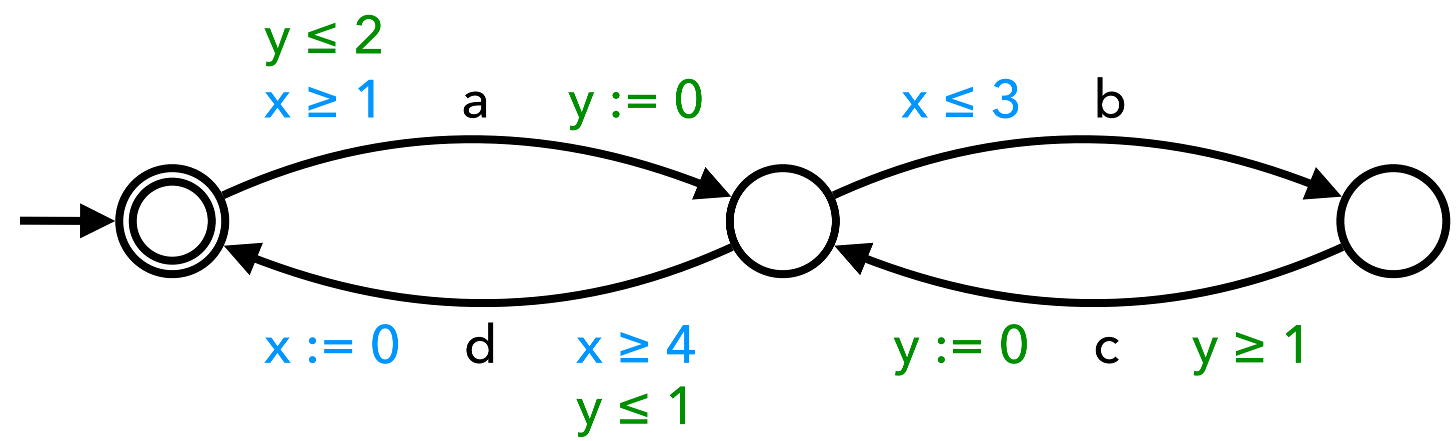
S. AKSHAY, IIT BOMBAY
PAUL GASTIN, LSV ENS PARIS-SACLAY (ENS CACHAN)
S. KRISHNA, IIT BOMBAY

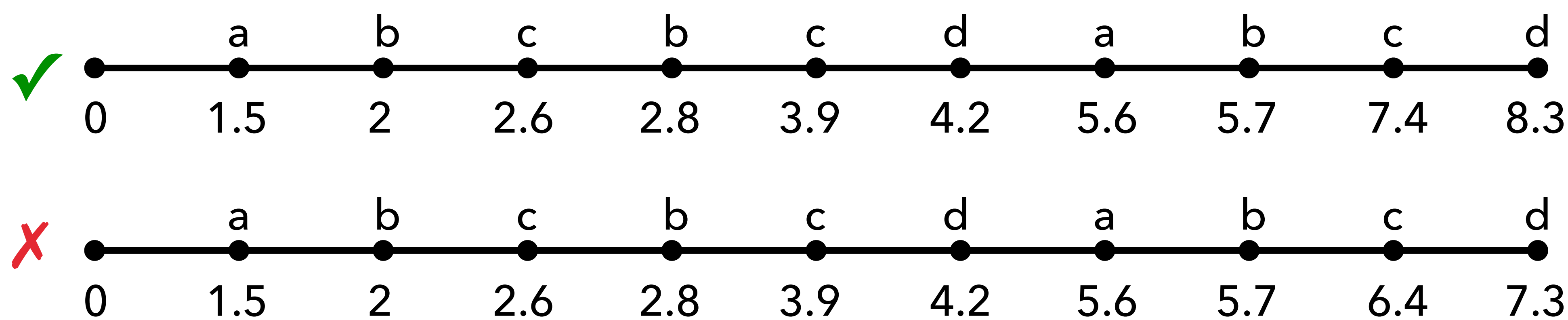
ANALYZING TIMED SYSTEMS USING TREE AUTOMATA

CONCUR 2016

TIMED AUTOMATON



TIMED WORD



TIMED WORD LANGUAGE

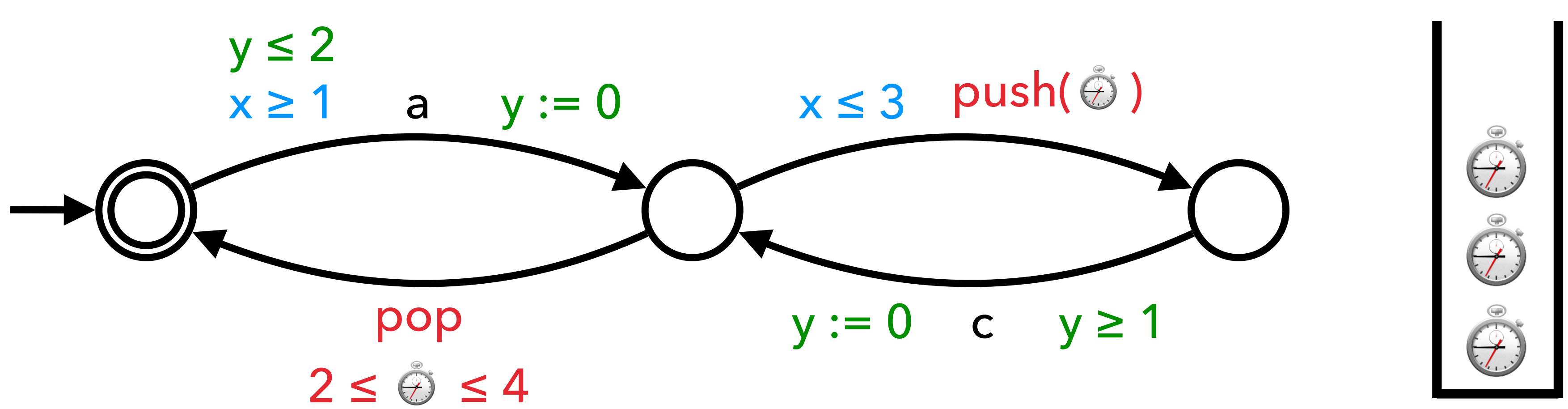
$\mathscr{L}_{\mathrm{T}}(\mathcal{A})$

NON-EMPTINESS / REACHABILITY PROBLEM

$$\mathscr{L}_{\mathrm{T}}(\mathcal{A}) \neq \emptyset$$

EMPTINESS FOR (PUSHDOWN) TIMED AUTOMATON

- Well-studied problem with standard approach
  - Timed automata (TA): Region construction [Alur-Dill'90]
    Many optimizations
  - Pushdown timed automata (PDTA): Lifting region construction [Bouajjani et al. '94] [Abdulla et al. '12]
  - Common feature:
    - represent behaviors as timed words
    - use abstractions to reduce to finite automata



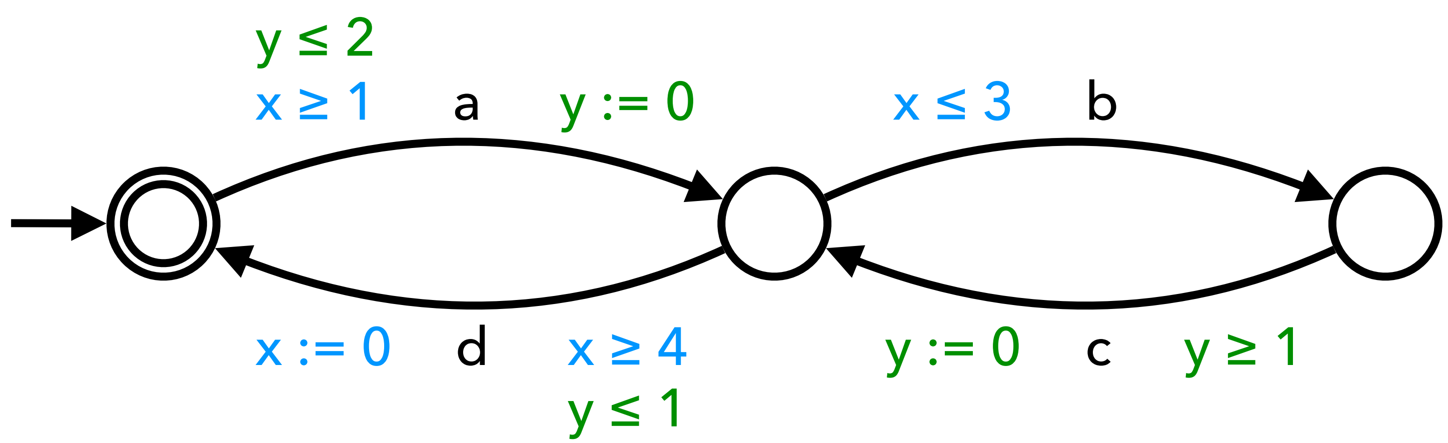
EMPTINESS FOR (PUSHDOWN) TIMED AUTOMATON

- Well-studied problem with standard approach
  - Timed automata (TA): Region construction [Alur-Dill'90]
    Many optimizations
  - Pushdown timed automata (PDTA): Lifting region construction [Bouajjani et al. '94] [Abdulla et al. '12]
  - Common feature:
    - represent behaviors as timed words
    - use abstractions to reduce to finite automata
- Our new approach
  - represent behaviors as graphs: words with timing constraints
  - Interpret graphs in trees to reduce to tree automata
    - High level and powerful technique
    - Simpler and uniform proofs for more complicated systems
    - New technique not relying on regions/zones

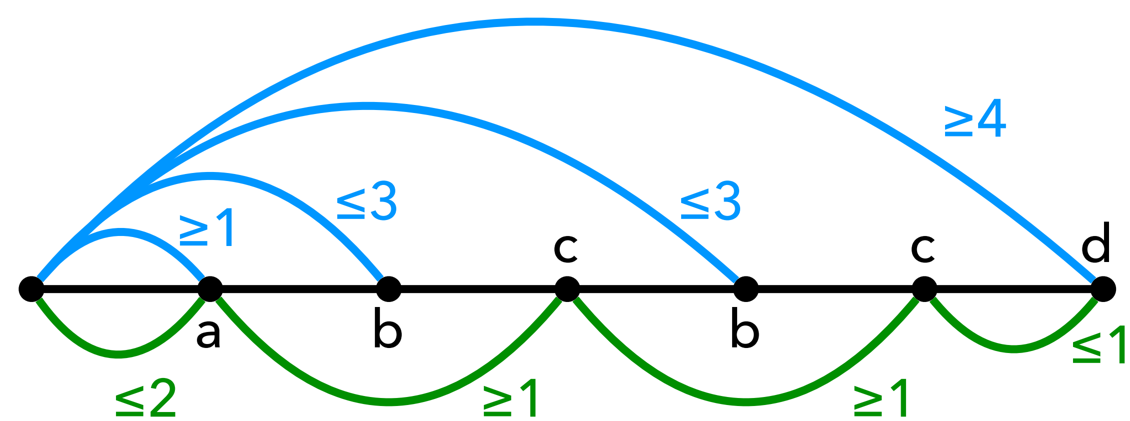
OUTLINE

- BEHAVIOURS AS GRAPHS
- DECIDING GRAPH PROPERTIES
- DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
- TREE-WIDTH FOR TIMED SYSTEMS
- INTERPRETING GRAPHS IN TREES
- CONCLUSION

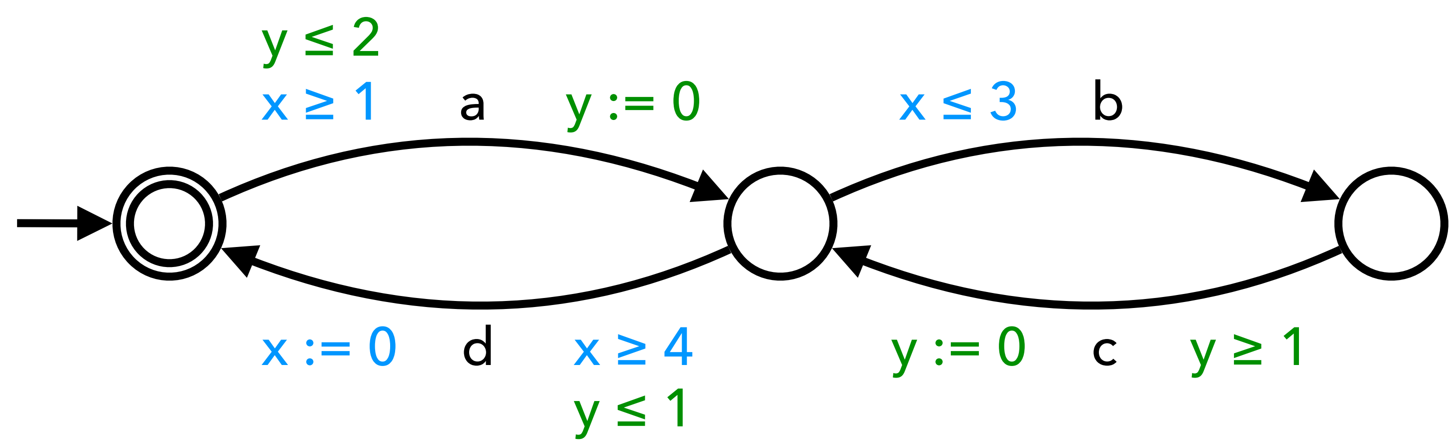
BEHAVIORS AS GRAPHS: TIMED SYSTEMS



TC-WORDS: WORDS WITH TIMING CONSTRAINTS



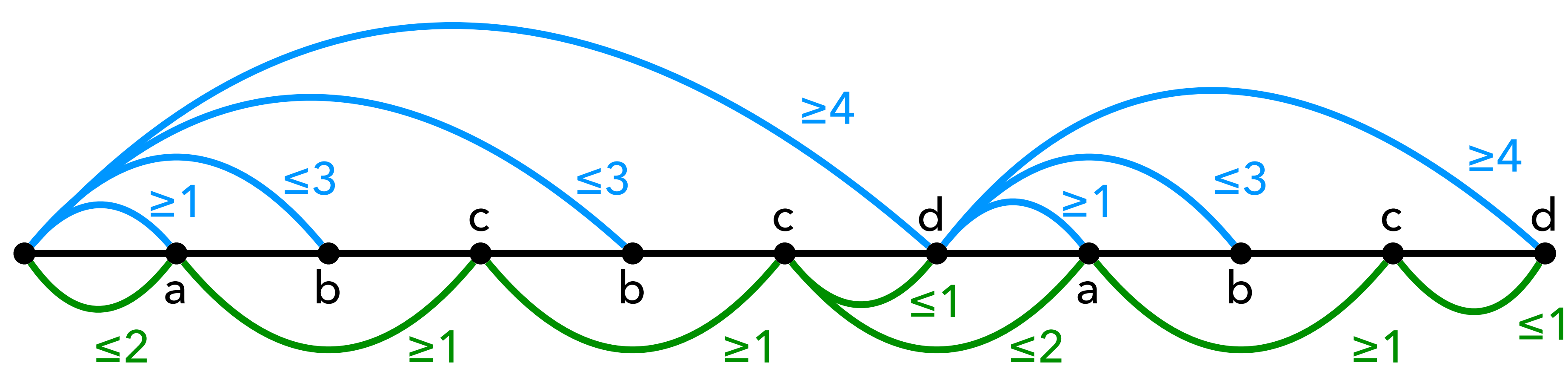
BEHAVIORS AS GRAPHS: TIMED SYSTEMS



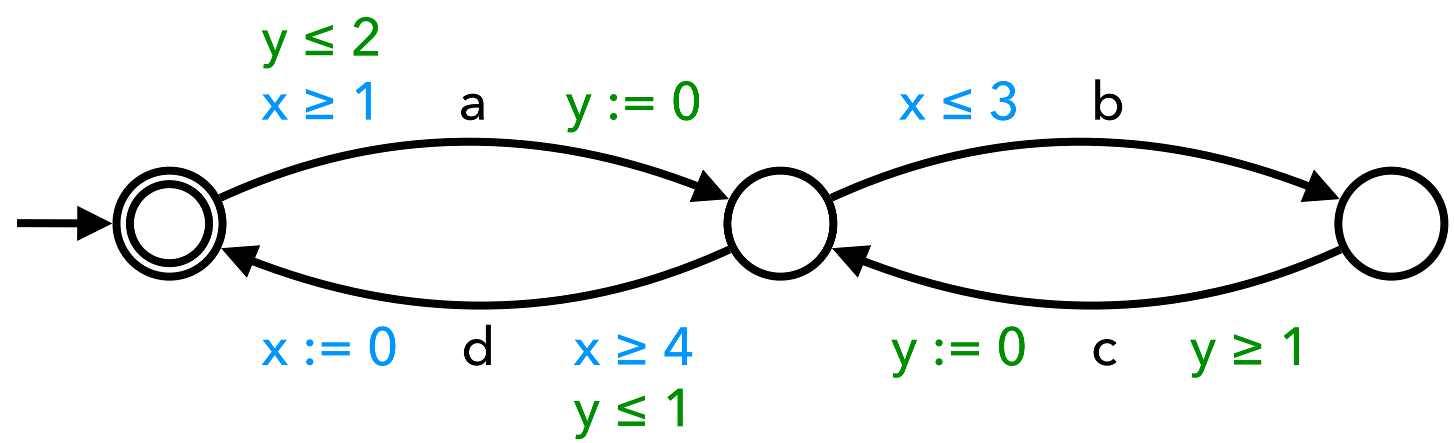
TC-WORDS: WORDS WITH TIMING CONSTRAINTS

TC-WORD LANGUAGE: $\mathscr{L}_{\mathrm{TCW}}(\mathcal{A})$

- Every accepting path ρ in the timed system generates one TC-word $\mathrm{tcw}(\rho) \in \mathscr{L}_{\mathrm{TCW}}(\mathcal{A})$



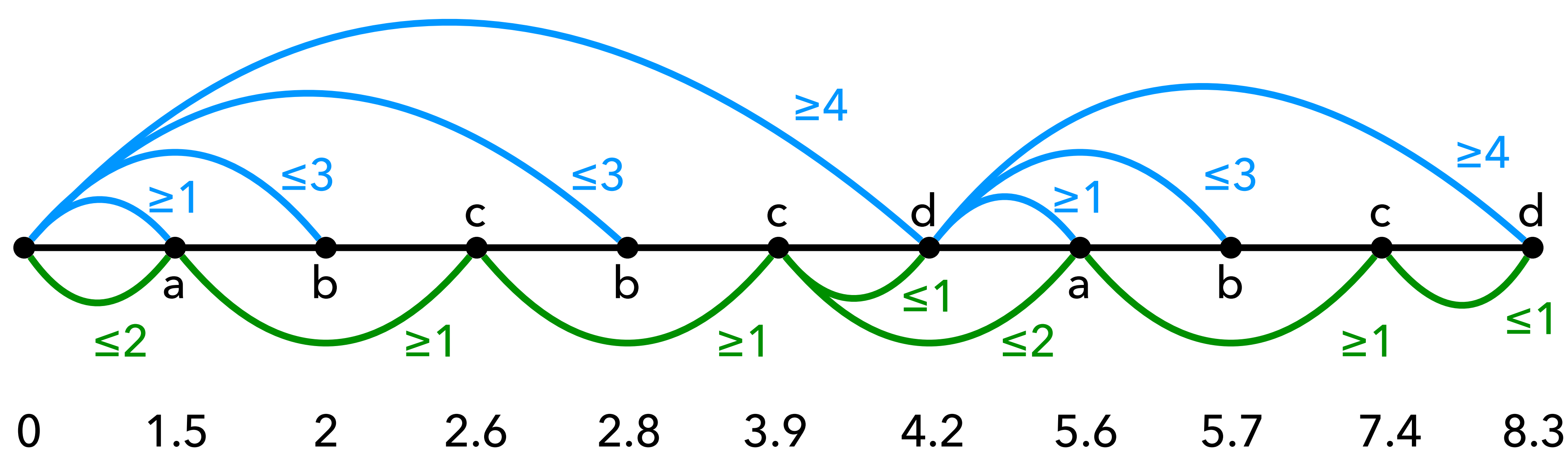
BEHAVIORS AS GRAPHS: TIMED SYSTEMS



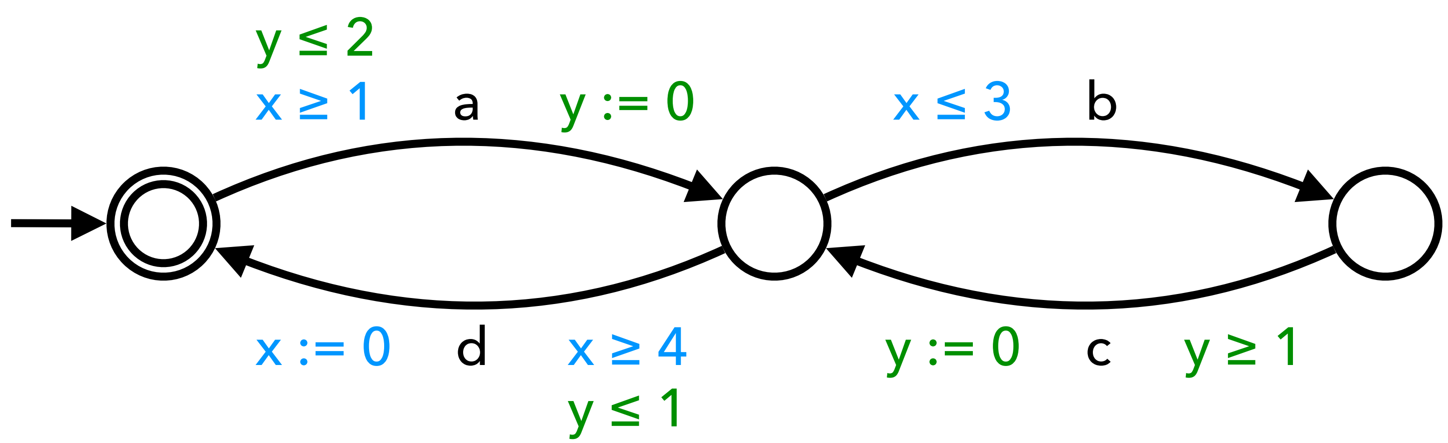
TC-WORDS: WORDS WITH TIMING CONSTRAINTS

TC-WORD LANGUAGE: $\mathscr{L}_{\mathsf{TCW}}(\mathcal{A})$

REALIZABLE TC-WORDS: $\mathfrak{Real}_{\mathsf{TCW}}$



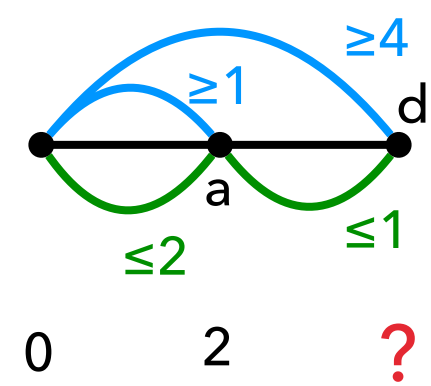
BEHAVIORS AS GRAPHS: TIMED SYSTEMS



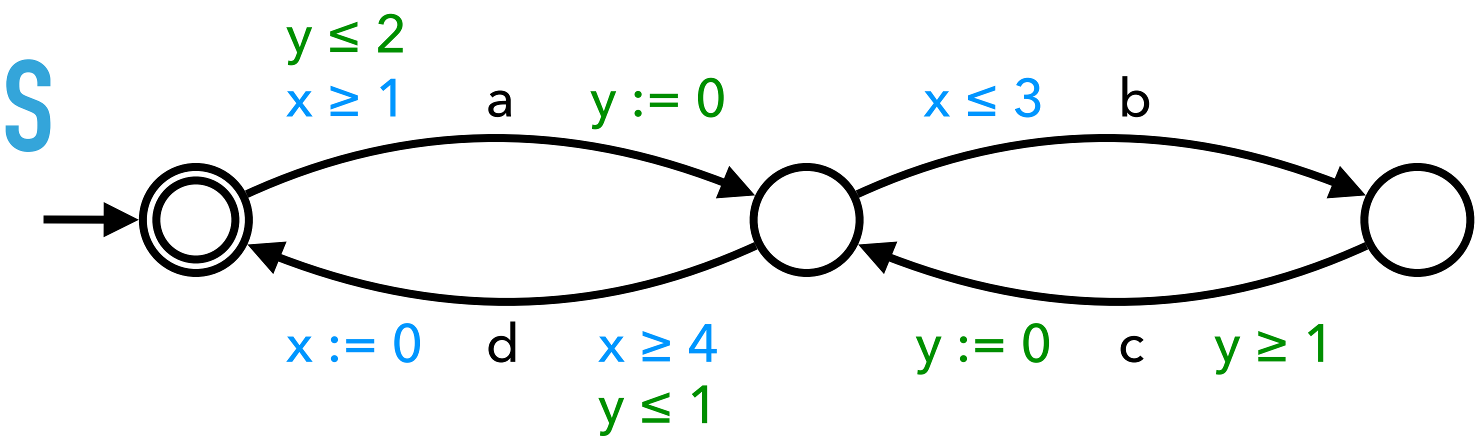
TC-WORDS: WORDS WITH TIMING CONSTRAINTS

TC-WORD LANGUAGE: $\mathscr{L}_{\mathsf{TCW}}(\mathcal{A})$

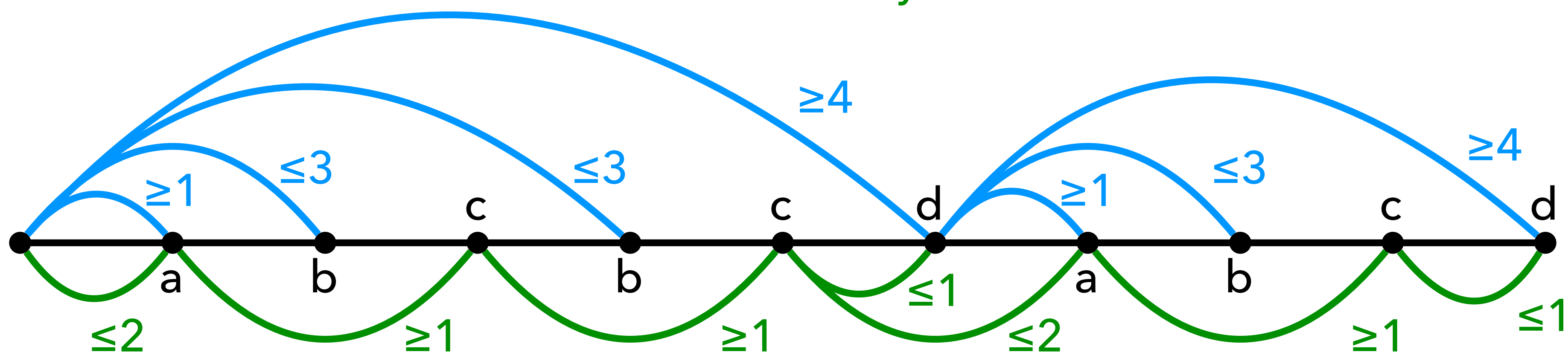
REALIZABLE TC-WORDS: $\mathfrak{Real}_{\mathsf{TCW}}$



