Kraków, 4.11.2015r.

Geometria obliczeniowa

Ćwiczenie trzecie

Piotr Oramus

1. Program do wizualizacji wyników oparty jest na bibliotece *graphics.py*. Dodawanie odcinków możliwe jest przez wczytywanie pliku w formacie, w którym każda linia reprezentuje jeden odcinek:

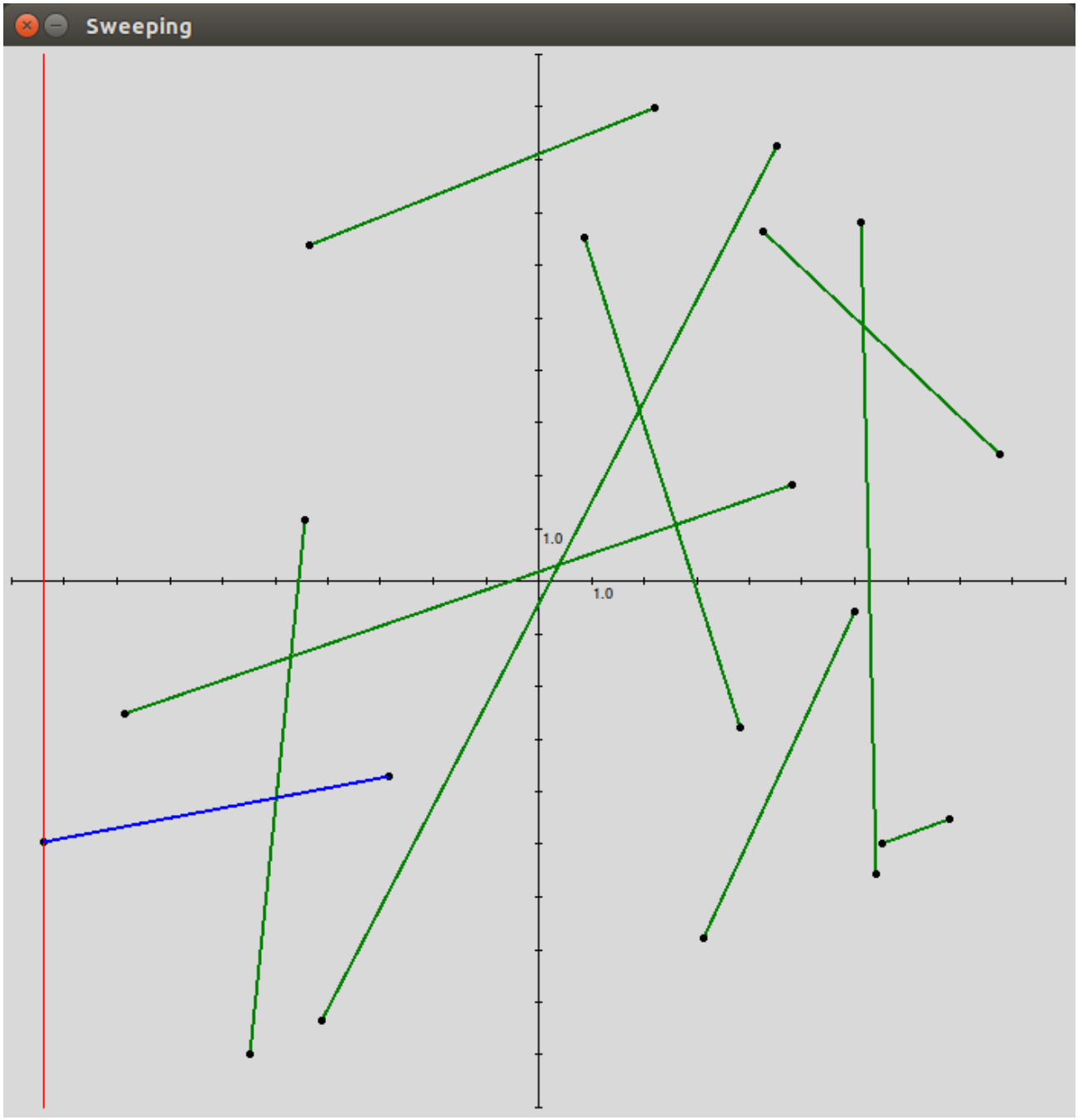
   x11 y11 x12 y12
   x21 y21 x22 y22
   …

