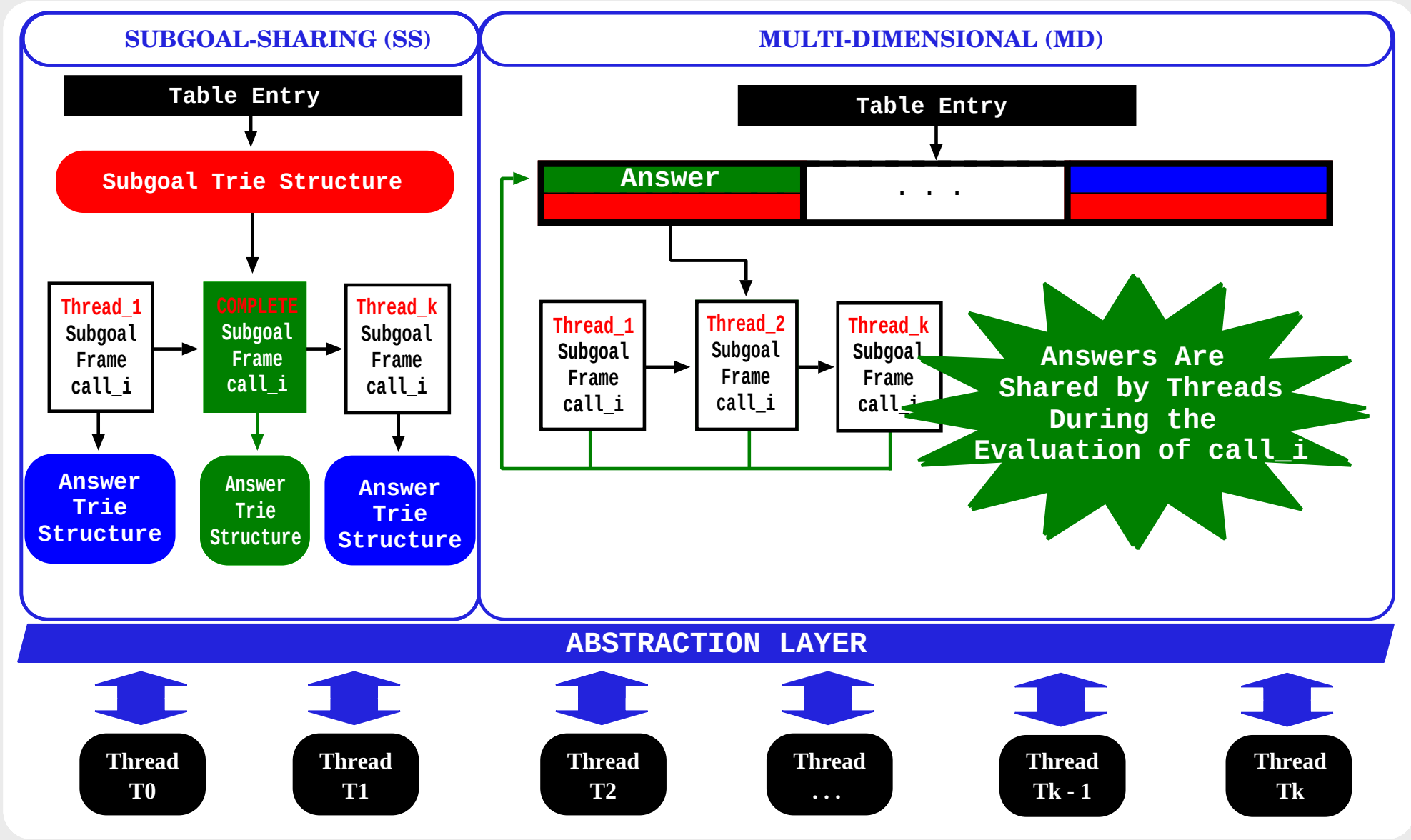
Multi-Dimensional Design (Threaded)



