

Géométrie Algorithmique

CALCUL D'INTERSECTIONS

Abdelkrim Mebarki

Université des Sciences et de la Technologie d'Oran

Faculté des Mathématiques et Informatique

Département d'Informatique

2019 – 2020

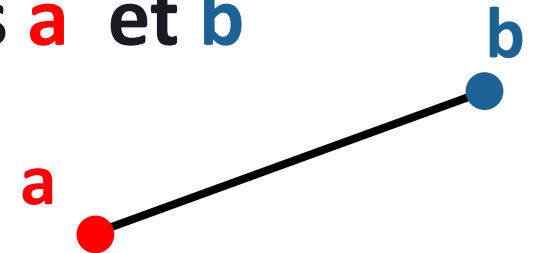
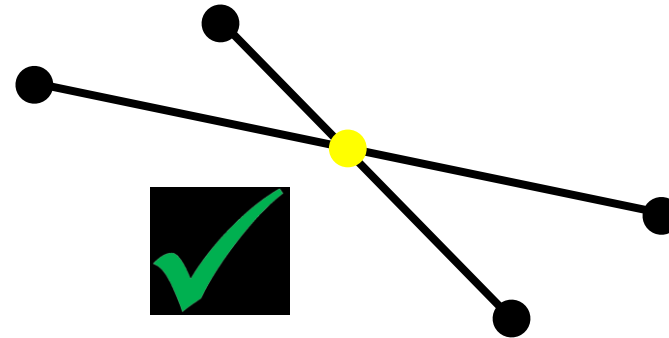
Segments

- Un segment $[a b]$ est défini par ses deux extrémités a et b

- Pour $\lambda \in [1,0]$

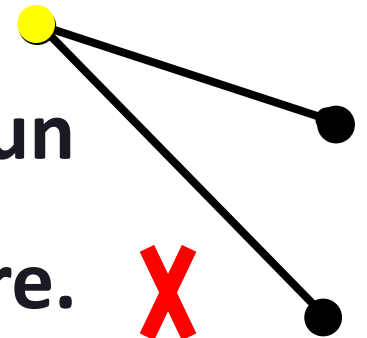
- $\lambda * a.x + (1 - \lambda) * b.x$

- $\lambda * a.y + (1 - \lambda) * b.y$

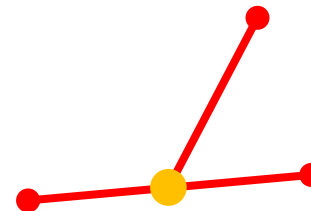
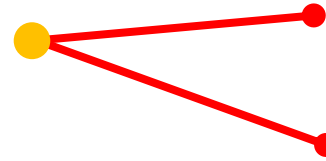
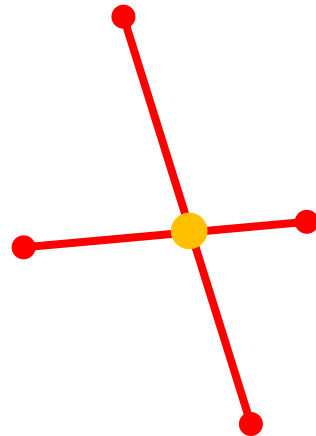
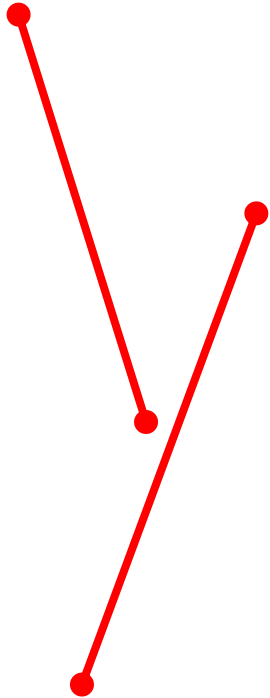


- Une intersection entre deux segments : Un point commun

- Intersection strictement à l'intérieur : intersection propre. **X**



Segments



Intersection de segments ?

- **Entrée :**

- Un ensemble de n segments $S = \{S_1, S_2, \dots, S_n\}$

- **Sortie :**

- **Détection de l'intersection :**

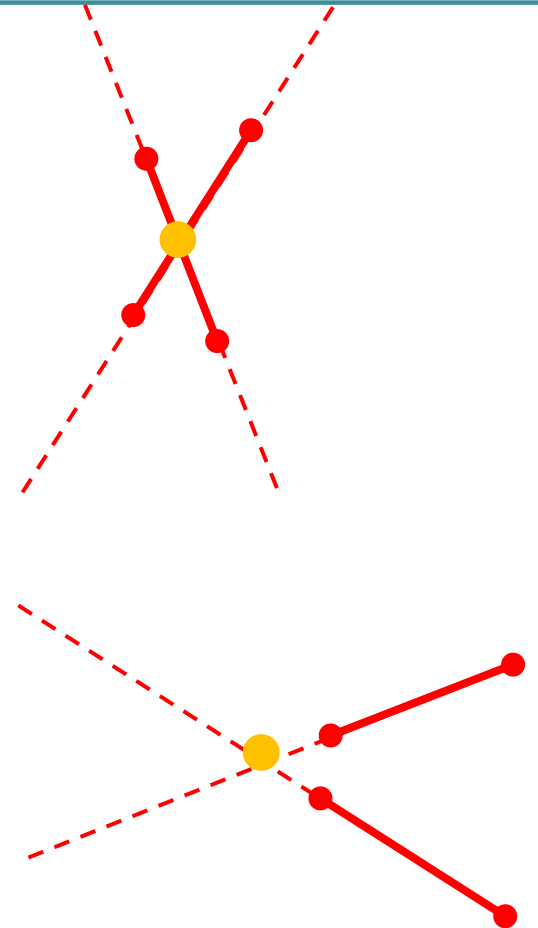
- Y-a-t-il une paire (S_i, S_j) de S^2 où $i \neq j$ et $S_i \cap S_j \neq \{\}$?

- **Calcul de l'intersection :**

- Trouver toutes les paires (S_i, S_j) de S^2 où $i \neq j$ et $S_i \cap S_j \neq \{\}$

Cas de 2 segments

- Calcul de l'intersection des deux supports:
 - Système linéaire de deux équations à deux variables
 - Comment ?
 - Comparer la solution avec les deux extrémités :
 - A l'intérieur : Il existe une intersection
 - Sinon : Intersection vide
- Complexité : $O(1)$
- Cas dégénéré : L'intersection est une ligne ?



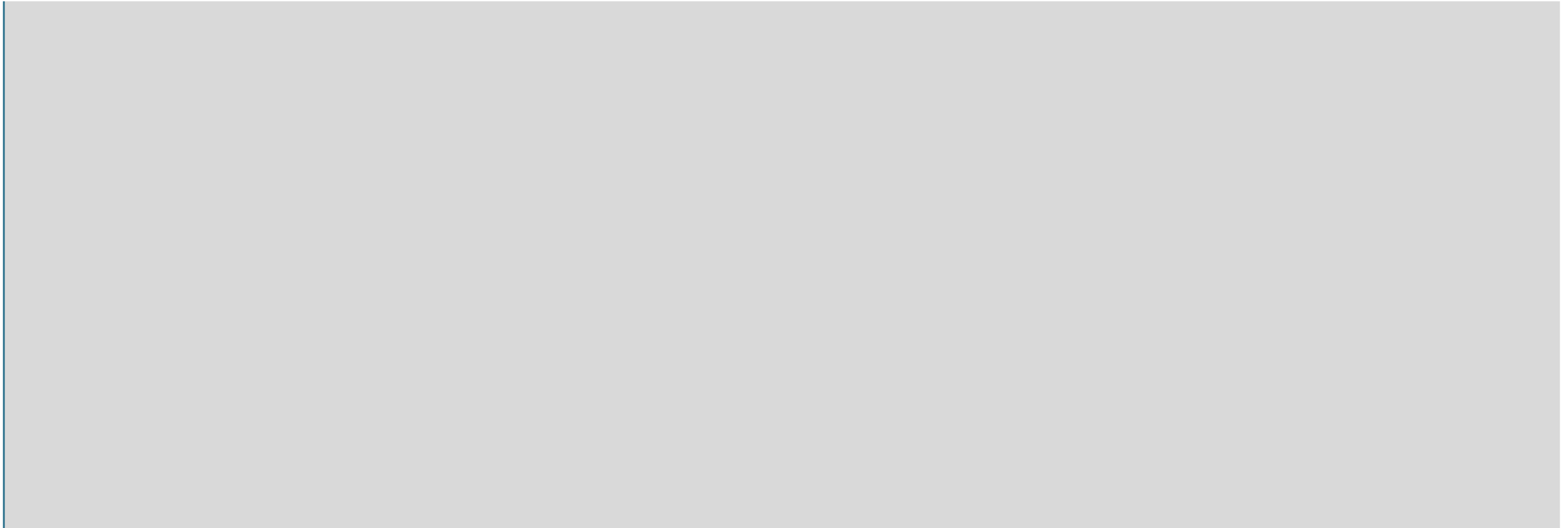
Calcul des intersections ?

- Calculer toutes les intersections entre tous les segments
 - $O(n^2)$?

Peut on faire mieux

- ~~Calculer toutes les intersection : $O(n^2)$~~
- Détecter l'intersection entre les segments . .
- Calculer les intersections s'ils existent . .
- Comment ?

Balayage



Balayage

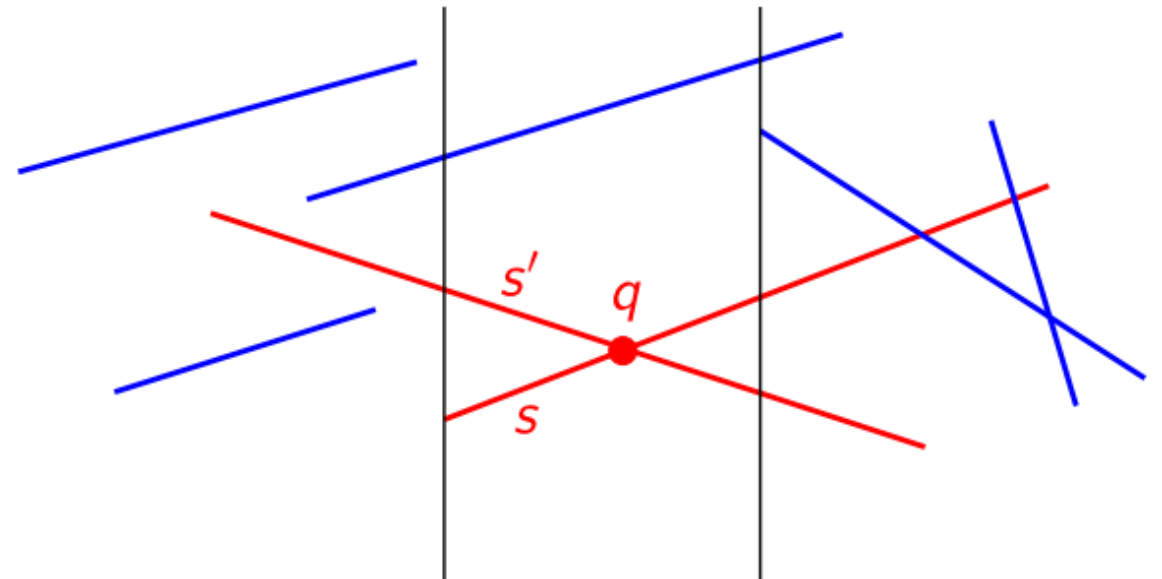
- **Principe de l'algorithme :**
 - Une ligne verticale traverse le plan de gauche à droite et affiche la scène
 - A chaque évènement (début ou fin d'un segment, une intersection), correspond un traitement
- **Définition des cas **dégénérés** :**
 - 3 points collinaires
 - Deux points avec la même abscisse (x)
 - 3 segments qui se coupent dans le même point

Balayage

- Soit une ligne verticale $l_t : x = t$
- $State_t$ est l'ensemble des Segments en intersection avec l_t
- L'algorithme définit un traitement pour chaque évènement : les extrémités des segments et les intersections
- Les extrémités des segments sont triés par leurs ordonnées y suivant leurs intersections avec l_t

Observation géométrique

- S'il existe une intersection entre deux segments, ces deux segments sont adjacents dans la liste $State_t$, L'état strictement antérieur à l'intersection
- Pourquoi ?