REALIZATIONS OF TC-WORDS

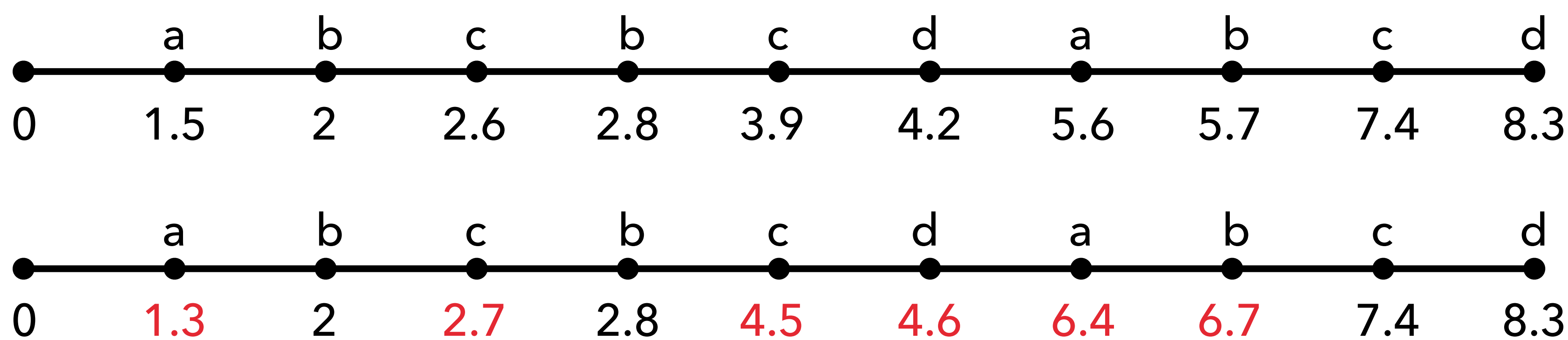


TC-WORD

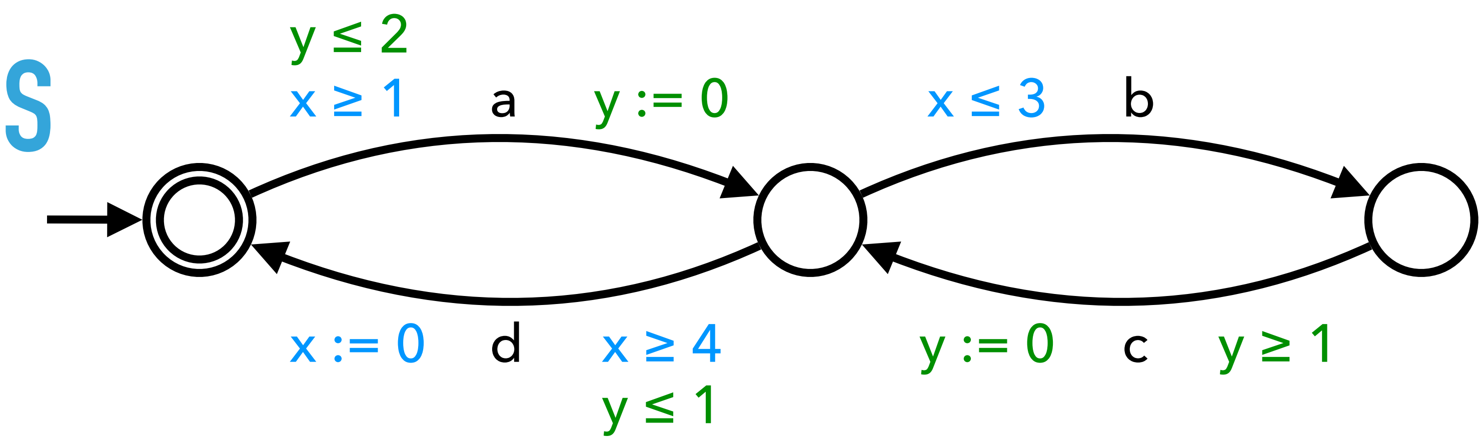


REALIZATION

TIMED-WORDS



TIMED WORDS vs TC-WORDS

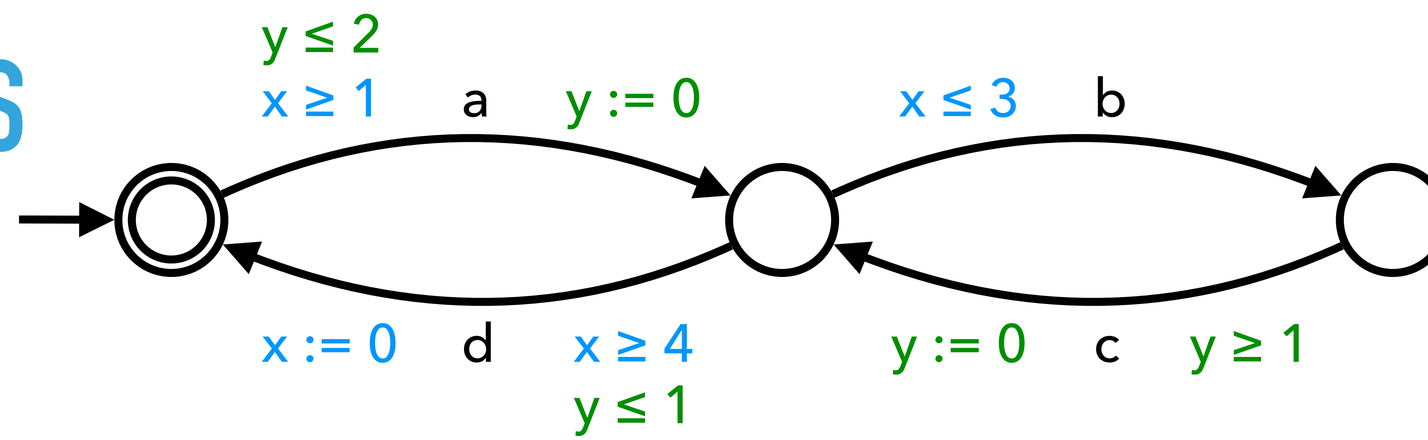


TIMED-WORDS	TC-WORDS
UNCOUNTABLY MANY REALIZATIONS	ONE TC-WORD
NO REALIZATIONS	ONE TC-WORD
WORDS OVER AN INFINITE ALPHABET	GRAPHS OVER A FINITE SIGNATURE

$$\mathscr{L}_{\mathrm{T}}(\mathcal{A}) = \mathrm{Realizations}(\mathcal{L}_{\mathrm{TCW}}(\mathcal{A}))$$

$$\mathscr{L}_{\mathrm{T}}(\mathcal{A}) \neq \emptyset \quad \longleftrightarrow \quad \mathscr{L}_{\mathrm{TCW}}(\mathcal{A}) \cap \mathfrak{Real}_{\mathrm{TCW}} \neq \emptyset$$

TIMED WORDS vs TC-WORDS



TIMED-WORDS	TC-WORDS
UNCOUNTABLY MANY REALIZATIONS	ONE TC-WORD
NO REALIZATIONS	ONE TC-WORD
WORDS OVER AN INFINITE ALPHABET	GRAPHS OVER A FINITE SIGNATURE

$$\mathscr{L}_{\mathrm{T}}(\mathcal{A}) = \mathrm{Realizations}(\mathcal{L}_{\mathrm{TCW}}(\mathcal{A}))$$

$$\mathscr{L}_{\mathrm{T}}(\mathcal{A}) \neq \emptyset \quad \longleftrightarrow \quad \mathscr{L}_{\mathrm{TCW}}(\mathcal{A}) \cap \mathfrak{Real}_{\mathrm{TCW}} \neq \emptyset$$

THIS IS A GRAPH PROPERTY!

OUTLINE

- BEHAVIOURS AS GRAPHS
- DECIDING GRAPH PROPERTIES
- DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
- TREE-WIDTH FOR TIMED SYSTEMS
- INTERPRETING GRAPHS IN TREES
- CONCLUSION

COURCELLE'S THEOREM

- Let TW_k be the set of graphs of tree-width at most k
- Let P be a property of graphs
- If P is MSO-definable then $P \cap TW_k \neq \varnothing$ is decidable

- Graphs in TW_k can be interpreted in trees (k-terms)
- Let P be an MSO-definable property of graphs
- Φ_P MSO over graphs ⇨ Φ^k_P MSO over trees (k-terms)
- Then $P \cap TW_k \neq \varnothing$ iff Φ^k_P satisfiable over trees (k-terms)
- THATCHER&WRIGHT'68: REDUCTION TO EMPTINESS OF TREE AUTOMATA

COURCELLE'S THEOREM

HOW DO WE GET A GOOD COMPLEXITY?

- Let TW_k be the set of graphs of tree-width at most k
- Let P be a property of graphs
- If P is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

- Graphs in TW_k can be interpreted in trees (k-terms)
- Let P be a property of graphs
- Build directly a tree automaton $\mathcal{A}^k{}_P$ accepting k-terms denoting graphs satisfying P
- Then $P \cap TW_k \neq \emptyset$ iff $\mathscr{L}(\mathcal{A}^k{}_P) \neq \emptyset$

CONCUR'16

COURCELLE'S THEOREM

- Let TW_k be the set of graphs of tree-width at most k
- Let P be a property of graphs
- If P is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

WE WANT TO SOLVE $\mathscr{L}_{\mathrm{TCW}}(\mathcal{A}) \cap \mathfrak{Real}_{\mathrm{TCW}} \neq \emptyset$

- Show that TC-words have bounded tree-width
- Show that our properties are MSO-definable
- Build directly tree automata for our properties

CONCUR'16
SPLIT-WIDTH

CONCUR'16

RELATED WORKS

TC-WORDS ARE QUITE DIFFERENT GRAPHS

REALIZABILITY IS THE MAIN CHALLENGE

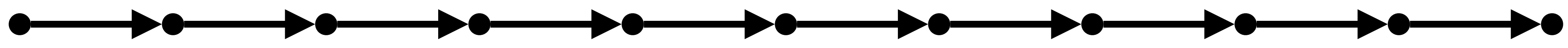
- P. Madhusudan & G. Parlato, POPL'11
  The tree-width of auxiliary storage
- C. Aiswarya, PG & K. Narayan Kumar, CONCUR'12
  MSO decidability of multi-pushdown systems via split-width
- C. Aiswarya PhD'14
  Verification of communicating recursive programs via split-width
- C. Aiswarya & PG, FSTTCS'14
  Reasoning about distributed systems: WYSIWYG

OUTLINE

- BEHAVIOURS AS GRAPHS
- DECIDING GRAPH PROPERTIES
- DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
- TREE-WIDTH FOR TIMED SYSTEMS
- INTERPRETING GRAPHS IN TREES
- CONCLUSION

MSO-DEFINABLE GRAPH PROPERTIES

G = (V , →) IS A WORD: LINEAR ORDER



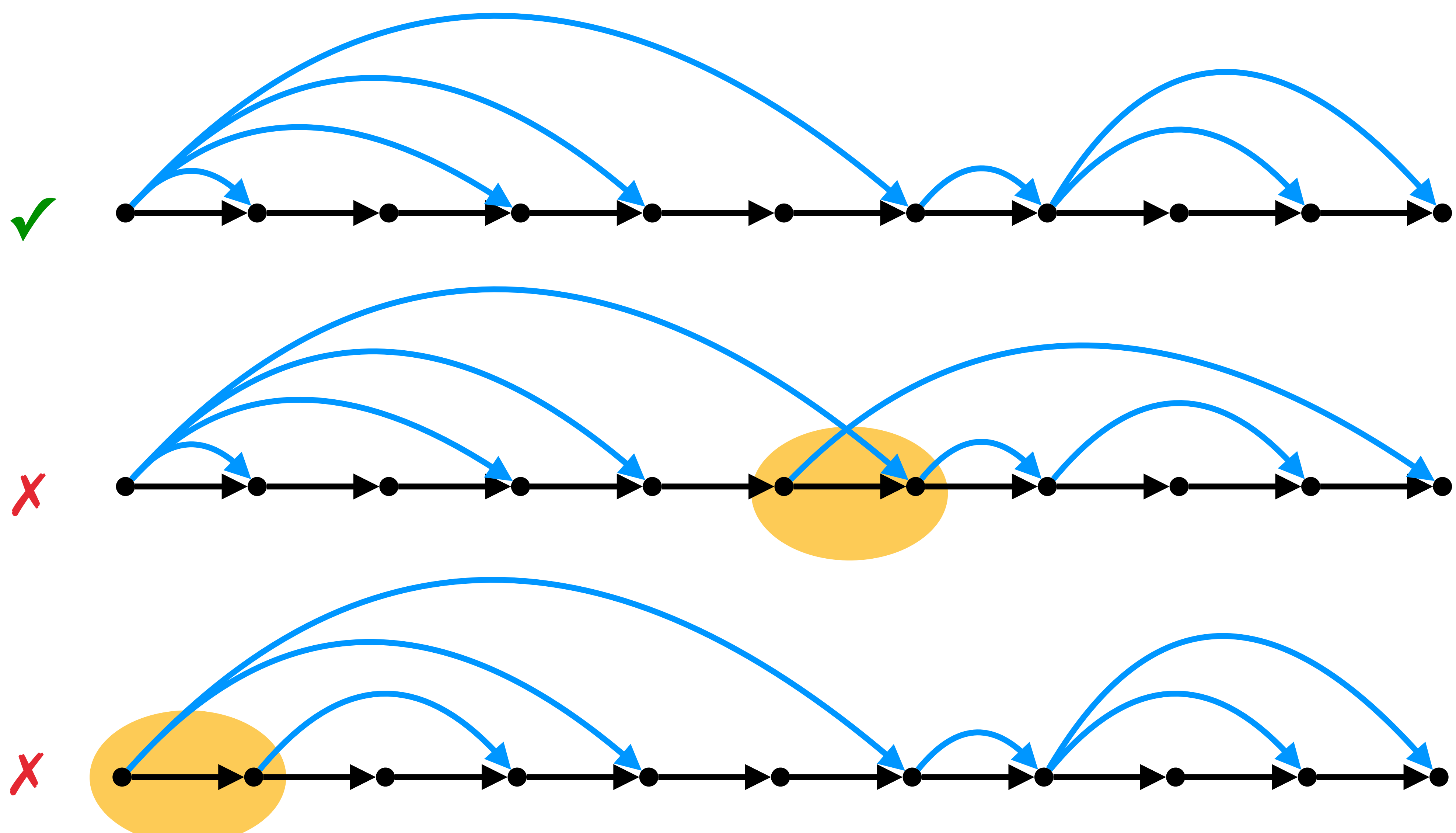
$$\mathsf{Word}(\rightarrow) ::= \forall x, y, z \left(\neg(x \rightarrow^+ x) \wedge (x = y \vee x \rightarrow^+ y \vee y \rightarrow^+ x) \wedge \neg(x \rightarrow z \vee x \rightarrow y \rightarrow^+ z)\right)$$

MSO-DEFINABLE GRAPH PROPERTIES

G = (V , → , $\curvearrowright$) IS A 1-CLOCK TC-WORD

$$\text{Forward}(\curvearrowright) ::= \forall x, y\ (x \curvearrowright y \implies x < y)$$

$$\text{Clock}(\curvearrowright) ::= \neg\exists x, y, x', y'\ (x \curvearrowright y \wedge x' \curvearrowright y' \wedge x < x' < y)$$

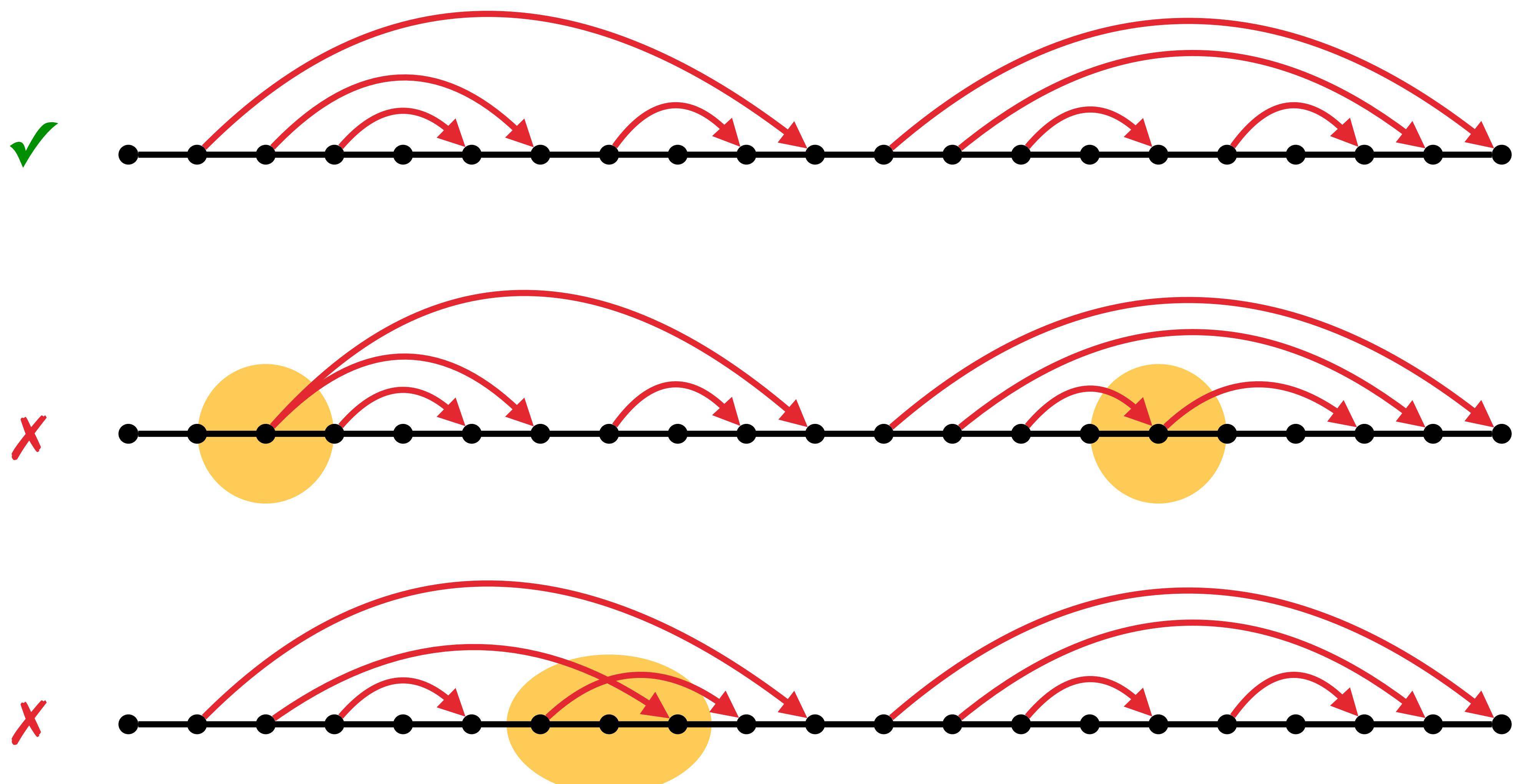


MSO-DEFINABLE GRAPH PROPERTIES

G = (V , → , ↷) IS A 1-STACK TC-WORD

$$\mathsf{Forward}(\curvearrowright) ::= \forall x, y\ (x \curvearrowright y \implies x < y)$$

$$\mathsf{Stack}(\curvearrowright) ::= \neg\exists x, y, x', y'\ (x \curvearrowright y \wedge x' \curvearrowright y' \wedge (y = x' \vee x \leqslant x' < y < y' \vee x < x' < y \leqslant y'))$$



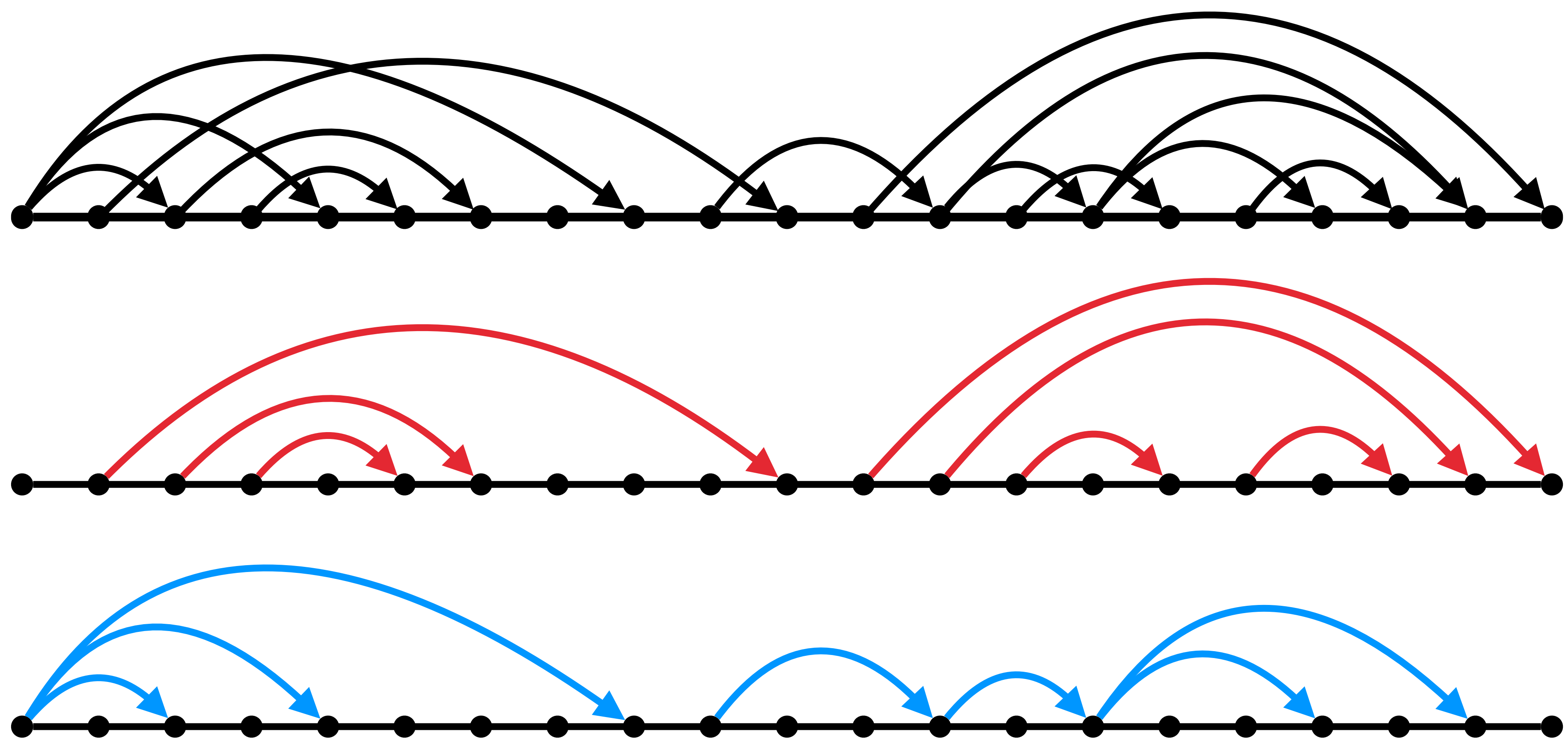
MSO-DEFINABLE GRAPH PROPERTIES

G = (V , → , ⌒) IS AN M-STACKS N-CLOCKS TC-WORD

$$\text{Forward}(\curvearrowright) \wedge \exists \overline{R} = (R_1, \ldots, R_n)\ \exists \overline{S} = (S_1, \ldots, S_m)$$

$$\text{Partition}(\curvearrowright, \overline{R}, \overline{S}) \wedge \bigwedge_{i=1}^{n} \text{Clock}(R_i) \wedge \bigwedge_{i=1}^{m} \text{Stack}(S_i)$$

???