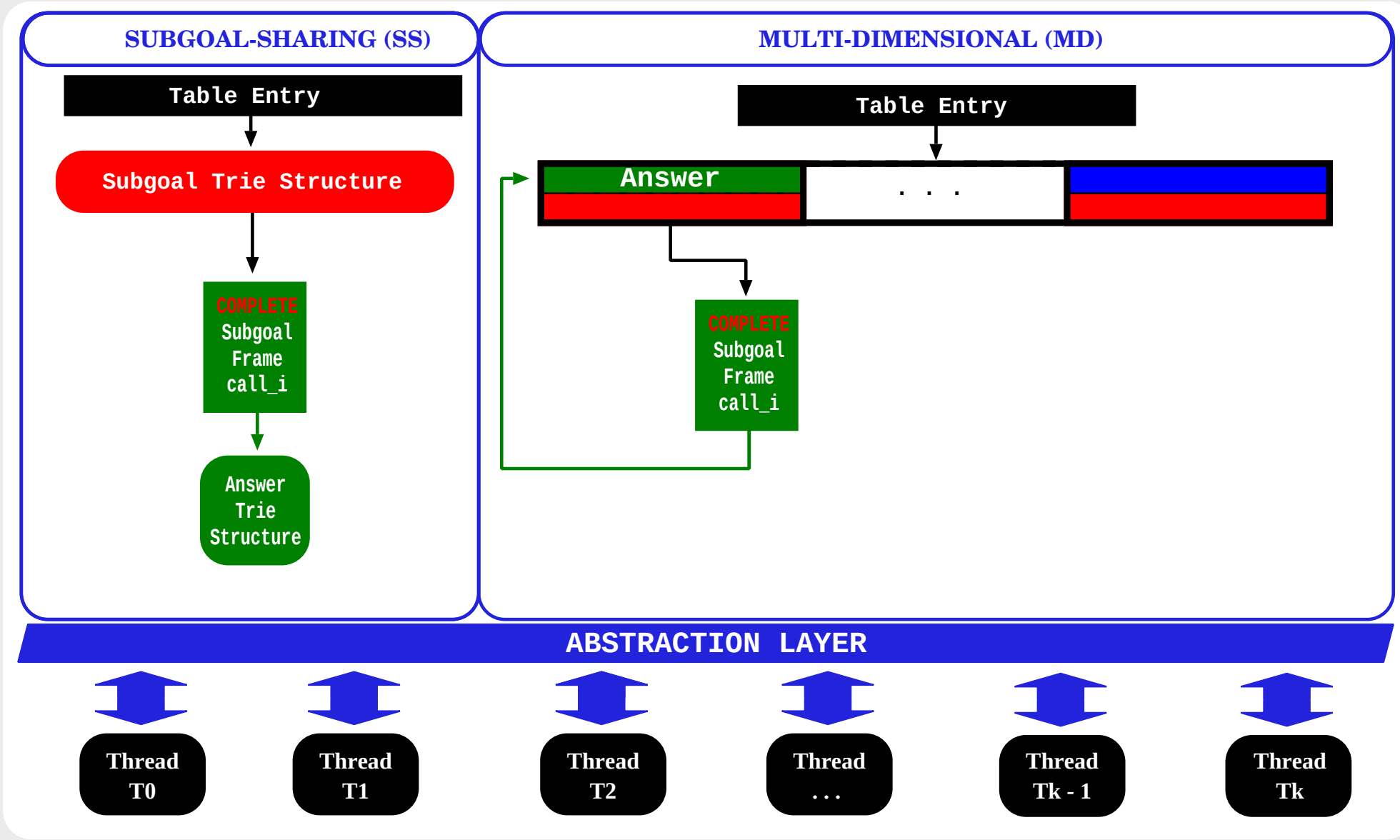
Multi-Dimensional Design (Threaded)



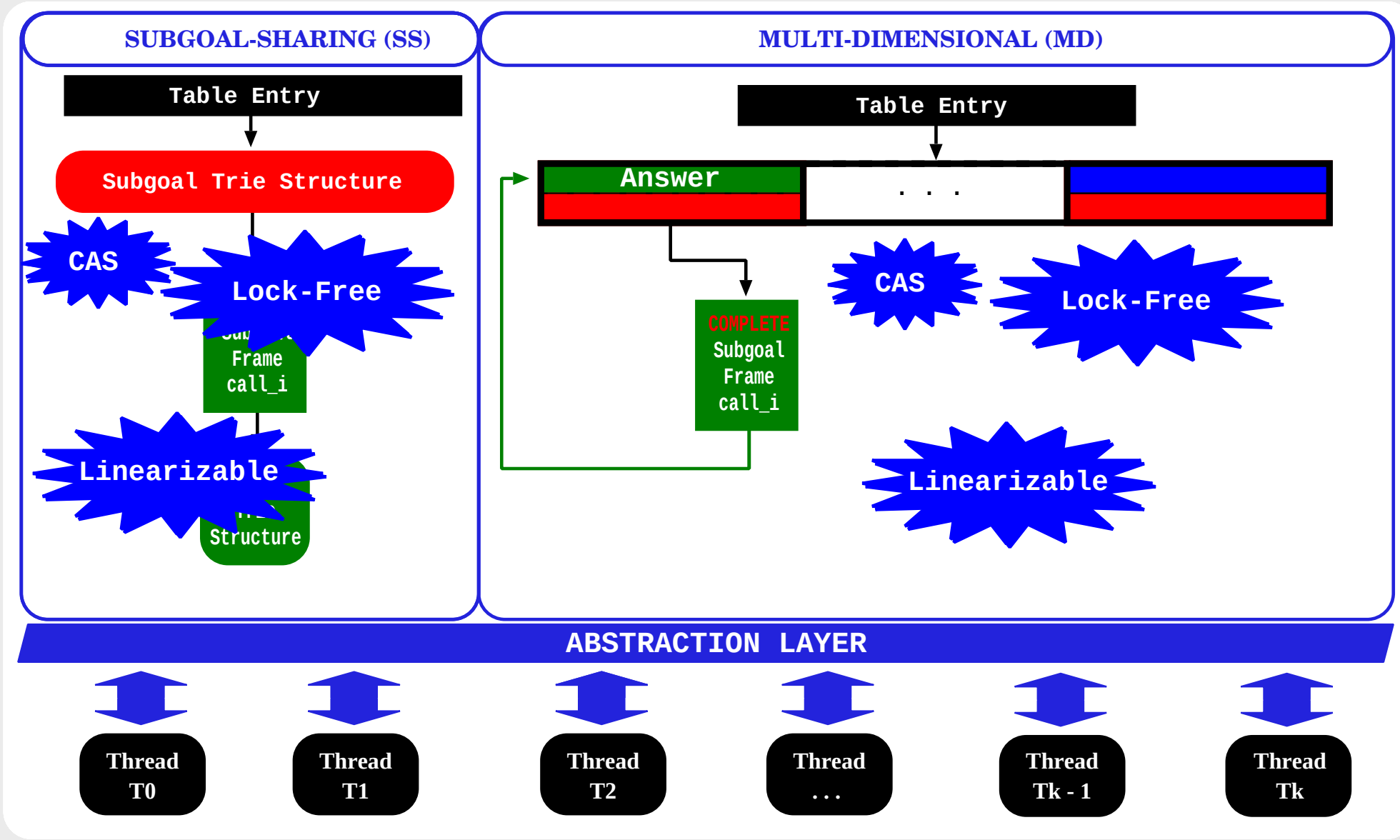
Multi-Dimensional Design (Threaded)