Les évènements

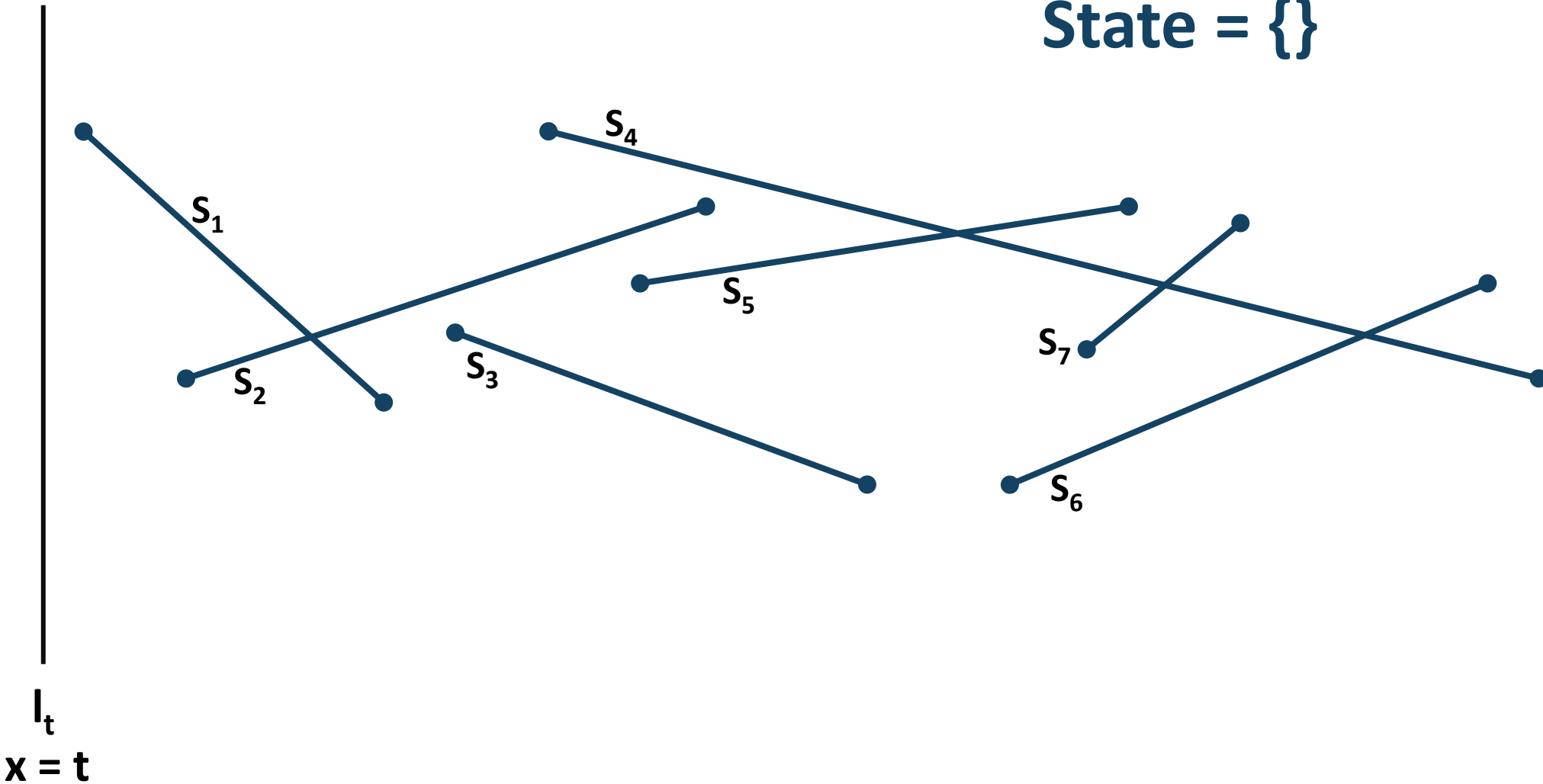
- **1^{er} cas : Extrémité gauche d'un segment**
 - Insertion dans la liste $State_t$
- **2^{ème} cas : Extrémité droite d'un segment**
 - Suppression de la liste $State_t$
- **Comment détecter les intersections ?**

Les évènements

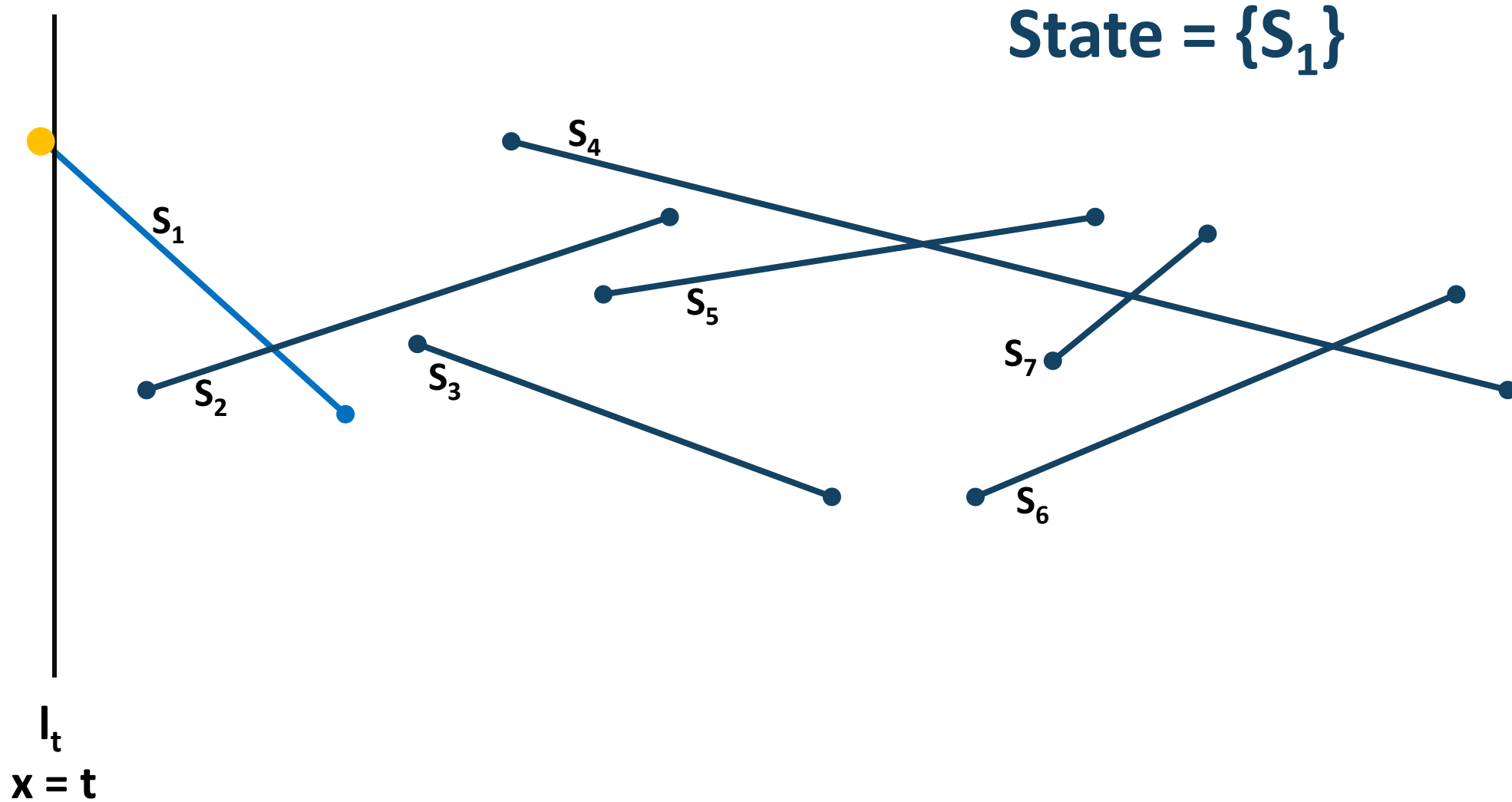
- **1^{er} cas : Extrémité gauche d'un segment**
 - Insertion dans la liste $State_t$ (Test d'intersection avec les voisins)
- **2^{ème} cas : Extrémité droite d'un segment**
 - Suppression de la liste $State_t$ (Test d'intersection avec les voisins)
- ~~Comment détecter les intersections ?~~