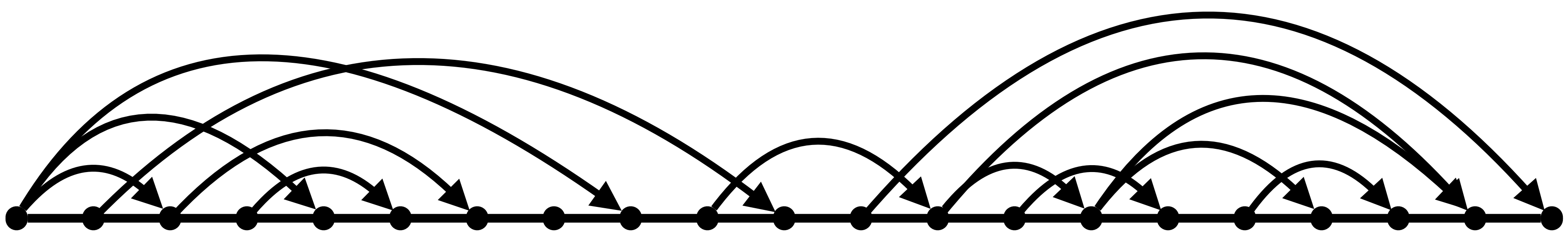
MSO-DEFINABLE GRAPH PROPERTIES

G = (V , → , ↷) IS AN M-STACKS N-CLOCKS TC-WORD

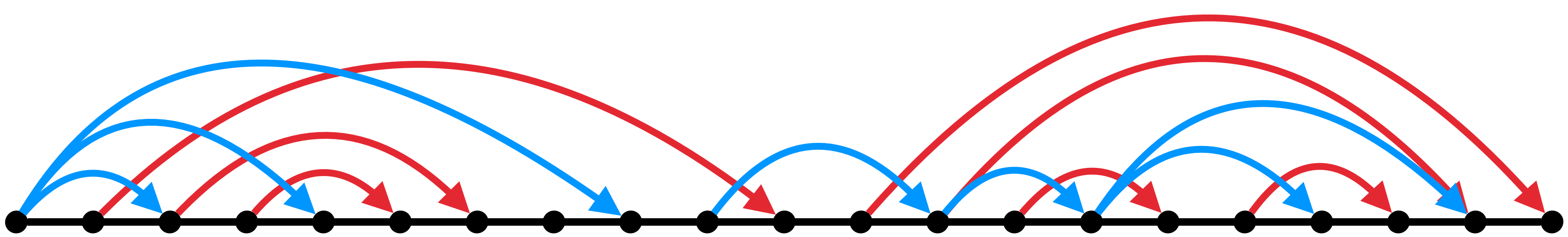
$$\mathsf{Forward}(\curvearrowright) \wedge \exists \overline{R} = (R_1, \ldots, R_n)\ \exists \overline{S} = (S_1, \ldots, S_m)$$

$$\mathsf{Partition}(\curvearrowright, \overline{R}, \overline{S}) \wedge \bigwedge_{i=1}^{n} \mathsf{Clock}(R_i) \wedge \bigwedge_{i=1}^{m} \mathsf{Stack}(S_i)$$

???



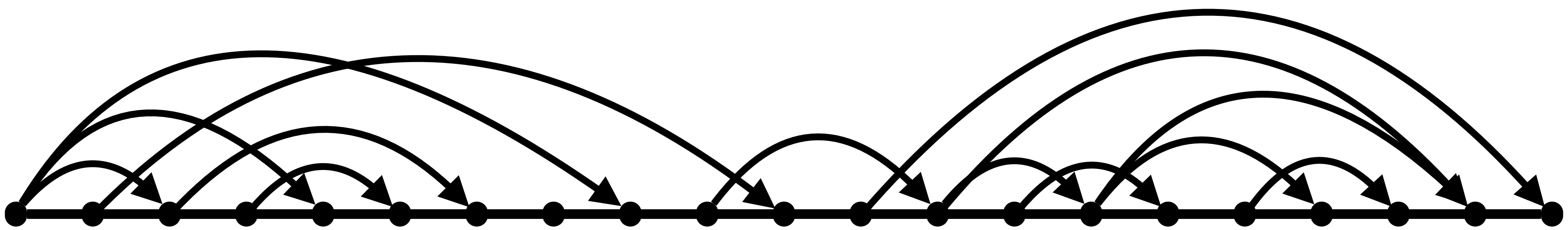
✓



MSO-DEFINABLE GRAPH PROPERTIES

G = (V , → , $\curvearrowright$) IS A TC-WORD ACCEPTED BY A TA $\mathcal{A}$

???



$$\exists \overline{X} = (X_\delta)_{\delta \in \Delta} \ \mathsf{Partition}(\overline{X}) \wedge \mathsf{AcceptingPath}(\overline{X})$$

$$\wedge \ \exists ({}^c\!\curvearrowright)_{c \in \mathsf{Clocks}} \ \mathsf{Partition}(\curvearrowright, ({}^c\!\curvearrowright)_{c \in \mathsf{Clocks}})$$

$$\wedge \bigwedge_{c \in \mathsf{Clocks}} \forall x, y \left(x \ {}^c\!\curvearrowright y \implies \mathsf{Reset}_c(x) \wedge \neg \exists (z \ x < z < y \wedge \mathsf{Reset}_c(z)) \right)$$

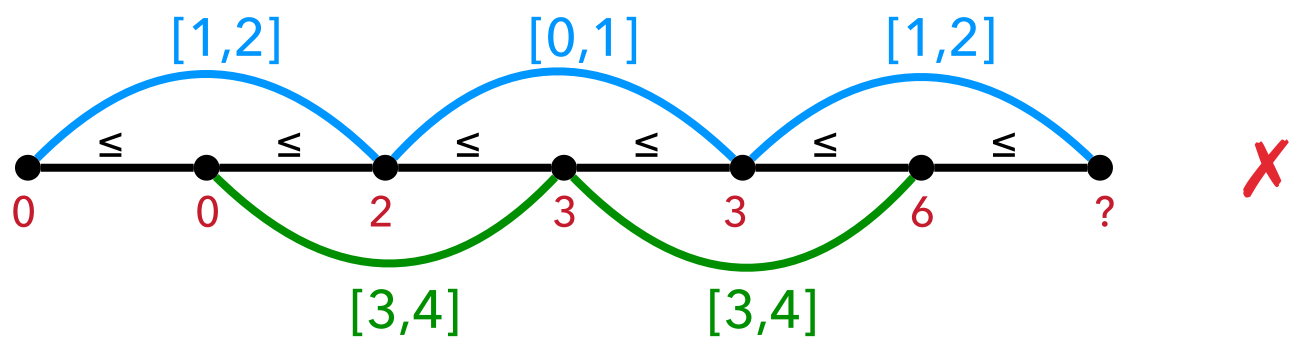
$$\wedge \bigwedge_{\substack{\delta \in \Delta \\ (c \in I) \in \delta}} \forall y \left(X_\delta(y) \implies \exists x \ (x \ {}^c\!\curvearrowright y \wedge x \curvearrowright^I y) \right)$$

$$\mathsf{Reset}_c(x) ::= \bigvee_{\substack{\delta \in \Delta \\ (c := 0) \in \delta}} X_\delta(x) \qquad\qquad \curvearrowright \ ::= \biguplus_I \curvearrowright^I$$

MSO-DEFINABLE GRAPH PROPERTIES

WITH VINCENT JUGÉ

THEOREM: REALIZABILITY OF TC-WORDS IS MSO-DEFINABLE

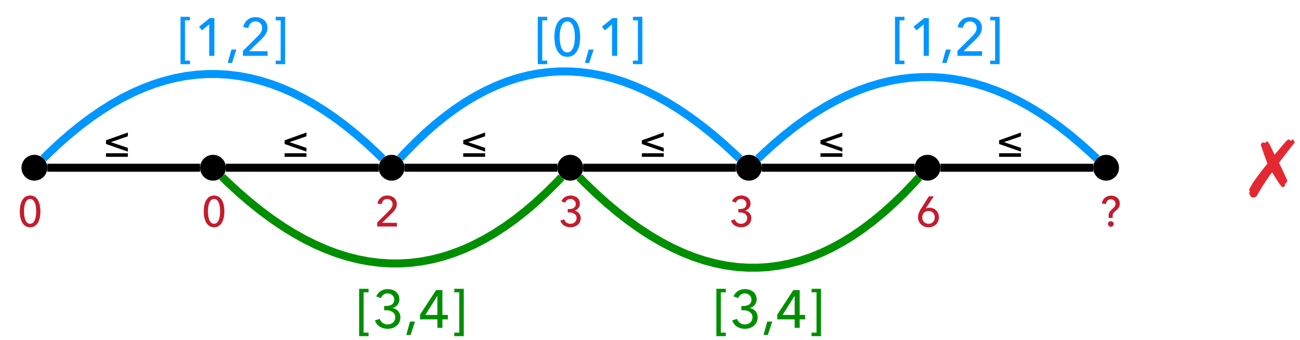


$$\exists \mathsf{ts}\colon V \to \mathbb{R},\ \forall x, y \ \ (x \curvearrowright^{I} y \implies \mathsf{ts}(y) - \mathsf{ts}(x) \in I) \wedge (x \to y \implies \mathsf{ts}(x) \leqslant \mathsf{ts}(y))$$

MSO-DEFINABLE GRAPH PROPERTIES

WITH VINCENT JUGÉ

THEOREM: REALIZABILITY OF TC-WORDS IS MSO-DEFINABLE

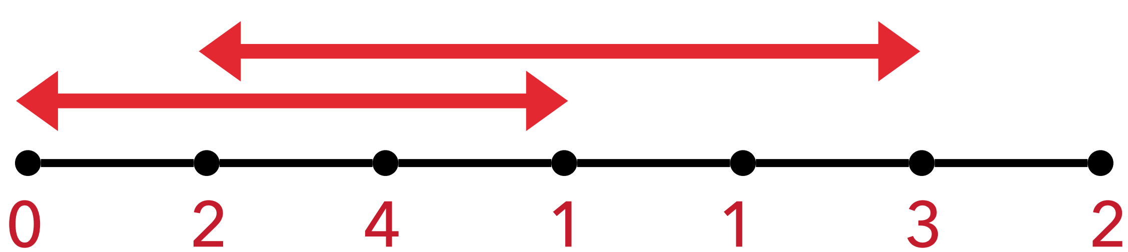


$$\exists \mathsf{ts}\colon V \to \overset{\mathbb{N}}{\cancel{\mathbb{R}}},\ \forall x,y\ \ (x \curvearrowright^{I} y \implies \mathsf{ts}(y)-\mathsf{ts}(x) \in I) \wedge (x \to y \implies \mathsf{ts}(x) \leqslant \mathsf{ts}(y))$$

$$\exists \mathsf{tsm}\colon V \to [M] = \{0,\dots,M-1\},\ \forall x,y\ \ x \curvearrowright^{I} y \implies$$

$$(\mathsf{Big}(x,y) \wedge I.\mathsf{up} = \infty) \vee (\neg \mathsf{Big}(x,y) \wedge (\mathsf{tsm}(y)-\mathsf{tsm}(x))[M] \in I)$$

M = 5

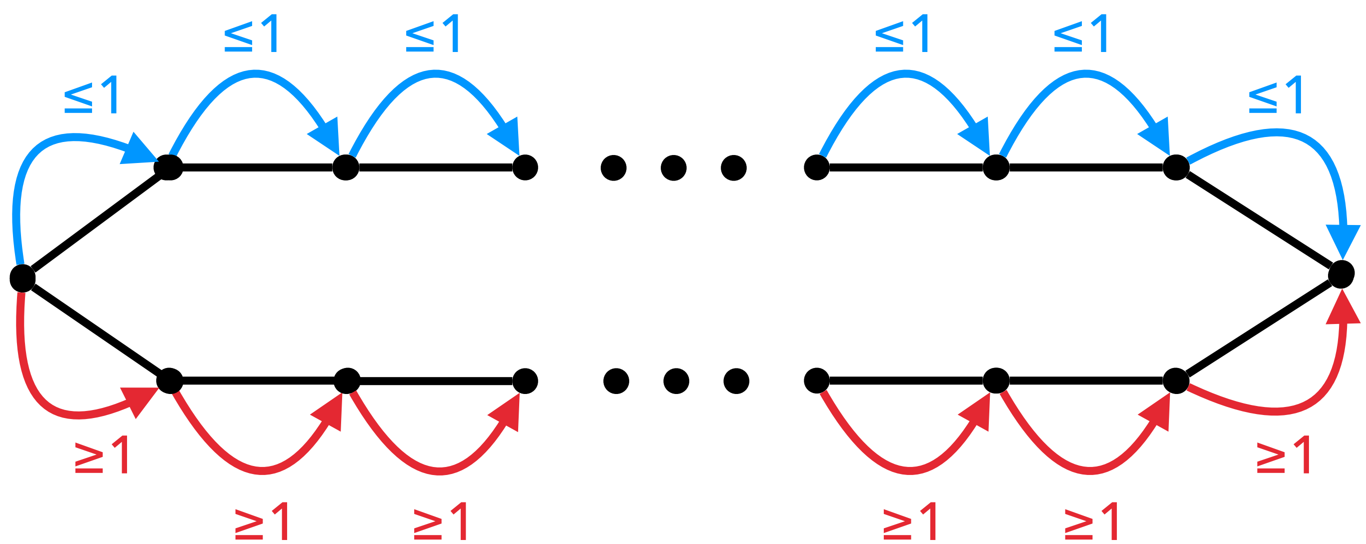


$$\mathsf{Big}(x,y) = \exists z,z',\ x < z < z' \leqslant y\ \wedge \bigvee_{\substack{a,b,c \mid \\ (b-a)[M]+(c-b)[M] \geqslant M}} \mathsf{tsm}(x) = a \wedge \mathsf{tsm}(z) = b \wedge \mathsf{tsm}(z') = c$$

MSO-DEFINABLE GRAPH PROPERTIES

G = (V , → , ↷) IS REALIZABLE

REALIZABILITY IS NOT MSO-DEFINABLE WITHOUT THE LINEAR ORDER



COURCELLE'S THEOREM

- Let TW_k be the set of graphs of tree-width at most k
- Let P be a property of graphs
- If P is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

WE WANT TO SOLVE $\mathscr{L}_{\mathsf{TCW}}(\mathcal{A}) \cap \mathfrak{Real}_{\mathsf{TCW}} \neq \emptyset$

- Show that TC-words have bounded tree-width — CONCUR'16 SPLIT-WIDTH
- Show that our properties are MSO-definable ✓
- Build directly tree automata for our properties — CONCUR'16

OUTLINE

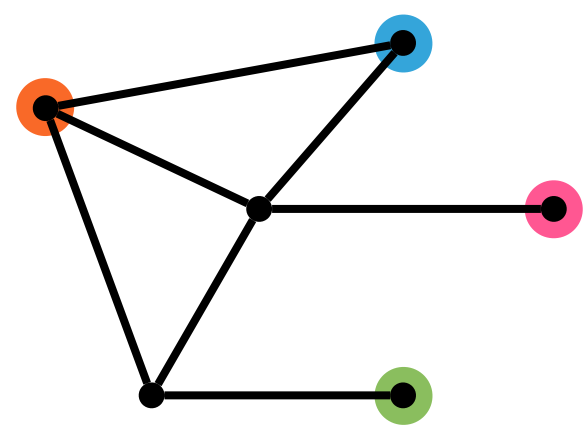
- BEHAVIOURS AS GRAPHS
- DECIDING PROPERTIES OF GRAPHS
- DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
- TREE-WIDTH FOR TIMED SYSTEMS
- INTERPRETING GRAPHS IN TREES
- CONCLUSION

TREE-WIDTH ALGEBRA

$$\tau ::= i \mid i \longrightarrow j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

ATOMIC

FORGET

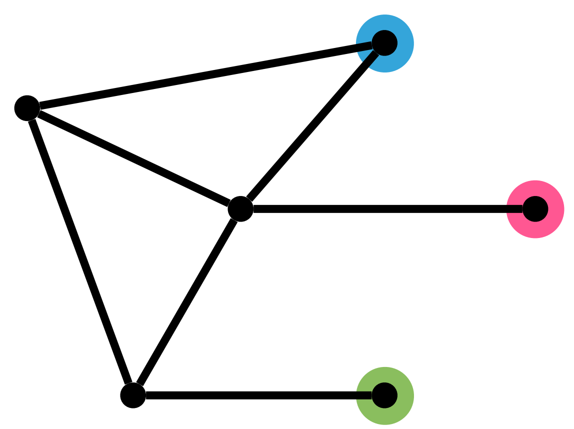


TREE-WIDTH ALGEBRA

$$\tau ::= i \mid i \longrightarrow j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

ATOMIC

FORGET

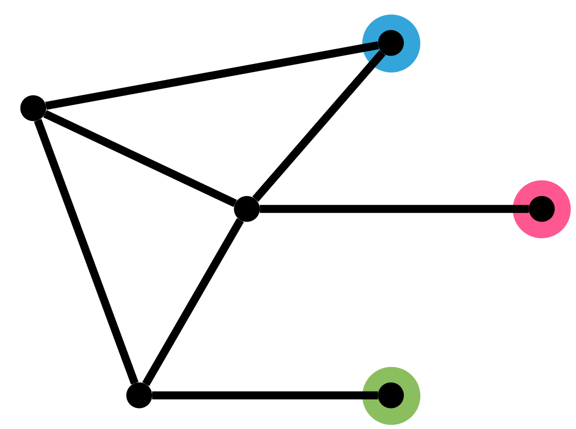


TREE-WIDTH ALGEBRA

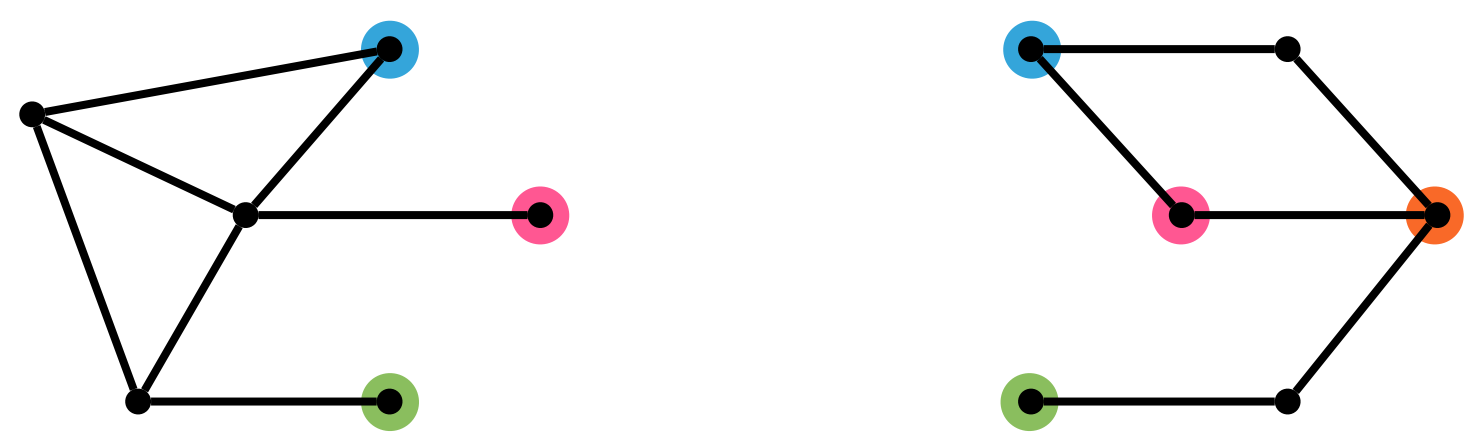
$$\tau ::= i \mid i \longrightarrow j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

ATOMIC

FORGET



COMBINE $\oplus$

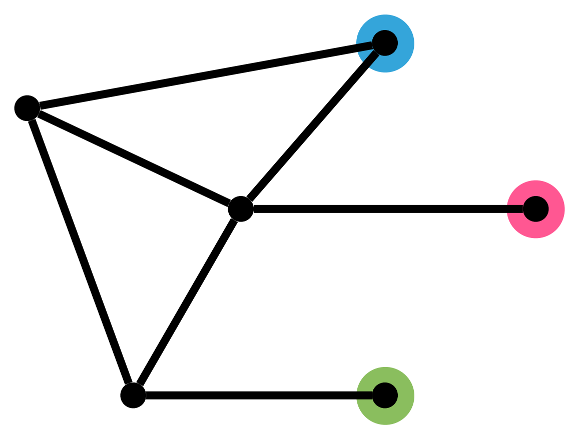


TREE-WIDTH ALGEBRA

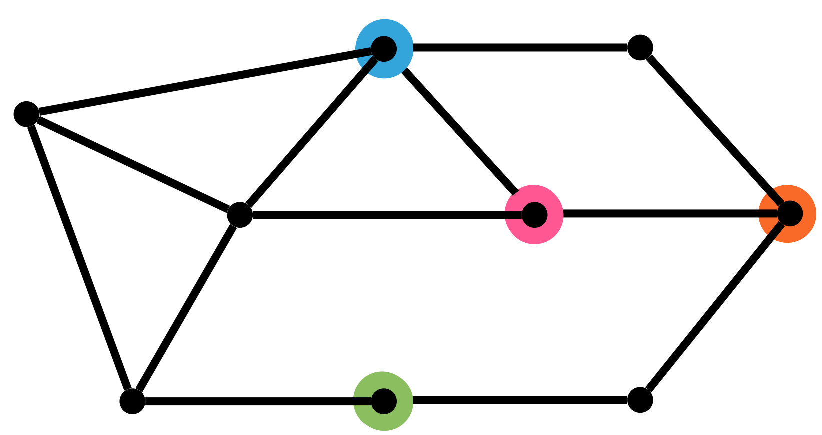
$$\tau ::= i \mid i \;\rule[0.5ex]{3em}{0.4ex}\; j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

ATOMIC

FORGET ●



COMBINE $\oplus$



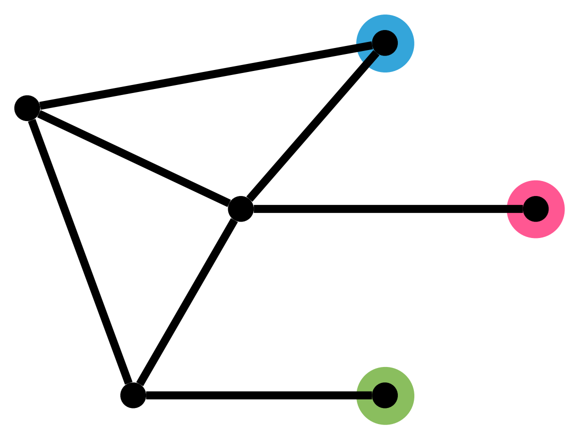
TREE-WIDTH ALGEBRA

TW_k : graphs of tree-width at most k

$$\tau ::= i \mid i \longrightarrow j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