Multi-Dimensional Design (Threaded)



Multi-Dimensional Design (Threaded)



Performance - Sequential vs One-Thread

Approach & Dataset		Subgoal Sharing (SS) Time T_{seq}	Subgoal Sharing (SS) Time T_1	Subgoal Sharing (SS) Ratio T_1/T_{seq}	Multi-Dimensional (MD) Time T_{seq}	Multi-Dimensional (MD) Time T_1	Multi-Dimensional (MD) Ratio T_1/T_{seq}	SS vs MD Ratio SS_{T_1}/MD_{T_1}
Knapsack Problem								
TD_{no}	D_{10}	9,508	12,415	1.31	4,275	5,241	1.23	2.37
	D_{30}	9,246	12,177	1.32	4,196	5,336	1.27	2.28
	D_{50}	9,480	12,589	1.33	4,275	5,457	1.28	2.31
TD_{rnd}	D_{10}	19,667	24,444	1.24	10,462	11,740	1.12	2.08
	D_{30}	19,847	25,609	1.29	10,508	11,959	1.14	2.14
	D_{50}	19,985	25,429	1.27	10,805	11,982	1.11	2.12
BU	D_{10}	12,614	17,940	1.42	7,001	7,668	1.10	2.34
	D_{30}	12,364	17,856	1.44	7,005	7,786	1.11	2.29
	D_{50}	12,653	17,499	1.38	6,775	7,637	1.13	2.29
LCS Problem (Longest Common Subsequence)								
TD_{no}	D_{10}	21,191	26,225	1.24	16,202	18,046	1.11	1.45
	D_{30}	20,809	26,146	1.26	16,006	18,067	1.13	1.45
	D_{50}	20,775	26,028	1.25	16,259	18,195	1.12	1.43
TD_{rnd}	D_{10}	34,565	44,371	1.28	21,525	23,635	1.10	1.88
	D_{30}	34,284	44,191	1.29	21,512	24,055	1.12	1.84
	D_{50}	33,989	44,158	1.30	21,477	23,736	1.11	1.86
BU	D_{10}	20,799	28,909	1.39	11,453	14,017	1.22	2.06
	D_{30}	21,174	28,904	1.37	11,218	14,189	1.26	2.04
	D_{50}	21,166	28,857	1.36	11,139	13,982	1.26	2.06

Performance - Sequential vs One-Thread

Approach & Dataset		Subgoal Sharing (SS) Time T_{seq}	Subgoal Sharing (SS) Time T_1	Subgoal Sharing (SS) Ratio T_1/T_{seq}	Multi-Dimensional (MD) Time T_{seq}	Multi-Dimensional (MD) Time T_1	Multi-Dimensional (MD) Ratio T_1/T_{seq}	SS vs MD Ratio SS_{T_1}/MD_{T_1}
Knapsack Problem								
TD_{no}	D_{10}	9,508	12,415	1.31	4,275	5,241	1.23	2.37
	D_{30}	9,246	12,177	1.32	4,196	5,336	1.27	2.28
	D_{50}	9,480	12,589	1.33	4,275	5,457	1.28	2.31
TD_{rnd}	D_{10}	19,667	24,444	1.24	10,462	11,740	1.12	2.08
	D_{30}	19,847	25,609	1.29	10,508	11,959	1.14	2.14
	D_{50}	19,985	25,429	1.27	10,805	11,982	1.11	2.12
BU	D_{10}	12,614	17,940	1.42	7,001	7,668	1.10	2.34
	D_{30}	12,364	17,856	1.44	7,005	7,786	1.11	2.29
	D_{50}	12,653	17,499	1.38	6,775	7,637	1.13	2.29
LCS Problem								
TD_{no}	D_{10}	21,191	26,225	1.24	16,202	18,046	1.11	1.45
	D_{30}	20,809	26,146	1.26	16,006	18,067	1.13	1.45
	D_{50}	20,775	26,028	1.25	16,259	18,195	1.12	1.43
TD_{rnd}	D_{10}	34,565	44,371	1.28	21,525	23,635	1.10	1.88
	D_{30}	34,284	44,191	1.29	21,512	24,055	1.12	1.84
	D_{50}	33,989	44,158	1.30	21,477	23,736	1.11	1.86
BU	D_{10}	20,799	28,909	1.39	11,453	14,017	1.22	2.06
	D_{30}	21,174	28,904	1.37	11,218	14,189	1.26	2.04
	D_{50}	21,166	28,857	1.36	11,139	13,982	1.26	2.06