2. Wersja algorytmu wykrywającego wszystkie przecięcia odcinków:

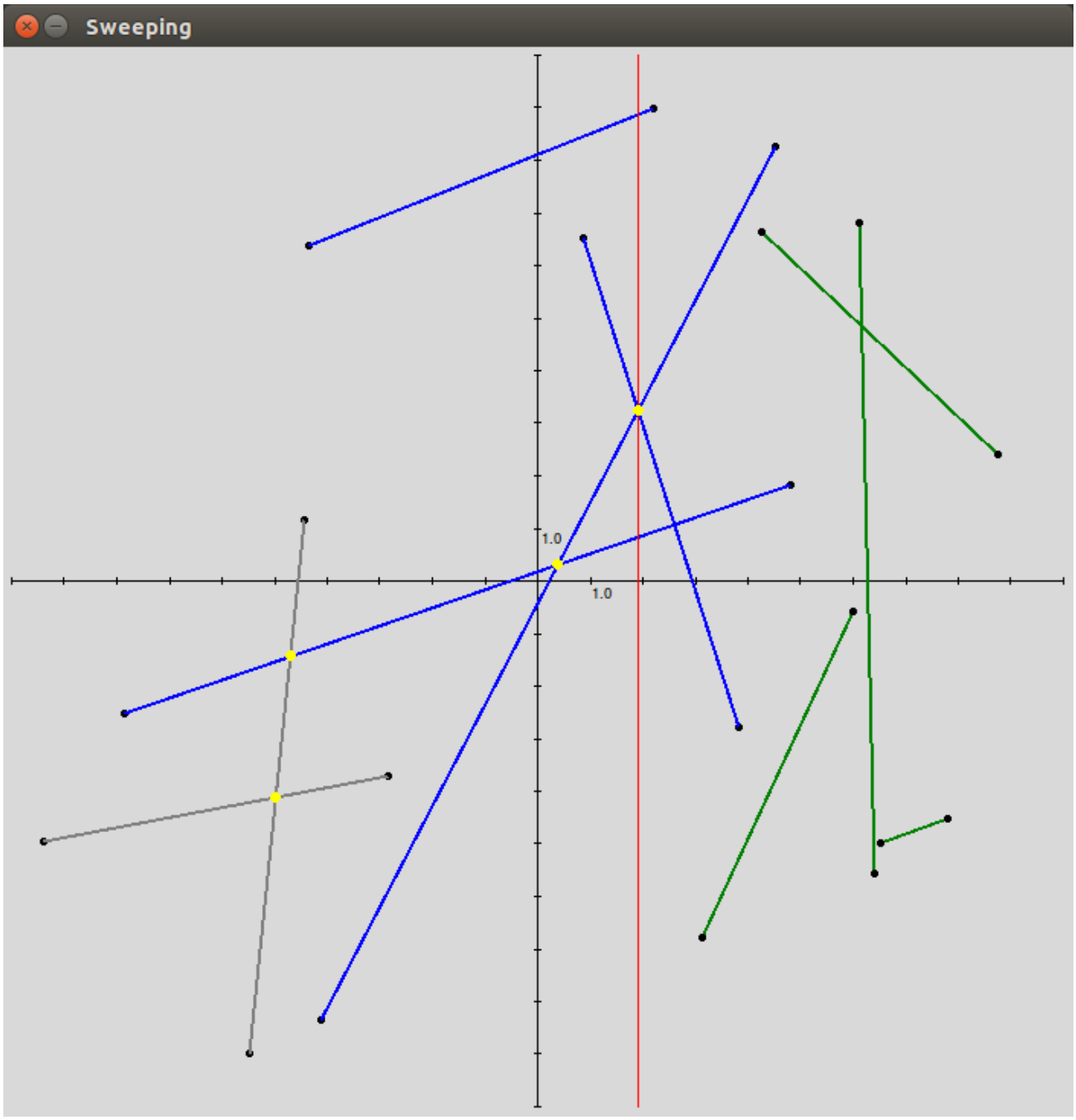
   Stan miotły jest klasą, która przechowuje aktualną pozycję *x*, aktywne odcinki w drzewie AVL oraz listę wszysktich odcinków w celu ułatwienia implementacji. Udostępnia metody, które pozwalają na wstawienie nowego aktywnego odcinka, usunięcie go oraz zamienienie ze sobą dwóch odcinków. Każda z tych metod zwraca nowo utworzone pary sąsiadów, jeżeli takowe istnieją.

   Struktura zdarzeń jest klasą zawierającą informację o typie zdarzenia (START_SEGMENT, INTERSECTION, SEGMENT_END), punkcie wystąpienia zdarzenia oraz odcinku lub odcinkach biorących udział w zdarzeniu. Eventy agregujące te informacje przechowywane są w kolejce priorytetowej, gdzie zdarzenia o najmniejszej współrzędnej x mają największy priorytet.