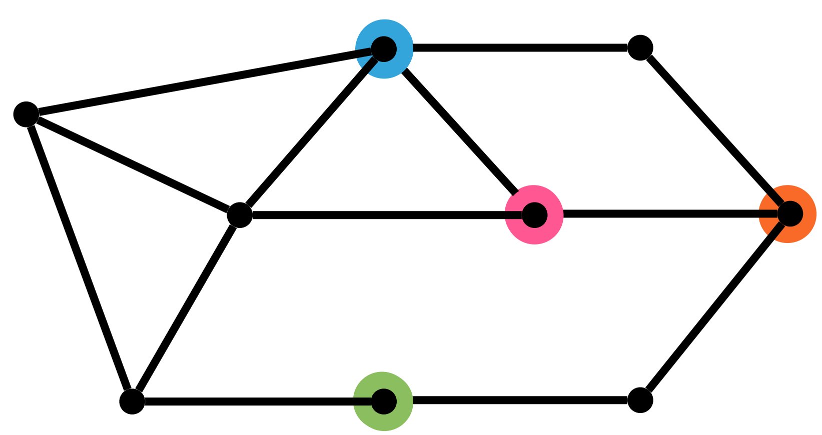
ATOMIC

FORGET ●



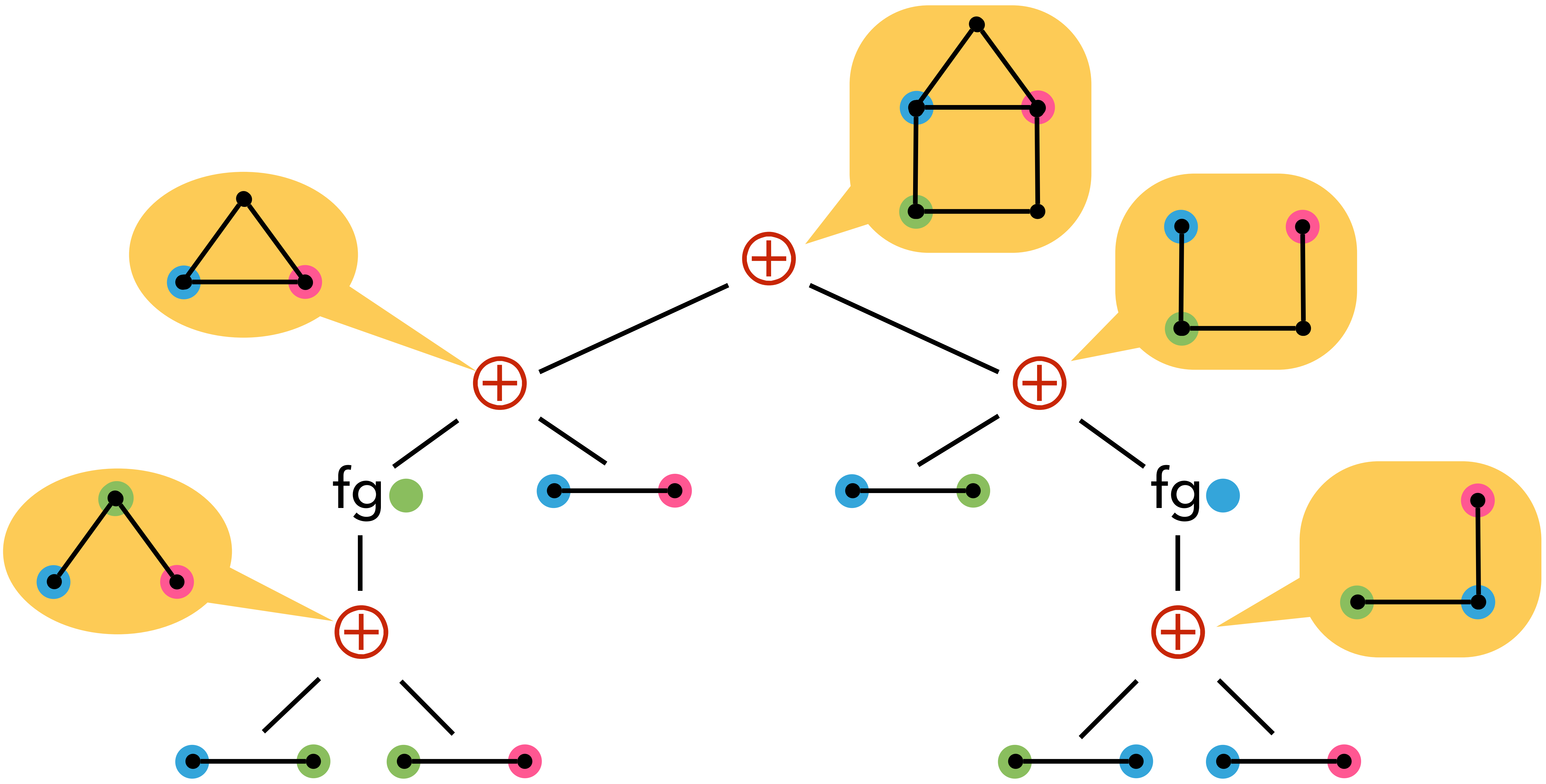
GRAPH G HAS TREE-WIDTH AT MOST k IF IT CAN BE CONSTRUCTED USING k+1 COLORS

COMBINE $\oplus$



TREE-WIDTH ALGEBRA

$$\tau ::= i \mid i \longrightarrow j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

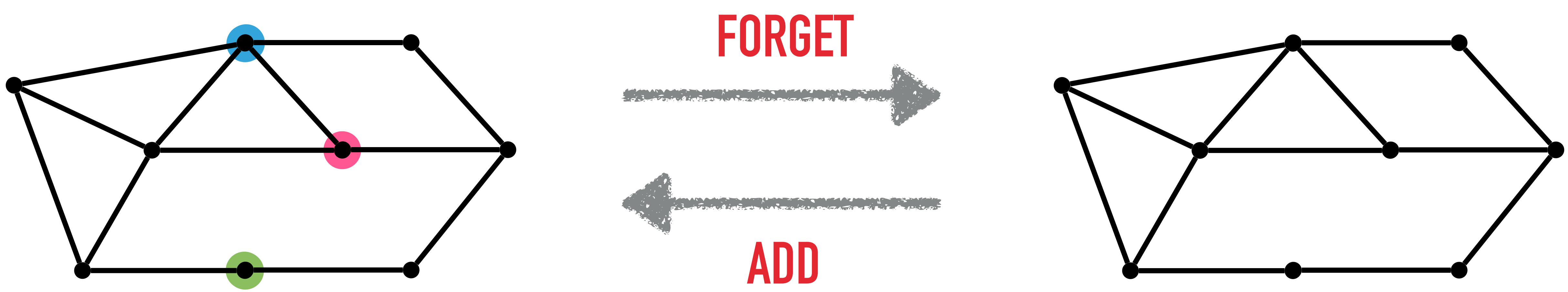


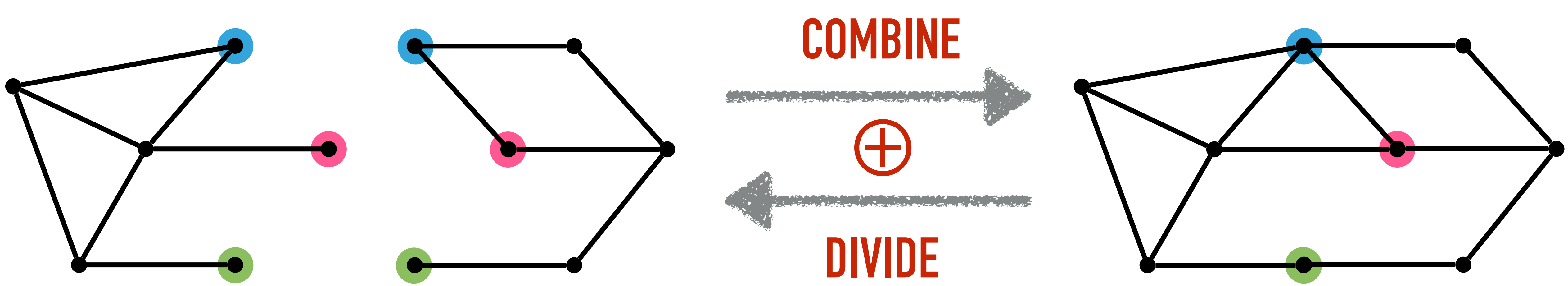
TREE-WIDTH ALGEBRA

GRAPH DECOMPOSITION

$$\tau ::= i \mid i \longrightarrow j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

ATOMIC

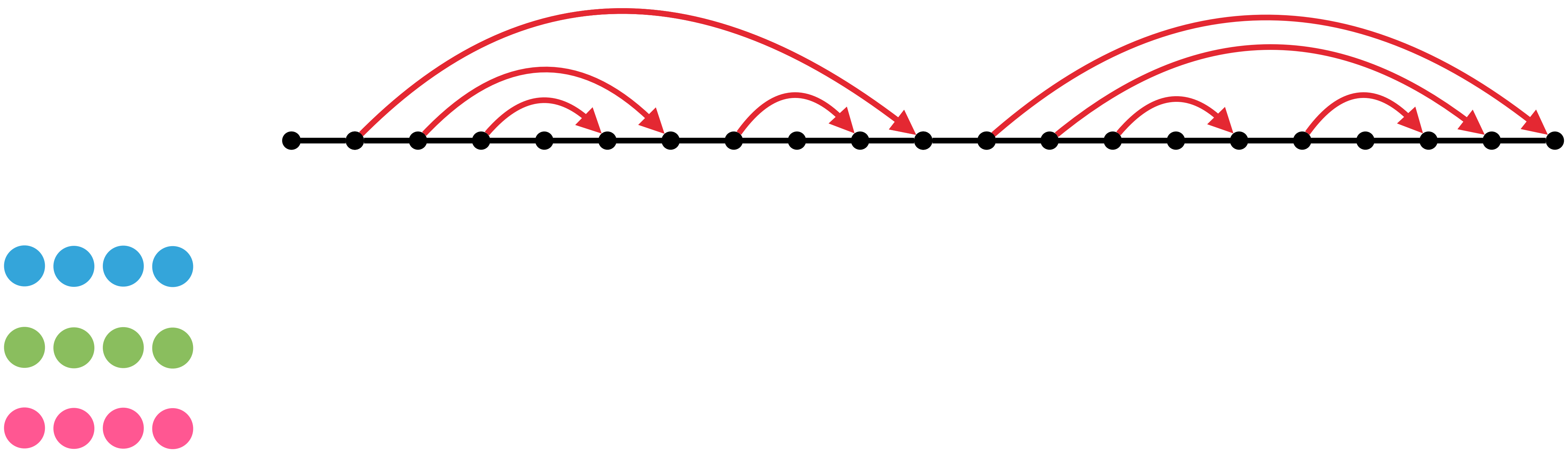




1-STACK TC-WORDS $\subseteq$ TW$_2$

G = (V , $\rightarrow$, $\curvearrowright$) TC-WORD

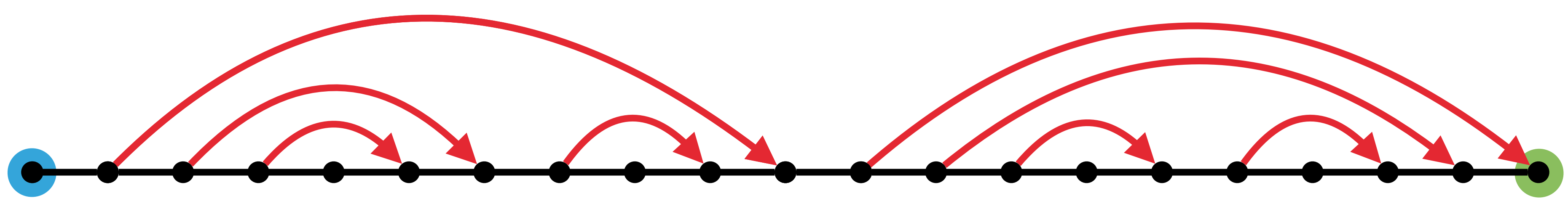
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK TC-WORDS $\subseteq$ TW_2

$G = (V, \rightarrow, \curvearrowright)$ TC-WORD

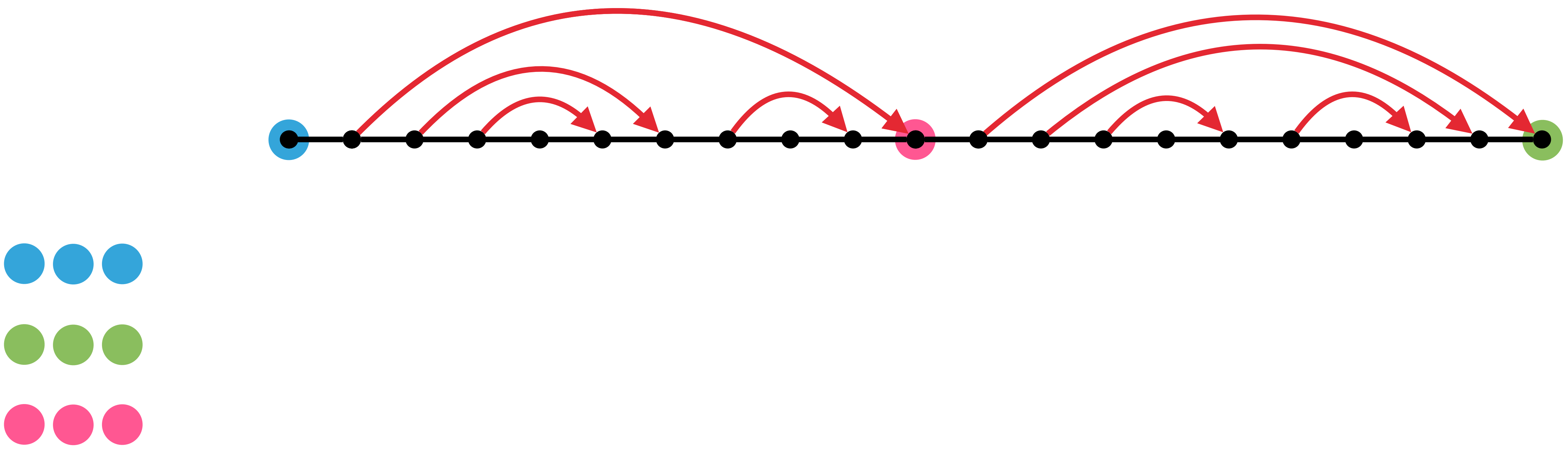
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK TC-WORDS $\subseteq$ TW$_2$

G = (V , → , ↷) TC-WORD

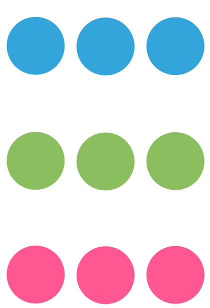
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK TC-WORDS $\subseteq$ TW_2

G = (V , → , ↷) TC-WORD

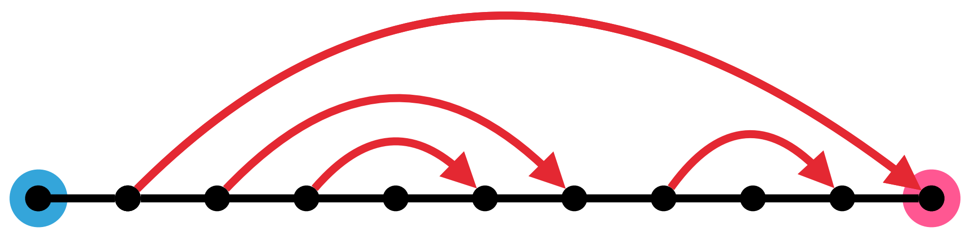
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

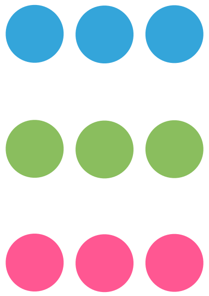


1-STACK TC-WORDS $\subseteq$ TW$_2$

G = (V , $\rightarrow$, $\curvearrowright$) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

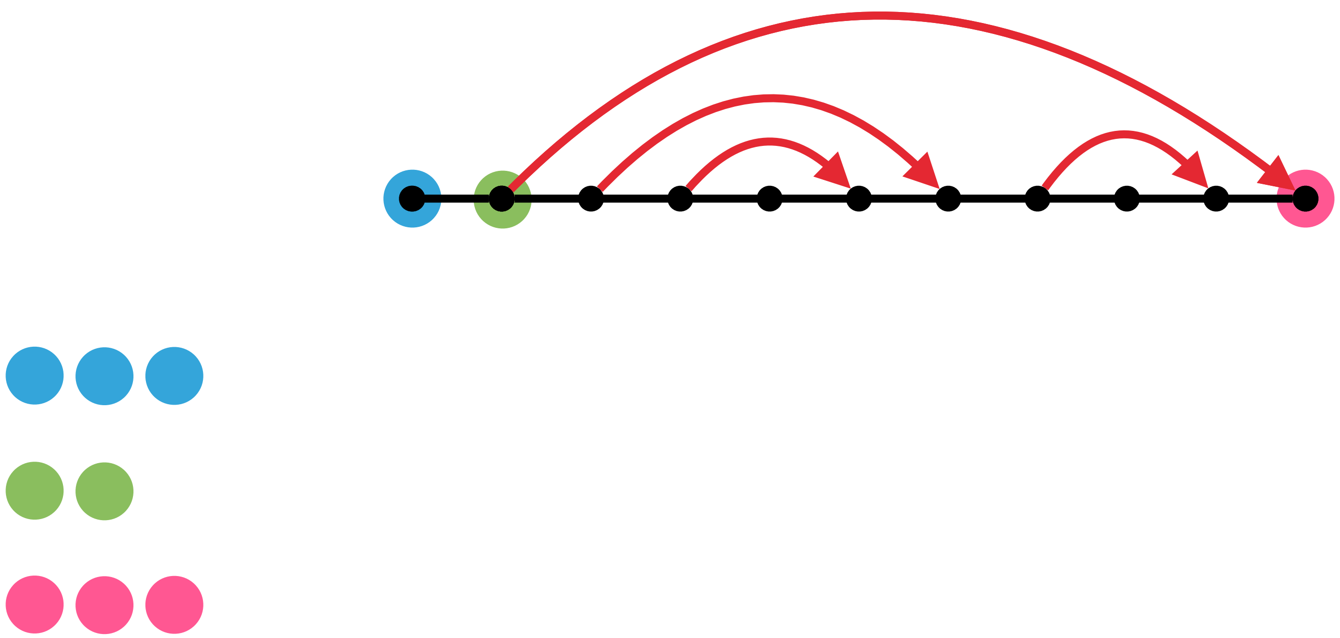




1-STACK TC-WORDS $\subseteq$ TW$_2$

G = (V , → , ↷) TC-WORD

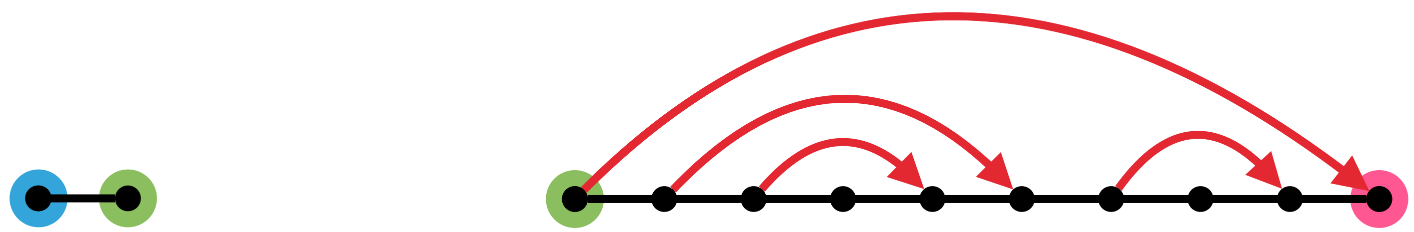
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

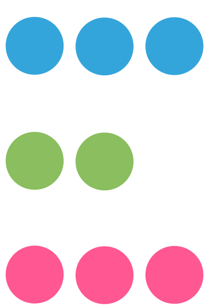


1-STACK TC-WORDS $\subseteq$ TW$_2$

$G = (V, \rightarrow, \curvearrowright)$ TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

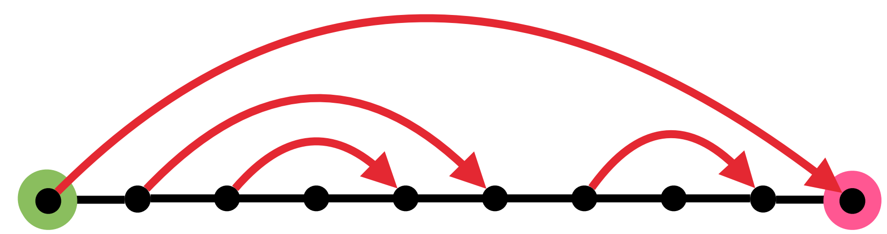


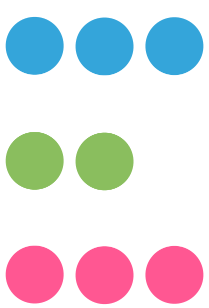


1-STACK TC-WORDS $\subseteq$ TW$_2$

$G = (V, \rightarrow, \curvearrowright)$ TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

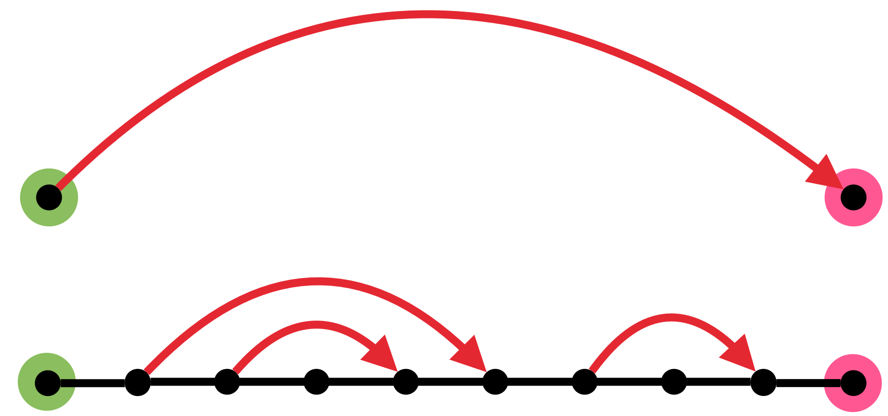




1-STACK TC-WORDS $\subseteq$ TW$_2$

G = (V , $\rightarrow$, $\curvearrowright$) TC-WORD

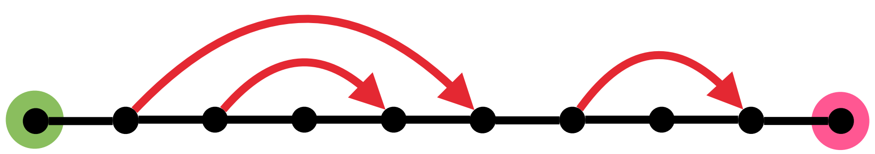
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK TC-WORDS $\subseteq$ TW$_2$

G = (V , → , ↷) TC-WORD

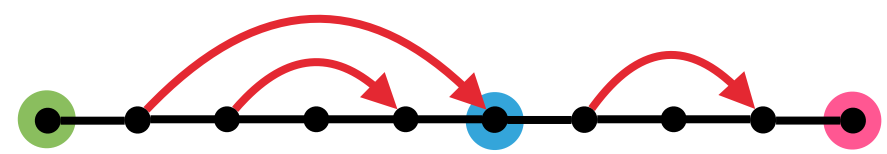
$$\tau ::= i \mid i \to j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

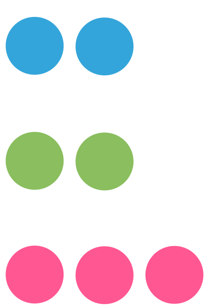


1-STACK TC-WORDS $\subseteq$ TW$_2$

$G = (V, \rightarrow, \curvearrowright)$ TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

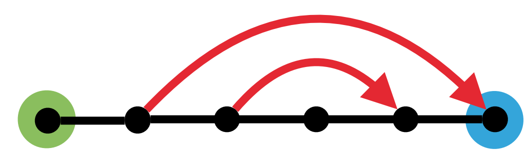


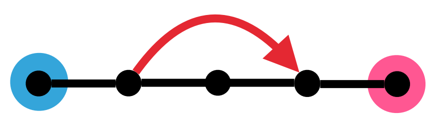


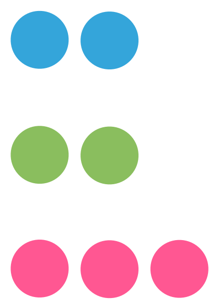
1-STACK TC-WORDS $\subseteq$ TW_2

G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$



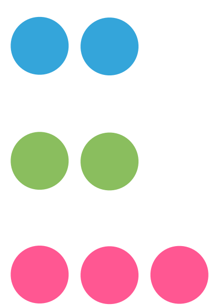




1-STACK TC-WORDS $\subseteq$ TW$_2$

G = (V , → , ↷) TC-WORD

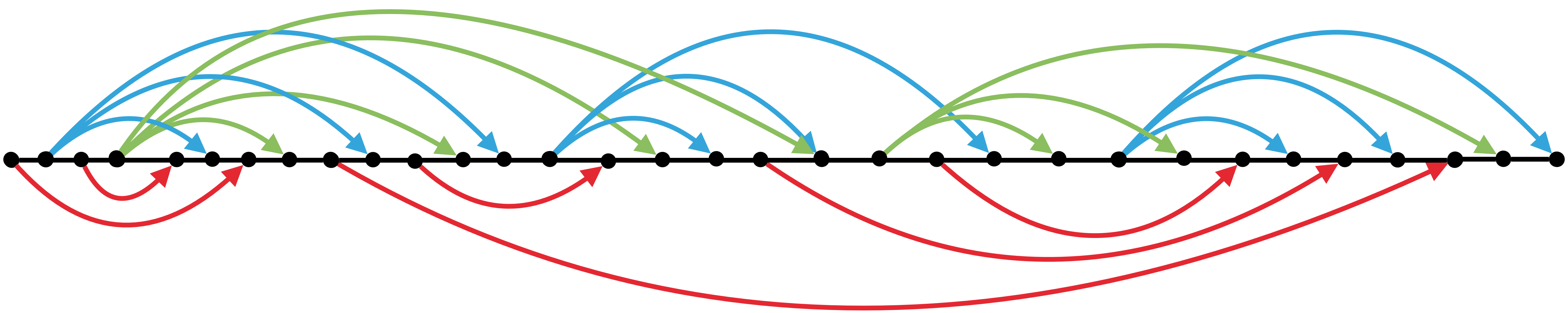
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