Performance - Sequential vs One-Thread

Approach & Dataset		Subgoal Sharing (SS) Time T_{seq}	Subgoal Sharing (SS) Time T_1	Subgoal Sharing (SS) Ratio T_1/T_{seq}	Multi-Dimensional (MD) Time T_{seq}	Multi-Dimensional (MD) Time T_1	Multi-Dimensional (MD) Ratio T_1/T_{seq}	SS vs MD Ratio SS_{T_1}/MD_{T_1}
Knapsack Problem								
TD_{no}	D_{10}	9,508	12,415	1.31	4,275	5,241	1.23	2.37
	D_{30}	9,246	12,177	1.32	4,196	5,336	1.27	2.28
	D_{50}	9,480	12,589	1.33	4,275	5,457	1.28	2.31
TD_{rnd}	D_{10}	19,667	24,444	1.24	10,462	11,740	1.12	2.08
	D_{30}	19,847	25,609	1.29	10,508	11,959	1.14	2.14
	D_{50}	19,985	25,429	1.27	10,805	11,982	1.11	2.12
BU	D_{10}	12,614	17,940	1.42	7,001	7,668	1.10	2.34
	D_{30}	12,364	17,856	1.44	7,005	7,786	1.11	2.29
	D_{50}	12,653	17,499	1.38	6,775	7,637	1.13	2.29
LCS Problem								
TD_{no}	D_{10}	21,191	26,225	1.24	16,202	18,046	1.11	1.45
	D_{30}	20,809	26,146	1.26	16,006	18,067	1.13	1.45
	D_{50}	20,775	26,028	1.25	16,259	18,195	1.12	1.43
TD_{rnd}	D_{10}	34,565	44,371	1.28	21,525	23,635	1.10	1.88
	D_{30}	34,284	44,191	1.29	21,512	24,055	1.12	1.84
	D_{50}	33,989	44,158	1.30	21,477	23,736	1.11	1.86
BU	D_{10}	20,799	28,909	1.39	11,453	14,017	1.22	2.06
	D_{30}	21,174	28,904	1.37	11,218	14,189	1.26	2.04
	D_{50}	21,166	28,857	1.36	11,139	13,982	1.26	2.06

Performance - Sequential vs One-Thread

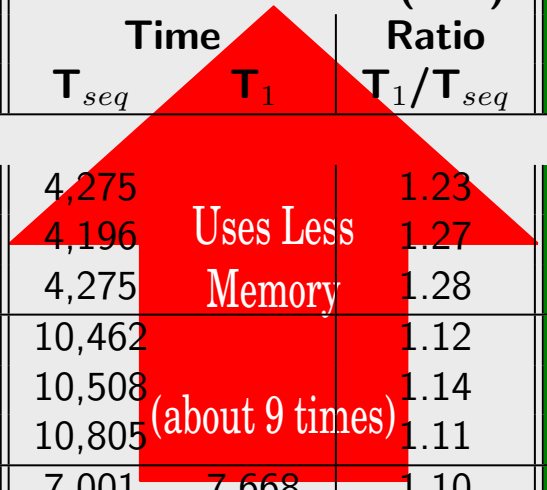
Approach & Dataset		Subgoal Sharing (SS) Time T_{seq}	Subgoal Sharing (SS) Time T_1	Subgoal Sharing (SS) Ratio T_1/T_{seq}	Multi-Dimensional (MD) Time T_{seq}	Multi-Dimensional (MD) Time T_1	Multi-Dimensional (MD) Ratio T_1/T_{seq}	SS vs MD Ratio SS_{T_1}/MD_{T_1}
Knapsack Problem								
TD_{no}	D_{10}	9,508	12,415	1.31	4,275	5,241	1.23	2.37
	D_{30}	9,246	12,177	1.32	4,196	5,336	1.27	2.28
	D_{50}	9,480	12,589	1.33	4,275	5,457	1.28	2.31
TD_{rnd}	D_{10}	19,667	24,444	1.24	10,462	11,740	1.12	2.08
	D_{30}	19,847	25,609	1.29	10,508	11,959	1.14	2.14
	D_{50}	19,985	25,429	1.27	10,805	11,982	1.11	2.12
BU	D_{10}	12,614	17,940	1.42	7,001	7,668	1.10	2.34
	D_{30}	12,364	17,856	1.44	7,005	7,786	1.11	2.29
	D_{50}	12,653	17,499	1.38	6,775	7,637	1.13	2.29
LCS Problem								
TD_{no}	D_{10}	21,191	26,225	1.24	16,202	18,046	1.11	1.45
	D_{30}	20,809	26,146	1.26	16,006	18,067	1.13	1.45
	D_{50}	20,775	26,028	1.25	16,259	18,195	1.12	1.43
TD_{rnd}	D_{10}	34,565	44,371	1.28	21,525	23,635	1.10	1.88
	D_{30}	34,284	44,191	1.29	21,512	24,055	1.12	1.84
	D_{50}	33,989	44,158	1.30	21,477	23,736	1.11	1.86
BU	D_{10}	20,799	28,909	1.39	11,453	14,017	1.22	2.06
	D_{30}	21,174	28,904	1.37	11,218	14,189	1.26	2.04
	D_{50}	21,166	28,857	1.36	11,139	13,982	1.26	2.06