Exemple

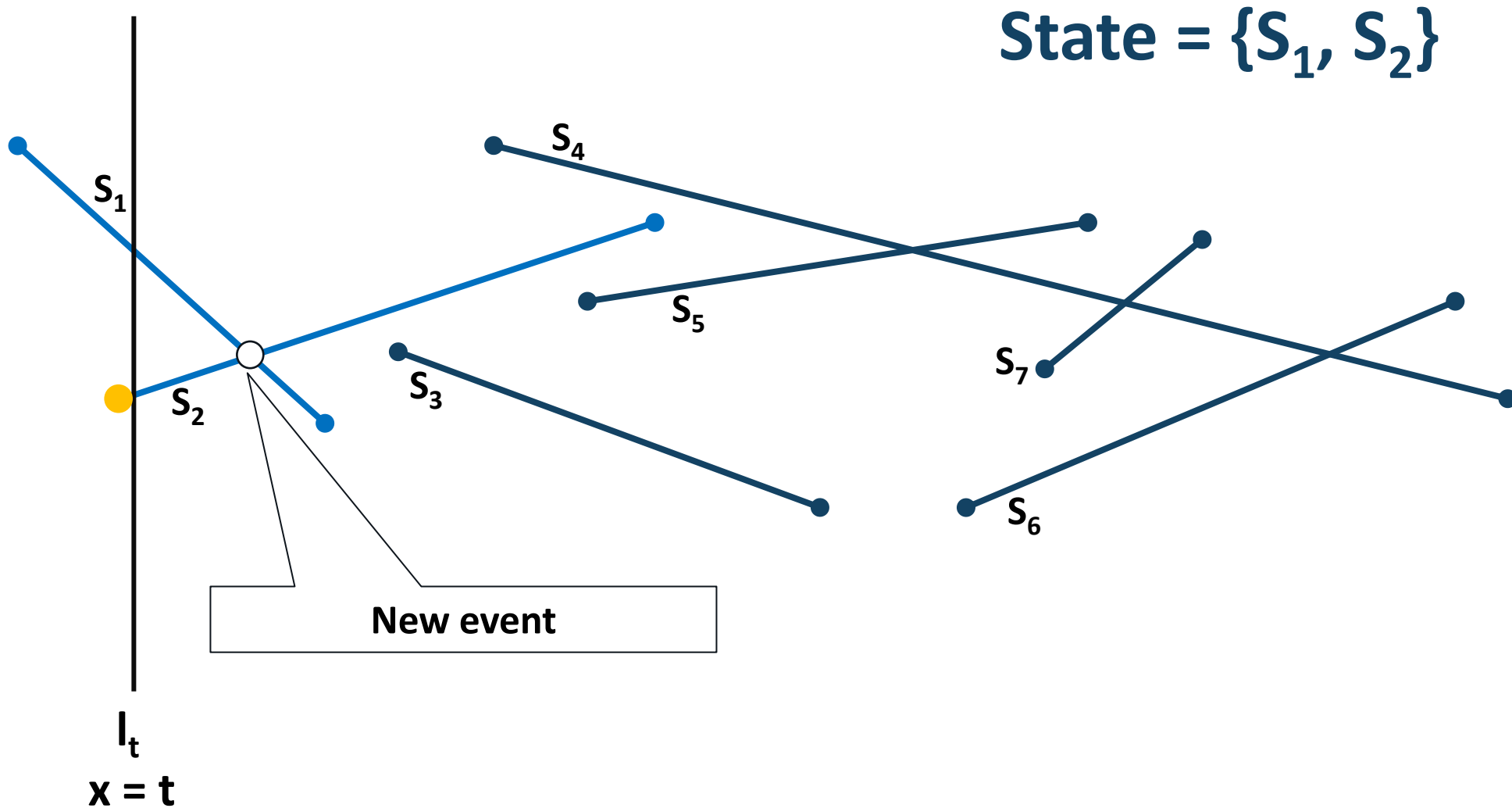
State = {}



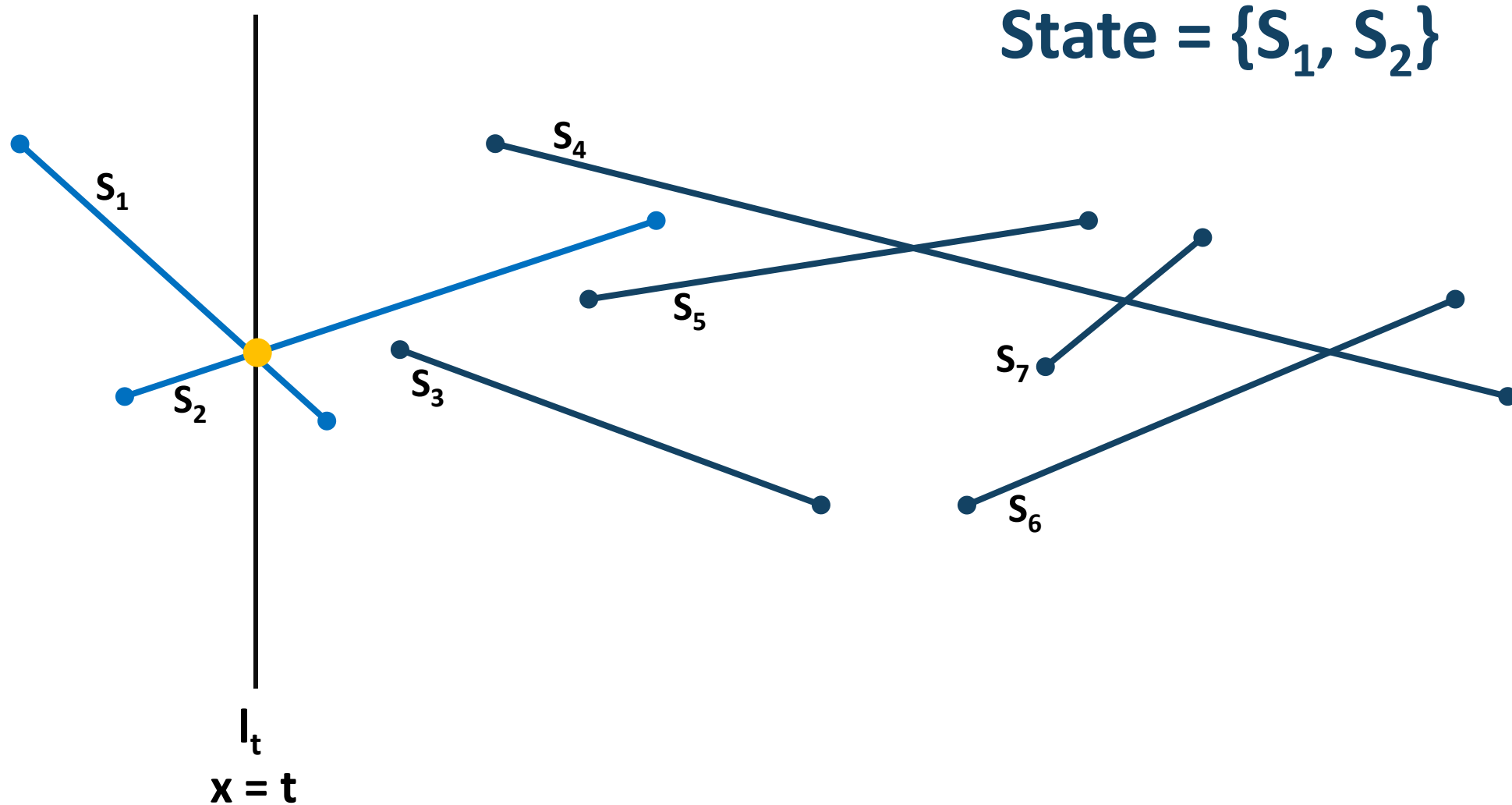
Exemple



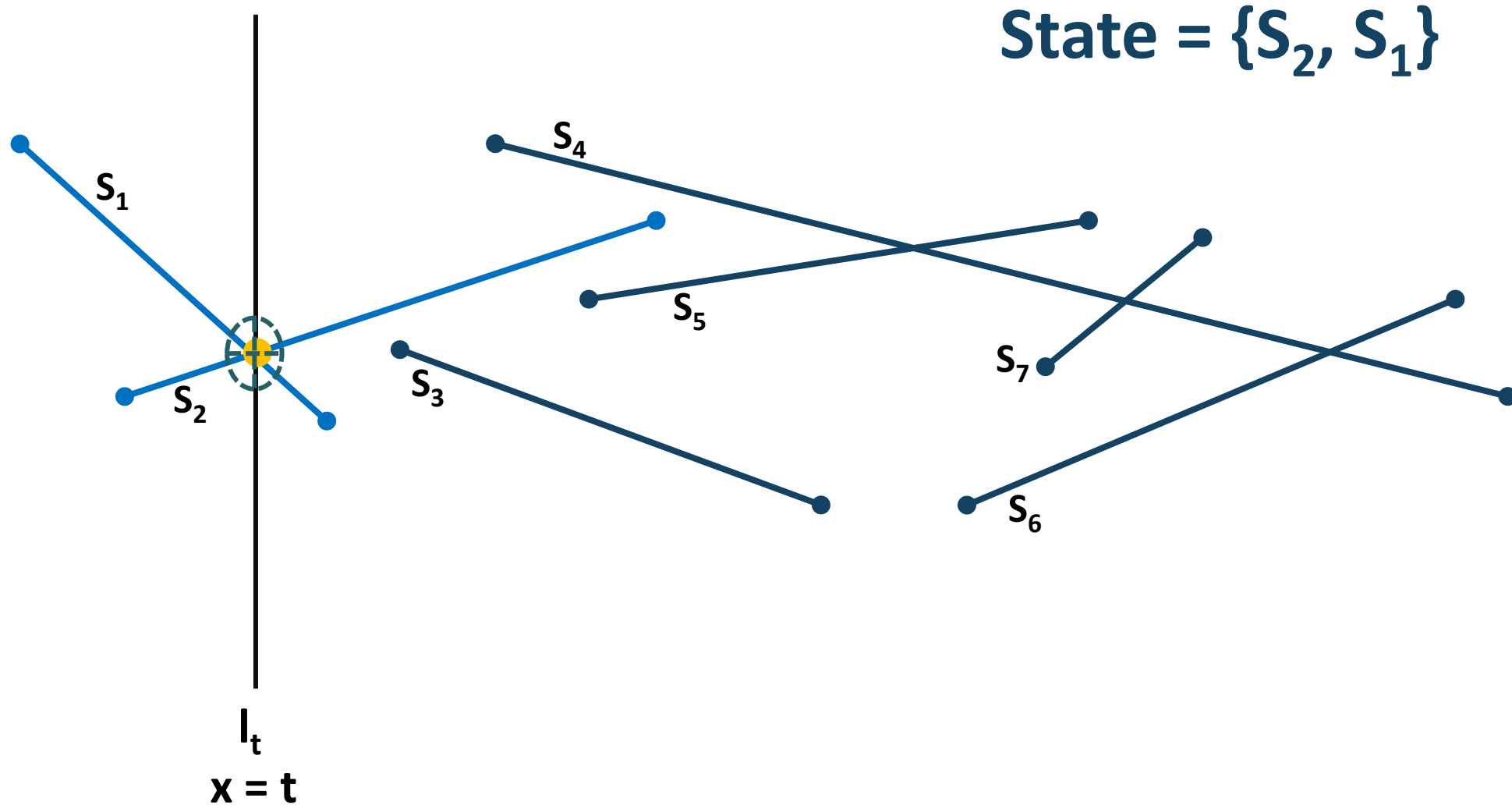
Exemple



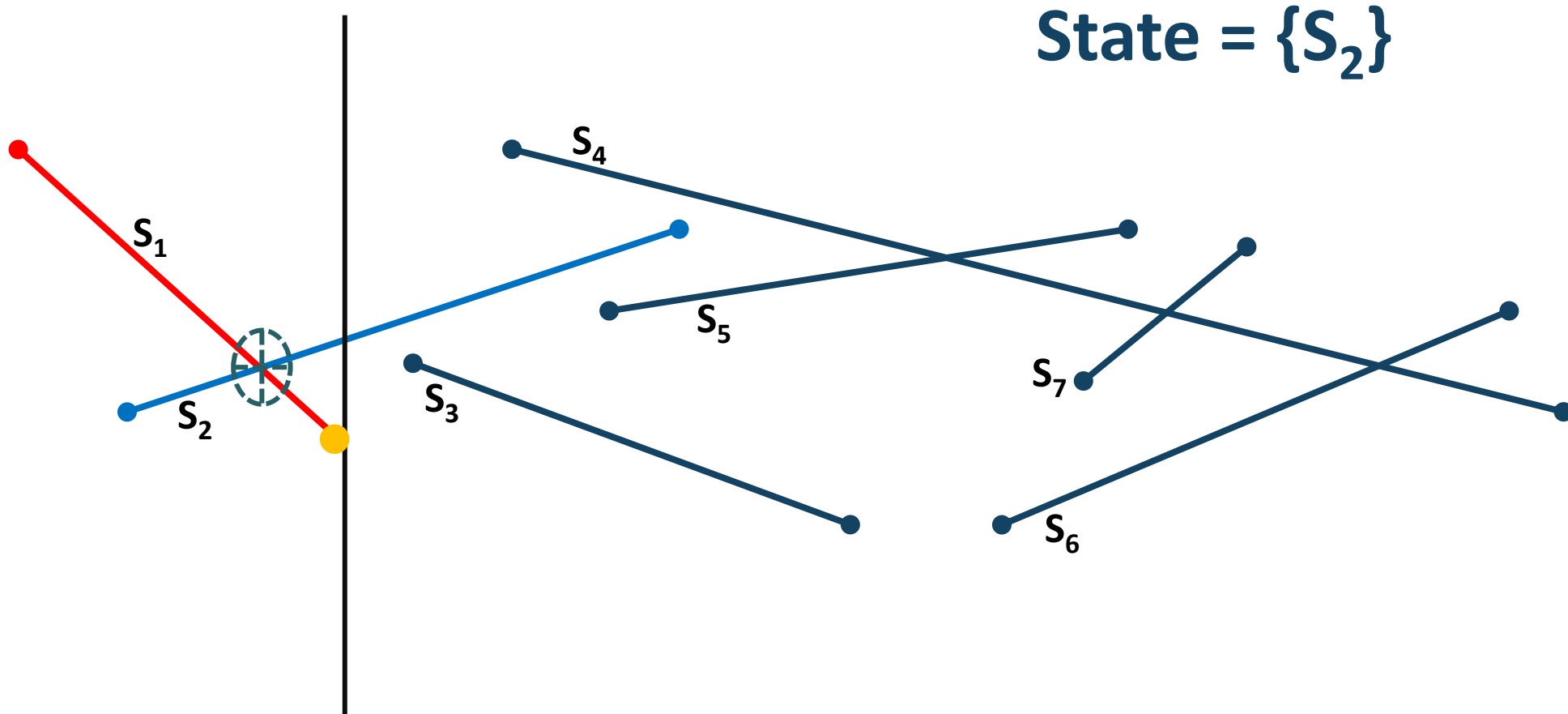
Exemple



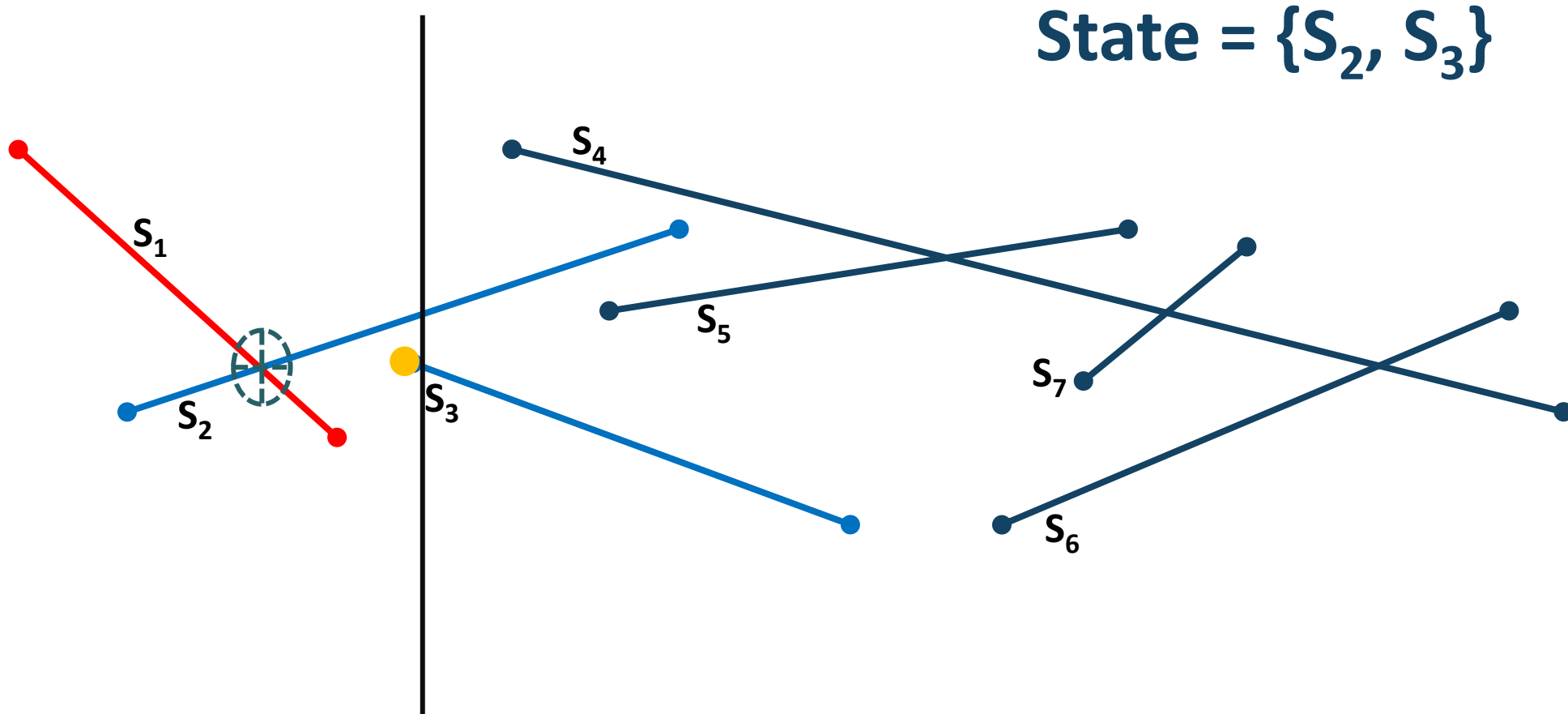
Exemple



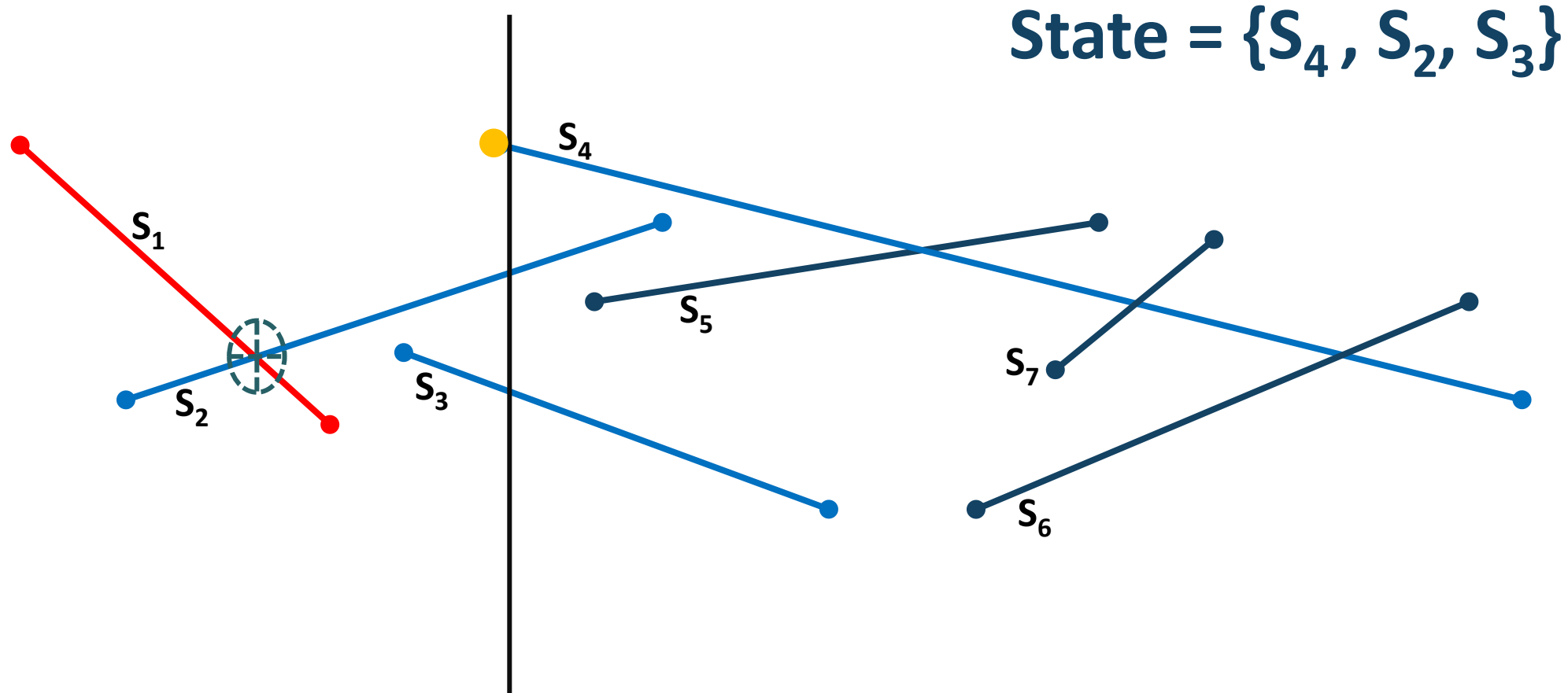
Exemple



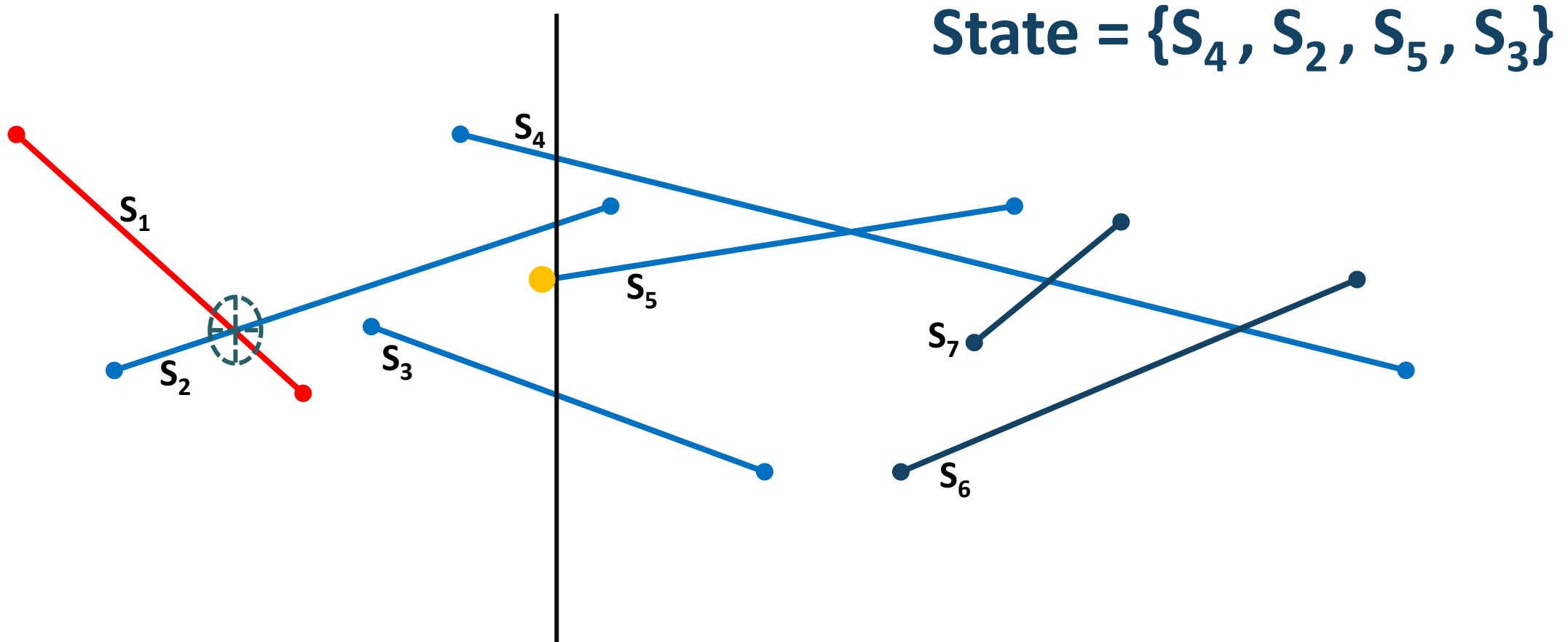
Exemple



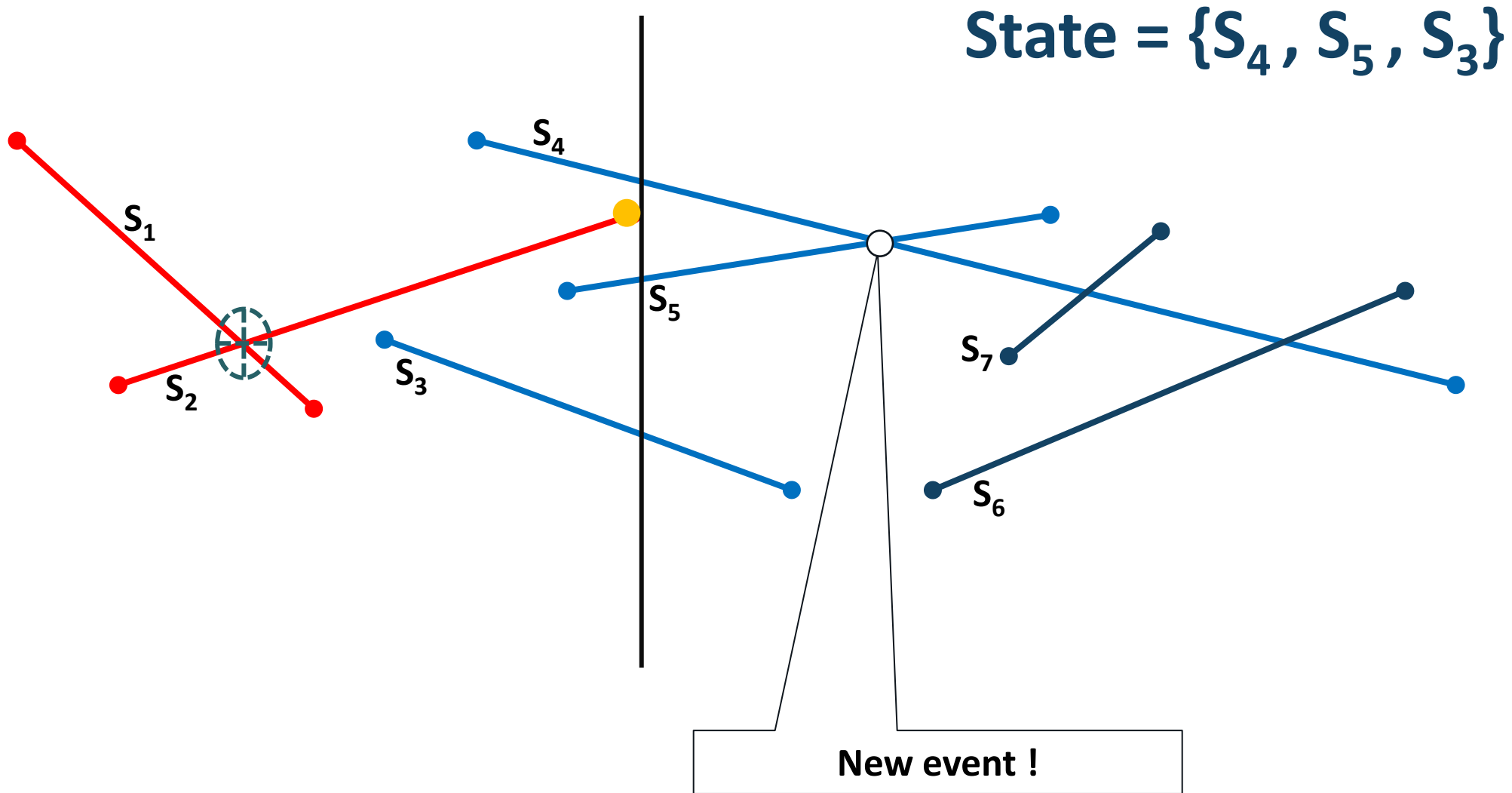
Exemple



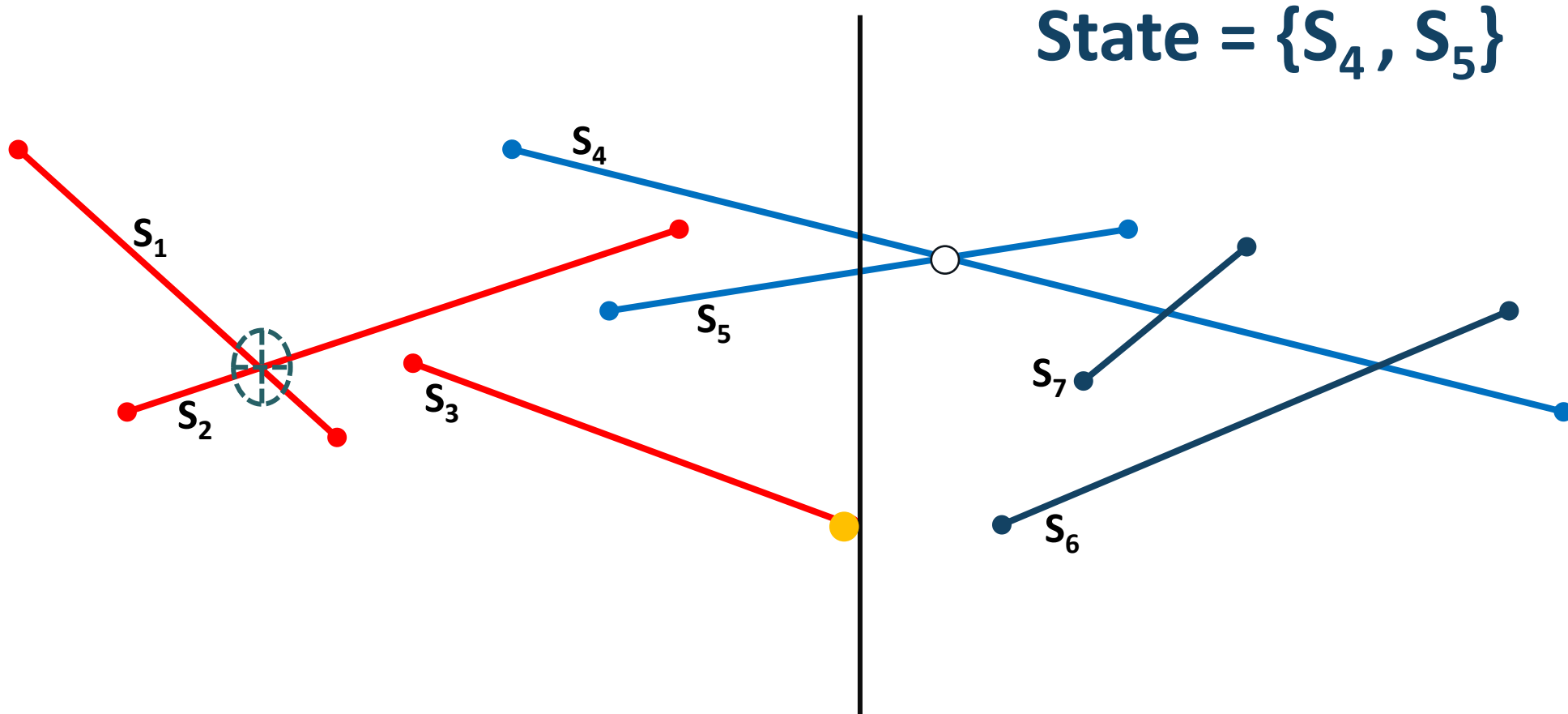
Exemple



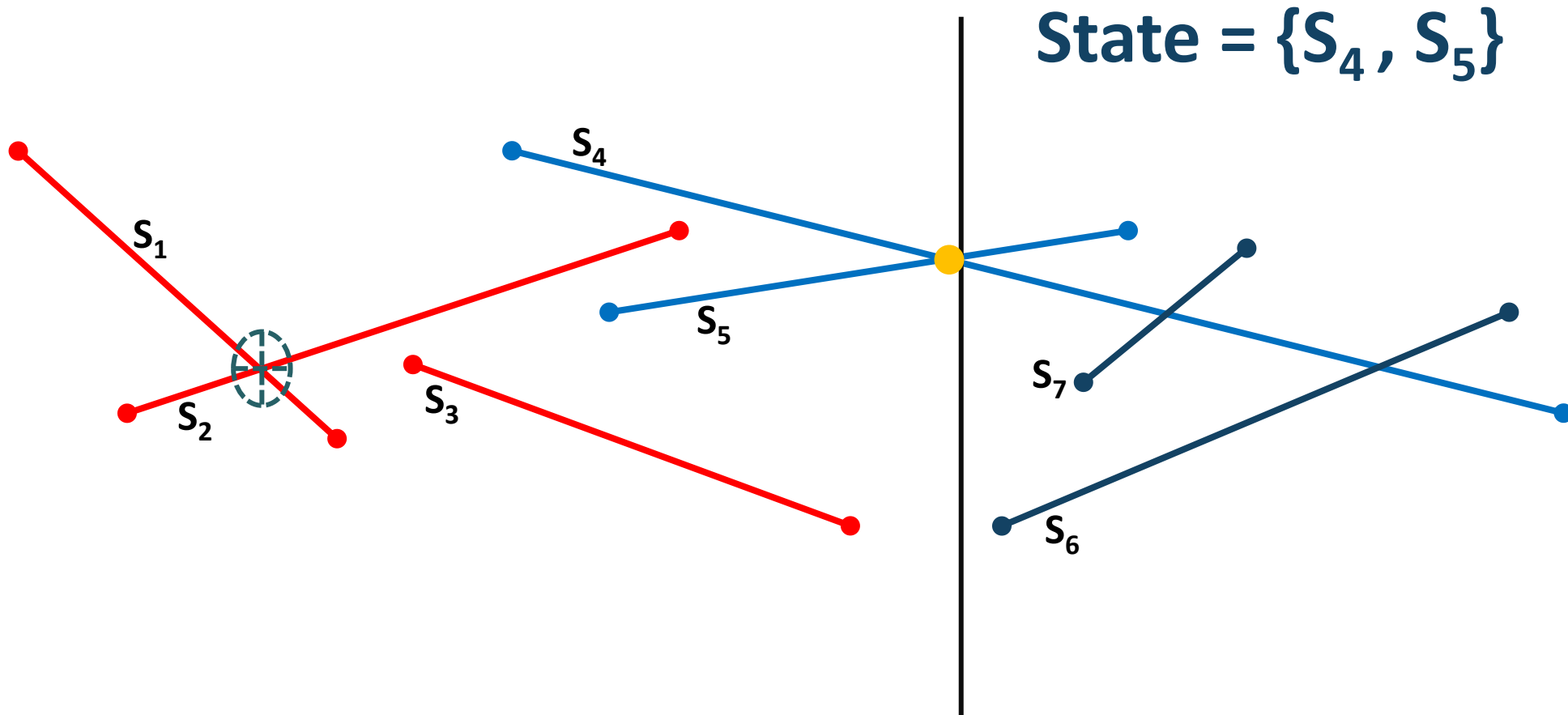
Exemple



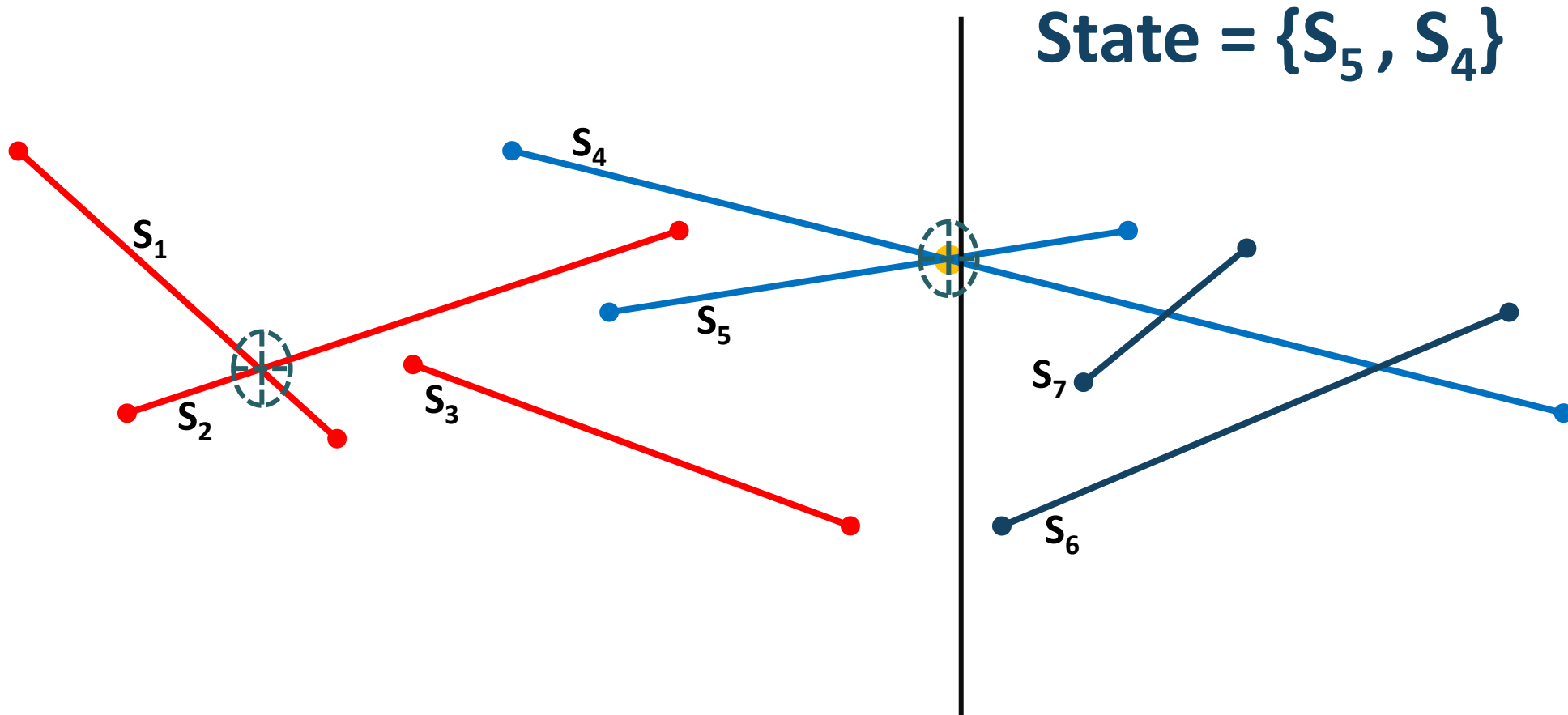
Exemple