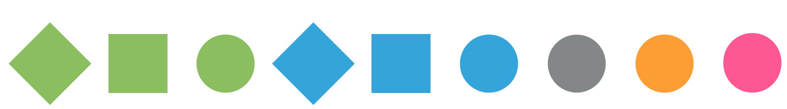


2-STACKS ⇒ UNBOUNDED TREE-WIDTH

1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$ G = (V , → , ↷) TC-WORD

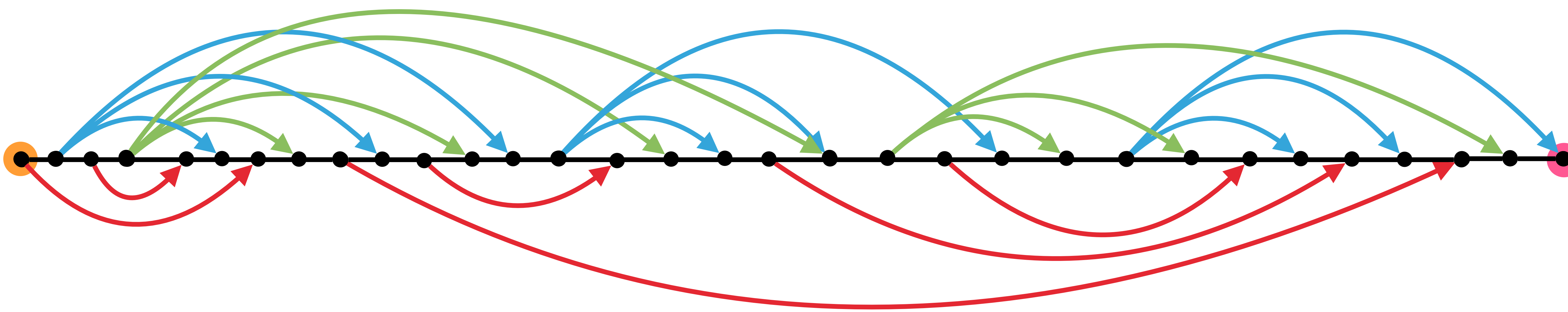
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

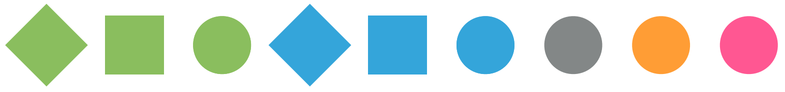




1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2} G = (V , → , ↷) TC-WORD

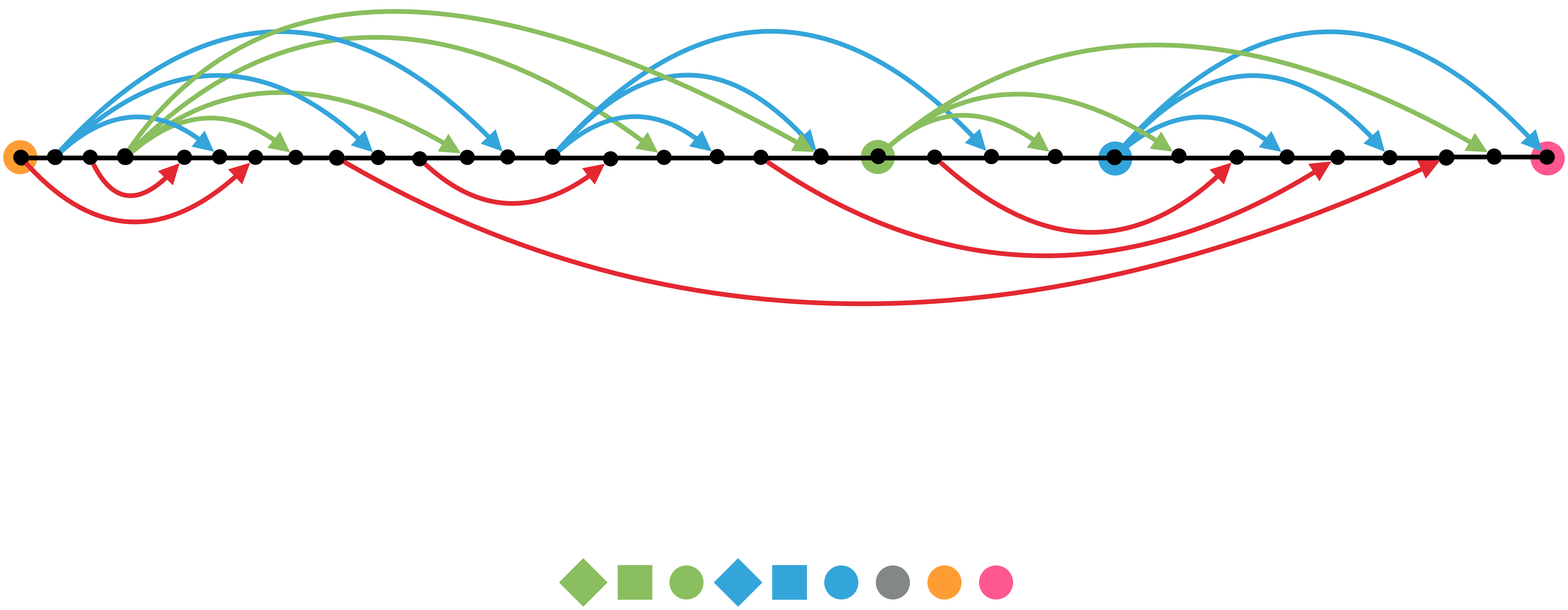
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$





1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$ G = (V , → , ↷) TC-WORD

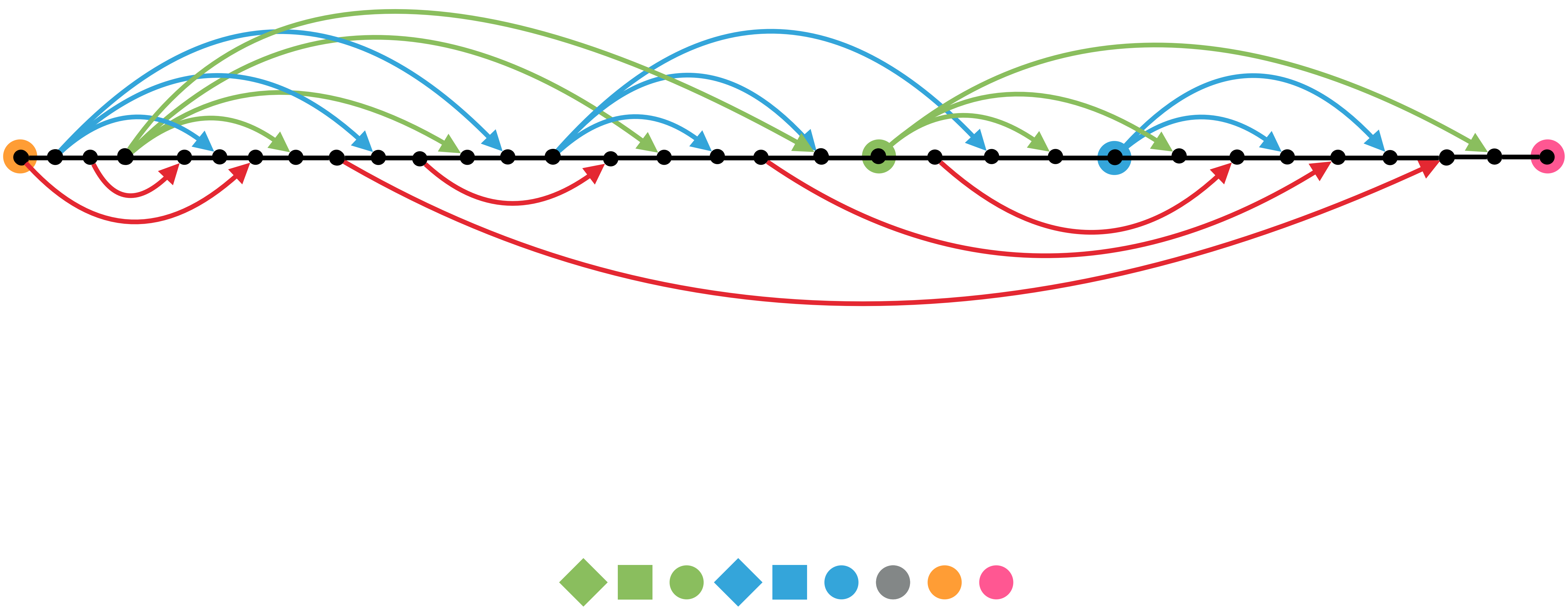
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$

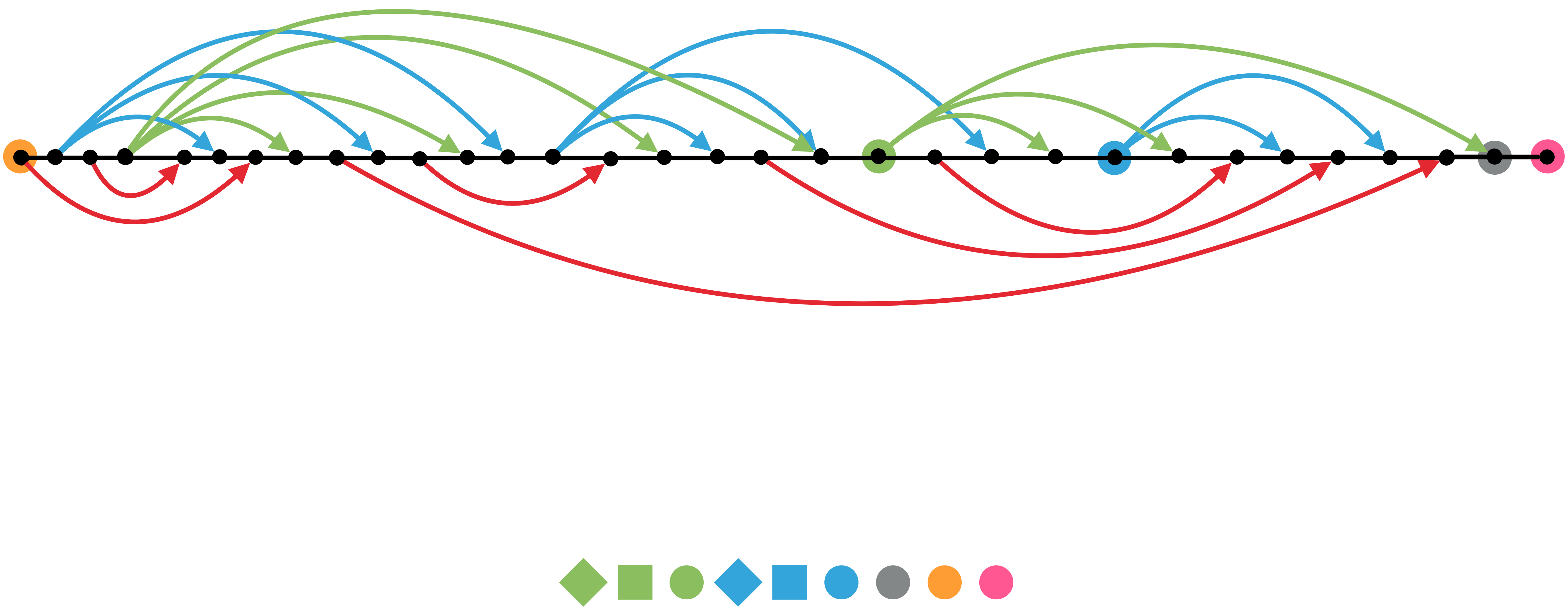
G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$ G = (V , $\rightarrow$, $\curvearrowright$) TC-WORD

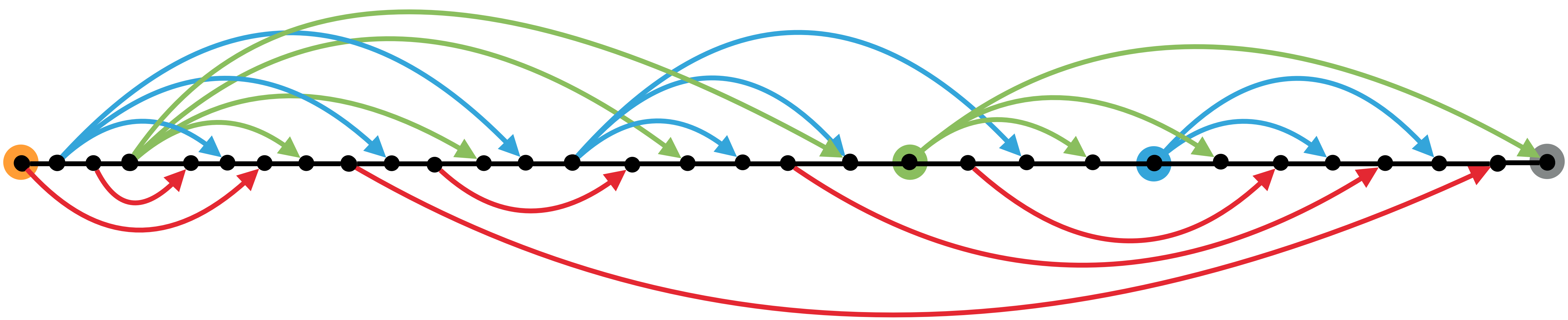
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

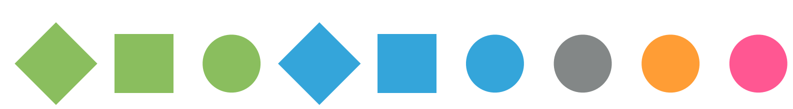


1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$

$G = (V, \rightarrow, \curvearrowright)$ TC-WORD

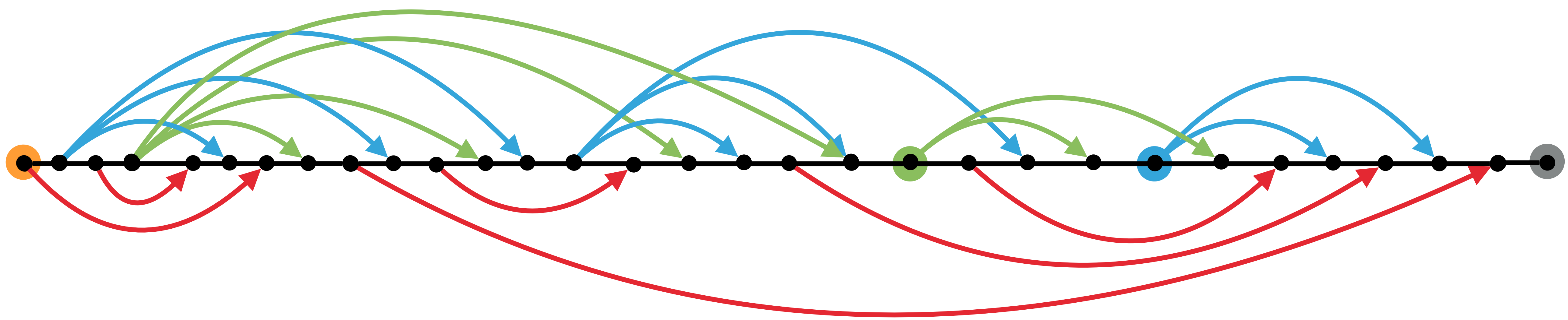
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$





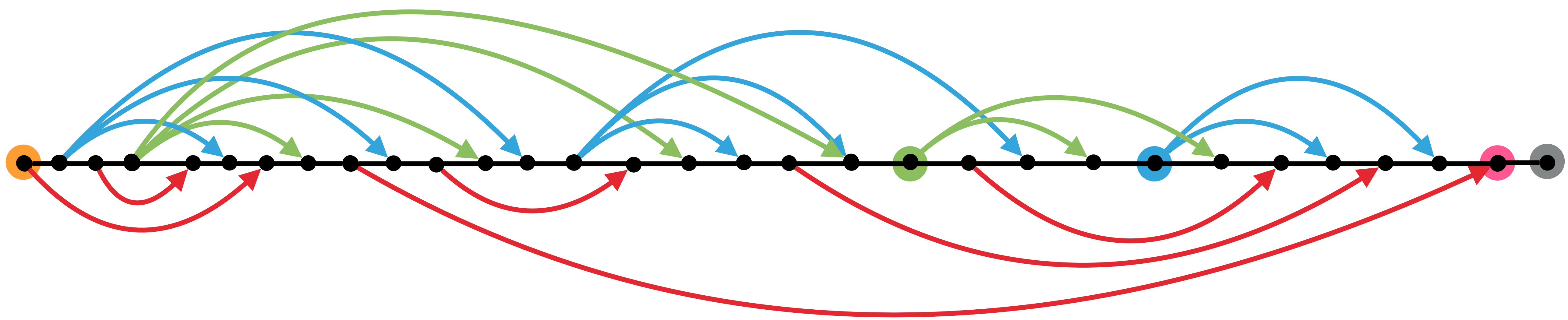
1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$ G = (V , → , ↷) TC-WORD

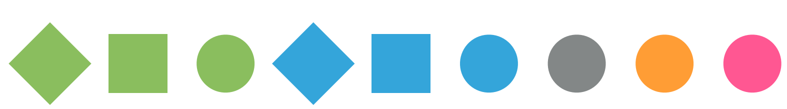
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2} G = (V , → , ↷) TC-WORD

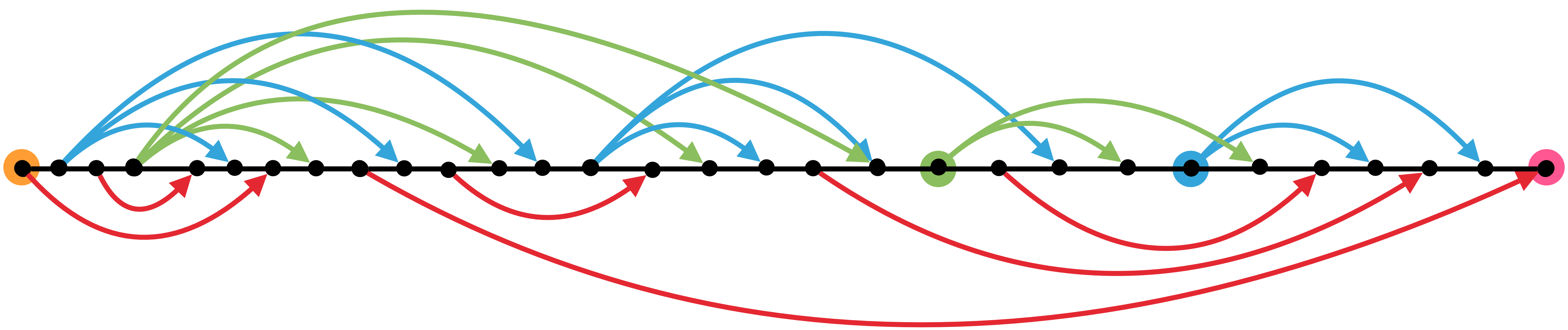
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

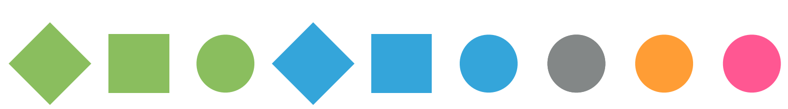




1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2} G = (V , → , ↷) TC-WORD

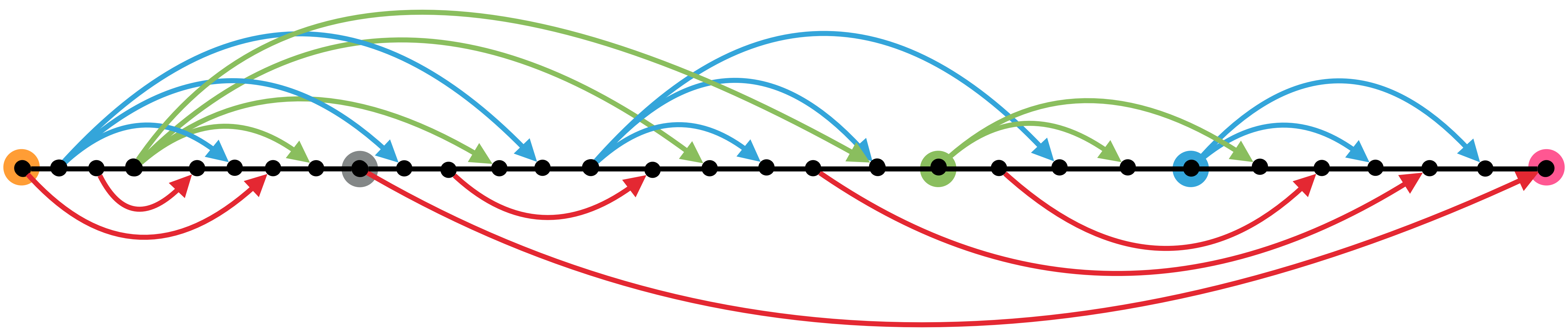
$$\tau ::= i \mid i \to j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

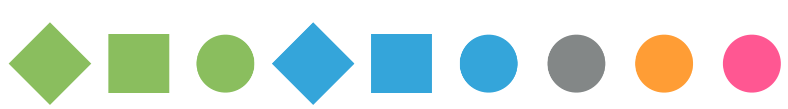




1-STACK k-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$ G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