Performance - Sequential vs One-Thread

Approach & Dataset		Subgoal Sharing (SS) Time T_{seq}	Subgoal Sharing (SS) Time T_1	Subgoal Sharing (SS) Ratio T_1/T_{seq}	Multi-Dimensional (MD) Time T_{seq}	Multi-Dimensional (MD) Time T_1	Multi-Dimensional (MD) Ratio T_1/T_{seq}	SS vs MD Ratio SS_{T_1}/MD_{T_1}
Knapsack Problem								
TD_{no}	D_{10}	9,508	12,415	1.31	4,275	5,241	1.23	2.37
	D_{30}	9,246	12,177	1.32	4,196	5,336	1.27	2.28
	D_{50}	9,480	12,589	1.33	4,275	5,457	1.28	2.31
TD_{rnd}	D_{10}	19,667	24,444	1.24	10,462	11,740	1.12	2.08
	D_{30}	19,847	25,609	1.29	10,508	11,959	1.14	2.14
	D_{50}	19,985	25,429	1.27	10,805	11,982	1.11	2.12
BU	D_{10}	12,614	17,940	1.42	7,001	7,668	1.10	2.34
	D_{30}	12,364	17,856	1.44	7,005	7,786	1.11	2.29
	D_{50}	12,653	17,499	1.38	6,775	7,637	1.13	2.29
LCS Problem								
TD_{no}	D_{10}	21,191	26,225	1.24	16,202	18,046	1.11	1.45
	D_{30}	20,809	26,146	1.26	16,006	18,067	1.13	1.45
	D_{50}	20,775	26,028	1.25	16,259	18,195	1.12	1.43
TD_{rnd}	D_{10}	34,565	44,371	1.28	21,525	23,635	1.10	1.88
	D_{30}	34,284	44,191	1.29	21,512	24,055	1.12	1.84
	D_{50}	33,989	44,158	1.30	21,477	23,736	1.11	1.86
BU	D_{10}	20,799	28,909	1.39	11,453	14,017	1.22	2.06
	D_{30}	21,174	28,904	1.37	11,218	14,189	1.26	2.04
	D_{50}	21,166	28,857	1.36	11,139	13,982	1.26	2.06

Performance - Sequential vs One-Thread

Approach & Dataset		Subgoal Sharing (SS) Time T_{seq}	Subgoal Sharing (SS) Time T_1	Subgoal Sharing (SS) Ratio T_1/T_{seq}	Multi-Dimensional (MD) Time T_{seq}	Multi-Dimensional (MD) Time T_1	Multi-Dimensional (MD) Ratio T_1/T_{seq}	SS vs MD Ratio SS_{T_1}/MD_{T_1}
Knapsack Problem								
TD_{no}	D_{10}	9,508	12,415	1.31	4,275		1.23	2.37
	D_{30}	9,246	12,177	1.32	4,196		1.27	2.28
	D_{50}	9,480	12,589	1.33	4,275		1.28	2.31
TD_{rnd}	D_{10}	19,667	24,444	1.24	10,462		1.12	2.08
	D_{30}	19,847	25,609	1.29	10,508		1.14	2.14
	D_{50}	19,985	25,429	1.27	10,805		1.11	2.12
BU	D_{10}	12,614	17,940	1.42	7,001	7,668	1.10	2.34
	D_{30}	12,364	17,856	1.44	7,005	7,786	1.11	2.29
	D_{50}	12,653	17,499	1.38	6,775	7,637	1.13	2.29
LCS Problem								
TD_{no}	D_{10}	21,191	26,225	1.24	16,202	18,046	1.11	1.45
	D_{30}	20,809	26,146	1.26	16,006	18,067	1.13	1.45
	D_{50}	20,775	26,028	1.25	16,259	18,195	1.12	1.43
TD_{rnd}	D_{10}	34,565	44,371	1.28	21,525	23,635	1.10	1.88
	D_{30}	34,284	44,191	1.29	21,512	24,055	1.12	1.84
	D_{50}	33,989	44,158	1.30	21,477	23,736	1.11	1.86
BU	D_{10}	20,799	28,909	1.39	11,453	14,017	1.22	2.06
	D_{30}	21,174	28,904	1.37	11,218	14,189	1.26	2.04
	D_{50}	21,166	28,857	1.36	11,139	13,982	1.26	2.06



Performance - The Knapsack Problem

Approach & Dataset		Sequential Time (T_{seq})	# Threads (p) Time (T_1)	Speedup (T_1/T_p) 8	16	24	32	Best Time (T_{best})
		1	1					
Subgoal Sharing (SS)								
TD_{no}	D_{10}	9,508	12,415	n.c.	n.c.	n.c.	n.c.	9,508
	D_{30}	9,246	12,177	n.c.	n.c.	n.c.	n.c.	9,246
	D_{50}	9,480	12,589	n.c.	n.c.	n.c.	n.c.	9,480
TD_{rnd}	D_{10}	19,667	24,444	6.78	12.35	15.44	18.19	1,344
	D_{30}	19,847	25,609	7.15	13.83	17.37	20.47	1,251
	D_{50}	19,985	25,429	7.27	13.70	17.35	20.62	1,233
BU	D_{10}	12,614	**17,940**	7.17	13.97	18.31	**22.15**	**0,810**
	D_{30}	12,364	**17,856**	7.23	13.78	18.26	**21.94**	**0,814**
	D_{50}	12,653	**17,499**	7.25	14.01	18.34	**21.76**	**0,804**

Performance - The Knapsack Problem

Approach & Dataset		Sequential Time (T_{seq}) 1	# Threads (p) Time (T_1) 1	Speedup (T_1/T_p) 8	16	24	32	Best Time (T_{best})
Subgoal Sharing (SS)								
TD_{no}	D_{10}	9,508	12,415	n.c.	n.c.	n.c.	n.c.	9,508
	D_{30}	9,246	12,177	n.c.	n.c.	n.c.	n.c.	9,246
	D_{50}	9,480	12,589	n.c.	n.c.	n.c.	n.c.	9,480
TD_{rnd}	D_{10}	19,667	24,444	6.78	12.35	15.44	18.19	1,344
	D_{30}	19,847	25,609	7.15	13.83	17.37	20.47	1,251
	D_{50}	19,985	25,429	7.27	13.70	17.35	20.62	1,233
BU	D_{10}	12,614	**17,940**	7.17	13.97	18.31	**22.15**	**0,810**
	D_{30}	12,364	**17,856**	7.23	13.78	18.26	**21.94**	**0,814**
	D_{50}	12,653	**17,499**	7.25	14.01	18.34	**21.76**	**0,804**
Multi-Dimensional (MD)								
TD_{no}	D_{10}	4,275	5,241	n.c.	n.c.	n.c.	n.c.	4,275
	D_{30}	4,196	5,336	n.c.	n.c.	n.c.	n.c.	4,196
	D_{50}	4,275	5,457	n.c.	n.c.	n.c.	n.c.	4,275
TD_{rnd}	D_{10}	10,462	11,740	6.90	12.90	16.22	19.09	0,615
	D_{30}	10,508	11,959	7.31	14.04	18.01	21.59	0,554
	D_{50}	10,805	11,982	7.36	14.03	17.96	21.63	0,554
BU	D_{10}	7,001	**7,668**	7.24	13.77	17.55	**21.42**	**0,358**
	D_{30}	7,005	**7,786**	7.40	14.13	18.02	**22.18**	**0,351**
	D_{50}	6,775	**7,637**	7.37	13.96	18.10	**21.95**	**0,348**