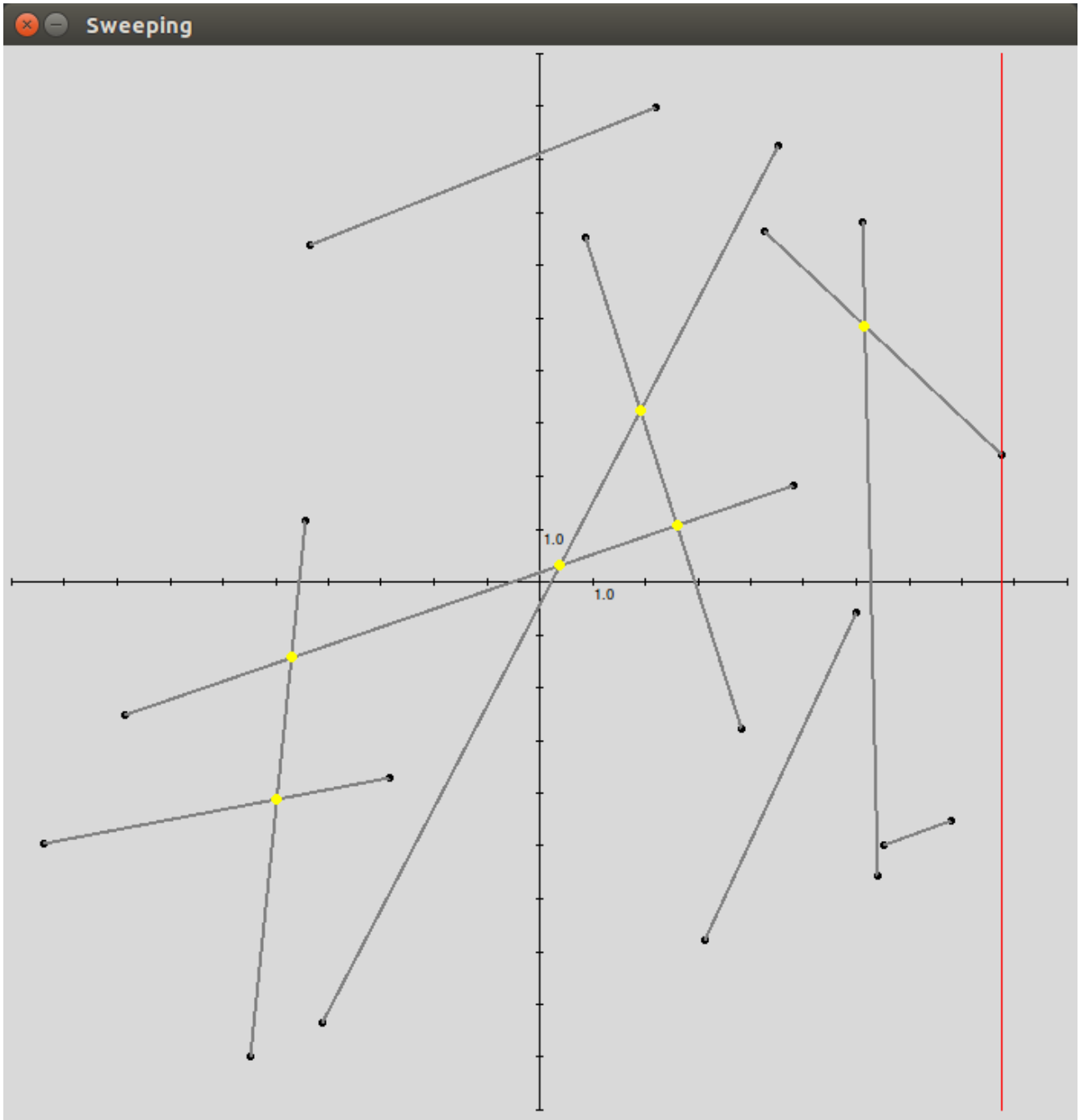
3. Przykład działania algorytmu (wizualizacja)



Rysunek 1: Algorytm na początku działania. Odcinki aktywne - niebieskie. Odcinki oczekujące - zielone.



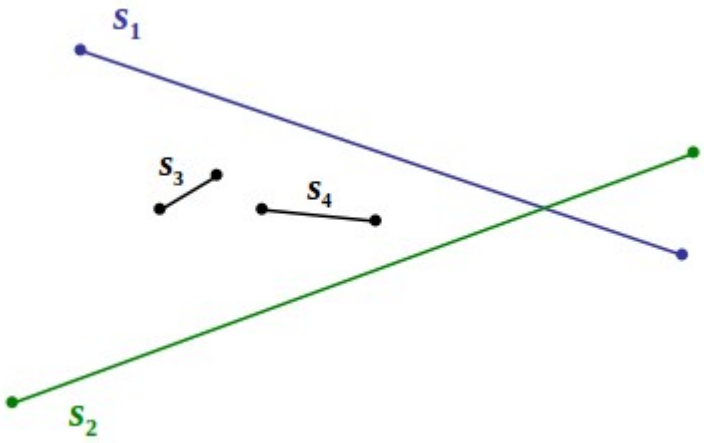
Rysunek 2: Algorytm w trakcie działania. Odcinki aktywne - niebieskie. Odcinki oczekujące - zielone. Odcinki przetworzone - szare. Wykryte przecięcia - żółte punkty.



Rysunek 3: Algorytm po zakończonym działaniu. Odcinki przetworzone - szare. Wykryte przecięcia - żółte punkty.

4. Wykrywanie przecięć więcej niż jeden raz.

Taka sytuacja jest możliwa np dla danego układu odcinków:



Wtedy przecięcie odcinków *s1* oraz *s2* będzie wykrywane:
- po dodaniu lewego końca *s1*
- po usunięciu prawego końca *s3*
- po usunięciu prawego końca *s4*

Zaimplementowany program sprawdza, czy wykryte przecięcie już istnieje w zbiorze rozwiązań – jeżeli tak, nie jest dodawane kolejny raz.