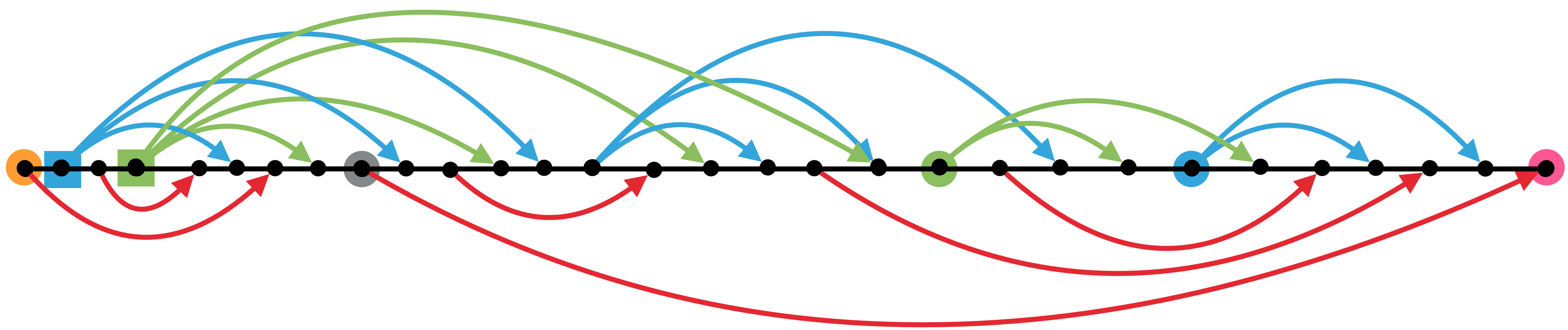


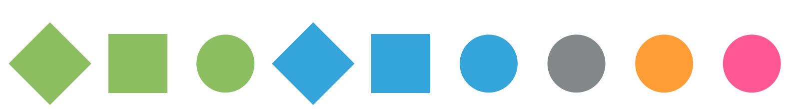


1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2}

G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

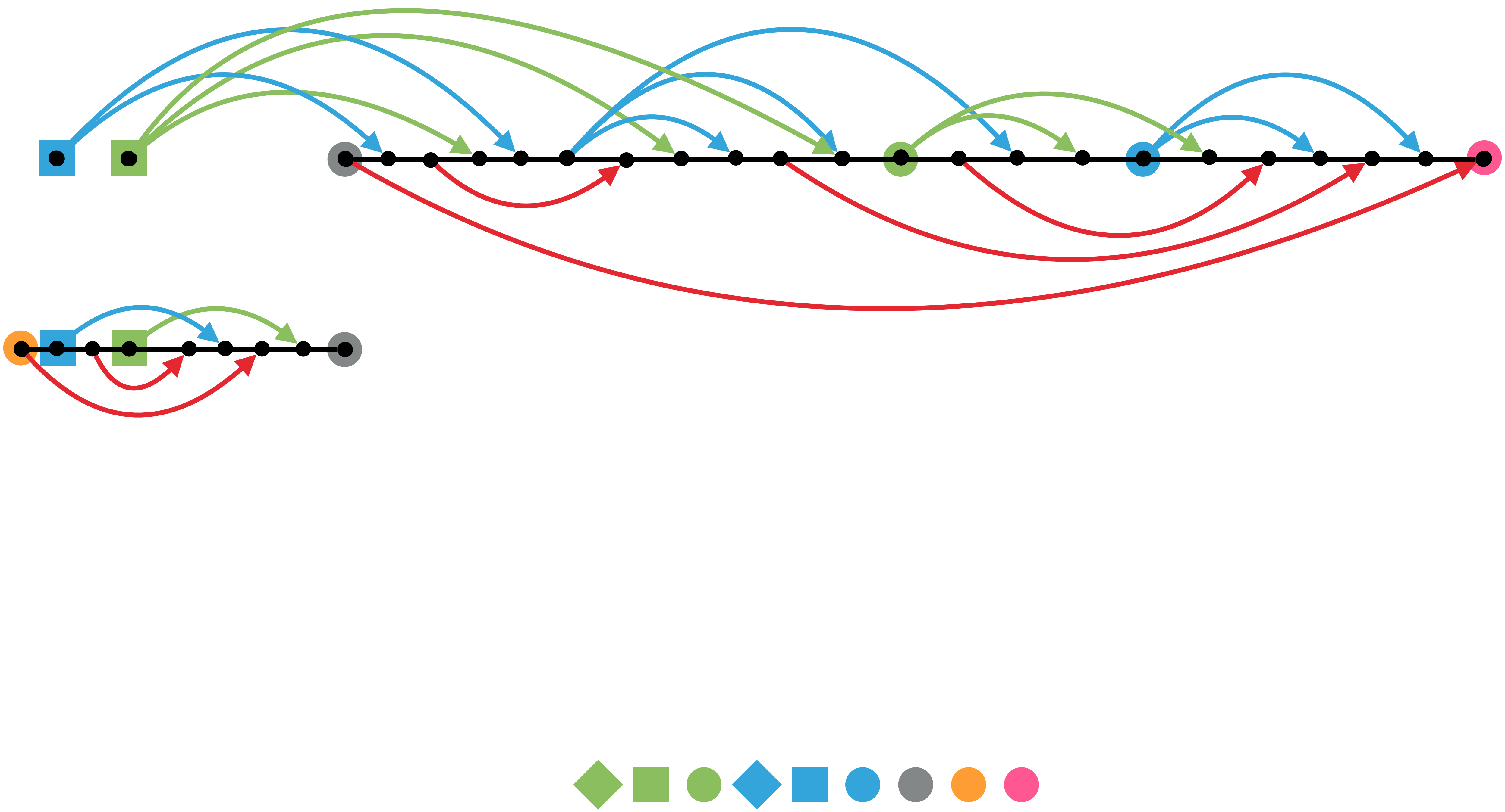




1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$

G = (V , → , ↷) TC-WORD

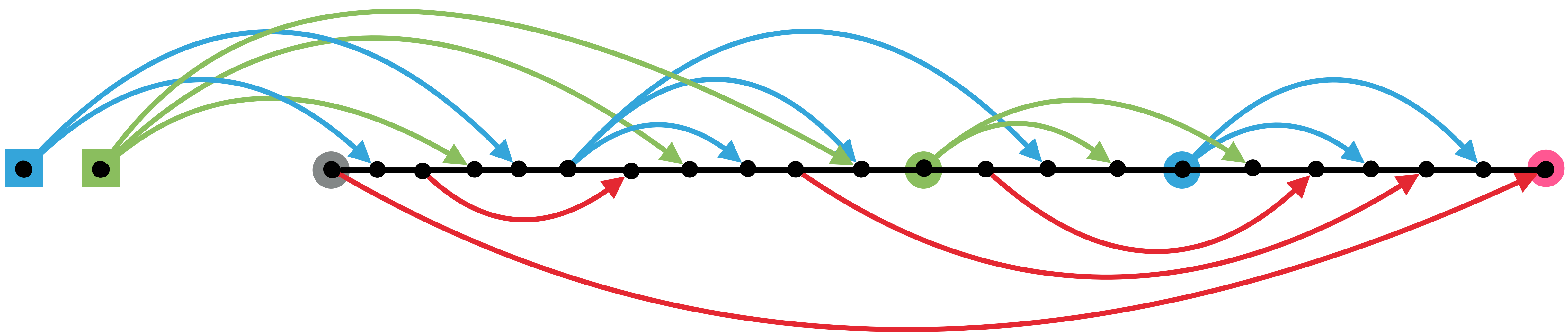
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2}

G = (V , → , ↷) TC-WORD

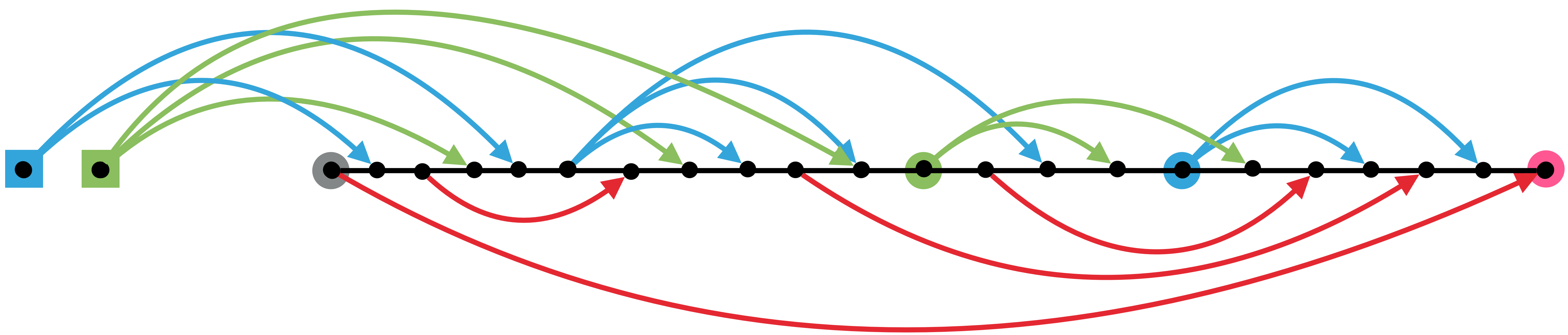
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK k-CLOCKS TC-WORDS $\subseteq TW_{3k+2}$

$G = (V, \rightarrow, \curvearrowright)$ TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

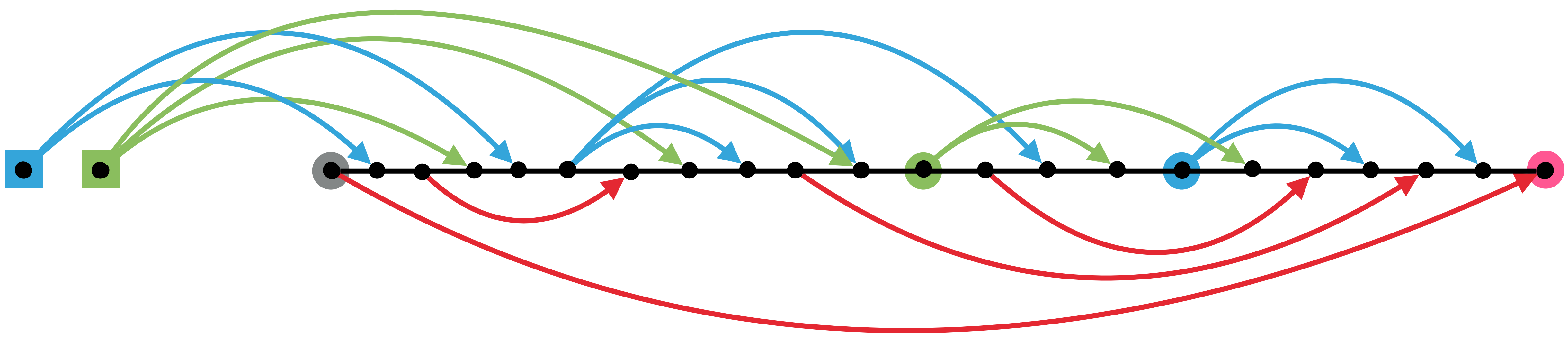


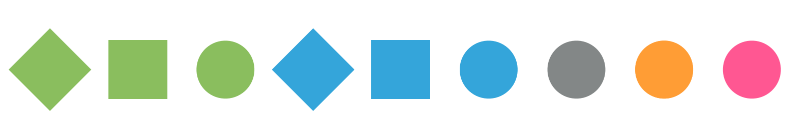
- At most one hanging reset node for each clock
- At most one Last reset node for each clock
- First and last points
- k+1 extra colors to maintain this invariant

1-STACK k-CLOCKS TC-WORDS ⊆ TW_{3k+2}

G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

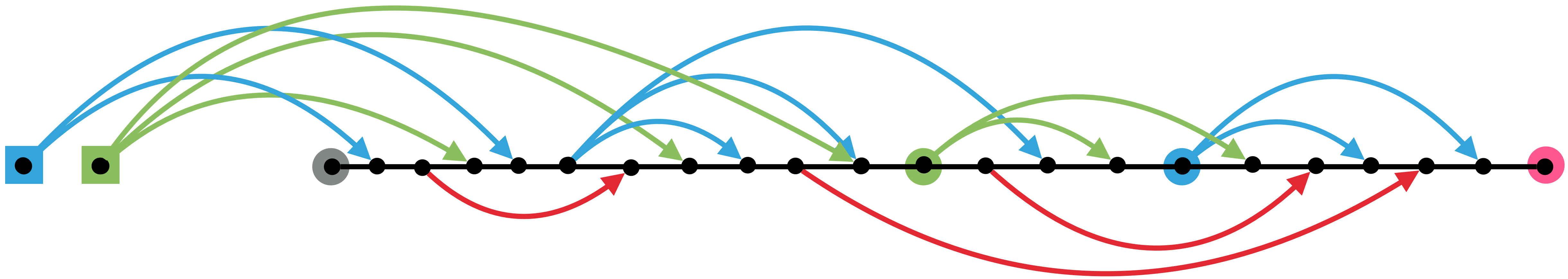


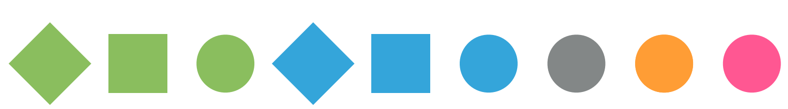


1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2}

G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

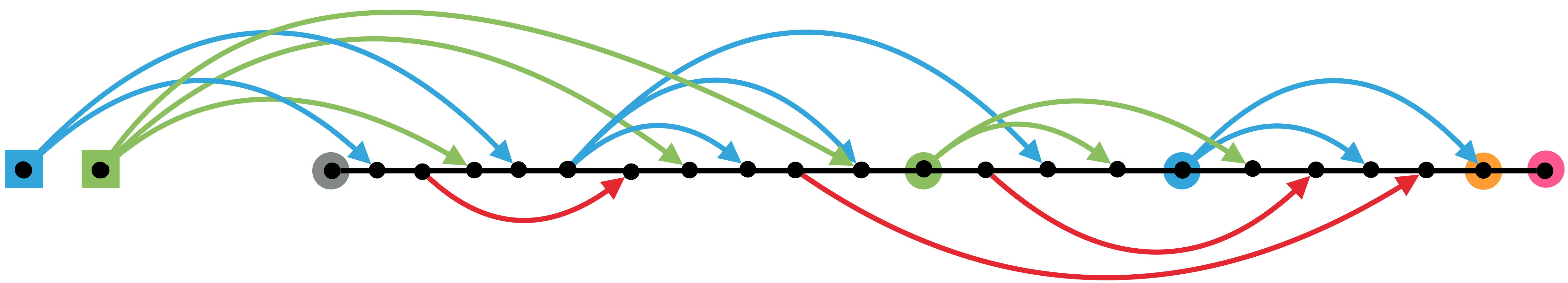


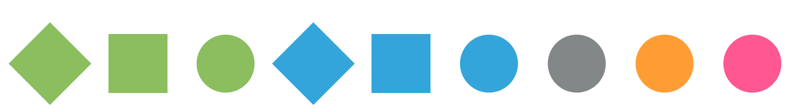


1-STACK k-CLOCKS TC-WORDS ⊆ TW_{3k+2}

G = (V , → , ↷) TC-WORD

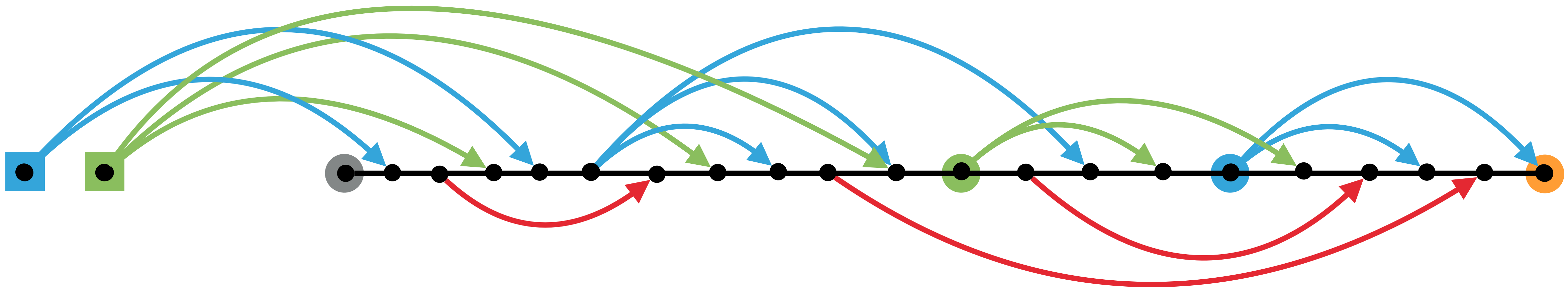
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

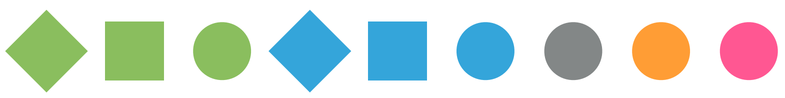




1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2} G = (V , → , ↷) TC-WORD

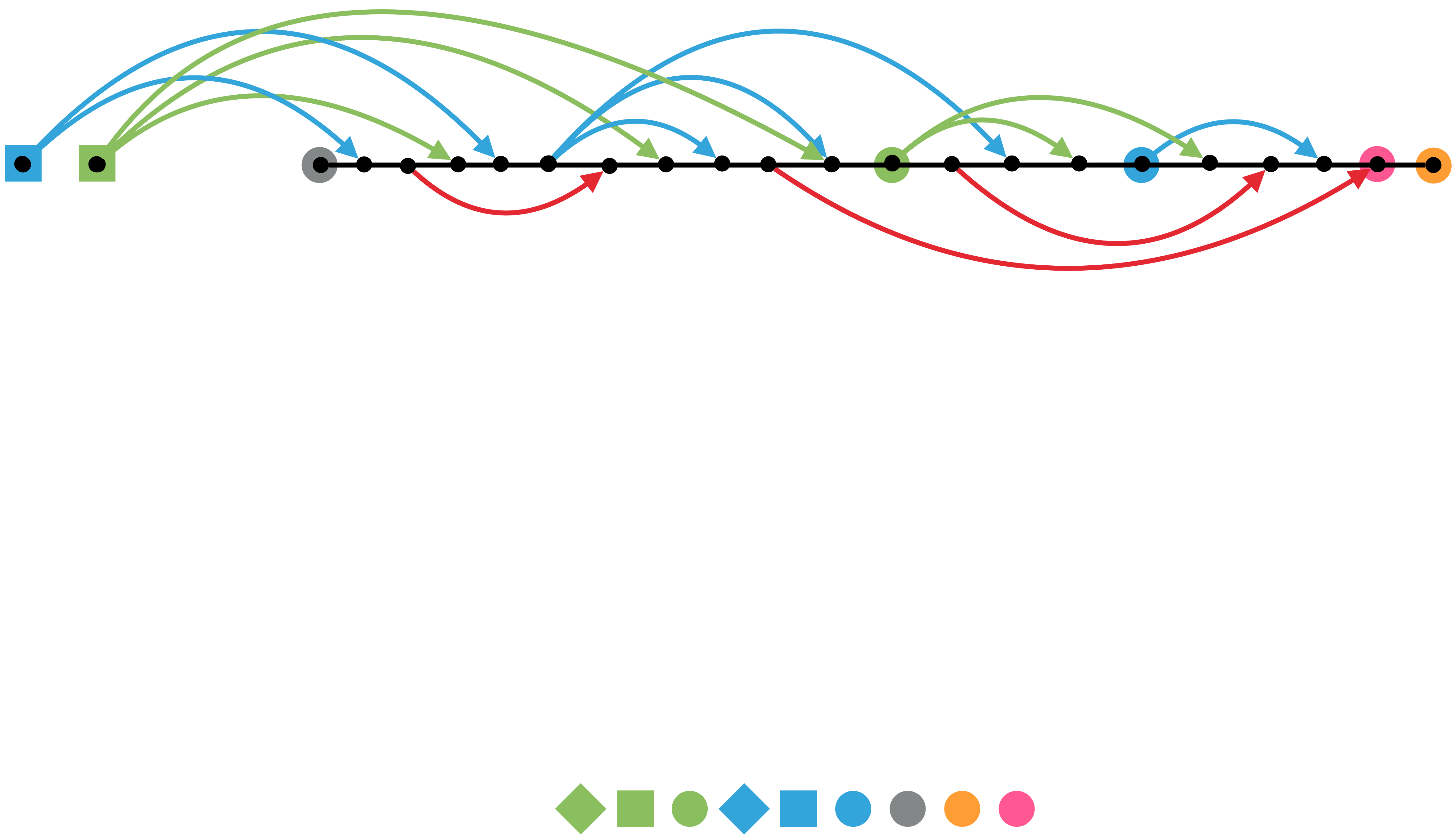
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$





1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2} G = (V , → , ↷) TC-WORD

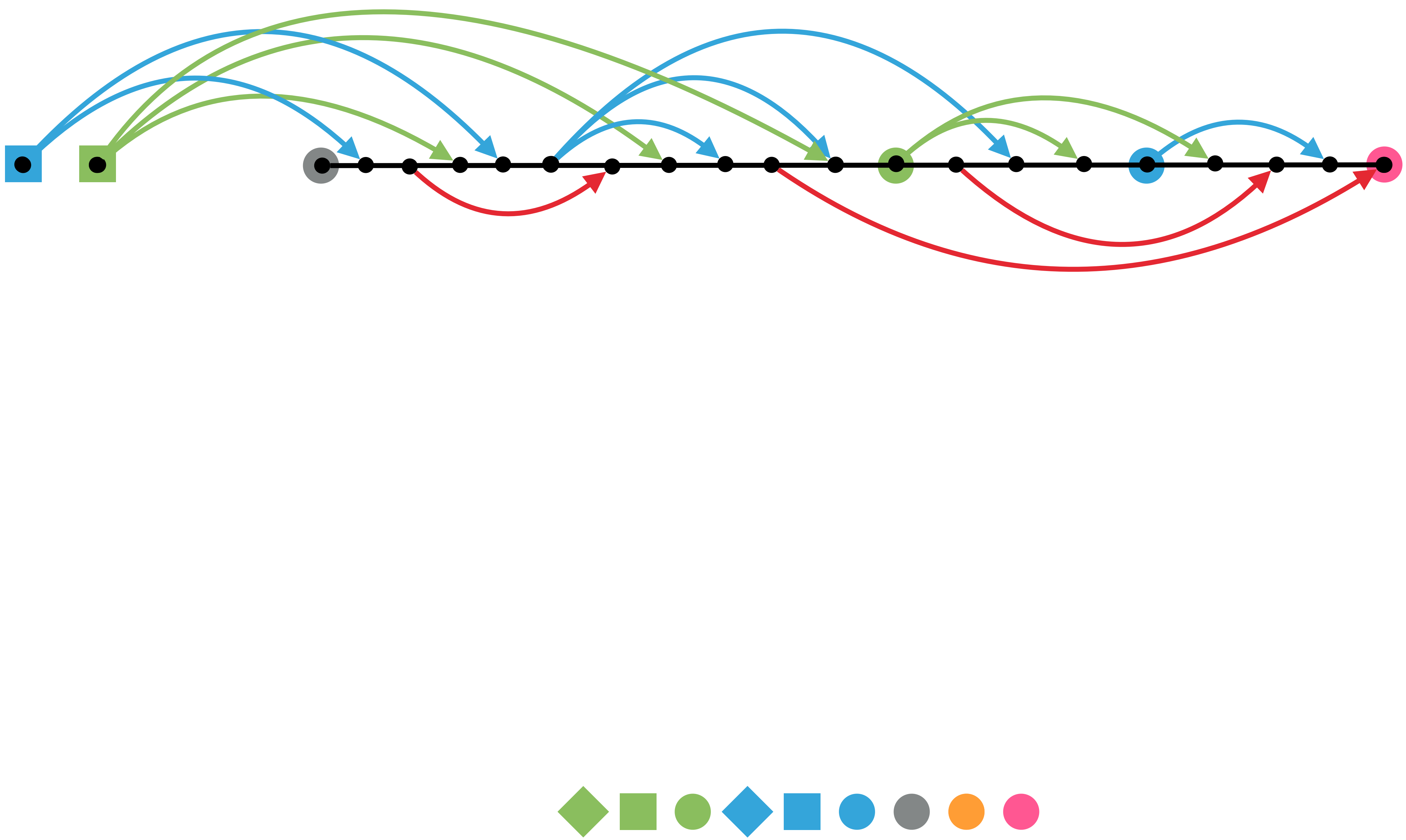
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$

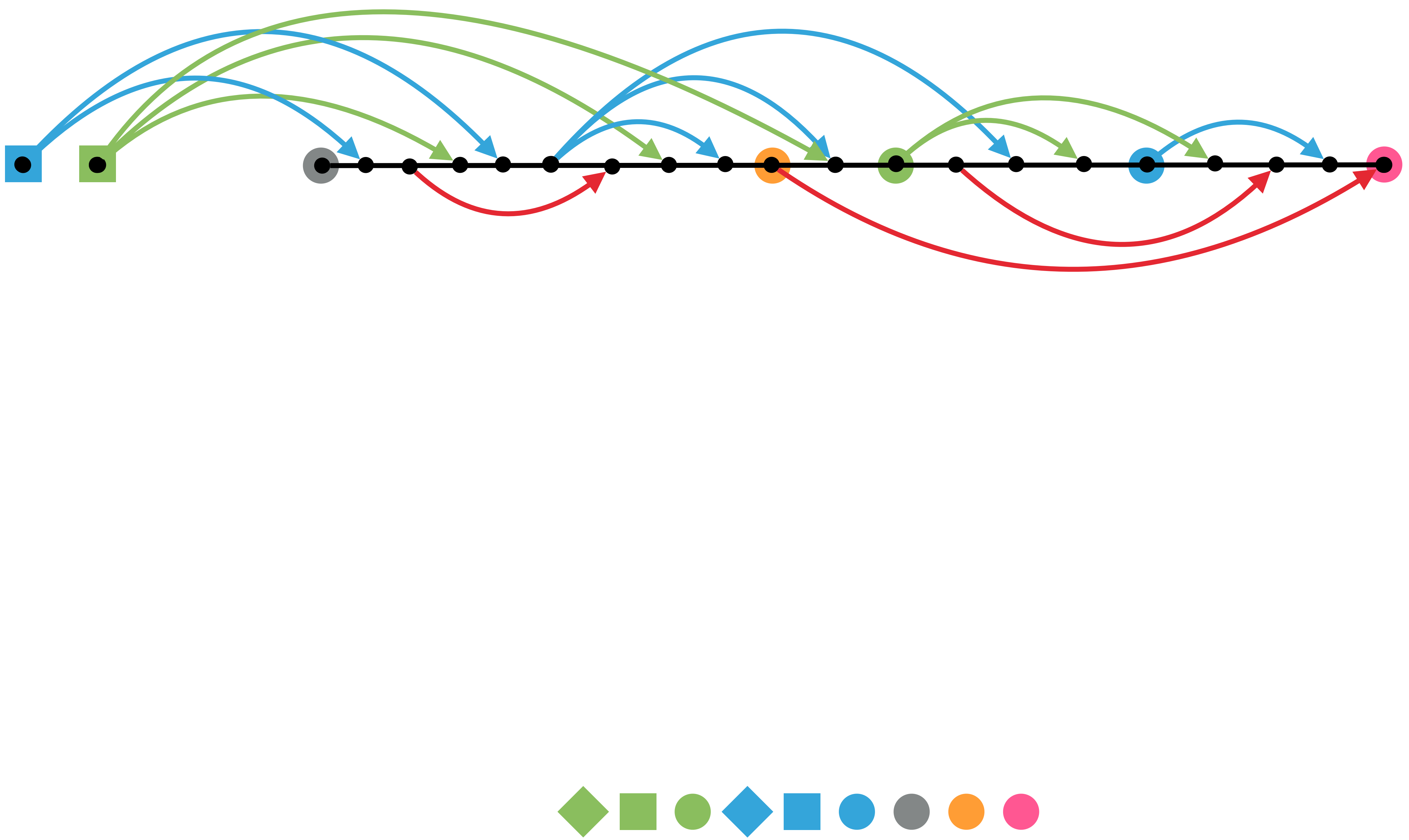
G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