Performance - The Knapsack Problem

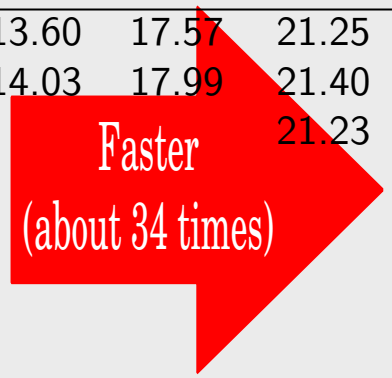
Approach & Dataset		Sequential Time (T_{seq}) 1	# Threads (p) Time (T_1) 1	Speedup (T_1/T_p) 8	16	24	32	Best Time (T_{best})
Subgoal Sharing (SS)								
TD_{no}	D_{10}	9,508	12,415	n.c.	n.c.	n.c.	n.c.	9,508
	D_{30}	9,246	12,177	n.c.	n.c.	n.c.	n.c.	9,246
	D_{50}	9,480	12,589	n.c.	n.c.	n.c.	n.c.	9,480
TD_{rnd}	D_{10}	19,667	24,444	6.78	12.35	15.44	18.19	1,344
	D_{30}	19,847	25,609	7.15	13.83	17.37	20.47	1,251
	D_{50}	19,985	25,429	7.27	13.70	17.35	20.62	1,233
BU	D_{10}	12,614	**17,940**	7.17	13.97	18.31	**22.15**	**0,810**
	D_{30}	12,364	**17,856**	7.23	13.78	18.26	**21.94**	**0,814**
	D_{50}	12,653	**17,499**	7.25	14.01	18.34	**21.76**	**0,804**
Multi-Dimensional (MD)								
TD_{no}	D_{10}	4,275	5,241	n.c.	n.c.	n.c.	n.c.	4,275
	D_{30}	4,196	5,336	n.c.	n.c.	n.c.	n.c.	4,196
	D_{50}	4,275	5,457	n.c.	n.c.	n.c.	n.c.	4,275
TD_{rnd}	D_{10}	10,462	11,740	6.90	12.90	16.22	19.09	0,615
	D_{30}	10,508	11,959	7.31	14.04	18.01	21.59	0,554
	D_{50}	10,805	11,982	7.36	14.03	17.96	21.63	0,554
BU	D_{10}	7,001	**7,668**	7.24	13.77	17.55	**21.42**	**0,358**
	D_{30}	7,005	**7,786**	7.40	14.13	18.02	**22.18**	**0,351**
	D_{50}	6,775	**7,637**	7.37	13.96	18.10	**21.95**	**0,348**

Performance - The LCS Problem

Approach & Dataset		Sequential Time (T_{seq}) 1	# Threads (p) Time (T_1) 1	Speedup (T_1/T_p) 8	16	24	32	Best Time (T_{best})
Subgoal Sharing (SS)								
TD_{no}	D_{10}	21,191	26,225	n.c.	n.c.	n.c.	n.c.	21,191
	D_{30}	20,809	26,146	n.c.	n.c.	n.c.	n.c.	20,809
	D_{50}	20,775	26,028	n.c.	n.c.	n.c.	n.c.	20,775
TD_{rnd}	D_{10}	34,565	44,371	7.23	13.23	16.45	19.74	2,248
	D_{30}	34,284	44,191	7.12	13.09	16.52	19.77	2,235
	D_{50}	33,989	44,158	7.06	13.30	16.49	19.58	2,255
BU	D_{10}	20,799	**28,909**	6.47	12.21	16.48	**20.32**	**1,423**
	D_{30}	21,174	**28,904**	6.94	12.61	16.63	**20.40**	**1,417**
	D_{50}	21,166	**28,857**	6.44	12.31	16.44	**20.52**	**1,406**
Multi-Dimensional (MD)								
TD_{no}	D_{10}	16,202	18,046	n.c.	n.c.	n.c.	n.c.	16,202
	D_{30}	16,006	18,067	n.c.	n.c.	n.c.	n.c.	16,006
	D_{50}	16,259	18,195	n.c.	n.c.	n.c.	n.c.	16,259
TD_{rnd}	D_{10}	21,525	23,635	7.31	13.60	17.57	21.25	1,112
	D_{30}	21,512	24,055	7.46	14.03	17.99	21.40	1,124
	D_{50}	21,477	23,736	7.33	13.76	17.78	21.23	1,118
BU	D_{10}	11,453	**14,017**	6.90	12.14	16.89	**22.21**	**0,631**
	D_{30}	11,218	**14,189**	6.88	13.10	17.01	**22.49**	**0,631**
	D_{50}	11,139	**13,982**	6.87	13.06	16.85	**22.55**	**0,620**