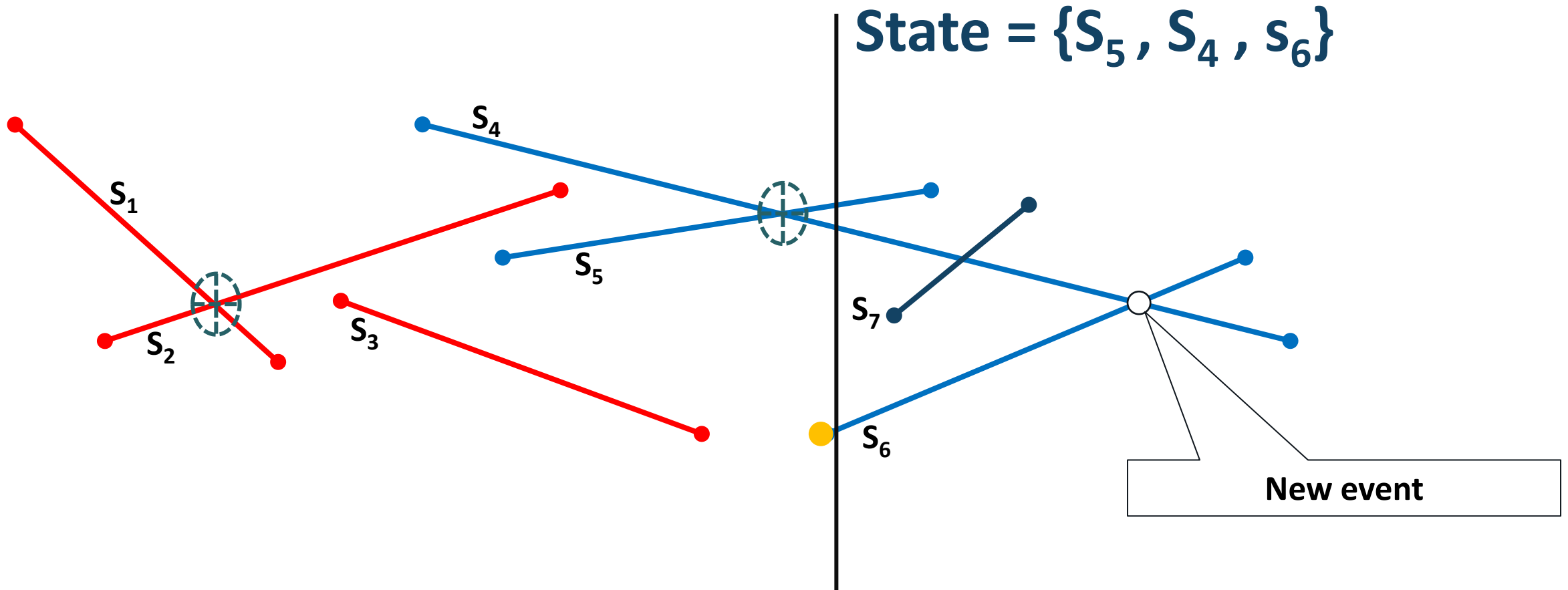
Exemple



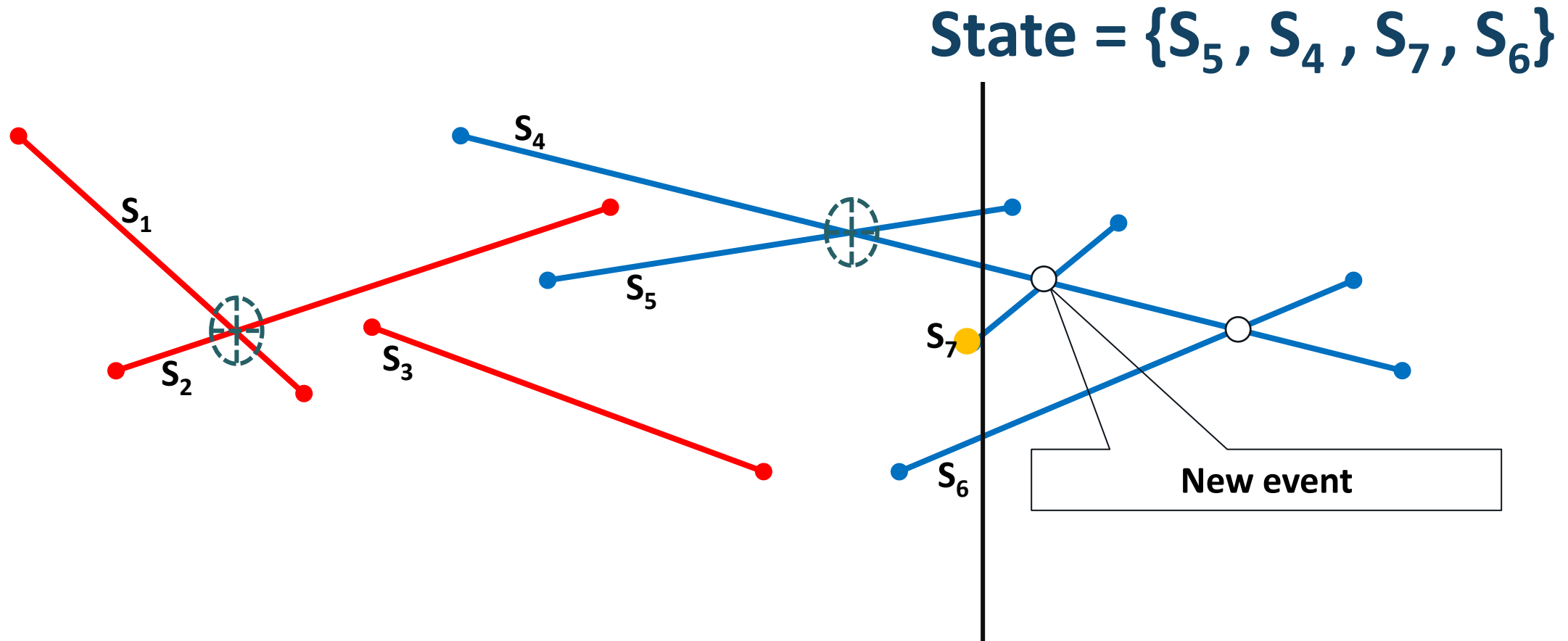
Exemple



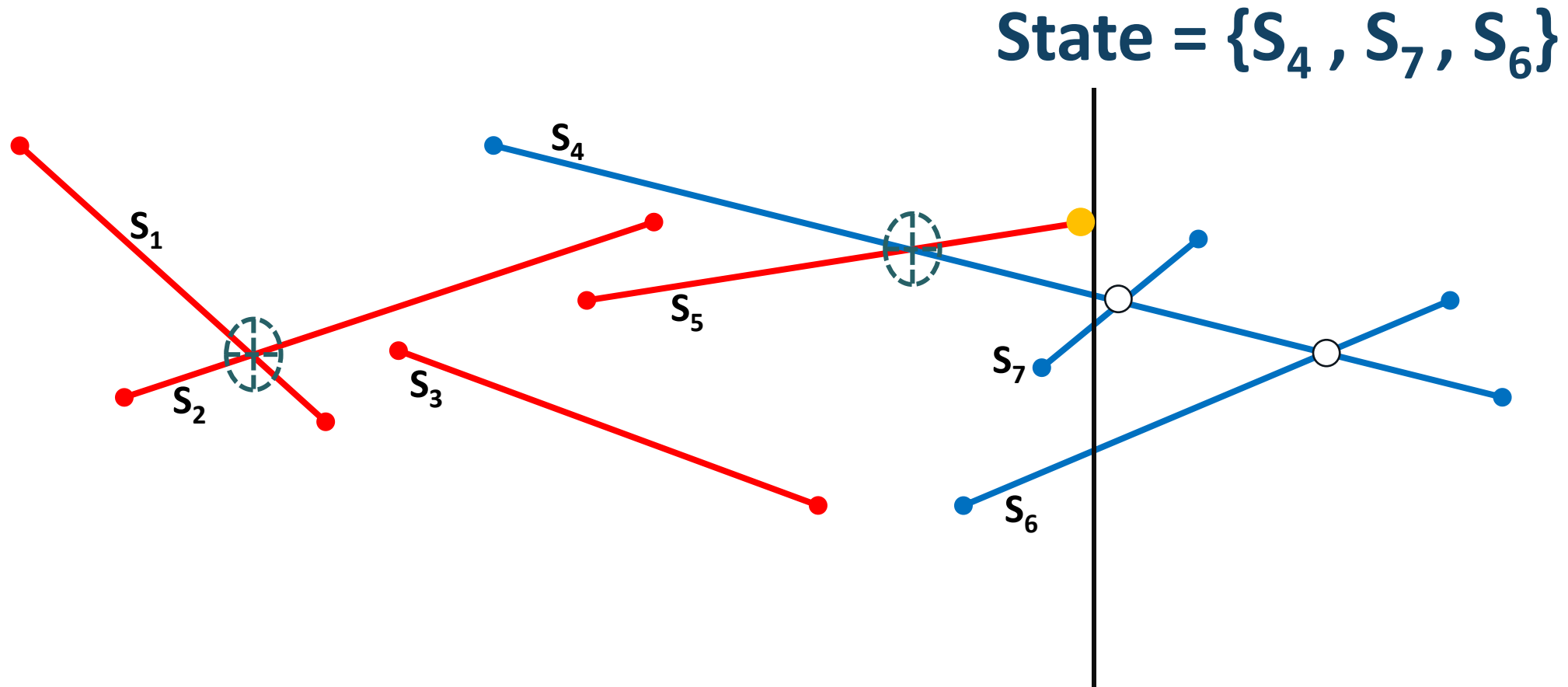
Exemple



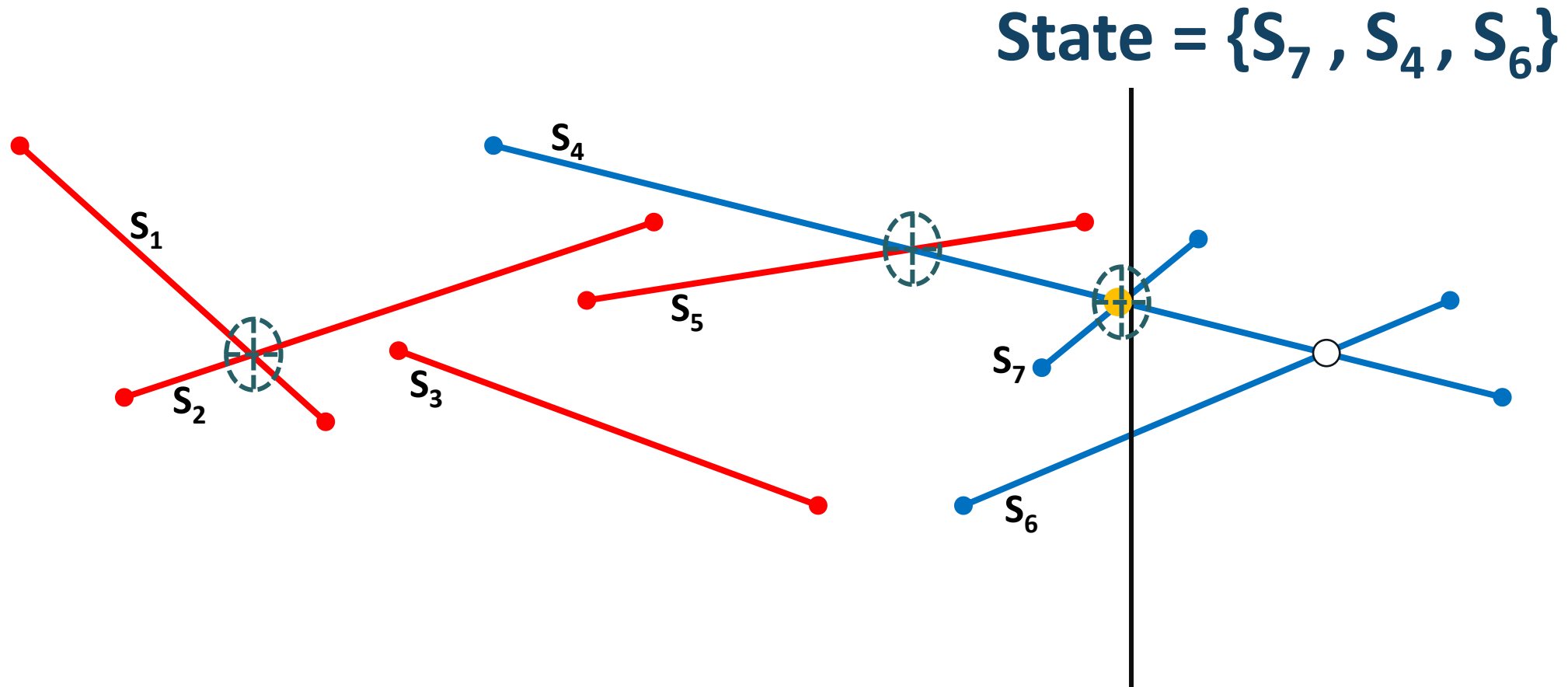
Exemple



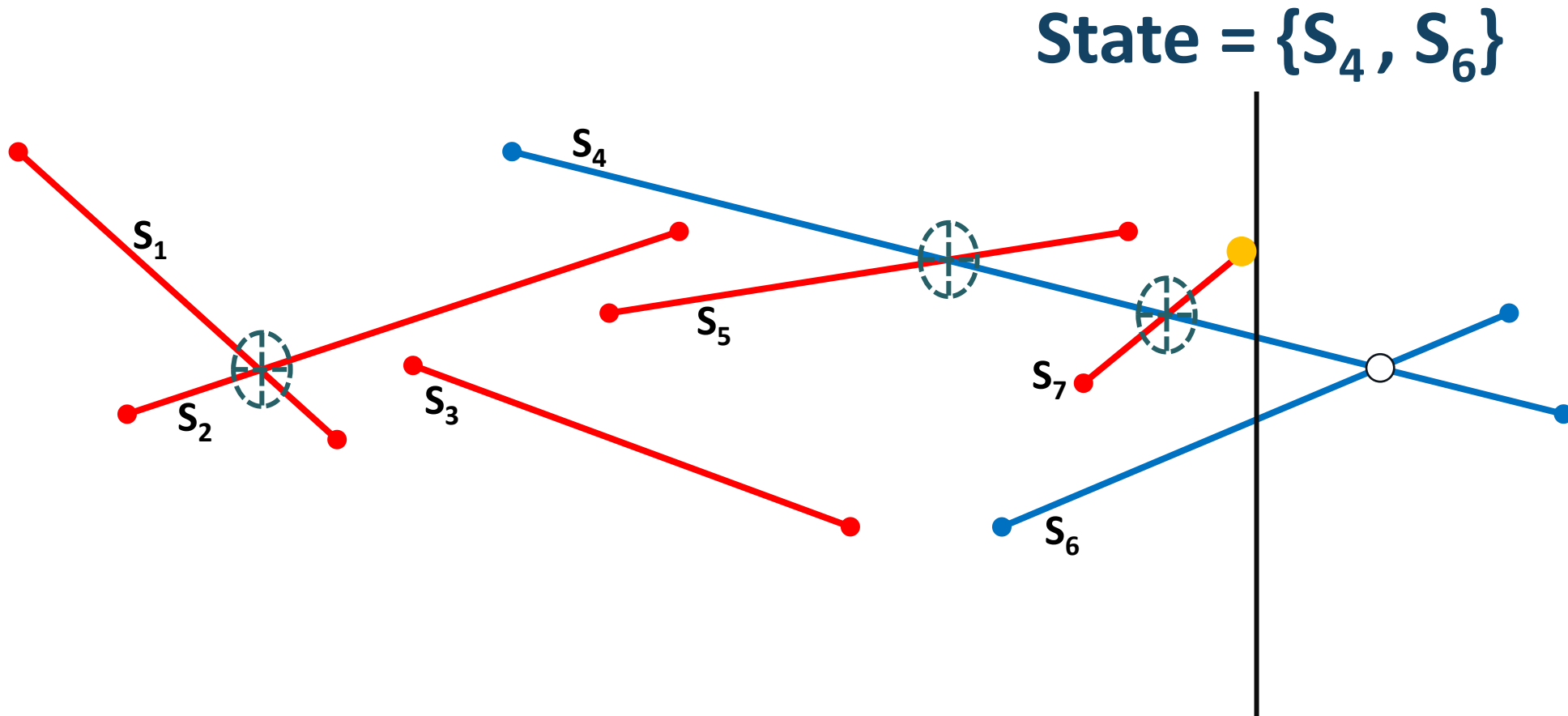
Exemple



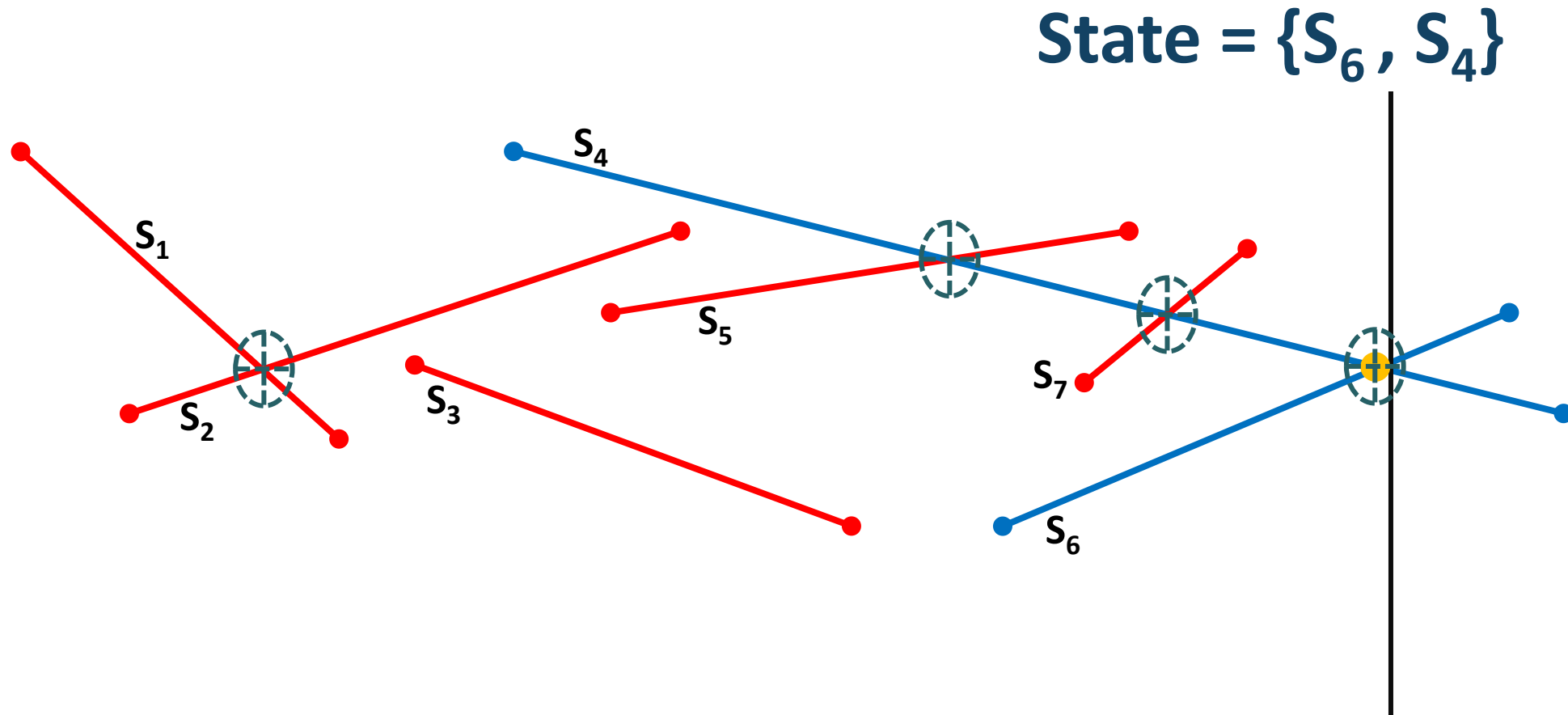
Exemple



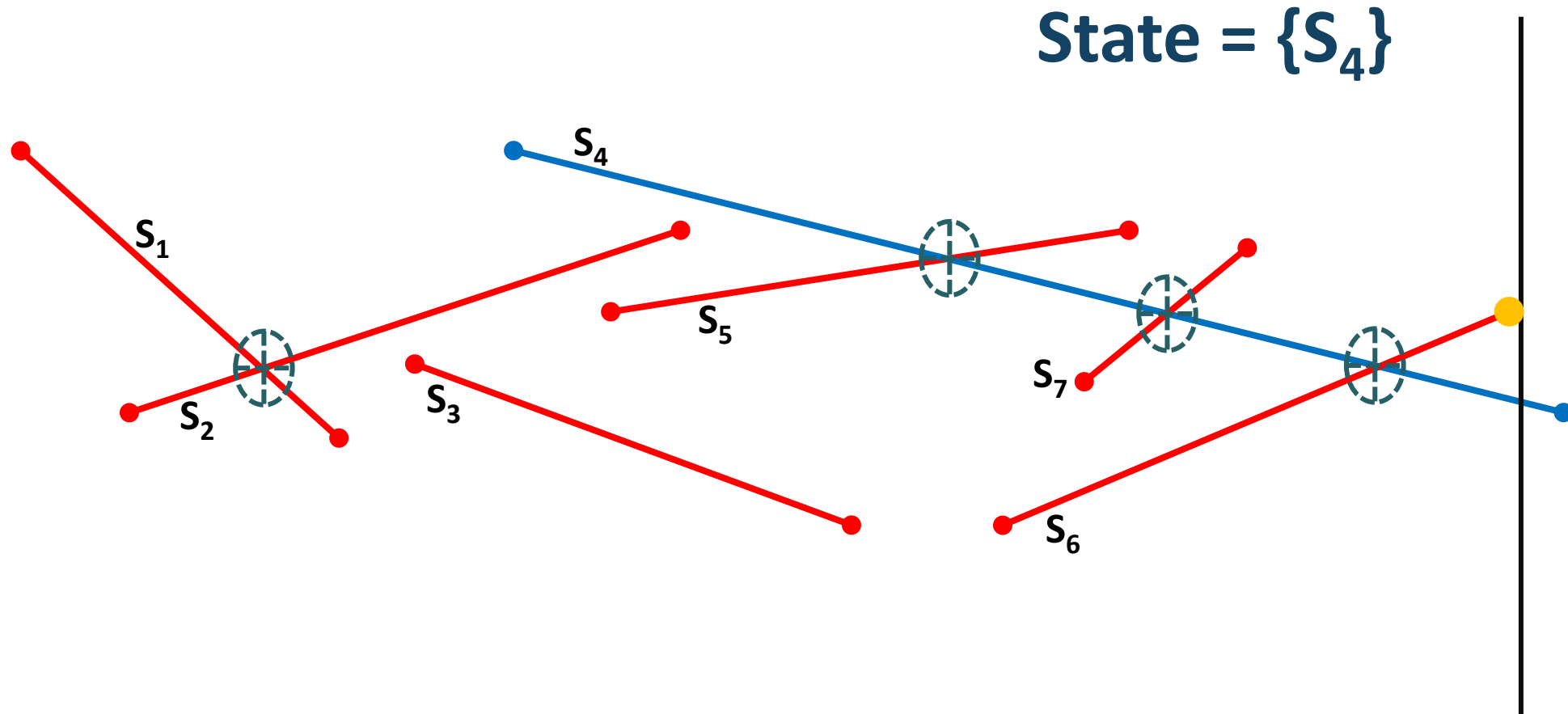
Exemple



Exemple

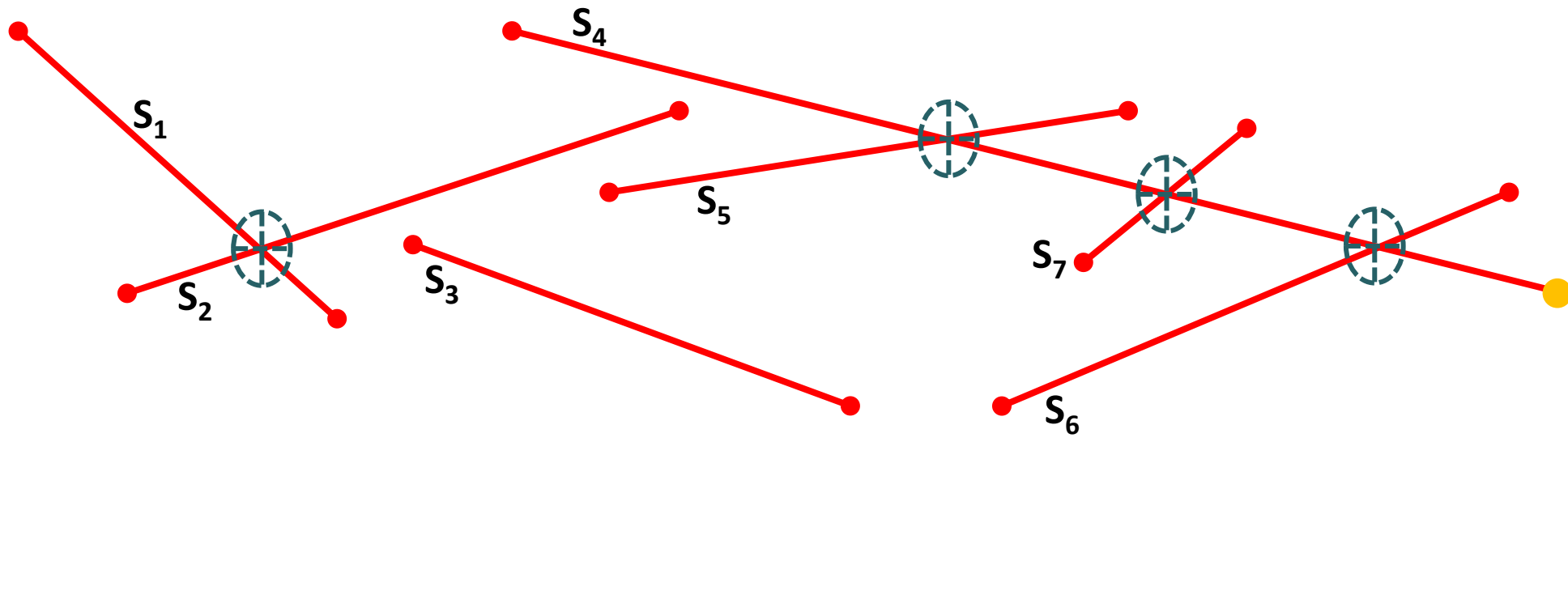


Exemple



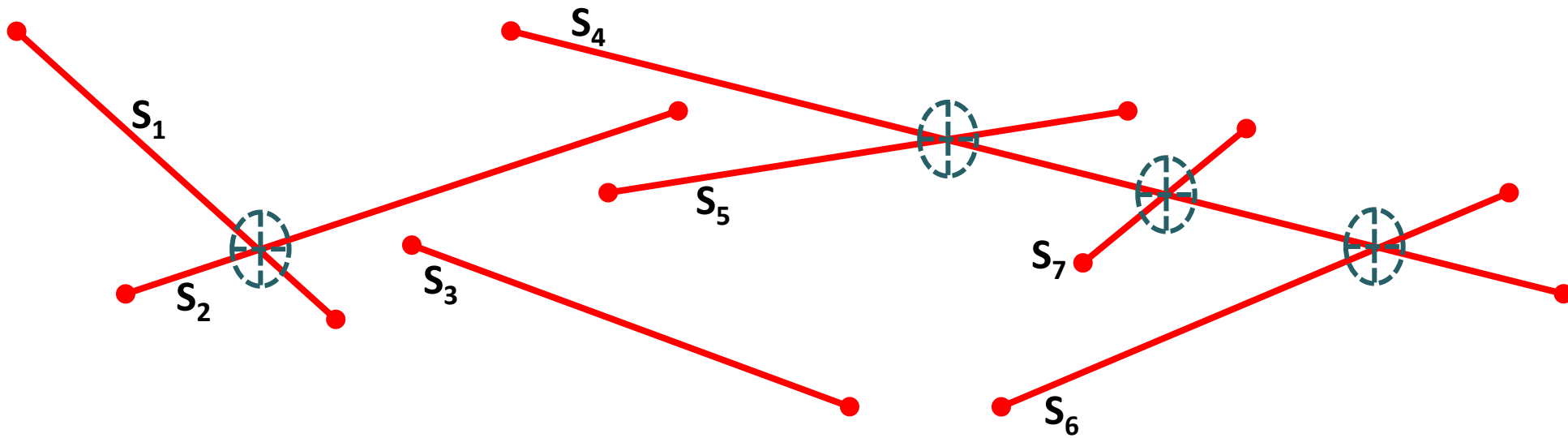
Exemple

State = {}



Exemple

State = {}

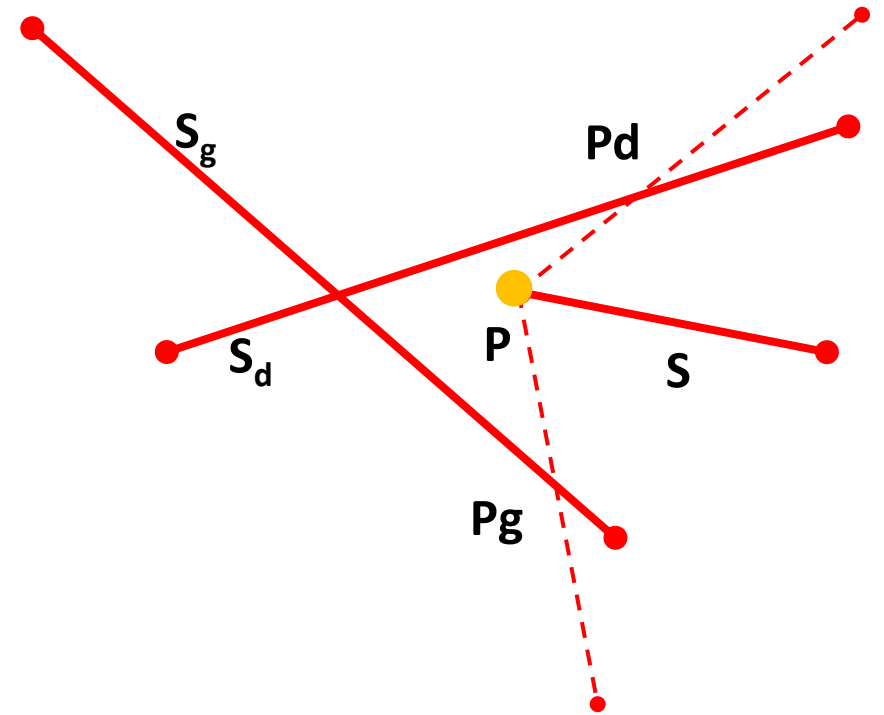


Formulation de l'algorithme