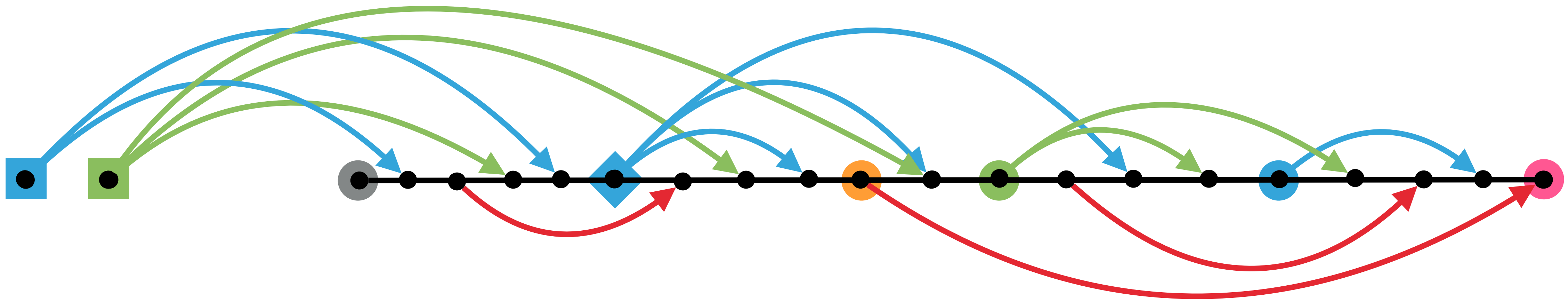
1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$ G = (V , → , ↷) TC-WORD

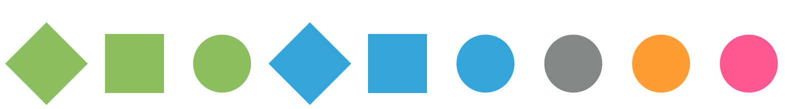
$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$



1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2} G = (V , → , ↷) TC-WORD

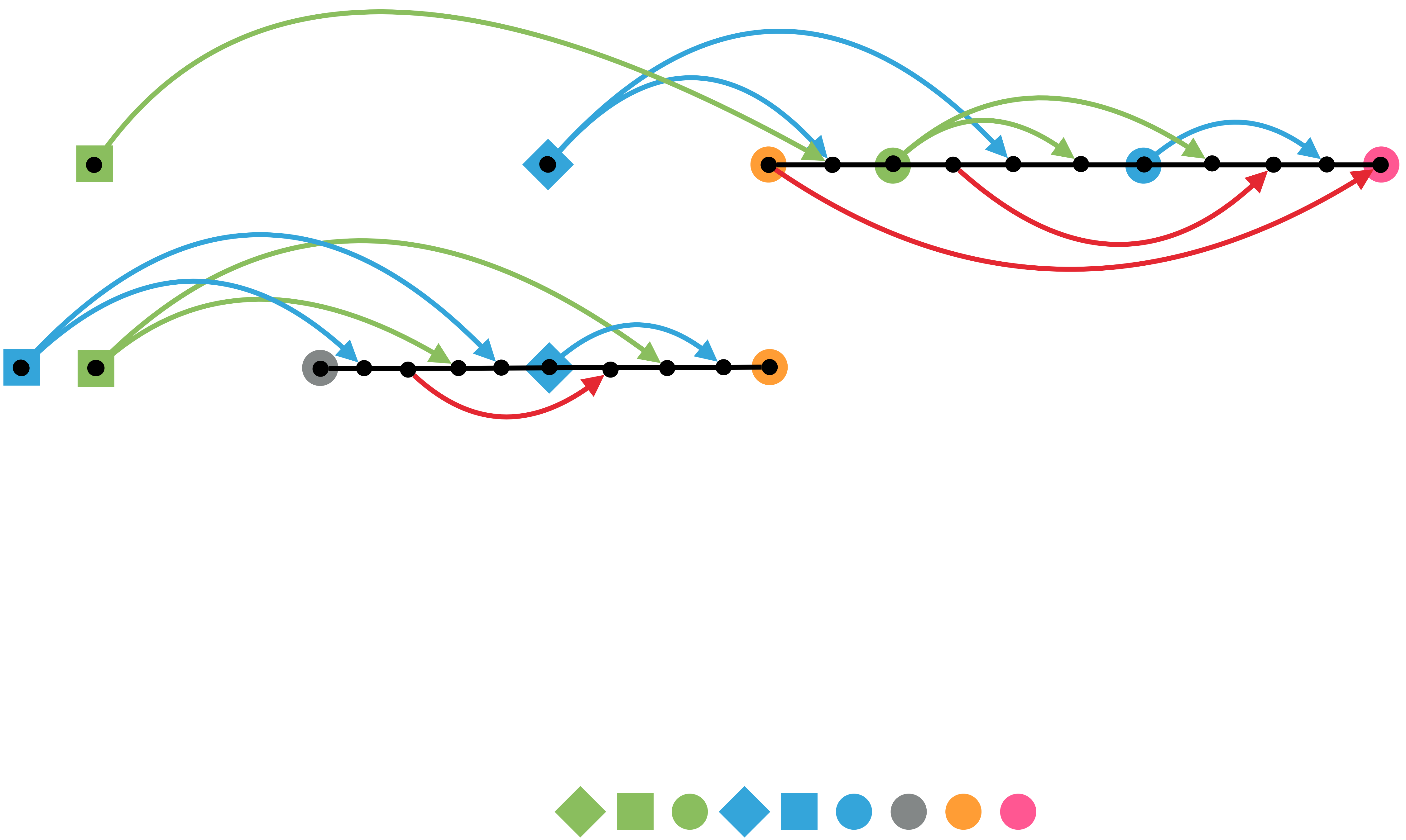
$$\tau ::= i \mid i \to j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$





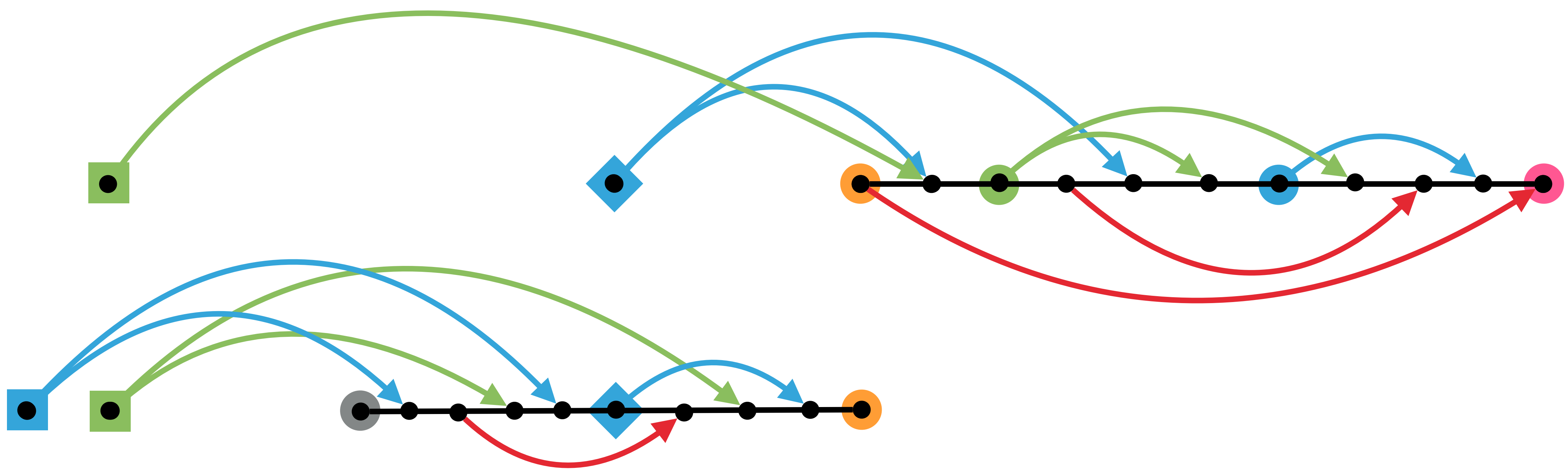
1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW$_{3k+2}$ G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

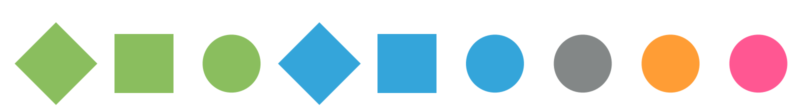


1-STACK k-CLOCKS TC-WORDS $\subseteq$ TW_{3k+2} G = (V , → , ↷) TC-WORD

$$\tau ::= i \mid i \rightarrow j \mid i \curvearrowright j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$



- At most one hanging reset node for each clock
- At most one Last reset node for each clock
- First and last points
- k+1 extra colors to maintain this invariant



COURCELLE'S THEOREM

- Let TW_k be the set of graphs of tree-width at most k
- Let P be a property of graphs
- If P is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

WE WANT TO SOLVE $\mathscr{L}_{\mathsf{TCW}}(\mathcal{A}) \cap \mathfrak{Real}_{\mathsf{TCW}} \neq \emptyset$

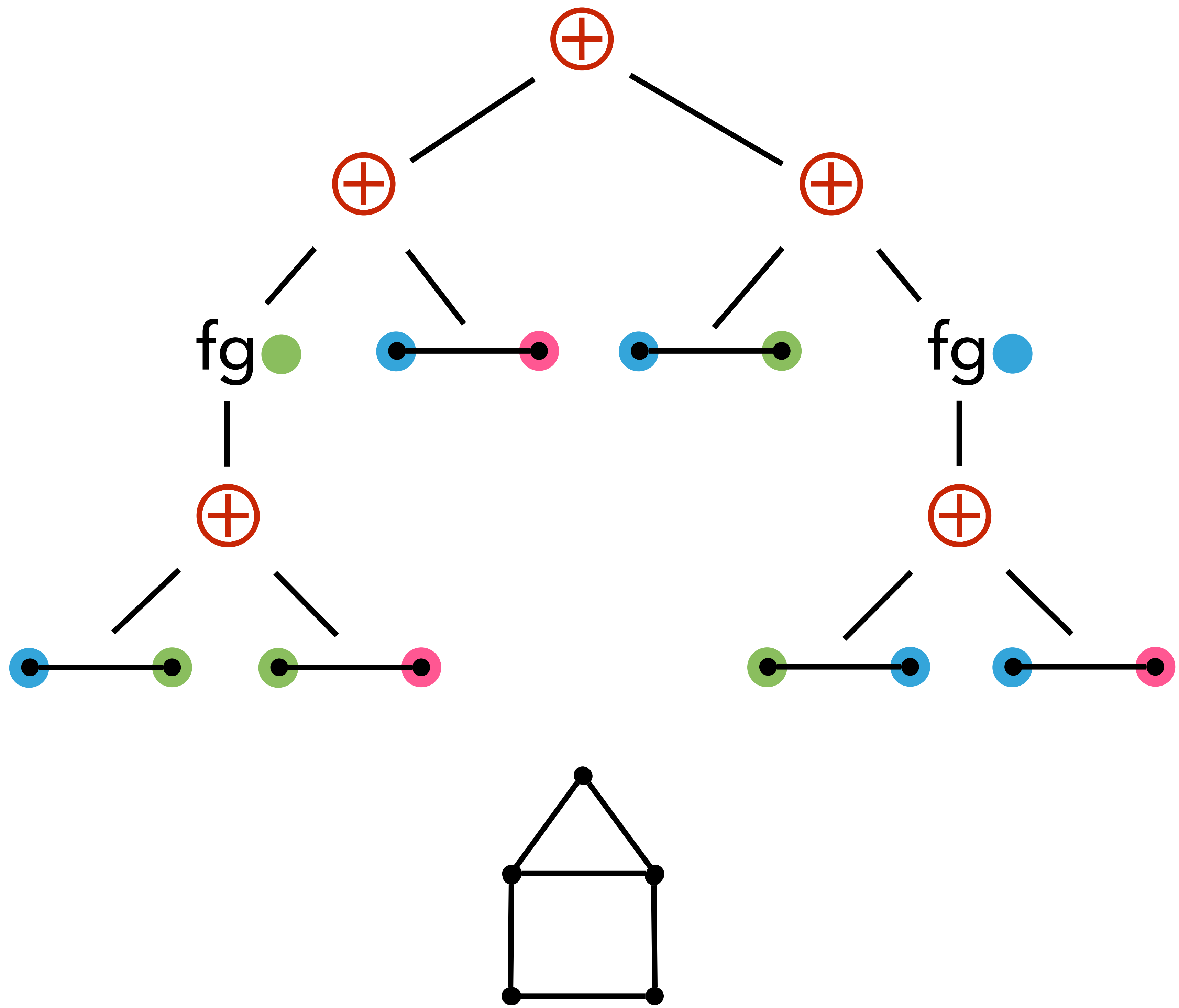
- Show that TC-words have bounded tree-width ✓ CONCUR'16 SPLIT-WIDTH
- Show that our properties are MSO-definable ✓
- Build directly tree automata for our properties CONCUR'16

OUTLINE

- BEHAVIOURS AS GRAPHS
- DECIDING PROPERTIES OF GRAPHS
- DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
- TREE-WIDTH FOR TIMED SYSTEMS
- INTERPRETING GRAPHS IN TREES
- CONCLUSION

TREE INTERPRETATION

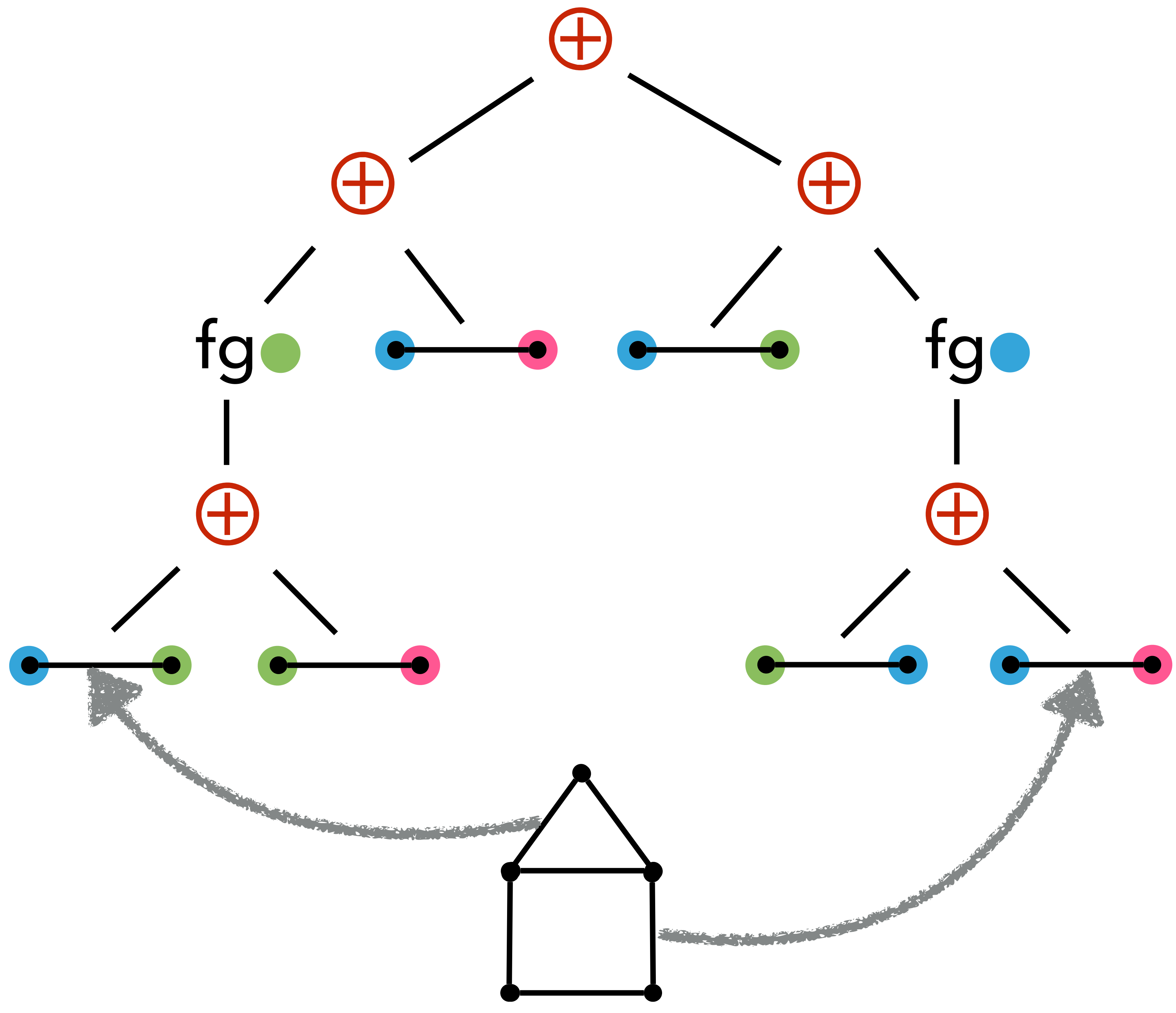
$$\tau ::= i \mid i \longrightarrow j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$



TREE INTERPRETATION

$$\tau ::= i \mid i \textbf{——} j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

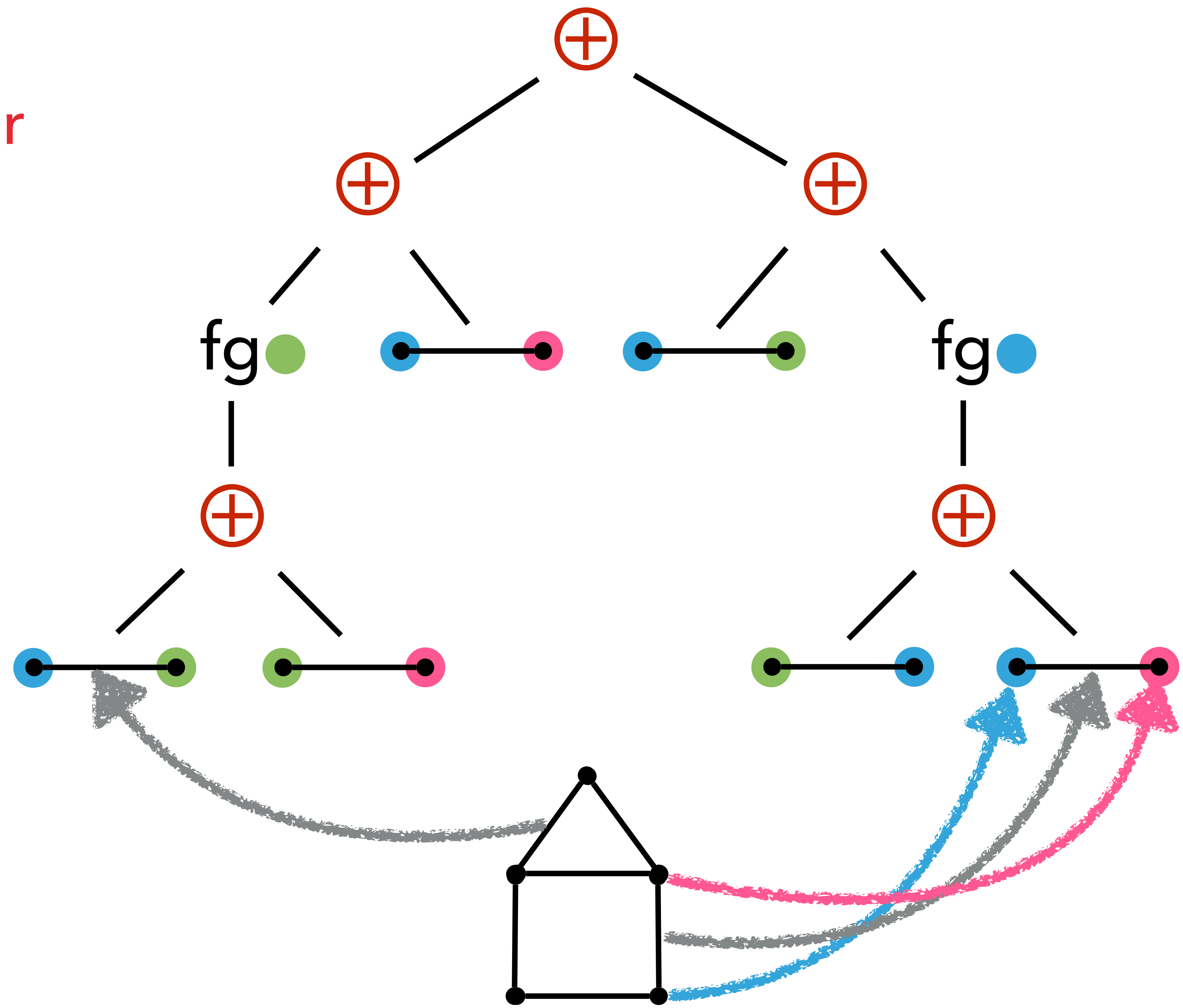
- Edge = leaf



TREE INTERPRETATION

$$\tau ::= i \mid i \longrightarrow j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

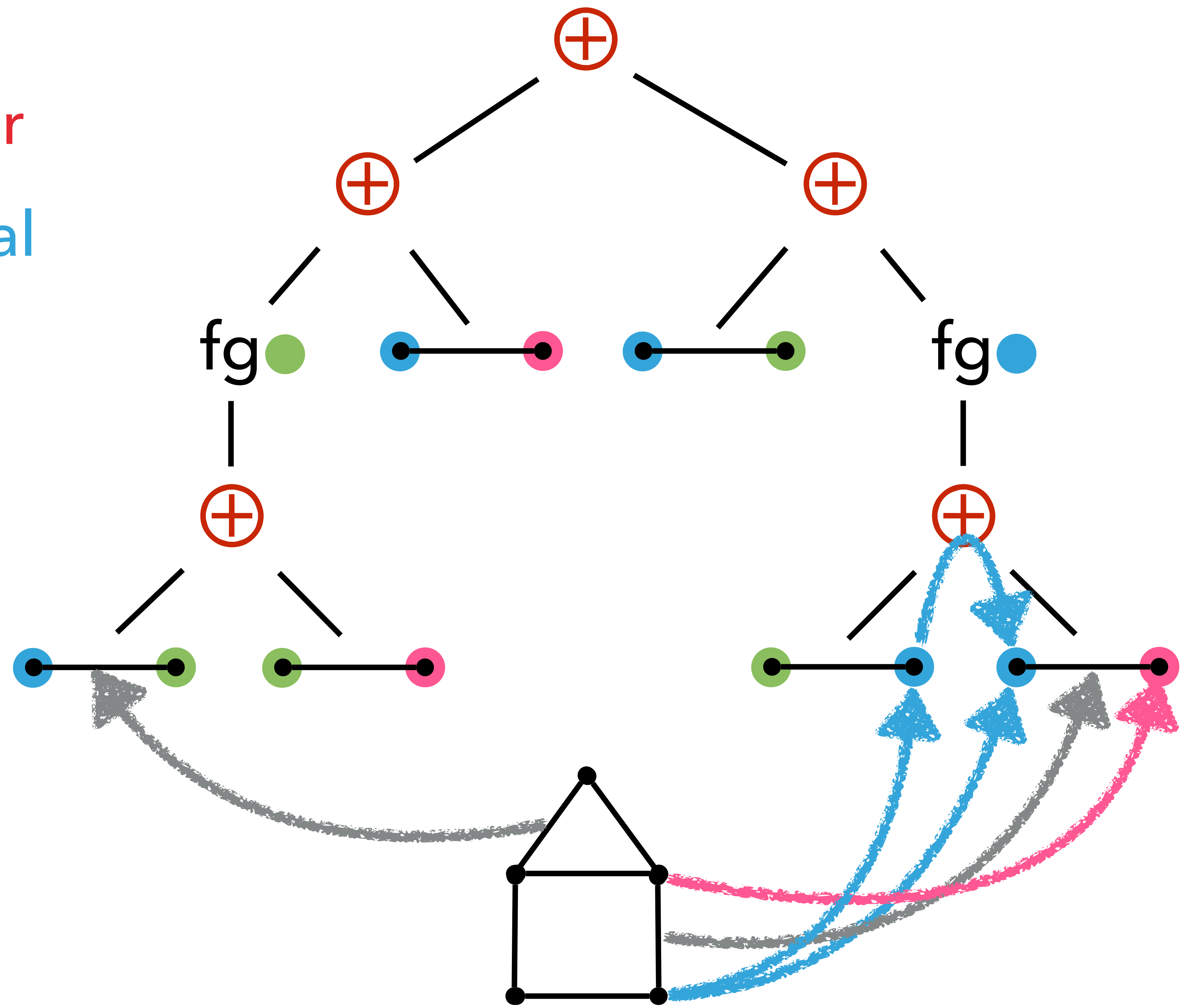
- Edge = leaf
- Vertex = leaf + color



TREE INTERPRETATION

$$\tau ::= i \mid i \text{ --- } j \mid \mathrm{fg}_i(\tau) \mid \tau \oplus \tau$$