Performance - The LCS Problem

Approach & Dataset		Sequential Time (T_{seq}) 1	# Threads (p) Time (T_1) 1	Speedup (T_1/T_p) 8	16	24	32	Best Time (T_{best})
Subgoal Sharing (SS)								
TD_{no}	D_{10}	21,191	26,225	n.c.	n.c.	n.c.	n.c.	21,191
	D_{30}	20,809	26,146	n.c.	n.c.	n.c.	n.c.	20,809
	D_{50}	20,775	26,028	n.c.	n.c.	n.c.	n.c.	20,775
TD_{rnd}	D_{10}	34,565	44,371	7.23	13.23	16.45	19.74	2,248
	D_{30}	34,284	44,191	7.12	13.09	16.52	19.77	2,235
	D_{50}	33,989	44,158	7.06	13.30	16.49	19.58	2,255
BU	D_{10}	20,799	**28,909**	6.47	12.21	16.48	**20.32**	**1,423**
	D_{30}	21,174	**28,904**	6.94	12.61	16.63	**20.40**	**1,417**
	D_{50}	21,166	**28,857**	6.44	12.31	16.44	**20.52**	**1,406**
Multi-Dimensional (MD)								
TD_{no}	D_{10}	16,202	18,046	n.c.	n.c.	n.c.	n.c.	16,202
	D_{30}	16,006	18,067	n.c.	n.c.	n.c.	n.c.	16,006
	D_{50}	16,259	18,195	n.c.	n.c.	n.c.	n.c.	16,259
TD_{rnd}	D_{10}	21,525	23,635	7.31	13.60	17.57	21.25	1,112
	D_{30}	21,512	24,055	7.46	14.03	17.99	21.40	1,124
	D_{50}	21,477	23,736	7.33			21.23	1,118
BU	D_{10}	11,453	**14,017**	6.90				**0,631**
	D_{30}	11,218	**14,189**	6.88				**0,631**
	D_{50}	11,139	**13,982**	6.87				**0,620**



Conclusions and Further Work

- In this work, we have integrated in the **YapTab-Mt** a novel table space design, which is based in **multi-dimensional arrays**:
  - Specially aimed for a **fast evaluation** of **dynamic programming** problems with single solutions and multiple integer dimensions.
  - A simple **dim** instruction allows the users to define the size of the **dimensions** in table space.