- Trier les extrémités et les mettre dans **Event**
- **State** $\leftarrow \{\}$
- Tant que **Event** n'est pas vide
 - $P \leftarrow \text{MIN}(\text{Event})$
 - Supprimer **P** de **Event**
 - Traitement selon le cas
- **Fin** Tant que

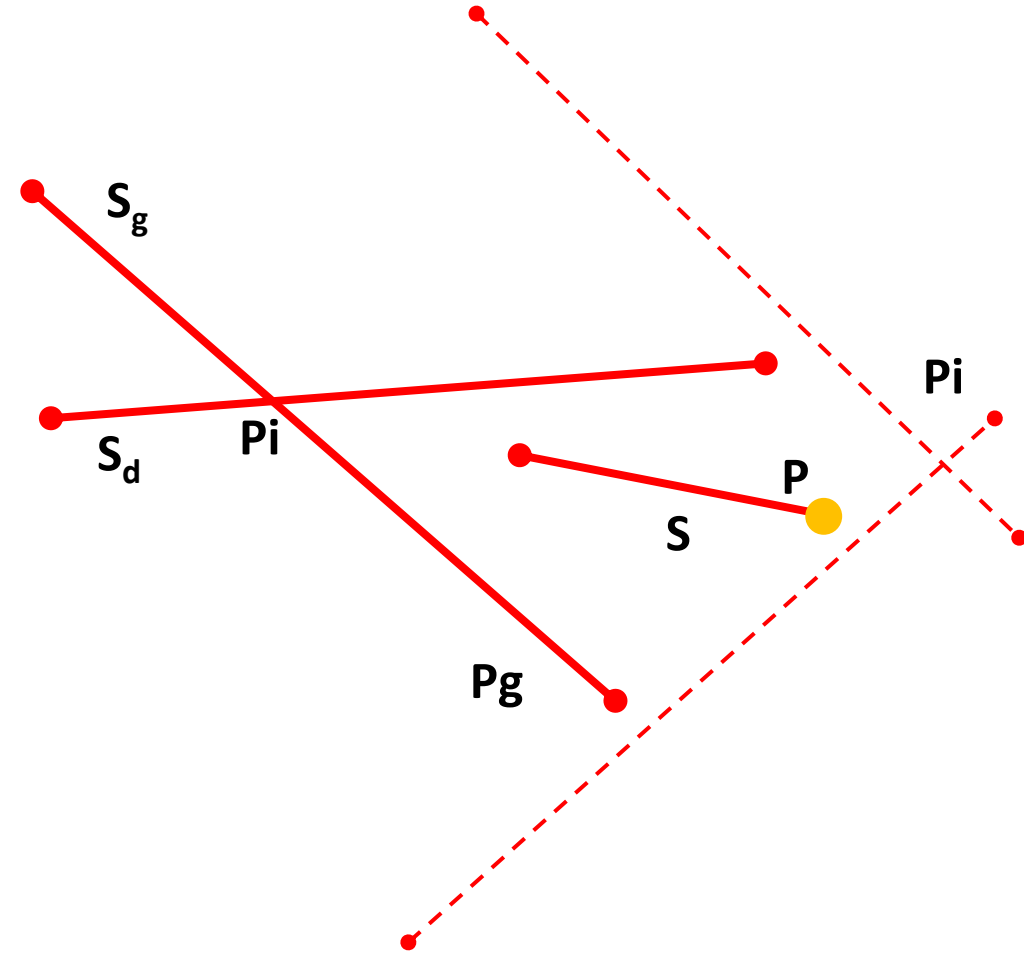
Traitement selon le cas

- **P** est début de **S**
 - **State** \leftarrow **State** + **S**
 - **S** \cap **S**_g = **P**_g , **S** \cap **S**_d = **P**_d
 - **Event** \leftarrow **Event** + {**P**_g , **P**_d }



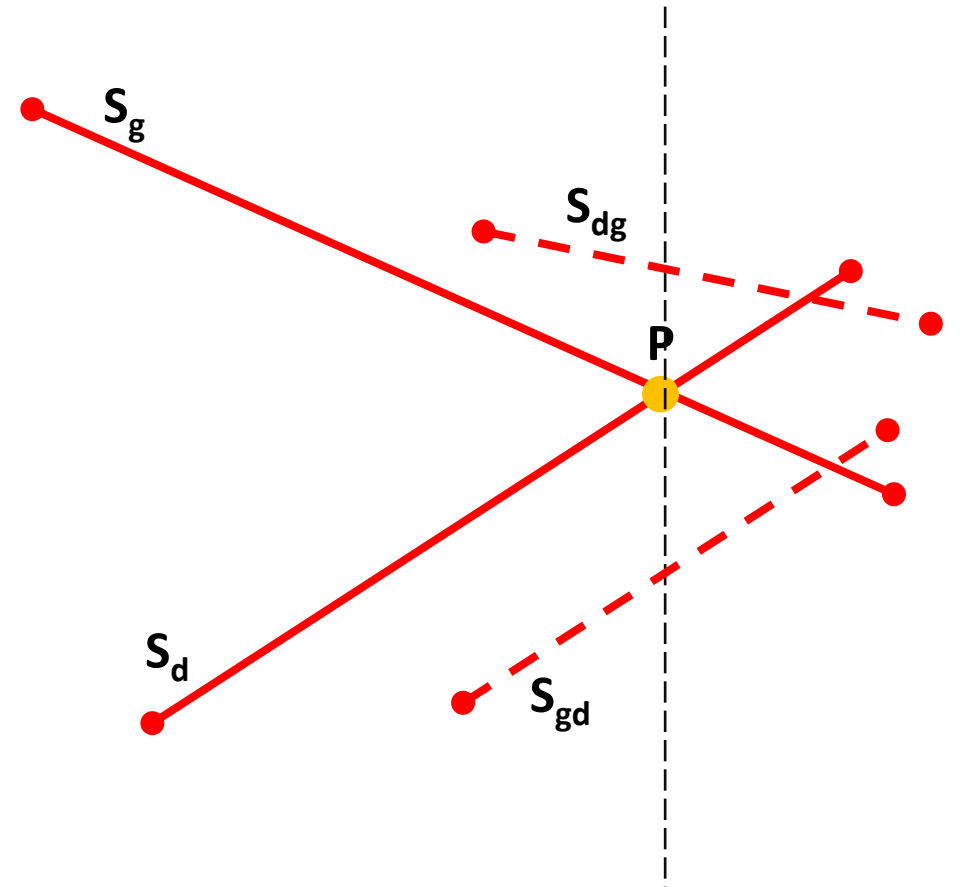
Traitement selon le cas

- **P est fin de S**