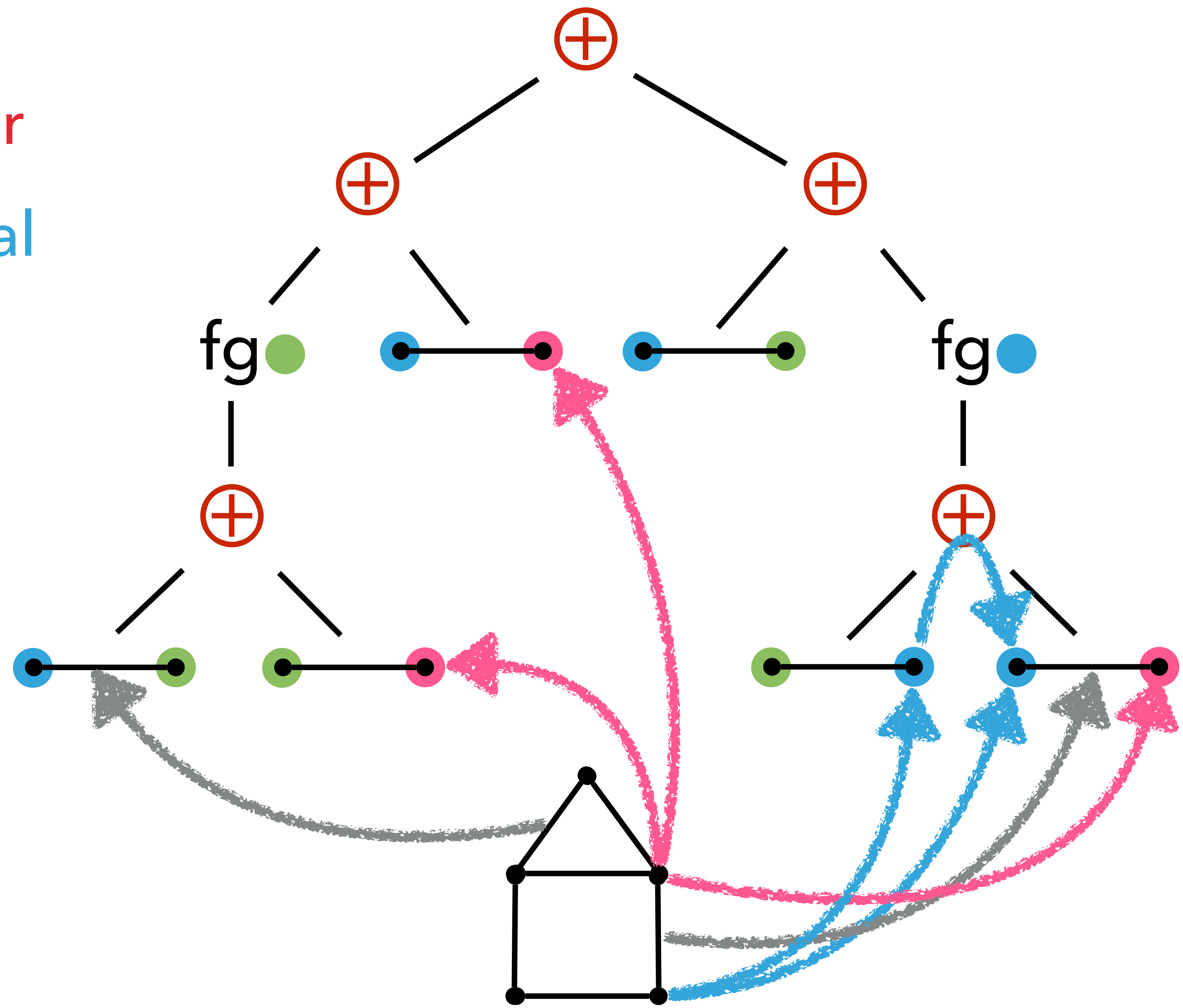
- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves



TREE INTERPRETATION

$$\tau ::= i \mid i \longrightarrow j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

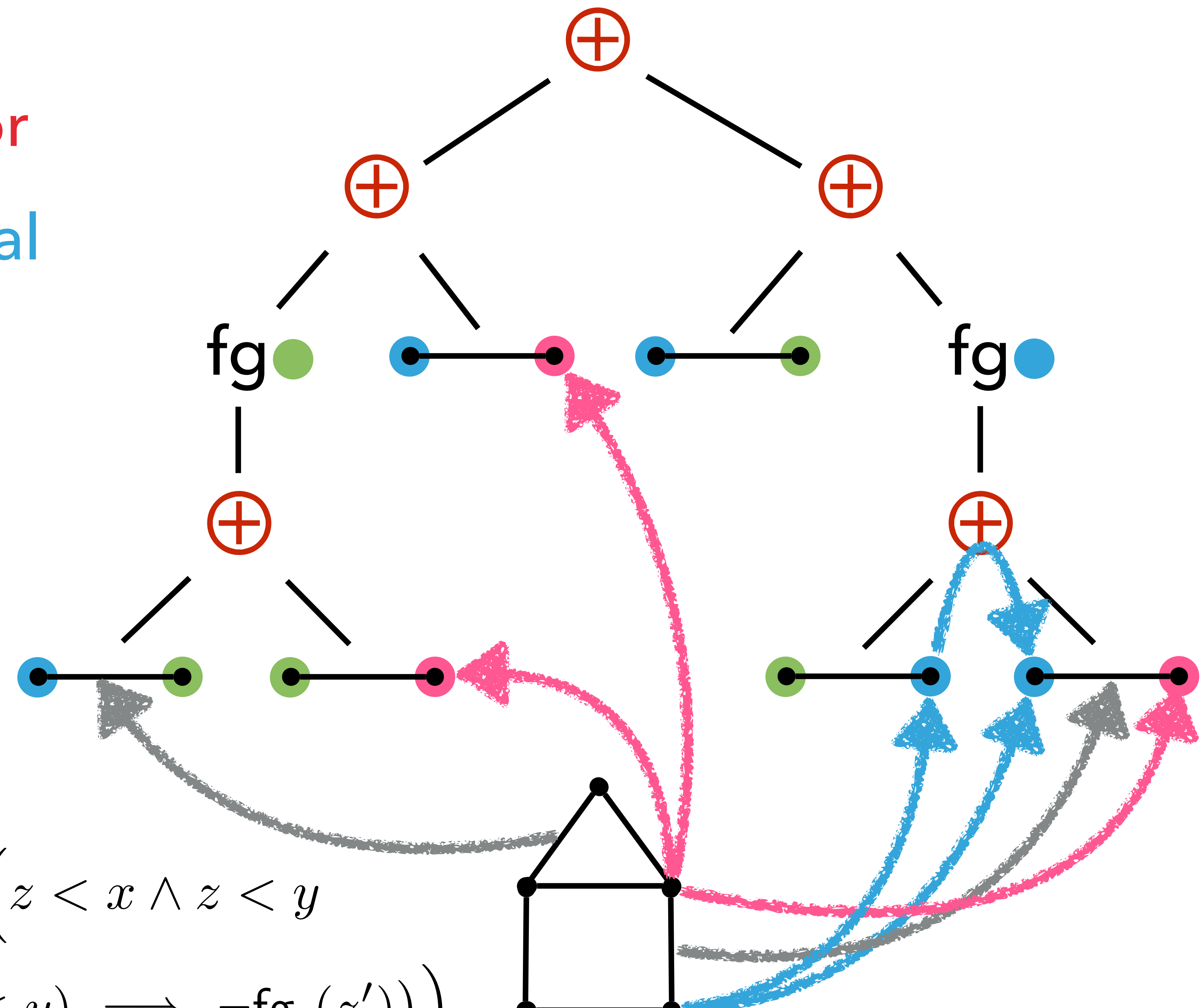
- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves



TREE INTERPRETATION

$$\tau ::= i \mid i \longrightarrow j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves
- SameVertex$_i$(x,y)

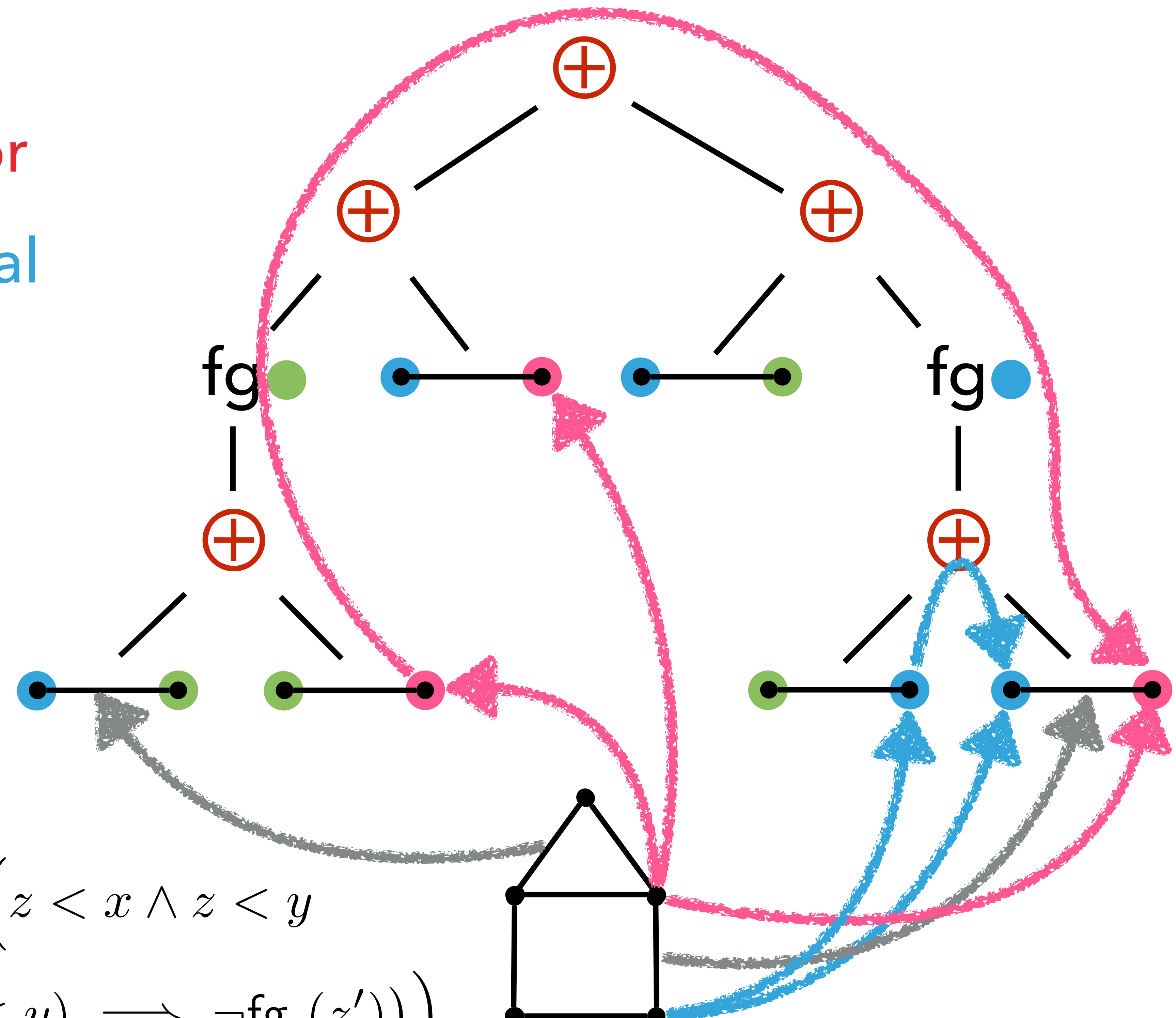


$$\mathsf{SameVertex}_i(x,y) ::= \exists z \Big(z < x \wedge z < y$$
$$\wedge \forall z' \big((z < z' < x \vee z < z' < y) \implies \neg \mathsf{fg}_i(z') \big) \Big)$$

TREE INTERPRETATION

$$\tau ::= i \mid i \text{ --- } j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves
- SameVertex$_i$(x,y)

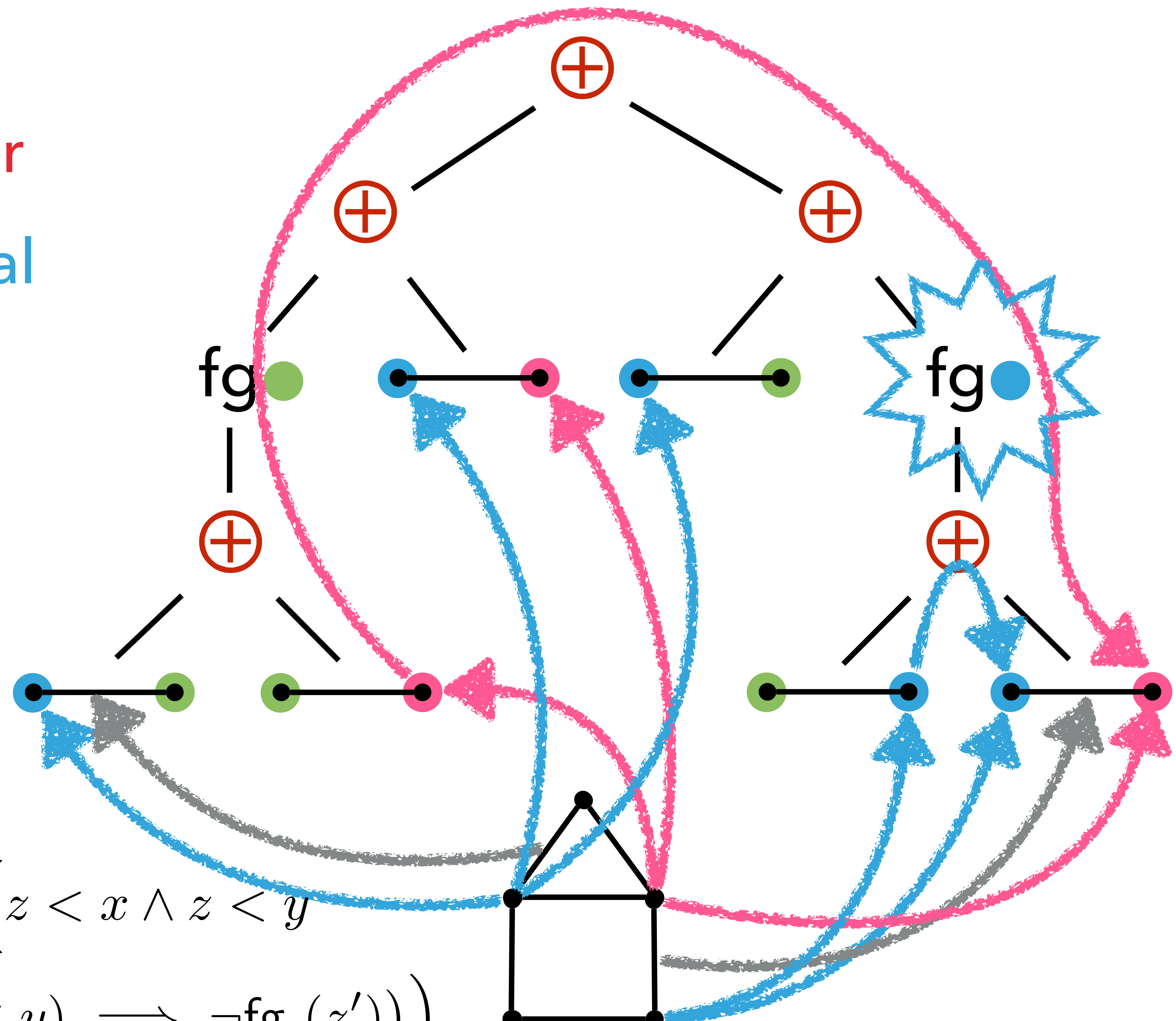


$$\mathsf{SameVertex}_i(x,y) ::= \exists z \Big(z < x \land z < y$$
$$\land \forall z' \big((z < z' < x \lor z < z' < y) \implies \neg \mathsf{fg}_i(z') \big) \Big)$$

TREE INTERPRETATION

$$\tau ::= i \mid i \longrightarrow j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves
- SameVertex$_i$(x,y)

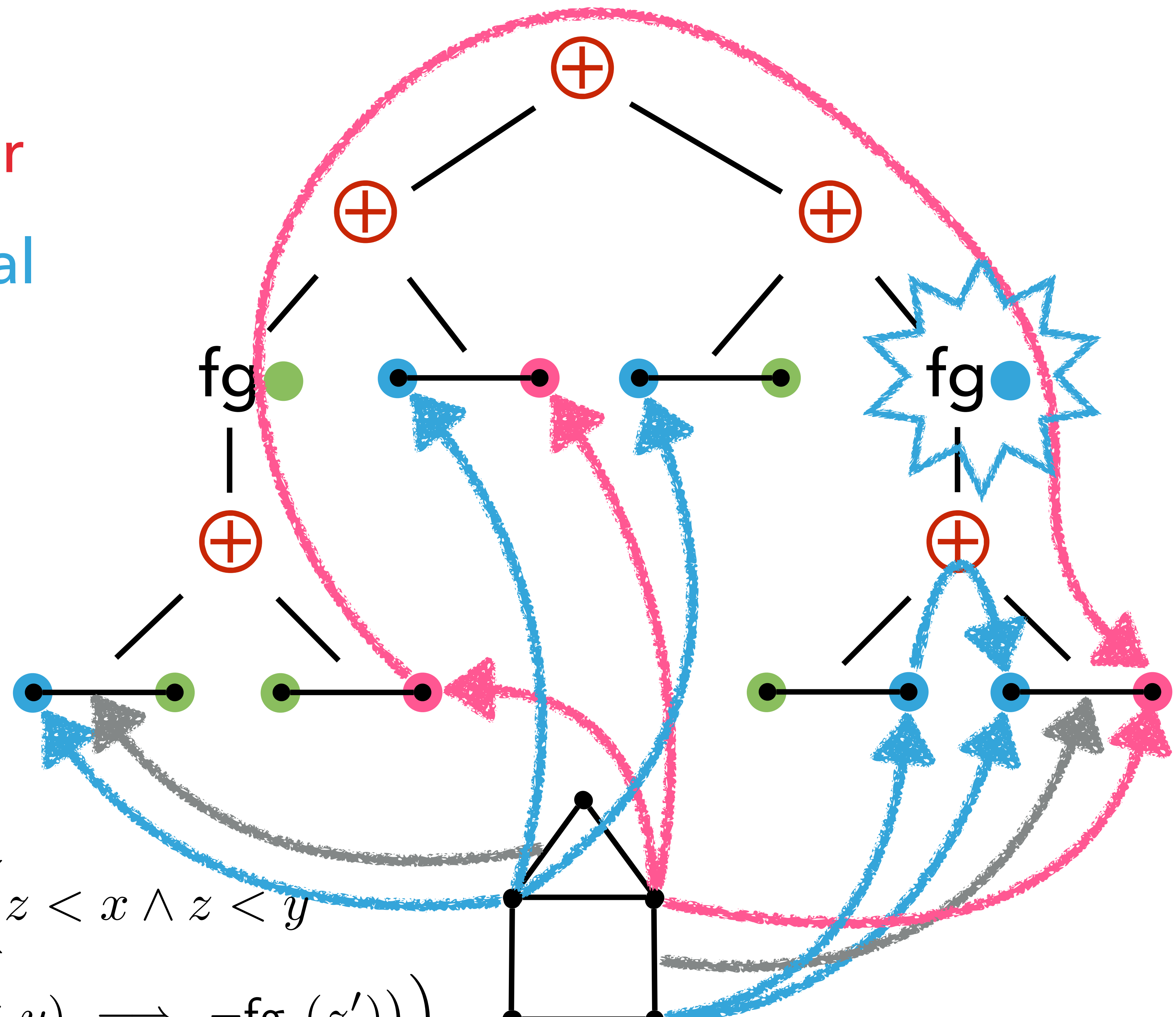


$$\mathsf{SameVertex}_i(x,y) ::= \exists z \Big(z < x \wedge z < y \wedge \forall z' \big((z < z' < x \vee z < z' < y) \implies \neg \mathsf{fg}_i(z') \big) \Big)$$

TREE INTERPRETATION

$$\tau ::= i \mid i \longrightarrow j \mid \mathsf{fg}_i(\tau) \mid \tau \oplus \tau$$

- Edge = leaf
- Vertex = leaf + color
- One vertex = several leaves
- SameVertex$_i$(x,y)



LCPDL/MSO OVER GRAPHS

LCPDL/MSO OVER TREES

$$\mathsf{SameVertex}_i(x,y) ::= \exists z \Big(z < x \wedge z < y \wedge \forall z' \big((z < z' < x \vee z < z' < y) \implies \neg \mathsf{fg}_i(z') \big) \Big)$$

DIRECTLY BUILDING TREE AUTOMATA

CONCUR'16

☞ We can build a tree automaton $\mathcal{A}^k_{\mathsf{valid}}$ of size $2^{\mathcal{O}(k^2)}$ which accepts all k-terms denoting valid TCWs.

☞ We can build a tree automaton $\mathcal{A}^{k,M}_{\mathsf{real}}$ of size $M^{\mathsf{poly}(k)}$ which accepts all k-terms denoting realizable TCWs using constants at most M.

☞ Let $\mathcal{S}$ be a pushdown timed automaton with set of clocks X.
Let $|\mathcal{S}|$ be its size (constants encoded in unary) and $k = 3|X| + 2$.

We can build a tree automaton $\mathcal{A}^k_{\mathcal{S}}$ of size $|\mathcal{S}|^{\mathsf{poly}(k)}$ which accepts all k-terms denoting TCWs in $\mathcal{L}_{\mathsf{TCW}}(\mathcal{S})$.

NON EMPTINESS / REACHABILITY $\mathcal{L}(\mathcal{S}) \neq \emptyset \iff \mathcal{L}(\mathcal{A}^k_{\mathsf{valid}} \cap \mathcal{A}^{k,M}_{\mathsf{real}} \cap \mathcal{A}^k_{\mathcal{S}}) \neq \emptyset$

COURCELLE'S THEOREM

- Let TW_k be the set of graphs of tree-width at most k
- Let P be a property of graphs
- If P is MSO-definable then $P \cap TW_k \neq \emptyset$ is decidable

WE WANT TO SOLVE $\mathscr{L}_{\mathrm{TCW}}(\mathcal{A}) \cap \mathfrak{Real}_{\mathrm{TCW}} \neq \emptyset$

CONCUR'16
SPLIT-WIDTH

- Show that TC-words have bounded tree-width ✓
- Show that our properties are MSO-definable ✓
- Build directly tree automata for our properties ✓

OUTLINE

- BEHAVIOURS AS GRAPHS
- DECIDING PROPERTIES OF GRAPHS
- DEFINABILITY OF PROPERTIES FOR TIMED SYSTEMS
- TREE-WIDTH FOR TIMED SYSTEMS
- INTERPRETING GRAPHS IN TREES
- CONCLUSION

CONCLUSION

NEW TECHNIQUE FOR ANALYZING TIMED SYSTEMS

1. Write behaviors as graphs with timing constraints
2. Show a bound on tree-width for these graphs
3. Show MSO-definability of the relevant properties, or
4. Build Tree automata directly

RESULTS

- PSPACE decision procedure for timed automata
- EXPTIME decision procedure for pushdown timed automata
- EXPTIME decision procedure for multi-pushdown timed automata with bounded rounds

CONCLUSION

NEW TECHNIQUE FOR ANALYZING TIMED SYSTEMS

1. Write behaviors as graphs with timing constraints
2. Show a bound on tree-width for these graphs
3. Show MSO-definability of the relevant properties, or
4. Build Tree automata directly

FUTURE WORK

- Efficient implementation
- Concurrent recursive timed programs
- MSO/LCPDL-definability of realizability and non-realizability
- Model-Checking wrt. timed specifications

THANK YOU