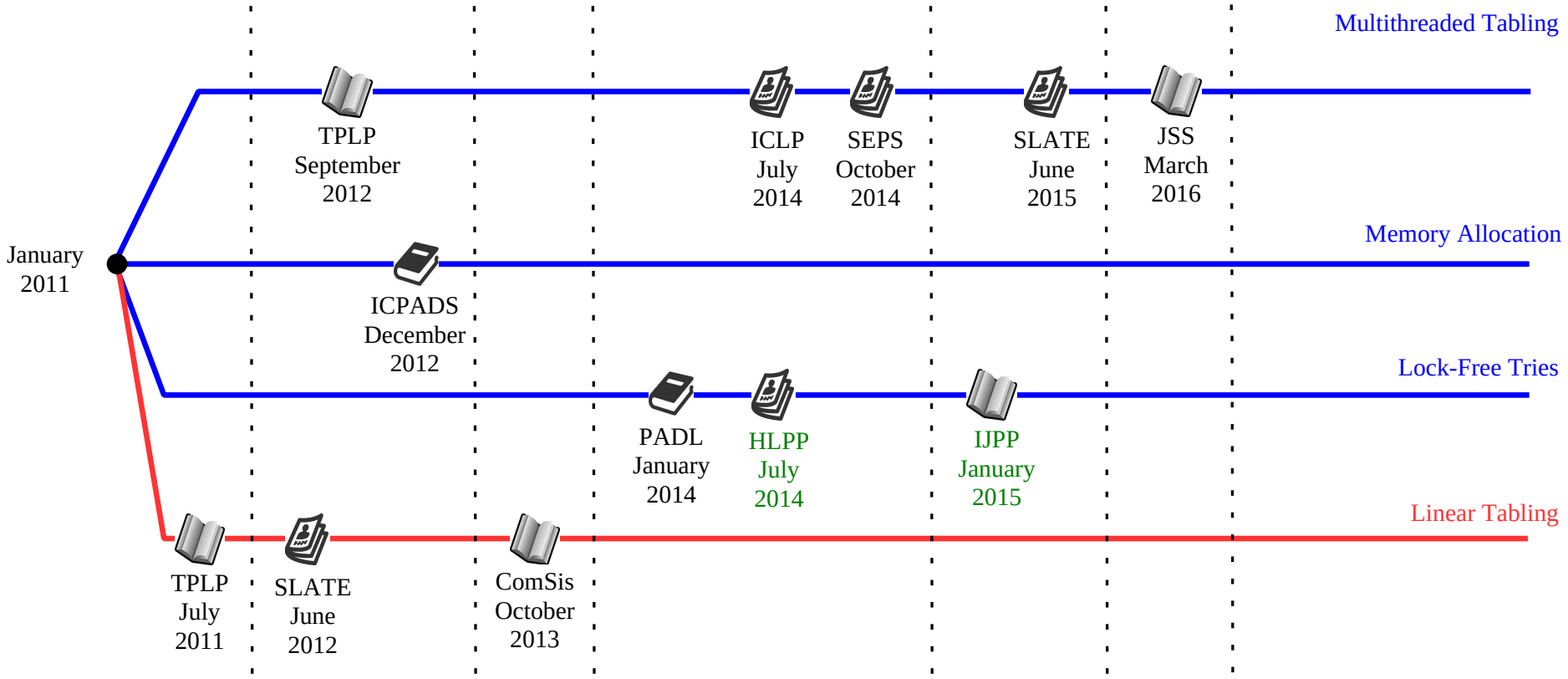
Conclusions and Further Work

- In this work, we have integrated in the **YapTab-Mt** a novel table space design, which is based in **multi-dimensional arrays**:
  - Specially aimed for a **fast evaluation** of **dynamic programming** problems with single solutions and multiple integer dimensions.
  - A simple **dim** instruction allows the users to define the size of the **dimensions** in table space.
  - **Single-Threaded** evaluations are **faster** than the original design.
  - **Multi-Threaded** evaluations **benefit** from an **highly-concurrent** design (at the implementation level we use **lock-free** techniques).

Conclusions and Further Work

- In this work, we have integrated in the **YapTab-Mt** a novel table space design, which is based in **multi-dimensional arrays**:
  - Specially aimed for a **fast evaluation** of **dynamic programming** problems with single solutions and multiple integer dimensions.
  - A simple **dim** instruction allows the users to define the size of the **dimensions** in table space.
  - **Single-Threaded** evaluations are **faster** than the original design.
  - **Multi-Threaded** evaluations **benefit** from an **highly-concurrent** design (at the implementation level we use **lock-free** techniques).
- **Further work** will include:
  - **Compare the performance** against domain-specific languages and libraries for dynamic programming, such as **DPSKEL** or **MALLBA**.
  - **Explore impact** of the design in other application domains (test-drive used the **Knapsack** and the **LCS** problems).

Research Outline



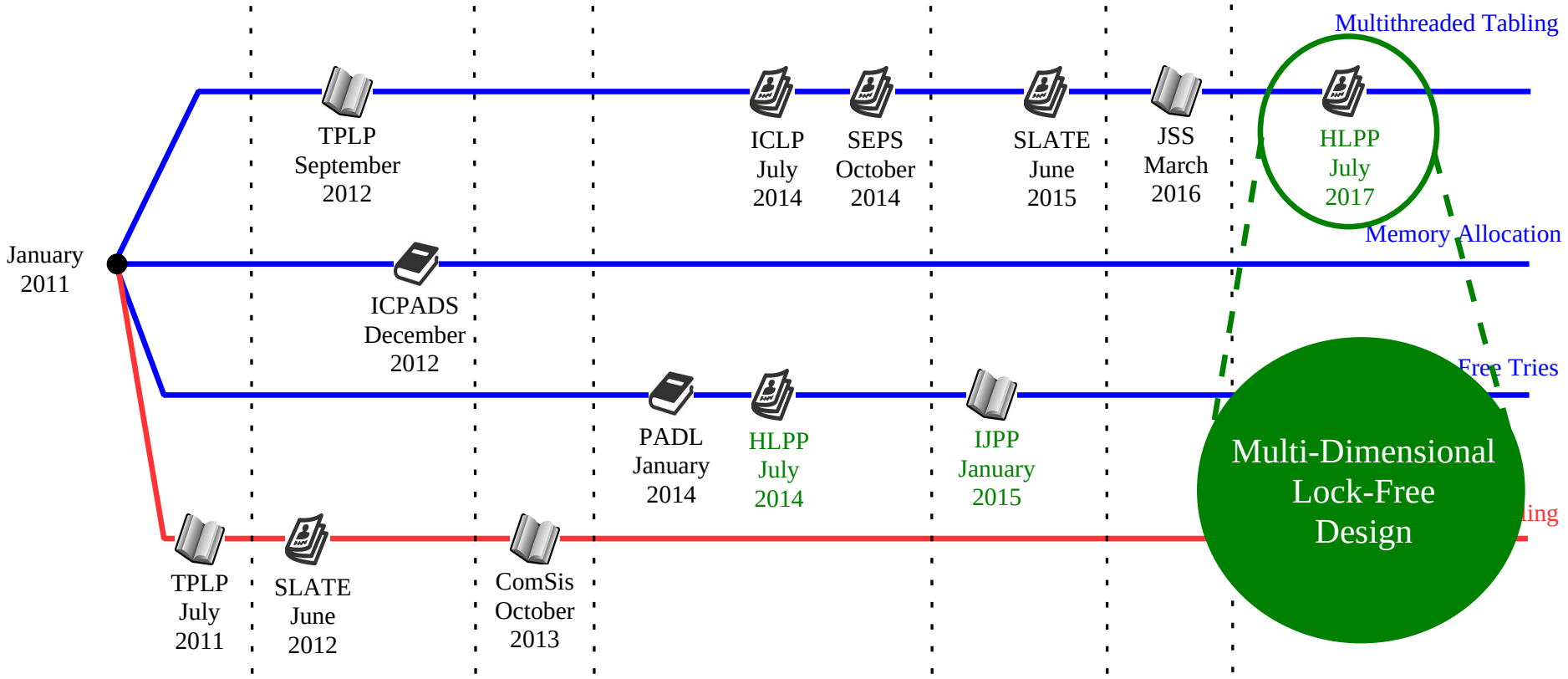
Work related:

- 4 Journals
- 2 Book Series
- 4 Workshop Proceedings

Others:

- 2 Journals
- 0 Book Series
- 1 Workshop Proceeding

Research Outline



Thank You !!!

FCT Grant: *SFRH/BPD/108018/2015*