 - **State** \leftarrow State - {S}
 - **$S_g \cap S_d = P_i$**
 - si $x(P_i) \geq x(P)$
 - **Event** \leftarrow Event + { P_i }



Traitement selon le cas

- $P \leftarrow S_g \cap S_d$
 - Marquer **P**
 - Inverser S_g et S_d dans State
 - $S_{dg} \cap S_d = P_1$, $S_{gd} \cap S_g = P_2$
 - $\text{Event} \leftarrow \text{Event} + \{P_1, P_2\}$



La complexité $O(n \log n)$? ... $O(n^2 \log n)$

- Trier les extrémités dans Event
 - State $\leftarrow \{\}$
 - Tant que Event n'est pas vide
 - $P \leftarrow \text{MIN}(\text{Event})$
 - Supprimer P de Event
 - P est début de S
 - State $\leftarrow \text{State} + S$
 - $S \cap S_g = P_g, S \cap S_d = P_d$
 - Event $\leftarrow \text{Event} + \{P_g, P_d\}$
 - P est fin de S
 - State $\leftarrow \text{State} - \{S\}$
 - $S_g \cap S_d = P_i$
 - si $x(P_i) \geq x(P)$
 - Event $\leftarrow \text{Event} + \{P_i\}$
 - $P \leftarrow S_g \cap S_d$
 - Marquer P
 - Inverser S_g et S_d dans State
 - $S_{dg} \cap S_d = P_1, S_{gd} \cap S_g = P_2$
 - Event $\leftarrow \text{Event} + \{P_1, P_2\}$
- Fin Tant que

La complexité $O(n \log n)$? ... $O(n^2 \log n)$

• Trier les extrémités de S par ordre croissant de x (Fin Tant que)

• State $\leftarrow \{\}$

• Tant que Event n'est pas vide (Fin Tant que)

• $P \leftarrow \text{MIN}(\text{Event})$

• Supprimer P de Event

• P est début de S

• State $\leftarrow \text{State} + S$

• $S \cap S_g = P_g, S \cap S_d = P_d$ $O(\log k)$

• Event $\leftarrow \text{Event} + \{P_g, P_d\}$

$n + m$ (nbre d'évènements)

m : nbre d'intersections

$k = n+m$

$O((n+m) \log(n+m)) = O((n+m) + \log(n))$

Le cas le pire : $O(n^2 \log n)$

• $S_{dg} \cap S_d = P_1, S_{gd} \cap S_g = P_2$

• Event $\leftarrow \text{Event} + \{P_1, P_2\}$ $O(\log k)$

• Fin Tant que

$O(\log k)$